MY BULLETING: MANAGE STUDENTS' ATTENDANCE TRACKING WITHOUT PAPERS

FACULTAD DE INFORMÁTICA,

UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo de Fin de Grado en el Grado de Ingeniería Informática

Junio 2019

Estudiantes:

YEVHENII LASHUTKO

BARTLOMIEJ KOZIEL

Director:

Iván Martínez Ortiz

Resumen en castellano

MyBulletin es un servicio diseñado para facilitar el trabajo diario de los docentes para gestionar la asistencia a las actividades docentes. El sistema ha sido creado como una herramienta fácil de utilizar y a la vez efectiva mediante el uso de la tecnología disponible en un teléfono inteligente. Combina las ventajas de movilidad ofrecidas por un teléfono inteligente compatible con la tecnología NFC, así como la comodidad de realizar la administración de las tareas desde una aplicación web. Toda la información se sincroniza entre los dispositivos de manera automática para asegurar la integridad de la fiabilidad y la facilidad de trabajo con la aplicación.

La aplicación está compuesta por teres módulos que trabajan de manera colaborativa. Los módulos son: una aplicación web, una aplicación móvil nativa y un servidor que gestiona y almacena la información.

Un docente que quiera gestionar la presencia de los estudiantes sólo necesita de un teléfono inteligente y de una tarjeta NFC. El teléfono detecta la proximidad de la tarjeta NFC. En este caso, los alumnos pasan la tarjeta NFC en el teléfono y automáticamente el alumno es añadido a la lista de asistencia. Estas listas son almacenadas y a partir de ellas algunas métricas relativas a la asistencia son calculadas para cada alumno.

Los estudiantes se organizan en diferentes grupos que son asignados asignaturas. Cada asignatura tiene sus propias actividades, que pueden requerir controlar la asistencia. La aplicación permite gestionar todos estos aspectos y permite añadir nuevas actividades de manera simple.

Palabras clave

Android, Laravel, asistencia, NFC, VUE, API REST.

Summary

MyBulletin is a service designed to facilitate the daily work of professors for tracking students' attendance. There was created as an easy-to-use and at the same time very effective and original application by using the available technologies in a smartphone. It combines the mobility advantages of a smartphone equipped with an NFC reader, and ease of data administration through the website. Everything is synchronized and automated to ensure reliability and ease of work with the application.

The application itself consists of three parts, three separate systems that work together. These are: a website, a mobile application and a server for managing and storing data.

A professor who wants to check his presence is enough for him to have a mobile phone with a valid NFC. The phone detects the NFC cards that are next to it. This way, each student who tap the card to the phone is added to the attendance list. At any time, the professor can add this list to the list of students assigned to her. Lists are saved, and some attendance metrics of each student is calculated.

Students are divided into groups that are assigned to courses. Each course has its own activities, which may have a list of presence. The application makes it easy to manage everything and adding an attendance list is very simple.

Keywords

Android, Laravel, attendance, NFC, VUE, REST API

Table of Contents

Table of figures

Chapter 1 - Introduction

Introduction and motivation

The advances on computer and mobile technologies has led to the simplification for everyday tasks. Mobile note taking, sending messages, searching for information on the Internet or filling in any form, etc. In addition, nowadays it is quite common to consume online services using multiple devices [1] and even the use a particular device depending on the needs or even the time of the day [2]. Hence, successful online services and social networks provide the same or quite similar user experience in different devices or platforms.

Currently the attendance tracking it is traditionally done on paper, that is, the teacher must carry a paper-based attendance list that is passed around the class and that the students must sign for every activity / class. In particular, this is the usual practice at Facultad de Informática of Complutense University of Madrid. However, this approach has some limitations:

- Teachers have to carry an imprint attendance list of their courses with the risk of being forgotten or lost.
- Teachers have to transcript the paper-based list to calculate the attendance statistics with the risk of making mistakes during transcription the process.
- Because the list is usually passed around to simplify the signing process, it is not too difficult to cheat.

Considering these limitations, the general objective of this project has been the development software application that help teachers with students’ attendance tracking on their courses. Particularly the resulting solution has to allow teachers to use their smartphone or laptop/PC to track the students’ attendance, although the functionality provided in each platform may not be the same. Smartphone should be the default or primary device for track students’ attendance and laptop/PC should be considered the primary device to do the administrative tasks (students’ enrollment, prepare course activity schedule, etc.)

Purpose, specific objectives and scope of work

Based on identified issues and general aim of the project described in the previous section, the specific objectives of this project are:

1. Facilitate the attendance tracking process by using NFC cards. The usage or compatibility with UCM's Tarjeta Universitaria Inteligente (TUI) should be considered.
2. Develop an Android application that allow the teacher to track the student attendance by using NFC cards. The Android application must work in an offline setting if the teacher has previously installed and configured the app.
3. Facilitate administrative tasks (course creation, students' enrollment, etc.) in a web application that is intended to be used mainly in a laptop / PC. The application must serve as a backup plan for tracking student attendance
4. The software products created as a result of this projects will be distributed under an Open Source license and available in a public repository.

Project Work Plan

Development methodology

During the project we have followed an agile approach based on Scrum but not following strictly due to the limited available time (more details below). We had weekly meetings with the project's tutor that was playing the role of Product Owner as a key stakeholder of the final solution, but also partially the role of Scrum Master acting as a facilitator to address the identified issues during the previous week.

During the whole project the aim has been to have usable prototypes of the different software components in order to facilitate the discussion during the weekly meetings.

Project schedule and milestones

This project has been developed from February to June 2019 as part of the Erasmus scholarship at UCM's Facultad de Informatica. Due to the limited available time to develop the project, we had to keep a constant balance between functionality, technical feasibility and time prioritizing the work, in order to have a functional and usable of the whole solution.

The main phases of the project have been:

February: Clarification of the intended problem to solve, definition of the projects' scope and functionality of the whole project identifying the "must to have" and "nice to have" features. Because during the early meetings was identified the requirement of using a native Android application in order to use a NFC card as a means of students' identification, we had to learn how to develop Android applications (we had no previous experience) and how the NFC technology works in general and how can be used inside the Android platform.

March-April: Main development effort of the identified "must to have" features. During early March some technical proof-of-concepts where required to be developed in order to verify the technical feasibility of some of the functionalities like using the UCM's TUI card (more details on next chapter). As a result, a complete (from features point of view) of the android native application, REST API and administrative web application was available.

May-June: Through testing and polishing of the final core solution. Moreover, during this period we try to implement some of the "nice to have" identified features. In addition, during this period we write down this report and prepared the final public presentation.

Chapter 2 - Introducción

Introducción y motivación

Los avances en informática y en las tecnologías móviles ha propiciado la simplificación de las tareas cotidianas de la vida. Desde tomar notas en el móvil, enviar mensajes, a la búsqueda de información en Internet, rellenar formularios, etc. Además, en la actualidad también es muy habitual utilizar los servicios de la nube desde múltiples dispositivos [1] e incluso existe una predilección en el uso de un dispositivo concreto dependiendo del momento del día [2]. Por tanto, los servicios en línea o redes sociales que quieran ser exitosos deben tener en cuenta este requisito adicional de diseñar una experiencia de usuario adaptada a los múltiples dispositivos que puedan utilizar sus potenciales usuarios.

En la actualidad la gestión de la asistencia a clase se lleva a cabo normalmente mediante papel, es decir, el profesor tiene que confeccionar, imprimir y llevar a clase la lista de asistencia que suele pasarse entre los estudiantes donde estos deben firmar cada día por cada actividad o clase. En particular, esta es la práctica habitual entre los docentes de la Facultad de Informática de la Universidad Complutense de Madrid. Sin embargo, esta aproximación tiene limitaciones:

- Los docentes deben imprimir, transportar y custodiar las hojas de asistencia, en el riesgo de que se le olviden o las pierdan.
- Los docentes deben transcribir las listas en papel para calcular las estadísticas de asistencia, con el riesgo de cometer errores durante la transcripción.
- Además, como la lista se pasa entre los estudiantes para simplificar el proceso, no es demasiado complejo hacer alguna trampa al firmar en nombre de otro estudiante.

Considerando estas limitaciones, el objetivo general de este proyecto ha sido el desarrollo de una aplicación informática que ayude a los docentes a gestionar asistencia de los alumnos a las diferentes actividades educativas de una asignatura. En particular, la solución resultante debe permitir a los docentes utilizar su propio teléfono inteligente o su PC o portátil para gestionar la asistencia, aunque la funcionalidad proporcionada en las diferentes plataformas deba de ser exactamente la misma. Más concretamente, el teléfono inteligente debería de ser el dispositivo principal para pasar lista durante las actividades y el portátil / PC debería de ser el dispositivo principal para realizar las gestiones administrativas (asignación de alumnos a los cursos / grupos, preparar con antelación las actividades que comprenden el curso, etc.).

Propósito, objetivos específicos y alcance del trabajo

Basándonos en los problemas identificados y el objetivo general propuesto en la sección previa, los objetivos concretos del proyecto son:

1. Facilitar la gestión de la asistencia a las actividades docentes utilizando tarjetas NFC. La compatibilidad con la Tarjeta Universitaria Inteligente (TUI) debe ser contemplada.

2. Desarrollar una aplicación Android que permite al docente pasar lista en las actividades mediante el uso de la TUI que tienen los estudiantes. La aplicación Android debe ser capaz de trabajar en un escenario semi-desconectado o desconectado una vez que el profesor se haya instalado y accedido a la aplicación al menos una vez.

3. Facilitar las tareas administrativas (creación de cursos, asignación de estudiantes a cursos, etc.) a través de una aplicación web pensada para ser utilizada en un portátil o PC. La aplicación también debe poder utilizarse como plan alternativo para pasar lista.

4. Las aplicaciones resultantes serán distribuidas bajo una licencia Open Source y disponible en un repositorio público.

Plan de trabajo

Metodología de desarrollo

Durante el proyecto se ha seguido una metodología ágil basada en Scrum pero no de manera exhaustiva y estricta debido al limitado tiempo (más detalles a continuación). Hemos tenido reuniones semanales con el tutor del proyecto que ha actuado como *Product Owner* y como cliente de la solución final resultante, aunque también ha realizado parcialmente el papel de *Scrum*

Master actuando como facilitador para abordar problemas encontrados o identificar posibles problemas durante el desarrollo del proyecto.

Durante todo el proyecto se ha intentado tener semanalmente prototipos ejecutables de los diferentes componentes que comprenden la solución completa, con el objetivo de facilitar las discusiones durante las reuniones semanales.

Cronograma e hitos

Este proyecto se ha desarrollado desde febrero a junio de 2019 como parte de la beca Erasmus realizada en la Facultad de Informática de la UCM. Debido a la limitación de tiempo disponible para desarrollar el proyecto, durante todo el proyecto ha sido necesario mantener un equilibrio entre funcionalidad de la aplicación, las posibilidades técnicas y el tiempo disponible. Se ha intentado priorizar el trabajo con el objetivo de tener una solución completa funcional a costa de tener que simplificar algunos aspectos del proyecto que pueden realizarse como extensión futura al trabajo actual.

Las principales fases del proyecto han sido:

Febrero. Clarificación del problema real a resolver, definición del alcance del proyecto y de las funcionalidades “necesarias” y “deseables”. Durante las reuniones iniciales se identificó el requisito de utilizar una aplicación nativa Android con el objetivo de poder utilizar la TUI de los estudiantes como mecanismo de identificación a la hora de pasar lista, de modo que fue necesario aprender a desarrollar aplicaciones Android (ya que no teníamos experiencia previa) y cómo funciona la tecnología NFC en general y más concretamente cómo puede ser utilizada en un dispositivo Android.

Marzo-Abril. Se llevó a cabo el principal esfuerzo de desarrollo de las características identificadas como “necesarias”. Al comienzo de marzo se desarrollaron pequeños prototipos de concepto para verificar la viabilidad técnica de algunas de las características propuestas como el uso de la TUI (más detalles en el siguiente capítulo). Como resultado de esta etapa se obtuvieron una solución completa (desde el punto de vista de características identificadas) de la aplicación nativa Android, de la API REST y de la aplicación de gestión.

Mayo-Junio. Se realizó un exhaustivo testeo de las aplicaciones. Además, durante este período también desarrollamos algunas de las características "deseables" de la aplicación. Además, durante esta fase también escribimos esta memoria y preparamos la presentación para la defensa pública.

Chapter 3 - Description of the Results and Discussion

My Bulletin

MyBulletin is a service designed to facilitate the daily work of university professors. There was created an easy-to-use and at the same time very effective and original application by using the available technologies. It combines the advantages of a smartphone equipped with an NFC reader, as well as mobility and ease of data administration through the website. Everything is synchronized and automated to ensure reliability and ease of work with the application. It is safe and allows you to safely store data, both shared, such as the student's first and last name, and confidential data.

The application itself consists of three parts, three separate systems that work together. These are: a website, a mobile application and a server for managing and storing data.

A professor who wants to check students’ presence only requires a mobile phone with NFC support. The phone detects the NFC cards that are next to it. The tracking of the attendance is done by tapping the student card to the phone, the applications identifies the student that is the owner of the card and it is automatically added to the attendance list. In addition to just the attendance list, some basic attendance statistics are calculated per course and per student, which can be reviewed using as teacher through the web management application.

Students are divided into groups that are assigned to courses. Each course has its own activities, which may have a required to track students’ presence. The application makes it easy to manage everything, particularly the management of adding new activities to track.

The name of the application (see Figure 1 Logo) suggests that this may be the professor’s electronic journal, which can replace the paper-based list that currently use professors at UCM’s Facultad de Informática, which need to be checked, sorted and archived.

Figure 1 Logo

Architecture / main components

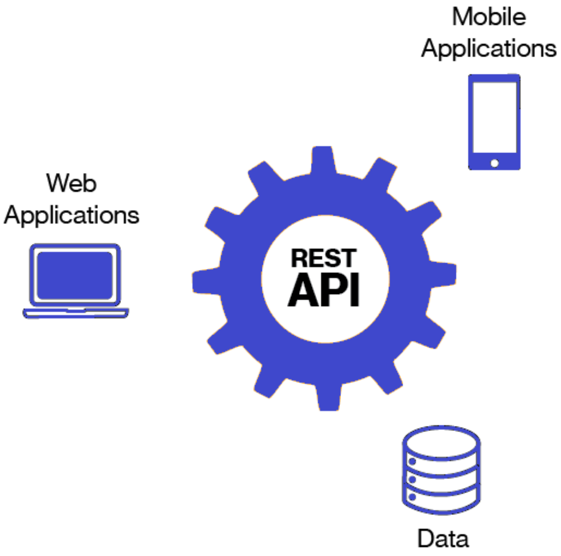


Figure 2 Component Diagram of the solution

The main elements of the solution are (see Figure 2 Component Diagram of the solution:

- Management web app

The main objective of this web application is to allow the professors to do a detailed and in-depth definition of their courses, assign students to the courses and define the educational activities that will be part of the course and require to track students' attendance. In addition, the views related to the administrative role allows an administrator to create new users' account for the system (professors and administrators). Moreover, administrators can define the students in the system, this way, it is possible to identify uniquely the same student across courses. Finally, professors can also view the attendance list from one subject and view the group rating of student attendance.

- Android Native application

The application is used by professors to manage attendance lists associated to an educational activity. A professor selects a course, and a group of students and finally an activity, then the application waits to detect cards with a built-in NFC chip, which are located next to the phone.

- Backend service REST API

The objective of this component is to provide a REST API endpoint that will be used by the two previous clients. Basically, the REST API provides a Create Read Update and Delete (CRUD) service and maintains the consistence and restrictions of application data model.

Following sections describe with more detail these 3 components.

VUE management application

The management application has been implemented as a Single Page Application (SPA) using the VUE framework. The application provides features for two different roles: professors and administrators.

The first role is the site administrator. This role has the following capabilities:

- Bulk addition of students, as well as editing and deleting selected students.
- Creation, editing and deleting professor accounts.
- Create an administrator account.

The second role is the professor. This role has these specific capabilities:

- Create and delete a course.
- Create, edit and delete groups of students.
- Create, edit and delete subjects.
- Add students to the group, as well as remove them from there.
- Creating an activity for one day and for a specified period, which this will be repeated more than once.
- Creating an activity for a particular course, as well as creating a unique activity.
- Adding students to the activity, as well as removing them from there.
- View attendance list.
- View the frequency of attendance to each student group as a percentage.

Tracking students' attendance using teachers' smartphone

At the beginning of the project the aim was to use UCM's identification card called Tarjeta Universitaria Inteligente (TUI) because students already have it, however we faced a huge problem associated with the MIFARE chip that is embedded in it. The MIFARE chip cannot be read by all smartphones, due to use some extensions to the NFC protocol that are proprietary and even so, most of the data stored in the card requires an encryption password to be read. After reviewing other alternatives to avoid the problems with the MIFARE chip, we finally decided to use other cards with an NFC chip to demonstrate this feature.

The mobile application on Android is available only for professors since they are the ones who collect student attendance. The developed application includes the following technological components and functions:

- Track students' attendance using NFC cards.
- Possible to use in a semi-disconnected mode. Automatic synchronization of phone files with the backend.
- Convenient authorization mechanism for the mobile application.

During the first authorization, a token is created in the application that links the phone with this account and with the server. This token replaces and hides the usual method of authorization with a login and password. After successful authorization, all account data will be downloaded to the phone, which will help to use the applications without connecting to the Internet. But when you first connect to the Internet, all data will be uploaded to the server by creating a single version both on the phone and on the server itself.

Backend REST API

The Laravel framework allows a developer to creat a REST API that follows the CRUD [3] (Create, Read, Update, Delete) model, that is created with resource controllers based on route Model Binding. A Laravel Controller is a class that include methods that are binded to different routes, for example: CoursesController classes and function index() in our project shows all courses assigned to professor. In Laravel a model class in Laravel used to segregate data from Database (each model is like table in Database). These model classes are extended using Laravel's

Resource classes, and they are used as Data Transfer Objects [4] for the REST API. These Resource and collection of Resources, hides the complexities and the large number of relationships between model classes, hence making the usage of the REST API more easier. For example, request (/courses) shows all courses with information about group and activities attached to this course. With one request we could get all data that we need and use it for example in native application.

All API resources are secured with JWT-auth token that is generated after user authentication. The authentication is done in the web-managed application by simply using a form with an email and password. In the native application the login procedure it is a two-step process firstly a professor login using email and password and secondly the professor has to use her UCM's TUI as a second authentication factor. Next time the professor opens the application needs only to use the TUI in order to get access to the application, this way if the professor is using a shared device and forgot to log out properly, the next user can not access to the data All API requests, except Login, need this token.

To create the Laravel backend we have used artisan command line tool that is provided as part of the Laravel SDK and let programmers to make: migration, controllers, seeders, resources, and models in easy and secure way. For example: "PHP artisan make:controller –resource CourseController" create Controller class with CRUD functions.

Figure 3 Backend Service UML Diagramshows a class diagram of the main elements that comprises the domain logic of the back-end API. It very same domain structure is used also in the native android application.

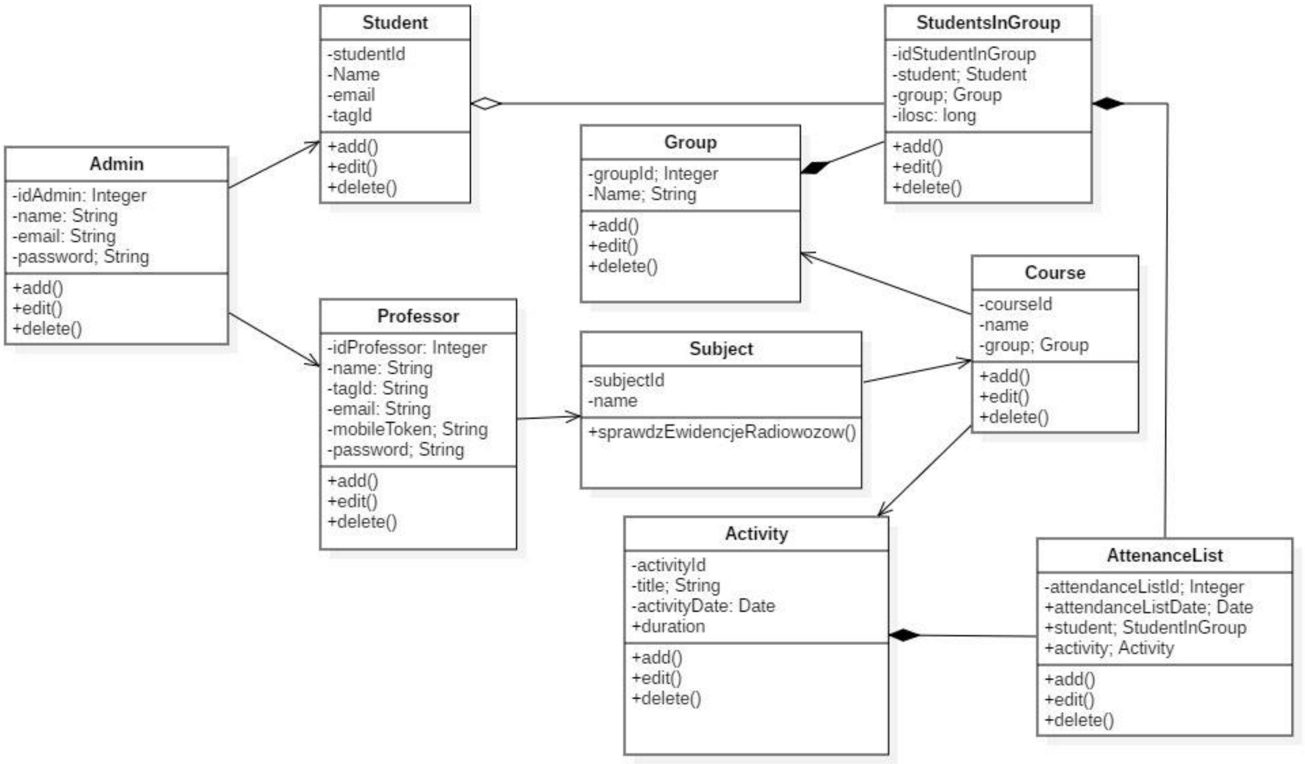


Figure 3 Backend Service UML Diagram

MyBulletig Data model

Figure 4 Backend Service data model depicts the data model of the backend service. This data model comprises the following elements:

- Administrator has the ability to create professor accounts, as well as add new students and manage information about them
- Professors has the ability to manage a range of functions, which are described below. It is this profile of our system that has the ability to authenticate to my application, as well as the creation of a token that replaces authentication.
- Subjects can take several professors to the same subject.
- Groups has the ability to admit students to his or her place, as well as to assign selected professors to a certain group.

- Courses have the ability to combine information about the subject and the group. And there is also a unique name for each course, which combines the name and number of the group through sign "/" .
- Students groups have the ability to store all the information about the whole group, as well as the professors attached to it.
- Students have a personal id card identifier, first and last name, and also have the ability to be marked in the attendance list.
- Activities are created for a certain day and date with a unique duration, as well as the opportunity to join the right group.
- Recurrences is a part of activity that allows you to do cyclic activities for every day or every week, where you can select the start and end of the activity
- Attendance list is created on the basis of the selected activity and the course assigned to it by a group of students.

These elements have the following dependencies. The administrator can add students and create accounts for professors. While the professor has the ability to create a subject, course, group, activities and attach student to group. There is no limit to the number of courses, and each course can consist of only one group and name many activities. Activity can only have one attendance list, but attendance list have (studentInGroup) and activity. StudentInGroup is a class that is relation between student and group.

This structure let us manage the attendance list application and check attendance.

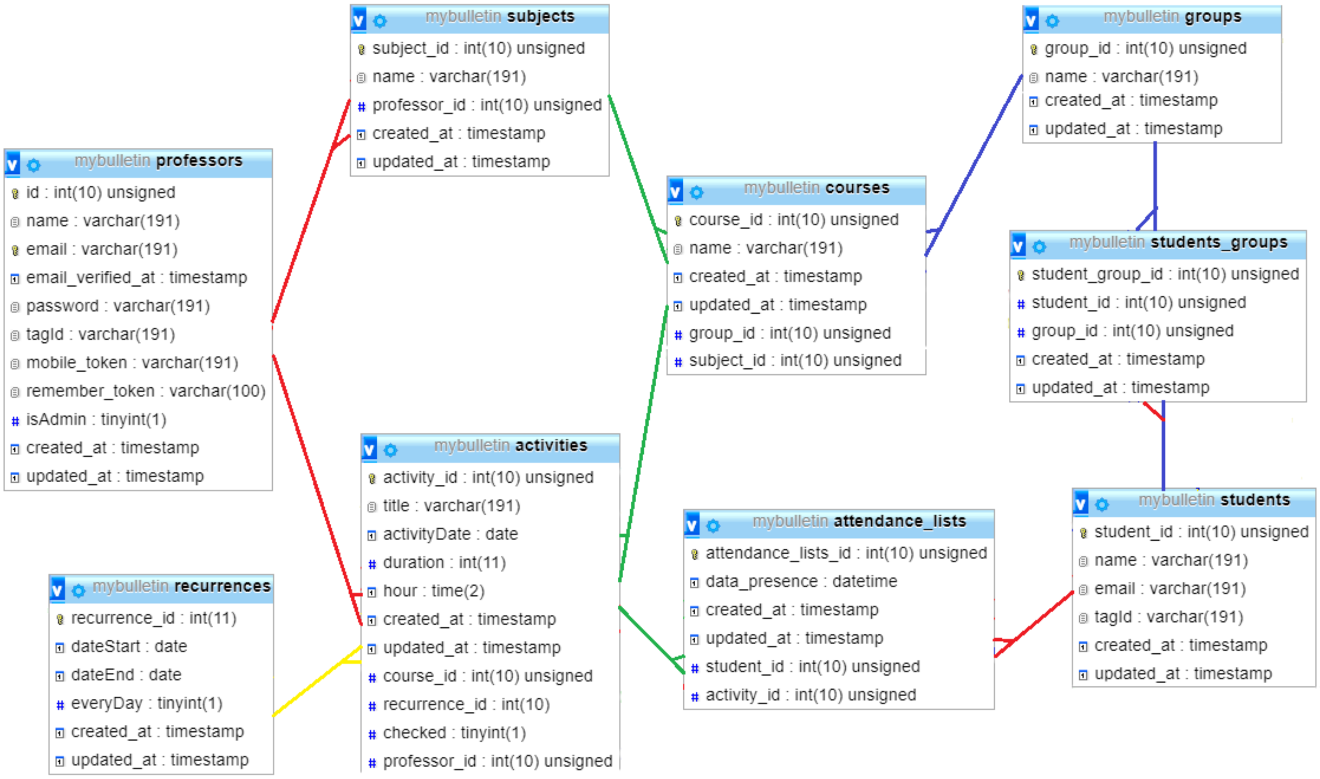


Figure 4 Backend Service data model

Technologies involved

This section provides a brief summary of the technologies and tools used during the development of the project, highlighting those that are more novel of that play a main role in the project.

Android Platform

The Android platform or simply Android is one of the most popular and fast developing operating systems and a platform in the world that is used as the main platform for most mobile devices, tablets and various systems. Platform of the Android is based on the Linux kernel and with Google's own Java Virtual Machine implementation [5]. The Android operating system runs on a virtual machine that performs the execution of bytecode commands. A bytecode is an intermediate representation in which a program can be translated.

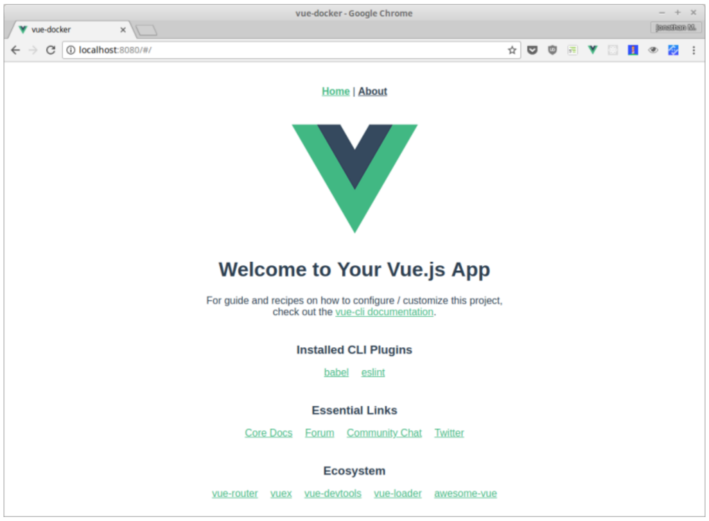
The undeniable advantages of the engine make Android currently one of the most popular engine and platform for mobile devices in the world. According to the website of the manufacturer, the global market share is about 85-86% [5][6] of users of devices with the Android operating system, which allows developers to earn good money in this market.

Android has its own integrated IDEA development environment for working with the Android platform called Android Studio. It contains layouts for creating an UI, where the work on the application begins, and also contains tools for developing solutions for smartphones and tablets and many other technological solutions. It is interesting to point that that Android Studio encourages the adoption of concept of continuous integration development process, which allows you to immediately detect existing problems. The development environment for the application has become really convenient even for novice developers. The main feature of this platform is the availability of its SDK emulator smartphone, which helps the developer to test his project without the need to download to his device.

The Android platform is unique in that on most mobile devices it is possible to get access to the NFC, RFID chips hardware since the earliest Android versions [7]. This is particularly interesting because IOS support for access the NFC hardware for non-Apple developers it is quite limited. On IOS devices, this feature appeared only in the latest models starting from 2018.

VUE

VUE is a progressive framework for building user interfaces. Unlike other monolithic frameworks, VUE is designed from the ground up to be incrementally adaptable. The core library is focused on the view layer only and is easy to pick up and integrate with other libraries or existing projects. VUE is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries. One of the greatest advantages of VUE is its small size. The size of this framework is 18–21KB and it takes no time for the user to download and use it. This does not mean that it has low speed because of small size. Instead, it beats all the bulky frameworks like React, Angular, and Ember.[8]



Home | About

Welcome to Your Vue.js App

For guide and recipes on how to configure / customize this project,
check out the vue-cli documentation.

Installed CLI Plugins

babel eslint

Essential Links

Core Docs Forum Community Chat Twitter

Ecosystem

vue-router vuex vue-devtools vue-loader awesome-vue

Figure 5 VUE interface (start page) [9]

VUE [10][11] is also popular among the web developers because it facilitates them to integrate with existing applications. This means that it is useful for developing new web applications as well as altering the pre-existing applications. This integration is possible because VUE has large set of already available UI components..

Advantages:

- Easy to understand and develop application.
- Simple integration
- Detailed documentation
- Flexibility
- Two-way communication
- Very small size
- Few restrictions
- Support
- Addresses the drawbacks of AngularJS and ReactJS

Disadvantages:

- Lack of support for large projects
- Lack of plugins

Laravel

Laravel is a PHP framework for web applications with expressive and simple syntax. [12] It will allow simplifying the solution of the main urgent tasks, such as authentication, routing, sessions and caching. Laravel is an attempt to combine all the best from other [13] PHP frameworks.

Laravel differs from other frameworks in that it uses the latest features of PHP. In fact, Laravel is suitable for all types of applications for which PHP can be used. But one of the characteristics that make Laravel unique among its competitors is the fact that compared to other PHP frameworks, it provides more opportunities for implementing backends for mobile applications. A Laravel application can be developed on any operating system. This framework consistently gets on the list of the most popular and used PHP frameworks and receives such titles as the best corporate leveling framework and the best personal project framework. Corporate-level projects occupy a leading position in the market, for example: rv.campingworld.com, startups.com, filmfed.com, etc.

Advantages:

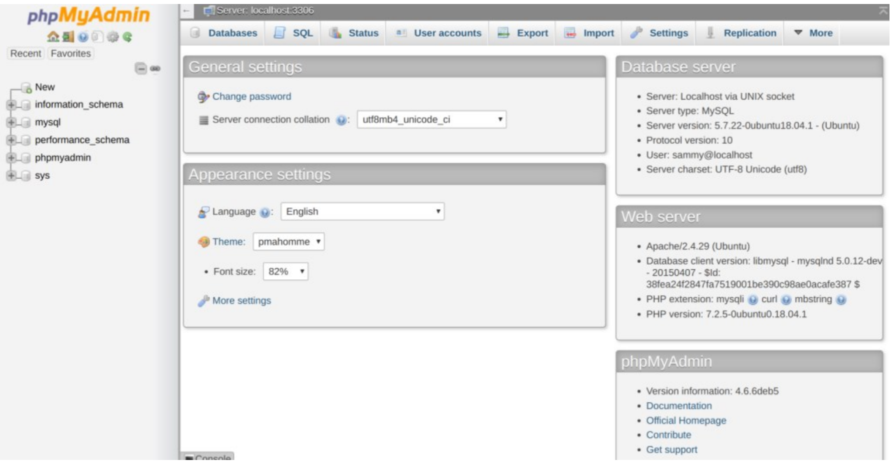
- The quick fixes for detected errors.
- Detailed documentation.
- Open source & powerful community
- Availability of the own engine for Blade templates
- Highly security
- Database migration support
- Object oriented libraries
- Easy unit testing
- Easy creating multi-lingual apps

Disadvantages:

- Speed and performance will be affected.
- To modify core behavior there will be a lack of options
- No smooth transition between versions

phpMyAdmin

phpMyAdmin is free and perhaps the most common MySQL management environment. [14] Since more than ninety percent of web resources run on content management systems (CMS) using the PHP language and are built on the principle of separating the graphical component of the interface and data, which are stored in separate files called SQL databases. Most of the problems of administrators and web site developers are solved by the phpMyAdmin server application, which presents databases in a conveniently readable form and allows you to manage them in one click, even without using any commands.



General settings

- Change password
- Server connection collation: utf8mb4_unicode_ci

Appearance settings

- Language: English
- Theme: pmahomme
- Font size: 82%
- More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MySQL
- Server version: 5.7.22-0ubuntu18.04.1 - (Ubuntu)
- Protocol version: 10
- User: sammy@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.29 (Ubuntu)
- Database client version: libmysql - mysqlnd 5.0.12-dev - 20150407 - $Id: 38fea24f2847fa7519001be390c98ae0acafe387 $
- PHP extension: mysqli curl mbstring
- PHP version: 7.2.5-0ubuntu0.18.04.1

phpMyAdmin

- Version information: 4.6.6deb5
- Documentation
- Official Homepage
- Contribute
- Get support

Figure 6 phpMyAdmin interface [15]

All operations with cells and their contents - viewing, deleting, pasting, copying are performed by one click. Most often, web developers and site administrators use the ability to instantly create a backup copy of the database to transfer the site from one hosting to another, there is an opportunity to get a copy in an archived form.

Advantages:

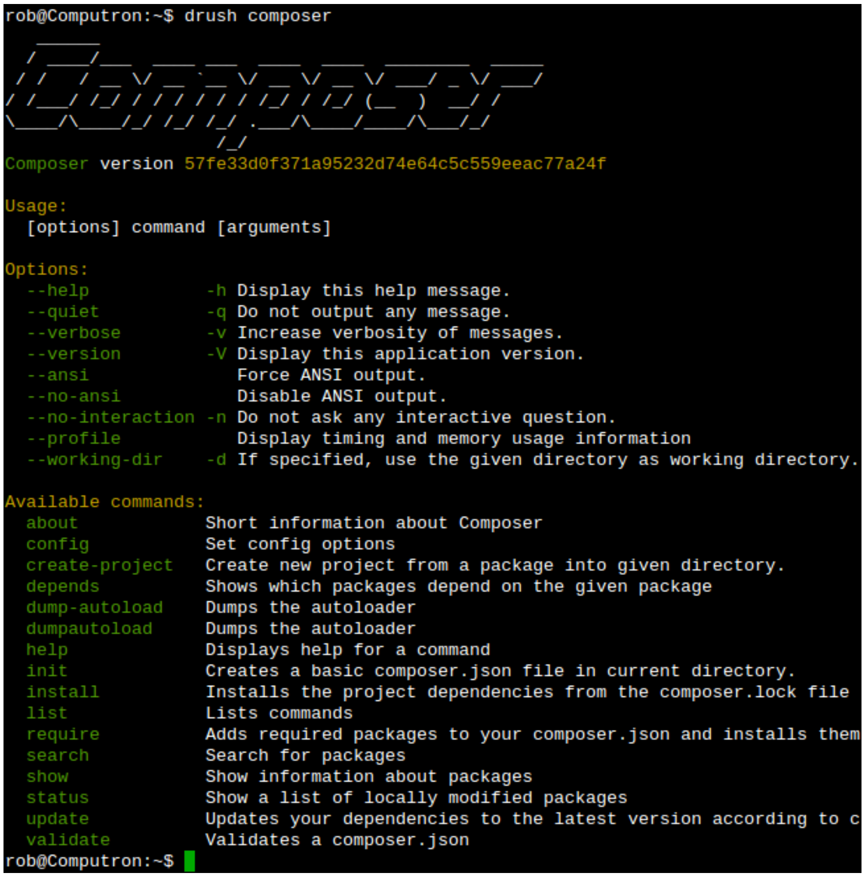
- Multi server administrations
- Easy data export
- Not very resource intensive
- Solid perfomance

Disadvantages:

- does not provide as complete a set of features as MySQL Workbench
- difficult for novice user

PHP Composer/Artisan Console

PHP Composer is the package manager for PHP [16]. It allows you to describe which libraries the project depends on and Composer will install the necessary libraries inside each project separately [17]. To work with Laravel fraemwork you need to install Artisan Console. Artisan is a command line interface that includes a set of useful commands for web application development. The main functionality of Artisan is, generation migration, publication of the package resources and many similar tasks. There is a list of pre-created commands to work with projects, but it is possible to create your own team for specific requests. What gives you new possibilities for working with PHP.



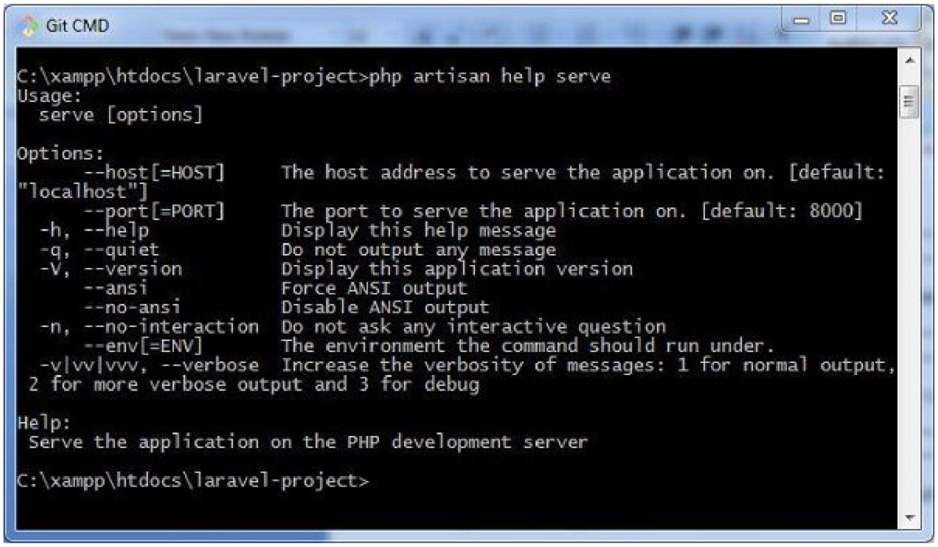
```
rob@Computron:~$ drush composer
   ______
  / ____/___  ____ ___  ____  ____  ________  _____
 / /   / __ \/ __ `__ \/ __ \/ __ \/ ___/ _ \/ ___/
/ /___/ /_/ / / / / / / /_/ / /_/ (__  )  __/ /
\____/\____/_/ /_/ /_/ .___/\____/____/\___/_/
                    /_/
Composer version 57fe33d0f371a95232d74e64c5c559eeac77a24f

Usage:
  [options] command [arguments]

Options:
  --help           -h Display this help message.
  --quiet          -q Do not output any message.
  --verbose        -v Increase verbosity of messages.
  --version        -V Display this application version.
  --ansi              Force ANSI output.
  --no-ansi           Disable ANSI output.
  --no-interaction -n Do not ask any interactive question.
  --profile           Display timing and memory usage information
  --working-dir    -d If specified, use the given directory as working directory.

Available commands:
  about            Short information about Composer
  config           Set config options
  create-project   Create new project from a package into given directory.
  depends          Shows which packages depend on the given package
  dump-autoload    Dumps the autoloader
  dumpautoload     Dumps the autoloader
  help             Displays help for a command
  init             Creates a basic composer.json file in current directory.
  install          Installs the project dependencies from the composer.lock file
  list             Lists commands
  require          Adds required packages to your composer.json and installs them
  search           Search for packages
  show             Show information about packages
  status           Show a list of locally modified packages
  update           Updates your dependencies to the latest version according to c
  validate         Validates a composer.json
rob@Computron:~$
```

Figure 7 PHP Composer commands [18]



```
C:\xampp\htdocs\laravel-project>php artisan help serve
Usage:
  serve [options]

Options:
      --host[=HOST]     The host address to serve the application on. [default:
"localhost"]
      --port[=PORT]     The port to serve the application on. [default: 8000]
  -h, --help            Display this help message
  -q, --quiet           Do not output any message
  -V, --version         Display this application version
      --ansi            Force ANSI output
      --no-ansi         Disable ANSI output
  -n, --no-interaction  Do not ask any interactive question
      --env[=ENV]       The environment the command should run under.
  -v|vv|vvv, --verbose  Increase the verbosity of messages: 1 for normal output,
 2 for more verbose output and 3 for debug

Help:
 Serve the application on the PHP development server

C:\xampp\htdocs\laravel-project>
```

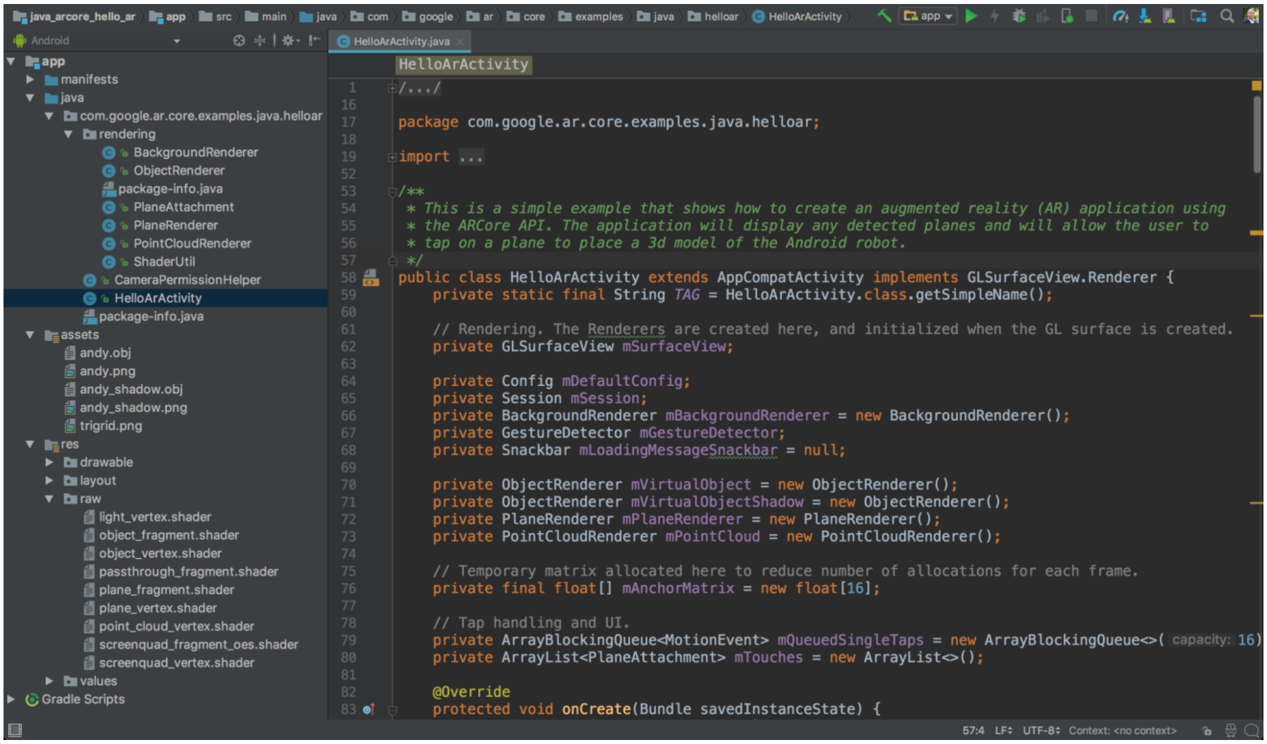
Figure 8 Artisan console interface [19]

PVS Code / IDE we used

To write this project, we needed various IDE. Since this project uses various technologies and programming languages, these systems need their own software development environments. Each IDE has its advantages and disadvantages. The rest of this section provides a brief discussion of the more relevant tools for this work.

Android Studio

Android studio is the official development environment created by Google and most of the applications that we use every day have been developed with it. Android Studio provides an interface for creating applications and takes over most of the complex file management [20]. Android Studio has a fast and feature-rich emulator that allows you to create an emulation of any device that supports the Android system. That is why Android Studio is number one for developers who want to create applications, in accordance with the material design of Google and access to advanced features of the platform.



```
/.../

package com.google.ar.core.examples.java.helloar;

import ...

/**
 * This is a simple example that shows how to create an augmented reality (AR) application using
 * the ARCore API. The application will display any detected planes and will allow the user to
 * tap on a plane to place a 3d model of the Android robot.
 */
public class HelloArActivity extends AppCompatActivity implements GLSurfaceView.Renderer {
    private static final String TAG = HelloArActivity.class.getSimpleName();

    // Rendering. The Renderers are created here, and initialized when the GL surface is created.
    private GLSurfaceView mSurfaceView;

    private Config mDefaultConfig;
    private Session mSession;
    private BackgroundRenderer mBackgroundRenderer = new BackgroundRenderer();
    private GestureDetector mGestureDetector;
    private Snackbar mLoadingMessageSnackbar = null;

    private ObjectRenderer mVirtualObject = new ObjectRenderer();
    private ObjectRenderer mVirtualObjectShadow = new ObjectRenderer();
    private PlaneRenderer mPlaneRenderer = new PlaneRenderer();
    private PointCloudRenderer mPointCloud = new PointCloudRenderer();

    // Temporary matrix allocated here to reduce number of allocations for each frame.
    private final float[] mAnchorMatrix = new float[16];

    // Tap handling and UI.
    private ArrayBlockingQueue<MotionEvent> mQueuedSingleTaps = new ArrayBlockingQueue<>( capacity: 16)
    private ArrayList<PlaneAttachment> mTouches = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

Figure 9 Android studio interface [20][21]

Advantages:

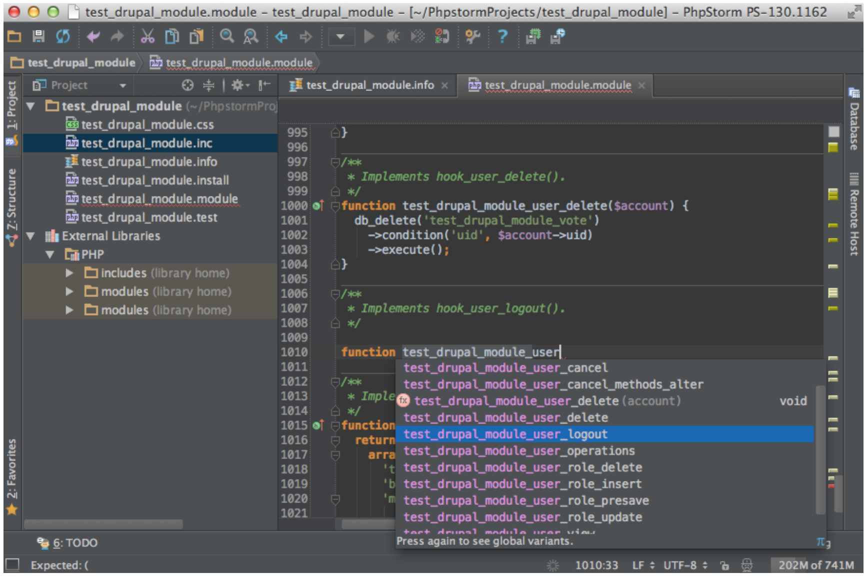
- Fast and feature-rich emulator
- Instant Run
- Free of charge
- Complete Support
- Simple integration
- Updates

Disadvantages:

- Brakes
- Need a good computer for the emulator
- Differences in control (when switching from Eclipse)

Php Storm

PhpStorm is an integrated PHP development environment with an intelligent editor that deeply understands the code and supports PHP 5.3-7.3 [22]. The PhpStorm environment offers functions for automatically completing PHP language constructs in code, inspecting code, and quick code navigation.



```
}

/**
 * Implements hook_user_delete().
 */
function test_drupal_module_user_delete($account) {
  db_delete('test_drupal_module_vote')
    ->condition('uid', $account->uid)
    ->execute();
}

/**
 * Implements hook_user_logout().
 */

function test_drupal_module_user
test_drupal_module_user_cancel
test_drupal_module_user_cancel_methods_alter
test_drupal_module_user_delete (account)                 void
test_drupal_module_user_delete
test_drupal_module_user_logout
test_drupal_module_user_operations
test_drupal_module_user_role_delete
test_drupal_module_user_role_insert
test_drupal_module_user_role_presave
test_drupal_module_user_role_update
Press again to see global variants.
```

Figure 10 PhpStorm interface [23]

<u>Advantages:</u>

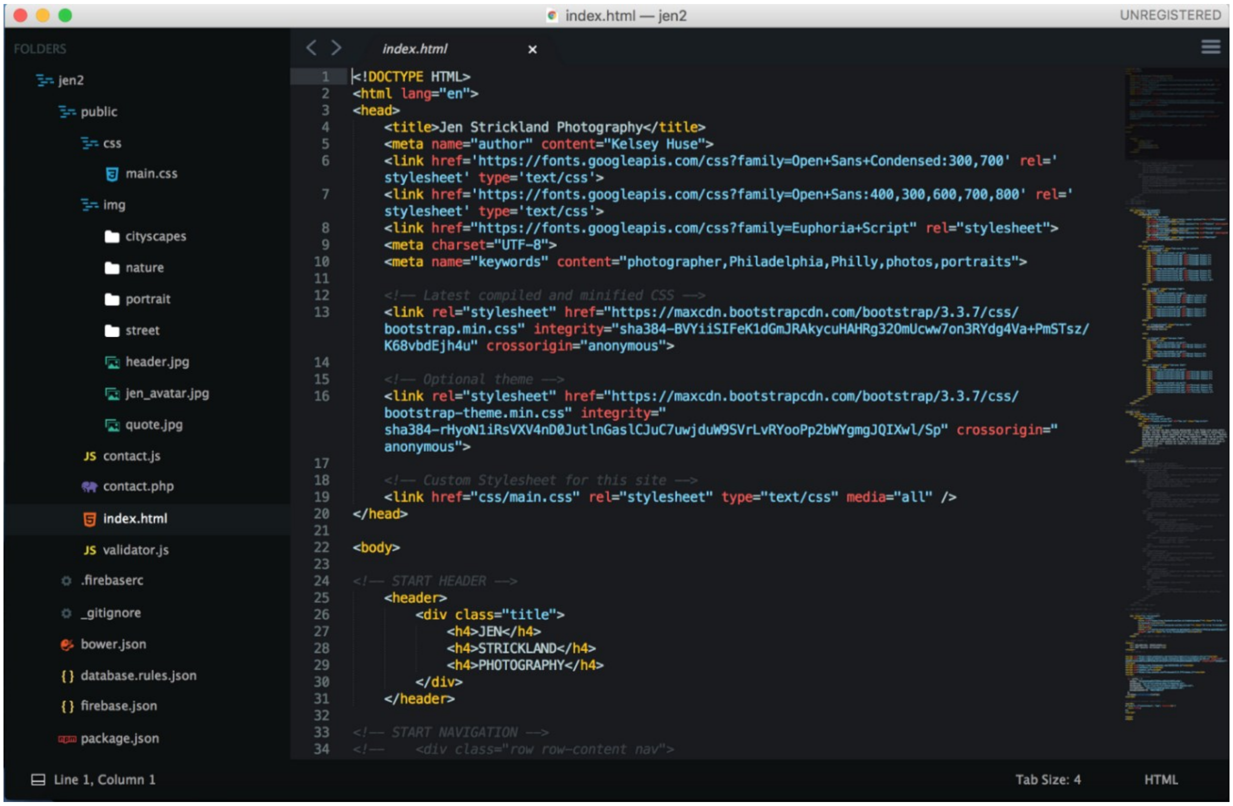
- Doesn't slow down
- Good interface
- Functionality and convenience
- Easy code highlighting
- Extraction of class variables

<u>Disadvantages:</u>

- Cannot be used as a fast editor for a single file
- High price
- Too complex for some developers

Sublime Text

Sublime Text is a paid cross-platform source code editor with an application programming interface that runs on Windows and IOS. The convenience of this editor is the use of various keyboard shortcuts.



```html
<!DOCTYPE HTML>
<html lang="en">
<head>
    <title>Jen Strickland Photography</title>
    <meta name="author" content="Kelsey Huse">
    <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300,700' rel='
    stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Open+Sans:400,300,600,700,800' rel='
    stylesheet' type='text/css'>
    <link href="https://fonts.googleapis.com/css?family=Euphoria+Script" rel="stylesheet">
    <meta charset="UTF-8">
    <meta name="keywords" content="photographer,Philadelphia,Philly,photos,portraits">

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/
    bootstrap.min.css" integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/
    K68vbdEjh4u" crossorigin="anonymous">

    <!-- Optional theme -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/
    bootstrap-theme.min.css" integrity="
    sha384-rHyoN1iRsVXV4nD0JutlnGaslCJuC7uwjduW9SVrLvRYooPp2bWYgmJQIXwl/Sp" crossorigin="
    anonymous">

    <!-- Custom Stylesheet for this site -->
    <link href="css/main.css" rel="stylesheet" type="text/css" media="all" />
</head>

<body>

<!-- START HEADER -->
    <header>
        <div class="title">
            <h4>JEN</h4>
            <h4>STRICKLAND</h4>
            <h4>PHOTOGRAPHY</h4>
        </div>
    </header>

<!-- START NAVIGATION -->
<!--    <div class="row row-content nav">
```

Figure 11 Sublime Text interface [24]

<u>Advantages:</u>

- multiple allocation
- assigning hotkeys to absolutely any action
- smart comments/recomments on hotkeys
- hidden character display
- personal settings are stored in a separate folder.

<u>Disadvantages:</u>

- high price

- loading time of the program

Chapter 4 - Conclusions and Future work

Conclusions

As a result of the Quality Assurance (QA) procedures put in place at UCM's Facultad de Informática it is required to track the students' attendance to the different educational activities that comprises the courses (e.g. lab sessions, classroom sessions, seminars, etc.). On regular basis this is done by generating a paper-based attendance list per activity that the students' must sign and teachers must collect and process in order to generate the stats required by the QA procedures.

This project alleviates the burden of managing the students' attendance tracking by creating a software application that systematize the traditional paper-based procedure by collecting in a central location the attendance data per educational activity for a specific course. Even more, it is possible to collect more data like specific time that the student clocks in, so it is possible to generate additional metrics that are not possible using the traditional paper-based process.

We created an Android native application that uses the available NFC hardware in the smartphone, so it is possible to use an NFC card like the UCM's TUI to facilitate even more the attendance tracking. Moreover, this solution it is scalable to other UCM's Faculties due to the penetration of the Android smartphones in Spain [25] (79,86%, data collected during last twelve months ending in May 2019) so teachers can used their personal smartphones. In addition, the proposed solution it is cost effective compare to other custom and hardware specific solutions if it is required to provide shared devices to teachers.

In addition to the Android application we have created a web-based management application for teachers, so they can plan ahead all the educational activities that comprise a specific course. This application offers a functionality similar to a personal calendar management application by allowing to create individual or recurrent activities for a specific course.

Both the Android app and the web application share a backend service that has been created as a REST API so all the restrictions and validations are performed in one place.

We have made not just an application, but a whole service around the students' attendance tracking. A lot of work has been done on creating solution simple to use and understanding how this should work from both the students' and professors' perspective. We thoroughly figured out this issue and imbued it in the final solution by our own daily-basis experience and the long discussions with our tutor.

At the beginning of the project we didn't have a clear vision of how it would look the final solution, so we applied an iterative and agile approach to manage the project by building playable solutions that incrementally add new features or change the existing ones. This has been a key point of the success of the project, because this not only help us to be confident around of having a solution, but also the playable software products facilitated the discussions with the client / tutor.

The project has enabled us to learn new technologies that we only heard of or use as regular users, such as the Android platform, the Laravel framework, VUE, and of course we had to understand how to design and secure a backend REST API in order to combine our native application with the management application to build the whole service.

Future work

The hardware embedded in UCM's TUI uses a MIFARE compatible chip that it is not compatible with all the NFC-enabled android smartphones. In fact, during the project we evaluated the possibility of integrating an alternative mechanism leveraging the opportunity offered by the TUI, because the back of the card includes a Code 128 1D barcode [26] that codifies the unique number assigned to this specific card. Hence it will be possible to integrate a barcode reader library to read the 1D code instead of using the NFC capabilities. Ultimately, we decided to use an alternative NFC card (not a MIFARE one) to test the application, however the barcode reader library could be a nice feature to have. In addition, the UCM's TUI NFC chip stores encrypted information about the user, however we couldn't take advantage of that feature that required to have access to a specific encryption key and a software SDK that is only provided to licensee.

Other interesting features to have in the Android app could be to add notifications previews, add a calendar view and the ability to create notes attached to the attendance lists.

Currently the communication between the Android application, the web management application and the backend REST API is done over HTTP connection that in a real setting must be over a secure TLS channel that must be enforced in both clients. In addition, the authorization between the client and backend REST API is done using an API access token [27] combined with a login endpoint that is used to authenticate the user. In order to deploy this solution at UCM it will required to integrate the applications with the corporate identity provider in a proper way, so this will require to change the current authentication and authorization scheme to use the OAuth2 OIDC protocol or OAuth2 protocol plus a user management service instead.

The current solution it is focused on teachers, however a possible evolution of the project could consider providing at least a web portal for students too, they can view the number the attendance classes or the number of absences. Finally, although the development has been done to address a real problem that our tutor suffers, there were no evaluation of the final prototype besides the tutor, so this could be an aspect to improve in a future evolution of this work..

Chapter 5 - Conclusiones y trabajo futuro

Conclusiones

Como resultado de la aplicación de los procedimientos de la calidad docente en los estudios de la Facultad de Informática de la UCM es necesario gestionar la asistencia de los alumnos en las diferentes actividades educativas realizadas como parte de una asignatura (e.g. sesiones de laboratorio, clases magistrales, etc.). De manera habitual este trabajo se lleva a cabo mediante la gestión de listas de asistencia en papel que los estudiantes deben firmar y los docentes deben recopilar y procesar para generar las estadísticas requeridas por estos procesos de calidad.

El propósito de este proyecto ha sido aliviar la carga de la gestión de la asistencia a clase mediante la creación de una aplicación informática que sistematizara el procedimiento habitual basado permitiendo recopilar la misma información en un único lugar para cada actividad educativa. Además, la solución resultante permite almacenar mayor información, como el momento exacto en el que el estudiante ficha, de modo que es posible generar otro tipo de métricas que no es posible generar con el procedimiento tradicional.

Hemos creado una aplicación nativa Android que hace uso del hardware NFC disponible en los teléfonos inteligentes, de modo que es posible utilizar una tarjeta NFC como la TUI de la UCM como mecanismo para facilitar aún más la gestión de la asistencia a clase. Además, esta solución también es escalable a otras Facultades de la UCM debido a la alta penetración del sistema operativo Android en el mercado español [25] (79,86%, dato recopilado durante los últimos doce meses terminando en mayo de 2019), de modo que los docentes pueden utilizar su propio teléfono inteligente. Además, la solución propuesta también es eficiente desde el punto de vista del coste comparada con otras soluciones de hardware específico, por ejemplo si fuese necesario proporcionar un dispositivo compartido por docentes para utilizar el servicio.

Tanto la aplicación Android como la aplicación web comparten un servicio común que se ha implementado mediante una API REST de modo que todas las restricciones y validaciones de lógica de negocio se realizan en un único lugar.

No hemos desarrollado únicamente una aplicación, sino un servicio completo alrededor de la gestión de la asistencia a clase. Se ha invertido una gran cantidad de tiempo en crear una solución simple de utilizar y que pueda ser entendida analizando el proceso tanto desde el punto de vista del docente como del estudiante.

Al comienzo del proyecto no teníamos una visión clara de la solución resultantes, de modo que hemos aplicado una metodología ágil e iterativa que nos ha permitido crear soluciones ejecutables con funcionalidades incrementales. Éste ha sido una de las claves del éxito del proyecto, ya que no solo nos ha permitido sentirnos más seguros al tener una solución ejecutable, sino que nos ha facilitado las reuniones de seguimiento durante el proyecto.

Este proyecto nos ha permitido explorar y trabajar con tecnologías que sólo habíamos escuchado hablar o de la que simplemente habíamos sido usuarios finales, como la plataforma Android y los armazones de aplicaciones como Laravel y VUE. Además, también ha sido necesario aprender como securizar una solución que tiene diferentes componentes que se ejecutan en diferentes dispositivos.

Trabajo futuro

El hardware que se utiliza como base para construir la tarjeta TUI de la UCM utiliza un chip MIFARE que no es compatible con todos los chips NFC que son utilizados en los teléfonos inteligentes. De hecho, durante el proyecto hemos evaluado la posibilidad de integrar algún mecanismo alternativo utilizando alguna de las oportunidades que ofrece la TUI. En la parte trasera de la tarjeta se encuentra un código de barras 1D basado en codificación 128 [26], que codifica el identificador único asignada a la tarjeta. Por tanto, sería posible integrar una biblioteca de lectura de códigos de barra para leer este código de barras en vez de utilizar el hardware NFC. Al final decidimos utilizar una tarjeta NFC diferente para poder probar la aplicación. Además, el chip NFC de la TUI de la UCM guarda información encriptada sobre el propietario de la tarjeta, aunque esta información sólo se puede leer utilizando una clave de encriptación concreta y utilizando un SDK de desarrollo que requiere licenciamiento.

Otra característica interesante a añadir en la aplicación Android sería la previsualización de notificaciones o el añadir una vista de calendario o integración con la aplicación de calendario del sistema.

En la actualidad la comunicación entra la aplicación Android, la aplicación web de gestión y la API REST utiliza el protocolo HTTP que en un escenario real de aplicación sería necesario que se utilizada sobre un canal de comunicación con cifrado TLS. Además, la autorización entre el cliente Android y la API REST se lleva a cabo mediante un token de acceso [27] combinado con una operación de login que permite identificar al usuario final. Para poder aplicar esta solución en la UCM sería necesario integrar los diferentes componentes con el sistema de gestión de identidad de la UCM de manera adecuada, de modo que sería necesario cambiar el actual sistema de autenticación y autorización para utilizar un mecanismo como OAuth2 OIDC o el protocolo OAuth2 más un servicio de gestión de usuario adicional.

La solución resultante de este proyecto está centrada en los docentes, sin embargo sería posible una evolución del proyecto en el que se considera al menos el desarrollo de un portal web donde los estudiantes también pudieran ver su porcentaje de asistencia y número de faltas. Finalmente, aunque el desarrollo se ha realizado abordando un problema real no ha habido una evaluación del prototipo final de la aplicación final aparte de la realizada por el tutor de este trabajo, de modo que este sería un aspecto a abordar en una extensión de este trabajo.

Chapter 6 - Authors' Individual Contributions

BARTLOMIEJ KOZIEL contributions to the project

The idea was created and implemented to make an application that can help and facilitate life in an educational institution, and in particular to help professors. Since I only had a phone with an Android platform, in connection with which I had to test more than thirty mobile applications that scan various cards with embedded NFC chips, RFID and other smart cards to try to recover at least some information from UCM university cards.

After finding out the problems that were discovered during the work on the native mobile application and their solution, the work was continued on studying the structure of building a mobile application. Since this is the first introduction to programming on Android, a lot of courses have been reviewed and a large amount of documentation related to this topic has been read. In the course of the study and attempts to create application that can accept only an id from a smart card and compare it with the application id entered, a lot of effort was spent. Next the application development was much faster.

Further, the work began on the design and implementation of a web page, there were created a lot of layouts on a regular A4 sheet, so that we could understand and fully represent the model of the web site. As each of us came up with new and new ideas, we had to come to a common decision to follow the layout that we now have.

The main task on my part was to select and choose a convenient framework to which later it would be possible to connect the REST API without any problems and link the fully mobile native application to the web site. After a long study of this issue, it was decided to use Laravel together with Vue.js, since this framework was better suited to our requests and the main advantage was that it was easier to understand.

All further work has been done on correcting the functionality of a mobile application and creating a web site based on the already ready idea of the native mobile application.

The idea was proposed to add a calendar to the website, which works on the principle of a Google calendar that is you, could create some tasks and add them to a specific day of the calendar.

I started working on this part of the project, since this topic was very interesting. There are many open-source calendars forums available on the Internet, but the main problem was that they were huge and larger than our project and difficult to modify. Then I had to collect my calendar with several existing ones, after which the most interesting thing began: how to add an event to the calendar and later be able to read or view this event.

All the further time I was developing the addition of an event to the calendar, and linking the activity of objects between the web site and the mobile native application. That is, the main task was to create an activity in the calendar and the ability to view this activity in the application, as well as on the website itself.

Later, the professors account was improved, so to speak, made for complete convenience and creating the right course, group and adding students which the administrator loads. A lot of previously added functionality for the service was reviewed and tested. A token was also created, which is created during the first authorization in the mobile native application and replaces the subsequent authorization using the login and password.

YEVHENII LASHUTKO contributions to the project

After observing Barltomiej long attempts, a problem was discovered which was related to his phone, since at the time of creating the application he had a Samsung S6, so I had to search for information on reading smart cards in various forums. There was the information that the Samsung S6 cannot read information from various chips that are in the cards, in particular the university UCM cards. But it is possible to read cards of this type like the university UCM cards as they are present on the younger version of the phone, and this is the Samsung S3.

After that, other smart cards, which at that moment were with us, were tested. The tarjeta transporte publico map was chosen, where the chip is located, which provides all the information we need about the card itself. Having found information on the Internet that functions that read and verify card id numbers can work with various chips and they are not limited to only 1 model of smart cards. After that, immediately checking the information I was convinced of the success of this model of the project.

After the main part of the application idea was already implemented, it was necessary to think about how to increase the capabilities of this project. The idea was invented, so that if every professor could have full access to editing the class schedule, to create an attendance list and that this would be not only from the application, but in comfortable conditions from a laptop.

Then the idea was born to make a web page that will be directly connected with the native Android mobile application. In which there will be an opportunity to create subjects, courses, groups and to have full access to editing them.

In addition to the fact that we have been creating a web site with a mobile application for a long time, we had to make the information on the web site and the mobile application shared so that it is constantly synchronized between these two small projects so that the result is one common project.

The authorization was added both to the site and to the application, which allowed us to create unique accounts for each user by our service. Then I had to delve into the REST API technology to make the account information one and the same at all sites. In the process of developing an application, we got the idea to divide users into 2 types, the first type is the administrator and the second one is a professor, each of them should have its own functionality. My task was to provide an administrator account with all the necessary management capabilities, such as creating students, professors and administrators who can add or change data.

So also auto-synchronization of data on the native mobile application was also added when connecting to the Internet. That is, after the general data was made both on the site and in the application, the Internet was needed to use the application, but it is not always possible either due to poor cellular connection or because of a bad connection Wi-Fi network. Then auto-synchronization was introduced, which did not require constant access to the Internet.

I also worked on checking the functionality of the application and correcting errors arising in creating new pages or various links between applications and all the shortcomings of the site have been corrected.

References

[1] https://www.thinkwithgoogle.com/advertising-channels/mobile-marketing/the-new-multi-screen-world-study/

[2] https://www.thinkwithgoogle.com/advertising-channels/mobile-marketing/device-use-marketer-tips/

[3] https://appdividend.com/2018/11/17/vue-laravel-crud-example-tutorial-from-scratch/

[4] https://martinfowler.com/eaaCatalog/dataTransferObject.html

[5] https://ru.wikipedia.org/wiki/Android

[6] http://www.dailycomm.ru/m/43799/

[7] https://medium.com/@ssaurel/create-a-nfc-reader-application-for-android-74cf24f38a6f

[8] https://hackernoon.com/what-is-vue-js-and-what-are-its-advantages-4071b7c7993d

[9] https://jonathanmh.com/deploying-a-vue-js-single-page-app-including-router-with-docker/

[10] https://medium.com/javascript-in-plain-english/implement-movie-app-with-vue-vuetify-axios-open-movie-database-api-d12290318cf9

[11] https://vuejs.org/v2/guide/instance.html

[12] https://www.youtube.com/watch?v=cy50mLPk-yo

[13] https://medium.com/employbl/build-authentication-into-your-laravel-api-with-json-web-tokens-jwt-cd223ace8d1a

[14] https://www.phpmyadmin.net/docs/

[15] https://www.hostingadvice.com/how-to/install-phpmyadmin-on-ubuntu/

[16] https://habr.com/ru/post/145946/

[17] https://laravel.com/docs/5.8/artisan/

[18] https://www.w3resource.com/php/composer/a-gentle-introduction-to-composer.php

[19] https://www.tutorialspoint.com/laravel/laravel_artisan_console.htm

[20] https://losst.ru/kak-polzovatsya-android-studio

[21] https://developers.google.com/ar/develop/java/quickstart

[22] https://jetbrains.ru/products/phpstorm/

[23] https://blog.jetbrains.com/phpstorm/2013/07/drupal-development-using-phpstorm/

[24] https://medium.com/@kelseyhuse/a-beginners-guide-to-beautifying-sublime-text-a3deda360e84

[25] http://gs.statcounter.com/os-market-share/mobile/spain

[26] https://www.barcodefaq.com/1d/code-128/

[27] https://auth0.com/docs/tokens/overview-access-tokens

Annex I: Installation Manual

Minimum system requirements for editing

	Windows	**OS X / macOS**	**Linux**
OS version	Windows 7	Mac OS X 10.10	GNOME or KDE desktop
RAM	3 GB RAM minimum		
Disc space	5 GB of disk space		
Screen resolution	1280x800 minimum		

Bitbucket source repository:

- https://bitbucket.org/796854/erasmus/src/master/

How to run project?

To run project you must follow the following instructions.

How to download project

The source code of the project is hosted in a Bitbucket repository where is located at https://bitbucket.org/796854/erasmus/src/master/.

Then you can use git client, like the git command line client, to execute the following command:

```
git clone https://bitbucket.org/796854/erasmus.git
```

If you need more info about how to use git commands, you can have a look to:

https://www.atlassian.com/git/tutorials/setting-up-a-repository/git-clone

After this we will have on our computer or server all the source code of the project. The project structure it is simple: MyBulletinAPI, MyBulletinNative and MyBulletinVue are the folders containing the, respectively, the backend REST API, the Android native application and the web management application. In addition, the repository also contains an demo APK binary for the Android application.

Installation instructions

Now when we have repository we could run each parts of project. Firstly, we should setup the Laravel API because it is required for the other two applications to run.

Laravel API installation

The API is inside the MyBulletinAPI folder. We need to copy this on server or run it locally. To install it you need composer (You can download it from https://getcomposer.org/). The first step is run console and go to folder with this app and run “composer install” to install all composer dependencies

```
composer install
```

Second You could install also npm (You can download it from https://www.npmjs.com/get-npm):

```
npm install
```

Third step is to configure .env file. It is configuration file of our Laravel API.

```
cp .env.example .env
```

Then we need to create in our MySQL database, for example by using phpMyAdmin database client and then **replace the example values inside the .env** file with access to database: DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, DB_PASSWORD.

Fourth step is generating application key:

```
php artisan key:generate
```

This key is randomly generated and stored in .env file.

At the end we have to migrate and seed database using “php artisan” that is embedded in Laravel.

```
php artisan migrate

php artisan db:seed
```

Those commands are creating Laravel API structure in your database and seed will fill essential and basic records to database.

You could find more about how to setup a Laravel project at:

https://devmarketer.io/learn/setup-laravel-project-cloned-github-com/

VUE API installation

To install VUE application you need to have Node.js (you could download it from https://nodejs.org/en/). Then you have to open the console and go to MyBulletinVUEapp. And run following commands.

```
npm install

npm run dev

npm run build

npm run build --report
```

As a reminder these commands can be also find out in the repository "README.md" file here:

https://bitbucket.org/796854/erasmus/src/master/MyBulletinVue/

After building the VUE client, you need to change the **`serverUrl`**: 'http://vps669485.ovh.net/', defined in the **`src/master/MyBulletinVue/src/store/store.js`** file.

Native app installation

To build the native application you need to have installed Android studio and SDK tools. You can download them from:

https://developer.android.com/studio

Then you have to run Android Studio and import project. After that you could compile project and Android Studio will do everything to run. You will need also phone with NFC to test application because emulators in Android Studio do not have NFC port.

Annex II: User Manual

Minimum system requirements for using

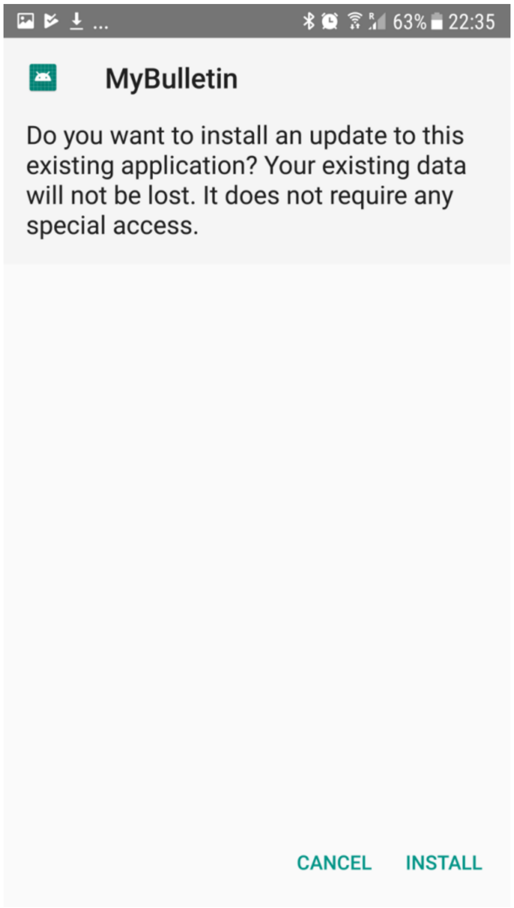
	Android	**Any operating system**
OS version	Android 5.0 +	Browser availability
Disc space	10 MB	-
RAM	500 MB	1GB

Bitbucket:

- https://bitbucket.org/796854/erasmus/src/master/

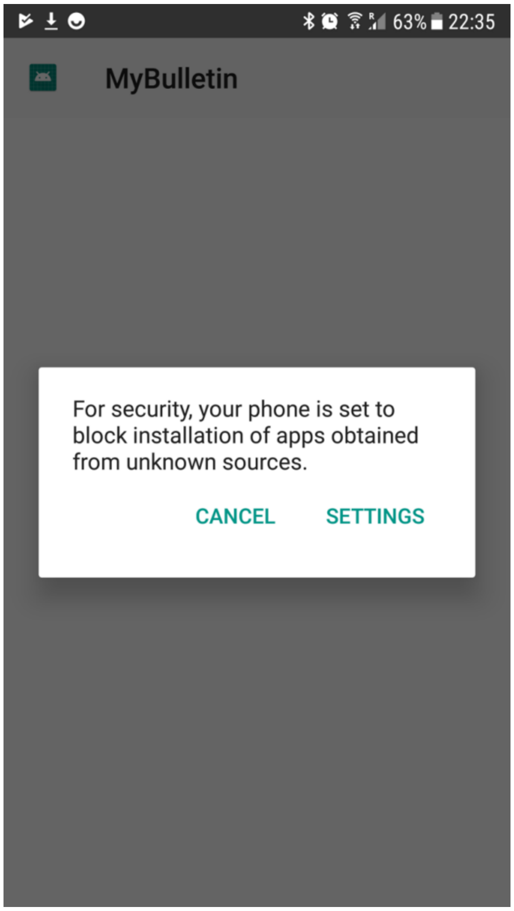
Installation for users

To start using the My Bulletin application on your Android mobile phone, you need to go to the Bitbucket repository (link below) and download the file “MyBulletin-download.apk” or go to this temporarily demo server http://vps669485.ovh.net/download/app.apk. and downloading will start automatically. After successfully downloading the file, you need to click on it and a similar screen should appear as in Figure 12, then click on the “SETTINGS” button.



MyBulletin

Do you want to install an update to this existing application? Your existing data will not be lost. It does not require any special access.



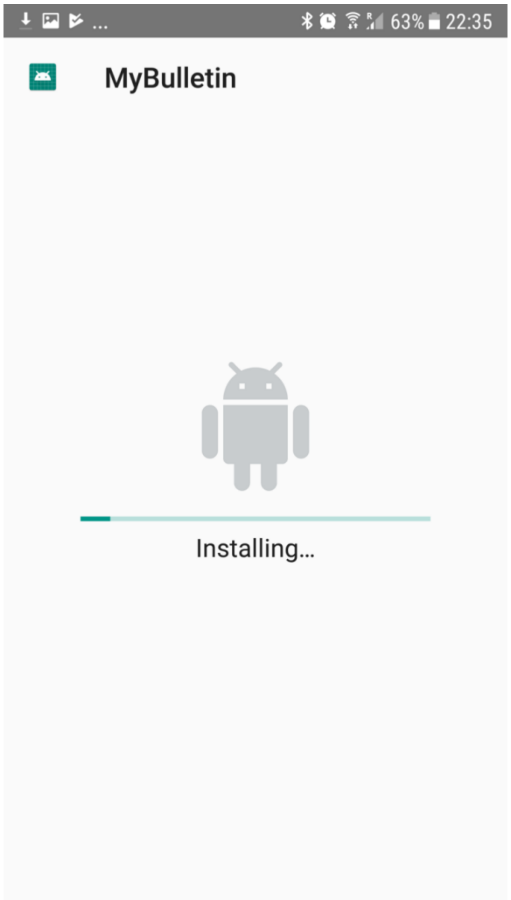
MyBulletin

For security, your phone is set to block installation of apps obtained from unknown sources.

Figure 12 Unknown sources

Figure 13 Install

Next, the Android system will ask you for permission to install a third-party program (see Figure 13), where you need to confirm the installation by clicking on the "INSTALL" button, after this action the installation will be performed Figure 14. After waiting a brief moment, the My Bulletin application will be installed on the smartphone and will be informed about the installation of the application (see Figure 15). To start the application, click on the "Open" button and then the authorization of the My Bulletin application will open (see Figure 16).



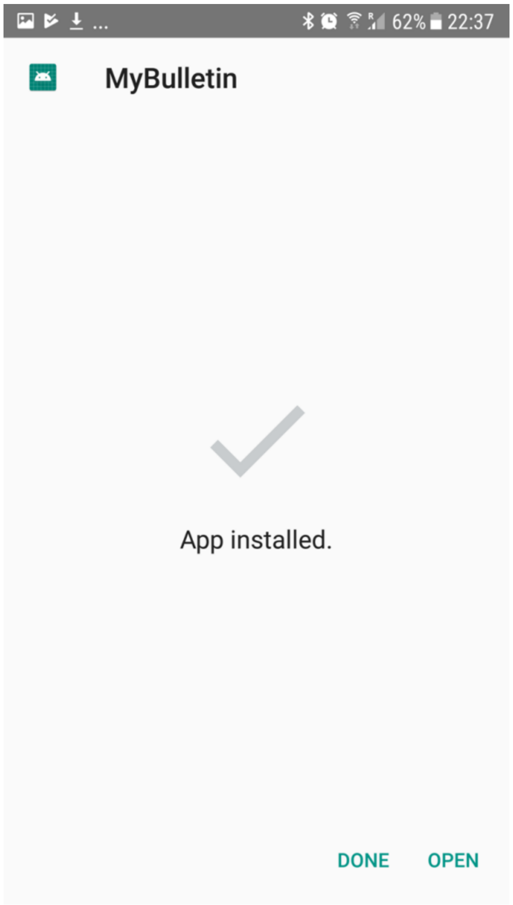


Figure 14 Installing

Figure 15 App installed

To start using the My Bulletin website you need to go to the Bitbucket repository (link below) and open the "temporary server.txt" file which contains the link to the temporary hosting and a list with logins and passwords for different users.

Android native application

User interface (android system)

If you have completed the previous installation steps described in Annex I, you can start using the app. We will start by using the mobile native application My Bulletin. To begin with, we will open the application My Bulletin, after which we will have an authorization screen (see Figure 16), where you need to enter the login and password provided by the administrator. After entering the data in the required fields and confirming with the "LOG IN" button, this device is linked to the professor account. Then the next page of the application opens (see Figure 17) in which you need to tap the NFC card to the phone in order to log into the application and get access to the selection of the subject (see Figure 18).

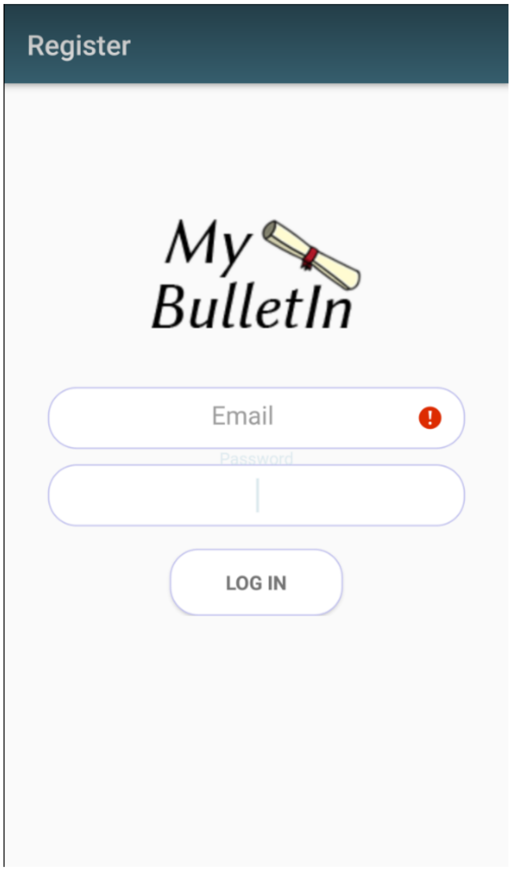
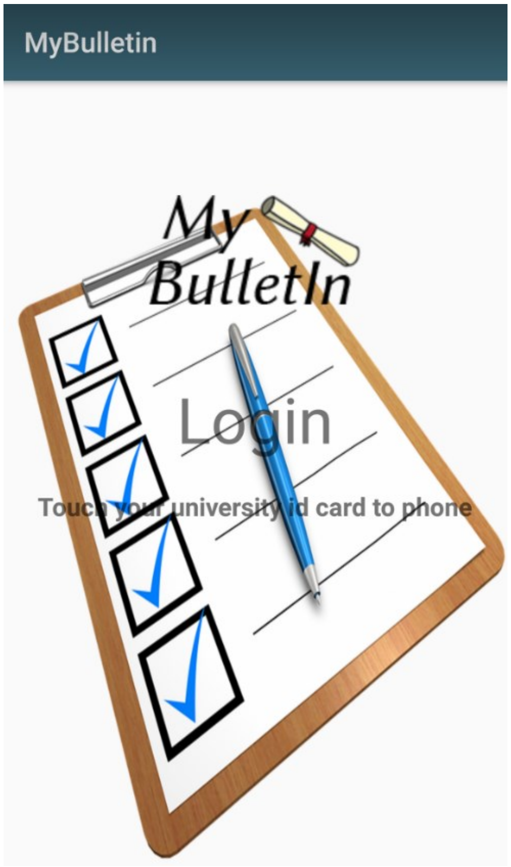


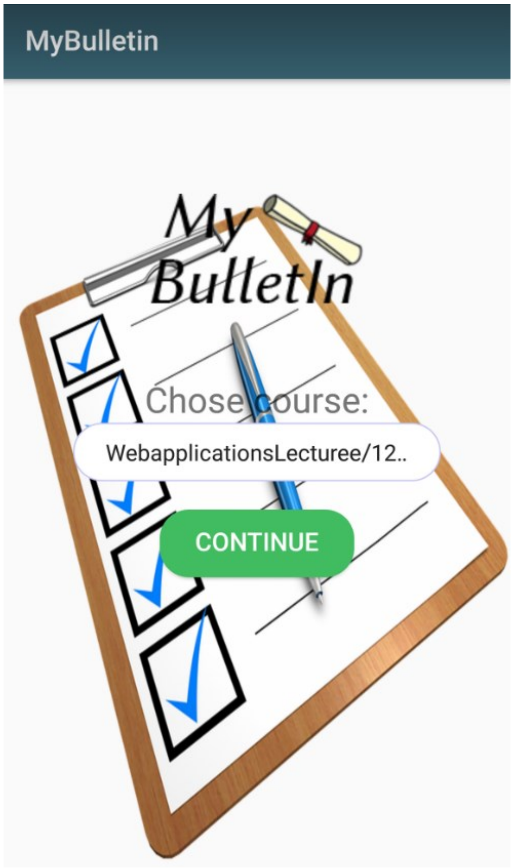
Figure 16 Backend Log in



Touch your university id card to phone

Figure 17 Backend Touch id card

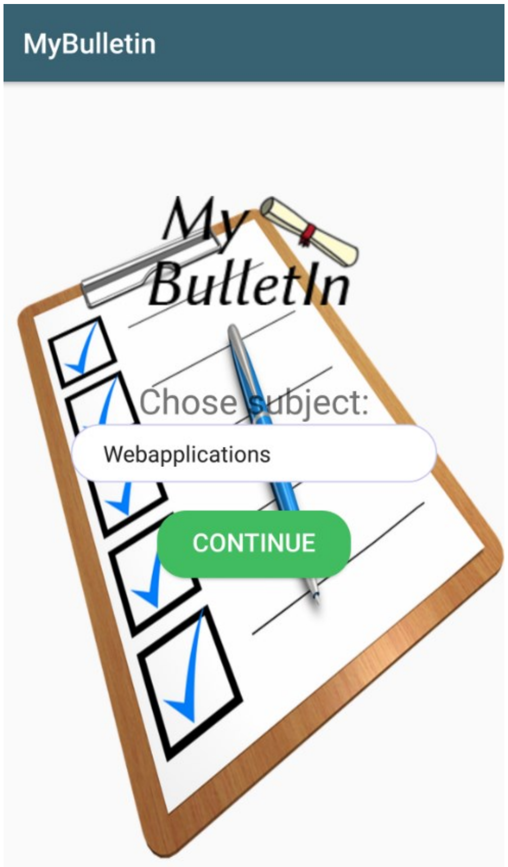
During the subsequent use of the application, as we said earlier, it is not necessary to enter a login and password again, so the log-in page ready to tap the NFC card will open immediately (see Figure 17).



MyBulletin

Chose course:

WebapplicationsLecturee/12..



MyBulletin

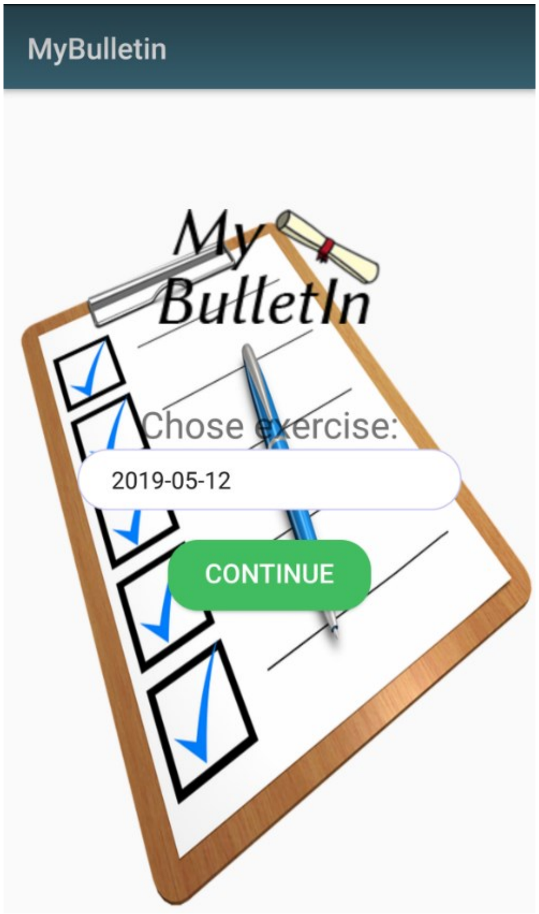
Chose subject:

Webapplications

Figure 18 Choose Subject

Figure 19 Choose Course

In Figure 18, you need to select the needed item with which to continue the operation of the application and confirm with the "Continue" button. Then the next menu opens with a choice of courses (see Figure 19) where you can choose the appropriate course. The name of the course includes: the name of the course and the "/" number of the group. Select the desired course and confirm with the "Continue" button and then go to the next menu where you need to select exercise (see Figure 20) for the attendance list and then confirm with the "Continue" button.

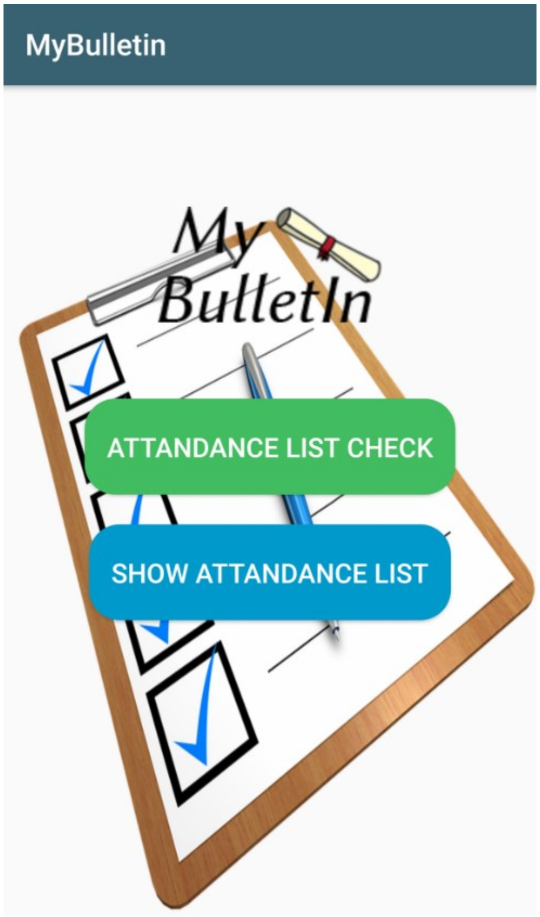


MyBulletin

Chose exercise:

2019-05-12

Figure 20 Choose Exercise

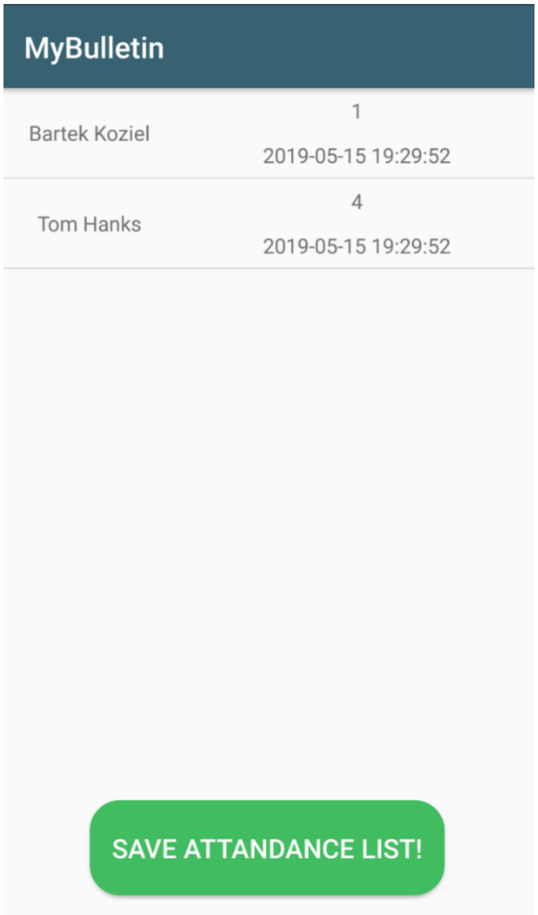


MyBulletin

Figure 21 Main menu

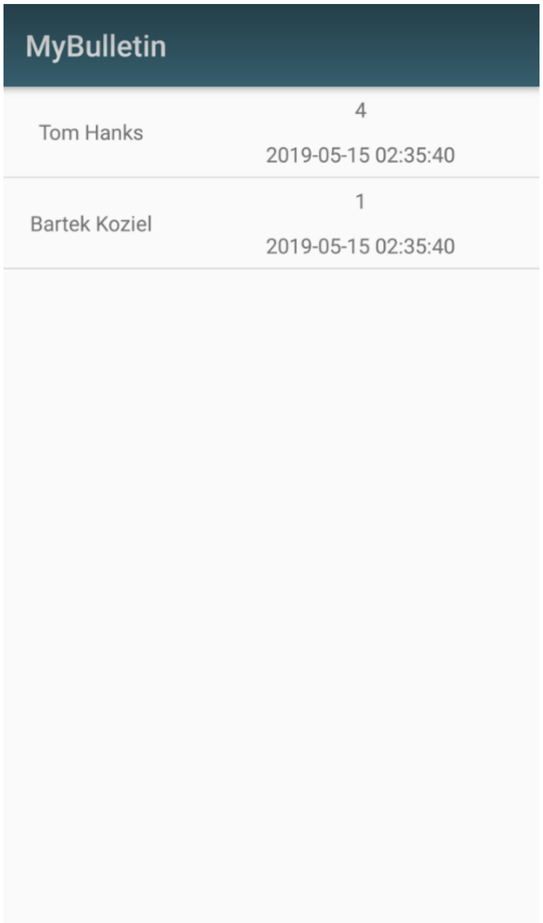
After the previous action, the application will redirect to the main menu (see Figure 21), where you need to choose to create or view the attendance list.

We will begin with the creation, on the smartphone screen, select the necessary button and click on it, and then the application redirects to the creation of the attendance list (see Figure 22). Now the attendance list is created and you need to note student attendance in the classroom, all you need is for each student to attach their student card to the telephone after the card is attached to the phone, in this list (see Figure 22) the name, the surname of the student and the date when the student was entered on the attendance list are displayed.



Bartek Koziel	1	2019-05-15 19:29:52
Tom Hanks	4	2019-05-15 19:29:52

Figure 22 Attendance lists



Tom Hanks	4	2019-05-15 02:35:40
Bartek Koziel	1	2019-05-15 02:35:40

Figure 23 Show Attendance list

After students were checked in the attendance list, you need to save this list by clicking on the "SAVE ATTENDANCE LIST" button and then the list will be completely saved on the server. To return to the main menu, you need to press the button on your smartphone back and then we will see this menu (see Figure 21).

Next, if we choose to view the attendance list and select the needed button and click on it, and then the application redirects to the next page (see Figure 23). This displays the entire the attendance list that was previously created on the previous list (see Figure 22).

To select another item or course, you need to click on the button back on your smartphone and select the needed item or course, then all actions are repeated as described above.

Web management application

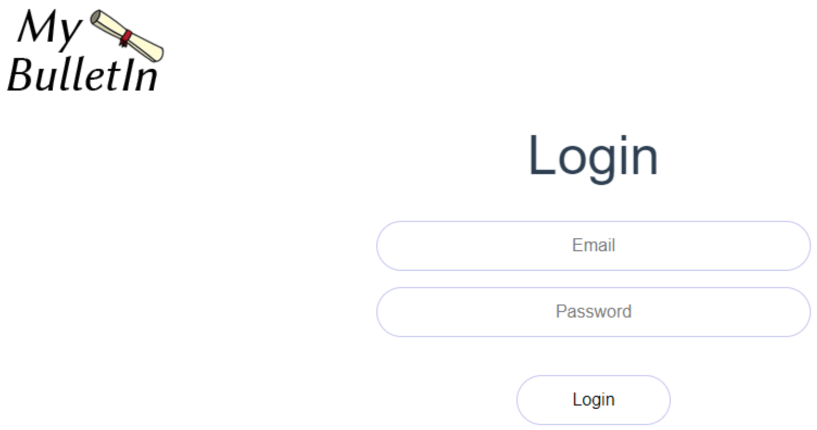


Figure 24 Website Start Page

There are two user roles on the web application, the administrator and the professor. Next sections will describe in more detail the different capabilities per role.

When you first connect to the web site, a login form appears (see Figure 24) so you need to enter a username and password and confirm interaction.

Web management application (administrator interface)

To begin with we will describe the different features of the administrator role. As mentioned above, you need to enter the login and password of the administrator account here (see Figure 24) and then the page will open (see Figure 25) with the main administrator menu. In this menu there are two main buttons "Students and Professors", when you click on the "Students" button, a page opens (see Figure 26), where we can see a list of students with Names, Surnames, E-mails and, accordingly, Tag id of university cards. On this page, you can delete any student from our server, you can change the information about her and also add a student to the server.

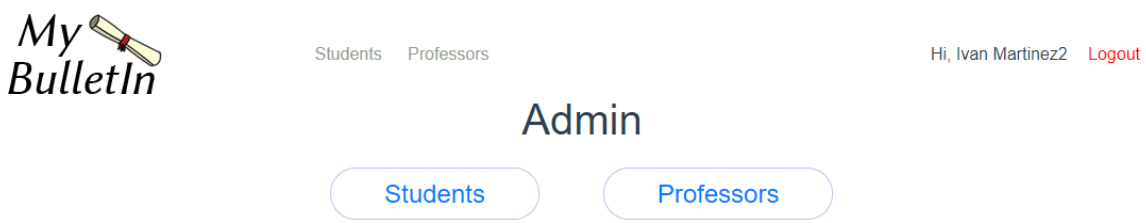
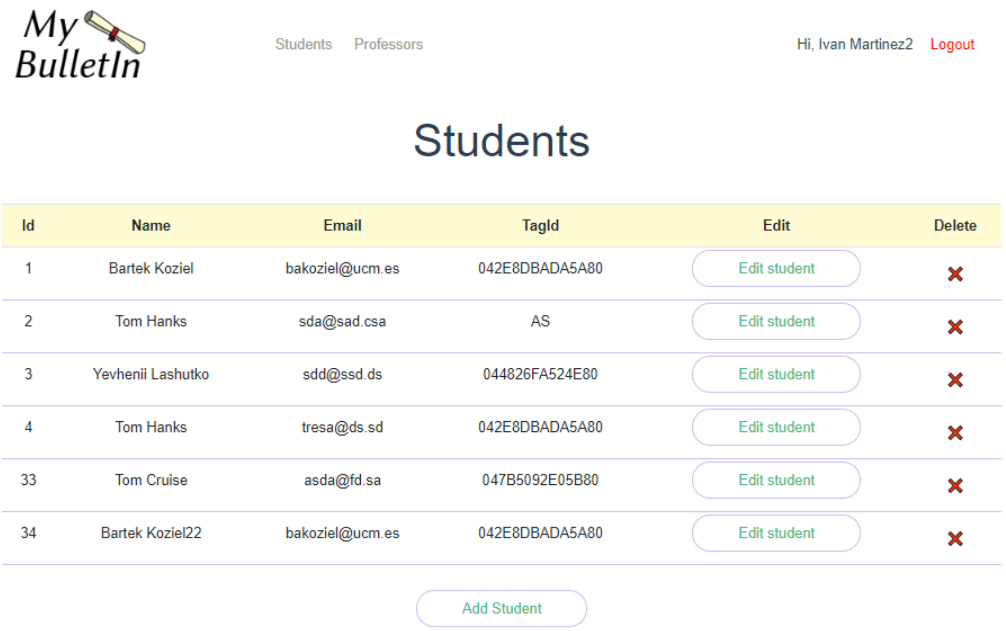


Figure 25 Website Main Menu

Clicking on the "Add student" button opens the page (see Figure 27), where there are several required fields to fill them in. If you need to add a large list of students and there are not enough fields, then you need to click on the "More" button, which will add new row of fields without losing the previously entered information on this page. After all the students have been defined, you need to click on the "Add student" button and all students will be added to the server.



Students

Id	Name	Email	TagId	Edit	Delete
1	Bartek Koziel	bakoziel@ucm.es	042E8DBADA5A80	Edit student	✗
2	Tom Hanks	sda@sad.csa	AS	Edit student	✗
3	Yevhenii Lashutko	sdd@ssd.ds	044826FA524E80	Edit student	✗
4	Tom Hanks	tresa@ds.sd	042E8DBADA5A80	Edit student	✗
33	Tom Cruise	asda@fd.sa	047B5092E05B80	Edit student	✗
34	Bartek Koziel22	bakoziel@ucm.es	042E8DBADA5A80	Edit student	✗

Figure 26 Students' list

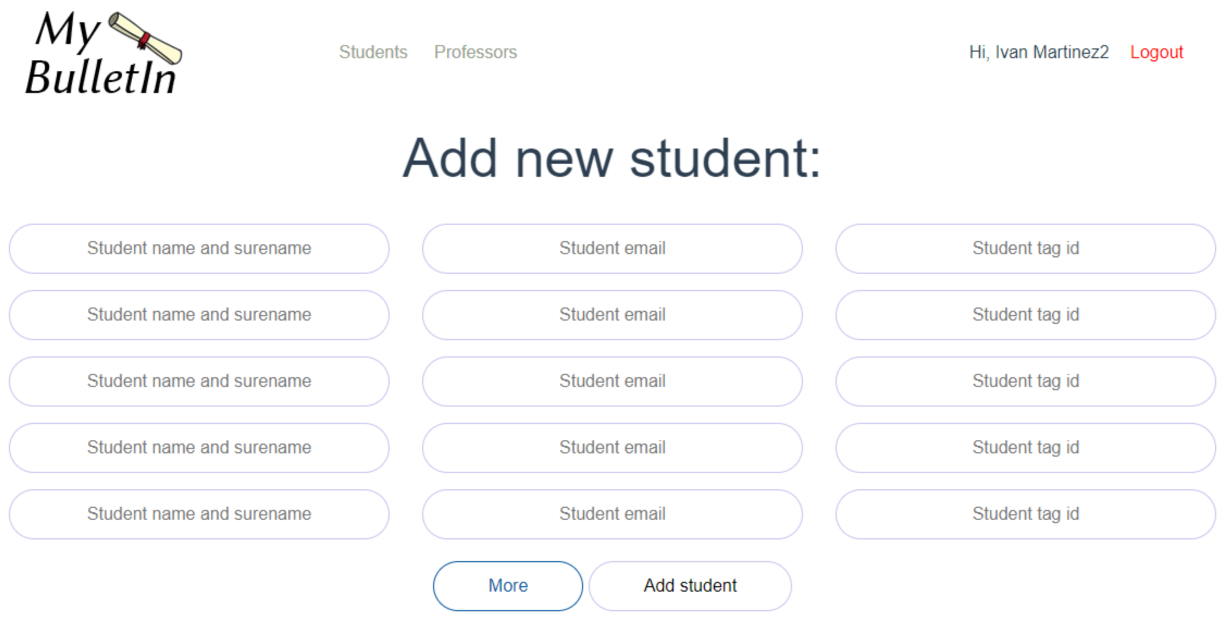
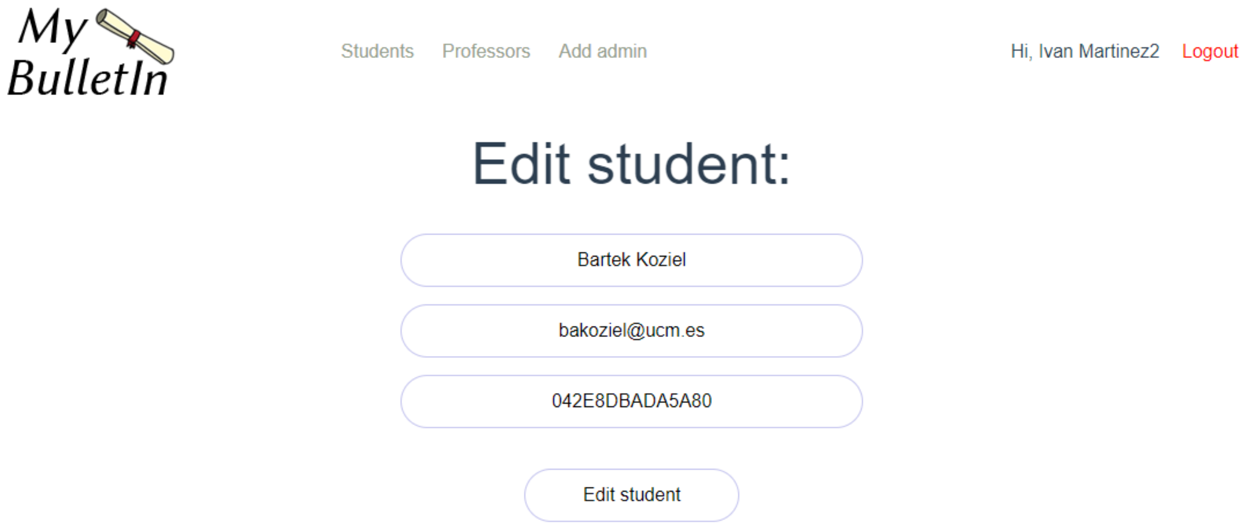


Figure 27 Add new student



Bartek Koziel

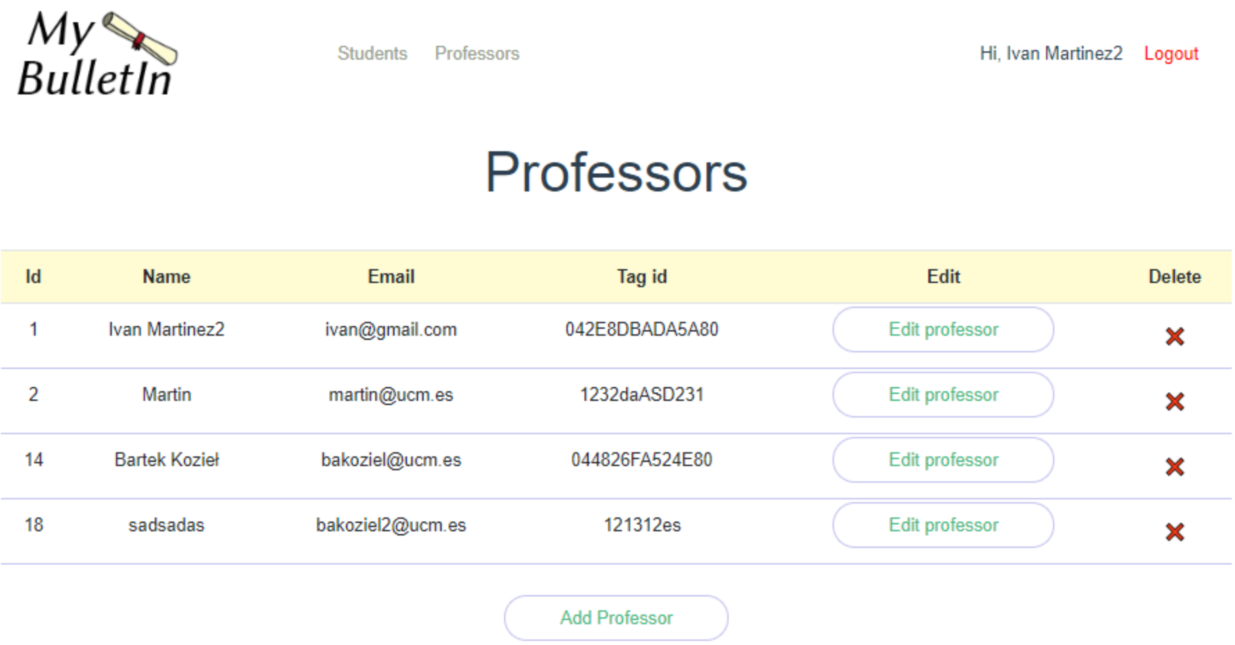
bakoziel@ucm.es

042E8DBADA5A80

Figure 28 Edit student

Another option is to edit the data of a particular student by clicking on the "Edit student" page (see Figure 28) where you can correct or even replace student data.

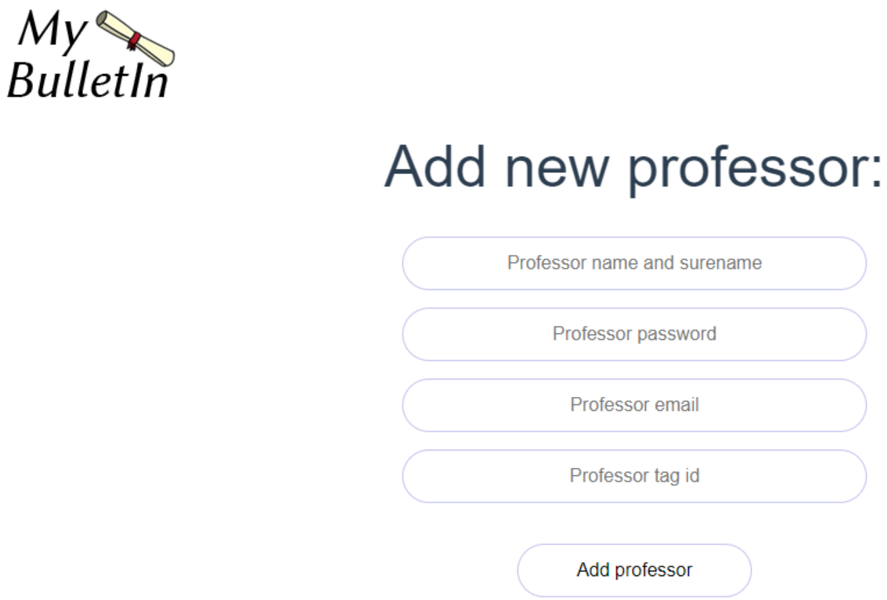
Now we will go to the Professors page by selecting the homonymous option that is located in the header of the site and clicking on it to open this page (see Figure 29). There is also a function to add, edit and delete professors account from the application. We will move on to creating a professor by clicking on the "Add professor" button and then the page will open (see Figure 30) where you can create only one professor by entering the data in the appropriate fields. Once the account has been created it is available to be used in the Android application. After entering all the data, you need to click on the "Add professor" button and a notification will show up to confirm the action. To return to the previous menu, you need to select and click on the "Professors" button in the header of the site and then the page with the list of professors (see Figure 29) will open again, where you can also check all professors and if someone is missing, you can add repeating the previous procedure again. You can also edit data of the professor by clicking on the "Edit professor" page (see Figure 31) where you can correct or even replace the data on the professor.



Professors

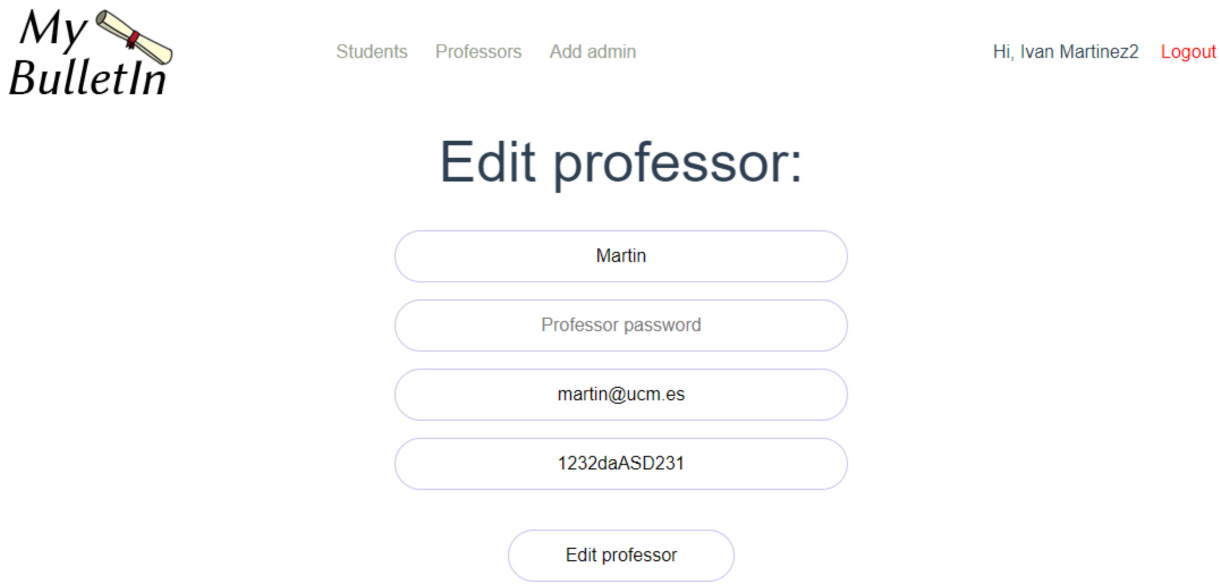
Id	Name	Email	Tag id	Edit	Delete
1	Ivan Martinez2	ivan@gmail.com	042E8DBADA5A80	Edit professor	✖
2	Martin	martin@ucm.es	1232daASD231	Edit professor	✖
14	Bartek Kozieł	bakoziel@ucm.es	044826FA524E80	Edit professor	✖
18	sadsadas	bakoziel2@ucm.es	121312es	Edit professor	✖

Figure 29 Professors' list



Add new professor:

Figure 30 Add new professor



Edit professor:

Martin

martin@ucm.es

1232daASD231

Figure 31 Edit a professor

Finally the administrator role create additional administrator accounts accounts, for this you need to select and click on “Add admin” in the header of the site, then the page will open (see Figure 32) with the addition of the site administrator. On this page you need to enter all the data that is needed and then click on the "Add admin" button and if everything is done correctly, it will be indicated that the data has been added.

To logout from the application you only need to click on "Logout" in the upper right part of the website header and then the login page will open again (see Figure 24).

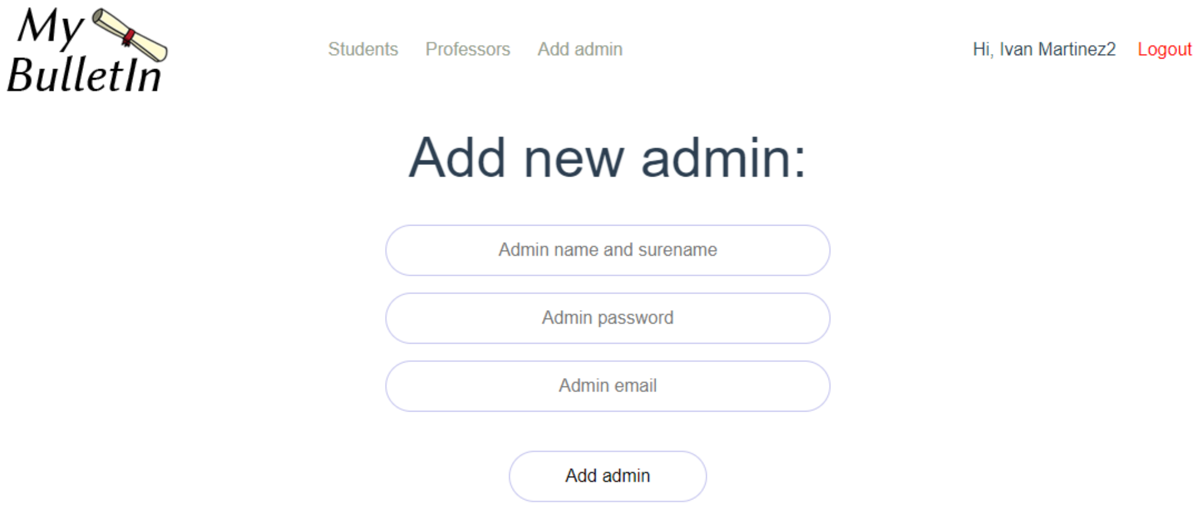
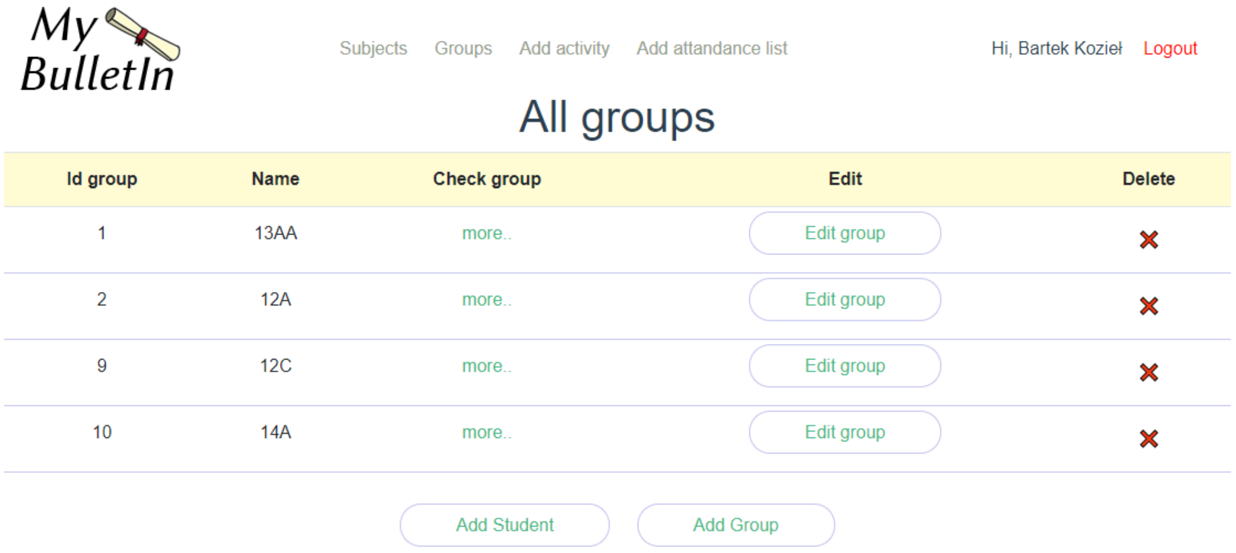


Figure 32 Add new administrator

Web management application (professor interface)



Id group	Name	Check group	Edit	Delete
1	13AA	more..	Edit group	✗
2	12A	more..	Edit group	✗
9	12C	more..	Edit group	✗
10	14A	more..	Edit group	✗

Figure 33 Main menu All groups

After you have entered the login and password for the account to the professor here (see Figure 24) and confirmed the login, a page will open (see Figure 33) where you can define the groups of students that will be attached to the account of the professor. On the Add Student page, you can add the email of the student account to the needed group as you can see in the screenshot of the (see Figure 35).

After entering the e-mail address, a drop-down menu will appear in which you need to select a group to which you want to link a particular student and then confirm adding the student by clicking on the "Add student" button and if everything is done correctly, the message will appear that the data has been added. To return to the previous menu, click on "Groups" in the header of the site and then the page will open (see Figure 33), to create a new group, click on the "Add Group" button and the page will open (see Figure 34), where you need to enter the name of the group and confirm the changes with the "Add group" button.

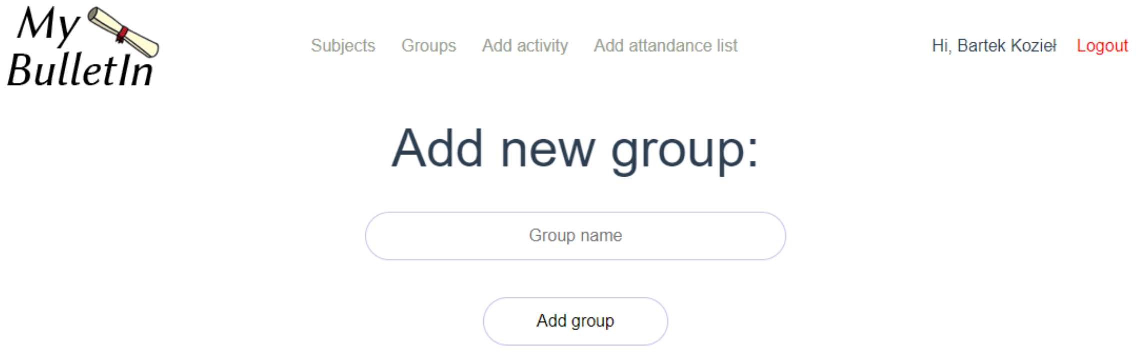
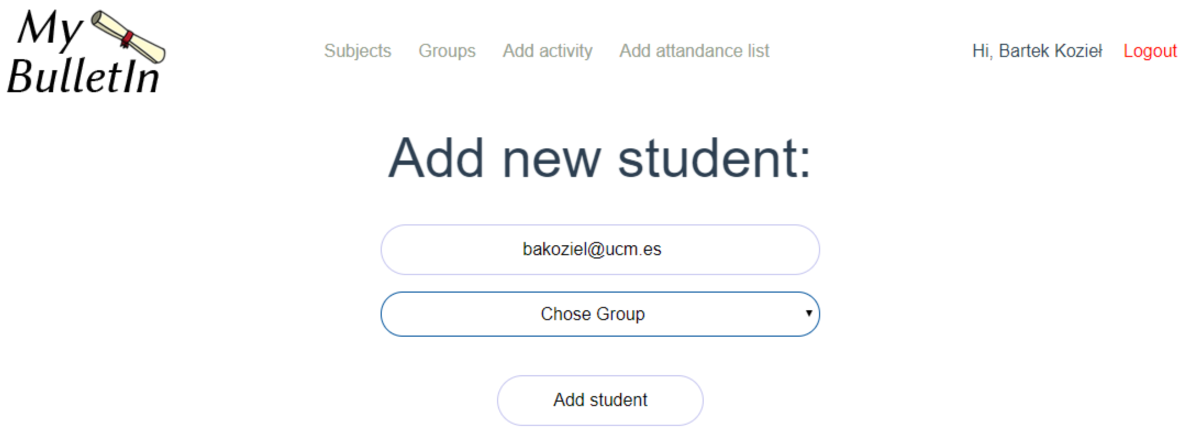


Figure 34 Add new group

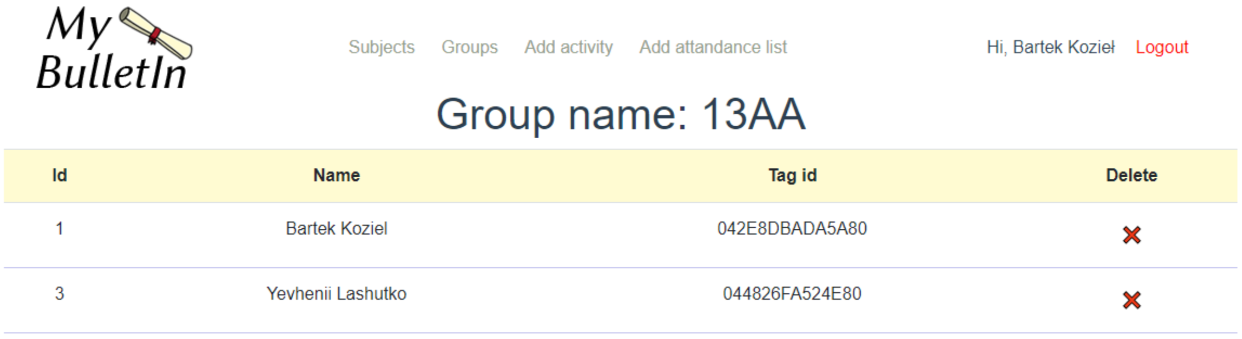


Add new student:

bakoziel@ucm.es

Figure 35 Add a student to a group

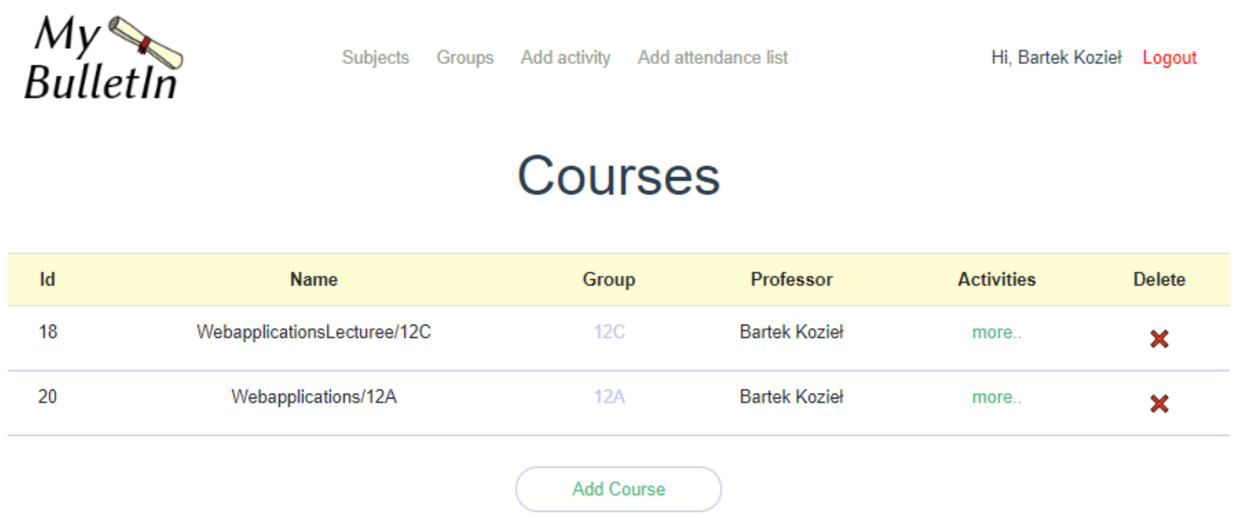
To view information about the group and check the students that are added to this group, then you need to click on the more.. inscription, and then this page will open (see Figure 36), where you can check the entire list of students in this group and remove the excess student, if there is one.



Group name: 13AA

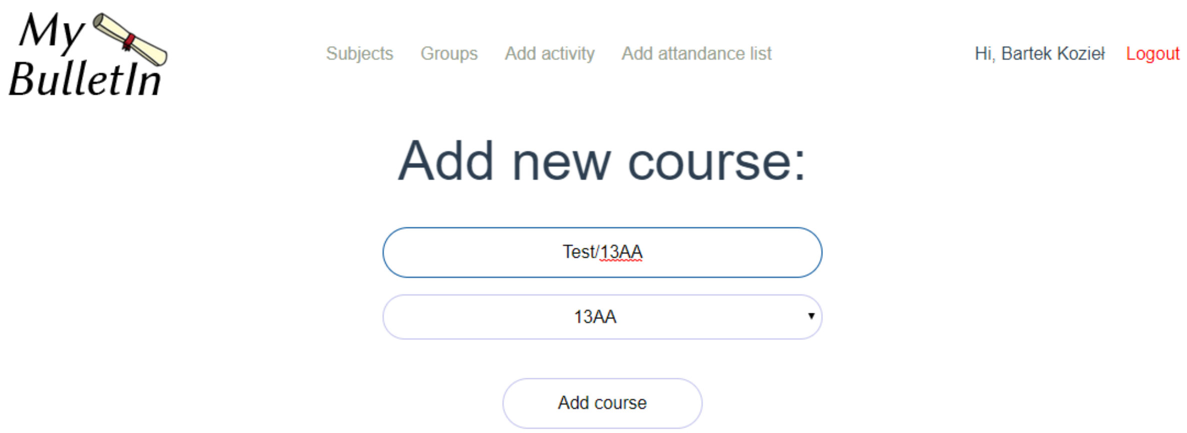
Id	Name	Tag id	Delete
1	Bartek Koziel	042E8DBADA5A80	✗
3	Yevhenii Lashutko	044826FA524E80	✗

Figure 36 Group participants



Id	Name	Group	Professor	Activities	Delete
18	WebapplicationsLecturee/12C	12C	Bartek Kozieł	more..	✖
20	Webapplications/12A	12A	Bartek Kozieł	more..	✖

Figure 37 Courses

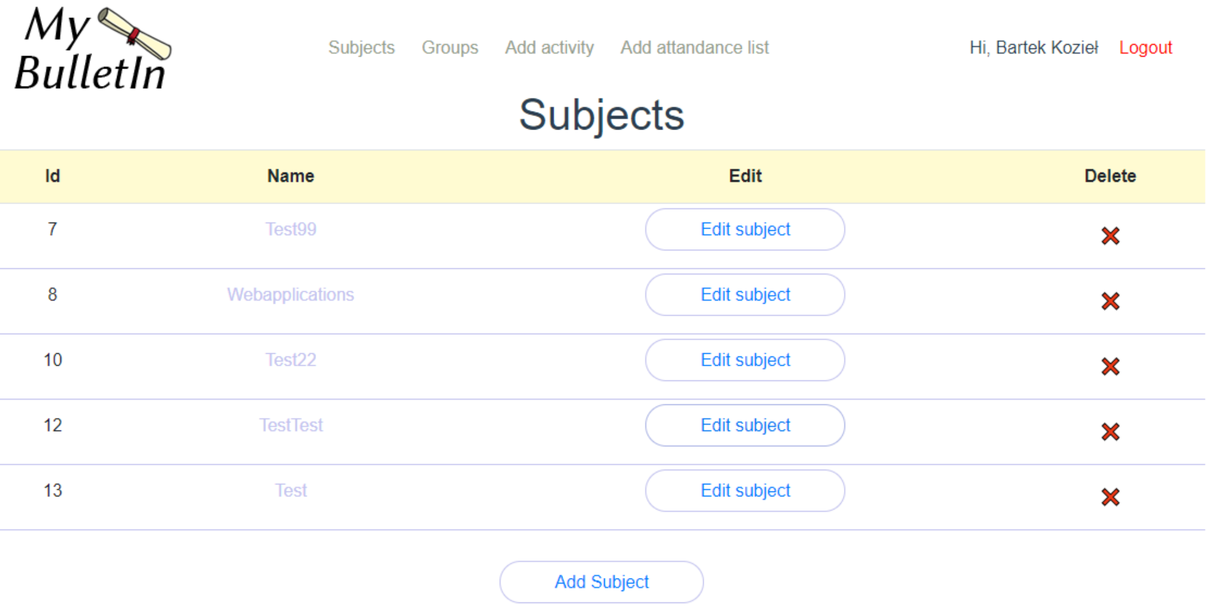


Test/13AA

13AA ▾

Figure 38 Add new course

To go to the course page, you need to click on the My Bulletin logo, which is located on the left side of the website header, after which the page will open on the website (see Figure 37). On this page you can delete the course, as well as create it. Let us begin proceed to the creation by clicking on the "Add course" button, after which a page will open (see Figure 38), where we should enter the name of the course, as well as select the needed group that will be linked to this course. The course name must be entered manually, and the group name is automatically added to the name using the delimiter "/".After entering the necessary data, we confirm the creation of a new course with the "Add course" button and the message should appear that the data was added to the server, if everything was done correctly. To return to the previous menu (see Figure 37), you need to click on the My Bulletin logo, which is located on the left side of the header of the site.



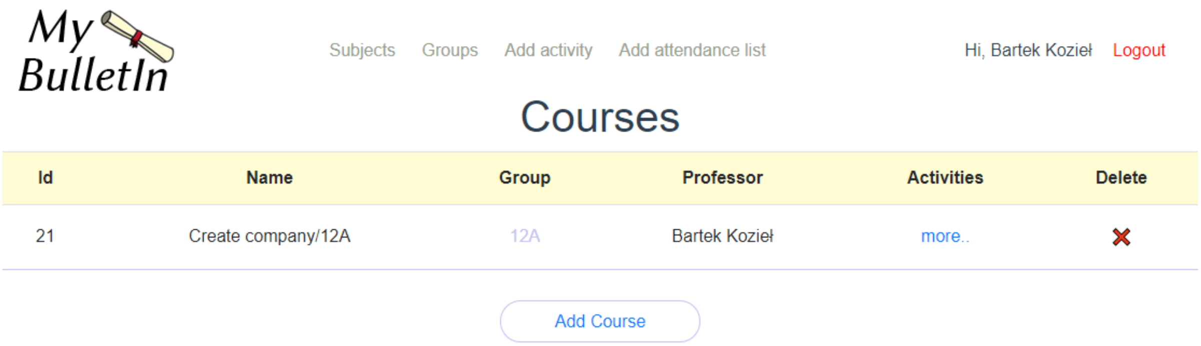
Subjects

Id	Name	Edit	Delete
7	Test99	Edit subject	✖
8	Webapplications	Edit subject	✖
10	Test22	Edit subject	✖
12	TestTest	Edit subject	✖
13	Test	Edit subject	✖

Figure 39 Subjects

Then go to the next page by clicking on "Subjects" in the header of the site, after which you will see the following (see Figure 39). On this page you can delete the subject, add and edit. When you click on "Add subject", you can to write only the name of the subject and confirm the changes by pressing the button, after which, if everything is done correctly, the inscription appears that subject has been added. To return to the previous page, click on the "Subjects" in the header of the site. The page with the editing of the subject looks the same as the creation of the subject, you can only change the name itself. You can also click on the red cross to remove an subject from this list.

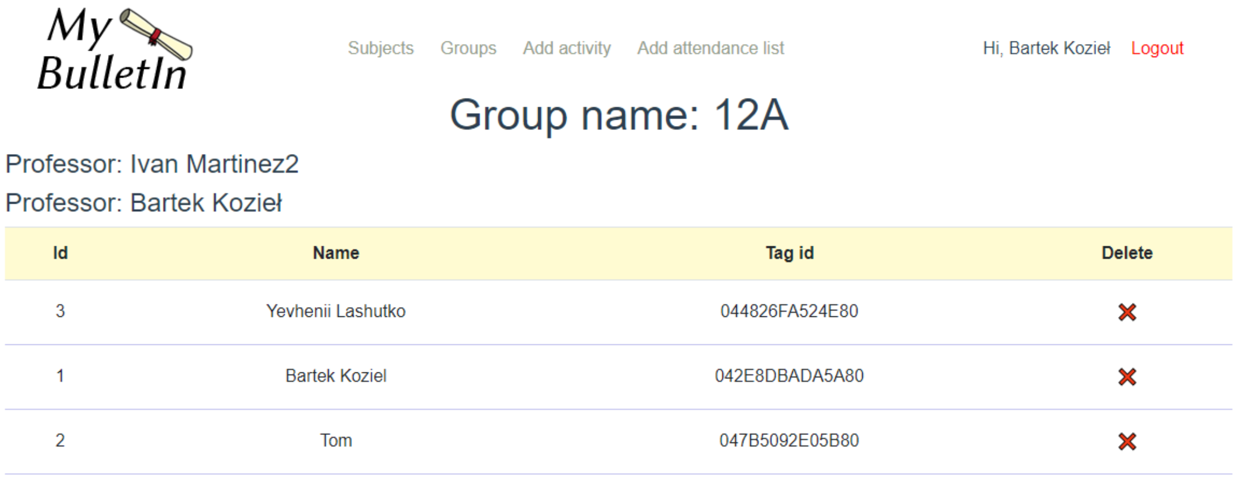
If you click on the name of the subject, for example, we take the subject "Create company" and click on it, the next page opens (see Figure 40), on which we see the Create company course with the already created group and the selected professor. If you click on the name or number of the group, the next page will open (see Figure 41), where you can see all the information about this group and the professors added to it. There is an opportunity to remove students from this group.



Courses

Id	Name	Group	Professor	Activities	Delete
21	Create company/12A	12A	Bartek Kozieł	more..	✖

Figure 40 Subject course



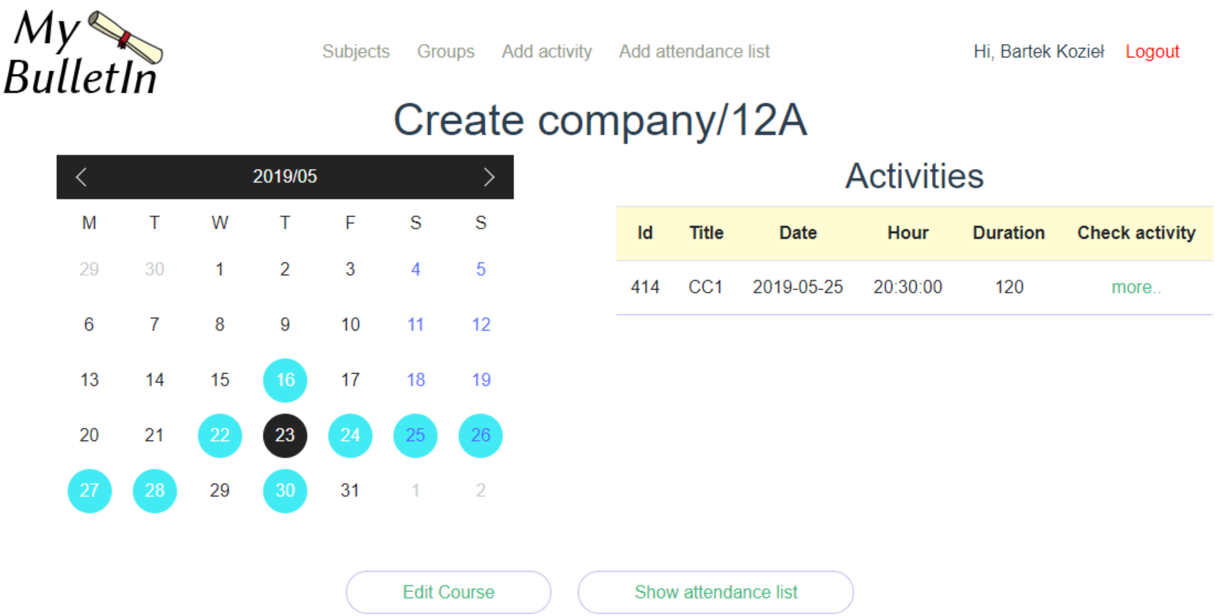
Group name: 12A

Professor: Ivan Martinez2
Professor: Bartek Kozieł

Id	Name	Tag id	Delete
3	Yevhenii Lashutko	044826FA524E80	✖
1	Bartek Koziel	042E8DBADA5A80	✖
2	Tom	047B5092E05B80	✖

Figure 41 Subject group info

When you go to the page (see Figure 40) on the inscription more.. this page opens (see Figure 42) on which the calendar is on the left side and the activities on the right side. Activities are the attendance list in which students should check in at the lesson, this is how the page looks like if you click on more .. (see Figure 43). Here we can see immediately the date when there will be a lesson and a list of students who need to be noted in the attendance list. There is a possibility to add a student and delete a student.



Create company/12A

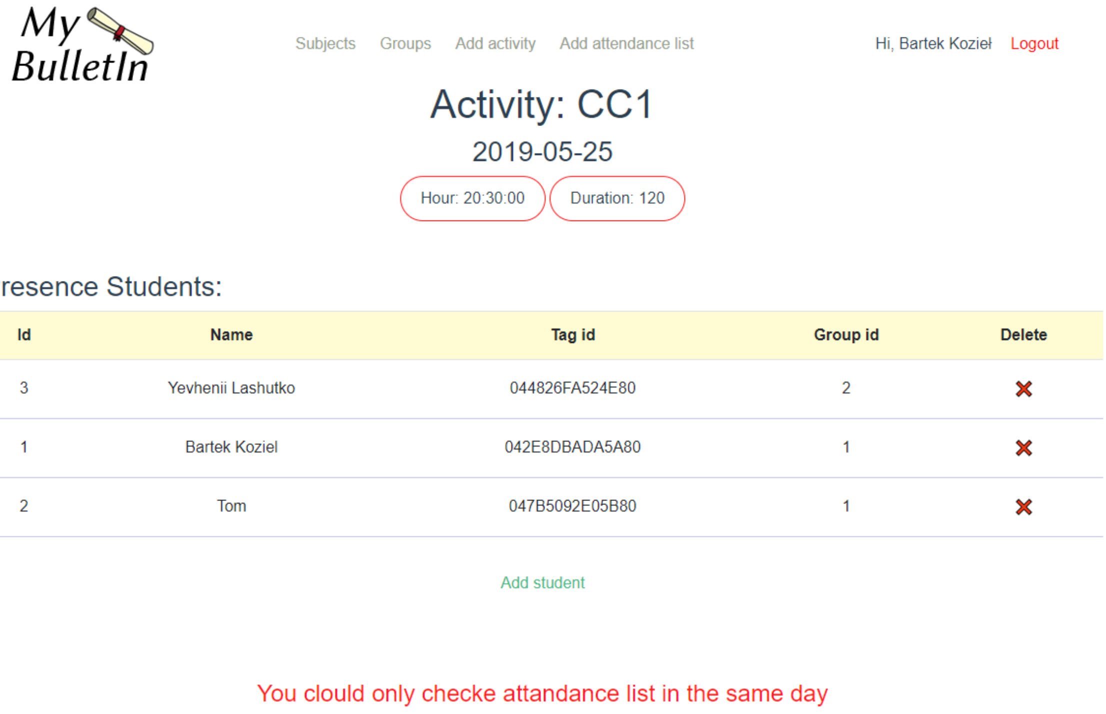
< 2019/05 >

M	T	W	T	F	S	S
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Activities

Id	Title	Date	Hour	Duration	Check activity
414	CC1	2019-05-25	20:30:00	120	more..

Figure 42 Course details



Activity: CC1

2019-05-25

Hour: 20:30:00 Duration: 120

Presence Students:

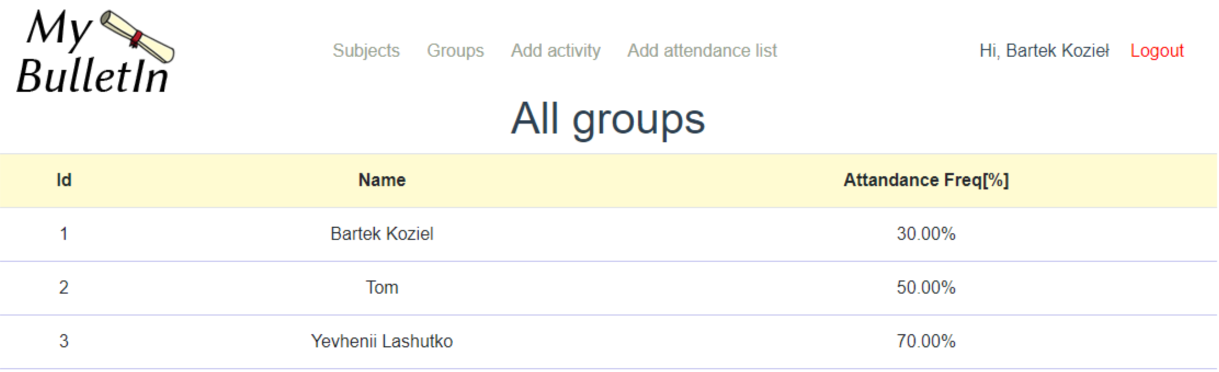
Id	Name	Tag id	Group id	Delete
3	Yevhenii Lashutko	044826FA524E80	2	✗
1	Bartek Koziel	042E8DBADA5A80	1	✗
2	Tom	047B5092E05B80	1	✗

You clould only checke attandance list in the same day

Figure 43 Activity detail

Returning to the page (see Figure 42), where we will select the "Show attendance list" and can see (see Figure 44) the percentage ratio of student attendance from all groups of this course.

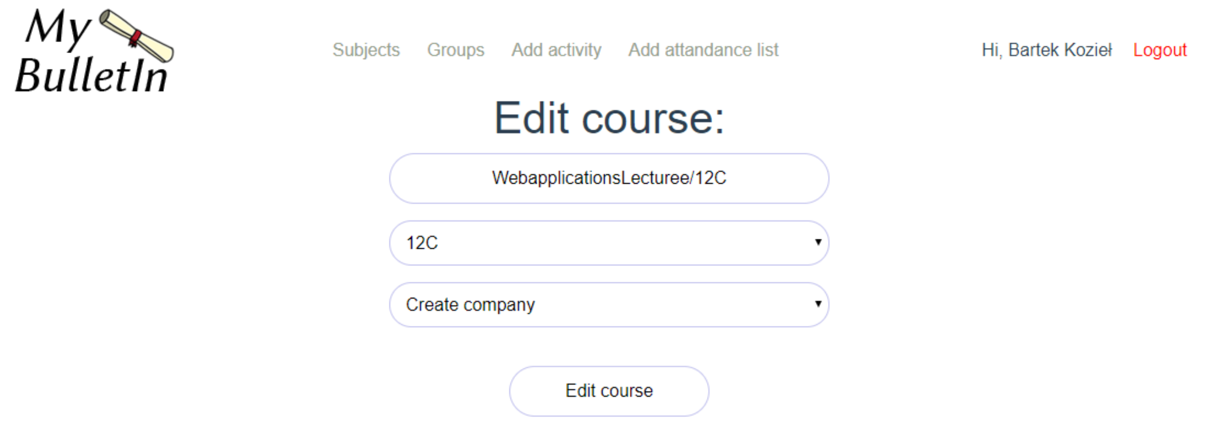
On the page (see Figure 42), if you select the "Edit course" button, we will see a page (see Figure 45) on which we can change the group and the subject to which this course is attached.



All groups

Id	Name	Attandance Freq[%]
1	Bartek Koziel	30.00%
2	Tom	50.00%
3	Yevhenii Lashutko	70.00%

Figure 44 All groups



Edit course:

WebapplicationsLecturee/12C

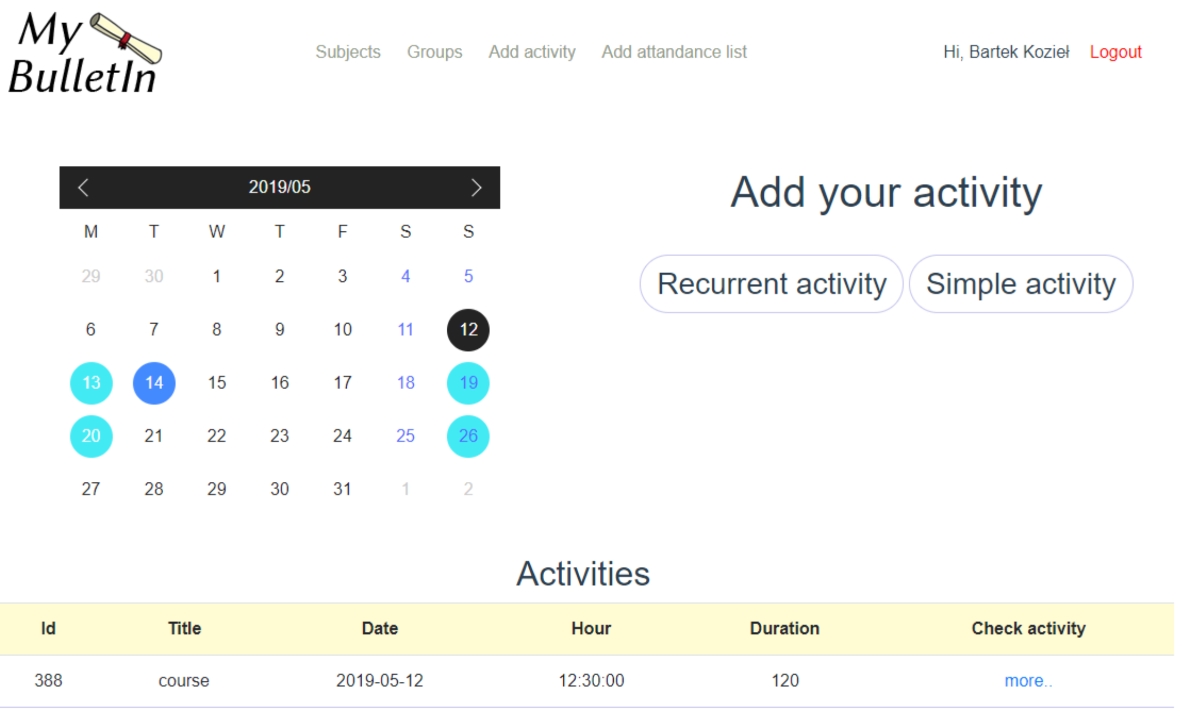
12C

Create company

Figure 45 Edit Course

When you select and click on "Add activity" in the header of the site, you will see (see Figure 46) the calendar on the left side and the opportunity to create an individual activity for the needed day and time.

An activity has already been created on the calendar in circles and when you click on the calendar, information about them appears as we see (see Figure 46) at the bottom of the screenshot.



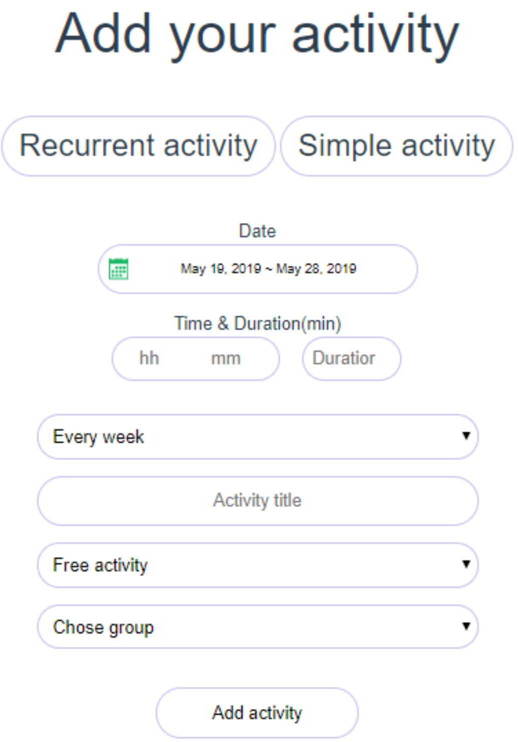
2019/05

M	T	W	T	F	S	S
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2

Activities

Id	Title	Date	Hour	Duration	Check activity
388	course	2019-05-12	12:30:00	120	more..

Figure 46 Add activity



Date: May 19, 2019 ~ May 28, 2019

Time & Duration(min): hh mm Duratior

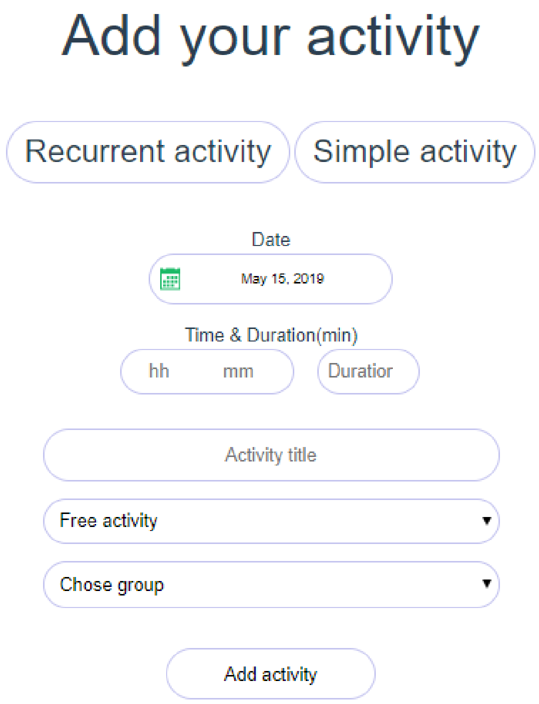
Every week

Activity title

Free activity

Chose group

Figure 47 Recurrent Activity



Date: May 15, 2019

Time & Duration(min): hh mm Duratior

Activity title

Free activity

Chose group

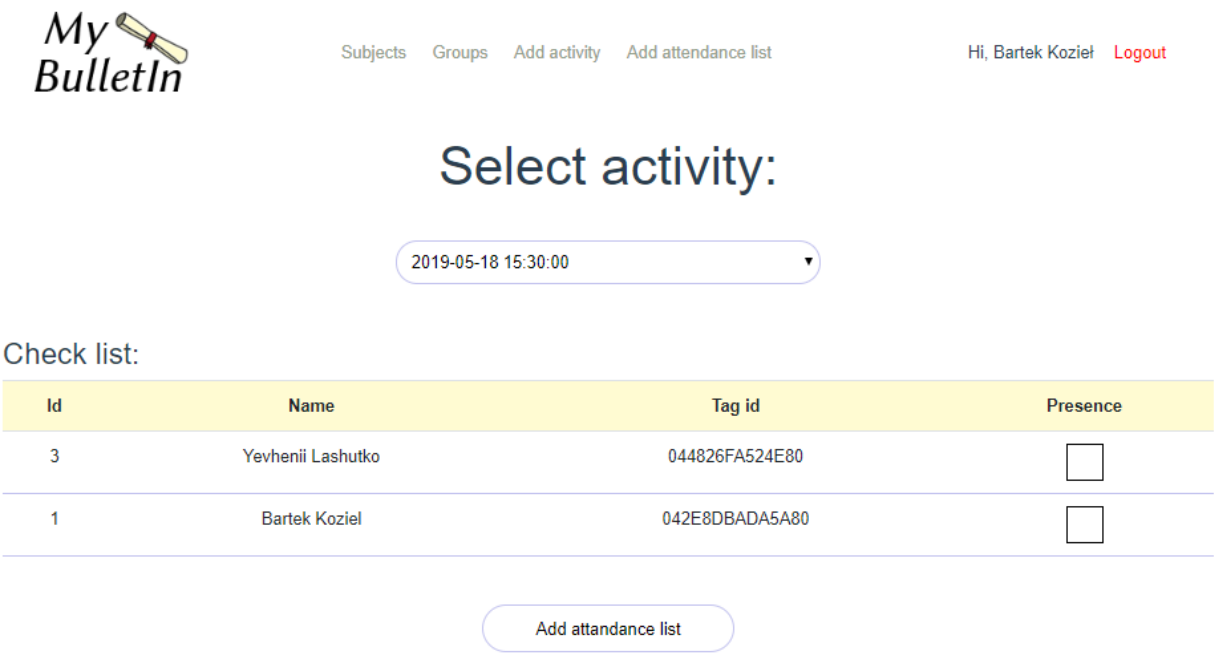
Figure 48 Simple Activity

On the page (see Figure 46) in the right part of "Add your activity" there are two buttons, this is "Recurrent activity" and "Simple activity".

In the Recurrent activity block (see Figure 47), we must select a period in the drop-down calendar by clicking on the needed dates. After that, you need to enter the time and duration of classes, as well as choose how often classes will occur every week or every day, this creates the needed activity. Next you need to write the name of this activity. You can do this as a separate activity to which you can link a specific group without a course, or choose a specific course with an already created group, after which you need to confirm all actions by clicking on the "Add activity" button.

In the Simple activity block (see Figure 48), you need to select 1 day on the drop-down calendar and choose the time and duration of the session. Further actions are the same as in the **Recurrent activity block**, it is possible to select an already defined course with an attached group or create an activity to the right group and then save it, whereupon all information will be displayed on the calendar which is located on the screenshot (see Figure 46).

Clicking on the more.. inscription on the page (see Figure 46) will open exactly the same page about which we spoke earlier (see Figure 43).



Select activity:

2019-05-18 15:30:00 ▾

Check list:

Id	Name	Tag id	Presence
3	Yevhenii Lashutko	044826FA524E80	□
1	Bartek Koziel	042E8DBADA5A80	□

Figure 49 Track students’ attendance list

Now we will go to the last page, to do this, select and click on “Add attendance list” in the header of the site and then the page will open (see Figure 49). On this page, first you must select the needed activity in the drop-down menu and then you will see the entire list of students who must attend classes. On the right side, we can see empty squares, and if we clicked in it, then it will be marked on the attendance list. To select a different activity, then you need to return to the drop-down menu and select a different date and time.