CSyllabus

Design Description Document

Version 1.0

Revision History

Date	Version	Description	Author
2017-10-23	0.01	Initial Draft	Emanuel Guberovic
2017-10-27	0.02	Introduction	Zvonimir Relja
2017-10-27	0.03	Architecture	Emanuel Guberovic
2017-10-27	0.04	Front end architecture	Zvonimir Relja
2017-11-08	1.0	Brushed up for a first public upload	Filip Turcinovic

Table of Contents

1. Introduction

CSyllabus is imagined as a web platform which should ease up process of finding and comparing courses on domestic and foreign faculties. It will enable users to discover and compare courses on interactive way through web application. This “one click” app will save time and provide very useful information to interested parties.

1.1 Purpose of this document

The purpose of this document is to expose the design decisions that have been made by team CSyllabus during the project which is being developed as part of the Distributed Software Development course simultaneously at Politecnico di Milano and FER Zagreb. The content of this document are not immutable: changes can happen during the implementation of the application and so new versions of this document will be produced.

1.2 Document organization

The document is organized as follows:

1. **Introduction** describes purpose of document, contents of this guide, intended audience and definitions and acronyms.
2. **Architecture and Design** describes architecture used in developing application and, in addition to that, shows diagrams so reader can understand why are architectural components and design choices taken.
3. **User interface mockups** shows graphical user interface of the web application separated in two chapters: initial mockups and new one.
4. **List of tables** shows list of tables used in this document.

1.3 Intended Audience

The intended audience is:

- Customer
- Team members
- Supervisors

Document represents the guide that team members of CSyllabus must use to develop the application. Also, besides team members, this document is for customer and supervisors. Their role is checking progress and read about the decisions the team has taken.

1.4 Scope

Scope of this document is to provide an insight into detailed design of the CSyllabus project. Database design, backend and frontend architecture is also explained.

1.5 Definitions and acronyms

1.5.1 Definitions

Keyword	Definitions
Django	It is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
REST API	Application endpoint which exposes all features from backend to frontend.
Django REST	Django REST framework is a powerful and flexible toolkit for building Web APIs.
Angular 4	Framework which allows to create complex, customizable, modern, responsive and user friendly web applications.
Postgres	One of the most advanced open source database.

1.5.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
FER	Faculty of Electrical Engineering and Computing, University of Zagreb
POLIMI	Politecnico di Milano, Italy
Backend	Refers to Django, Django REST and REST API
Frontend	Refers to Angular 4 framework
Git	Refers to version control system
Push/Pull	Refers to Git, push - code is in version control, ready to pull by other members of the team

1.6 References

More information about Csyllabus like more documentation and team members can be found here: http://www.fer.unizg.hr/rasip/dsd/projects/csyllabus_score

Project documentation:
http://www.fer.unizg.hr/rasip/dsd/projects/csyllabus_score/documents

2. Architecture and Design

2.1 Basic priorities and high-level system architecture

When choosing the project architecture, the fact that it's users are distributed yet they all have to have access to the same data was taken into consideration. The best solution for such as system considered is a web application with a central database. The advantage of web applications as opposed to let say mobile application is the ability to have the same system on all platforms with a single requirement that they have a WWW browser and access to Internet. This choice introduces savings in the project development because one developed product is used on multiple platforms.

More over the app has five main modules with distinct functionalities completely separated from each other. This besides being a good practice in the view of the overall system scalability is also good for the distributed incremental development process.

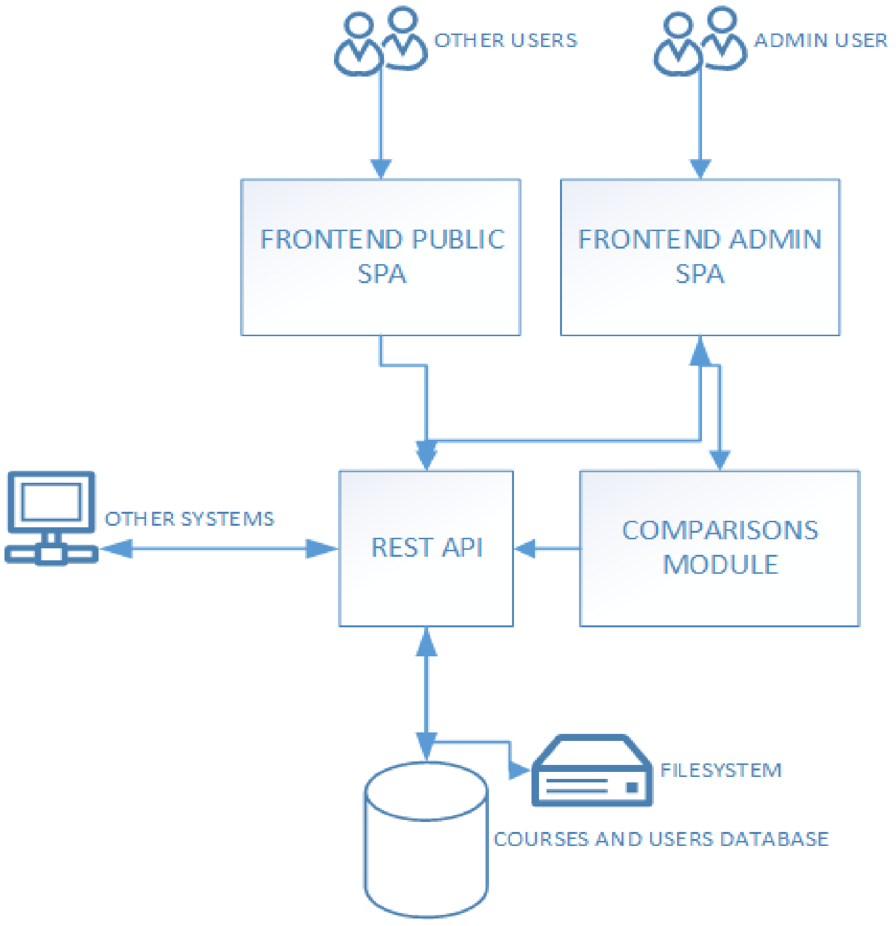


Illustration 1: System architecture diagram

Five modules are organised into three layers:

- **The presentation layer** is built with two separated single page application modules, one for public access and another for administration purposes. SPA modules are developed using Angular4+ framework in Typescript language.

- **The business logic layer** is separated into two modules, one serving as a REST interface to the data layer and the other as a module which on event runs course comparisons algorithms and using the REST interface caches the result to the data layer. Our business logic layer is built using Django framework with Python programming language.

- **The data layer** is composed of the SQL database and a server filesystem. The DBMS used in our app is PostgreSQL.

2.2 Component diagram

Component diagrams show system components and their interrelations. The following figure shows the realization of our backend system which is reachable to the frontend SPAs as well as other subsystems through it’s REST interface.

Faculty data management, course data management and search engine together serve the data used in syllabus exploration and comparison functionalities of the system. User post management and user management modules serve the socialization part of the system. Statistics viewer serves the statistics and log functionalities of the system. All of the API calls which require some form of authorization go through authentication module.

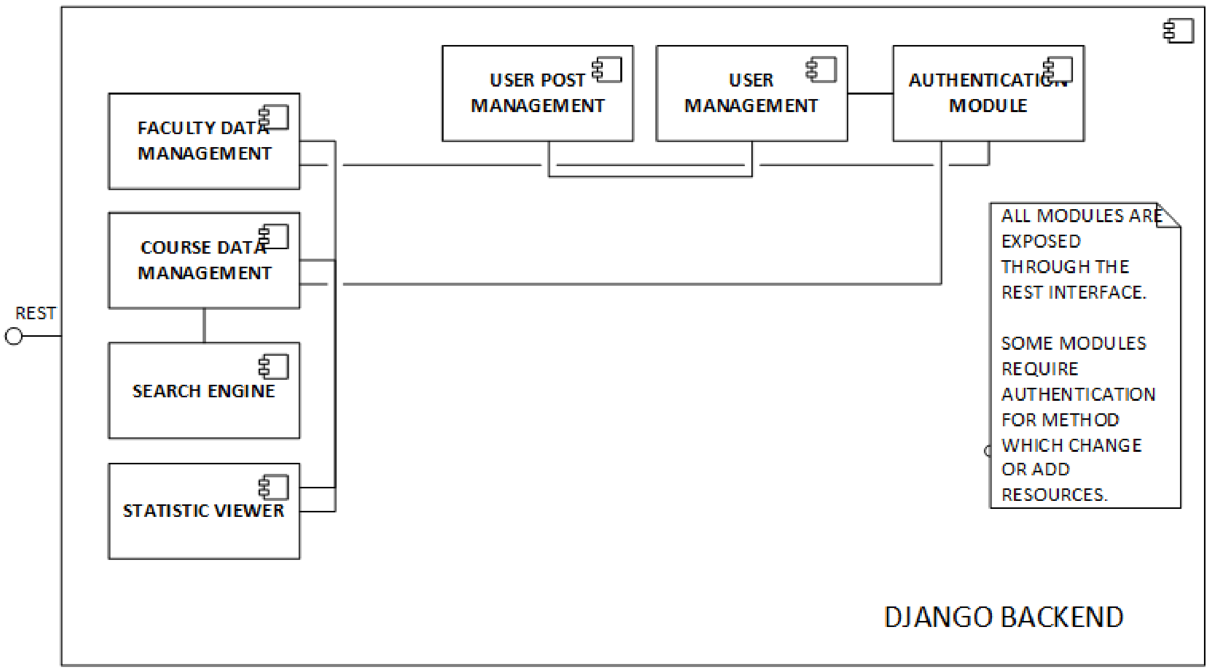


Illustration 2: Component diagram

2.2.1 User stories to architectural components mapping

ID	User Story	Component
US1	As a student I want to search for syllabi by name, faculty or country	Search Engine
US2	As a student I want to see syllabi details so I can see what it contains	Course Data Management
US3	As a student I want to see details of course so I can see what they offer	Course Data Management
US4	As a student I want to see details of faculty so I can know more about faculty	Faculty Data Management
US5	As a student I want to compare my syllabi with other syllabuses	Search Engine
US6	As a student I want to choose country and faculty I am studying at so I can choose my own courses for comparison	Search Engine
US7	As a student I want to see which are the best faculties for my chosen courses of interest	Search Engine
US8	As a student I want to see which faculties are most similar to my own so I can see what is my best option	Search Engine
US9	As a student I want to see other subjects that are relevant for me	Search Engine
US10	As a student I want to choose destination country and faculty so I can compare my courses with them.	Faculty Data Management
US11	As a student I want to see recommended courses so I can maybe come to know something new	Search Engine
US12	As a student I want to comment and evaluate a course	User Post Management
US13	As a student I want to share a syllabus in social media so others can see it	Third party APIs
US14	As a Professor I want to have option to add new syllabi to the database so that database can expand	Course Data Management
US16	As a User I want to have a profile with my personal data	User Management

US17	As a User I want to know what is the most searched faculty, subject and tag	Statistics Viewer
US18	As a Professor I want to see statistics data in graphic format	Statistics Viewer
US19	As a Guest I want to see the main functionalities without an account	Authentication Module

2.3 Deployment diagram

The Deployment Diagram is a static and structured view of computer resources essential to the functionality of the system. It consists of actual devices (other systems of human users), application components running (frontend public and admin single page applications, REST API, comparisons modules and the database and filesystem) and connections between these components (REST calls).

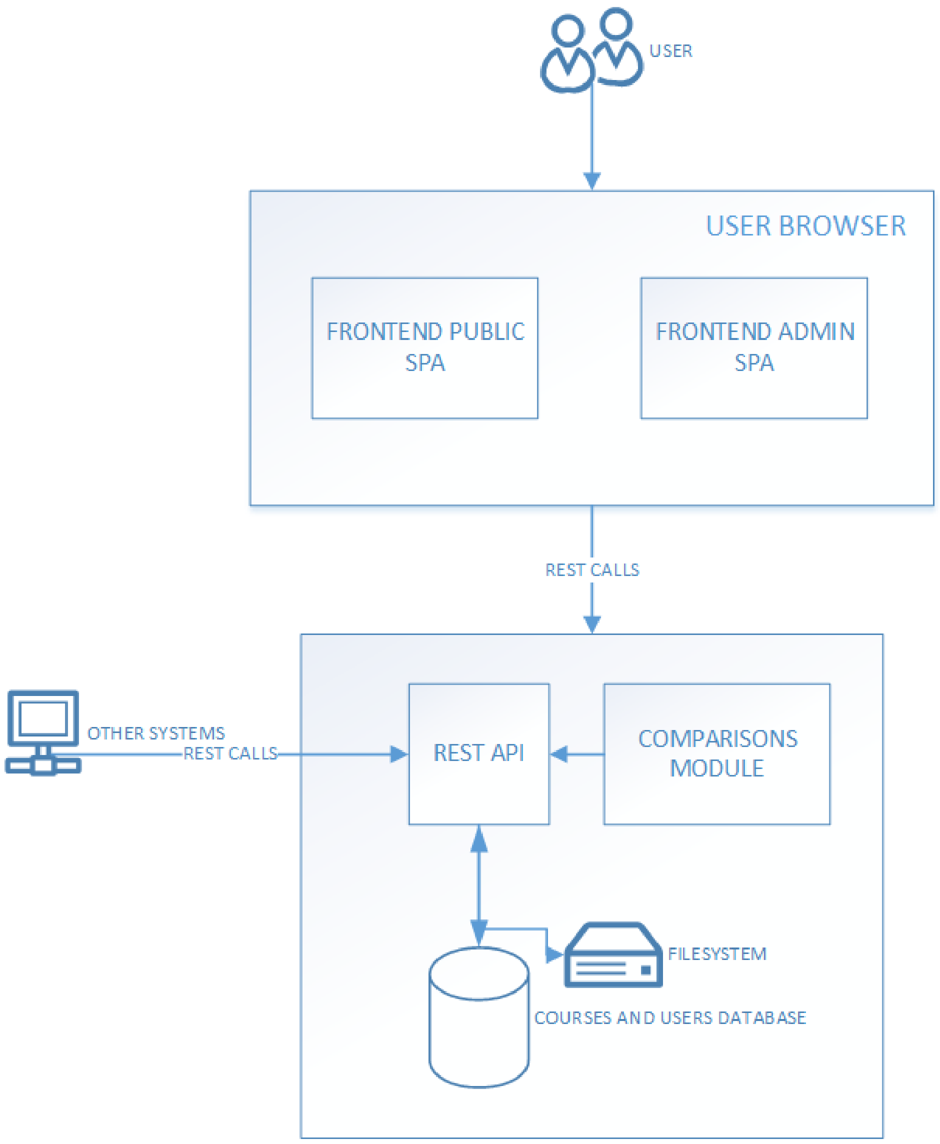


Illustration 3: Deployment diagram

2.4 REST API documentation

More detailed REST API documentation can be found in Swagger.

Resources list: country, university, faculty, program, course, course_comparison_results, user_post and user.
Nested resources: country/university, university/faculties, faculty/programs, program/courses and user/courses.

Authentication for calling resource methods which require a certain authorization is done using JSON Web Tokens.

2.5 Database model (ER diagram)

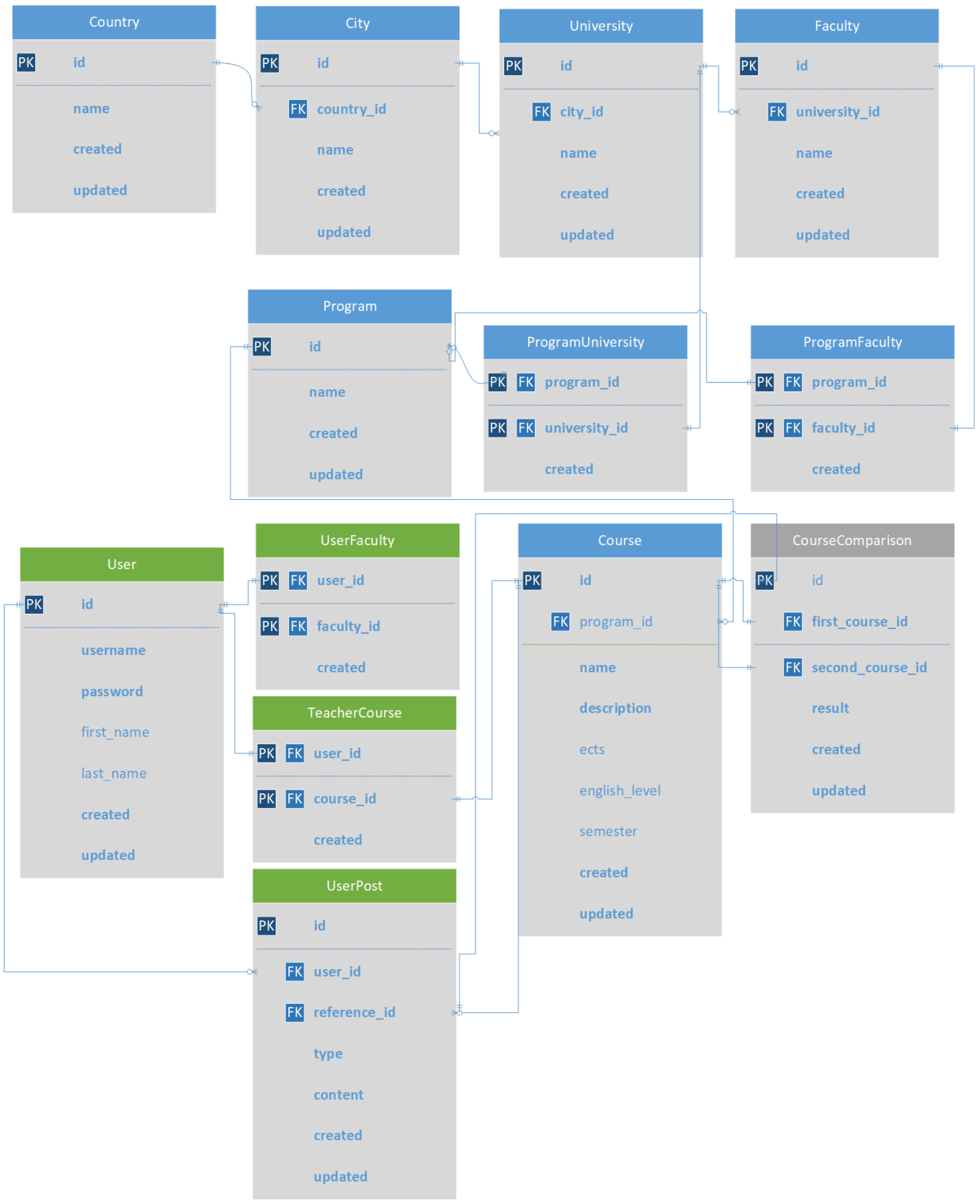


Illustration 4: ER Diagram

The ER diagram shows relationships of entities stored in the database. In light blue color we have entities which belong to faculty and course data management components, green colored are entities which are a part of user related components (user management, user post management and authentication components). In grey color we have an entity which belong to the search engine component. All of these entities coexist together and have relationships between them weather they belong to same components or.

Relationship cardinalities:

- Country to City is 1 to 0 or more
- City to University is 1 to 0 or more
- University to Faculty is 1 to 0 or more
- University to ProgramUniversity is 1 to 0 or more
- Faculty to ProgramFaculty is 1 to 0 or more
- Program to ProgramUniversity is 1 to 0 or 1
- Program to ProgramFaculty is 1 to 0 or 1
- Program to Course is 1 to 0 or more
- User to UserFaculty is 1 to 0 or more
- User to TeacherCourse is 1 to 0 or more
- User to UserPost is 1 to 0 or more
- Faculty to UserFaculty is 1 to 0 or more
- Course to TeacherCourse is 1 to 0 or more
- Course to UserPost is 1 to 0 or more
- CourseComparison to UserPost i 1 to 0 or more
- CourseComparison to Course is 1 to 2

Entities:

- Country holds the data of a country for which there is course data in the system. It is connected to syllabuses through the relations with city entities.
    - Attributes:
        - id: auto-increment primary key
        - name: name of the country
        - created and updated: timestamp atribute for logging changes in the system
- City holds the data of a city for for which there is course data in the system. It is connected to syllabuses through the relations with university entities.
    - Attributes:
        - id: auto-increment primary key
        - name: name of the city
        - country_id: foreign key to country entity
        - created and updated: timestamp atribute for logging changes in the system
- University holds the data of a university for for which there is course data in the system. It is connected to syllabuses either through programs entities or through a relationship with faculty entities.
    - Attributes:
        - id: auto-increment primary key

        - name: name of the university
        - city_id: foreign key to city entity
        - created and updated: timestamp atribute for logging changes in the system
- Faculty holds the data of a faculty for which there is course data in the system. It is connected to syllabuses directly through programs entities.
    - Attributes:
        - id: auto-increment primary key
        - name: name of the university
        - university_id: foreign key to university entity
        - created and updated: timestamp atribute for logging changes in the system
- Program holds the data of a faculty for which there is course data in the system. It is connected to syllabuses directly through programs entities.
    - Attributes:
        - id: auto-increment primary key
        - name: name of the university
        - created and updated: timestamp atribute for logging changes in the system
- Course holds the syllabus data of the course.
    - Attributes:
        - id: auto-increment primary key
        - name: name of the university
        - description: course description
        - english_level: level of English at which course is taught
        - ects: ECTS credit value of the course
        - semester: semester at which course is taught
        - created and updated: timestamp atribute for logging changes in the system
- User holds the user login data.
    - Attributes:
        - id: auto-increment primary key
        - username: username to be used to log in
        - password: hashed version of a password used to log in
        - firstn_name: user's first name
        - last_name:users's last name
        - created and updated: timestamp atribute for logging changes in the system
- UserPost holds the user post data.
    - Attributes:
        - id: auto-increment primary key
        - user_id: foreign key to user
        - reference_id: foreign key to entity it is posted to
        - type: post type (types to be decided later)
        - content: post's content
        - created and updated: timestamp atribute for logging changes in the system

Entities ProgramUniversity, ProgramUniversity, UserFaculty, TeacherCoursehold two different foreign keys to which they reference and timestamp attributes.

2.6 Frontend architecture

2.6.1 Angular 4 + Material

CSyllabus web application will be developed using Angular 4 framework. Angular 4 is a Typescript based frontend web application framework for building client applications in HTML and Typescript that compiles to JavaScript. Framework combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Our choice was version 4 of Angular because it is backward compatible with version 2 but has some new features such as HttpClient library for HTTP Requests. Angular 5 was released on November 1, 2017.

With Angular 4 we will use Material UI. Angular Material is a set of modern UI components designed by the Angular team and based on Google's Material design specification. Google describes Material design as a set of guidelines, icons, and components that combine to create a unified user experience across platforms.

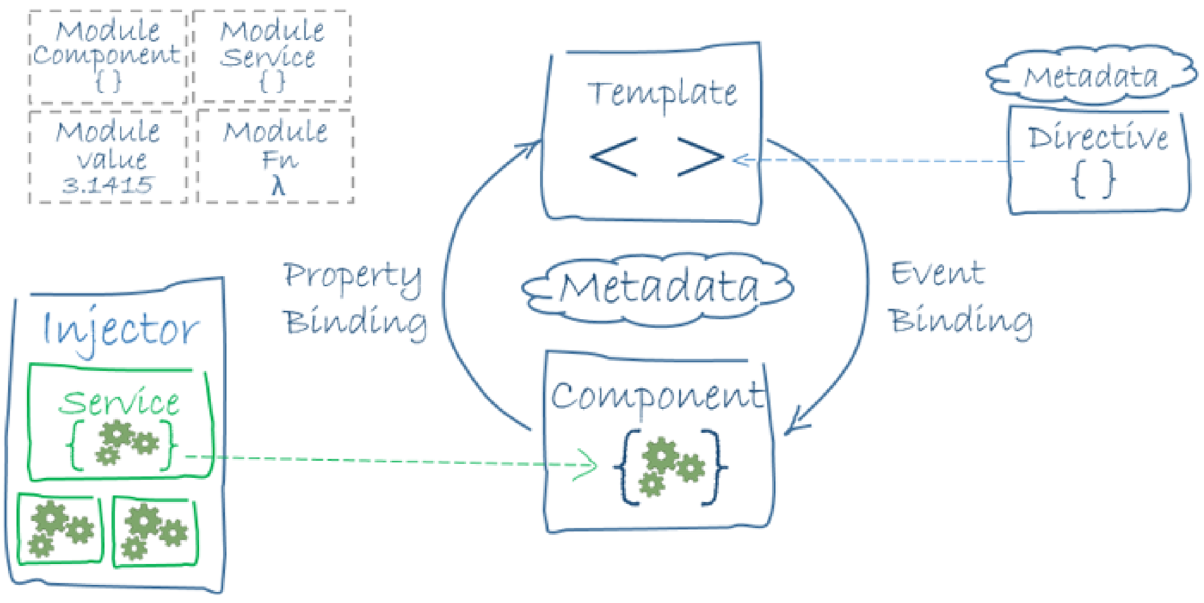


Illustration 5: Angular 4 architecture overview

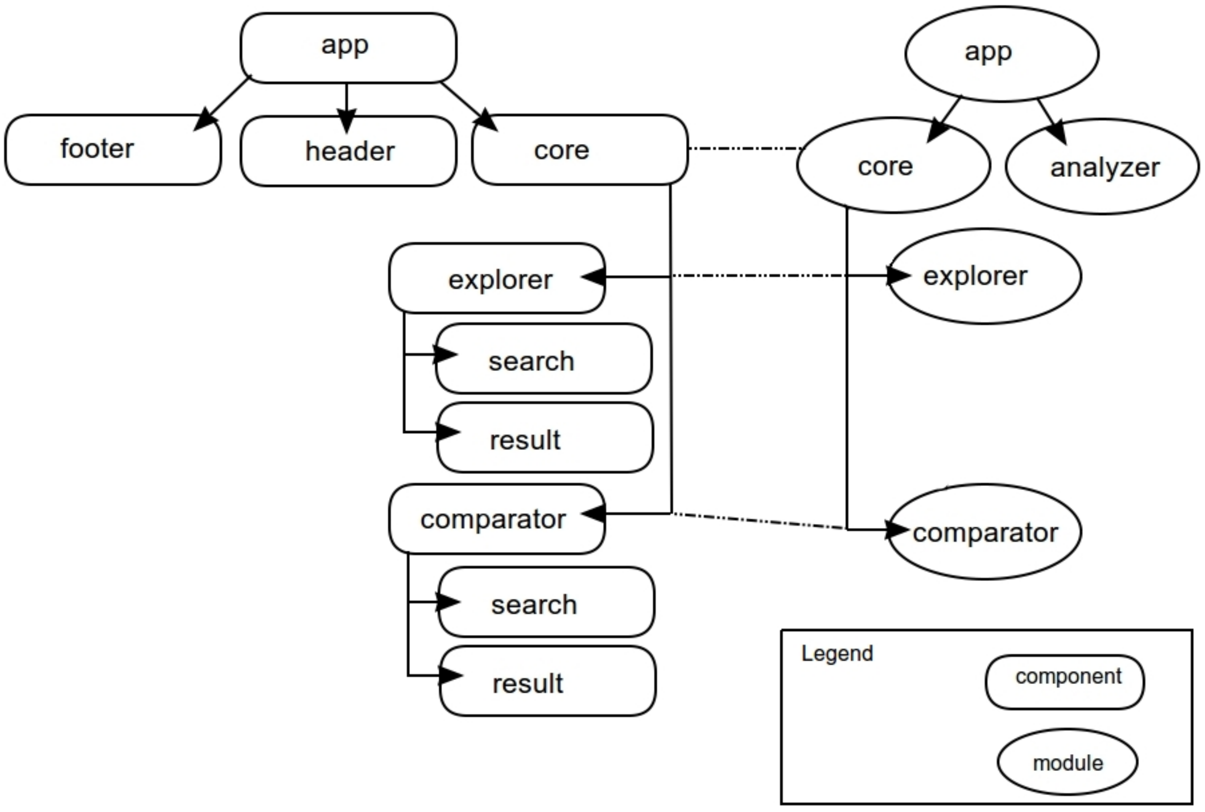


Illustration 6: Angular 4 organization diagram

2.6.2 Components

Components are the most basic building block of an UI in an Angular application. An Angular application is a tree of Angular components. A component controls a patch of screen called a view. CSyllabus modules will contain multiple components. If component is reusable, for example list of faculties, it will go to SharedModules, and will be reused throughout the whole application. Components we plan to have (as shown on illustration 6) are AppComponent, FooterComponent, HeaderComponent, CoreComponent which has 2 children components ExplorerComponent and ComparatorComponent and those two each have SearchComponent and ResultComponent.

2.6.3 Modules

Angular applications are modular and Angular has its own modularity system called *NgModules*. Every Angular application has at least one NgModule class, the root module, conventionally named AppModule. While the *root module* may be the only module in a small application, most applications have many more *feature modules*, each a cohesive block of code dedicated to an application domain, a workflow, or a closely related set of capabilities. CSyllabus will have multiple modules divided by features. For now we are going to have AppModule, CoreModule, AnalyzeModule, ExplorerModule and ComparatorModule as shown on illustration 6 and additionally SharedModule.

2.6.4 Services

As said, components should focus on presenting data and not fetch or save data directly. For those tasks we use services. Services are a great way to share information among classes. CSyllabus web application will have services which get data from API connected to database. Services used in application are CountriesService, CitiesService, UniversitiesService, FacultiesService, ProgramsService and CoursesService.

3. User interface mockups

3.1 Initial mockups

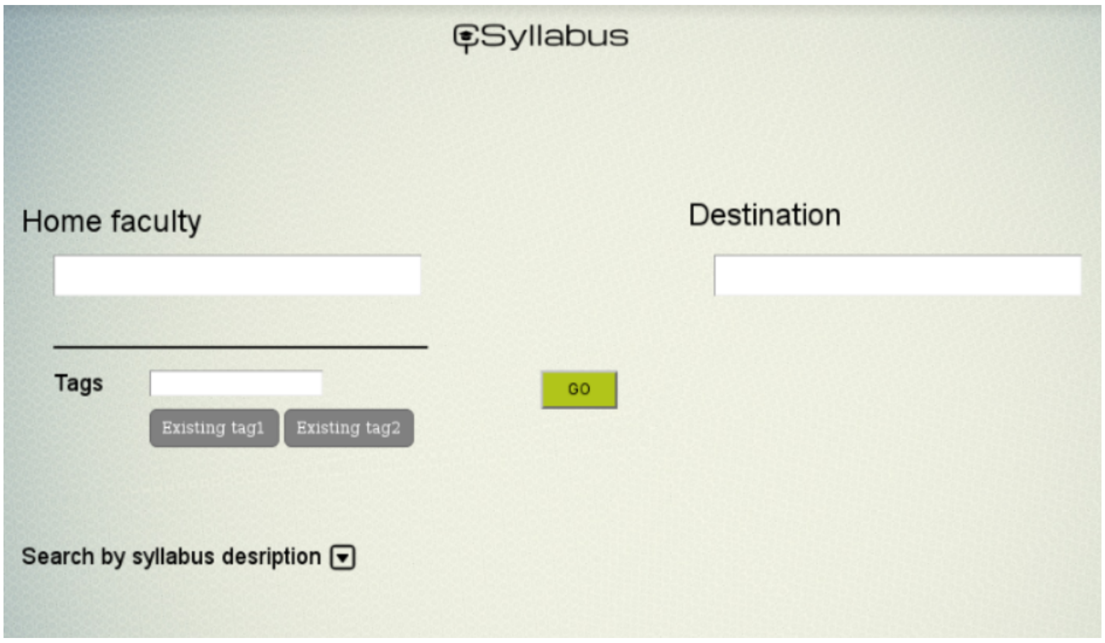


Illustration 7: Initial mockup 1

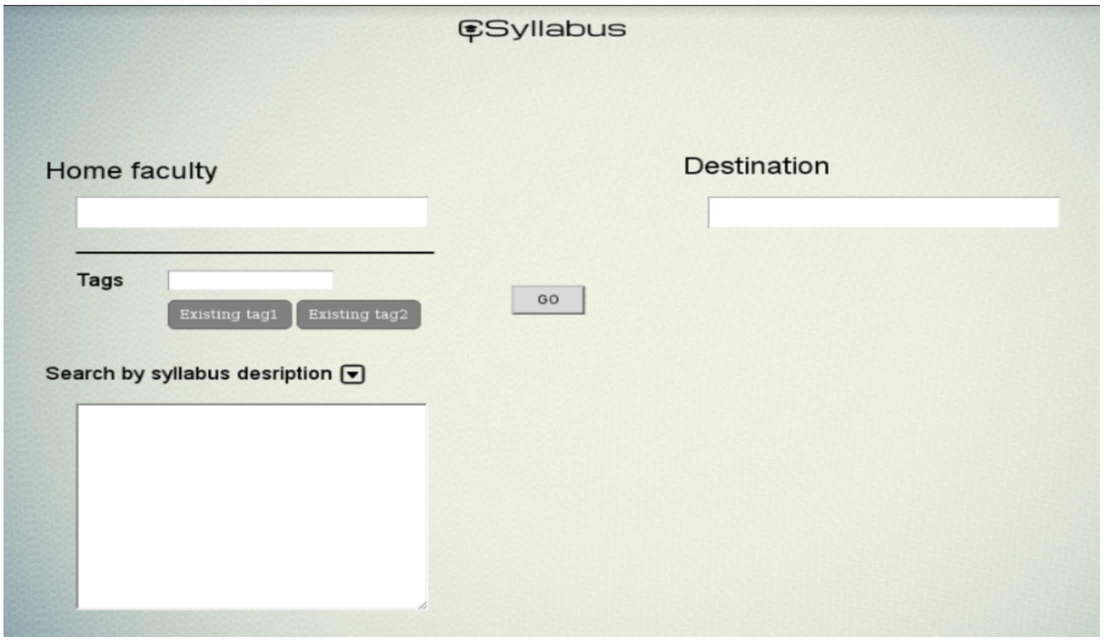
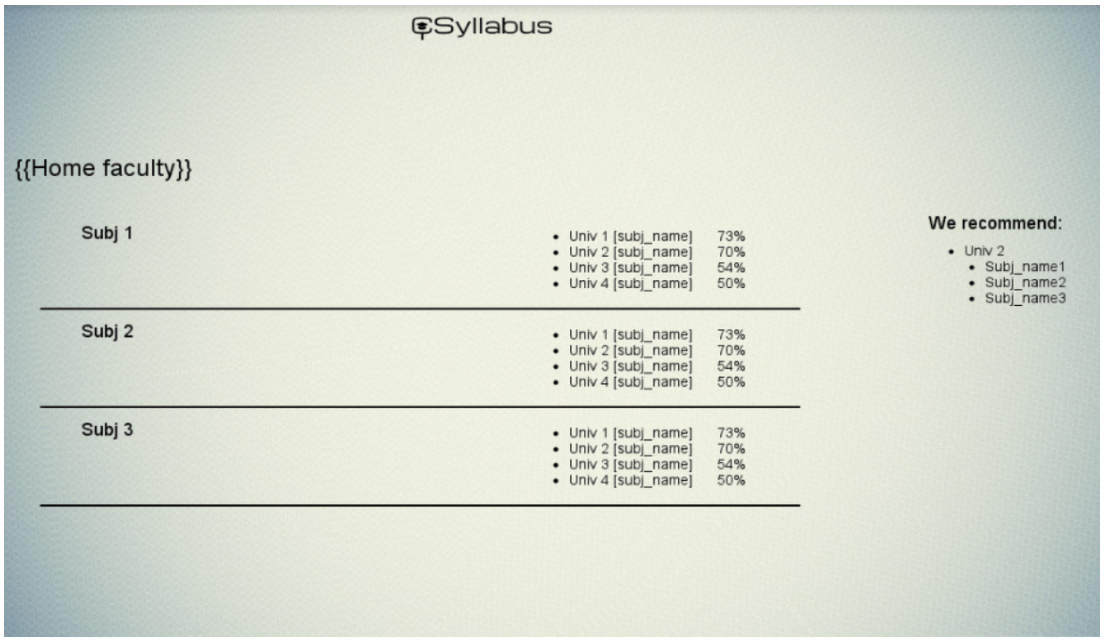


Illustration 8: Initial mockup 2



CSyllabus

{{Home faculty}}

Subj 1

- Univ 1 [subj_name] 73%
- Univ 2 [subj_name] 70%
- Univ 3 [subj_name] 54%
- Univ 4 [subj_name] 50%

Subj 2

- Univ 1 [subj_name] 73%
- Univ 2 [subj_name] 70%
- Univ 3 [subj_name] 54%
- Univ 4 [subj_name] 50%

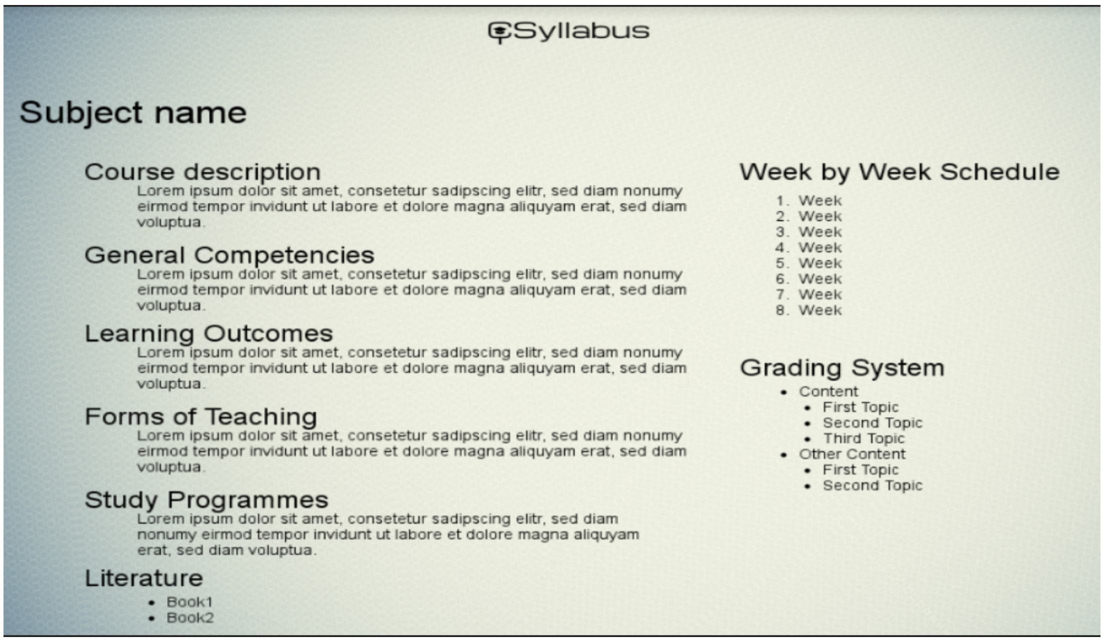
Subj 3

- Univ 1 [subj_name] 73%
- Univ 2 [subj_name] 70%
- Univ 3 [subj_name] 54%
- Univ 4 [subj_name] 50%

We recommend:

- Univ 2
  - Subj_name1
  - Subj_name2
  - Subj_name3

Illustration 9: Initial mockup 3



CSyllabus

Subject name

Course description
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

General Competencies
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Learning Outcomes
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Forms of Teaching
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Study Programmes
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.

Literature

- Book1
- Book2

Week by Week Schedule

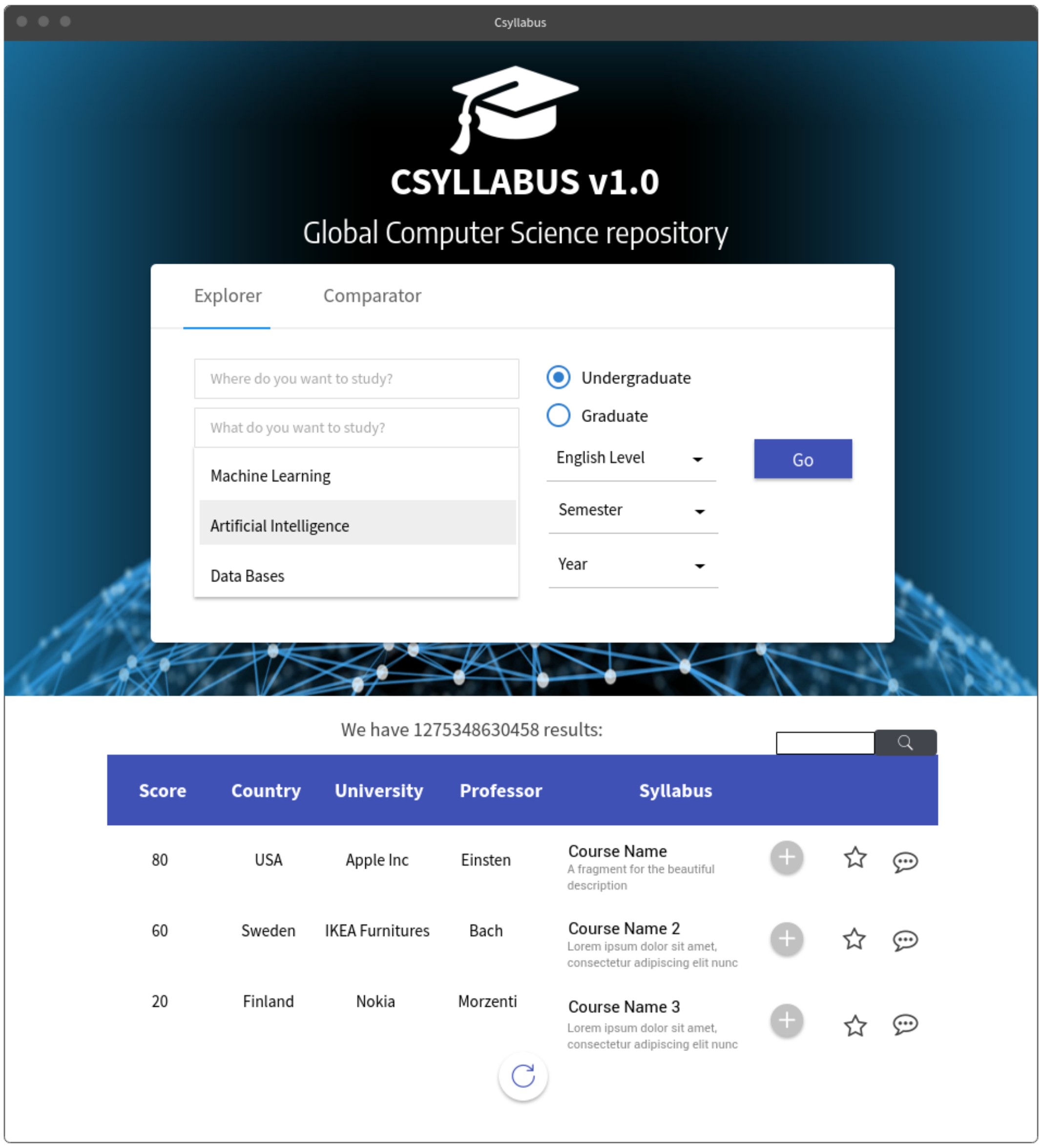
1. Week
2. Week
3. Week
4. Week
5. Week
6. Week
7. Week
8. Week

Grading System

- Content
  - First Topic
  - Second Topic
  - Third Topic
- Other Content
  - First Topic
  - Second Topic

Illustration 10: Initial mockup 4

3.2 New mockups



We have 1275348630458 results:

Score	Country	University	Professor	Syllabus
80	USA	Apple Inc	Einsten	Course Name A fragment for the beautiful description
60	Sweden	IKEA Furnitures	Bach	Course Name 2 Lorem ipsum dolor sit amet, consectetur adipiscing elit nunc
20	Finland	Nokia	Morzenti	Course Name 3 Lorem ipsum dolor sit amet, consectetur adipiscing elit nunc

Illustration 11: New mockup 1

4. List of Tables

Index of Tables