# 

# 

# 

# 

CSYLLABUS

**Requirements definition**

Draft

Version 0.9.0

# 

# 

# 

# 

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 2017-10-23 | 0.1 | Initial Draft | Sebastian Mayoral |
| 2017-10-23 | 0.2 | Initial Ideas and backlog | Matej Vukosav |
| 2017-10-24 | 0.3 | General writing | Sebastian Mayoral |
| 2017-10-25 | 0.4 | Description Use Cases and Stories | Sebastian Mayoral |
| 2017-11-08 | 0.5 | Use cases and user stories re-write | Matej Vukosav |
| 2017-11-10 | 0.6 | Revision | Sebastian Mayoral |
| 2017-11-14 | 0.7 | Revision and data update | Matej Vukosav |
| 2017-11-18 | 0.7.1 | User stories revision and field template creation | Matej Vukosav |
| 2017-12-06 | 0.8.0 | Improved introduction, domain analysis and non-functional requirements | Matej Vukosav |
| 2018-01-07 | 0.9.0 | Removed abandoned user stories: US5, US7, US15, US16, US17 | Matej Vukosav |

Contents

[1 INTRODUCTION 1](#_Toc500367400)

[1.1 Purpose of this document 1](#_Toc500367401)

[1.2 Intended audience 1](#_Toc500367402)

[1.3 Scope 1](#_Toc500367403)

[1.4 Document Structure 1](#_Toc500367404)

[1.5 Definitions and Acronyms 2](#_Toc500367405)

[2 DOMAIN ANALYSIS 3](#_Toc500367406)

[2.1 Problem Description 3](#_Toc500367407)

[2.2 Goals 3](#_Toc500367408)

[2.3 Domain 4](#_Toc500367409)

[3 FUNCTIONAL REQUIREMENTS 4](#_Toc500367410)

[3.1 Users 4](#_Toc500367411)

[3.2 User Stories 4](#_Toc500367412)

[3.3 User Story Descriptions 6](#_Toc500367413)

[3.4 Use Cases Diagram 13](#_Toc500367414)

[3.5 Use Cases Descriptions 13](#_Toc500367415)

[4 NON-FUNCTIONAL REQUIREMENTS 18](#_Toc500367416)

[4.1 Usability and Portability 18](#_Toc500367417)

[4.2 Availability 18](#_Toc500367418)

[4.3 Privacy 18](#_Toc500367419)

[4.4 Performance 19](#_Toc500367420)

[4.5 Interoperability 19](#_Toc500367421)

# 

# 1 INTRODUCTION

## 1.1 Purpose of this document

The purpose of this document is to analyze the project domain and to define the functional and nonfunctional requirements. In this document will be described what is supposed to develop viewed from the user perspective. This is a crucial part of the development being that it permits to clarify and understand project domain.

## 1.2 Intended audience

* Development team, as a guidance during the development activities and for the team to ensure they understand the requirements of the project.
* The supervisors who can use this document to understand the future process of the project.
* The customer who can ensure that all the requirements are captured by the team.

## 1.3 Scope

This document provides high level description of requirements for the CSyllabus project. Requirements definition focus on what to do and not how to do it. Also it will provide both functional and nonfunctional requirement descriptions based on client inputs along with UML diagrams.

## 1.4 Document Structure

The document is organized in three parts:

* Domain analysis: describes the problem and analyzes the domain from different user perspectives.
* Functional requirements: defines the requirements in a form of user stories. If user story does not provide enough details, there will be UML diagrams to ensure a clear assignment.
* Non-functional requirements: describes requirements that shows how the system should work. They serve as constraints or restrictions of the system design that should be met. Defines system attributes such as availability, security, privacy and performance.

## 1.5 Definitions and Acronyms

* User *noun*: single name for all different roles of the system. User can also be a student, professor or a guest depending on the context.
* App *noun*: refers to CSyllabus software.
* User story *noun:* short description of functionality told from user’s perspective
* Use Case *noun*: list of actions or event steps typically defining the interactions between a role and a system. Can include various requirements.
* Actor *noun*: who interacts with the app.
* System *noun:* set of components working together
* Graph *noun*: shows data distribution in an intuitive form where each node represents a field.
* Administrator *noun*: who can manage the server and the app.
* DSD: Distributed Software Development
* Syllabus *noun:* an academic document that contains faculty information’s including detailed list of courses available at the faculty
* Program *noun:* a list of courses in a particular subject (*e.g Computer Science*)
* Course *noun:* shaped area of knowledge studied on university

# 2 DOMAIN ANALYSIS

## 2.1 Problem Description

In today’s modern world where educational boundaries slowly disappear transit of students from one university to another is increasing. Most of time they want to conduct an experience of studying abroad. Often different county includes different language or culture what can be challenging or even too hard to overcome. Exploring or choosing suitable faculties involves adaptation to different searching tools, sites and portals. Their mechanism of showing relevant data can be confusing and not understandable at first and demand exhaustive work to get familiar with. Even if they found similar faculty to their own or one that matches their preferences comparing available courses can be challenging. Such data are not centralized and easily offered as they should be. To remove unnecessarily work and unpleasant experience before they even start this project aim to enable and provide all relevant information’s and insure that they choose best possible destination for their further education.

## 2.2 Goals

G1. To assist students in making good educational choices

G2. To assist instructors in sharing best practices

G3. To assist education researchers in understanding the evolution of our field

## 2.3 Domain

* Students are interested in information’s about other universities and their study plans. Students want to go to the best university that suits their needs to do so, they need to get information’s about other faculties and their courses. Because foreign faculties are not often well known, and courses can have different names they will have opportunity to compare their familiar courses with available.
* Professors are in constant research in the academic world. A good professor is updated about the new ways to teach others. However, often he lacks quality information’s and good feedback on his work. Professors can have possibility to see other feedbacks on their work in a view of comments and votes.

# 3 FUNCTIONAL REQUIREMENTS

## 3.1 Users

It is important to clarify that the actors mentioned in this document are in scope of User. User is high level of abstraction representing a single role in the system.

* Guest: is a person that belongs to the academic community who does not have an account registered in CSyllabus system. Guest is user who has high level of overview on the main functionalities of the system but has no user-identification-required features available such as user profile or can build user engagement.
* Student: is the registered user with verified CSyllabus account. He has all features as a guest plus user-identification-required features such as user profile and possibility to leave comments or vote for course they are familiar with.
* Professor: is the registered user with verified CSyllabus account. He has all features as a guest plus user-identification-required features such as user profile except possibility to leave comments or vote for course they are familiar with. Professors can learn from comments and use them to improve their personal skills. Professors can have option to upload new syllabus.

## 3.2 User Stories

|  |  |  |
| --- | --- | --- |
| **ID** | **Name** | **User Story** |
| US1 | Syllabus search | As a student I want to search for syllabi by name, faculty or country. |
| US2 | Syllabus details | As a student I want to see syllabi details, so I can see what it contains. |
| US3 | Course details | As a student I want to see details of course so I can see what they offer |
| US4 | Faculty details | As a student I want to see details of faculty, so I can know more about faculty. |
| US6 | Country and faculty choose | As a student I want to choose country and faculty I am studying at, so I can choose my own courses for comparison |
| US8 | Similar faculty | As a student I want to see which faculties are most similar to my own, so I can see what is my best option |
| US9 | Relevant subjects | As a student I want to see other subjects that are relevant for me |
| US10 | Destination country and faculty | As a student I want to choose destination country and faculty, so I can compare my courses with them. |
| US11 | Recommended courses | As a student I want to see recommended courses, so I can maybe come to know something new |
| US12 | Course comments | As a student I want to comment and evaluate a course |
| US13 | Syllabus share | As a student I want to share a course in social media so others can see it |
| US14 | Add syllabus to database | As a Professor I want to have option to add new syllabi to the database so that database can expand |
| US18 | Guest view | As a Guest I want to see the main functionalities without an account |

## 3.3 User Story Descriptions

|  |  |
| --- | --- |
| **Story ID** | US1 |
| **Story** | As a student I want to search for syllabi by name, faculty or country. |
| **Source** | Team |
| **Detailed Description** | Student wants to search through the syllabuses using a filter. This filter can be the subject name, technology or any course related value, for example “Artificial Intelligence”. Also, Student can use a faculty name to refer one University specifically or one country. When User does this, the app should show him a list of the syllabuses according to given criteria. |
| **Validation Criteria** | * The User can choose the search criteria * The app provides the results according the chosen criteria |

|  |  |
| --- | --- |
| **Story ID** | US2 |
| **Story** | As a student I want to see syllabi details, so I can see what it contains. |
| **Source** | Team |
| **Detailed Description** | User can see all data that single syllabus provides. Such data if available are: faculty information’s, available courses with details, study level etc. |
| **Validation Criteria** | * User can see more details about single syllabus after choosing it. |

|  |  |
| --- | --- |
| **Story ID** | US3 |
| **Story** | As a student I want to see details of course so I can see what they offer |
| **Source** | Team |
| **Detailed Description** | User can see details of single course. Details include following if available: course name, belonging syllabus id, belonging faculty name, course short and long description, course tags, comments and ratings from other users and graphic statistics showing course popularity. |
| **Validation Criteria** | * User can see details of course * User can see comments and ratings from other users |

|  |  |
| --- | --- |
| **Story ID** | US4 |
| **Story** | As a student I want to see details of faculty, so I can know more about faculty. |
| **Source** | Team |
| **Detailed Description** | User wants to know more about the faculty related with the syllabus. The app must show a short description of the university and a link to the official website |
| **Validation Criteria** | * User can reach the information about the faculty associated to the syllabus * User can go from the syllabus description |

|  |  |
| --- | --- |
| **Story ID** | US6 |
| **Story** | As a student I want to choose country and faculty I am studying at, so I can choose my own courses for comparison |
| **Source** | Team |
| **Detailed Description** | User can choose country and faculty and then he will have a list of available courses of that faculty. User can choose courses of interest to be compared with other syllabuses. |
| **Validation Criteria** | * User can choose his faculty and country * A list of his subjects is showed * He can see similar courses |

|  |  |
| --- | --- |
| **Story ID** | US8 |
| **Story** | As a student I want to see which faculties are most similar to my own, so I can see what is my best option |
| **Source** | Team |
| **Detailed Description** | User can compare whole faculty syllabus with other faculties |
| **Validation Criteria** | * The User can see his best suggested option. * The User can review all the possible suggestions |

|  |  |
| --- | --- |
| **Story ID** | US9 |
| **Story** | As a student I want to see other subjects that are relevant for me |
| **Source** | Team |
| **Detailed Description** | User can see suggested courses without need to manually enter courses for comparison. System can suggest data based on other users. |
| **Validation Criteria** | * The User can see a list of suggested courses. |

|  |  |
| --- | --- |
| **Story ID** | US10 |
| **Story** | As a student I want to choose destination country and faculty, so I can compare my courses with them. |
| **Source** | Team |
| **Detailed Description** | User can choose destination country and faculty, so he will get compared data based on chosen input. |
| **Validation Criteria** | * User can choose his destination faculty and country which are relevant in comparison |

|  |  |
| --- | --- |
| **Story ID** | US11 |
| **Story** | As a student I want to see recommended courses, so I can maybe come to know something new |
| **Source** | Team |
| **Detailed Description** | User can see courses that system recommend to him. He can use that data in further decisions. |
| **Validation Criteria** | * User can see similar courses |

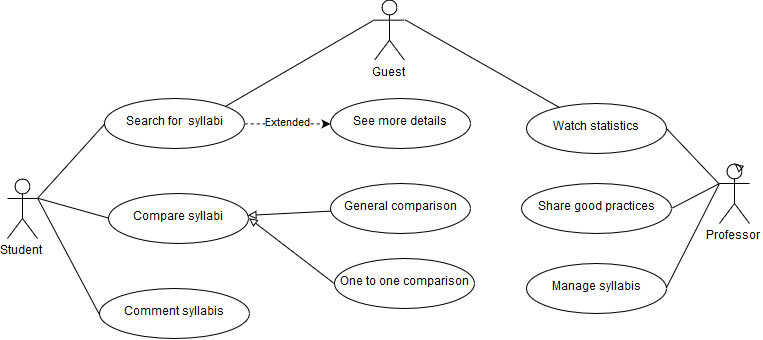
|  |  |
| --- | --- |
| **Story ID** | US12 |
| **Story** | As a student I want to comment and evaluate a course |
| **Source** | Team |
| **Detailed Description** | When students have attended a course, they can leave comment how did it looks like. Also, they can evaluate the course with a ranking system. |
| **Validation Criteria** | * User can add comments to each subject * User can evaluate each subject * App has filters for the comments. |

|  |  |
| --- | --- |
| **Story ID** | US13 |
| **Story** | As a student I want to share a course in social media so others can see it |
| **Source** | Team |
| **Detailed Description** | User can share with others syllabus with other users on other social networks |
| **Validation Criteria** | * User can share syllabus |

|  |  |
| --- | --- |
| **Story ID** | US14 |
| **Story** | As a Professor I want to have option to add new syllabi to the database so that database can expand |
| **Source** | Team |
| **Detailed Description** | Professor will have option to contribute to database with his version of syllabus. |
| **Validation Criteria** | * Professor can submit a form to create a new syllabus. * Professor can send a syllabus with a specific format to the system. |

|  |  |
| --- | --- |
| **Story ID** | US18 |
| **Story** | As a Guest I want to see the main functionalities without an account |
| **Source** | Team |
| **Detailed Description** | A guest doesn't need to have an account. However, he must be able to see statistics using the analyzer and do the simple searching’s using the syllabi explorer. |
| **Validation Criteria** | * The main functionalities must be available for the guest without the need of creating account |

## 3.4 Use Cases Diagram



## 3.5 Use Cases Descriptions

|  |  |
| --- | --- |
| **Use case name** | Search for syllabus |
| **User stories related** | US1, US18 |
| **Actor** | Student, Professor, Guest |
| **Pre-conditions** | User is in the app home |
| **Flow of events** | 1. User writes in the search engine search criteria as name, country, faculty and click in go. 2. System performs a query in the database according the criteria. 3. System shows to User in the interface the results found with a list of syllabi. |
| **Post conditions** | User can navigate across results |
| **Exceptions** | System did not find results: Show to User the problem and come back to home.  User wrote a indecipherable word: Show to User the problem and come back to home. |

|  |  |
| --- | --- |
| **Use case name** | See more details |
| **User stories related** | US2, US3, US4, US18 |
| **Actor** | Student, Guest |
| **Pre-conditions** | User has done a search |
| **Flow of events** | 1. User selects a syllabus 2. System shows to User a completed description of the syllables 3. System shows to User the faculty associated and a link to his website |
| **Post conditions** | User can see details |
| **Exceptions** | A syllable does not have one detail, for example the name of faculty: System sends a message to administrator informing about the error. |

|  |  |
| --- | --- |
| **Use case name** | Compare Syllabi |
| **User stories related** | US6 |
| **Actor** | Student |
| **Pre-conditions** | User is already registered, User is in the comparator interface. |
| **Flow of events** | 1. User submits his faculty and course 2. User selects his subjects 3. User select the type of comparison: General or one to one. 4. User select a destination university or faculty (Optional) |
| **Post conditions** | System must know if it´s a general comparison or directed. |
| **Exceptions** | The faculty is not in the database: It´s registered and the User must write his syllabi subjects manually or sends to administration the descriptions files.  The User does not select a destination faculty: The system does a general comparison |

|  |  |
| --- | --- |
| **Use case name** | Compare Syllabi: General comparison |
| **User stories related** | US8, US9 |
| **Actor** | Student |
| **Pre-conditions** | User already has submitted his subjects |
| **Flow of events** | 1. User adds country or city preference. 2. System performs the comparison algorithm with all the syllabi in the database and according to the filters 3. System shows to User the most similar faculties to home. 4. System shows to User the most important or better ranked syllabi corresponding his subjects. 5. System shows subjects suggested according the student profile. |
| **Post conditions** | User can see all the possible comparisons |
| **Exceptions** | A subject does not have a possible equivalent: Systems shows to User the error. |

|  |  |
| --- | --- |
| **Use case name** | Compare Syllabi: One to one comparison |
| **User stories related** | US10, US11 |
| **Actor** | Student |
| **Pre-conditions** | User already has submitted his subjects, User already has submitted the destiny faculty and course. |
| **Flow of events** | 1. System shows to User the syllabi of the destiny faculty. 2. System shows to User which are the similar subjects between both and which ones are different. 3. System shows a percent of similarity |
| **Post conditions** | User can knows if the two syllabuses are different or no. User can navigate across the syllabus and know more about the faculties |
| **Exceptions** | User wants to change the destination faculty: The system provides it in the same interface |

|  |  |
| --- | --- |
| **Use case name** | Comment course |
| **User stories related** | US12,US13 |
| **Actor** | Student |
| **Pre-conditions** | User is looking at course |
| **Flow of events** | 1. User selects add comment 2. User writes and submits the comment 3. User rate course 4. System verify if the comment is entered 5. System add the comment to the course page |
| **Post conditions** | The comment and the rank are added |
| **Exceptions** | The comment is missing: System shows user the error |

|  |  |
| --- | --- |
| **Use case name** | Manage syllabi |
| **User stories related** | US14 |
| **Actor** | Professor |
| **Pre-conditions** | Professor is in the manage section |
| **Flow of events** | 1. System shows to User his subjects 2. User selects an action: Delete, upgrade or add a subject. 3. In the case of add and upgrade the system shows to User a form to fill out. 4. User fills out the form and submit the new syllables 5. System verifies if all the fields are completed 6. System add the syllables to the database |
| **Post conditions** | A new Syllabus is added |
| **Exceptions** | User does not want to submit the form: System shows to user a option to upload a file with free format, After, this is analyzed by de the administrator |

|  |  |
| --- | --- |
| **Use case name** | Watch statistics |
| **User stories related** | US18 |
| **Actor** | User |
| **Pre-conditions** | User is in the explorer section |
| **Flow of events** | 1. User go to explorer page 2. In the page he can see statistics of other users engagement |
| **Post conditions** | Reports and summary are shown to User |
| **Exceptions** | The data for statistics does not exists so graphs cannot be shown. |

# 4 NON-FUNCTIONAL REQUIREMENTS

## 4.1 Usability and Portability

* CSyllabus will be a web-based app responsive for desktop and mobile usage. In this way, User can use the app with a different browser from various locations if has internet connection.
* The application must offer interface following one of popular design guidelines such as material or angular design.
* The application must be easy to use and understandable. It must not require specific knowledge of new technologies.
* The application must not require more than one hour of training to overcome its functionalities

## 4.2 Availability

System will be available through web application. Interaction with the data system will be available through system API. The system will be available for normal use every single day except in maintenance days that will be shorter than 2 days per occasion.

## 4.3 Privacy

The data used in this project will be publicly visible to all Users except in situations where data owner insist otherwise. The personal data of the user will be publicly visible in user profile except when user manually hide information’s. Syllabuses will be available only with the authorization of the respective faculties or universities.

## 4.4 Performance

The System will be scaled according to user’s acquisition and content growth so that every page loads under 2 seconds. Architecture will be designed per latest architectural guidelines. Since this application will be created in the context of the DSD course, our team will not build or require any dedicated infrastructure for it. Database will have minimum 2 syllabuses for comparison.

## 4.5 Interoperability

System will provide public API, so others can use its functionalities. Special plugin for FER faculty will be created.