



CSyllabus Final report

Version 1.0

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

Revision History

Date	Version	Description	Author
2017-01-17	1.0	Initial Draft	Thomas Benetti
2017-01-19	1.1	Added positive experiences and improvement possibilities	Zvonimir Relja
2017-01-19	1.2	Final version	Thomas Benetti

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

Contents

1. Introduction.....	5
1.1 Purpose of this document.....	5
1.2 Document organization.....	5
1.3 Intended Audience.....	5
1.4 Scope.....	5
1.5 Definitions and acronyms.....	5
2. Background and objectives.....	6
2.1 Background.....	6
2.2 Project goals.....	6
2.3 High-level requirements.....	6
3. Project results.....	7
3.1 Produced deliverables.....	7
3.2 Requirements satisfied.....	7
3.3 Improvements and extensions possibilities.....	8
3.4 Guidelines for software installation and operation.....	8
4. Project work.....	9
4.1 Organization and routines.....	9
4.2 Total project effort.....	10
4.3 Worked hours per group member.....	11
4.4 Positive experiences.....	11
4.5 Improvement possibilities.....	12

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

1. Introduction

1.1 Purpose of this document

The purpose of this document is to summarize the experiences of the project, both in terms of the produced results and of the project work.

1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction*, describes contents of this guide, used documentation during developing process etc.
- Section 2, *Background and objectives*, describes the project goal and the high-level requirements defined at the beginning of the project.
- Section 3, *Project results*, describes the produced deliverables and the requirements satisfaction.
- Section 4, *Project work*, describes our organization and routines using the SCRUM method and metrics associated to our sprints.

1.3 Intended Audience

The intended audience is:

- The supervisors
- The team
- The customer

1.4 Scope

This document addresses the retrospective of the whole project work.

1.5 Definitions and acronyms

1.5.1 Definitions

Keyword	Definitions
SCRUM	An iterative and incremental agile software development framework for managing the development of a product.
Sprint	Basic unit of time of the project development cycle used in scrum. Usually this set of time is of 1 week to 1 month maximum.

1.5.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
NTR	Nothing to Report. There is no information to a specific topic available or necessary.
JSON	JavaScript Object Notation
JWT	JSON Web Tokens
FER	Faculty of Electrical Engineering and Computing, University of Zagreb
POLIMI	Politecnico di Milano, Italy

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

2. Background and objectives

2.1 Background

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 countries includes different languages or cultures, what can be challenging or even too hard to overcome. Exploring or choosing a suitable university 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 a similar university or one that matches their preferences, comparing 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 provide all relevant information and insure that they choose best possible destination for their further education.

2.2 Project 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 High-level requirements

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 admin 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

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

3. Project results

3.1 Produced deliverables

Project delivers features based on user stories also with technical improvements through time of developing product. Deliverable items in this project are:

- **Collected syllabuses** from 4 countries and 5 universities
 - Croatia:
 - Faculty of Electrical Engineering and Computing
 - Italy:
 - Politecnico di Milano
 - University of L'Aquila
 - United States of America:
 - The University of Texas in Austin
 - University of California, Los Angeles
 - University of Stanford
- **Created web page** where user can explore and compare courses
 - In explore part user can see what are available courses based on his input
 - In comparator part user can compare courses from his own university with courses from some other university from the database
- **Application publicly visible** by leasing a server and domain:
 - <https://www.csyllabus.com>
- **Bug reporting page** where team members along with other testers can report found bugs
 - <https://bugs.int-rev.com/>
- **Rest API** in Google JSON format with Google API standard, using authentication with JWT
- **User identification** with Facebook and Google account
- **Social course sharing** = users can share their favorite courses with others through
 - Facebook, Twitter, LinkedIn, WhatsApp and Google Plus
- **Administrator panel** = user with admin privileges can modify existing syllabuses
- **About us page** = page describing what we are doing so common user can easily understand what CSyllabus is all about.
- **Connect (Developer center)** = page with technical information's where are described technical part of implementation and also listed contact and documentation sources.
- **Mobile view** = web application is fully adapted to mobile view. Content is rearranged based on screen size to enable seamless experience in common usage.

3.2 Requirements satisfied

From initial 18 user stories, 13 were finished. Five user stories were abandoned as become irrelevant or deprecated in project vision. Major features are preserved. Requirement document passed through 10 stages from initial draft to fully implemented document.

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

3.3 Improvements and extensions possibilities

One of the major improvement/extension we would like to do is expand the list of universities available on our website. As said in the questionnaire that we made for a few users, they said it would be better to have more courses so that they have a lot of choices to compare and therefore choose wisely.

3.4 Guidelines for software installation and operation

Software setup consists of setting frontend and backend parts. Both parts can be run individually for debugging process.

Backend setup guide:

- install PostgreSQL 9.6x (<https://www.postgresql.org/download/>)
- install with pgAdmin
- create new server hostname -> "localhost"
 - o write down password and username for root
 - o (usually username = postgres)
 - o after installation create a database in pgAdmin to be used with the opensyllabus and write down the name used
 - o if you create a new username and password for the database write it down too
- install python 2.7 (<https://www.python.org/downloads/>)
 - o check python version in command line with 'python -V'
- install pip for python 2.7 (it already comes shipped with python 2.7.9+)
 - o check pip version with 'pip -V'
- install django with 'pip install django'
 - o check django version with

```
$ 'python -c "import django; print(django.get_version())"'
```

Install project requirements

- position yourself in CSyllabus root folder
- pip install -r backend/requirements/devl.pip
- in file backend/settings/devl.py field change DATABASES according to database name, username and password you wrote down in first

Steps for migrating and loading data

- *python manage.py migrate*
- *python manage.py loaddata backend/apps/csyllabusapi/fixtures/epfl_fixtures_json.json*
- *python manage.py loaddata backend/apps/csyllabusapi/fixtures/fer_fixtures_json.json*
- *python manage.py loaddata backend/apps/csyllabusapi/fixtures/laquila_fixtures_json.json*
- *python manage.py loaddata backend/apps/csyllabusapi/fixtures/mockup_fixtures_json.json*
- *python manage.py loaddata backend/apps/csyllabusapi/fixtures/polimi_fixtures_json.json*
- *python manage.py loaddata backend/apps/csyllabusapi/fixtures/stanford_fixtures_json.json*
- *python manage.py loaddata backend/apps/csyllabusapi/fixtures/texas_fixtures_json.json*
- *python manage.py loaddata backend/apps/csyllabusapi/fixtures/ucla_fixtures_json.json*

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

- *python manage.py createsuperuser*

Running backend with command:

- *python manage.py runserver*

Frontend setup guide:

- install nodeJS (<https://nodejs.org/en/download/>)
- position yourself in the frontend/csyllabus folder
- run *'npm install'*
- serve angular app with *'ng serve'*

4. Project work

4.1 Organization and routines

4.1.1 Meetings

4.1.1.1 Team meetings

We had two types of team meetings:

- Daily meetings: with the SCRUM method, a meeting must be done every day of work, which includes the work done yesterday, the work that will be done today and what problems have been encountered by each team member. But as we are not working every day on this project, we are doing the daily meetings on Wednesday, Friday and Sunday.
- Weekly meetings: These meetings are done on Sunday and lasts 1 hour and a half. The planning of these meetings are :
 - If it is the end of a sprint: to do the sprint retrospective and plan the next sprint.
 - If it is the middle of a sprint (1 week): to check the current sprint's progress.
 - If there is a presentation on the next Tuesday: to do the assignment of the 2 team members that will do the next oral presentation.

4.1.1.2 Team and supervisors meetings

They are held on Fridays. They are used to control the progress of the project with the supervisors, but also to have their opinions.

4.1.1.3 Product Owner and customer meetings

The customer meetings are not held as team and supervisors meetings. They are held when its necessary for the project progress, and if the team has interrogations on some project characteristics.

4.1.2 Forms of contact

At first, all the meetings were done via Skype but we occured to have some troubles regarding Skype so we used Google Hangouts as a backup plan for internal meetings. The off-board communication is done via Slack, Google Drive or e-mail. We also use Kanban Flow for project management and Toggl for time tracking.

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

4.1.3 Worked time reports

The worked time on any task is reported on Kanban Flow. Each team member use the task time log to enter the time spent on each task every day. When the task is done, it is compared to the estimated time to know how much and why it may be different.

Kanban Flow also has a time tracking tool that is used by several team members. It allows them to launch a timer with pre-defined work time and breaks. In that way, they can define their schedule and after each time spent know why they overflowed the estimated work time.

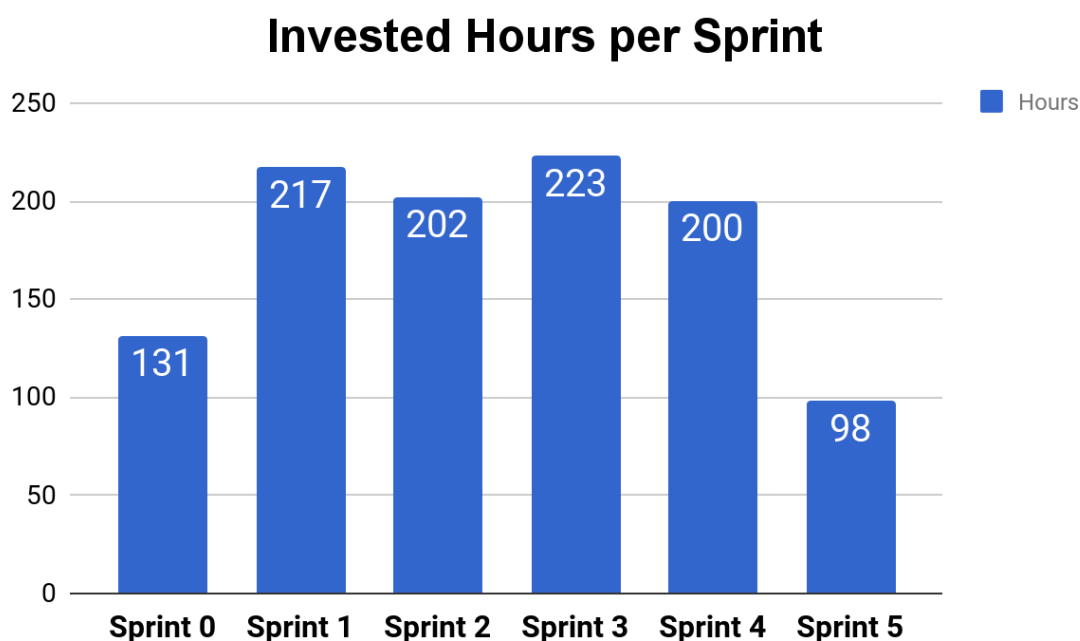
But we had an issue with Kanban Flow regarding worked time : it cannot generate reports/analysis to use for sprint reports. So we used Toggl to enter our working hours and to be able to visualize metrics concerning our worked time.

4.1.4 Configuration management

We use Git to manage multiple versions of our application, with everyone working on separate branches, it allows the work to be done without interferences and to keep stable versions of the application.

Also we use GitHub as the repository of Git applications versions. The repository is public as it is a SCORE project.

4.2 Total project effort

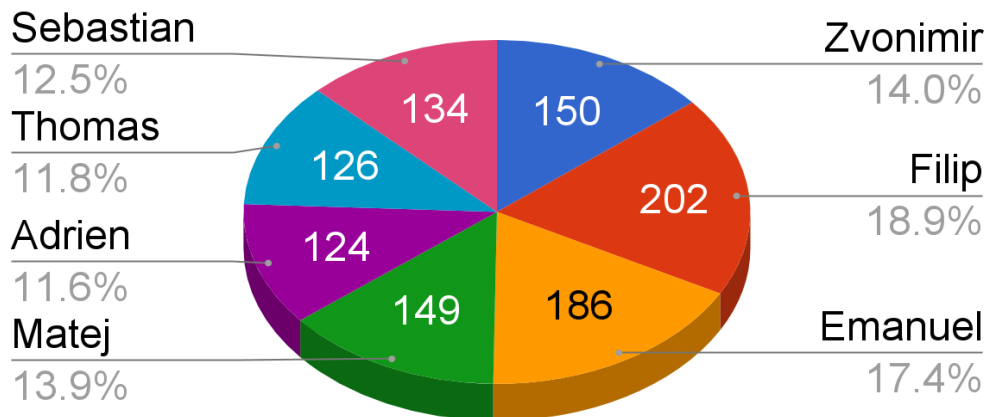


As you can see in the figure below, we worked an average of 214 hours per sprint. And as for the whole project, we worked for over 1 000 hours.

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

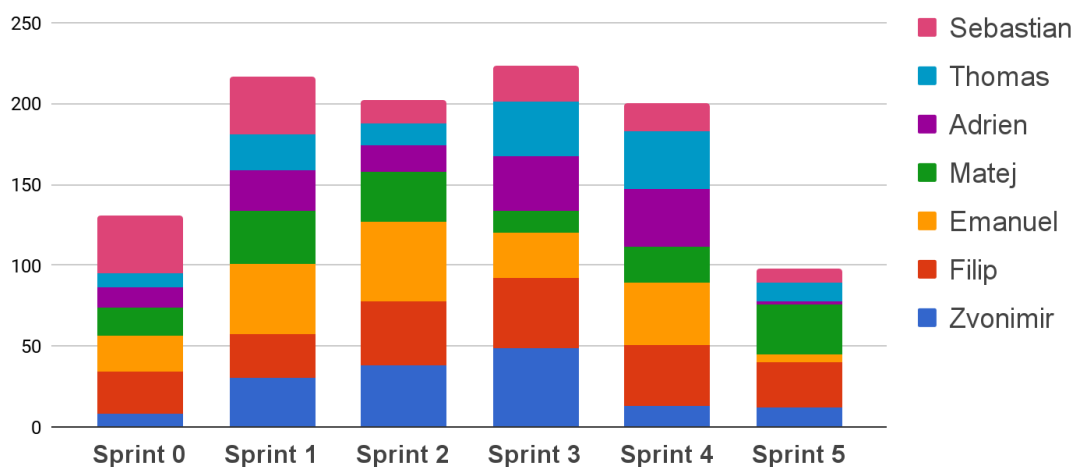
4.3 Worked hours per group member

Invested Hours per Member



As for each team member, the invested hours depended on how complicated were the tasks assigned.

Invested hours per Sprint for each member



4.4 Positive experiences

There are many good experiences within the Csyllabus team during whole project:

- improved English skills
- improved public presentation skills
- experienced working in a distributed team
- experienced teamwork
- learned how to overcome all sorts of problems
- learned new technologies including: Angular 4 framework, Django framework

CSyllabus	Version: 1.0
{Document Type}	Date: 2017-01-17

4.5 Improvement possibilities

4.5.1 Technical

Angular Material wasn't the happiest choice for UI design. We thought and researched a lot what to use and at the end of planning decided to go with Angular Material. It is relatively new framework and when we stumbled upon code problems the community wasn't much helpful and there wasn't many solutions on the internet.

4.5.2 Documentation and presentations

Documentations and presentations were too stressful for the entire team because they would usually be done last few days prior to the deadline. If we could change things, we would definitely make higher priorities to do this things earlier.