

CHARLES UNIVERSITY
Faculty of mathematics
and physics

Software project - analysis

Artur Finger, Slávka Ivaničová, Zoltán Betteš, Marek Beňovič, Gergely Tóth

The new website for the software project committee at the Informatics section of MFF UK (PRKNO)

Katedra softwarového inženýrství

Project supervisor: doc. RNDr. Petr Hnětynka, Ph.D.

Prague 2017

Contents

1	Introduction	2
1.1	Context	2
1.2	Goal(s)	3
2	Specification	4
2.1	Users	4
2.1.1	Guest	4
2.1.2	Student	8
2.1.3	Academic	12
2.1.4	Committee member	15
2.1.5	Committee admin	18
2.2	Shared functionality	24
2.2.1	Data export	24
2.2.2	Internationalization	24
2.2.3	Email notifications	24
2.2.4	Anonymization	24
2.2.5	Accounts	24
3	Analysis	28
3.1	Technologies	28
3.1.1	Backend (PHP, Yii)	28
3.1.2	Frontend (JS, jQuery)	28
3.1.3	Database (MySQL, Yii Database)	29
3.1.4	Typesetting (Markdown)	31
3.2	Architecture	31
3.3	Risk analysis	33
3.4	Milestones	33
3.5	Work split	34

1. Introduction

Glossary

- **CAS** - A central authentication system for students and staff of Charles University

1.1 Context

As a part of the mandatory curricula of most master's degree programmes at the Informatics section of MFF UK there is a subject called the Software Project (code *NPRG023*). This subject does not have any lectures and the way to pass it is to form a team of four to eight students and implement a complex, full-featured, ready-for-shipment piece of software. The problem which the software solves must be well analysed by the team, design decisions must be made with careful consideration and the resultant software must be well-documented.

The detailed life cycle of a software project is as follows. First a team must assemble and come up with an idea. The idea can be original, it can be some software that some of the faculty's research teams need implemented or it can even be on contract for some external company - its origin does not matter. Once an idea is agreed upon, the team must find a faculty member who would agree to supervise them and help them with this project. The supervisor then drafts a short proposal describing the aim of the project and sends it to the project committee for evaluation. If the committee agrees that the project is reasonable and complex enough, they accept the proposal. Otherwise, the supervisor has a chance to improve the proposal, maybe add some extra features and resend it or the team can come up with another idea. After a proposal is accepted, the team has to come to an agreement about when they will start working on the project, their time starts ticking - they have nine months to complete the project. Once the team has agreed upon the start date, the supervisor sends a request to start the project to the committee. (The request includes a list of students that will comprise the team and the start date.) If the committee accepts the request, the team can start their work on the project on the specified start date. Two months into the project the team must present their analysis of the problem to the committee. After additional 7 months (9 months in total since the beginning of the project) the team stops working on the project and soon after they stop, they present the complete software to the committee (this is known as the defence of the project). The committee will have questions and the team must answer them convincingly. If everything is OK, the team passes. In the unlikely event that the software or anything related to it (e.g. the documentation) is not satisfactory, the team continues to work on the project for additional 3 months and then presents/defends it again. During any of these two defences, if the committee does not find any harsh flaws with the software, but the software or its documentation are not completely satisfactory, the team passes on a condition - which means they have to finish up some details (perhaps refactor some code, rewrite the documentation, maybe improve test coverage) but they do not need to defend anything again. After the team makes satisfactory corrections, they

pass.

If there is a problem with the analysis, the team must rework it and defend it again. Also, same as with the final defence, if the problems are only minor ones it may be sufficient for the team to submit their reworked documents and diagrams, without the need to attend a new defence. Unlike the final defence, the defence of the analysis may be repeated an unlimited amount of times. It should be noted however, that the deadline for the implementation does not change, so repeating the defence of the analysis takes away time that should be used for implementation.

1.2 Goal(s)

The purpose of this project is to implement a web application for the project committee, which would serve as an information system for managing software projects and related tasks. In short, the application will allow academics to create projects and proposals, send them to the committee, assign students to a project and so on. Students will be able to add more information to their projects. Committee members will be able to discuss proposals of defence dates. The admin of the committee will be able to manage defences, search users, accept and decline projects during a defence and much more. In case of any problem the committee admin will be able to override anything in the system. One feature of this application will also give anyone (who is registered) the ability to advertise their idea for a potential software project and another feature will allow students to look for a team via an advertisement-like system.

2. Specification

After a careful analysis and disambiguation of the project committee's functional requirements for the website, we propose the following specification. The functionality listed in the specification should cover the needs of any person who takes part in software projects (meaning those that are part of the *NPRG023* course).

2.1 Users

The users of this web application can be split into six categories according to what role they play in software projects. A person can assume one of the following roles:

- Guest 2.1.1
- Student 2.1.2
- Academic 2.1.3
- Committee member 2.1.4
- Committee admin 2.1.5


Different functionality is offered to each category of users. A guest is someone who does not take part in software projects at all. Such people will only have access to the bare minimum of information and functionality. (Naturally, all users are perceived as guests before they authenticate themselves.) A student is a master's degree student of the Charles University. *Technically, it could be a bachelor's degree student or a student from a different university, but that is highly unlikely, since the course is not intended for them.* A student has all the privileges of a guest and more. An academic is a PhD student or staff of the Charles University. *Again technically they could be from a different university, but that is highly unlikely.* Academics have all the privileges of a student and more - with two exceptions: they cannot work on a project as part of the team and therefore they also cannot create a student advertisement about themselves, since they will not be needing it. What they can do is to supervise a project or oppose one. Committee members have all the privileges of an academic and more. Most importantly, they are the ones that decide the fate of newly proposed projects. A committee admin has all the privileges of a committee member and more. In fact, an admin is able to create, delete, change and override anything in the system and there may be multiple committee admins. This role is usually assigned to the chair of the committee or the secretary. The functionality offered to each category of users follows below.

2.1.1 Guest

Guests cannot edit or change anything on the web site. The only thing they can do is to view content. Generally they can view as much information about

projects as possible, but nothing that could be considered sensitive information. The intended use case for a guest is just to browse the website a bit and check out a few projects to gain some understanding of software projects before taking the course.

1. **News** - All important events appear in a news feed. News are generated when a static page changes (eg. when the official rules for the software project course (*NPRG023*) change), when a new date is open for defences to take place on or when a defence is cancelled. More on static pages later. Guests are able to view most of the news, but some are intended for and visible only to the users who are logged in.
2. **Advertisements** - There are two types of advertisements - ones about potential projects (so called project advertisements) and others about students who are looking for a team (so called student advertisements). A student advertisement is a short summary that a student posts about himself when he is looking for a team. Such an advertisement should contain a list of technologies the student would like to use on the project, his preferred topics and his time/schedule preferences. Other students can then contact the author of the advertisement and add him to their team. Guests can view all advertisements, but cannot comment on them. Advertisements can be filtered by keywords.
3. **Projects** - Guests can view a list of all projects and also the details of each. The details contain a short description, names of the team members, the state of the project 2.2, the starting date and the deadlines, the name of the supervisor, possibly a detailed description and a list of uploaded files of any kind. Furthermore there is always the specification of the project and - if the project is finished - an advice document (giving tips to other students on how to work efficiently on such projects, based on the experience the team has gained during this course). It is possible to filter projects by keywords, supervisor, date/time interval and by state or look for them by a full-text search. Projects in the proposal stage cannot be searched or listed because they are not formally considered to be a project yet.
4. **Defenses** - Guests are allowed to see basic info about upcoming defences, including the date and a list of projects scheduled for a defense.
5. **Static pages** - Guests can also view several static pages including the official rules of the *NPRG023* course, a guide through the life cycle of software projects, a list of downloadable document templates and a page about the committee.



Web projektovoj komise

ENG ▼
Sign In

Home
Projects
Advertisements
Documents

Advertisements

[Show/hide filter](#)

Project ads
Students ads

Date from to
Keywords

[Filter](#)

Title of advertisement
Project

Publication date [Author of Advertisement](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Keywords: # web # PHP

[2 Comments](#)

[Hide comments](#)

[Author of the first comment](#)
Publication date

I'm interested in this project.

[Author of the last comment](#)
Publication date

I'm interested in this project. It looks cool.

[Sign in to write comment...](#)

Title of advertisement
Students

Publication date [Author of Advertisement](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in.

Keywords: # web # PHP

[0 Comments](#)

News

Publication Date

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

News Title

Publication Date

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

New open projects

ID abbr.	Proposal accept	Supervisor	Project name
Value 1	10.2.2017	Supervisor name	Value 5
Value 2	5.2.2017	Supervisor name	Value 6
Value 3	2.2.2017	Supervisor name	Value 7
Value 4	30.1.2017	Supervisor name	Value 8


Sign In

User Name:

Password:

[SIGN IN](#)

[Forgot Password?](#)



[About Terms](#)

[Press & Blogs Privacy](#)

[Copyright Policy & Safety](#)

[Creators & Partners Send feedback](#)

[Advertising Try something new!](#)

[Developers](#)

Figure 2.1: Guest page

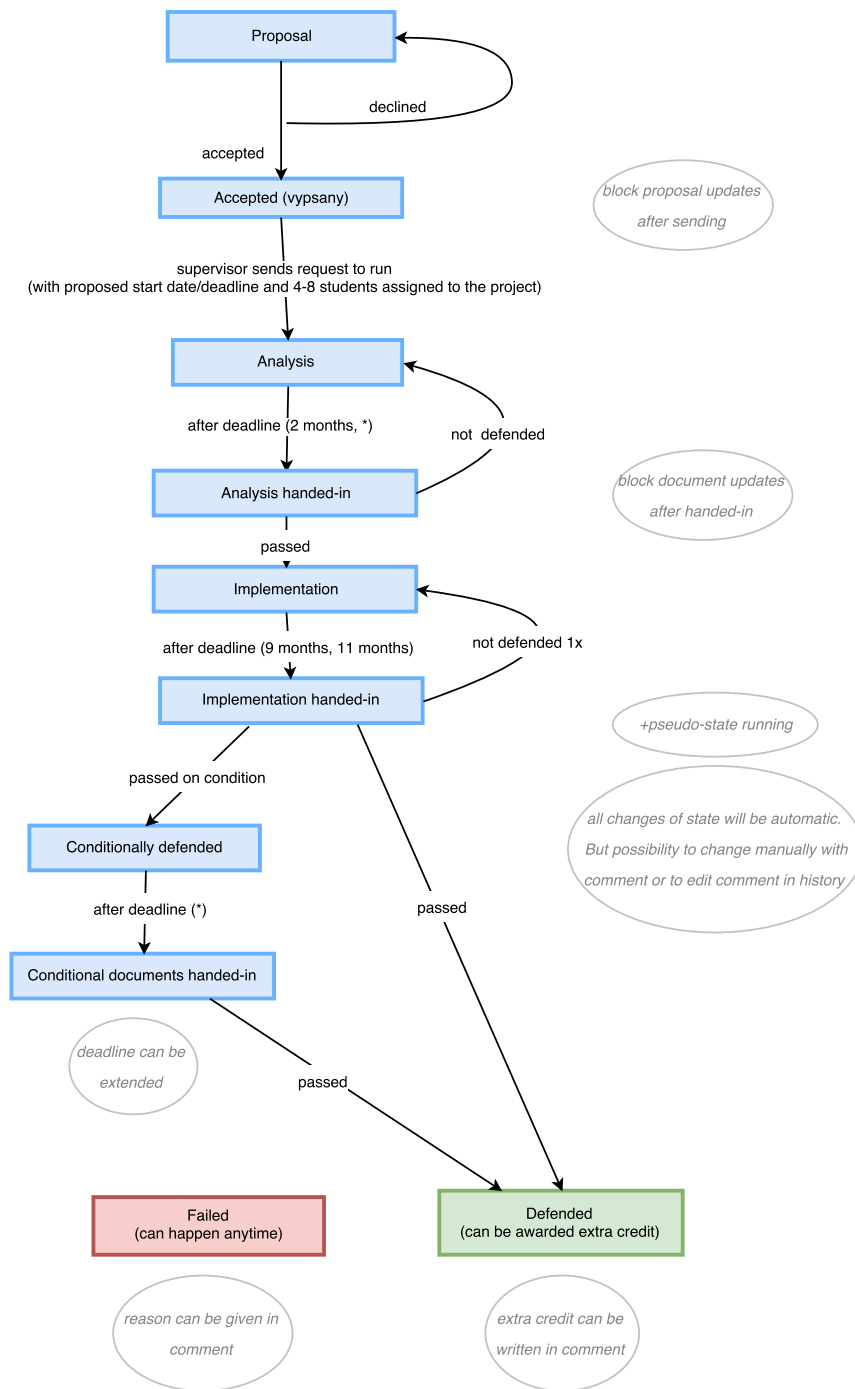


Figure 2.2: Stages of a project

A project starts in the proposal stage. Once a proposal is accepted the project proceeds to accepted stage and becomes visible to everyone. After a team gathers and the project starts to run it goes to the analysis stage and after a two month deadline the analysis is handed in. Implementation follows with a seven month deadline after which code and documentation are handed in. If a project passes, its state is changed to defended and the students are awarded their credits. Otherwise the project may be forced to attend another defence (at most one for implementation and any number for analysis). A conditionally defended project is followed by minor rework, which is handed in after a specified deadline. If everything checks out, the project is defended. At any point the committee admin can kill a project, sending it to the failed stage, in which case he or she must supply a reason.

2.1.2 Student

Students can do everything a guest can and more, after they log in. The main addition is that students can edit the content of the project they are working on and do a few more things connected to that. The intended use case for a student is to create a student advertisement, get together a team, find a project with a supervisor and then work on the project for nine months, uploading relevant files before each deadline. If the student has a more creative spirit they can think of their own project, create a project advertisement and form a team around it.

1. **News** - Students can view all news and also subscribe to them by email.
2. **Advertisements** - Students can create, edit, hide and delete any number of project advertisements. Students can view and comment on the advertisements or show interest in an advertised project. The author of an advertisement receives an email each time someone comments on it, but it is also possible to unsubscribe from these.
 - (a) Concerning person advertisements, the functionality is similar, but the student can create only one such advertisement, no one can comment on it and once the student starts working on a project, the advertisement gets hidden automatically.
3. **Project** - The author (supervisor) of a project adds team members to it. The team can view their project directly on the homepage. They can edit the content of the project in markdown, view and upload documents, edit basic info and create/edit/send proposals (same as the supervisor (2.1.3-1)). Documents can be just general ones, which can be hidden from anyone who is not part of the team (more specifically, this does not include the supervisor of the project and neither the committee admin). But there are also some documents that are formally required. These are the documents that get handed in to the committee after a part of the work is finished (i.e. before each defence) - theses always consist of one PDF file for the documentation and a zip file containing any code, diagrams and similar. These documents are hidden from anyone but the committee and they can be re-uploaded any number of times before the deadline. There is also another special kind of file, which can be uploaded by each team member after the project is finished - this is the advice document for the next generation of students. Each student can upload only one of these. It is also worth mentioning that students can view the history of the states their project went through.

When the time is right, team members can schedule their project for a defence, if a defence date is available. It is necessary for the team to upload all relevant documents for a defence to their project a week before the defence takes place. There is also a deadline for each chunk of work (the analysis, the implementation, corrections of either one) which cannot be exceeded either. The documents for defences have to be uploaded before these deadlines pass. *Technically, one person can be working on multiple projects simultaneously, but that is not recommended as it would be too time consuming.*

4. **Defense** - As mentioned earlier, team members can schedule their project for a defence. This must happen at least twice - once for the analysis and once for the implementation. If there are some problems with the project, the team will be asked to rework it and defend it again. The defence of the analysis can be repeated arbitrarily many times, but the defence of the implementation only once at most. The team members must schedule their project for a defence each time. After the supervisor's and the opponent's reviews are uploaded, students can view them immediately. Additionally, after a defence, students can view the committee admin's statement.
5. **Users** - Students can change some details in their profile, such as the subscription to the news published on the site and the URL of their personal webpage. In case the student doesn't log in via CAS, they can edit their email and change their password.

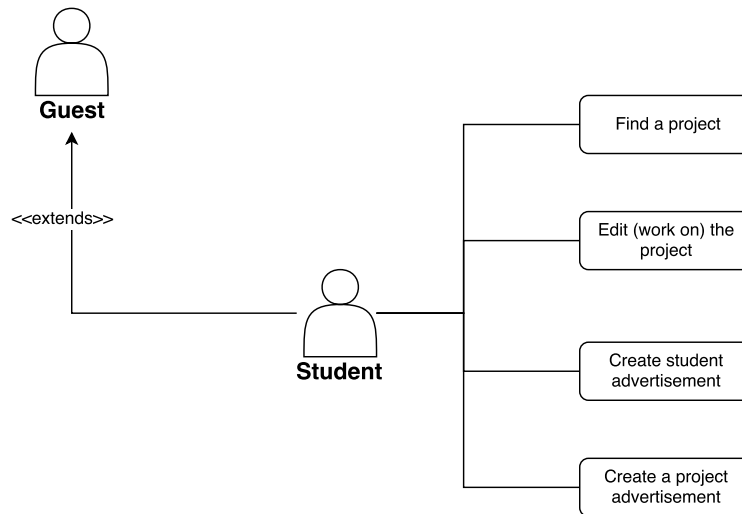


Figure 2.3: Student use case

Use Case	Find a project
<i>Primary Actor:</i>	Student
<i>Main Success Scenario:</i>	<ol style="list-style-type: none"> 1. After the student has logged in, the advertisements are displayed directly on the homepage. In order to show only the project ads the student enables the filter and checks the <i>Project ads</i> field. The results are refined by specifying the important dates (when is the project introduced and when is it due) and keywords. By clicking the <i>Filter</i> button the list is refreshed to show only the appropriate entries. After perusing the description the student signals his or her interest by clicking the <i>Interested</i> button. (A1)
<i>Alternate scenario:</i>	A1: To gain more information, the student can: <ol style="list-style-type: none"> (a) post a comment about the project (ask a question) (b) view the comments about it (if there are any) by clicking on the <i>Comments</i> link presented below the description.

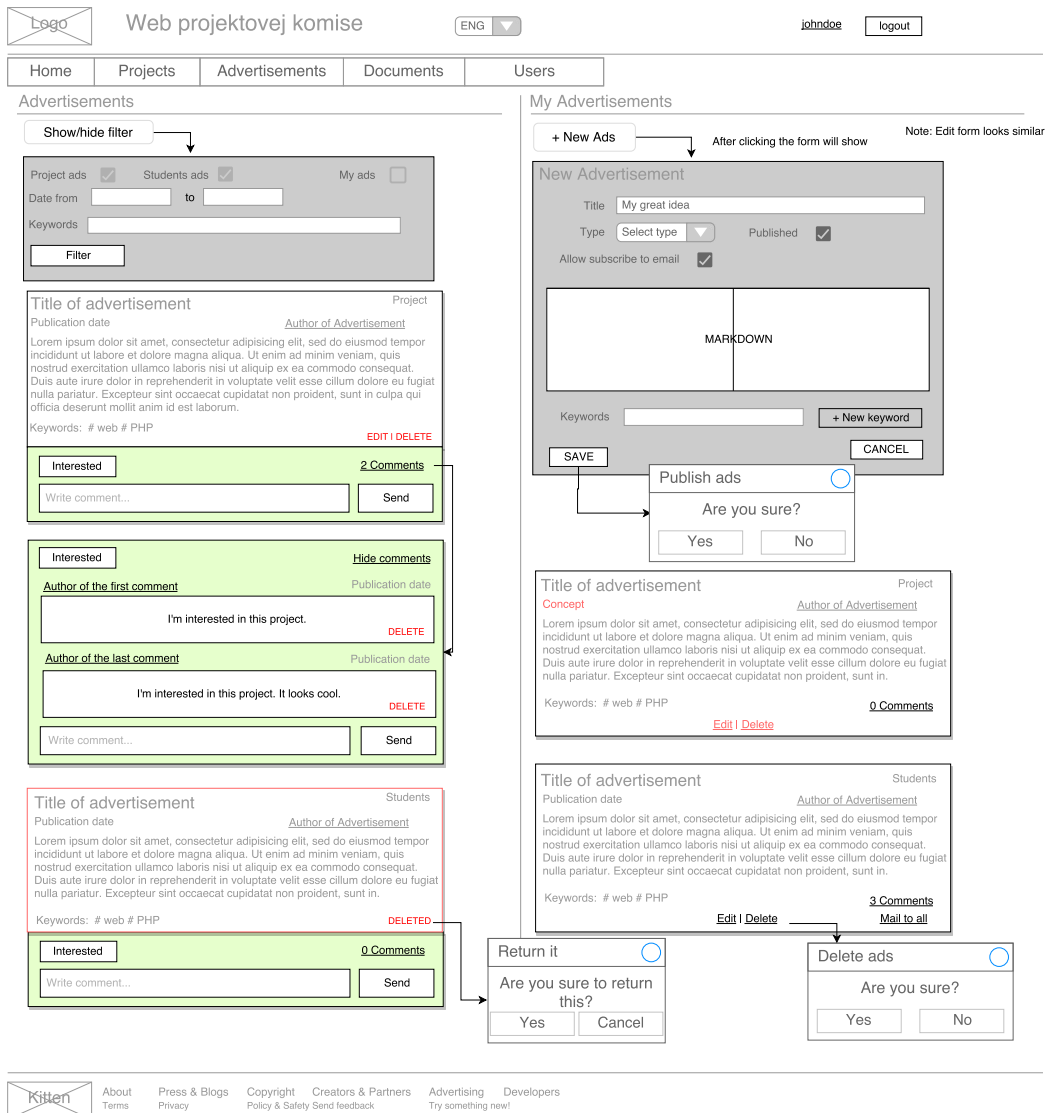


Figure 2.4: Advertisements page

Use Case **Create student advertisement**

Primary Actor: Student

Main Success Scenario:

1. After navigating to the *Advertisements* page, the column titled *My Advertisements* is presented to the student on the right side. The user clicks on the *New ad* button and a form appears asking the user for the details of the advertisement. The student gives it a title, selects the type of it to be *Student advertisement* and writes a short description about the things he is looking for. Additionally, the user can also specify keywords for the ad (which helps in the filtering process). The user publishes the ad by clicking the *Save* button and then confirming it in the pop-up dialog. (A1)

Alternate scenario:

- A1: The student can choose not to publish the advertisement by clicking on the *Cancel* button or by not confirming the publishing in the pop-up dialog.

Web projektovej komise

ENG

john Doe Logout

Showing on the home page depends on the user Student with project

Home Projects Advertisements Documents Users

My project

Name: Nový web projektové komise
Abbr.: PRKNO

Project progress: [Progress bar]

State: Running

Next deadline: 1.12.2017

Supervisor: Petr Hnětynka

Consultants: Student 1, Student 2, Student 3, Student 4, Student 5

Description: [Text]

Documents to defense: ZIP file, Upload zip, PDF file, Upload pdf

file_name.pdf	Comment	Delete
file_name2.zip	Comment	Delete

+ Add document

New private document

Upload form: file_name.pdf

Comment...

ADD CANCEL

Proposal: Show markdown | Download pdf

Defence: 9.12.2017

Defence: 9.12.2017

Defence 2: 9.12.2017

News

News Title

Publication Date

News Title

Publication Date

Defenses

Date	Attending	Active vote	Log in project
1.5.2017	Yes 5, No 4, Null 2	YES	Log in
2.5.2017	Yes 3, No 6	NO	Log in
5.5.2017	No votes	NO	Log in
8.5.2017	No votes	NO	Log in

About Terms Press & Blogs Privacy Copyright Policy & Safety Send feedback Creators & Partners Advertising Developers Try something new!

Figure 2.5: Homepage of the student with a project

Use Case Edit (work on) the project

Primary Actor: Student

Main Success Scenario:

1. The current state of the project is displayed directly on the homepage of the student after he or she logs in. In order to move the project forward, the student uploads private documents to the server by clicking the *Add document* button in the *Documents* section. After the selection of the file from the disk, it is submitted by clicking on the *Add* button. (A1) Optionally, the student can add a comment to the particular document for all his team to see and provide more information about it, thus improving team communication. When the student (and the team) feels that the project is ready to be defended, they upload the necessary ZIP and PDF file to the appropriate fields in the *Documents* section.

Alternate scenario:

- A1: By clicking on the *Cancel* button the submitting process is aborted.

2.1.3 Academic

Academics can do anything students can, except for two things. They cannot create student advertisements (2a) and cannot be part of a team (3). On the other hand they can supervise a project, oppose one or play the role of a consultant. The intended use case for an academic is to come up with new projects, make suitable proposals for them (that would get accepted by the committee) and once an interested team shows up, supervise them. Another scenario for an academic is that when a team approaches them with their own original idea for a project and asks for supervision, the academic must make a proposal based on that idea and see to it, that the project is complex enough to be accepted by the committee. A supervisor must upload a review of the project two days ahead of the defence of the implementation. Also, from time to time an academic is asked by the committee to oppose a finished project. In such case, he or she must revise the implementation and the documentation of the software and upload their review at least two days before the defence of the project.

1. **Project** - As far as searching and browsing projects goes, academics can do anything a guest can do. In addition to that, academics can do quite a few things. As mentioned earlier, they can create proposals for new projects and send them to the committee for approval. But before a proposal can be sent, the author of the project must specify a basic description including the name of the project, a short name, a short description and keywords (the description and keywords can be edited later but not after the project starts running). The author of the project also automatically becomes its supervisor. During the time the committee is deciding about whether the proposal will be accepted or not, no additional proposals about the same project can be sent. If the proposal is declined, the academic can make suitable adjustments and resend it. Otherwise, the project becomes visible in searches and eligible to run.
 - (a) The supervisor can view the history of all the declined proposals and the comments from the committee about the reason a specific proposal was not accepted. Proposals can be created and edited on site in markdown or uploaded as PDF. Proposals in markdown can be saved as drafts to be finished and sent later.

Once the supervisor of a new project has a team ready to tackle it, he or she can issue a request to run the project, listing the members of the team. The date of start is specified by the supervisor in the request. It is not possible to specify members of the team who are not students. The supervisor can add consultants to the project, but that must be done before the request to run is sent (after that, none of the basic info can be edited anymore). A consultant is another interested party - if a project is commissioned by some company, a representative from that company would be a consultant of the project. Consultants can either be registered users (academics) or not part of the information system at all, in which case they are identified simply by their email address. When a supervisor adds a consultant, he or she must add a comment explaining the consultant's relationship to the project. The supervisor can view any file that the team has uploaded to

the project and do anything a team member can do including scheduling their project for a defence.

Once the implementation is finished, an opponent is assigned to the project with the task of finding out any shortcomings. The opponent can view only that which a guest is allowed to see with one exception - he can also access the files uploaded for the defence.

2. **Defence** - Much like a student working on a project, the supervisor of a project can schedule it for a defence. They cannot pick the hour, only the day of the defence. No later than two days before a defence of the implementation both the supervisor and the opponent must upload their reviews of the project for the defence.

The screenshot shows the 'Web projektové komise' project details page. The page includes a navigation menu (Home, Projects, Advertisements, Documents, Users), a project name 'Nový web projektové komise', and a progress bar. Key sections include:

- Project progress:** State 'Running', Next deadline '1.12.2017', Supervisor 'Petr Hnětynka', and a list of consultants and students.
- Documents:** A list of documents to defend (ZIP and PDF files) and a 'New private document' form.
- Proposals:** A 'Proposals' section with a 'Show Proposal history' table containing two entries: '1.1.2017 Decline Proposal' and '21.2.2017 Decline Proposal'.
- Defence:** A 'Defence' section with three entries for '9.12.2017' and a 'Link to defence page'.

 Annotations include:

- 'Comment shows in bubble' pointing to a comment field.
- 'Hide State history' pointing to a table with columns 'State from' and 'State'.
- 'Hide Proposal history' pointing to the proposal history table.
- 'Link to proposal page with comments' pointing to a green button.
- 'Link to new project form with pre-filled data' pointing to a green button.
- 'Link to defence page' pointing to a green button.

Figure 2.6: Project details page

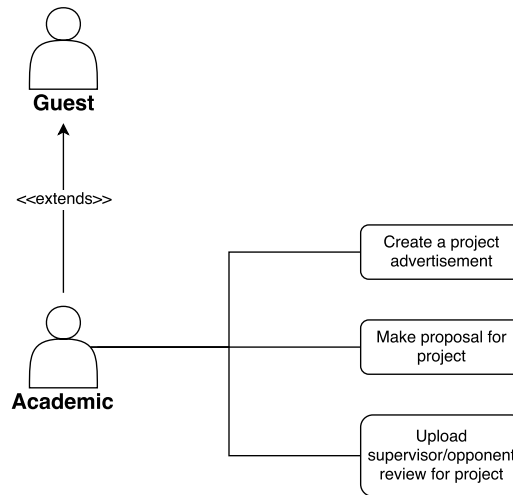


Figure 2.7: Academic use case

Use Case **Make proposal for project**

Primary Actor: Academic

Main Success Scenario:

1. The academic navigates to the project page where he clicks the *New Project* button. The user fills out the appearing form with the project details, which are the name of the project, a compulsory abbreviation for it and a link to the consultants and the students assigned to the project. Furthermore, he provides a short description of the project and types the proposal directly on the webpage (A1). After clicking the *Send* button (A2), a pop-up dialog appears to ask the academic to confirm his choice to send the proposal (A3).
-

Alternate scenario:

- A1: The proposal can also be uploaded as a PDF file by clicking the *Upload proposal* button, which will display a file-selection window. The academic selects the file from the disk and clicks on the *Open* button to upload it.
 - A2: The academic can choose not to send the proposal to the committee right away and save it as a draft instead.
 - A3: By clicking *No* on the confirmation dialog the user can return to the form to modify the values of the fields or to rather save the proposal as a draft.
-

2.1.4 Committee member

Members of the project committee can do anything any academic can do. Specifically, they can supervise and oppose projects as well. Other than that, they have some additional privileges. The intended use case for a committee member is the same as for an academic. In addition to that, he or she takes part in deciding the fate of every project proposal that is sent to the committee. A large enough portion of the committee also has to take part in every defence. To that end, each time a new date for a defence is opened, each committee member lets the others know whether he will be able to be present during the event.

1. **Project** - Committee members can view any project - even those that are still in the proposal stage (their proposal has not been accepted yet). They can also view the history of the states any project went through and display the list of team members (even though it may have been hidden from anyone not working on the project). In addition, they can view the history of all proposals for each project and all documents that were submitted by the students for any defence the project went through.

Concerning proposals, committee members have access to all proposals sent to the committee for evaluation. They can comment on them among themselves and vote for or against the acceptance of the proposed project. Each member must vote and until they do so, they are being notified repeatedly. If any of the comments is intended for the author of the proposal, it will be sent back to him along with the verdict.

All comments made about any proposal remain visible to the committee forever. However, no member of the committee can see comments on a proposal that was sent to the committee and responded to during an interval the committee member did not have this or a higher role (committee member, committee admin). This way privacy is protected even in such cases when a student becomes a member of the committee later on and then tries to view comments on the projects from his or her student years.

2. **Defence** - As mentioned above, each committee member can and must let the others know whether he or she can be present during a defence. If they cannot be present, they must specify a reason why. Otherwise, they can add an additional comment (eg. when they can be present, but are only available in the afternoon). Again, until a committee member announces his or her presence during a defence, he or she is being reminded of it repeatedly (via email).

Committee members have complete knowledge about any defence. Specifically, they can also view the supervisor's and opponent's reviews - which they should skim through before they take part in a defence of a project. These documents are available two days before the defence. After the defence is finished, committee members can view the admin's final statement. This can be viewed even by those who were not present during the defence.

3. **Users** - Committee members can search for users by name or filter them by role in order to view their info. This way they can write an email to anyone or find out what their role is. In comparison, students only see the

info (including the email address) of users they meet on the website, such as the authors of advertisements, their comments etc.

4. **Dashboard** - Committee members have a dashboard containing the most pressing affairs right on the homepage. It includes new proposals the committee member hasn't yet voted about and defence dates, for which his or her participation has not been confirmed.

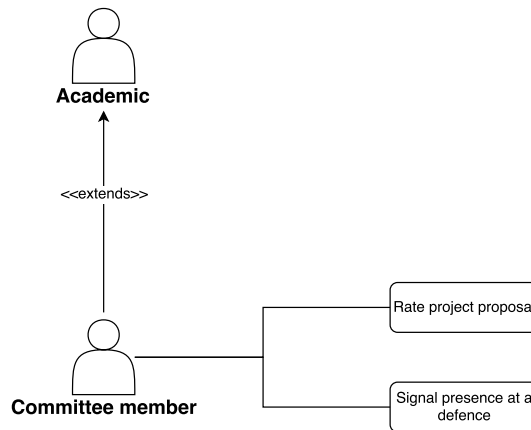


Figure 2.8: Committee member use case

Use Case	Rate project proposal
<i>Primary Actor:</i>	Committee member
<i>Main Success Scenario:</i>	<ol style="list-style-type: none"> 1. The committee member navigates to the page of the proposal by selecting the project on his homepage and clicking the <i>Proposal</i> link. After reading and examining it the committee member votes to accept the proposal by clicking on the <i>Accept</i> button below the displayed text of the proposal (A1)(A2). His vote will be shown to other committee members.
<i>Alternate scenario:</i>	<ol style="list-style-type: none"> A1: The committee member can also decline the proposal by clicking the <i>Decline</i> button below the text. A2: The user can also choose to ignore the request to rate the proposal by leaving the page of the proposal (in which case he will be reminded to rate it later).



Showing on the home page depends on the user
Committee member

Inbox

ID abbr.	State from	State	Project name	What
Value 1	10.2.2017	Proposal send	Value 5	Why is the project in Inbox
Value 2	5.2.2017	Running	Value 6	
Value 3	2.2.2017	Defended	Value 7	
Value 4	11.1.2017	Canceled	Value 8	

Defenses

Defense Date	Time	Project	Attending	Active vote	My vote
1.5.2017			Yes 5, No 4, Null 2	YES	No vote
	9:00	Project ID			
	13:00	Project ID			
	15:00	Project ID			
		Project ID			
2.5.2017			Yes 3, No 6	NO	Yes
5.5.2017			No votes	NO	
8.5.2017			No votes	NO	

Defense date: 2.5.2017

Votes

Name	Vote	Comment
Member Name 1	YES	
Member Name 2	YES	
Member Name 3	NO	Lorem ipsum dolor sit amet, consectetur
My Name	No vote	

News

News Title

Publication Date

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.

My projects

<< 1 2 3 4 5 6 7 8 9 >>

+ New Project

Name: Nový web projektové komise
Abbr.: PRKNO
State from: 1.3.2017 State: Running Show State history

Hide State history

State from	State
1.3.2017	Running

Supervisor: Petr Hnětynka

Consultants

Students: Student 1, Student 2, Student 3, Student 4, Student 5

Description

Edit

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Documents

ZIP file PDF file

Add documents Upload form

Proposal Show markdown Download pdf Show Proposal history

Hide Proposal history

Date of sending	State	Link
1.1.2017	Decline	Proposal
21.2.2017	Decline	Proposal

New proposal

Defence: 9.12.2017

<< 1 2 3 4 5 6 7 8 9 >>



Figure 2.9: Committee member homepage

2.1.5 Committee admin

The admin of the project committee can do anything a committee member can. Unlike them though, he or she has the power to override anything in the whole system. The intended use case for the admin is to accept and decline project proposals, open new dates for defences and select a subset of the committee to attend a defence. From time to time, he or she also creates new users (in case they cannot use CAS), issues news when something extraordinary happens and edits static pages (eg. when the official rules of the *NPRG023* course change).

1. **News** - Most news are generated automatically, but the admin can also issue news if something out of the ordinary happens (eg. when the server should go offline for maintenance.) The admin must issue news in both the Czech and English language and can specify whether the news is important, whether it should be visible to guests or not and after how long it should be discarded. The admin can also edit and delete any news, even the ones that were generated automatically.
2. **Advertisements** - The admin can delete any advertisements or their comments (eg. when improper content is presented).
3. **Project** - The committee admin can view anything about any project, including files that are visible for the team only. He or she can also carry out file-modifying operations, including deleting uploaded files, removing team members from a project, changing the supervisor, the opponent or even the consultants, or deleting the whole project. He or she is the one who accepts and declines proposals, but they should only be accepted if all members of the committee agree. Last but not least, the admin accepts requests to run a project. All the checks - whether the team comprises only students, whether the date of start of the project is reasonable and whether the size of the team is between four and eight - are carried out automatically. Therefore, the request presented to the admin should not require any investigation.
4. **Defence** - The admin opens a new date for defences anytime he sees fit. After someone from a team schedules their project for a defense (on some date), the admin chooses the exact time of day for the defence and assigns an opponent. Once enough committee members have let the admin know that they can be present during the defence, the admin selects them to attend obligatorily (not everyone who can be present has to be selected). In the event there are not enough committee members available, or some have fallen ill, the admin can restart the attendance inquiry requiring everyone to let him or her know again whether they can come or not. In case this doesn't help or some problem arises, the admin can cancel a defence or all defence scheduled for one day. If a defence is cancelled, all concerned parties are notified via email. After a defence is finished, the admin uploads (or creates and edits in markdown) a final statement and decides whether the project has passed, failed to defend or passed on a condition. The process is the same for the defence of the implementation and the analysis, except that no reviews are made for the analysis and everything about it is a little

more lightweight (the deadlines are shorter and less committee members are required to attend).

5. **Static pages** - The admin can create, edit (in markdown), show, hide and delete static pages. All static pages should be present in both the Czech and English language, even though the application works even if they are not. They are versioned and the admin decides which version is currently displayed. There is a distinction between rewriting a version of page or using copy-on-write to create a new version. Both options are available. Static pages are also available for download as PDF documents and the order of static pages in the menu can be specified by the admin. The versions work in the following way: each document with the same logical name is just a different version of the same document and new versions can be created based on an arbitrarily old version. Also, language versions of the same document have no connection to each other whatsoever from the application's point of view.

6. **Users** - As mentioned before, the admin can create new users (eg. when they refuse to use their CAS credentials to log in with this application, or if they are not members of Charles University, but for some reason they still need access to the application). He or she can edit anything - with the exception of the password - about anyone. This includes changing a user's name, role, email, expiration date of their role or login type 2.2.5. He or she can also ban a user or reload their info from CAS.

Regarding the changing of roles, it is the admin's responsibility to upgrade their roles when an academic becomes a member of the committee. When the admin wants to retire, he or she chooses a new admin, who in turn degrades the old admin's role back to committee member or academic.

7. **Dashboard** - Much like any member of the committee, the admin also has the same dashboard, but his or hers also includes projects that have sent a request to run, conditionally defended projects whose deadline has passed and finally, dates which should become open for defences, when a project needs a defence, but no suitable date is open.

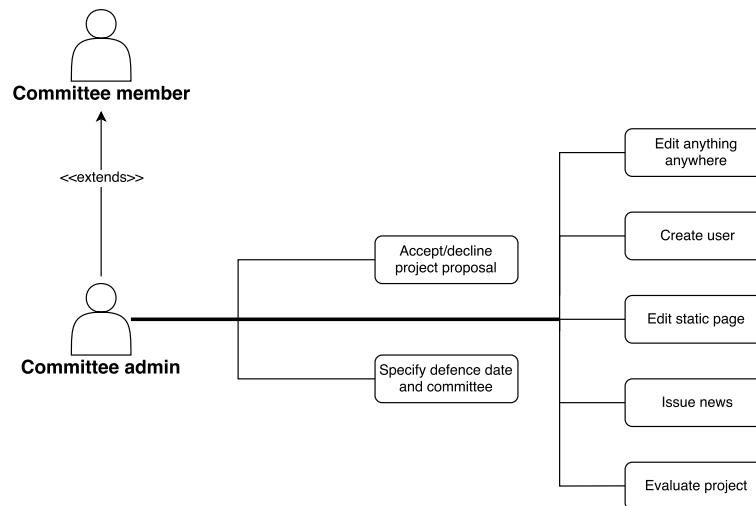


Figure 2.10: Committee admin use case

Use Case	Specify defence date and committee
<i>Primary Actor:</i>	Committee admin
<i>Main Success Scenario:</i>	<ol style="list-style-type: none"> 1. After the committee admin receives the notification that a team is ready to defend their project, the admin adds a new defence date by clicking on the <i>New Date</i> button under the <i>Defences</i> header on their homepage. He selects the date and clicks the <i>Save</i> button (A1) after which the date appears in the <i>Defense dates</i> list below. The admin clicks on the date just added to the list which displays a form where he can specify the committee members invited to the defence. He confirms the selection by clicking the <i>Save</i> button.
<i>Alternate scenario:</i>	<ol style="list-style-type: none"> A1: The admin can cancel the selection of date by clicking the <i>Cancel</i> button, in which case the date is not added to the <i>Defence dates</i> list.

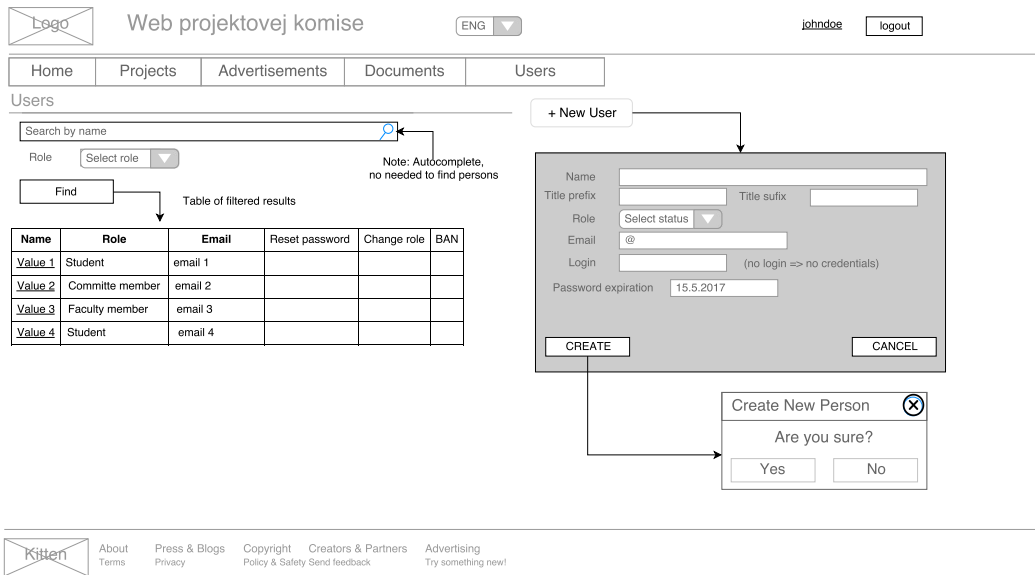


Figure 2.11: Users page

Use Case Create user


Primary Actor: Committee admin

Main Success Scenario:

1. The admin navigates to the *Users* page where he clicks the *New User* button. The admin fills in the requested information about the person he or she wishes to add to the database, including the name, titles, role in the system, e-mail address, login alias and the date when the person's password will expire. When the information is entered the user clicks the *Create* button (A1) causing a dialog to appear to confirm the creation of the new user. By confirming the process (A2), a new user is added to the system.
-

Alternate scenario:

- A1: The admin can cancel the process by clicking on the *Cancel* button.
- A2: By clicking *No* on the confirmation dialog the admin can further modify the information about the person.
-



Web projektové komise

ENG ▼
john DOE
logout

Home
Projects
Advertisements
Documents
Users

Defense 9.12.2017, Nový web projektové komise

Project Name [Nový web projektové komise](#) → Link to project page

Abbr. PRKNO
 Date 9.12.2017 Time 13:00 [Edit](#)
 Status Defended
 Supervisor [Petr Hnětynka](#)
 Consultants
 Students [Student 1](#), [Student 2](#), [Student 3](#), [Student 4](#), [Student 5](#)
 Opponent [John Doe](#) [Edit](#)
 Committee [Name1](#), [Name2](#), [Name3](#), [Name4](#), [Name5](#)

Documents [ZIP file](#) [PDF file](#)

Defense statement	<input type="button" value="Add/Edit"/>	Show! Download pdf	When the defence paper is uploaded active voting is set to NO
Opponent review	<input type="button" value="Add/Edit"/>	Show! Download pdf	
Supervisor review	<input type="button" value="Add/Edit"/>	Show! Download pdf	

Supervisor review

MARKDOWN

Upload file

review.pdf

SAVE

CANCEL

Accept

Accept on condition

Decline


Accept ○

Are you sure to accept?

Bonus credit

Yes

Cancel



About Terms
Press & Blogs Privacy
Copyright Policy & Safety
Creators & Partners Send feedback
Advertising Try something new!
Developers

Figure 2.12: Defence evaluation page

Use Case	Evaluate project (defence)
-----------------	-----------------------------------

Primary Actor: Committee admin

Main Success Scenario:

1. After navigating to the page of the defence, the committee admin presents a defence review either by typing it directly in the markdown field or uploading it as a PDF file and saves it by clicking on the *Save* button. (A1) After this, the user makes the decision either to *Accept* the project directly or *Accept it on condition*. (A2) In the pop-up dialog he can also grant extra credits for the work done after which with a click on the *Yes* button the project is approved. (A3)

Alternate scenario:

- A1: The admin can cancel the process by clicking on the *Cancel* button.
 - A2: The user can also *Decline* the project, in which the defence is unsuccessful.
 - A3: The admin can cancel the confirmation process to correct himself by clicking on the *Cancel* button.
-

2.2 Shared functionality

The rest of the specification is common to everyone, regardless of their role. It includes data export, multi-language support, email notifications and user accounts.

2.2.1 Data export

Data available to a guest is exportable via a simple REST API 2.13. The REST API is public and read-only (it only serves GET requests). The intended use of the REST API is for other web sites of the faculty to display up-to-date information about software projects. The offered output formats are XML, JSON and iCal for defence dates.

2.2.2 Internationalization

The web site is available in two languages - Czech and English. For that reason all news and static pages supplied by the committee admin should be supplied in both languages as well.

2.2.3 Email notifications

Any time something of importance happens, the concerned users are notified via email (2.9). The emails always contain a relevant link (or multiple links) to the cause of the notification.

2.2.4 Anonymization

Users can have their identity (including name and email) removed from the application after they left the system (eg. after a student leaves the university).

2.2.5 Accounts

User accounts differ greatly based on which login option the user uses. The login option also determines how the account is created and how information gets updated later on. The two login options are login via CAS or classic login where the password is stored in the local database.

1. Login via CAS -

Login via the universal CAS credentials is the preferred option, which is used by approximately 98% of the userbase, because every student studying and every professor teaching at Charles University is in possession of an account. Our system adopts the CAS system completely, meaning the "Login via CAS" button on the project's website redirects to the CAS homepage. This way the security viewpoint of our application is strengthened greatly, because we never work with user's password in any way. The user accesses his identity through the official university system (by using his login alias) after which he gets redirected back to the project's website.

After the first successful login, the user's account is created and his or her email address and login alias is downloaded from CAS. A user's role automatically expires at the end of the academic year and if they try to log in after the expiration date, their role is newly derived from the CAS info. That is, unless they are part of the committee (a member or the admin), in which case their role never expires and it prevails until it is manually changed by the admin.

2. **Classic login** - The other option is the classic login method where the password is stored in the local database. This options is intended for exceptional situations when either the user does not have CAS credentials or he or she refuses to use them. A situation like this can occur, when an external academic or an MFF alumnus needs access to the system. In both cases their accounts have to be created by the admin himself, because of which the admin has to be confident in the identity and the intentions of the person in order to not risk the system's integrity.

The users requesting access must specify their name and email address. When a user's account is created, an email is sent to them containing a link to a page where they can change their password. The same mechanism is also used when a user forgets his password and cannot log in. After the creation of the account, the person can log in using the website's GUI by specifying his email address and password.

A user's role is automatically set to expire at the end of the academic year (again, only if they are not part of the committee) and it is not possible to log in after the expiration date. To regain control of their account, these users must ask the committee admin to either prolong or change their role (eg. from student to academic).

No user can use both login methods at the same time (the committee admin can look up the method utilized by the user). In case a user who uses the classic login method, acquires a CAS account and wants to start using the login method via CAS (eg. when the user is an academic who later becomes a member of Charles University), he or she must seek the help of the committee admin who can then switch their method of login.

In case it is the other way around (eg. when the user is no longer a member of Charles University, but wants to stay in touch with software projects), the committee admin can similarly modify the type of login. As a result of this switch, the user receives an email in which he or she is asked to change their password. *Technically, one person can have multiple accounts - either if they forget that they have one and ask the committee admin to create another one or if they have a classic-login account and then acquire a CAS account and use it to log in, thus creating a new account. Both these cases (and other similar cases) are very unlikely and are the doing of a confused user. The solution to such a situation is to spot it early on and ask the committee admin to delete one of the accounts.*

```

HTTP GET export/v1/defences/id/1

{
  "id":1,
  "time":"2018-02-14T09:30:00.419Z",
  "room":"S3",
  "projectID":20,
  "projectName":"Complexity analysis of important algorithms",
  "projectShortName":"COMPA",
  "projectDescription":"This project deals with the complexity analysis of
    important algorithms.",
  "projectKeywords":"complexity, algorithm",
  "projectRunStartDate":"2017-09-24T11:12:02.419Z",
  "defenceSuccessful":"true"
}

HTTP GET export/v1/defences

[ /* array of all defences */ ]

HTTP GET export/v1/news

[ {
  "id":31,
  "titleEN":"No defences during 12th week",
  "titleCS":null,
  "contentEN":"There will be no defence dates issued during the 12th week of
    the year due to a conference in Brno.",
  "contentCS":null,
  "createdAt":"2018-01-13T15:31:06.419Z",
  "updatedAt":"2018-01-14T10:26:39.419Z"
}, ... ]

HTTP GET export/v1/projects/id/20

{
  "id":20,
  "name":"Complexity analysis of important algorithms",
  "shortName":"COMPA",
  "description":"This project deals with the complexity analysis of important
    algorithms.",
  "keywords":"complexity, algorithm",
  "team":"John Doe, Sam Williams, Jessica Twain, Rick Swan",
  "supervisor":"doc. RNDr. Petr Hnetynka, Ph.D.",
  "state":"defended",
  "runStartDate":"2017-09-24T11:12:02.419Z",
  "deadline":"2018-06-24T11:12:02.419Z"
}

HTTP GET export/v1/projects

[ /* array of all projects */ ]

HTTP GET export/v1/ads

[ {
  "id":12,
  "title":"Looking for Java/C# projects",
  "authorName":"Bc. Jan Vesely",
  "authorEmail":"jan@vesely.cz",
  "type":"person",
  "keywordsCSV":"java;c#;csharp",
  "content":"Looking for an interesting project written mainly in Java or C#.
    I'm proficient in both languages.",
  "createdAt":"2017-10-21T14:12:35.419Z",
  "updatedAt":"2017-10-24T19:11:48.419Z"
}, ... ]

```

Figure 2.13: REST API JSON data

This shows the structure of the returned data. Other format will be analogous.

Cause	Sent when	Recipients
A proposal sent to the committee	immediately	all committee members
A committee member hasn't made a decision about a proposal yet	every Monday at 8:00	the committee member in question
A committee member commented a proposal	immediately	all committee members
The committee admin accepted or declined a proposal	immediately	all committee members, the supervisor of the project
A new date open for defences	immediately	subscribers to news
A defence was scheduled for a concrete time of day or the schedule changed	immediately	the team of the project and their supervisor
At least one project is scheduled for defence on some date	every Monday at 8:00	committee members who haven't let the admin know about their presence or absence yet
Voting about defence attendance reactivated	immediately	all committee members who did not vote yes
A member of the committee was selected to attend defences on some date	immediately	the committee member in question
A defence is coming	1 week, 2 days and 2 hours before the defence	all committee members who will attend, the team of the project and their supervisor
A defence was cancelled	immediately	all committee members, the team and the supervisor of the project
A project's deadline for implementation has passed and no date is open for a defence in the next 30 days	immediately	the committee admin (also in dashboard) (a date will be offered to the admin the upcoming Friday)
A project's deadline for analysis has passed and no date is open for a defence in the next 15 days	immediately	the committee admin (also in dashboard) (a date will be offered to the admin the upcoming Friday)
A conditionally defended project's deadline for rework has passed	immediately	the committee admin
A project transferred into a new stage	immediately	the team of the project and their supervisor
A static page was published	immediately	all committee members and news subscribers
News were added	immediately	subscribers to news
An advertisement was commented	immediately	the author of the advertisement (if not disabled)
A user's role was changed	immediately	the user in question
A user stopped being a supervisor, an opponent or a team member on some project	immediately	the user in question
A new temporary URL for changing a user's password was generated	immediately	the user in question
The application cannot connect to CAS	immediately	the committee admin
The application cannot connect to the database	immediately	the committee admin
An error occurred	immediately	the committee admin

Table 2.9: Email notifications

Users are immediately notified via email about anything they could be interested in. There are notifications about proposals, defences, the state of a project, miscellaneous (about users' roles, advertisements, static pages and news) and also ones about technical problems.

3. Analysis

3.1 Technologies

The choice of programming languages is greatly influenced by our choice that the application will be a classic web application (not single-page), that uses ajax to speed things up. SPAs are extremely responsive, but they are also harder to design and implement and have trouble when the internet connection drops. Both single-page and classic design are for this project equally suitable, but the main reason for choosing the classic design is that an application of this size needs a framework (to speed up development, reduce bugs and decrease the volume of tests) and our team has good experience with classic (backend) frameworks and limited experience with SPA ones.

3.1.1 Backend (PHP, Yii)

For backend development, the most prominent languages are PHP, C# (asp), JavaScript (node.js) and Python. Each of them has some pros and cons. Our choice was PHP due to several reasons like the popularity and the size of the community of developers. Based on the current statistics, it is the most common language for the backend web development. Another good indicator is the number of posts on stackoverflow.com web page with number more than twice as for any other language for backend development mentioned before. The next and fairly important reason is that our team has the experience with PHP and the same is true for those who will take over the project after it is finished.

We came to a decision of using Yii as a PHP framework. It is one of the most popular frameworks supporting the latest version of PHP with typing/type hinting feature and is fully object-oriented. The coding experience is comparable to Java due to a great code readability and maintainability. Framework solves most common backend problems - DB abstraction, application configuration, routing, security against all common attacks, sessions, caching, object-relational mapping, etc. Other PHP frameworks has similar architecture, but do not cover some of the needed functionality that Yii has. This choice was made also due to the fact, that some members of the team have more experience with it than with any other PHP framework. Additional libraries or Yii plugins will be used as needed (eg. a CAS library, etc.).

Nette framework was considered, but refused in favor of Yii. Both frameworks cover the same basic functionality (routing, security, config, DI, DB abstraction, etc.) and both have their pros and cons - there is definately not a clear winner. However, Yii was chosen because it has faster pageloads, is more robust (comes with ORM, i18n and widgets out-of-the-box), has better documentation and it probably has more/better plugins.

3.1.2 Frontend (JS, jQuery)

The programming language for frontend is JavaScript (ES5). Using an elevated form of JS, such as TypeScript, CoffeeScript or newer version - ES6 would

bring unnecessary complexity due to the need for code transpilation. This increased complexity is not worth of using those languages taking into consideration the fact, that the application is orientation more to backend than frontend. The jQuery framework/library will be used. JQuery became basically a standard for frontend development providing a vast number of methods for DOM object manipulation, UI components, AJAX request, etc. Other libraries (eg. a markdown library, UI libraries and jQuery plugins) will be used as needed.

3.1.3 Database (MySQL, Yii Database)

Based on the project structure, SQL database came to be the best fit for the project. The other option is NoSQL databases, which are designed for a system with huge amounts of data and lack of ACID transactions, which are essential for this project. Due to the fact that there are only 420 software projects from the last 20 years, we may assume, the size of the data in the database is not going to be big. The only NoSQL databases offering ACID transactions are graph databases. The project needs a strict database schema with integrity constraints. We were considering several common SQL engines like MySQL, MS SQL, PostgreSQL, SQLite and Oracle DB. Oracle does not have a good support in Yii framework and MS SQL is paid, so these options were rejected. SQLite is suitable for the small application and does not provide as wide functionality as MySQL. MySQL is an open source database, which came to be the most suitable choice due to the functionality, experiences by team members and a wide community of developers and the usage in the world of web development. Yii framework also provides object-relational mapping as a database abstraction layer, which will be used in combination with MySQL. ORM is a good choice, because it simplified development and decreases the amount of testing.

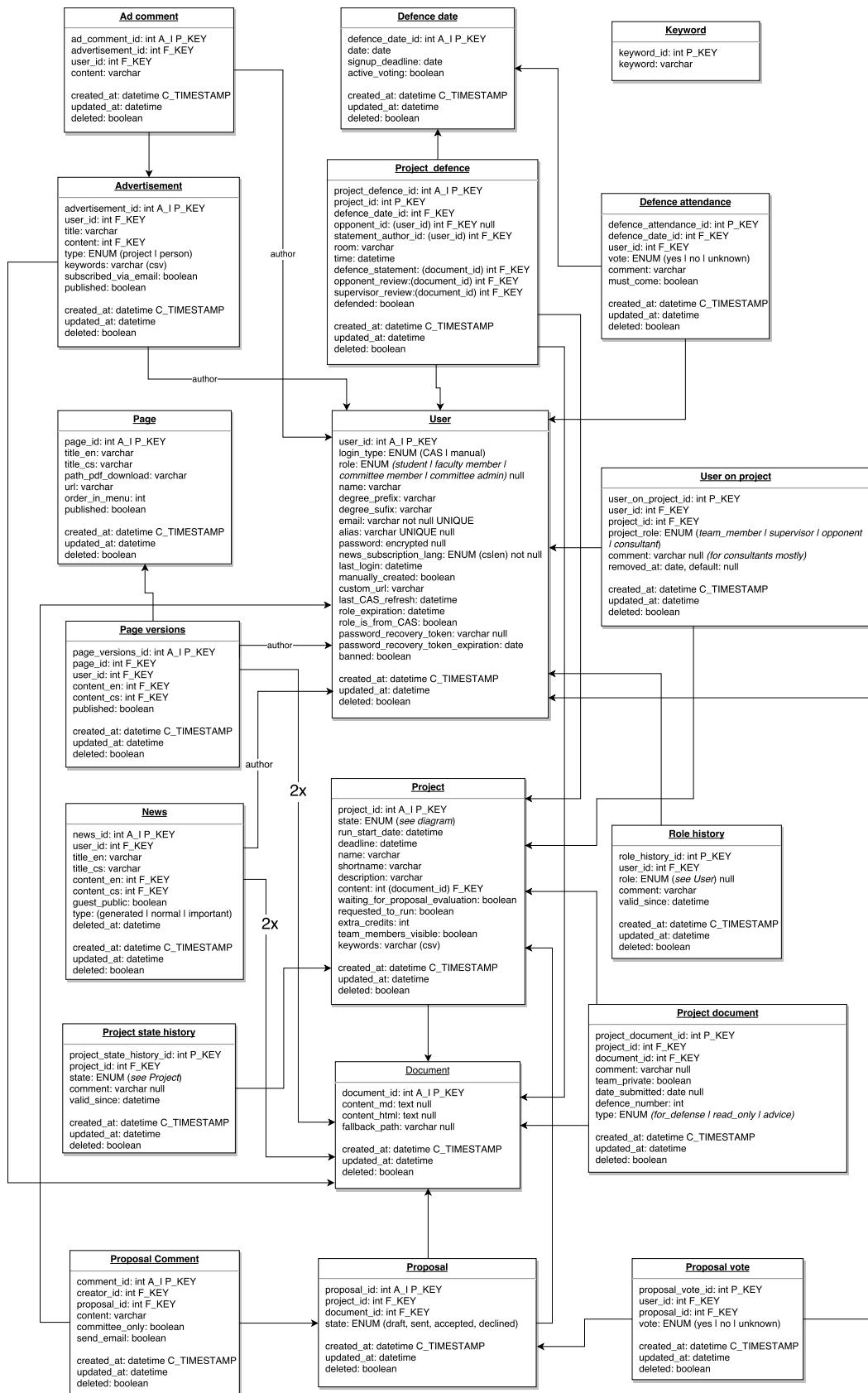


Figure 3.1: Database model

3.1.4 Typesetting (Markdown)

Markdown will be used for typesetting. It is a lightweight markup language that covers everything needed for simple typesetting (bullet points, numbered lists, different levels of headings, boldness, italics, strike-through, hypertext links, etc). There are several free JavaScript and PHP libraries that can render it.

There are a lot of makrups with very similar syntax and capabilities to Markdown. From the technical point of view there is no clear winner - on the contrary. The languages are almost exactly the same and most of them have a PHP engine. However Markdown is chosen for a different reason - its popularity. It is being used on github.com and bitbucket.com for wikis and readme files (by default) and it can also be used for typesetting stackoverflow.com question. Therefore, current and future students are most likely to be familiar with Markdown and not anything else and therefore Markdown is the most user-friendly option.

Here are a few examples of Markdown competitors. Creole - an effort to create a markup language as the intersection of all current markups, syntax close to Markdown, has a PHP engine. ReStructuredText - syntax close to Markdown, has a PHP engine. AsciiDoc - syntax close to Markdown plus a little more options than Markdown (color, tables, even variables, built in image icons), does not seem to have a PHP engine. Mediawiki markup - syntax different from Markdown (different characters, everything is a little bit longer to write), does not seem to have a PHP engine. Textile - syntax close to Markdown plus more options (custom CSS, subscript, superscript, ctation, footnotes, etc.), has a PHP engine. The syntax and features of Textile are actually quite nice, if Markdown wasn't so much more popular, Textile would be a great candidate.

Other alternatives are HTML and TeX. HTML is too hard to write and Markdown supports a subset of inline HTML anyway, so if someone really wants to use it they can. TeX is very popular, but first, it's meant for very precise typesetting of text for print, not HTML and second, it relies heavily on downloading packages and that would bring in unnecessary problems.

3.2 Architecture

As mentioned in the previous section, the application will be backend-oriented. Diagram 3.2 shows the big picture and interaction with CAS, diagram 3.3 describes the architecture of the backend with a PHP framework in mind. Frontend may be quite complex, since it will use quite a few libraries and the potential problems with using JS libraries cannot be overlooked. However the actual custom code on frontend will be minimal and that is why there is no diagram for it.

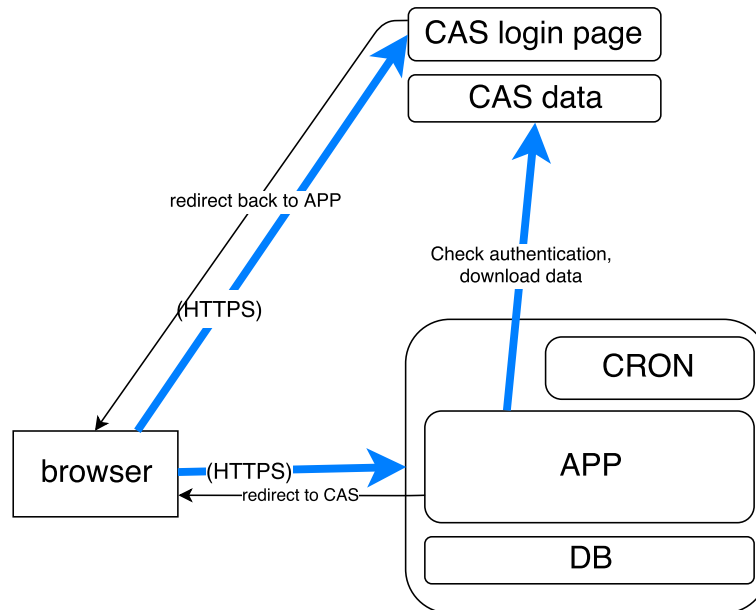


Figure 3.2: High level architecture, interaction with CAS

This is a static layered modular usage viewpoint of the architecture. Each layer uses the one directly below it, arrows represent HTTP requests and responses. Users will be redirected to CAS to log in and then back to our application after a successful login. At first login a user will be automatically registered and his data downloaded from CAS without him noticing. CRON jobs will be used to send out periodic emails and perform other tasks.

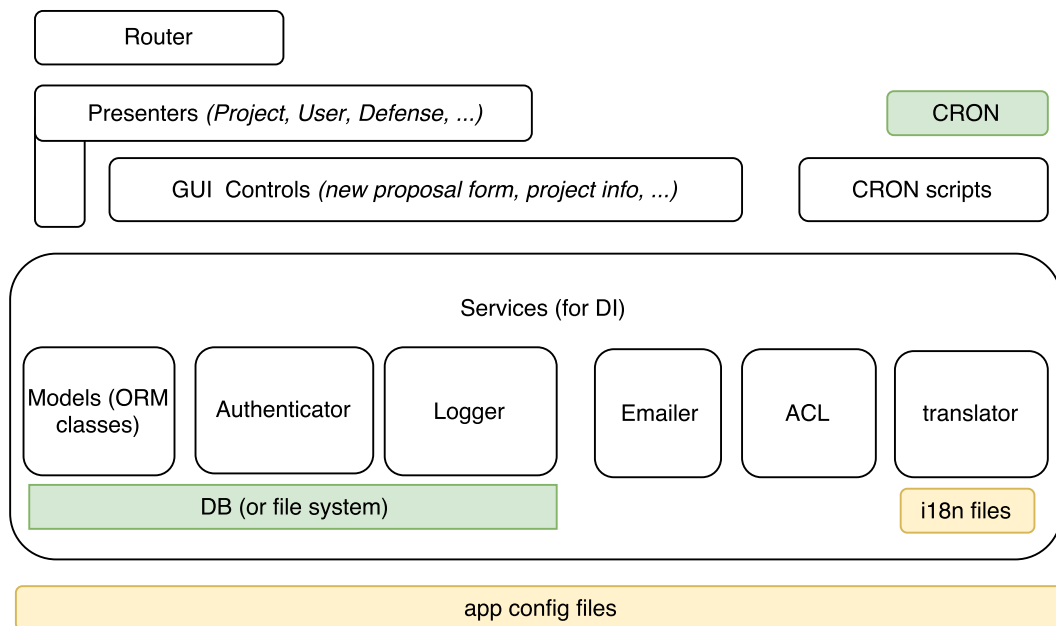


Figure 3.3: Backend architecture

This is a static layered modular usage viewpoint of the architecture. Each layer uses the one directly below it. Green modules are external programs, yellow modules are files. Services are classes that contain application logic and are injected (via dependency injection) into GUI controls and presenters (which render GUI and react to user's input). A router routes to the correct presenter. Controls are just reusable parts of a presenter. All services are configured via a config file. The DB can be accessed directly or via models (ORM-mapped classes).

3.3 Risk analysis

Table 3.1 lists the most prominent adversities that may arise during development along with advice on how to handle them.

problem	prob.	consequence (1 inconvenient - 9 critical)	mitigating action
one teammate works only half the time	30%	see (2)	remove them, goto (2)
(2) one teammate stops working	24%	7 (4 if spec is reduced), not enough time	reduce spec, work gets split among the rest of the team, expect to finish after 9 months
two or more teammates work only half the time	10%	7, not enough time	bring the whole team to their level, expect to finish after 12 months
Markdown library fails/is bugged	30%	1, inconvenience	find another and use it (about 1 week work)
Yii turns out to be seriously bugged	3%	9 (2 if workaround exists), cannot use it	report bug and try to work around. If not successful, learn and switch to a similar framework (perhaps Laravel) and expect to finish after 10 months
two teammates stop working	2%	9 (4 if spec is reduced), not enough time	reduce spec, work gets split among the rest of the team, expect to finish after 12 months (1st defense will fail)(*)
not using ORM turns out to be a huge nuisance	10%	1, inconvenience	consider switching to Doctrine (about 2 weeks work)
MySQL turns out to lack some needed functionality	1%	4 (2 if workaround exists), cannot use it	try to work around, otherwise switch to postgresSQL (also consider SQLite)
three teammates stop working	0%	9, not enough time	quit

Table 3.1: Risk analysis

The most consequential and the most probable adverse situations are those where a part of the team fails to put in sufficient effort or leaves the project. This can be caused by an illness, problems at work or in school, etc.

3.4 Milestones

- 1.2. - 1.3. - write a complete specification and analysis, study PHP framework and libraries
- 1.3 - 1.4 - set up development environment, implement functionality concerning users and projects, create test data for the DB
- 1.4. - 1.5. - write an import script (for existing XML data), implement REST API for data export and functionality concerning defenses
- 1.5. - 1.6. - write half of the documentation, implement functionality concerning static pages and advertisements
- 1.6. - 1.7. - finish the documentation, implement homepage/dashboard and functionality concerning news and finish end-to-end tests

(unit tests will be done along the way)

3.5 Work split

- Artur Finger - specification, documentation, helping others, code reviews, merging, deployment
- Zoltán Betteš - main programmer
- Slávka Ivaničová - second main programmer, misc
- Gergely Tóth - DB, data import and export, misc
- Marek Beňovič - testing, documentation