Call Calendar

Requirements Definition Document

Version 0.1

Title:
Call Calendar

Course:
Distributed Software Development

Document:
Requirements Definition

Participants:

Biljana Stanić
Md. Abdur Razzaque
Hrvoje Pavlović
Marko Vuglec
Marko Veličković
Damian Marušić

Supervisors:
Federico Ciccozzi
Ivana Bosnić

Date:
14th November 2014

Revision History

Date	Version	Description	Author
2014-11-12	0.1	Initial Draft	Biljana Stanić
2014-11-12	0.2	Added use cases	Hrvoje Pavlović
2014-11-12	0.3	Added list of requirements	Marko Vuglec
2014-11-12	0.4	Created backlog table	Marko Veličković
2014-11-12	0.5	Division of work	Abdur Razzaque
2014-11-12	0.6	Modified user stories	Marko Veličković
2014-11-12	0.7	Introduction part	Biljana Stanić
2014-11-12	0.8	Short background, related documents	Biljana Stanić
2014-11-12	0.9	High-level design introduction, requirements	Biljana Stanić
2014-11-12	0.10	Extended the content of the requirements table (id, motivation, source)	Biljana Stanić
2014-11-13	0.11	Assigned priority for requirements	Damian Marušić
2014-11-13	0.12	Extended use case diagrams	Damian Marušić
2014-11-14	0.13	Wrote detailed explanation for the use case diagrams	Abdur Razzaque, Damian Marušić
2014-11-14	1.0	Transferred everything to word/pdf and publish	Biljana Stanić

Table of Contents

1. Introduction

1.1 Purpose of this document

The purpose of this document is to describe requirements of the system that need to be developed. This will be used initially to ensure that we have common view of the tasks (within the group, and between the group and the customer). It will later be used to guide the development, and it is also one criterion that can be used to measure the quality of the delivered system.

1.2 Document organization

The document is organized as follows:

- Section 1, *Introduction,* describes purpose and audience of this document, scope, definitions and acronyms.
- Section 2, *Short background,* introduces project, customers, supervisors and related documents.
- Section 3, *High level description of the domain*, includes prerequisites for development process
- Section 4, *Requirements*.
- Section 4, *Requirements*.
- Section 4, *Use case diagrams*.
- Section 4, *Division of work*.

1.3 Intended Audience

The intended audience is:

- Team members;
- Local and remote supervisors;
- Customers.

1.4 Scope

This document provides a detailed list and description of the requirements of the system that will be developed. Requirements are defined with the customers, priorities have been assigned, motivation for their use, etc. Here are presented use case diagrams where are listed functional requirements. Furthermore, non-functional requirements have been identified and explained.

1.5 Definitions and acronyms

1.5.1 Acronyms and abbreviations

In the following table will be present and explained abbreviations that will be used in the document.

Acronym or abbreviation	Definitions
MDH	Mälardalen University, Västerås, Sweden
FER	Faculty of Electrical Engineering and Computing, Zagreb, Croatia
DSD	Distributed Software Development
ES	Embedded Systems
RECO	Research Coordination

Table 1. Abbreviations with their explanations

2. Short background

The division for Research Coordination (RECO) constantly looks out for funding possibilities, helps the researchers with updated information about conditions, budget, consortia, and, also, helps putting the applications together. It is crucial to keep calls and deadlines in mind. RECO and the ES researchers would all benefit from a ”Call calendar” containing information about funding bodies, different calls and their respective deadlines.

The goal of this project is to extend current functionalities of existing system at ES. Currently, there are researchers profiles/areas of interest, funding bodies and funding programs. System is divided into back and front office. Back office is used to manage researchers, projects and funding agencies. In front office researchers profiles/areas of interest and funding bodies/programs are shown.

System will be upgraded so it could support adding funding agencies calls, as well as researches' applications to specific calls. It will also match calls with researchers profiles/research interests in order to ensure maximum user satisfaction. Moreover, it will contain information about different calls and their respective topics, deadlines and links to more information. Alongside viewing available calls list, in front office, functionality of sorting and filtering calls will be added. Users will be able to get various generated reports about calls and applications as well.

2.1 Customer

Customers for the "Call Calendar" project are from research center Embedded Systems:

- Malin Rosqvist, Research Coordinator, Web project manager at ES

Email: malin.rosqvist@mdh.se
Web page: http://www.es.mdh.se/staff/215-Malin_Rosqvist

- Irfan Šlijvo, Web master & Presta shop developer at ES

Email: irfan.slijvo@mdh.se
Web page: http://www.es.mdh.se/staff/380-Irfan_Sljivo

2.2 Supervisors

The team has two supervisors, one local and one remote (based on the location of the team) situated in Sweden and Croatia. They are:

- Federico Ciccozzi

Email: federico.ciccozzi@mdh.se

- Ivana Bosnić

Email: ivana.bosnic@fer.hr

2.3 Related documents

In the document "*Project Plan Document.pdf*" is presented a detailed description of the project.

3. High level description of the domain

“Call Calendar” will be an upgrade of the existing system at ES that is powered by a Prestashop database. Researchers’ profiles/areas of interest can be found here and also funding bodies and funding programs. For this project we have to use model-view-controller (MVC) architecture pattern, because Prestashop is based on it. MVC splits application on three logical components and defines interactions between them.

In our “Call Calendar” project, every object that will be created should have a corresponding class where will be specified its properties. To use object, we will create controllers for front- and back-office. In these controllers will be specified information that will be presented in the specific pages in the front/end and forward this information to the smarty templates. Smarty templates are designed to display the information in the desired way. The back-office comes with predefined templates. When adding a page to the front, corresponding redirect page should be added in the root directory prestashop-skeleton. Notice the naming convention between this file, the corresponding front controller and the included template file.

3.1 Prerequisites for development process

Preconditions that have to be fulfilled during development process of the project are listed below.

Mirror server:
- Finish configuration;
- Secure it.

Product development environment:
- PHP editor - notepad++;
- Install Github support;
- Install and configure XAMPP.

Documents:
- Project Plan v.1;
- Requirements Definition v.1;
- Design Description v.1.

Modifications to existing database:

Add appropriate tables and fields (after conversation with customers)

4. Requirements

In this section will be presented all requirements that are captured from customers. For each of them, functional and non-functional, will be assigned certain priorities. Priorities can be high, medium and low. This means that requirements with that are labeled as “high” priority will be implemented first because they are crucial for those that have “medium” and “low” priority. Moreover, “medium” priority requirements are those that are listed as desirable to be implemented. While, “low” priority requirements can be implemented but they are not so important for the normal functioning of the system (some additional features).

In the following table are listed all requirements, with corresponding ids, priorities, names, description, motivation and source.

ID	Functional	Name	Description	Motivation	Priority	Source
FR-01	Yes	Add new call	Add the call to the specific funding agency	1	High	Malin and Irfan
FR-02	Yes	View all calls	View all past/active calls	2	High	Malin and Irfan
FR-03	Yes	Filtering calls by	Filter calls by funding	2	Medium	Malin and

		agencies	agency's name			Irfan
FR-04	Yes	Filtering calls by status	Filtering calls by status (past/active)	2	Medium	Malin and Irfan
FR-05	Yes	View specific call	Detail view of specific call	5	High	Malin and Irfan
FR-06	Yes	Send notification to researchers	Send email notification to all researchers who are interested in call	8	Medium	Malin and Irfan
FR-07	Yes	Adding application form to the call	Researcher's application is added to the specific call	6	High	Malin and Irfan
FR-08	Yes	Storing application's forms	All created applications are saved and stored	11	High	Malin and Irfan
FR-09	Yes	Viewing and generating reports about the call	Generate report with all important information about the specific call	7	Low	Malin and Irfan
FR-10	Yes	Generating reports about all applications for the call	Generate report about all applications for the specific call	11	Low	Malin and Irfan
FR-11	Yes	Custom filters and possibility to export results to excel	Filter calls and export retrieved data to excel file	14	Medium	Malin and Irfan
FR-12	Yes	Custom filter - option to choose which fields to return	Filter calls using custom fields	16	Low	Malin and Irfan
FR-13	Yes	Create a project	The project should be created once the application is labeled as "granted"	17	High	Malin and Irfan
FR-14	Yes	Create a news feed	For every granted application should be created a news feed on the web page	18	Medium	Malin and Irfan
NF-01	No	Performance (SQL optimization)	Faster query at runtime	13	High	Project group
NF-02	No	User experience	Quick response of the given function	13	Medium	Project group
D-01	No	MoM	Minutes of meeting, document that contains information about the meetings that were held in the past week	15	High	Supervisors
D-02	No	Weekly reports	Document that contains information about team activities of the past week	15	High	Supervisors
D-03	No	Project plan	Document that contains details about the team, project vision and goals to be achieved	15	High	Supervisors
D-04	No	Requirements definition	Document that contains what is supposed to develop	15	High	Supervisors

D-05	No	Test report	The test report captures and summarizes the test results	15	High	Supervisors

Table 2. Requirements definition

Our product backlog consists of functional requirements, non-functional requirements and documents. The ID determines each entry. The motivation for the specific requirement is shown in the following table.

ID	User story	Priority	Status
1	As a funding agency, I want to add call for a specific topic, so that researchers can apply for funding	High	**Not started**
2	As the funding agency, I want to view all applications for a specific call, so that I can select researcher among them	High	**Not started**
3	As the funding agency, I want to generate a report of all applications for a specific call, so I can have a file that can be stored	Low	**Not started**
4	As the funding agency, I want to grant for a certain application, so the researchers can work on the project	High	**Not started**
5	As the funding agency, I want to view all applications for a specific call and filter those applications by money requested, application status (submitted or In progress), so I can choose the suitable one	Medium	**Not started**
6	As a researcher, I want to see all active calls, so that i can decide if I am interested in it	High	**Not started**
7	As the researcher, I want to see details about each call with all documents, so that I can apply	High	**Not started**
8	As the researcher, I want to get notification for calls offered by funding agencies, so I can choose the interesting ones	Medium	**Not started**
9	As the researcher, I want to generate a report of all calls offered by funding agencies, so I can have a insight for all possible offers	Low	**Not started**
10	As an admin (RECO), I want to filter funding agencies and researchers, so that I can show them in calendar or in front office	High	**Not started**
11	As the admin (RECO), I want to view all past & active calls, so that I can generate a report	Low	**Not started**
12	As the admin (RECO), I want to match calls with researcher profiles, so I can ensure a maximum user satisfaction	High	**Not started**
13	As an user, I want a fast page load, so I can have a lower response time	Medium	**Not started**
14	As the user, I want to get a certain information and export it, so I can have it in a form of a file	Low	**Not started**
15	As a supervisor, I want to have insight into the progress and development of the project, so I can advise and improve possible difficulties	High	**Not started**
16	As the user, I want to specify which information about calls I want to get as a result, so I do not get unnecessary information	Medium	**Not started**
17	As the founding agency, when I approve the application, I want a project to be created automatically, so I can have the project related to the application	High	**Not started**
18	As the founding agency, when I approve the application, I want a news feed to be created automatically, so I can have that information displayed publicly on the web page	Medium	**Not started**

Table 3. Motivation for the specific requirement

5. Use case diagrams

5.1 Actors

Funding agency: Institution which wants to publish funding calls and attract researchers. Has to be previously registered in system and have access to back office.

Researcher: Person that does researches and has interest in finding funding agency that may finance their research/project. Has to be previously registered in system and have access to back office.

There may be also some other parts of the system considered as actors which are not shown on use case diagrams below. The reason for not displaying them is to keep diagram simple and also because their main activity is performing some background tasks. "System" is one such actor and it performs various tasks in background (sending notifications, identifying potentially interested researchers, communicating with database, etc.). Database itself may also be considered as actor because it is used to stored various data.

5.2 Use case models

Two examples of use cases will be considered in this section. One is Funding Agency Use Case and the other is Researcher Use Case.

5.2.1 Funding Agency Use Case

We have identified funding agency as one of the users on the system. In the following figure will be shown all use cases that are captured for this actor.

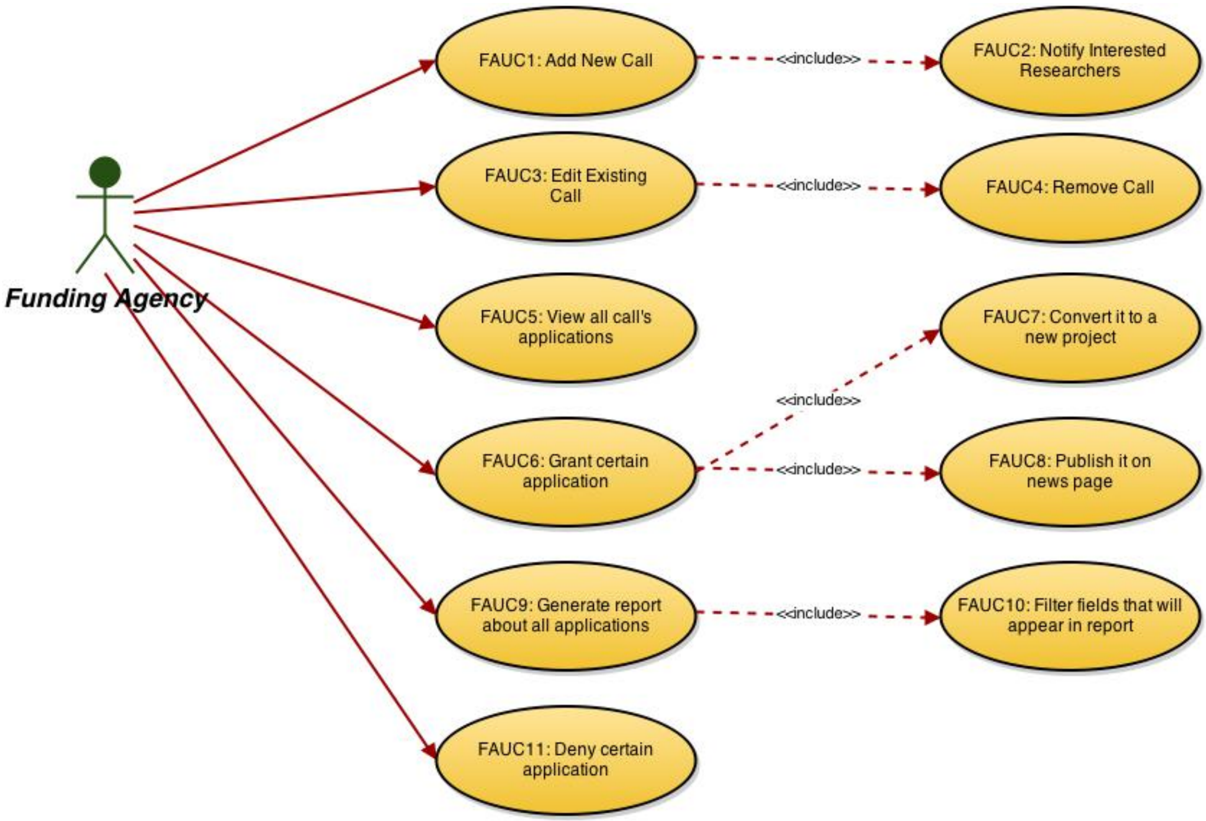

Figure 1. Use case diagram for the funding agency

Other actor is researcher that has more limited functions on the system comparing to the funding agency. In the following figure are shown use cases for the researchers as actors on the system.

Use case name:	**FAUC1: Add New Call**
Participating actor(s):	Funding Agency
Preconditions:	Funding Agency has access to back office
Basic Flow:	1. Funding agency decides to make a new call 2. Funding agency fills in form 3. Funding agency submits filled in form 4. New call is made and stored
Alternative Flows:	3a. Requested fields are incorrectly filled in 1. Alert about incorrectly filled fields is shown
Postcondition:	New call stored
Special requirements:	

Use case name:	**FAUC2: Notify Interested Researchers**
Participating actor(s):	Funding Agency System
Preconditions:	Call is made
Basic Flow:	1. Funding agency makes a call 2. System identifies all the researchers that may be interested in call

	3. System notifies all the researchers that may be interested in call
Alternative Flows:	2a. No interested researcher found 1. No notifications sent
Postcondition:	Potentially interested researchers are notified about certain call
Special requirements:	

Use case name:	**FAUC3: Edit Existing Call**
Participating actor(s):	Funding Agency
Preconditions:	Funding Agency has access to back office Funding Agency has created call
Basic Flow:	1. Funding agency chooses to edit previously made call 2. Funding agency makes changes 3. Funding agency submits changes
Alternative Flows:	3a. Some fields are incorrectly filled in 1. Alert about incorrectly filled fields is shown
Postcondition:	
Special requirements:	

Use case name:	**FAUC4: Remove Call**
Participating actor(s):	Funding Agency
Preconditions:	Funding Agency has access to back office Funding Agency has created call
Basic Flow:	1. Funding Agency decide to remove existing call
Alternative Flows:	
Postcondition:	Researchers will be notified
Special requirements:	Call is made

Use case name:	**FAUC5: View all call's applications**
Participating actor(s):	Funding Agency
Preconditions:	Funding Agency has access to back office Funding Agency has created call
Basic Flow:	1. Funding Agency views all applications for a specific call
Alternative Flows:	
Postcondition:	Funding Agency has insight in all applications made for certain call
Special requirements:	Researcher’s applied for a call

Use case name:	**FAUC6: Grant certain application**
Participating actor(s):	Funding Agency
Preconditions:	Funding Agency has access to back office Funding Agency has created received application(s) to previously made call
Basic Flow:	1. Funding Agency picks application to grant 2. Funding Agency grant application for a certain call
Alternative Flows:	
Postcondition:	Application is granted and will be become new project
Special requirements:	

Use case name:	**FAUC7: Convert granted application to a new project**
Participating actor(s):	Funding Agency System
Preconditions:	There is granted application
Basic Flow:	1. Granted application is automatically converted to new project and stored
Alternative Flows:	
Postcondition:	New project in database
Special requirements:	

Use case name:	**FAUC8: Publish granted application on news page**
Participating actor(s):	Funding Agency System
Preconditions:	There is granted application
Basic Flow:	1. Granted application automatically generates news about it on the news page
Alternative Flows:	
Postcondition:	News on the news page
Special requirements:	

Use case name:	**FAUC9: Generate report about all call's applications**
Participating actor(s):	Funding Agency
Preconditions:	Funding Agency has access to back office Funding Agency has created call
Basic Flow:	1. Funding Agency wants generated report about all calls from application 2. Funding Agency gets report
Alternative Flows:	
Postcondition:	Generated report for Funding Agency
Special requirements:	

Use case name:	**FAUC10: Filter fields that will appear in report**
Participating actor(s):	Funding Agency
Preconditions:	Funding Agency has access to back office Funding Agency has created call
Basic Flow:	1. Funding Agency wants generated report about all calls from application 2. Funding Agency select which fields from application it wants 3. Funding Agency gets report
Alternative Flows:	
Postcondition:	Generated report with filtered fields is created
Special requirements:	

Use case name:	**FAUC11: Deny certain application**
Participating actor(s):	Funding Agency
Preconditions:	Funding Agency has access to back office

	Funding Agency has created call
Basic Flow:	1. Funding Agency selects application that it wants to deny 2. Funding Agency denies selected application
Alternative Flows:	
Postcondition:	Application is marked as denied
Special requirements:	

5.2.2 Researcher Use Case

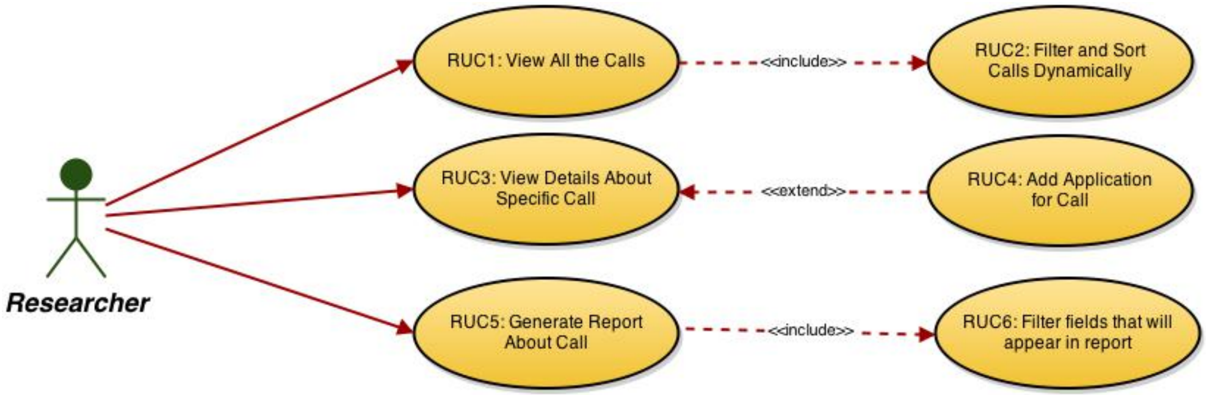

Figure 2. Use case diagram for the researcher

Use case name:	**RUC1: View All the Calls**
Participating actor(s):	Researcher
Preconditions:	
Basic Flow:	1. Researcher opens all calls 2. List of all calls is shown
Alternative Flows:	2a. Database connection is not available 1. Appropriate message is generated
Postcondition:	Listed all calls
Special requirements:	

Use case name:	**RUC2: Filter and sort calls dynamically**
Participating actor(s):	Researcher
Preconditions:	
Basic Flow:	1. Researcher wants to get filtered and/or sorted call 2. Researcher chooses filtering and/or sorting parameters 3. Researcher get list of filtered/sorted calls
Alternative Flows:	
Postcondition:	Filtered and/or sorted calls are shown
Special requirements:	

Use case name:	**RUC3: View details about specific call**
Participating actor(s):	Researcher
Preconditions:	There is at least one call

Basic Flow:	1. Researcher chooses specific call 2. Details about that call is displayed
Alternative Flows:	
Postcondition:	Details about specific call are shown
Special requirements:	

Use case name:	**RUC4: Add application for call**
Participating actor(s):	Researcher
Preconditions:	Researcher is logged in into the back office
Basic Flow:	1. Researcher decides he wants to apply for certain call 2. Researcher fills in a form 3. Researcher submits form 4. Application is made and stored
Alternative Flows:	3a. Requested fields are incorrectly filled in 1. Alert about incorrectly filled fields is shown
Postcondition:	New call application
Special requirements:	

Use case name:	**RUC5: Generate report about call**
Participating actor(s):	Researcher
Preconditions:	Researcher has access to back office
Basic Flow:	1. Researcher wants report about specific call 2. Report about call is generated
Alternative Flows:	
Postcondition:	Generated report about call
Special requirements:	

Use case name:	**RUC6: Filters fields that will appear in report**
Participating actor(s):	Researcher
Preconditions:	Researcher has access to back office
Basic Flow:	1. Researcher wants report about specific call with filtered fields 2. Researcher chooses which fields he wants in report 3. Researcher wants report about specific call
Alternative Flows:	
Postcondition:	Generated report about call with filtered fields
Special requirements:	

6. Division of the work

In order to better handle the implementation process, we have agreed to split the work into several parts. Moreover, for every specific part is assigned a small group that consists of two-three team members. This division is based on the experience and interests of the members for the specific task.

Role	Team Member(s)
Database Manager	Marko Vuglec, Hrvoje Pavlović
Front-End Manager	Biljana Stanić, Marko Veličković
Back-End Manager	Damian Marušić, Abdur Razzaque

Table 4. Division of work