Requirements Analysis, Software Design and Test Documentation

Mehrdad Khodaverdi mehr@itu.dk

Simon Alexander Kern siak@itu.dk

Thomas Bo Christiansen tbch@itu.dk

Nicklas Doolewerdt Mariager Johansen ndom@itu.dk

GlitterVersion 1.0

Analysis, Design and Software Architecture (BDSA)

September 25, 2015

Revision history

Date	Ver.	Description	Ref.	Author
08/09-15	0.1	Added requirements and scattered notes		siak tbch ndom
10/09-15	0.2	Creating scenarios and use cases		mehr siak tbch ndom
12/09-15	0.3	Finishing scenarios and use cases for first hand in		mehr siak tbch ndom
12/09-15	0.4	Setting up TeX for all project		mehr
25/09-15	0.5	Changing RAD according to feedback		mehr siak ndom

Preamble

This report is done as part of the Analysis, Design, and Software Architecture (BDSA) course at the IT University of Copenhagen. It provides an outline and some initial examples of the content of a report designed to document software systems. In particular, this outline is a useful starting point for the report, which should document the BDSA project done in the end of the course.

Contents

1 Introduction

1.1 Purpose of the System

To create a support tool for researchers in order to perform SLR and SMS. The problems with manually creating SLR and SMS, is that they are time consuming, highly susceptible to errors and require cooperation of researchers (maybe even multiple teams).

1.2 Scope of the System

The scope of this project is mainly the backend server side (Study Configuration Server) but also the UI for configuring the study (Study Configuration UI).

1.3 Objectives and Success Criteria of the System

Has the following success criteria:

- The system shall provide distributed research teams with a support tool to conduct SMS or SLR studies.
- The system shall support the activities of planning the review and conducting reviews. The system shall allow for easy deployment and dissemination.
- The system must be a better alternative to Excel and other spreadsheet software.
- The system must fulfil all functional and non-functional requirements defined in the section proposed system.

1.4 Definitions, acronyms and abbreviations

- SLR: Systematic literature reviews
  - SLR aims to precisely answer one or more focused research questions.
- SMS: Systematic mapping studies
  - SMS aims to provide a wide overview of a research area.
- UI: User Interface
  - UI is the space where the interaction between humans and machines occur

Part I

Requirement Analysis Document

2 Current System

It is assumed that there is currently no software available for the task, however, it is mentioned that spreadsheet software such as Microsoft Excel, can provide basic support.

3 | Proposed System

3.1 Overview

3.2 Functional Requirements

The system supports the following:

- The researcher must be able to conduct SMS and SLR through the use of the system
- The system must be able to manage multiple research teams
- The system must be able to manage multiple studies
- The system must be able to review progress
- The system must be able to generate research protocols.
- The system must be able to divide research work between team members (different assignments and percentages).
- The system must be able to create and manage inclusion / exclusion criteria.
- The system must be able to review the status of the review at any point.
- The system must be able to create and manage classification criteria (research type facets and contribution type facets).
- The system must be complemented with an installation and a user manual.
- A research team must have an admin/leader
- A study must have an admin/leader

3.3 Nonfunctional Requirements

3.3.1 Usability

The system should be easy to use for all users. This boils down to the following usability requirements:

- A user should be able to fill out the study configuration UI in "x" seconds.
- should be able to see information...
- The user should be able to have a complete overview of the their personal tasks.
- The user should be able to quickly and easily update information...

3.3.2 Reliability

The system should be reliable (dependable, robust, and safe) in the following manner:

- No bugs should crash the system but instead display an error message with information regarding the error.
- The system should be able to recover from a system failure (restart).

3.3.3 Performance

- The system should retrieve the papers which passed the inclusion / exclusion criteria within 1 seconds.
- The system should add an entry into the DB within "x" seconds
- The system should be available 99

3.3.4 Maintainability

- The system should be extensible.
- The database should be maintainable / updatable
- The system should be easy to deploy.
- The system should be flexible in terms of being able to deploy on different/ new hardware and software technology.

To be determined.

3.3.5 Implementation

- The system should be publishable as Open Source Software.
- The system should fit the imposed interface.
- The system should be developed using C# and WPF.
- The system should be testable.

3.3.6 Interface

- The system should import data via (.bib).
- The system should export data in .csv as a minimum (multiple formats?).

3.3.7 Packaging

- The end user should be able to install the program.

3.3.8 Legal

- Should be able to be published as an open source software, and therefore adhere to the danish copyright act.
  - Intellectual property law
    - `http://mwblaw.dk/Doing%20Business%20in%20Denmark/Intellectual%20property%20law.aspx`
  - Consolidated Act on Copyright 2010 (Consolidated Act No. 202 of February 27, 2010)
    - `http://www.wipo.int/wipolex/en/details.jsp?id=7394`

4 | System Models

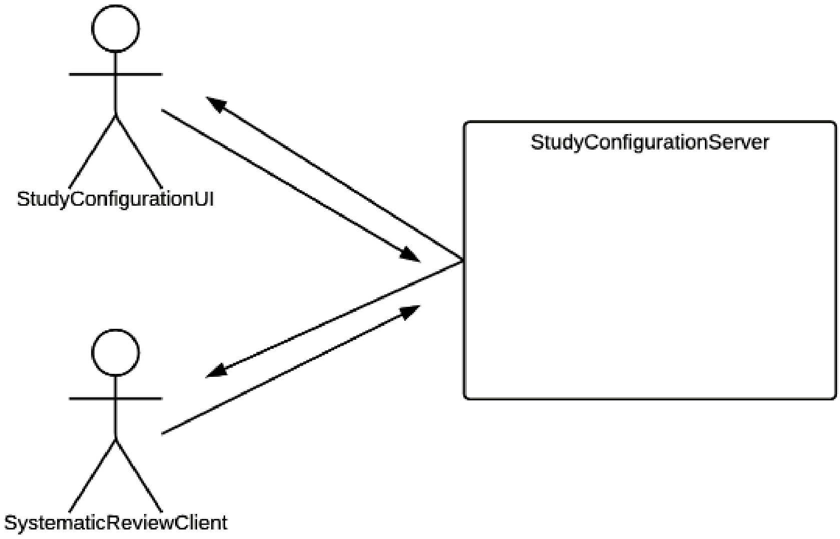

Figure 4.1: System Model for Glitter.

4.1 Use Case Models

4.1.1 Scenarios

Scenario name	**Conducting a review**
Participating Actor instances	Bob: Researcher Alice: Reviewer
Flow Of Events	1. Alice just had lunch with Bob, when he told Alice that he had assigned Alice to review a study about waves and how they can produce electricity. Bob wanted Alice to review and screen papers for the study by next week. 2. As soon as Alice turns on the program she immediately sees that she has been assigned for screening for the wave study. 3. Alice reads up on what the review wants her to do, this time she has to go through a list of 51 articles and look for relevant papers that match the topic of harnessing electricity from waves. The articles she find corresponding with the topic she marks off. 4. At 4 pm Alice has a appointment so she shuts down her pc and decides to finish reviewing the last 9 papers tomorrow.

Scenario name	**Retrieving papers for screening**
Participating Actor instances	Bob: Researcher, Alice: Researcher
Flow Of Events	1. Bob needs to retrieve papers for a seismic analysis article he got yesterday. 2. Bob is wondering if there are many people whom wrote on the topic before? He starts searching the system for papers with the topics ?seismic activity? and ?volcanic eruptions?. 3. Bob gets a long list of papers shown on the screen, he finds that some of the articles are written by Alice. 4. Bob downloads the articles Alice has published after 2011 and are under 6 pages, so he can read them on the way home.

Scenario name	**Planning a study**
Participating Actor instances	Bob: Researcher/Study manager
Flow Of Events	1. Bob wants to create a new study of a subject, he decides that waves producing electricity could be an intriguing subject. 2. Bob creates a new study and enters the research question: ?how can waves produce electricity?, together with some inclusion and exclusion criteria that older papers than 2005 are to old for this study to have any relevance. 3. Bob must choose a team to conduct the study. Bob chooses his own team with him and Alice. 4. After Bob chose a team he now has to set up the workload and phases of the study, Bob Chooses to have 3 phases and he wants to distribute the work equally among him and Alice. 5. Bob is now done with his study planning and can submit his study and begin working.

Scenario name	**Exporting the data**
Participating Actor instances	Bob: Researcher
Flow Of Events	1. Bob and his research team has just finished conducting the review and now they want to report it. 2. He decides that they want bubble graphs, bar charts and a couple of tables to illustrate their findings. 3. Bob receives the illustrations as figures which he can now use for their report.

4.1.2 Use Case Model

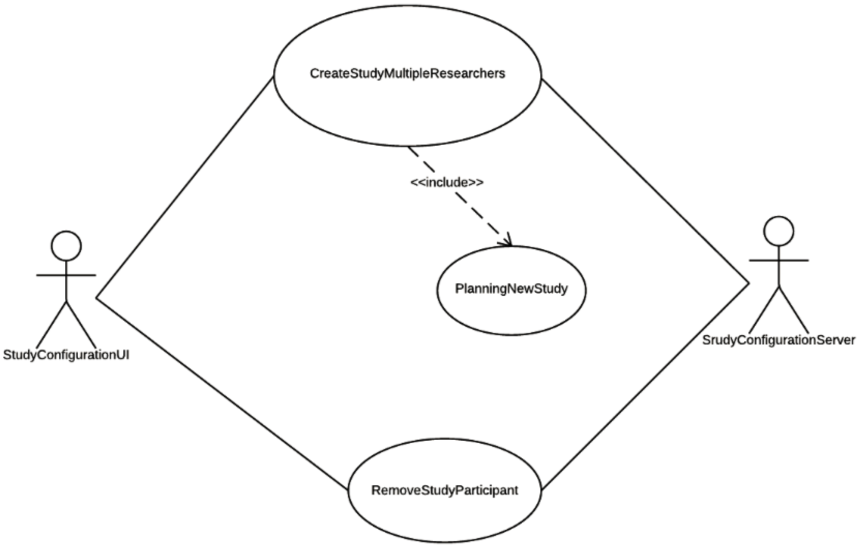

Figure 4.2: Use Case model for Glitter

4.1.3 Use Cases

Use Case Name	**Marking papers off for a study**
Participating Actors	Initiated by reviewer, Communicates via StudyConfigurationUI, StudyConfigurationServer
Flow Of Events	1. The reviewer presses the button "See my tasks" in the StudyConfigurationUI. 2. StudyConfigurationUI shows an overview of tasks that are assigned to the reviewer. 3. The reviewer double-clicks on a specific task to see the details of that task. 4. StudyConfigurationUI opens the task, and shows the study plan for this specific task. 5. The reviewer selects to start the review process. 6. StudyConfigurationUI retrieves the papers corresponding to the specified criteria of the study from the StudyConfigurationServer and lists the papers on the screen. 7. The reviewer reads one paper at the time and marks the ones that are relevant to the study off. 8. When done, the reviewer presses "submit" and the papers marked off are stored on the task. 9. StudyConfigurationUI prompts if reviewer want to change the task to "Done" 10. Reviewer confirms that the task has been done.
Entry Condition	- A task asking to find papers with a given criteria is assigned to the reviewer. - The Researcher is connected to the internet/System Configuration Server.
Exit Condition	- The reviewer has successfully submitted a list of papers and changed the status of the task to "done". - The reviewer has aborted the reviewing process.
Quality Requirements	No more than 5 seconds should pass before the researcher receives a response after pressing a button. The user should be able to cancel the process of reviewing at any time and still be able to save the progress made.

Use Case Name	**CreateStudyMultipleResearchers**
Participating Actors	Initiated by Researcher, Communicates via StudyConfigurationUI, StudyConfigurationServer
Flow Of Events	1. The researcher presses the “Create new study” button in the StudyConfigurationUI. 2. The StudyConfigurationUIresponds by asking the researcher for a name and a topic for the study. 3. The researcher enters a name and topic and presses “next”. 4. The StudyConfigurationUI responds by asking if any other researchers are to be assigned to the study, prompting the researcher with a field for a name, as well as a button saying add. 5. The researcher enter the name of a fellow researcher, and presses the “add” button. 6. The StudyConfigurationUI verifies the name of the user with the StudyConfigurationServer. 7. If the name of the new researcher is present in the StudyConfigurationServer, the researcher is added to the new study. 8. Otherwise, the StudyConfigurationUI prompts the researcher that the name of the new researcher could not be verified. 9. When the researcher has finished adding all of his coworkers, he presses the “Finish” button, and the study is assigned to the study in the StudyConfigurationServer.
Entry Condition	- The Researcher wants to create a new study. - The Researcher is connected to the internet/System Configuration Server.
Exit Condition	- The researcher has successfully assigned a team to the study. - The researcher has aborted the assignment of a team to the study.
Quality Requirements	No more than 10 seconds should pass before the researcher receives a response after pressing “next” “add” or “finish”. The user should be able to cancel the creation of a study at any time.

Use Case Name	**PlanningNewStudy**
Participating Actors	Initiated by Researcher, Communicates via StudyConfigurationUI, StudyConfigurationServer
Flow Of Events	1. The researcher enters the criteria he wishes to impose on the study if any into the StudyConfigurationUI and presses “add”. 2. The StudyConfigurationUI responds with a messages saying the criteria was added and clears the criteria field so a new criteria can be added. 3. The researcher selects the number of phases required for the study, and how the workload should be distributed through each phase, in the StudyConfigurationUI. 4. When the researcher is finished selecting phases and workload distribution and has added the desired criteria to his study, he presses “Create study” and the study information is sent via the StudyConfigurationUI to the StudyConfigurationServer. 5. The StudyConfigurationUI responds with a message saying the study was created. 6. The Study is created on the StudyConfigurationServer.
Entry Condition	- The Researcher wants to create a new study. - The Researcher has completed the “CreateStudyMultipleResearchers” use case. - The Researcher is connected to the internet/System Configuration Server.
Exit Condition	- The researcher has successfully planned a study. - The researcher has aborted planning a study.
Quality Requirements	No more than 10 seconds should pass before the researcher receives a response after pressing a button. The user should be able to cancel the creation of a study at any time.

Use Case Name	**RemoveStudyParticipant**
Participating Actors	Initiated by Researcher with Admin flag, Comminucates via StudyConfigurationUI, StudyConfigurationServer
Flow Of Events	1. The researcher presses the “Manage studies” button in the StudyConfigurationUI 2. The StudyConfigurationUI responds by presenting the researcher with a list of studies for which he is one of the registered admins. 3. The researcher selects the study for which he wants to edit participants, then presses the “Edit” button. 4. The StudyConfigurationUI responds by presenting the reasercher with a list of all participants, as well as other options for the study. 5. The researcher select the participant he wants to remove from the study, and then presses the “Remove” button. 6. When the researcher is finished editing the study, he presses the “Apply” button, followed by the “Exit” button and is returned to the main menu.
Entry Condition	- The researcher wants to remove a participant from a study. - The researcher is connected to the internet/System Configuration Server.
Exit Condition	- The researcher successfully removed the participant from the study.
Quality Requirements	No more than 10 seconds should pass before the researcher receives a response after pressing “Edit”, “Remove”, “Apply” or “Exit”. The researcher should be able to revert any changes, up until the point of pressing “apply”

4.2 Domain Object Models

4.3 Dynamic Models

4.3.1 Use Case Sequence Diagrams

4.3.2 State Diagrams

4.4 User Interfaces

4.4.1 Graphical User Interface for Operator

Part II

Software Design Document

5 Current Software Architecture

6 Proposed Software Architecture

6.1 Design Goals

6.2 Overview

6.3 Subsystem Decomposition

6.4 Persistent Data Management

6.5 Access Control and Security

6.6 Global Software Control

6.7 Boundary Conditions

Part III

Object Design Document

7 | Object Design and Patterns

7.1 Object Design

7.2 Interfaces

7.3 Design Patterns

Part IV

Test Document

8 | Test Plan and Results

8.1 Overall Test Approach

8.2 Component / Unit Testing

8.3 Integration Testing

8.4 System Testing