

Requirement Analysis Document - Yellow Project

William Diedrichsen Marstrand, Dennis Thinh Tan Nguyen,
Jacob Mullit Miniche, Thor Valentin Olesen

September 27, 2015

1 Introduction

1.1 Purpose of the System

Today, the amount of data has grown to a level where it is starting to challenge researchers who need to extract the best available research on a specific question. As a result, researchers apply systematic studies on big data sets where they exploit meta data to classify subsets with useful data. This requires smart data processing tools that can use meta data to narrow down relevant research data. The purpose of such a systematic review is to sum up the best available research on a specific question. This can be achieved by combining the results of several studies. In this regard, the system in this project is comprised of two parts, client and server side. Our system scope is restricted to the server side and shall support the configuration of summarized research data relevant to a given research question.

1.2 Scope of the System

The server shall provide teams with tools to conduct secondary studies (SMS or SLR). It should support activities of planning and conducting a review distributed in a research team. The server shall make sure that all data is stored for use in setting up a study configuration requested by the client. The system should support .bibtex files as a minimum. Security matters (e.g. user authentication) are not taken into primary account due to the scope of the project. In order for the system to fulfill the previously described purpose, it has to support the tasks described in the "Proposed System" section ???. Among these, the system shall support management of distributed research systems to work on a study. The reviewers should be able to export data sets and filter them with inclusion and exclusion criteria. Finally, the system should allow specific sets of data to be reviewed and screened by specific members of a research team.

1.3 Objectives and Success Criteria of the Project

The system should be easy to deploy and install. It should include an installation and user manual used to describe how to configure and prepare research papers for screening. It should be easy and quick to distribute relevant data and the overview should outcompete the one achieved in other third-party programs such as e.g. Excel. The system should define rules for which data goes to whom to achieve a successful screening of paper and efficient data extraction. The yellow system has to provide a user interface from which the blue team can extract data based on user roles and rules. The system has succeeded if users in the blue team can query the yellow system for relevant studies and tasks based on a given study configuration. Specifically, the yellow system should collect research and aggregate stacks of research material based on a research question. The blue system should then efficiently be able to extract a subset of primary studies provided from the screening of the search hits in the yellow system. The configured data may then be used by reviewers in the blue system (visualize, sort, export and categorize). Finally, the system should be able to replicate an existing study.

1.4 Definitions, Acronyms, and Abbreviations

- Systematic Studies: methodology used to sum up the best available research on a specific research question or topic.
- Yellow system: server side also referred as "server"
- Blue system: client side also referred as "client"

1.5 References

Tell, Paolo, and Steven Jeuris. Autosys: Supporting Distributed Teams Performing Systematic Studies. 1st ed. Copenhagen: ITU, 2015. Print.

1.6 Overview

The rest of the document will describe the current systems available and the proposed system in our project along with the requirements collected from users, customers and stakeholders. The system requirements are used to generate system models such as potential scenarios and use cases.

2 Current System

The current is a Excel based application which facilitates searches on academic topic. The system does not contain the articles, but data regarding the articles content. The system support multiple users to contribute data and makes it possible to identify if another user is contributing data on the same article. The Data contributed is mostly keywords, which makes it easier to search and compare articles. ===== HEAD To search for articles, a user must submit keywords, which the articles will be ranked after. Articles are also ranked after the amount of keywords they have in common with other articles.

The current system can work, but become increasingly hard to manage. Therefor a new system which solves the negatives while keeping all of the functionality the system already support is recommended. This can be done with a separation of the data and the user interface, by creating a dedicated database and a dedicated user interface. This will ensure that new functionality will not be limited by the database structure.

The negatives and pros of the current system are described below: ===== To search for articles, a user must submit keywords, which the articles will be ranked after. Articles are also ranked after the amount of keywords they have in common with other articles.

The current system can work, but become increasingly hard to manage. Therefor a new system which solves the negatives while keeping all of the functionality the system already support is recommended. This can be done with a separation of the data and the user interface. This will ensure that new functionality will not be limited by the database structure.

The negatives and pros of the current system are described below: ===== origin/master

2.1 Pros of current system

The systems pros are as following:

1. Excel licenses are cheap and is widely used by customer base
2. Its simple and easy to augment and implement changes.
3. If the excel files are Read Only, then data the is safe from hostile users.
4. It's easy to implement version control on .csv files.

2.2 Negatives of current system

And the systems Negatives are as following:

1. The data structures can be hard to manage, especially as the system grows in scope and content.
2. The system does not support multiple users editing the same files at the same time.
3. The system interface leaves a lot to be desired.
4. New features becomes hard to implement as the systems grow.

3 Proposed System

3.1 Overview

This section will describe the requirements of the system. In other words, it will clarify the purpose of the system with functional and non-functional requirements. The functional requirements will describe the services that the system should provide and how the system will behave. The non-functional requirements will describe the constraints of these services and are used to measure the quality of these.

3.2 Functional Requirements

The system must support the following functional requirements:

- The user must be able to define the phases of a study.
- The user must be able to create and manage classification criteria for different studies.
- The user must be able to create and manage inclusion and exclusion criteria.
- The user must be able to configure future phases in a study.
- The user must be able to store information about delivered tasks.
- The user must be able to store data about Users and Teams.
- The user must be able to filter research papers based on demographic information specified by the user.
- The user must be able to screen imported research papers. The result should be an export of papers consisting of primary studies (set of relevant papers).
- The user should be able to export studies as plain data sets in different formats, e.g. as comma-separated-values(cvs) format.
- The system should only accept tasks sent from the same user who received them in the first place.
- The system should support a role engine which identifies user rights. Possible roles to be supported could be a "viewer" and "validator" role.
- The system should be able to handle multiple client requests concurrently.
- The system should be able to store quality ratings of research papers based on coverage according to specific research criteria.
- The system should be able to extract data samples from specific studies for further validation by a validator role.
- The system should be able to generate a research protocol that describes the rationale, objectives, design and methodology of the data and configuration of a study.
- The system should restart automatically upon system failure.
- The system must be complemented with an installation manual and a user manual.

3.3 Nonfunctional Requirements

The system must support the following nonfunctional requirements:

3.3.1 Usability

- The Users must be able to configure a study through the use of at most 3 application windows.
- Users with no background in IT should be able to understand the entire set of tools, which the system provides in approximately 30 minutes.
- The user manual must provide knowledge about the interfaces provided by the system.

3.3.2 Reliability

- System restart upon system failure should be done within 30 seconds.
- The system should strive to always be available.
- The latest version of the system should always be available through a back up

3.3.3 Performance

- The system should identify the role of a user and load the appropriate studies within 15 seconds
- A user request for specific data should be acknowledged no more than 10 seconds after it is received
- The time before a response is send should be no more than 30 seconds after the request is received
- The system should support at least 200 users

3.3.4 Supportability

- The server is maintained by the supplier of the system
- The system must be fully functional independent of the other subpart of the complete system (the blue component)

3.3.5 Implementation

- The system should run flawlessly on any Windows operating system newer than Windows XP

3.3.6 Interface

- The system user interface for study configuration must at any time be replaceable by a newer version

3.3.7 Packaging

- The set up and installation of the server is done by the provider of the system independent of the client
- An employee with no prior knowledge about the system must be able to install and set up the server within a period of 15 min

3.3.8 Legal

- The system should be publishable as Open Source Software in accordance to the GNU General Public License v2.0
- The system should store data on a MySQL server provided by the IT University of Copenhagen following the rules issued by the institution

4 Scenarios

This section contains various scenarios regarding operations between the user and the system.

<i>Scenario name</i>	<u>bobStartsNewStudyConfiguration</u>
<i>Participating actor instances</i>	<u>bob:Researcher</u> <u>System:AutoSys System</u> <u>Configuration UI:StudyConfigurationUI</u>
<i>Flow of events</i>	<ol style="list-style-type: none">1. Bob the Researcher has to start a new research and opens the client from StudyConfigurationUI.2. Bob logs into the client and navigates to the "Study Configuration" page.3. Bob specify two reviewer and one validator and defines a research question based on some inclusion- and exclusion criteria to specify what papers should be returned.4. Bob confirms his study configuration and sends the request to the server by pressing "ok".5. The AutoSys System returns an overview of the study configuration to the client.

Table 1: Scenario when a user creates a new study configuration

<i>Scenario name</i>	<u>ClientFilteringOperation</u>
<i>Participating actor instances</i>	<u>server:Server</u> <u>client:Study Review UI</u>
<i>Flow of events</i>	<ol style="list-style-type: none">1. Server is receiving a request from Study Review UI: Filtering keywords Design pattern, 2005, A gang of four2. Server validates clients authentication. User credential is accepted.3. Server measures all studies based on the given keywords.4. Server finds 20 studies and a list is formed. Each study element in the list contains data about it.5. Server replies to the clients request by sending the article list of found articles.6. Server returns an overview of the Study Configuration to the Study Review UI,

Table 2: Scenario when a user sends a request with given filtering keywords trough the Study Review UI.

<i>Scenario name</i>	<u>ClientGetsToManyRelevantPapers</u>
<i>Participating actor instances</i>	<u>server:Server</u> <u>client:Study Review UI</u>
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. Server is receiving a request from Study Review UI: Search keywords 2001,2002,2003,2003,2004,2005,2005,2006,,2007 2. Server validates User authentication. User credential is accepted 3. Server measures all articles based on the keywords. 4. The list containing papers exceeds 10.000 papers, and the exception ToManyHitsException was thrown. 5. A response is sent to the Study Review UI that there in fact was too many papers returned.

Table 3: Scenario when a user has requested to many papers during one request.

<i>Scenario name</i>	<u>ExcludingPapersAboutDesignPatterns</u>
<i>Participating actor instances</i>	<u>server:Server</u> <u>client:Study Review UI</u>
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. Server is receiving a request from Study Review UI: Search keywords Design pattern, 2005, A gang of four, ExcludingAllNonAssignedArticles. 2. Server validates User authentication. User credential is accepted. 3. Server measures all articles based on the keywords. 4. 10 relevant articles was found but 5 was excluded because of the exclusion criteria. 5. A list with the remaining 5 articles is returned from Server to the Study Review UI

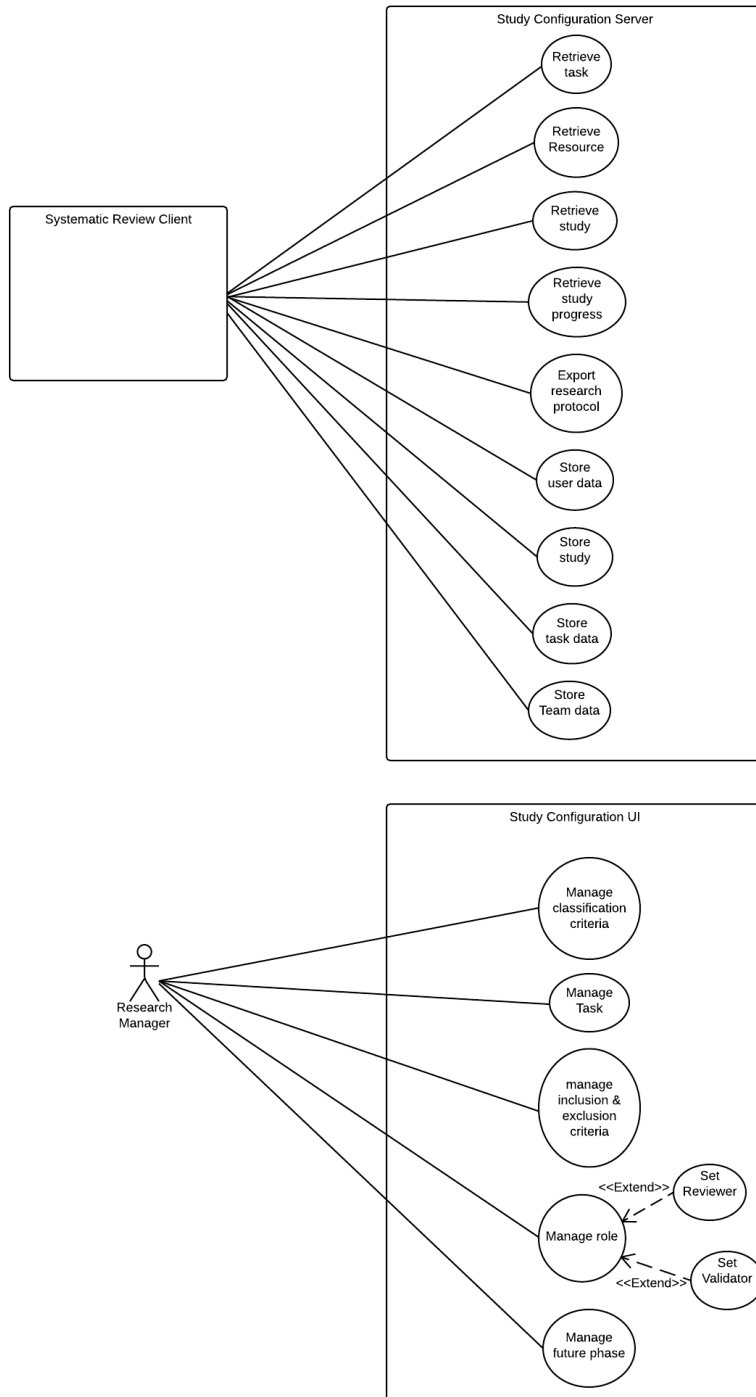
Table 4: Scenario when a user wants to exclude some papers.

<i>Scenario name</i>	<u>ClientRequestWithInvalidUser</u>
<i>Participating actor instances</i>	<u>server:Server</u> <u>client:Study Review UI</u>
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. Server is receiving a request from Study Review UI on task retrieval for a User named "bob" which is an invalid User. 2. Server validates User authentication. 3. The given User is not valid because it does not exist in the database. 4. A response is sent to the Study Review UI detailing why bob does not have access to the AutoSys System.

Table 5: Scenario when a invalid user is trying to get access to the server.

5 Use Case

5.1 Use Case Diagram



5.2 Use Case Description

<i>Use case name</i>	ExportProtocol
<i>Participating actors</i>	Initiated by Client Communicates with Server
<i>Flow of events</i>	<ol style="list-style-type: none">1. The Client sends a request for a specific study.2. The Server validates the request and sends back the relevant study data.3. The Client requests a research protocol from the study.4. The Server receives the request and makes a validation of the Client's user account.5. The Server exports the requested research protocol to the Client.6. The Client receives the research protocol.
<i>Entry condition</i>	<ul style="list-style-type: none">• The Client is logged in and connected to the internet.
<i>Exit conditions</i>	<ul style="list-style-type: none">• The Client has received the selected data from the Server
<i>Quality requirements</i>	<ul style="list-style-type: none">• The Client's request is received by the Server within 10 seconds.• The Client starts receiving the study report no later than 40 seconds after it has been requested.

Table 6: Use case: ExportProtocol

<i>Use case name</i>	RetrieveStudyInformation
<i>Participating actors</i>	Initiated by Client Communicates with Server
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. The Client activates the Manage Study module. 2. The Client uploads task data about a completed task. 3. The Client specifies the study information to retrieve. 4. The Client confirms the retrieval of details regarding the Study. 5. The message is sent to the server where the senders credentials are validated in the database. 6. The Server sends back the study information 7. The Client receives the requested study information.
<i>Entry condition</i>	<ul style="list-style-type: none"> • The Client is logged in and connected to the internet.
<i>Exit conditions</i>	<ul style="list-style-type: none"> • The Client has received the study information.
<i>Quality requirements</i>	<ul style="list-style-type: none"> • The Client should receive the study information maximum 30 seconds after requesting them.

Table 7: Use case: RetrieveStudyInformation

<i>Use case name</i>	StoreTaskData
<i>Participating actors</i>	Initiated by Client Communicates with Server
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. The Client selects a specific study. 2. The Client chooses to upload a completed task. 3. The Server receives the request and makes a validation of the user account. The validation goes through because the user uploading the completed task is the same as the one who received it in the first place. 4. The Server stores the data from the completed task in the database. Then it sends back a respond to the Client letting it know, that the request was received. 5. The Client receives the respond from the Server.
<i>Entry condition</i>	<ul style="list-style-type: none"> • The Client is logged in.
<i>Exit conditions</i>	<ul style="list-style-type: none"> • The Client has received a response from the Server indicating that the completed task was received and stored in the database.

Table 8: Use case: StoreTaskData

<i>Use case name</i>	ManageStudy
<i>Participating actors</i>	Initiated by Client Communicates with Server
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. The Client navigates to the Server page. 2. The Client invokes the ManageRole use case to define Role for each User that are appart of the StudyConfiguration. 3. The Client invokes the ManageClassificationCriteria use case to define classification criterias for the StudyConfiguration. 4. The Client invokes the ManageInclusionAndExclusionCriteria use case to define what to exclude or include to the StudyConfiguration. 5. The Client invokes the ManageTask use case to specify which Task a User should have. 6. The Client reviews all changes made and submits the StudyConfiguration to the Server. 7. The Server invokes the ValidateUser use case to validate access rights for the User of the Request 8. The Server accepts the User and the Request proceeds. 9. The Server invoke the StoreUserData use case to update the Role for each User which was specified in the StudyConfigurationUI. The Server will also invoke the StoreStudyData, StoreTaskData use cases to save the study and the tasks for each User. 10. The Server returns an overview of the StudyConfiguration to the client.
<i>Entry condition</i>	<ul style="list-style-type: none"> • The Client wants to create a new study.
<i>Exit conditions</i>	<ul style="list-style-type: none"> • The Client has received a StudyConfiguration overview
<i>Quality requirements</i>	<ul style="list-style-type: none"> • The Client's study is created within 10 sec.

Table 9: Use case when a Client wants to create a new study through the StudyConfigurationUI

6 Glossary

- Systematic Review Client: Is the client part of the system which broadly provides the User Interface for requesting data about papers, reviewing papers, and validating papers.
- Study Configuration Server: Is the server (including the database) where all data is stored, and user requests regarding papers are handled.
- Study Configuration UI: Is the User Interface supporting only work tasks regarding the configuration of studies e.g. study tasks, study phases etc.
- Research Protocol: the rules on how a given study is configured (work flow).
- Task: a subtask in a given study, e.g. a Validator may have to analyze similarities between the work of two different Reviewers on the same set of data.
- Study: the whole work process from initiating a research to narrowing down relevant research evidence. This implies different phases (subtasks) where data is continuously synthesized and approved by users with different roles. The end result is a final data set of primary studies used to clarify the research question.
- Request: all operations that invoke server functionalities.