Requirement Analysis Document - Yellow Project

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

November 17, 2015

Contents

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 be able to import information from a bibtex file to a database. The reason is we want to be able to populate our database with existing data. 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 succesful 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". The main responsibility of the yellow system are to store, send, validate and manipulate data from the database and also configurate a study.
- Blue system: client side also referred as "client" The main responsibility of the blue system are to visualize data from the database according to the user's demands and manage teams.
- Study: the whole work process from initiating a research to narrowing down relevant research evidence. A study consists of different phases where data is continuously synthesized and approved by users with different roles. The end result is a final set of primary studies used to clarify the research question.
- Primary and Secondary study: secondary research is defined as an analysis and interpretation of primary research. The method of writing secondary research is to collect primary research that is relevant to a given topic and interpret what the primary research found.
- Phase (stage): is a given set of review tasks. Each phase is dependent on each other sequentially and is completed in a fixed order. Each phase details how task requests are handled and handed out. Optionally, a user role can proceed to the next phase even though other user roles have not finished the same phase yet (e.g. two concurrent review phases).
- Task: an assignment in a given phase in a study. A task is defined by a unique id, a set of visible data fields (unmodifiable), a set of requested data fields (modifiable) and a type. A taks type can either be request to fill out data field(s) or a request to handle conflicting data field(s). By way of example, a phase could involve review tasks assigned for two reviewers. A validator could then analyze any inconsistencies between the work of both reviewers in a second phase.
- Data Field: a data type that determines the type of a criteria used in a task to fill out data or solve conflicts. Defined as either a String, Boolean, Enumeration (select one out of list), Flag (select multiple out of list) or Resource (pdf).
- Research Protocol: the rules on how a given study is configured (work flow).
- Study Configuration: the configuration of a given study. A study configuration involves defining the research question, phases, assigning a team for the study, choosing data fields and assigning roles for all team members.
- Inclusion/Exclusion Criteria: criteria that is evaluated throughout the whole lifetime of a given study. By way of example, a criteria could be whether the data is from later than 2005. The criteria is used along the way to synthesize the data. As opposed to the classification criteria that is only used in the end of the study.
- Classification Criteria: classification criteria is used to group data fields in a given study. By way of example, one could classify two groups for the same set of primary studies. The classification criteria determines what one is looking for in the data. It is defined in the study configuration and is used in the end of a study upon data extraction.
- User: a person who is either a manager or a researcher. A manager can create a team while also being a researcher. A researcher can only have one role in a specific phase of a study.

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 system is an Excel based application that contains data about contents of articles and facilitates searches on academic topics. It is possible for multiple users to contribute data and one can identify if another user is contributing data on the same article. The contributed data is mostly keywords, which makes it easier to search and compare articles. 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 is increasingly difficult to manage. Thus, a new system which solves the negatives while keeping all of the functionality that the system already supports 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:

2.1 Pros of current system

The systems pros are as following:

1. Excel licenses are cheap and widely used by customer base
2. Simple and easy to augment and implement changes.
3. Data is safe from hostile users if excel files are read only.
4. 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 system grows.

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 create and manage teams and members.
- The user must be able to assign teams and members in the configuration of a study.
- 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 must be able to export studies as plain data sets in different formats, e.g. as comma-separated-values(cvs) format.
- The system must support a role engine which identifies user rights. Possible roles to be supported could be a "viewer" and "validator" role.
- The system must be able to handle multiple client requests concurrently.
- The system must be able to store quality ratings of research papers based on coverage according to specific research criteria.
- The system must be able to extract data samples from specific studies. These are used to clarify any inconsistencies between different reviews of the same study. The data sample is accepted or rejected given in percentage terms - and is part of a phase in the study.
- The system must be able to generate a research protocol that describes the rationale, objectives, design and methodology of the data and configuration of a study.

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. This includes how a study is configured, how to manage teams and members and operations that the system support in general.

3.3.2 Reliability

- System restart should be done within 30 seconds upon system failure.
- The latest stored data should always be available through a backup.

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 handled 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)
- The system study configuration user interface must be replaceable by a newer version.

3.3.5 Implementation

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

3.3.6 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

3.3.7 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.

Scenario name	bobStartsNewStudyConfiguration
Participating actor instances	bob:`Researcher` System:`AutoSys System` Configuration UI:`StudyConfigurationUI`
Flow of events	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

Scenario name	ClientFilteringOperation
Participating actor instances	server:`Server` client:`Study Review UI`
Flow of events	1. `Server` is receiving a request from `Study Review UI`: Filtering keywords Design pattern, 2005, A gang of four 2. `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`.

Scenario name	ClientGetsToManyRelevantPapers
Participating actor instances	server:`Server` client:`Study Review UI`
Flow of events	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 an error will be displayed on the users screen. The error will also instruct the user to narrow his search to be able to display his search results

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

Scenario name	ExcludingPapersAboutDesignPatterns
Participating actor instances	server:`Server` client:`Study Review UI`
Flow of events	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` searches for articles with the given Search Keyword, but are excluding articles which are not assigned to the user. 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`

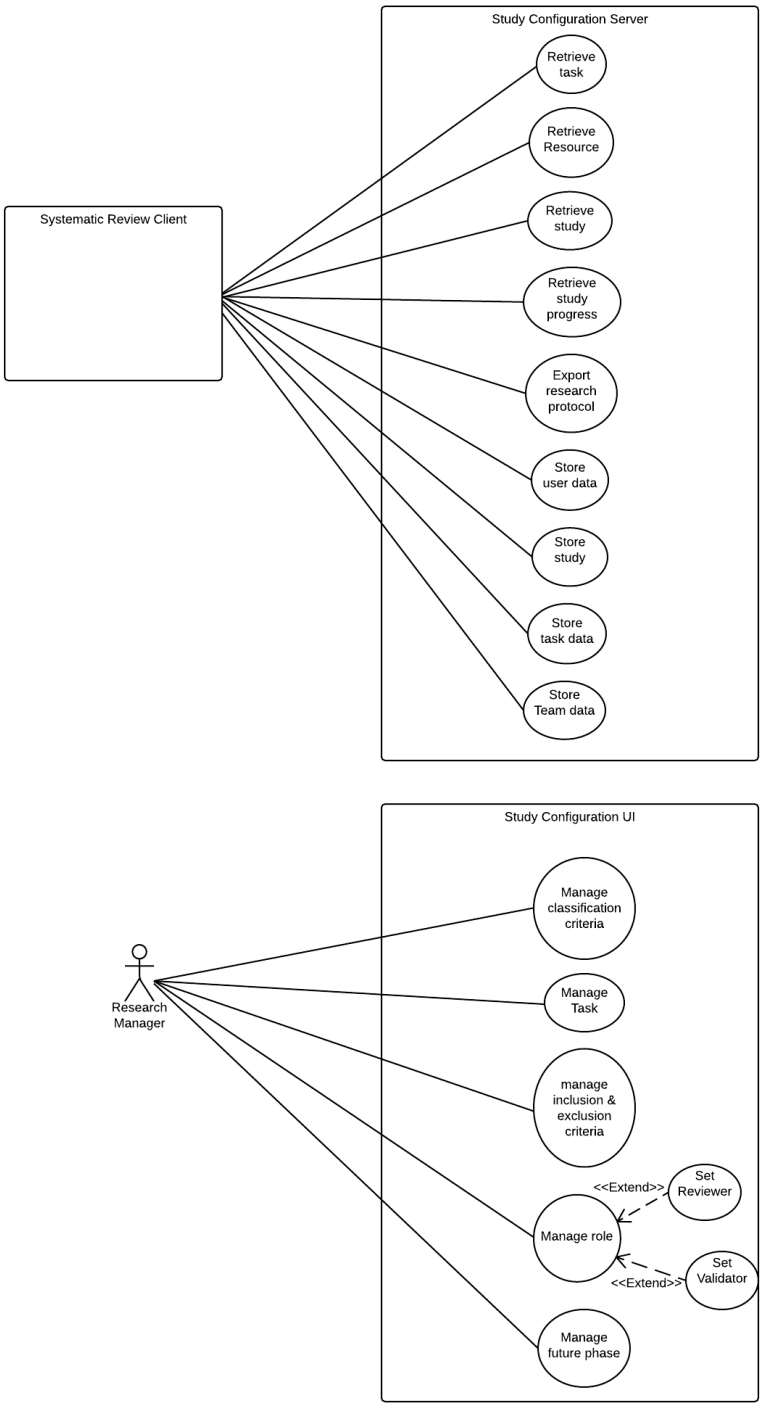
Table 4: Scenario when a user wants to exclude some papers.

Scenario name	ClientRequestWithInvalidUser
Participating actor instances	server:`Server` client:`Study Review UI`
Flow of events	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

Use case name	`ExportProtocol`
Participating actors	Initiated by `Client` Communicates with `Server`
Flow of events	1. The `Client` sends a request for a specific study. 2. The `Server` validates the request sends back data detailing the study. This include which protocol the Study use and which phases it has been through. 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.
Entry condition	• The `Client` is logged in and connected to the internet and click on the export protocol menu (provided by the Blue Team)
Exit conditions	• The `Client` has received the selected data from the `Server`
Quality requirements	• 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

Use case name	`RetrieveStudyInformation` using: `ManageStudy`
Participating actors	Initiated by `Client` Communicates with `Server`
Flow of events	1. The `Client` activates the Manage Study use case. 2. The `Client` specifies the study information to retrieve. 3. The `Client` confirms the retrieval of details regarding the Study. An error message is displayed if the provided data is insufficient 4. The message is sent to the server where the senders credentials are validated in the database. 5. The `Server` sends back the study information 6. The `Client` receives the requested study information.
Entry condition	• The `Client` is logged in and connected to the internet.
Exit conditions	• The `Client` has received the study information.
Quality requirements	• The `Client` should receive the study information maximum 30 seconds after requesting them.

Table 7: Use case: RetrieveStudyInformation

Use case name	`StoreTaskData`
Participating actors	Initiated by `Client` Communicates with `Server`
Flow of events	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`.
Entry condition	• The `Client` is logged in.
Exit conditions	• The `Client` has received a response from the `Server` indicating that the completed task was received and stored in the database.

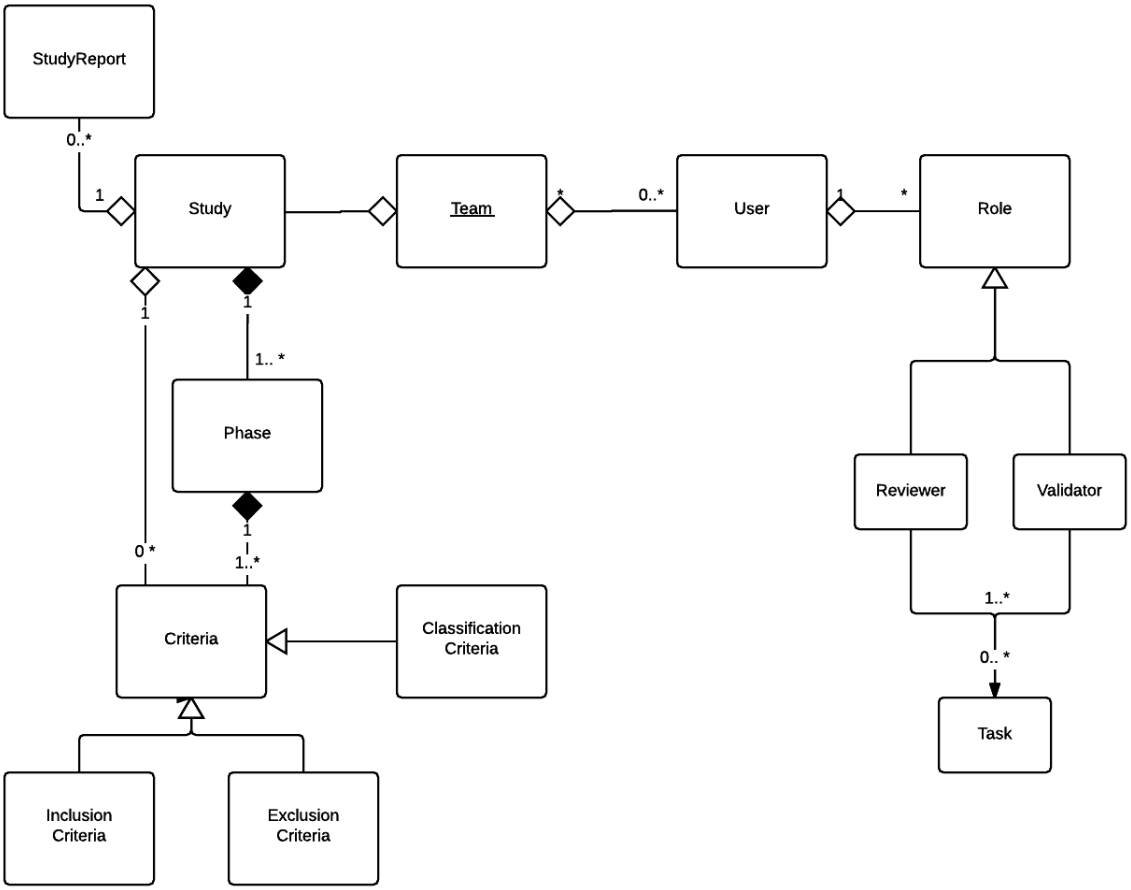
Table 8: Use case: StoreTaskData

Use case name	`ManageStudy` using: `ManageClassificationCriteria`
Participating actors	Initiated by `Client` Communicates with `Server`
Flow of events	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.
Entry condition	• The `Client` wants to create a new study.
Exit conditions	• The `Client` has received a `StudyConfiguration` overview
Quality requirements	• The `Client`'s study is created within 10 sec, after the server has received the request.

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

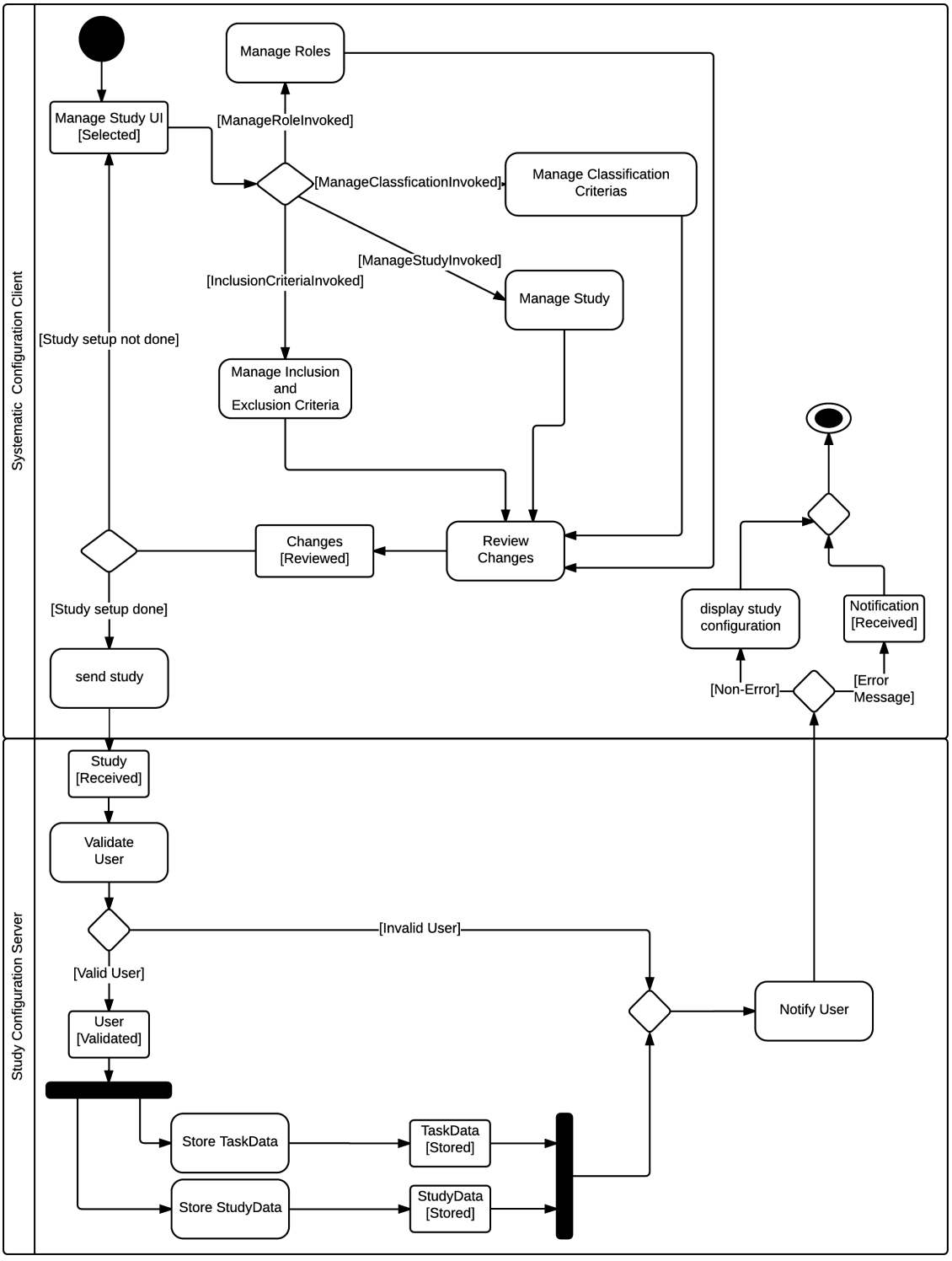
6 Object Model

6.1 Entity Object Model



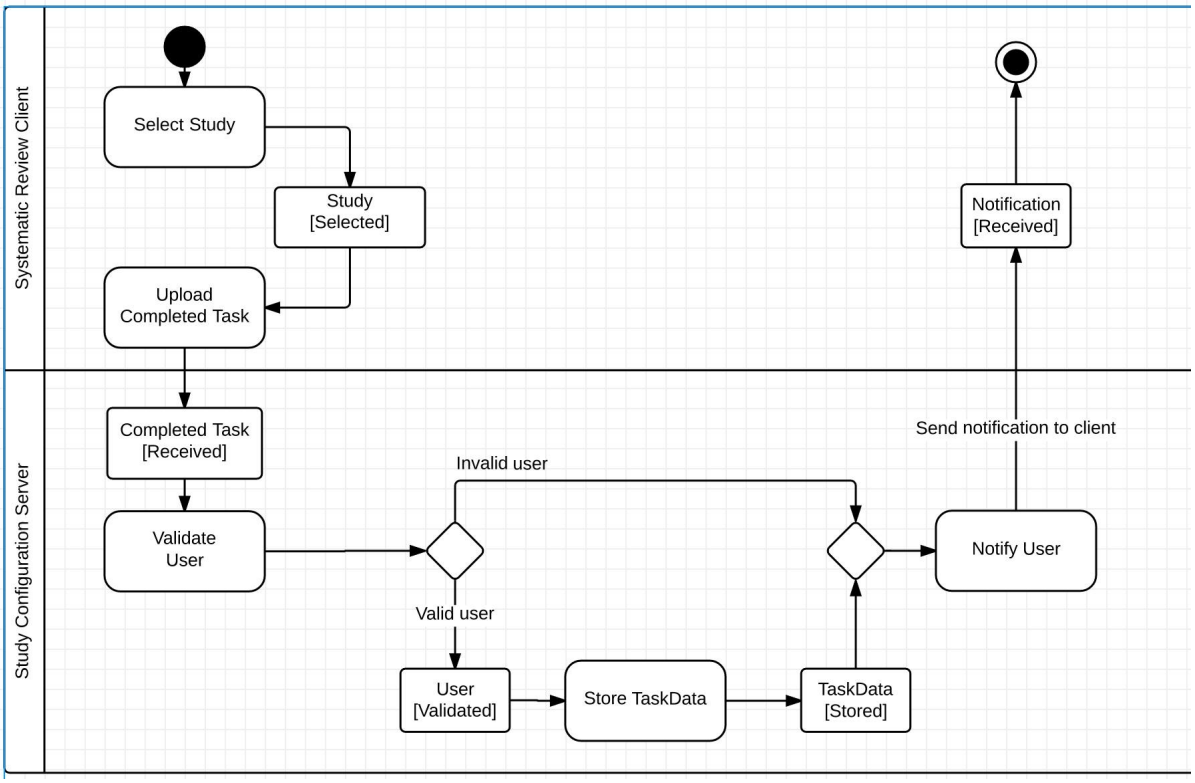
7 Dynamic Model

7.1 Activity Diagram of ManageStudy Use Case

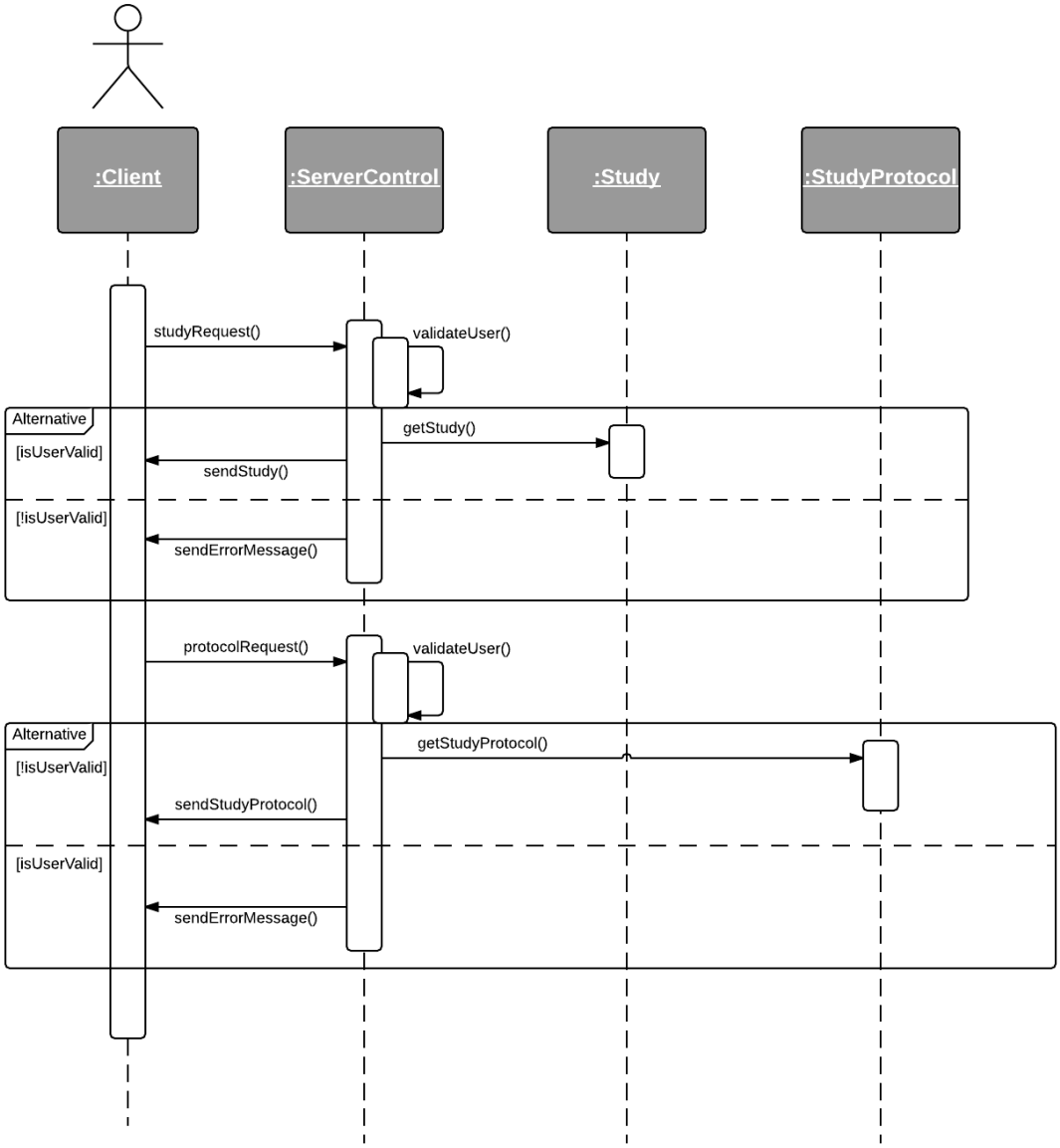


7.2 Activity Diagram of Retrieve Study Information Use Case

7.3 Activity Diagram of StoreTaskData Use Case



7.4 Sequence Diagram of ExportProtocol Use Case



8 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.
- Request: all operations that invoke server functionalities.