System Models

September 13, 2015

1 Scenarios

This section containts various scenarios regarding operations between `StudyConfigurationServer` and `StudyConfigurationUI`, with focus on the `StudyConfigurationServer`.

Scenario name	bobStartsNewStudyConfiguration
Participating actor instances	bob:`Researcher` server:`StudyConfigurationServer` client:`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. From the `StudyConfigurationServer`, the server extracts the information from the request and stores the information about the study configuration and the team roles. 6. The server 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:`StudyConfigurationServer` client:`StudyConfigurationUI`
Flow of events	1. Server is receiving a request: Filtering keywords Design pattern, 2005, A gang of four 2. Server validates clients authentication. User credential is accepted. 3. The server measures all studies based on the given keywords. 4. The server finds 20 studies and a list is formed. Each study element in the list contains data about it 5. The server replies to the clients request by sending the article list of found articles. 6. The server returns an overview of the study configuration to the client,

Table 2: Scenario when a user sends a request with given filtering keywords.

Scenario name	ClientRequestWithInvalidUser
Participating actor instances	bob:`InvalidUser` server:`StudyConfigurationServer` client:`StudyConfigurationUI`
Flow of events	1. Server is receiving a request on task retrieval for bob which is an invalid user. 2. Server validates clients authentication. 3. The given user from the client is not valid because it does not exist in database 4. A response is sent to the client detailing why bob does not have access to the server.

Table 3: Scenario when a invalid user is trying to get access to the server.

Scenario name	ClientGetsToManyRelevantPapers
Participating actor instances	server:`StudyConfigurationServer` client:`StudyConfigurationUI`
Flow of events	1. Server is receiving a request: Search keywords 2001,2002,2003,2003,2004,2005,2005,2006,,2007 2. Server validates clients authentication. User credential is accepted 3. The server measures all articles based on the keywords. 4. The list containing papears exceeds 10.000 papers, and the exception ToMany-HitsException was thrown. 5. A response is sent to the client that there was too many papers returned.

Table 4: Scenario when a user has requested to many papers during one request.

Scenario name	ExcludingPapersAboutDesignPatterns
Participating actor instances	server:`StudyConfigurationServer` client:`StudyConfigurationUI`
Flow of events	1. Server is receiving a request: Search keywords Design pattern, 2005, A gang of four, ExcludingAllNonAssignedArticles. 2. The Server validates clients authentication. User credential is accepted 3. The server measures all papers based on the keywords. 4. 10 relevant papers was found but 5 was excluded because of the exclusion criteria. 5. A list with the remaining 5 papers is returned to the client

Table 5: Scenario when a user wants to exclude some papers.