Requirements analysis

PULSU

Fall 2008
Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

Course

581260 Software Engineering Project (9 cr)

Project group

Heikki Manninen
Noora Joensuu
Sami Vuorivirta
Joel Kaasinen
Erno Liukkonen

Client

Peter von Etter
Roman Yangarber

Instructor

Peter von Etter

Homepage

http://www.cs.helsinki.fi/group/pulsu/

Version

Version	Date	Description
1.0	30.9.2008	Phase I requirements ready
2.0	19.11.2008	Phase 2 requirements ready

Table of contents

1. Introduction

This is the requirements document of the software engineering group PULSU that is taking part to software engineering course on fall 2008.

PULS is a natural language processing system, which is developed at Department of Computer Science. It is used for extract facts from unstructured texts, and store them in database. The system can be used on many scenarios. Currently it is mainly used on disease and business scenarios.

The current user interface of PULS is designed for a user who is familiar with the system and its functionality. It now has web-based user interface, which can be used to make different kinds of searches to the database. In disease scenario it is also possible to use map to help to locate disease areas. Software engineering group PULSU is going to enhance the user interface of PULS.

There are many areas where improvements for user interface are needed. User interface has to be easier to use for average users. The map must be embedded in the user interface better. There should be seen current situation of diseases for example a list of diseases and amounts of diseased. Snippets should still be on new user interface. Hierarchical browsing would be good, for example in business scenarios browsing could be done by industry and geographic and in medical scenarios by geographic and disease. On error conditions it would be good if error notifications would be stored to log files. User interface must be available also for other scenarios even though this project concentrates on medical scenario. Later it is possible to extend user interface to other scenarios and do other improvements.

2. Glossary

Administrator	Systems Administrator who has enhanced search functionality in System, which helps to do work better.
Expert	A user who is an expert of scenario, for example in disease scenario expert is doctor. He or she has privileges to modify cases when logs into system.
Average user	Anyone who uses the system. He or she does not necessarily have any experience of scenario.
Domain	Tells what particular domain this user interface is supporting. Examples of domains: Medical and financial
Scenario	Tells what particular scenario this user interface is supporting. Examples of scenarios: Diseases, investments
View	Individual view shows information from certain aspect. Examples of views: Overview, table, enhanced search
Map	Map which includes marks where different diseases have been found.
Charts	Charts which show information about diseases, for example which diseases have recently been most active in documents.
Document	Document which has been found from internet. Individual document may contain many cases.
Case (IE event)	Fact, which has been collected from document by the system. Doctors use word "case" because word "event" means the whole process of a disease for them. Only on medical domain.
Unified case	Group of cases which have similar attributes (Level 1).
Group	Group of unified cases which are related to same disease in same country on period of two weeks (Level 2).
Plain cases	Cases without any grouping (Level 0).
Attribute	Disease, country, time, total, status etc. Disease: Tells the name of disease in this case. Country: Tells the name of country in this case. Time: Tells when this case happened. Total: Tells how many people have got disease. Status: Tells the current situation of disease.
Search	Basic search which all users may use.
Enhanced search	A extended search which is not necessarily available for average users. All other users may use this search functionality to define better what they want to search.
Hierarchical browsing	User may for example search only documents that are from certain continent.
Help	Help information for all users.

3. User groups

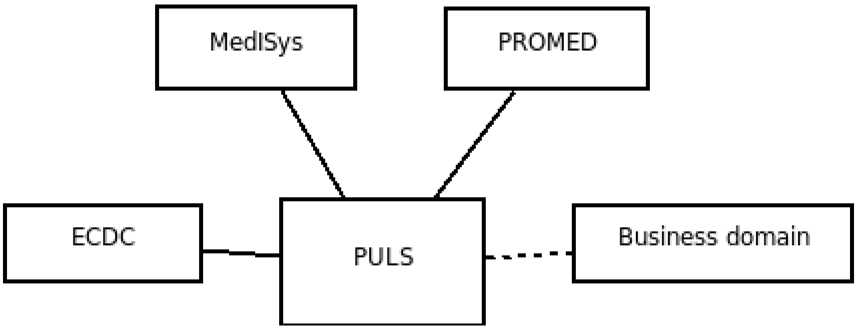
There are three user groups in PULS system:

Average user just uses the system. He may search and read information which he is interested in.

Expert may also search and read information. He can check that cases are correct and correct them if they are wrong. He may also send comments.

Administrator may add and remove users and he has more views to the system.

4. Stakeholders



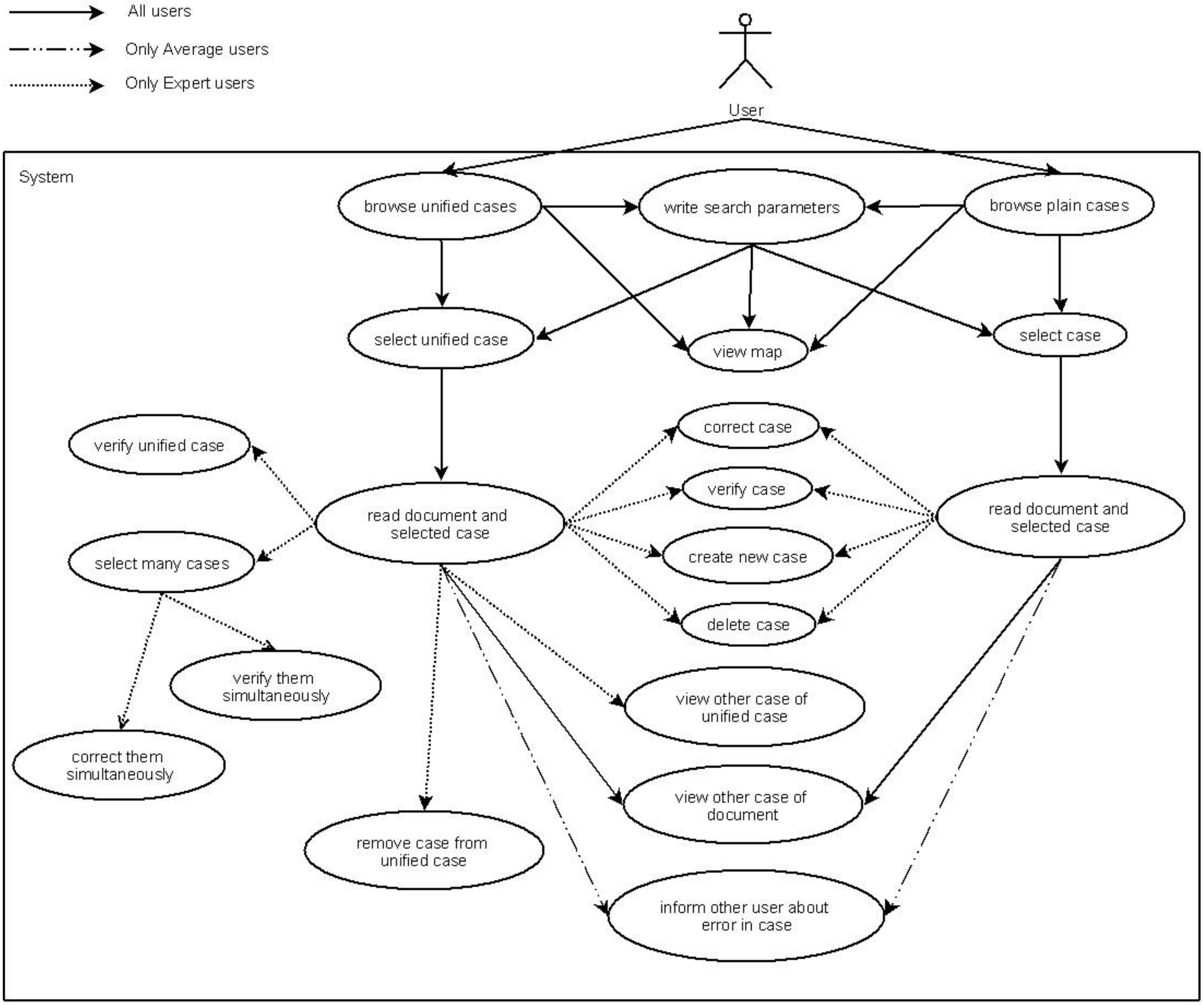
Medisys (Medical Information System) is main co-operation partner. Choose articles for Puls.

PROMED choose articles for Puls.

ECDC is main customer. Uses Puls program to get cases from articles.

Business domain is one of the systems main domains. It is alternative domain, which is planned to be used in the future.

5. Use cases


All users
Only Average users
Only Expert users
User
System
browse unified cases
write search parameters
browse plain cases
select unified case
view map
select case
verify unified case
read document and selected case
correct case
verify case
create new case
delete case
read document and selected case
select many cases
verify them simultaneously
correct them simultaneously
view other case of unified case
view other case of document
remove case from unified case
inform other user about error in case

See Functional requirements in section 7.

5.1 Phase I

5.1.1

User browses **unified cases**.

Functional requirements: **T1, T2**

Preconditions	User wants to browse unified cases.
Basic flow	1. User goes to Unified case view and begins to browse unified cases.
Exceptions	1. Some kind of connection problem occurs between user and the database. 2. There is problem with the database.
Postconditions	1. User got information which he or she requested.

5.1.2

User browses unified cases, selects one case and **reads the document** of the case.

Functional requirements: **T1, T2, T3**

Preconditions	User wants to read documents.
Basic flow	1. User goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case and continues to read the document of the case.
Exceptions	1. Some kind of connection problem occurs between user and the database. 2. There is problem with the database.
Postconditions	1. User got information which he or she requested.

5.1.3

Expert user browses unified cases, selects one case and checks its attributes and **verify case** if they are correct.

Functional requirements: **T1, T2, T3, T5, T6**

Preconditions	Expert wants to verify cases.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case and checks its attributes. 3. He or she verify the case. 4. He or she send a message of verification to the database. 5. The database receives it and marks the case to be verified.
Exceptions	1. Some kind of connection problem occurs between Expert and the database. 2. There is problem with the database.
Postconditions	1. The case has been verified.

5.1.4

Expert user browses unified cases, selects one case and **checks its attributes and corrects them** if they are incorrect.

Functional requirements: **T1, T2, T3, T5, T6**

Preconditions	Expert finds error on the case and wants to correct it.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case and checks its attributes. 3. He or she corrects attributes and sends them to the database. 4. The database receives new values of attributes and corrects the case.
Exceptions	1. Some kind of connection problem occurs between Expert and the database. 2. There is problem with the database. 3. The correction is invalid.
Postconditions	1. The case has been corrected.

5.1.5

Expert user browses unified cases, selects one case, checks its attributes and **deletes case**.

Functional requirements: **T1, T2, T3, T5, T8**

Preconditions	Expert finds an incorrect case and wants to delete it.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case, checks its attributes and notices that it shouldn't be a case at all. 3. He or she sends a request of deletion to the database. 4. The database receives Expert's request and deletes the case.
Exceptions	1. Some kind of connection problem occurs between Expert and the database. 2. There is problem with the database.
Postconditions	1. The case has been deleted.

5.1.6

Expert user browses unified cases, selects one case and reads the document and **creates a new case**.

Functional requirements: **T1, T2, T3, T5, T7**

Preconditions	Expert finds a new case from the document and wants to create it.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case, reads its document and finds one new case from it. 3. He or she writes attributes of the new case and sends a request to the database. 4. The database receives Expert's request and creates the case.
Exceptions	1. Some kind of connection problem occurs between Expert and the database. 2. There is problem with the database. 3. Attributes of the new case are invalid.
Postconditions	1. The case has been created.

5.1.7

User browses unified cases, selects one case and checks it and **selects other case of the unified case**.

Functional requirements: **T1, T2, T3, T4, T5**

Preconditions	User wants to check all cases of the unified case.
Basic flow	1. User goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case and checks its attributes. 3. He or she selects other case of unified case and checks it too.
Exceptions	1. Some kind of connection problem occurs between user and the database. 2. There is problem with the database.

Postconditions	1. User got information which he or she requested.

5.1.8

User browses newest unified cases, selects one case and checks it and **selects other case of the document**.

Functional requirements: **T1, T2, T3, T5**

Preconditions	User wants to check all cases of the document.
Basic flow	1. User goes to Unified case view and begins to browse newest cases. 2. He or she finds an interesting case and checks its attributes. 3. He or she selects other case of document and checks it too.
Exceptions	1. Some kind of connection problem occurs between user and the database. 2. There is problem with the database.
Postconditions	1. User got information which he or she requested.

5.2 Phase II

These use cases should be prepared for Phase I.

5.2.1

User browses unified cases and writes **search parameters**.

Functional requirements: **T1, T2, U1**

Preconditions	User wants to browse unified cases and find cases, for example, from specific disease.
Basic flow	1. User goes to Unified case view and begins to browse unified cases. 2. He or she writes search parameters and sends them to the database
Exceptions	1. Some kind of connection problem occurs between user and database. 2. There is problem with the database. 3. Search parameters are invalid.
Postconditions	1. Expert got information which he or she requested.

5.2.2

Expert user browses unified cases, selects one case and checks it. Then he or she selects other case of the unified case or the document and checks it too. After that he or she **chooses many cases and verify them all simultaneously**.

Functional requirements: **T1, T2, T3, T4, T5, T6, U4**

Preconditions	Expert wants to verify many cases simultaneously.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case and checks its attributes. 3. He or she also checks some other cases of the unified case and/or the document. 4. He or she choose these cases and verify them simultaneously. 5. He or she sends a message of verifications to the database. 6. The database receives it and marks cases to be verified.
Exceptions	1. Some kind of connection problem occurs between Expert and the database. 2. There is problem with the database.
Postconditions	1. Cases have been verified.

5.2.3

Expert user browses unified cases, selects one case and checks it. Then he or she selects other case of the unified case or the document and checks it too. After that he or she **chooses many cases and corrects them all simultaneously**.

Functional requirements: **T1, T2, T3, T4, T5, T6, U4**

Preconditions	Expert finds same error on many cases and wants to correct them simultaneously.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case and checks its attributes. 3. He or she also checks some other cases of the unified case and/or the document and notices that they all have same errors. 4. He or she chooses these cases and corrects their attributes simultaneously. 5. He or she sends new values of attributes to the database. 6. The database receives them and corrects cases.
Exceptions	1. Some kind of connection problem occurs between Expert and the database. 2. There is problem with the database. 3. The correction is invalid.
Postconditions	1. Cases have been corrected.

5.2.4

Expert user browses unified cases, checks its every cases and **verifies that unified case is unified correctly**.

Functional requirements: **T1, T2, T3, T4, T5, U5**

Preconditions	Expert wants to verify the unified case is unified correctly.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting unified case and checks its every cases. 3. He or she verify that the unified case is unified correctly. 4. He or she sends a message of verification to the database. 5. The database receives it and marks the case to be verified.
Exceptions	1. Some kind of connection problem occurs between Expert and the database.

	2. There is problem with the database.
Postconditions	1. The unified case has been verified.

5.2.5

Expert user browses unified cases, selects one case and checks it and **remove it from the unified case**.

Functional requirements: **T1, T2, T3, T4, T5, U6**

Preconditions	Expert wants to remove the case from the unified case.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case and checks its attributes. 3. He or she notice that the case doesn't belong to the unified case at all. 4. He or she removes it from the unified case. 5. He or she sends a message of removing to the database. 6. The database receives it and removes the case from the unified case.
Exceptions	1. Some kind of connection problem occurs between Expert and the database. 2. There is problem with the database.
Postconditions	1. The case is removed from the unified case.

5.2.6

User browses unified cases, **chooses one case and views it on the map**.

Functional requirements: **T1, T2, U2**

Preconditions	User wants to see unified case on the map.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she clicks "Map"-link. 3. He or she views results on the map
Exceptions	1. Some kind of connection problem occurs between user and database. 2. There is problem with database. 3. There is problem with Google Maps.
Postconditions	User views results on the map.

5.2.7

User browses unified cases and writes **search parameters and view results on the map**.

Functional requirements: **T1, T2, U1, U2**

Preconditions	Expert wants to view a certain unified case on the map.
Basic flow	1. Expert goes to Unified case view and begins to browse unified cases. 2. He or she writes search parameters. 3. He or she clicks "Map"-link.

	4. He or she views results on the map
Exceptions	1. Some kind of connection problem occurs between user and database. 2. There is problem with database. 3. Search parameters are invalid. 4. There is problem with Google Maps.
Postconditions	User views search results the map.

5.2.8

Average user discovers that there is problem with case and **informs other users about the problem**.

Functional requirements: **T1, T2, T3, T4, T5, U8**

Preconditions	Average user discovers problem with the case.
Basic flow	1. Average user goes to Unified case view and begins to browse unified cases. 2. He or she finds an interesting case, checks its attributes and notice that there is error. 3. He or she fills problem form and sends it to the database. 4. The database receives report about problem and shows it to other users.
Exceptions	1. Some kind of connection problem occurs between user and database. 2. There is problem with database.
Postconditions	1. The report of problem is shown to other users.

6. User interface

6.1 Phase I

Unified case view

This view provides user the newest unified cases. Each unified case consists of one or more cases which have identical attributes. There is some information about each group like disease and country and maybe a small description. Each unified case has also lists of cases and links to these documents where cases were found. By clicking a case of the unified case, user can open document view, where it is possible to view a document containing the case in question.

Document view

This view shows a document and all its cases and informs the user somehow of all the other cases of the unified case. Links are provided to each document where these cases were found. The user can correct the case and confirm that the case has been grouped correctly. Cases from identical documents are automatically confirmed.

6. 2 Phase II

Plain case view

Allow users to see all cases without any grouping. Use cases of this view are quite same as use cases of Unified case view.

Groups view

This view allows users to see groups of unified cases which are related to same disease in same country on period of two weeks. The view will be implemented only if time scale of the project allows it.

Map view

Allow user to view search results on the map.

Search pane

User can use both simple search bar and filters. Filters allows users to create views where are certain cases. Search pane can be embedded to many views.

7. Functional requirements

7.1 Phase I

Code	Requirement	Priority
T1	On Unified case view each table line corresponds to a unified case which consist of one or more cases. Cases are grouped if they have identical attributes.	High
T2	By moving mouse cursor over subject of news, user can see small summary of news on snippet. Same functionality could be done by other ways, so this might not be final solution.	Low
T3	Document view shows a document and all its cases.	High
T4	The Document view informs the user about the existence of all the other cases that belongs to same unified case. Links are provided to these cases.	High
T5	On Document view Expert can check if attributes of the case are correct.	High
T6	On Document view Expert can correct attributes of the case and verify them.	High
T7	On Document view Expert can create new cases from documents.	High
T8	On Document view Expert can reject false cases from documents.	High

7.2 Phase II

Code	Requirement	Priority
U1	User can search cases by writing search parameters. This functionality can be used in many views.	High
U2	User can use map to help to get better picture of current situation. Search functionality is also available in this view.	High
U3	User can use charts to help to get better picture of current situation.	Low
U4	On Document view Expert can correct and verify many cases simultaneously.	Medium
U5	On Document view Experts can verify that the unified case has been grouped correctly.	Medium
U6	On Document view Expert can remove the case from the unified case.	Medium
U7	System support many different user groups but their rights to use to the system won't be implement.	Medium
U8	Average user can inform other users about error in a case.	Low

8. Non-functional requirements

Code	Requirement	Priority
E1	User interface is going to be programmed in LISP.	High
E2	English is used in naming of programming variables, comments and other.	High

E3	System environment is on CSL linux on db or sysdb.	High
E4	System has to react in under 5 seconds on 90 % of cases, only for about 10 users who are using system at the same time.	Medium
E5	Project is going to be implemented with MYSQL (v.4.1.12) database and SBCL-lisp environment.	High
E6	Web implementations will be validated with following standards: xhtml 1.0 transitional and css 2.1.	Medium
E7	Pages have to work with Internet Explorer.	Low
E8	Later it is possible to extend user interface to other scenarios and do other improvements.	High