Software Requirements Specification

for

SLIPPS – Sharing Learning Events about Patient Safety

Version 1.0 approved

Prepared by <author>

<organization>

<date created>

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Product Scope 1

1.3 Definitions, acronyms and abbreviations 1

1.4 References 1

2. Overall Description 1

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Software Interfaces 4

4. System Features 4

4.1 Feature: Search by keyword 4

4.2 Search by content 5

4.3 View search result in detail page 5

4.4 Translate keyword 5

5. Other Nonfunctional Requirements 5

5.1 Safety Requirements 5

5.2 Security Requirements 5

Appendix A: Glossary 6

Appendix B: Analysis Models 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This project is a part of SLIPPS (Shared Learning from Practice to Improve Patient Safety) – a 3-year Erasmus+ funded Patient Safety education project. The aim of this project is to create a platform where the learning events collected from patient safety experiences can be shared among users and services for medical studies.

## Product Scope

* To create a multilingual platform to collect and share experiences of medical students.
* To create a unified list of medical terms (keywords) in English with translations to other languages.
* To create a search engine based on keywords.
* To support translation of medical terms in a set of available languages.

## Definitions, acronyms and abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| csv | Comma separated values. |
| Medical translator | Translator who works on medical documents. |
| Learning event | Description of an experience shared by user. |
| Keywords | A medical term in this context. |
| LERT | An application to extract medical keywords in learning event collected from patient safety experiences. |
| DB | Database |

## References

[1] IEEE, “IEEE Recommended Practice for Software Requirements Specification,” *IEEE Std 830-1993*, vol. 1998. p. 32, 1998.

# Overall Description

## Product Perspective

This platform is part of a European Union project called **SLIPPS** which aims towards sharing knowledge in the medical field. The purpose of this project is to create a database of events experienced by various users and make them available for other people to access. This will create a knowledge community allowing to produce ideas and solutions easily and effectively. This platform will enable the realization of this database and will have certain various functionalities that will allow users to utilize the learning experiences in an efficient way.

The platform is an extend service that will integrate with some available tools or services (i.e. crowd sourcing) as input or output. In the first stage of the implementation, we will connect with LERT.

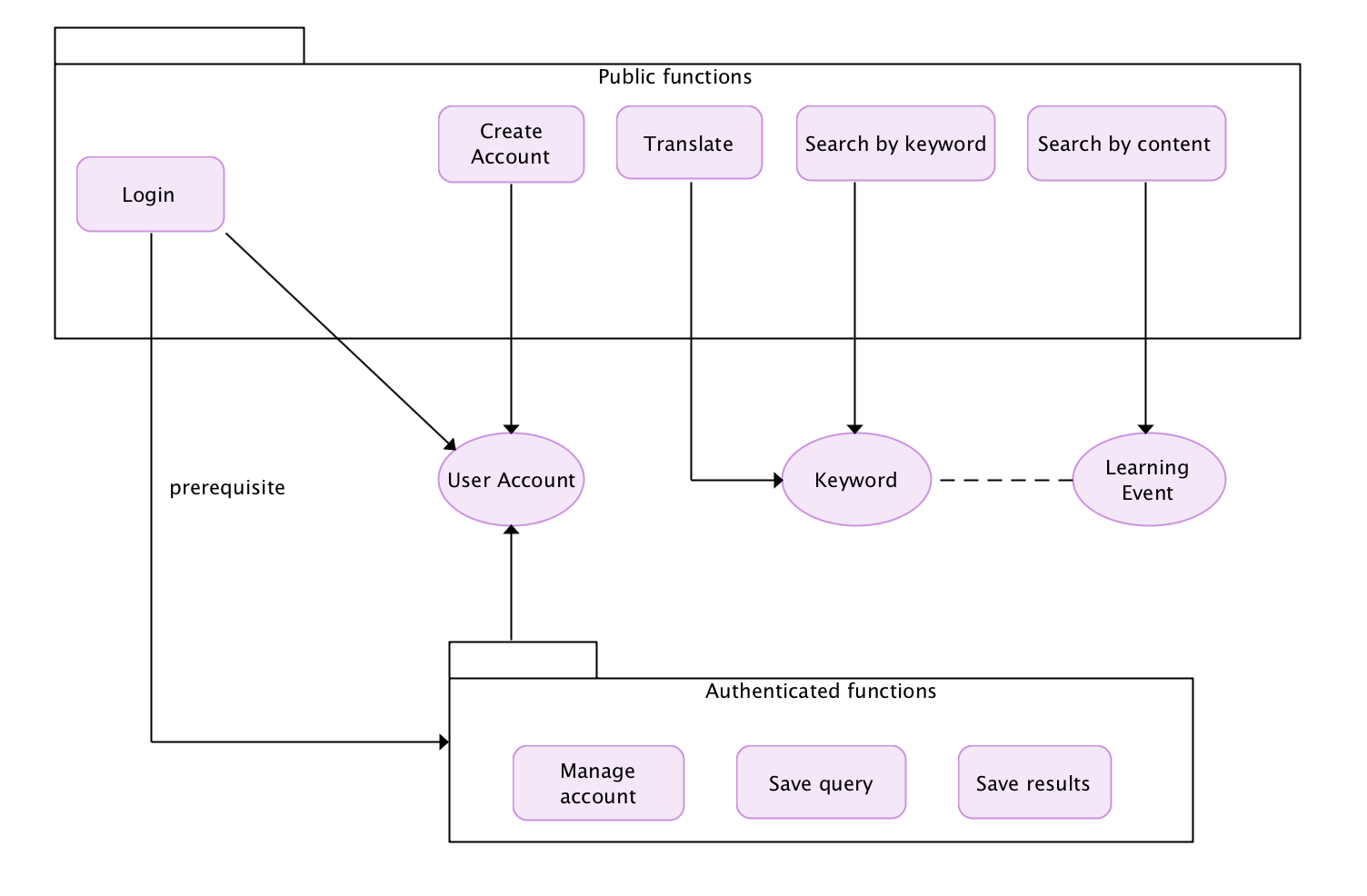
## Product Functions

### Public functions – users can perform without an authentication:

* Search by keywords.
* Search free text in content.
* Find translation of keywords.
* Create account for personalized functions.
* Sign in

### Authenticated functions – functions that require user to have an account and logged in:

* Manage account.
* Save search query and results.
* Save keywords.
* Get update on change (if any) of the saved keywords.



## User Classes and Characteristics

*The users for this platform are specifically limited to people involved with the medical field. There are three main classes of users regarding this platform:*

* **Primary Users**: The primary users of this product are the medical students and doctors. They will the ones using this platform the most frequently and interacting with the platform the most.
* **Secondary Users**: The secondary users are the users who use the application in a less frequent manner. These includes:
  + Medical translators: translate the keywords and the learning events from one language to another in order to make possible multilingual services.
  + Nurse: search for learning events.
  + Medicine Faculty in university: search for patient safety learning events for education or research purpose.
* **Tertiary Users**: The tertiary or indirect users are the patients. They will not make use of the platform directly and will not interact with it. But they are affected by the use of this platform by the primary and secondary users.

## Operating Environment

Functions are accessible for use through web application. Thus, our application will be operating system independent.

## Design and Implementation Constraints

*In order to integrate with other services, there are design constraints that need to be followed:*

* Process .csv files: our application will receive input from LERT under csv format. In order to read and use keywords from LERT, our application needs to be able to work with csv file.

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

* Search page
* Results page
* Login page
* Create account page
* Manage account page
* Translation page? Or translation control menu?

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

* LERT
* Database
* Sketch (design sketch).
* Frontend libraries.

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## Feature: Search by keyword

4.1.1 Description and Priority

Allow user to search by any keywords. This is the main function of our application and will perform search on keyword field only.

4.1.2 Stimulus/Response Sequences

* User goes to website as guest or as registered user.
* User enters keyword in “Keyword search box”.
* The system searches through “Keyword” field in DB and returns relevant results.

4.1.3 Functional Requirements

REQ-1: User can search for keyword.

REQ-2: User can see the relevant results from their query.

## Search by content

4.2.1 Description and Priority

Allow user to search by free text. This function will perform search in content field.

4.2.2 Stimulus/Response Sequences

* User goes to website as guest or as registered user.
* User enters text to search in “search box”.
* The system searches through content field in DB and returns relevant results.

4.2.3 Functional Requirements

REQ-1: User can search for content.

REQ-2: User can go through the relevant results from their query.

## View search result in detail page

4.2.1 Description and Priority

Allow user to view search result in a new page with full detail of the learning event.

4.2.2 Stimulus/Response Sequences

* User goes through search results.
* User can click on one of the results.
* Redirect to learning event detail page.

4.2.3 Functional Requirements

REQ-1: User can search for content.

REQ-2: User can go through the relevant results from their query.

## Translate keyword

4.2.1 Description and Priority

Allow user to translate keywords to available languages.

4.2.2 Stimulus/Response Sequences

* User goes to website as guest or as registered user.
* User selects the target language.
* Enter keyword.
* Translate.

4.2.3 Functional Requirements

REQ-1: User can translate keywords to other language.

# Other Nonfunctional Requirements

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

* Backup of external services that are integrated in our system.

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

* Anonymity
* User authentication.
* User authorization.

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>