Software Requirements Specification

for

SLIPPS – Sharing Learning from Practice to improve Patient Safety

Version 4.0 approved

Prepared by Anastasiia Grishina

SLIPPS team

12.04.18

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** |
| An Pham, Ijlal Niazi, Amir Rahafrouz | 6.02.18 | Creation of the document | 1.0 |
| Anastasiia Grishina | 23.02.18 – 07.03.18 | New system requirements acquisition | 2.1, 2.2 |
| Anastasiia Grishina | 07.03.18 – 23.03.18 | Admin requirements elicitation | 2.3 |
| Anastasiia Grishina | 23.03.18 – 15.04.18 | Technical requirements documentation | 3 |
| Anastasiia Grishina | 15.04.18 – 18.04.18 | Adjustment after the final meeting | 4 |

# 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 in order to collect and share experiences of medical students.
* To integrate a unified list of medical terms (keywords) in English with translations to other languages, currently Finnish.
* 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 a practical experience shared by user. |
| Keyword | A medical term in this context. |
| LERT | Learning Evet Recording Tool  An application to submit answers to the questionnaire about practical experience. It further extracts 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 several 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 by User Roles

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

*User role: guest.*

* Search by keywords.
* Search by multiple criteria: keywords, language, country, date.
* Get translation of keywords.
* Create account for personalized functions.
* Sign in.
* Preview learning events.

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

*User role: registered user.*

* Manage user`s profile.
* Save search query and results.
* Save keywords.
* Get full text of learning events.
* Upload learning events.
* Download learning events.

### Management administrator functions – functions that require user recognition as an administrator of the system

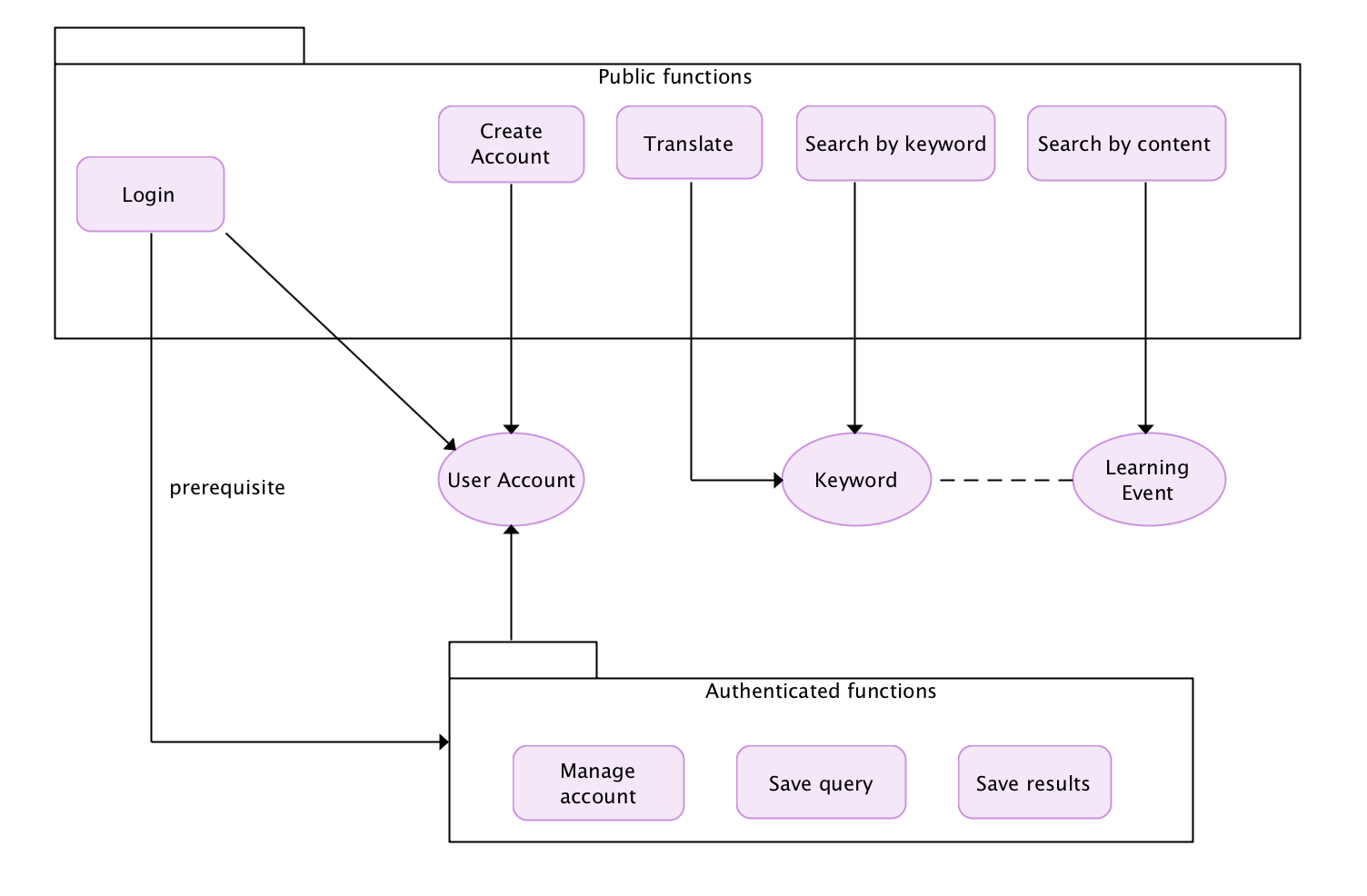
*User role: management administrator.*

* Get notifications of uploaded events which require verification.
* Check the content of uploaded learning events.
* Discard uploaded learning events in case of duplicates, lack of anonymity, irrelevance, conflicts with terms of use.
* Change the content of uploaded learning events to anonymize them.
* Notify and ban users who do not respect terms of use.
* Confirm users` credentials.
* Give administrator rights to other users.
* Manage statistics gathered from the platform use.

### Technical administrator functions

*User role: technical administrator.*

* Maintain system
* Develop technical support
* Upgrade the system



## User Classes and Characteristics

*The users for this platform are specifically limited to people involved in 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 be 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.
  + Governmental healthcare system: monitor development of the project with a future goal to integrate the practice of sharing medical learning experiences in a broader range of institutions.
* **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 files.
* Integrate localization functions
* Integrate analytical algorithm for search results sorting by relevance.

# External Interface Requirements

## User Interfaces

* Search page
* Advanced search page
* Results page
* Detailed learning event page
* Login page
* Sign up page
* Create account page
* Manage account page
* Language localization
* Uploading page

## Software Interfaces

* LERT – get learning events
* Database – store learning events, language, country, translation
* Frontend architecture
* Backend architecture

### Backend

An API server is built to serve requests from frontend server, backboned by PostgreSQL as database management and ElasticSearch for indexing and searching.

* Language: Python in combination with the following libs/frameworks
  + Django-admin
    - One of the most powerful parts of Django is the automatic admin interface. It reads metadata from your models to provide a quick, model-centric interface where trusted users can manage content on your site. We use this framework to build SLIPPS admin interface.
  + Django Rest Framework
    - Django REST framework is a powerful and flexible toolkit for building Web APIs.
  + elasticsearch-dsl
    - Elasticsearch DSL is a high-level library whose aim is to help with writing and running queries against Elasticsearch. It is built on top of the official low-level client (elasticsearch-py).

### Frontend

* React-Redux App Architecture
  + Entry Point
    - File: index.js Imports: agent.js, store.js, ./components/\*
    - Renders routes are pointing to their associated components.
  + Agent - API CLient
    - **File**: agent.js
    - **Purpose**: manage requests to api server.
    - **Usage**: Exports an object where each key is a "service" and a service has methods that internally run a request: > GET, PUT, POST, DELETE These methods have been defined.
  + Redux

Here is summary of how we apply this in our context.

* + - ***Store***

File**: store.js**

Imports**: reducer.js, middleware.js**

Fairly simple store setup, applies promiseMiddleware (handle asynchronous requests), logger-redux (for development only).

* + - ***Middleware***

**File**: middleware.js

**Imports**: agent.js

promiseMiddleware Intercepts all actions where action.payload is a Promise.

Then, for success and error, using the modified action object: store.dispatch(action)

localStorageMiddleware Runs after promiseMiddleware. Intercepts REGISTER | LOGIN to manage session data.

* + - ***Reducers***

**File**: reducer.js

Imports: ./reducers/\*.js

Uses combineReducers to export a reducer where each key is the reducer of the file with the same key.

**General Reducer Patterns**: map payload into piece of state

e.g: When user enter a keyword (which is sent in action.payload), save this keyword into state variable to share between components

# System Features

## Feature: Search by keyword

* + 1. Description and Priority

Allow user to search by keywords stored in the system. This is the main function of our application. It will perform search on keyword field only.

High priority

* + 1. Stimulus/Response Sequences
* User goes to the website as guest or as registered user.
* User enters a keyword in “Keyword search box”.
* The system searches through “Keyword” field in DB and returns relevant results.
* The system analyzes entered keywords and provide suggested keywords of the same language or from the same category.
  + 1. Functional Requirements

REQ-1: User can search for an event by keywords.

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

REQ-3: Search is performed by entered keywords and their equivalents in all possible languages.

REQ-4: Results are sorted by the relevance to entered keywords.

## Feature: Advanced search

* + 1. Description and priority

Allow the user to search by criteria other than keywords.

Medium priority

* + 1. Stimulus/Response Sequences
* User goes to website as guest or as registered user.
* User wants to use more searching criteria than only keywords and chooses “Advanced search” option.
* The system suggests fields of advanced search.
* The system searches for matches in the names of events stored in the system.
  + 1. Functional Requirements

REQ-1: User can search for an event by category of keywords or one keyword, a country, language.

REQ-2: Language, country and category fields should be drop-down lists of available options of advanced search.

REQ-3: Results are sorted by keywords relevance if they are stated, or by category, or by language, or by country, or by combination of those stated by the user.

## Feature: View search result details

* + 1. Description and Priority

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

High priority

* + 1. Stimulus/Response Sequences
* User goes through search results.
* User can click on one of the results.
* Redirect to learning event detail page.
  + 1. Functional Requirements

REQ-1: User can search for content.

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

REQ-3.1: If a user is a guest, details of the learning events are hidden.

REQ-3.2: If a user is registered in the system, this user can view full-detailed learning events and download them.

## Feature: Download a learning event

* + 1. Description and Priority

Allow a registered user to download a learning event.

High priority

* + 1. Stimulus/Response Sequences
* User is on the page of a learning event.
* User wants to download a file with the learning event.
* Guest user cannot download the event.
* Registered user clicks the download button.
* Registered user is asked to complete a questionnaire about his/her interest in downloading the learning event.
* Registered user completes the questionnaire or skips it.
* Registered user downloads the learning event.
  + 1. Functional Requirements

REQ-1.1: Guest user must not have an option to download a learning event file.

REQ-1.2: Guest user is suggested to register if he/she wants to get access to details of learning events and download files.

REQ-2: Registered user must have an option to download a learning event file.

REQ-3: Registered user can download an event only upon completion of a questionnaire or after skipping this step.

REQ-4: Reasons for providing answers to a questionnaire are explained to a registered user.

## Feature: Upload a learning event

* + 1. Description and Priority

Allow a registered user to upload a learning event.

Medium priority

* + 1. Stimulus/Response Sequences
* User is on the profile page.
* User has some information to share and to upload a file with the learning event.
* Registered user clicks the upload button.
* Registered user is asked to agree with terms of use and specifically agree that the anonymity check has been done.
* Registered user uploads an event.
  + 1. Functional Requirements

REQ-1. Registered user must be able to upload a learning event.

REQ-2. Guest user does not have a possibility to upload an event.

REQ-3. Registered user must agree with the terms of use and uploading.

REQ-4. The uploaded file must be stored in the system, and the administrator must be notified about a new file submission.

REQ-5. Administrator must check the file for anonymity. If the event is not confidential, the file should be sent back to the user`s email with notification about not complying to terms of use. Otherwise, the administrator changes the text and publishes it on the SLIPPS website.

## Feature: Providing statistics of search activities

* + 1. Description and priority

Store search statistics and show it to the user.

Low priority

* + 1. Stimulus/Response Sequences
* User goes to the website as guest or registered user.
* User performs advanced or basic search.
* The system shows suggested documents based on analysis of what is searched for together with the entered search information.
  + 1. Functional Requirements:

REQ-1: User can see the suggested learning events and keywords.

REQ-2: Statistics is managed by the administrator.

REQ-3: User can see how many other users viewed the same learning event.

## Feature: statistics collection by administrator

* + 1. Description and priority

Collect website use statistics for further analysis and partial publishing.

Medium priority

* + 1. Stimulus/Response Sequences

1. Search statistics: criteria and viewed events association

* The learning events, which a user opens after searching, are stored in association with the search keywords or advanced search criteria

1. Website access times/day

* The number of users` website access per day should be stored to measure users` activity

1. Up-/download events/day

* The number of information exchange acts, i.e. of uploading and downloading events per day, should be measured to assess the information share rate

1. Pre-downloading questionnaire results

* Graphs of commitment per country and of the reasons for downloading should be shown in the admin profile
  + 1. Functional Requirements:

REQ-1: Administrator dashboard should include statistical results.

REQ-2: Association between search criteria, primarily, keywords, and learning events should be stored in additional fields/tables in a database. Administrator should be able to accept suggested association rules to avoid random association.

REQ-3: Website access per day, uploaded events/day, downloaded events per day should be displayed in a form of weekly line graphs.

REQ-4: Answers on pre-downloading questionnaire should be shown in a form of column charts.

## Feature: administrator control of other users` profiles

* + 1. Description and priority

Profiles management.

High priority

* + 1. Stimulus/Response Sequences

1. Ban a user

* Administrator analyzed a learning event.
* Analysis shows that the event is not relevant (written for not-intended purpose, like a joke, or contains information which humiliates other users, or contains abusive/inappropriate language).
* Administrator bans the profile

1. Upgrade a registered user to be an administrator

* Administrator gets notification about a user`s request to be an administrator
* Administrator verifies the user`s credentials
* Administrator gives admin rights to the user in case the user is proved to be able to manage a branch of the system associated with the user`s native language
  + 1. Functional Requirements:

REQ-1: Administrator should have an opportunity to disable other user`s profiles and notify users about the reason

REQ-2: Administrator should be able to give administrator rights to other users.

# Other Nonfunctional Requirements

## Safety Requirements

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

## Security Requirements

### Anonymity

REQ-1: Before a registered user uploads a learning event, the user is asked to confirm that the document is checked for anonymity.

REQ-2: The administrator checks anonymity of the uploaded learning events before publishing.

### User authentication.

REQ-1: Users can sign up via ordinary email, Google or Facebook. They get an email informing they have created an account with a link to confirm it.

REQ-2: Users log in with their credentials and a password they have chosen.

REQ-3: Registered users are asked to specify their job position and field of interest.

### User authorization.

REQ-1: Unauthorized users are considered as guests

REQ-2: Authorized users who have a registered profile are able to:

* Change their profile
* View details of learning events
* Upload learning events
* Download learning events

REQ-3: Authorized users, who act as administrators, are able to:

* Manage other users` profiles, namely, give admin rights, ban users, check credentials
* Check learning events on anonymity, anonymize them or send back to the user who uploaded it with notification of irrelevance/lack of anonymity.
* Observe and manage statistics
* Manage the website, including the technical level