

System Requirement Specification

Lezioni alla pari

April 19, 2020

Team Members

Ovidiu Andrioaia
David Cirdan
Luciano Mateias
Zhiyang Xia

Document Control

Change History

Revision	Change Date	Description of changes
V1.0	04/20/2020	Initial release

Document storage

This document is stored in the project's GIT repository at:
<https://github.com/KilliKrate/Software-Documentation-G6/blob/master/docs/Software%20Requirements%20Specification/index.md>

Document Owner

Group 6 is responsible for developing and maintaining this document.

[Table of contents](#)

[Introduction](#)

[Overview](#)

[Goals and Objectives](#)

[Scope](#)

[Definitions](#)

[Document Conventions](#)

[Assumptions](#)

[General Design Constraints](#)

[Product Environment](#)

[User Characteristics](#)

[Mandated Constraints](#)

[Potential System Evolution](#)

Nonfunctional Requirements

Operational Requirements

Performance Requirements

Security Requirements

Safety Requirements

Legal Requirements

Other Quality Attributes

Documentation and Training

External Interface

User Interface

Software Interface

System Features

Authentication

Description and Priority

Use case: Registration

Additional requirements

Description and Priority

Use case: Login

Additional requirements

Lessons

Description and Priority

Use case: create a lesson

Additional requirements

Description and Priority

Use case: manage a lesson

Additional requirements

Quizzes

Description and Priority

Use case: create a quiz

Additional requirements

Description and Priority

Use case: manage a quiz

[Additional requirements](#)

[Inviting](#)

[Description and Priority](#)

[Use case: Inviting user to a course](#)

[Additional requirements](#)

Introduction

Overview

Lezioni alla pari is an application that focuses on simplifying the sharing of knowledge between various parties inside a closed environment (e.g. the departments of a company, a classroom, a sports club) easier and within everyone's reach. The software streamlines the creation of courses that can contain documents, videos and quizzes on a wide range of topics.

This document provides information on the requirements of the Lezioni alla pari application, and aims to deliver an intelligible documentation of the software to the developers that can use it as a reference, as well as their customers, that can have a deeper insight of the project and its sophistications.

Goals and Objectives

The main objective of this project is to develop a platform that will provide the following:

1. Login and registration system for users
2. Intuitive graphical user interface
3. Course creation system with lessons that can contain documents, videos or quizzes
4. Automated quiz correction

Scope

Lezioni alla Pari will store user information and documents inside an SQL database, while video lessons will be delivered through a dedicate content streaming server. Content is meant to be private to the group, the only way to access someone else's documents is to be invited by the author of the course, or the collaborators themselves. The application is open-source and can be downloaded through Github, and can be installed on any desktop that supports Python-based applications (this includes the main operating systems, like Windows, Linux and Mac OS)

Definitions

- **Use Case:** Describes a goal-oriented interaction between the system and an actor. A use case may define several variants called scenarios that result in different paths through the use case and usually different outcomes.
- **Actor:** Person or program that receives an output from a use case, be it in the form of an UI (person) or value (software).
- **System:** Set of features provided by the application that are directly associated with each other.

- **Scenario:** One path through a set of possible use cases.
- **Lezioni alla Pari:** What is being described here, the software that is being developed, specified in this document.
- **Project:** Activities that will lead to the production of the application described in this document.
- **User Interface:** Visual component that enables interaction with a person that is using the software.

Document Conventions

TBD (to be defined / to be discussed) - information that still needs to be gathered, framed or that it is incomplete.

Unknown - development feature, option that hasn't been implemented yet.

Assumptions

- Team members will work from home, since no office space is available. Collaboration will be made possible through Github.

General Design Constraints

Product Environment

Lezioni alla Pari is the first product in our series dedicated to online learning and does not require any third party software. It is portable and usable by any operating system that supports python3.

User characteristics

In order of increasing priority, the following categories of users can be distinguished for this application:

- Companies and other work areas, who can use our product to simplify the learning of information (for example the safety courses), provide easy access to any type of project documentation, and also provide a way for interns and new employees to learn skills relevant to their position
- Schools, who can use our product to provide a streamlined learning experience. This can include teachers and professors, who can use lessons and tests as extra reading and homework, and as a way to grade students.

Mandated Constraints

The end-user must have a device with an OS that supports python3, and can reliably handle video decoding, in order to watch video lessons.

Potential System Evolution

The final product will be a desktop applications, so UI and UX are optimized for this platform only. As successful adoption of our platform increases, Lezioni alla Pari will transition to a web-based platform, and finally an Android and iOS application.

Nonfunctional Requirements

Operational Requirements

- The application will allow users to concurrently access and edit documents. In case a lesson is edited, all users currently accessing said lesson will be notified and offered to switch to the new version, or keep reading/watching the current one.

Performance Requirements

- The server will accommodate, at release, up to 100TB of data, delivered through a distributed server system, which provides efficient document delivery and video streaming all over the world. This will of course be scalable as the application becomes more popular.
- Response times may vary based on server workload. All database operations like login, registration and document reading should take <1 second.
- Document editing may take more based on the size of the uploaded images, if any.
- Video lesson streaming is guaranteed to be fluid with resolutions up to 720p, 1080p and more will be available, but may result too slow if the content delivery servers are under heavy workload.
- Video lesson upload will take more, based on server workload and user connection speed.

Security Requirements

- The application must use the most up to date standards and practice for its security: sensible information created and shared by users must be kept private and confidential. The software must adhere to the NIST, OWASP and CWE standards, in order to protect data from malevolent attacks to the platform.
- The application must guarantee absolute privacy: data shared through our servers will be encrypted and inaccessible by anyone but the intended recipients of the information. This also includes the team members themselves.

Safety Requirements

No safety requirements were identified for this application.

Legal Requirements

All data delivered through the platform is private, and will thoroughly adhere to GDPR laws and OECD Privacy Principles. Copyright will not be enforced on user-made content, since it is of a private, non-commercial and non-redistributable nature.

Other Quality Attributes

- The software is portable on all desktop computer platforms, since the application heavily relies on Python. A mobile app will be made available if demand justifies it.
- UI is built from the ground-up to be intuitive and easy to understand without need to consult a user guide or manual. This is intended to provide ease of

access to a wide demographic.

Documentation and Training

- No training will be delivered when the application is delivered. A short tutorial will introduce the new users to the main features of the application. The application should be extremely intuitive to use, so UX will be a big focus during development.

External Interface

User Interface

- The user interface for the application will follow a modern and minimalist style, which makes important elements easily noticeable, and keeps the main features always in view.
- The login and registration forms will be quick and straightforward, with the purpose of introducing the user to the application immediately.
- An user will easily be able to determine the courses and lessons he has created, thanks to smart color use, and will also have easy access to any content he needs.
- Accessibility options, such as increased font sizes and color blindness support will be available at launch.

Software Interface

- All operations and logic defined by the use cases will be achieved through Python and SQL queries. SQL will take care of login, registration and course access, while a cooperation between SQL and Python will achieve document delivery and video streaming

System Features

Authentication

Description and Priority

Cost: Low

Risk: Medium

Benefit: High

Value: High

Use case: Registration

Actors: any unauthenticated in user

Description: This use case begins when an unauthenticated user wants to create a new account to use the Lezioni alla Pari platform.

Basic path:

1. The user enters the registration form by pressing "register" button.

2. The program requires the user to insert some personal information: name, surname, password, email and birthdate. Each field must match a specific pattern specified by the program
3. The program pops up a confirmation and redirects the user to the homepage.

Alternate path:

1. If the pattern in step 2 is not matched, the user will be presented with a dialog box with an error that specifies the cause.
2. The user can try again to register.

Additional requirements

N/A

Description and Priority

Cost: Low

Risk: Low

Benefit: High

Value: High

Use case: Login

Actors: any registered user

Description: This use case begins when a registered user wants to log in to the system with an already existing account.

Basic path:

1. The user enters the login form by pressing "login" button
2. The program requires the user to insert email and password of his account. Each field must match a specific pattern specified by the program
3. The program pops up a confirmation and redirects the user to the homepage.

Alternate path:

1. If the credentials used in step 2 are not correct, the user will be presented with a dialog box with an error that specifies the cause.
2. The user can try again to login.

Additional requirements

N/A

Lessons

Description and Priority

Cost: Medium

Risk: Low

Benefit: High

Value: High

Use case: create a lesson

Actors: any authenticated user

Description: This use case begins when an authenticated user wants to create a lesson.

Basic path:

1. The user selects a course and a topic, then presses the "+" button on top of the 3rd column.
2. The program displays the lesson creation form and requires the user to insert the name of the lesson, after that the lesson is created. A name is required in order to proceed.

Alternate path:

1. If the name used in step 2 is not valid, the user will be presented with a dialog box with an error that specifies the cause.
2. The user can try again to create a lesson.

[Additional requirements](#)

N/A

[Description and Priority](#)

Cost: Medium

Risk: Medium

Benefit: High

Value: High

[Use case: edit a lesson](#)

Actors: any authenticated user

Description: This use case begins when an authenticated user wants to edit a lesson previously created (see 4.2.2).

Basic path:

1. The user right-clicks on the lesson, this causes a new menu to appear, close to mouse cursor. On the new menu the user right-clicks on "Modifica" to edit a lesson's content.
2. The program displays a new page with a RTF Editor where the user can freely customize the lesson and then submit the changes.

Alternate path: N/A

[Additional requirements](#)

It is assumed that the user that wants to manage the lesson is authorized to do so. This is possible only if the user is the author or a collaborator of the course

[Quizzes](#)

[Description and Priority](#)

Cost: Medium

Risk: Medium

Benefit: High

Value: High

[Use case: create a quiz](#)

Actors: any authenticated user

Description: This use case begins when an authenticated user wants to create one or more quizzes.

Basic path:

1. The user selects a course and a topic, then presses the "+" button on top of the 3rd column.
2. The program displays the quiz creation form and requires the user to insert the name of the quiz, after that the quiz creation form is displayed to the user.
3. The user creates the quiz using the fields of the form and then submits it, after that the quiz is created.

Alternate path:

1. If the name used in step 2 is not valid, the user will be presented with a dialog box with an error that specifies the cause.
2. The user can attempt to create a quiz.

[Additional requirements](#)

It is assumed that the user that wants to manage the quiz is authorized to do so. This is possible only if the user is the author or a collaborator of the course

[Description and Priority](#)

Cost: Medium

Risk: Medium

Benefit: High

Value: High

[Use case: edit a quiz](#)

Actors: any authenticated user

Description: This use case begins when an authenticated user wants to edit a quiz previously created (see 4.3.2)

Basic path:

1. The user right-clicks on the quiz, this causes a new menu to appear, close to mouse cursor. On the new menu the user right-clicks on "Modifica" to edit a quiz's content.
2. The program displays a new page with an intuitive quiz editor where the user can freely customize the quiz and then submit the changes.

Alternate path:

N/A

[Additional requirements](#)

N/A

Description and Priority

Cost: Medium

Risk: Low

Benefit: High

Value: Medium-high

Use case: results of a quiz

Actors: any authenticated user

Description: This use case begins when an authenticated user wants to see the results of a quiz previously created (see 4.3.2)

Basic path:

1. The user right-clicks on the quiz, this causes a new menu to appear, close to mouse cursor. On the new menu the user right-clicks on "Revisione" to review a quiz's results.
2. The program displays a review panel where the user can easily access the result of every quiz participant by left-clicking on his username.

Alternate path:

1. If no user participated no list will be displayed.

Additional requirements

It is assumed that the user that wants to see the results is authorized to do so. This is possible only if the user is the author or a collaborator of the course

Inviting

Description and Priority

Cost: Low

Risk: Low

Benefit: High

Value: High

Use case: Inviting user to a course

Actors: any author, the course creator

Description: This use case begins when a course creator wants to invite another user to his course via mail.

Basic path:

1. The user right-clicks on the course, this causes a new menu to appear, close to mouse cursor. On the new menu the user right-clicks on "Modifica".
2. A new page will be displayed with a form asking the user to type the email of the user he intends to invite.

3. The user submits the form and the invite is sent.

Alternate path:

N/A

[Additional requirements](#)

N/A