SKPL-xxxx

SOFTWARE REQUIREMENTS SPECIFICATION

eLab

Directed to:

RPL/GDC Laboratory

Prepared by:

Khaidir Mauladan	1301192327
Aulia Rahman Arif	1301194195
Putu Budi Sukarya Putra Purnawan Oka	1301194252
Anggi Rodesa Sasabella	1301193161

Program Studi S1 Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

Table of Contents

1. Introduction

1.1 Document Purpose

The purpose of this Software Requirement Specification (SRS) document is to present a detailed description of the application-based archive for Telkom University laboratories known as eLab. This document will explain the purposes and features of the software, the interfaces of the software, what the software will do, and the constraints under which the software must operate.

This SRS documents is targeted for the use of all individuals that are involved in the development of the eLab software. This document will be utilized as guidance and reference for the software development as well as an evaluation device in the final process of development as well as during the development process.

1.2 Document Conventions

- Underline – Determines an important point
- CRUD – short for Create, Read, Update, and Delete

1.3 Product Coverage

eLab is an application-based method of archiving documentations of Telkom University laboratories. Its main purpose is to allow users, which in this case the students at Telkom University to find, recap, or upload documentations of previous and upcoming project or achievement of a lab in Telkom University with ease in the form of Instagram-like posts that can be filtered.

There are two major features which mainly focused on filtration and authorization. eLab provides full functionality to assigned laboratories member to upload, edit, or remove a post from the feed, while limiting default users to filtered searching, personal archive saving, and other basic functions of the application.

Current application that provides similar functionality include Instagram and LINE Official Accounts. Instagram has a very clean and straightforward system to make and search a post parallel to LINE Official Accounts, but both lack the capability to filter posts especially in a certain range of creation time. eLab propose a similarly straightforward system with the capability to filter posts and the addition of archive saving.

1.4 References

- ISO/IEC 25010:2011 (https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en)
- React Native (https://reactnative.dev/)
- Node.js (https://nodejs.org/en/)
- MySQL (https://www.oracle.com/mysql/)
- Simple Mail Transfer Protocol (https://www.smtp.com/)

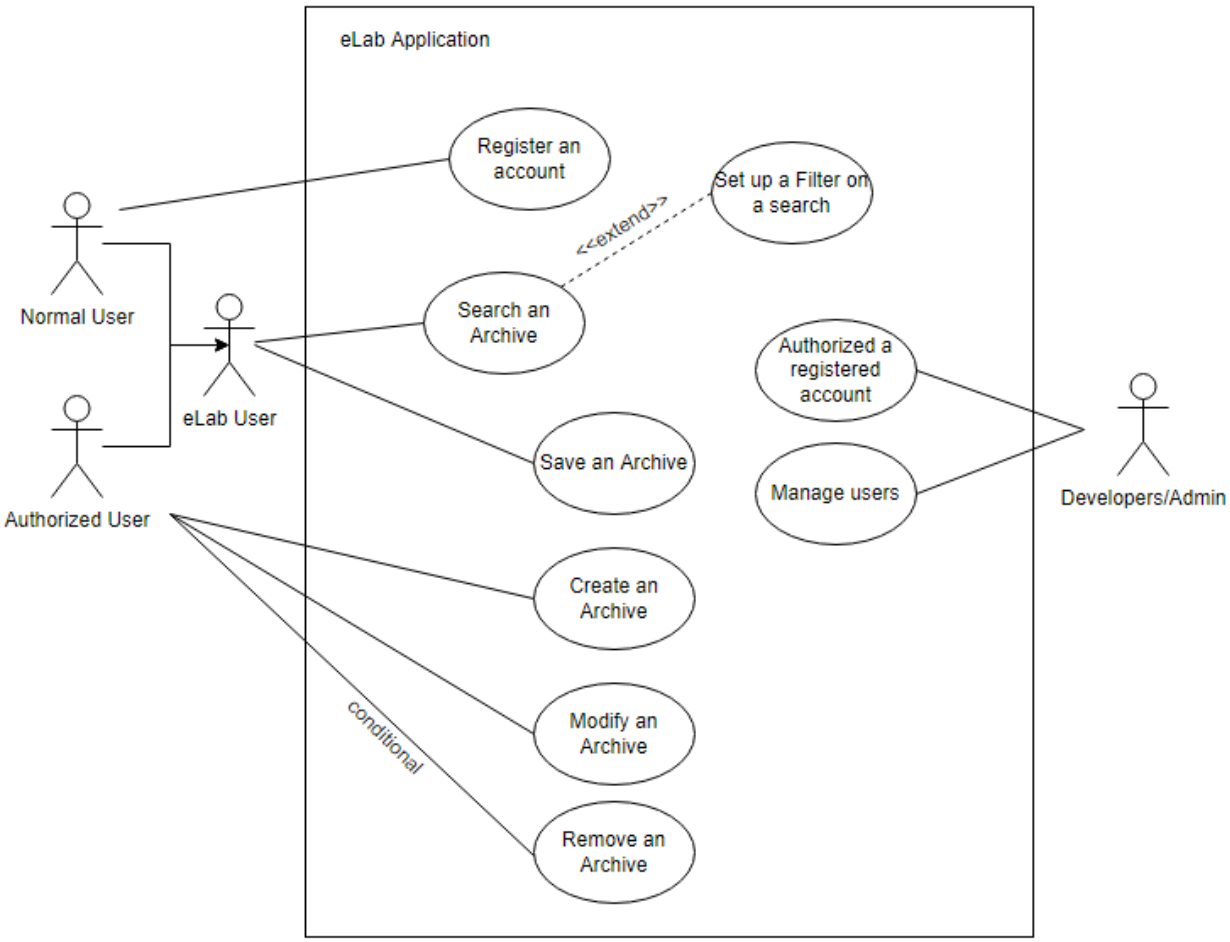
2. Overall Description

2.1 Product Perspective

Since documentation is a very important thing to do, especially for projects and events. archiving data is essential so that we can review them or as an indication that we have done various of projects and activities. But, without a proper and organized system, data from the documentation can be easily lost or mixed in with other data. Additionally, for security reasons, documentation are usually restricted to certain personnel.

ELab is a software hoped to fixed that certain problem. With an organized archiving system, we can upload documentation of projects and activities without any worry that they will be lost or mixed in with other data. And with the open access of the archive with a login of an account, many people can view the documentation without worry of security reasons since there are restriction of what a normal user and an authorized user can do.

2.2 Product Functionality



The diagram above shows the general functionality of the product. Furthermore, for a deeper explanation:

- A user can login to 2 different type of account, a normal user (a guest) or an authorized user
- Users can access and view archive documentations
- Users can set up a filter on their search
- Users can save previously seen archive so users don't have to re-search for them
- An archive post has pictures and descriptions or details about the documentation

2.3 User Class and Characteristics

- Authorized users can use this to archive documentation
  - Create a new archive post (upload pictures, create text description or details about the documentation)
  - Update an existing archived post
  - Delete an existing archived post
- Normal user can use this to view archived documentation
  - Search up an archived document by name
  - Search up an archived document by a certain filtration specification
- Administrator or developers will manage the system
  - Authorize a registered account
  - Manage members

2.4 Operating Environment

Our software is an application-based software focused on the mobile devices, therefore our intended operating environment are:

- Android
- iOS

2.5 Design and Implementation Constraint

The constraint that the system and the user have to face are as the following:

- The user has to have an internet connection
- Only an authorized user can create, modify or delete an archive

2.6 User Documentation

The eLab software will focus on a user-friendly environment in mind, with a little bit of style thrown into it. The software will have a simple yet stylish GUI that are easy to use and easy to learn for the user. In addition, there will be a help section for the user to read, in case they need help operating the software.

2.7 Assumption and Dependencies

The Assumption and Dependencies of the software are as following:

- The user has access to an internet connection
- The user is a student or laboratories member at Telkom University

3. External Interface Requirements

3.1 User Interfaces

The eLab user interface will focus on designing simplicity with a little bit of style. This design is not the final product as there will be more features added in the future.

Figure 3.1: Mockup of the Sign In and Sign Up page screen

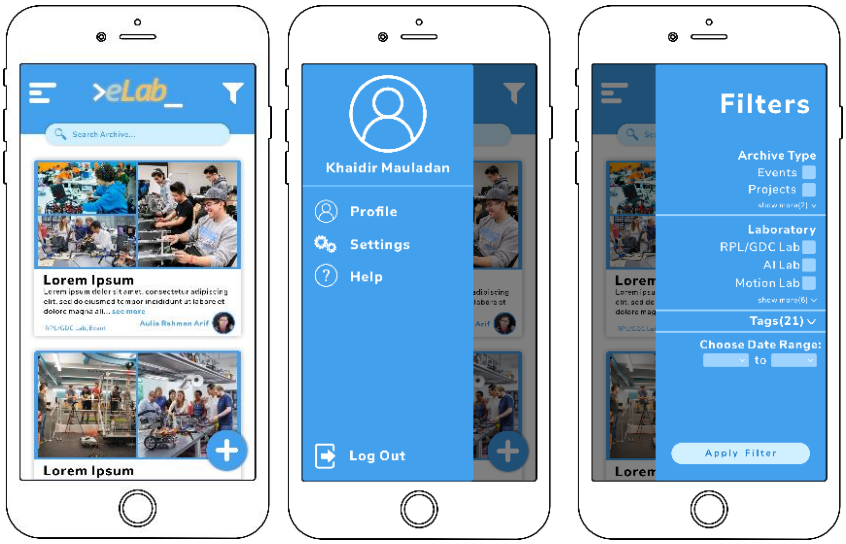


Figure 3.2: Mockup of the Home page screen

3.2 Software Interfaces

For our front-end development, we would be using React Native as it allows app development for both iOS and Android. As for our back-end development, we would be utilizing Node.JS while our servers would run on MySQL.

3.3 Communication Interfaces

The communication interfaces used by this app would include HTTP/HTTPS. An SMTP (Simple Mail Transfer Protocol) would also be utilized to allow activities such as e-mail account integration, verification as well as notification for various features this application would need.

4. System Feature

4.1 Authentication

4.1.1 Description:

The authentication feature consists of sign in and sign up function where users can register and sign in with an account to use eLab or and continuing as guest without an account. The user authorized by the corresponding laboratories must have an account registered previously and must sign in to use eLab, it is to differentiate the normal user and the authorized users.

4.1.2 Trigger:

This authentication feature will automatically be triggered when the user launched the software for the first time or when the account condition is being logged out. The user does not have to sign in every time they launch the software.

4.1.3 Input:

4.1.3.1 Sign in:

To sign in, the user input their email and password to the corresponding bar.

4.1.3.2 Sign Up:

To register an account, the user input their full name, email, and password to the corresponding bar after pressing the register button.

4.1.3.3 Sign in as Guest:

To sign in as guest, the user only needs to press ‘continue as guest’ button

4.1.4 Output:

4.1.4.1 Sign In:

After the email and password has been inputted by the user, the system will verify the input with the server. If they are valid, then the user is given the authorized access to the software by the system.

4.1.4.2 Sign Up:

After the personal data has been inputted by the user, the system will verify whether the email is a previously registered account. If the email is new, the system will create the corresponding account and automatically sign in to eLab.

4.1.4.3 Sign in as Guest:

After pressing ‘continue as guest’ the user goes straight to eLab homepage.

4.1.5 Main Scenario 1:

The user logged in successfully.

4.1.5.1 Pre-condition: The user is on the sign in page

4.1.5.2 Post-condition: The user is given access to the home page

4.1.5.3 Steps:

- Input a valid email in the email bar
- Input a valid password in the password bar
- Press the ‘Log in’ button
- User is moved to the home page screen

4.1.6 **Main Scenario 2:**

The account has been created.

4.1.6.1 Pre-condition: The user is on the sign in page

4.1.6.2 Post-condition: The account creation has been successful

4.1.6.3 Steps:

- User presses the ‘Sign Up’ button
- User input their full name, email, and password
- Press the ‘REGISTER’ button
- Account created and the user is moved to the home page screen

4.1.7 **Main Scenario 3:**

Use eLab as guest.

4.1.7.1 Pre-condition: The user is on the sign in page

4.1.7.2 Post-condition: The user use eLab without an account

4.1.7.3 Steps:

- User presses the ‘continue as guest’ button
- User is moved to the home page screen

4.1.8 **Exponential Scenario 1:**

The user not given access to the software since they either input the wrong email or the wrong password or both.

4.1.8.1 Pre-condition: The user is on the sign in page

4.1.8.2 Post-condition: The user stays on the sign in page, receiving a warning that the account details they had given is wrong

4.1.8.3 Steps:

- Input user’s email in the email bar
- Input user’s password in the password bar
- Press the ‘Log in’ button
- The user is given a warning that the account details they had given is invalid

4.1.9 **Exponential Scenario 2:**

The user is not able to create an account because they are using the same email.

4.1.9.1 Pre-condition: The user is on the register page

4.1.9.2 Post-condition: The user stays on the register page, receiving a warning that the email has already been registered

4.1.9.3 Steps:

- Input user's full name in the full name bar
- Input user's email in the email bar
- Input user's password in the password bar
- The user is given a warning that the email has been used

4.2 Archive CRUD

4.2.1 **Description:**

This CRUD feature is utilized so that only the authorized user can create, modify, or remove an archive documentation of a certain project or event, so that it safe from unauthorized modifications.

Creation, modification, and removal have different method of utilization. To create an archive, an authorized user presses a button on the homepage, while to modify and remove an archive, one must view an archive then the option to modify or remove will be available.

4.2.2 **Trigger:**

4.2.2.1 **Archive Creation:**

The archive creation is triggered when an authorized user pressed the '+' (plus) button to add an archive. The system then will move the user to an archive creation page where the user will fill in the details of the documentation.

4.2.2.2 **Archive Viewing:**

The archive viewing is automatically triggered when a user signed in.

4.2.2.3 **Archive Modifying:**

The archive modifying is triggered when an authorized user pressed the 'edit' button after viewing an archive.

4.2.2.4 **Archive Removal:**

The archive removal is triggered when an authorized user pressed the 'delete' option after viewing an archive.

4.2.3 **Input:**

4.2.3.1 **Archive Creation:**

The authorized user input a description of the documentation in the corresponding text bar and upload a picture or a video (optional).

4.2.3.2 **Archive Viewing:**
The user presses an archive post.

4.2.3.3 **Archive Modifying:**
The authorized user presses the ‘edit’ button on the archive viewing page.

4.2.3.4 **Archive Removal:**
The authorized user presses the ‘delete’ option on the archive viewing page.

4.2.4 Output:

4.2.4.1 **Archive Creation:**
After all the details of the documentation is filled in by user authorized, the system then will upload the documentation to the server where it can be viewed by other users.

4.2.4.2 **Archive Viewing:**
After a user pressed a documentation archive, the archive is viewed in detail.

4.2.4.3 **Archive Modifying:**
After an authorized user modified the description, picture, or video of a documentation archive, the system will update the corresponding archive with the server.

4.2.4.4 **Archive Removal:**
After an authorized user chose the remove option, the archive is removed from the server.

4.2.5 Main Scenario 1:

An authorized user successfully created an archive.

4.2.5.1 Pre-condition: The authorized user is on the home page screen

4.2.5.2 Post-condition: The authorized user created an archive and moved back to the home page screen

4.2.5.3 Steps:
- Authorized user pressed the ‘+’ (plus) button
- The authorized user is moved to archive creation page
- The authorized user fills the description and upload a picture or a video
- The authorized user pressed the ‘create’ button
- The authorized user is moved back to home page screen and an archive is created

4.2.6 Main Scenario 2:

The user views an archive in detail.

4.2.6.1 Pre-condition: The user is in the home page screen

4.2.6.2 Post-condition: The user viewed the archive in detail

4.2.6.3 Steps:
- The user presses an archive to view it
- The user is moved to an archive viewing page
- The archive is viewed in detail

4.2.7 Main Scenario 3:

An authorized user modifies an archive.

4.2.7.1 Pre-condition: The authorized user is on the home page screen

4.2.7.2 Post-condition: The authorized user updated the archive and moved back to the home page screen

4.2.7.3 Steps:
- Authorized user views an archive
- The authorized user is moved to an archive viewing page
- The authorized user chooses to modify the archive
- The authorized user modifies the contents of the archive, then save it
- The archive is updated, and the user is moved back to home page screen

4.2.8 Main Scenario 4:

An authorized user successfully removed an archive.

4.2.8.1 Pre-condition: The authorized user is in the home page screen

4.2.8.2 Post-condition: The authorized user removed an archive

4.2.8.3 Steps:
- Authorized user views an archive
- The authorized user is moved to an archive viewing page
- The authorized user chooses to remove the archive
- The authorized user removed the archive, and the user is moved back to home page screen

4.2.9 Exponential Scenario:

A user fails to create, modify, and remove an archive.

4.2.9.1 Pre-condition: The user is not an authorized user

4.2.9.2 Post-condition: The user cannot create, modify, or remove the archive

4.2.9.3 Steps:
- User is in the home page screen
- There are no '+' (plus) button to create an archive

- User views an existing archive
- There are no options to modify or remove the archive

4.3 Archive Saving

4.3.1 Description:

All user types can save an archive of a laboratories by using the save option available in the archive viewing page for every laboratories archive created. Every archive saved is stored in user's 'saved' page reducing the search time of saved archive.

4.3.2 Trigger:

The archive saving function is triggered when a user is in the archive viewing page of an archive.

4.3.3 Input:

The user presses the 'save' button to save the archive.

4.3.4 Output:

After pressing the 'save' button on an archive viewing page, the user stays on the archive viewing page and the saved archived is stored in user's 'saved' page.

4.3.5 Main Scenario:

A user saved a certain archive.

4.3.5.1 Pre-condition: The user is in the home page screen

4.3.5.2 Post-condition: The user has saved an archive and stays in the archive viewing page

4.3.5.3 Steps:

- User views an archive
- User chooses the 'save' button
- The archive is saved and stored in user's 'save' page
- User stays in archive viewing page

5. Nonfunctional Requirements

5.1 Quality Attributes

1. **Reliability:** The software can be used effectively
2. **Availability:** The software can be accessed anytime as long as it is not being maintained
3. **Security:** The software periodically does system checks of any abnormalities on the software to avoid any security threats. Users that want to use authorized access will have to authenticate their account with the server via log in or sign up
4. **Maintainability:** Software developers periodically does system check whether the software needed any maintaining.

5.2 Legal Requirements

1. **Account:** A user of the software can create account to receive authorization on a restricted feature of the software. However, under certain condition, the assigned administrator of the software can access and/or deactivate the account. User are prohibited to share their account information to other third parties, as the software are not responsible for the use of their account
2. **Privacy Protection**: The software handler will try their best keep the privacy and any protect any copyright law according to the existing terms and constitutions. Privacy policy will explain how the software collects and manage data of the user personal information. By using this software, the user agrees to the terms and condition.

Appendix A: Data Dictionary

- Administrator: Someone who manages a certain department in a whole
- Android: A mobile operating system based on a modified version of the Linux kernel and other open source software
- Archive: A depository containing historical records and document
- Authentication: Validating the authenticity if something or someone
- Authorization: Official permission or approval
- Documentation: Confirmation that some fact or statement is true through the user of documentary evidence
- Event: something that happens at a given place and time
- Filtration: The act of filtering out certain object from whatever they pass through
- GUI (Graphical User Interface): a form of user interface that allows users to interact with electronic devices
- Instagram: An American photo and video sharing social networking
- iOS: A mobile operating system created and developed by Apple inc.
- LINE: a freeware app for instant communications on electronic devices
- Project: a piece of work that is undertaken or attempted