DPPL-xx

SOFTWARE DESIGN DESCRIPTION

eLab

for:

RPLGDC Laboratory

Prepared by:
Aulia Rahman Arif Wahyudi (1301194195)
Putu Budi Sukarya (1301194252)
Khaidir Mauladan (1301192327)
Anggi Rodesa Sasabella (1301193161)

Informatics Study Program

Faculty of Informatics

Jl. Telecommunications 1, Dayeuhkolot Bandung

LIST OF CHANGES

Revision	Description
A	
B	
C	
D	
E	
F	
G	

INDEX DATE	-	A	B	C	D	E	F	G
Written by								
Review by								
Approved by								

List of Changes

Pages	Revised	Pages	Revised

Table of Contents

After the Table of Contents There may be a list of tables and a list of pictures ar

1. Introduction

1.1 Purpose of Document

The purpose of this Software Design Description (SDD) document is to present a detailed description of the application-based archive for Telkom University laboratories known as eLab. This document will help in describing the software specifications with object oriented design.

This SDD document 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 Scope of the Problem

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

1.3 Definitions and Terms

All definitions and abbreviations used in this document and their explanations

1.4 References

- Khaidir M., Aulia R. A., P. Budi S. *"eLab SRS"*.

1.5 Systematic Discussion

This section will describe the general systemic discussion of this SDD Document. Chapter 1 will introduce our app, the eLab. Chapter 2 will then proceed to outline the description of the eLab at large. We will give a detailed elaboration of the design of eLab in chapter 3

This section is a general description of the document. Write down the systematic discussion of this DPPL document.
example: for example, in chapter 1 we talk about what, in chapter 2 we talk about what etc.

2 Description of Global

2.1 Design Implementation Environment Design

eLab	Specification
Operating System	Windows 10
DBMS	MySQL
Development tools	
Filing System	Date, name, type (video, photo, document), …
Bahasa Pemrograman	

Mention the operating system, DBMS, development tools, filing system, programming language used

2.2 . Architectural Description

Give a brief description of the /L architecture to be built. Draw it in the form of a component diagram.

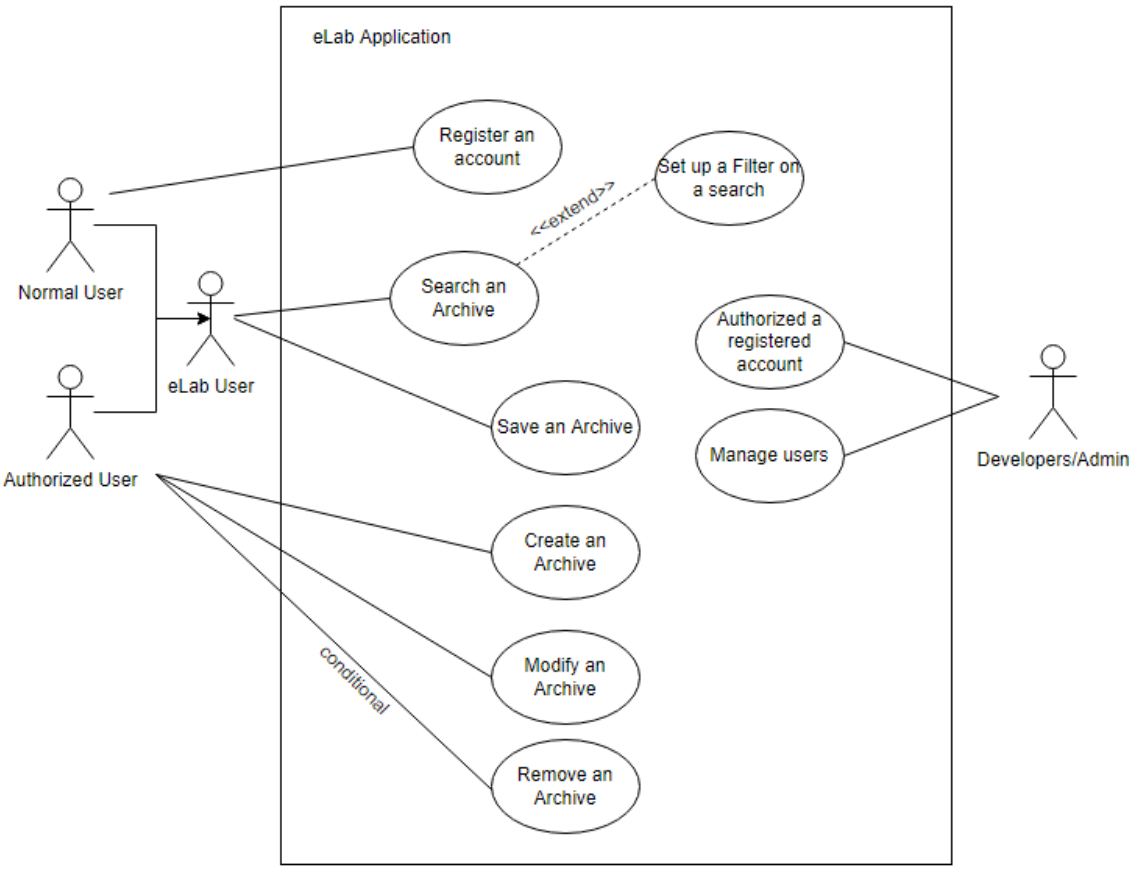
2.3 Component Description

Filled with a list of modules. The list of modules can be in the form of the following table:

No	Component Name	Detailed
1	Admin	Super user focused on user management
2	Normal user	Basic user, can be guest or registered user
3	Authorized user	Authorized by admin. Usually lab assistant
4	Login	Login to use more features
5	Register	Register to open more features
6	Search Archive	Search function for uploaded archives
7	View Archive	View function for searched archives
8	Create Archive	Create function specified for authorized user to create new archives
9	Modify Archive	Modify function specified for authorized user to modify archives
10	Remove Archive	Remove function specified* for authorized users to remove archives. *Can be registered user conditionally
11	Save Archive	Save function for viewed archive available to registered users
12	User Management	Specified for admins to manage authorized users

3 Design

3.1 Realization Use Case



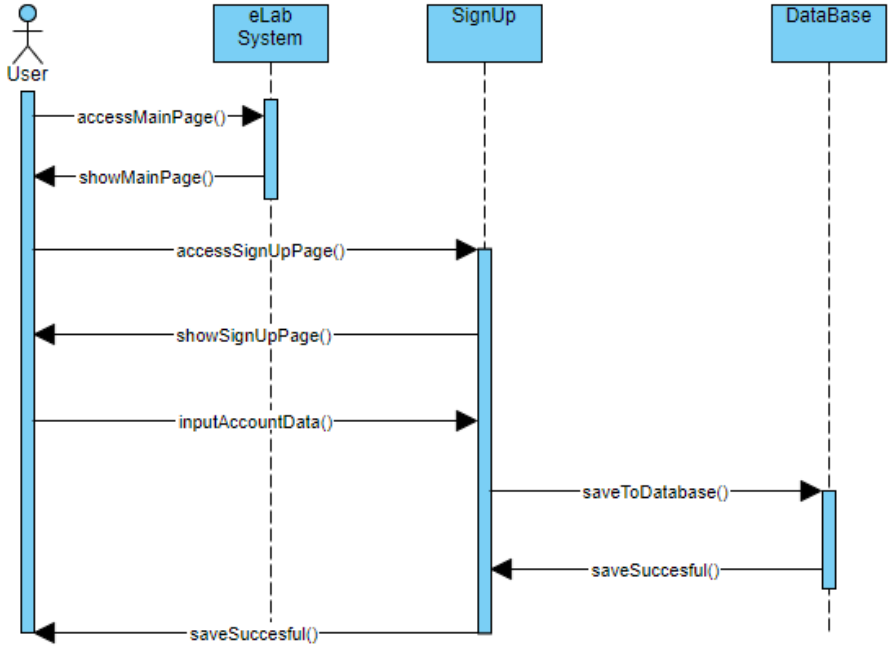
3.1.1 Use Case Register an Account

3.1.1.1 Class

Identification Identify the class associated with the use case. Classes in the design phase may differ from those in the analysis phase. You can use the table below:

No	*Class Name Design Class*	*Type*
1	*User*	*User*
2	*Admin*	*Admin*
3	*Database*	*Database*

3.1.1.2 Sequence Diagram



3.1.1.3 Class Diagram

*Make a class diagram for the use case. create class diagram **NOT ENTIRE, but PER USE CASE***

3.1.2 Use Case Search an Archive

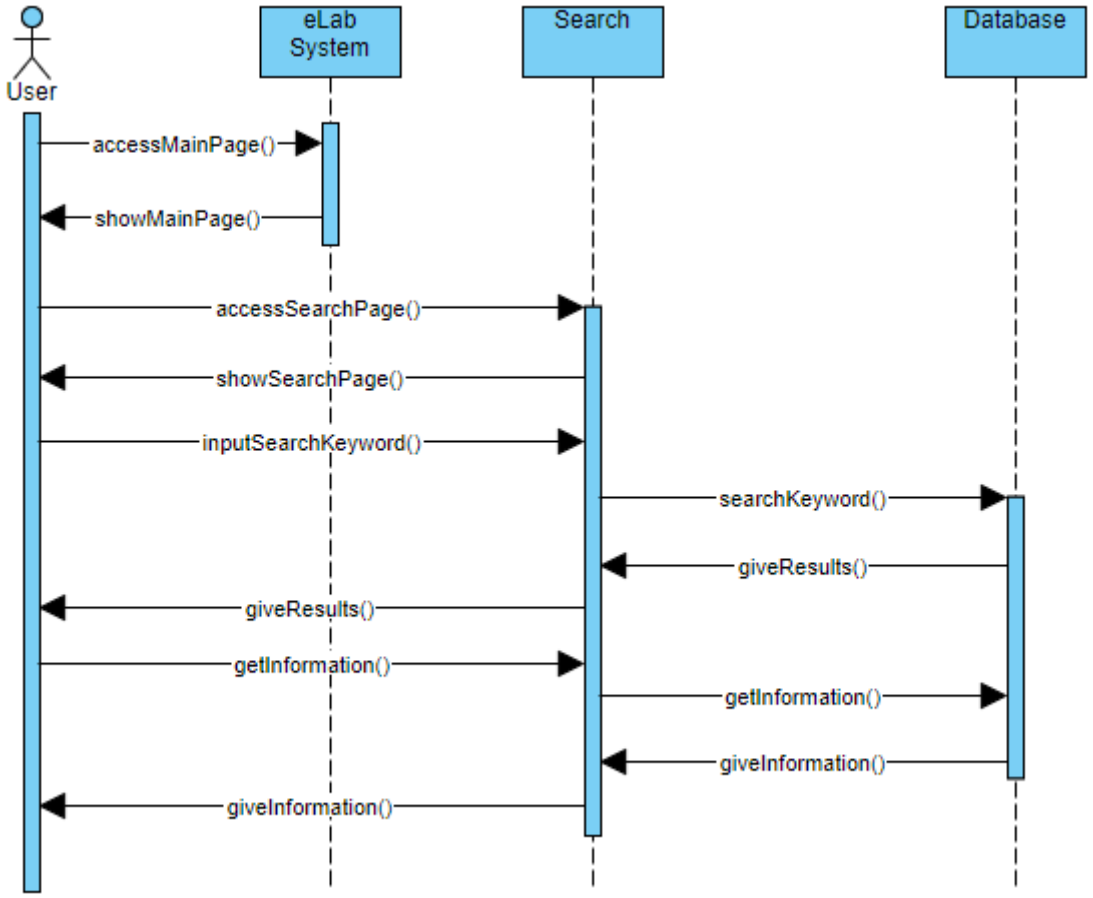
3.1.2.1 Class

Identification Identify the class associated with the use case. Classes in the design phase may differ from those in the analysis phase. You can use the table below:

No	*Class Name Design Class*	*Type*
1	*User*	*User*
2	*Admin*	*Admin*
3	*Database*	*Database*

**Class types such as Boundary(Interface), Entity(Database), Controller*

3.1.2.2 Sequence Diagram



3.1.2.3 Class Diagram

Make a class diagram for the use case. create class diagram ***NOT ENTIRE, but PER USE CASE***

3.1.3 Use Case Save an Archive

If this use case will be realized in the form of a web-based application, then the sub-chapters related to designing web-based application elements must be filled out.

3.1.3.1 Class

Identification Identify the class associated with the use case. Classes in the design phase may differ from those in the analysis phase. You can use the table below:

No	*Class Name Design Class*	*Type*
1	*User*	*User*
2	*Admin*	*Admin*
3	*Database*	*Database*

**Class types such as Boundary(Interface), Entity(Database), Controller*

3.1.3.2 Sequence Diagram

Make **a sequence diagram for each use case scenario**. *The scenario involves the classes that have been identified.*

3.1.3.3 Class Diagram

Make a class diagram for the use case. create class diagram ***NOT ENTIRE, but PER USE CASE***

3.1.4 Use Case Create an Archive

If this use case will be realized in the form of a web-based application, then the sub-chapters related to designing web-based application elements must be filled out.

3.1.4.1 Class

Identification Identify the class associated with the use case. Classes in the design phase may differ from those in the analysis phase. You can use the table below:

No	*Class Name Design Class*	*Type*
1	*User*	*User*
2	*Admin*	*Admin*
3	*Database*	*Database*

**Class types such as Boundary(Interface), Entity(Database), Controller*

3.1.4.2 Sequence Diagram

Make **a sequence diagram for each use case scenario**. *The scenario involves the classes that have been identified.*

3.1.4.3 Class Diagram

Make a class diagram for the use case. create class diagram ***NOT ENTIRE, but PER USE CASE***

3.1.5 Use Case Modify an Archive

If this use case will be realized in the form of a web-based application, then the sub-chapters related to designing web-based application elements must be filled out.

3.1.5.1 Class

Identification Identify the class associated with the use case. Classes in the design phase may differ from those in the analysis phase. You can use the table below:

No	*Class Name Design Class*	*Type*
1	*User*	*User*
2	*Admin*	*Admin*

3	*Database*	*Database*

**Class types such as Boundary(Interface), Entity(Database), Controller*

3.1.5.2 Sequence Diagram

Make **a sequence diagram for each use case scenario**. *The scenario involves the classes that have been identified.*

3.1.5.3 Class Diagram

Make a class diagram for the use case. create class diagram ***NOT ENTIRE, but PER USE CASE***

3.1.6 Use Case Remove an Archive

If this use case will be realized in the form of a web-based application, then the sub-chapters related to designing web-based application elements must be filled out.

3.1.6.1 Class

Identification Identify the class associated with the use case. Classes in the design phase may differ from those in the analysis phase. You can use the table below:

No	*Class Name Design Class*	*Type*
1	*User*	*User*
2	*Admin*	*Admin*
3	*Database*	*Database*

**Class types such as Boundary(Interface), Entity(Database), Controller*

3.1.6.2 Sequence Diagram

Make **a sequence diagram for each use case scenario**. *The scenario involves the classes that have been identified.*

3.1.6.3 Class Diagram

Make a class diagram for the use case. create class diagram ***NOT ENTIRE, but PER USE CASE***

3.1.7 Use Case Authorized a Registered Account

If this use case will be realized in the form of a web-based application, then the sub-chapters related to designing web-based application elements must be filled out.

3.1.7.1 Class

Identification Identify the class associated with the use case. Classes in the design phase may differ from those in the analysis phase. You can use the table below:

No	*Class Name Design Class*	*Type*

1	*User*	*User*
2	*Admin*	*Admin*
3	*Database*	*Database*

**Class types such as Boundary(Interface), Entity(Database), Controller*

3.1.7.2 Sequence Diagram

Make **a sequence diagram for each use case scenario**. *The scenario involves the classes that have been identified.*

3.1.7.3 Class Diagram

Make a class diagram for the use case. create class diagram ***NOT ENTIRE, but PER USE CASE***

3.1.8 Use Case Manage Users

If this use case will be realized in the form of a web-based application, then the sub-chapters related to designing web-based application elements must be filled out.

3.1.8.1 Class

Identification Identify the class associated with the use case. Classes in the design phase may differ from those in the analysis phase. You can use the table below:

No	*Class Name Design Class*	*Type*
1	*User*	*User*
2	*Admin*	*Admin*
3	*Database*	*Database*

**Class types such as Boundary(Interface), Entity(Database), Controller*

3.1.8.2 Sequence Diagram

Make **a sequence diagram for each use case scenario**. *The scenario involves the classes that have been identified.*

3.1.8.3 Class Diagram

Make a class diagram for the use case. create class diagram ***NOT ENTIRE, but PER USE CASE***

3.2 Design Detailed Classes

This section is filled with a list of all classes in the following table:

No	*Design Class*	*Name Related Analysis*

For each class:

- *identify operations (refer to class responsibilities), including visibility- its*
- *attribute identification, including its visibility*

3.2.1 Class <class name>

This section is filled with a list of operations and Create attributes for each class.

Name of Class : ……..

Operation Name	***Visibility (private, public)***	***Description***
Filled with operation signature		
Attribute Name	***Visibility (private, public)***	***Type***
Filled with attribute name		*Write the type according to what is known in the programming language used*

3.2.2 Class <class name>

3.3 Diagram Overall Class

This section is filled with the overall class diagram.

3.4 Algorithms/Query

This section is filled only for the algorithm framework for **methods of a class** *that is considered quite important. Implementation of skeleton code can also be done for classes defined in certain programming languages. You can make sub-chapters per class.*

Example:
Class :
Operation Name :
Algorithm : (Algo-xxx)

{If referring to a specific query, complete the query table below}
Query :

No Query	*Query*	*Description*
Q-xxx		*Write down the function of the query*

3.5 Interface Design

This section is filled with the initial version of the interface prototype .

Next, for each interface/screen, write down the detailed specifications, for example as below:

Interface : {insert no. screen or interface design image number}

Id_Objek	Type	Name	Description
		Filled with the string that appears on the screen	*Filled with an explanation of the system reaction, for example what screen to open, where to link. When it comes to a code that is quite complex, refer to the algorithm described above.*
Button1	*Button*	*OK*	*If clicked, will activate the AlgoXXX Process.*
RTF1	*RTF Box*		*Contents of Text stored in File xxx*

If the object is linked to another File (eg image file, text file), provide the associated file name and brief description in the description column

3.6 Design of Class Persistence Representation

This section is filled with database schema design and its traceability to the entity class. (RELATIONSHIP SCHEME DEVELOPMENT)

4 Traceability Matrix

Mapping use cases with related classes

Requirements	Related Use Cases	Class
FR-01	Register account	eLab user
FR-02	Log in account	eLab user
FR-03	Create an archive	Authorized user
FR-04	View an archive	eLab user
FR-05	Modify an archive	Authorized user
FR-06	Remove an archive	Authorized user
FR-07	Save an archive	eLab user
FR-08	Search an archive	eLab user
FR-09	Manage users	Admin