DPPL-xx

SOFTWARE DESIGN DESCRIPTION

<Inventory Management Software For SE Laboratory >

for:

<User Name>

Prepared by:

Putra Dharma Bangsa – 1301180523,

Naufal Xeela P – 1301194177,

Meizan Arthur Alfianto – 1301194178,

Andi Achmad Adjie – 1301194264

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

This Software Design Description Document is intended to provide an explanation of the software to be built in the form of a general description as well as a detailed and thorough explanation of the specifications of the SE Storage web, as well as to fulfill the tasks of the Software: Design and Implementation course. This document is intended to be used as a reference material for the two parties concerned, between: developers and users. For developers this document is used as a reference in creating the software, while for users this document is used to record all requirements specifications that will be needed later by the user. By making this document, it is hoped that the application development process will run smoothly and become more efficient in time, energy and cost.

1.2 Scope of the Problem

SE Storage is an inventory management software for SE Laboratory. This website is designed to help SE laboratory members to store their work in one place and it can be accessed anywhere anytime. Also there is a borrowing feature which can help members to request and borrow furniture from the SE Laboratory.

1.3 Definitions and Terms

1. User: People who are directly involved with the software.
2. Lab Member: People who can only input the data and borrow furniture.
3. Lab Assistant: People who manage the software.
4. Lab Supervisor: People who supervise the work of the lab assistants.
5. Use Case: A graphical depiction of several actors, and their interactions that introduce a system
6. Use Case Scenario: explanation of use case diagram
7. MySQL: Structured Query Language is a SQL database management system software.
8. HTML: A markup language used to create a web page, display various information in an Internet web browser and simple hypertext formatting written in ASCII format files in order to produce an integrated layout.
9. CSS: The abbreviation for Cascading Style Sheet which functions to adjust the appearance of elements written in a markup language
10. Javascript: Web programming language that is Client Side Programming Languages.
11. PHP: Programming languages that are often embedded in HTML

1.4 References

The reference documents used in writing this document are:

1. Software Design Description document template (SDD)
2. Software Requirements Specification (SKPL)
3. Lecture slides (Pressman)

1.5 Systematic Discussion

This SDD document for Inventory Management Software for SE Laboratory contains a detailed description of software development requirements. Document organization is grouped into four main parts, namely:

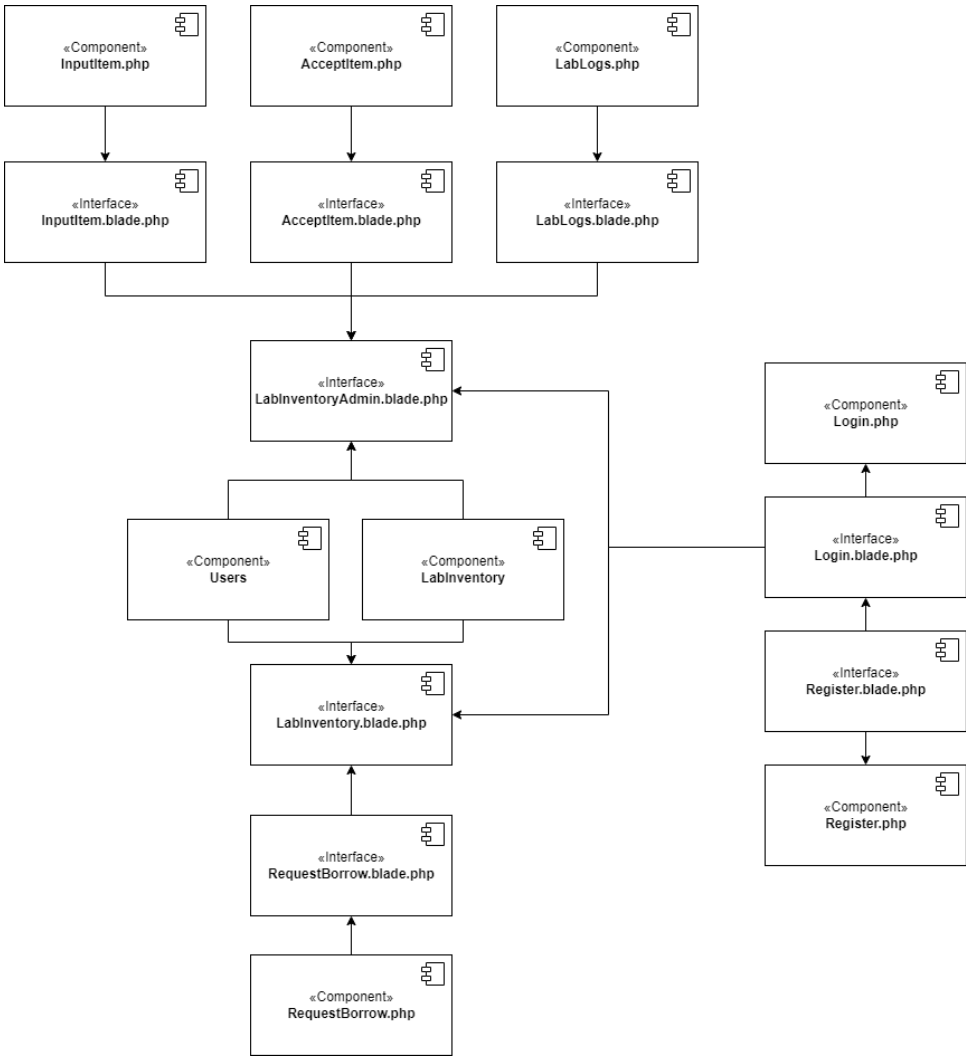
1. Chapter 1 Introduction, contains the purpose of writing the document, the scope of the problem to be handled on the software to be built, and a general description of the document.
2. Chapter 2 Global Description of the Software, contains a description of the software that will be implemented in a global user environment and the architecture of the software.
3. Chapter 3 Design, contains an explanation of the use case realization, class diagram, overall class details, algorithms, interfaces, messages, reports, and class persistence representations.
4. Chapter 4 Traceability Matrix, contains related use cases and related classes in functional requirements.

2 Description of Global

2.1 Design Implementation Environment Design

System	Specification
Operating System	Windows 7, 8, 10
DBMS	MySql
Development Tools	Visual Studio, GitHub, PhpMyAdmin
Programming Language	HTML, CSS, JavaScript, PHP

2.2 Architectural Description



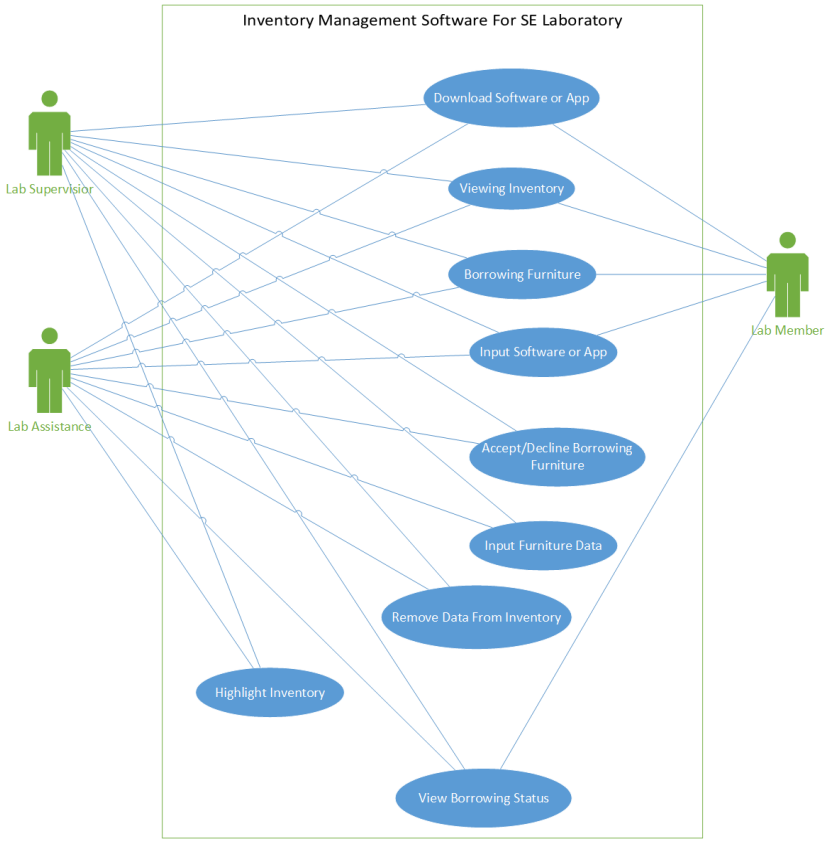
2.3 Component Description

The list of modules can be in the form of the following table:

No	Component Name	Detailed
1	Lab supervisor	Contain lab supervisor entity as well as its additional information
2	Lab assistant	Contain lab assistant entity as well as its additional information
3	Lab member	Contain lab member entity as well as its additional information
4	Lab Inventory Database	Store data related to lab inventory
5	Transaction	Store information about borrowed and returned item
6	Lab member Console	Web page where member can request to borrow an item
7	Lab supervisor and assistant Console	Web page where supervisor and assistant can input and accept item borrow request

3 Design

3.1 Realization Use Case



3.1.1 Use Case <name of use case 1>

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.

3.1.1.1 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	Team Admin	
2	Candidate	
3	Lab Admin	

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

3.1.1.2 Sequence Diagram

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

3.1.1.3 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 Usecases**	**Class**
FR-01		
FR-02		