DPPL-xx

SOFTWARE DESIGN DOCUMENT

QnAI

for:

AI Laboratory - Telkom University

Prepared by:

Rizqi Khoir (1301194266)

Hajarot Najiha (1301194259)

Pernanda Arya Bhagaskara S. M. (1301190184)

Rifaldy Bintang Ramadhan (1301194247)

Informatics Major

Informatics Faculty

Jl. Telekomunikasi 1, Dayeuhkolot, Bandung

Telkom University	**Informatics Study Program** **Telkom University**	**Document Number**		**Page**
		DPPL-xx ***<xx:no grp>***		*<#>/<number #*
		Revision	*<revision number>*	*Date: <fill in date>*

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								
Approve d by								

List of Changes

Pages	Revised	Pages	Revised

Table of Contents

After Table of Contents There may be a list of tables and a list of figures

1. Introductory

1.1 Document Purpose

The purpose of this document is to present a detailed description of the Web Application QnAI (Question and Answer Artificial Intelligence). It will explain the purpose and features of the QnAI, the interfaces of the system, what the system will do, the constraints under which it must operate, and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to all members of the AI Laboratory and society that want to know about AI.

This document is used by software system developers and other interested members. We hope that by making this DPPL document you can ensure that all the features described in the Software Requirements Specification (SKPL) document have been implemented properly.

1.2 Problem Scope

This application will be a web based application and this app will belong to Artificial Intelligence of Telkom University and AI enthusiasts. It is designed to be a forum that will be used for AI Lab and AI enthusiasts to share information around AI tech. We hope this app will help students in the AI lab of Telkom university find resources, study cases, datasets, and information around Artificial intelligence. The complete information about the app will be in 2. Overall description.

1.3 Definition and Term

NO	Terms, Acronyms, and Abbreviations	Explanation
1	SDD	Software Design Description is a representation of a software design used to record and communicate the software design to stakeholders.
2	SRS	A Software Requirements Specification is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality of the product
3	DBMS	Database Management System is a data keeping system that will be used in the designed software.
4	Web Browser	A software to enter the world wide web and can be used to access websites.

1.4 Reference

1. IEEE

IEEE Std 830–1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2. Medium
vincetran-28429.medium.com/software-requirements-specification-srs-document-fd9ab103b18#_Toc77487651
3. krazytech
https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database
4. Requirement Engineering Box
http://isg.inesc-id.pt/REBox/WiegersTemplate@17.aspx?page=4ExternalInterfaceRequirements#:~:text=According%20to%20Richard%20Thayer%20(2002, communicate%20properly%20with%20external%20components.
5. SKPL:https://docs.google.com/document/d/169sAwiSBEIPqcDXDoOG_qiP5e6KYd4Fvlx4ES_RbM90/edit?usp=sharing

1.5 Systematic Discussion

This document contains all descriptions related to QnAI software development. The sections in this document consist of:

1. Chapter 1 contains the purpose of writing and designing DPPL, problem scope, the definition of the terms, acronyms, or abbreviations, as well as the references used to create DPPL.
2. Chapter 2 contains a global design description consisting of architectural descriptions, component descriptions, and implementation environment design.
3. Chapter 3 contains a detailed design of the software consisting of the realization of use cases, user interfaces, and others.

2 Global Planning Description

2.1 Implementation Environment Design

The software is designed and coded through Windows and Linux as its operation system, coded in javascript and HTML. The database will be created in MySql, while the user interface design will use Figma, Laravel, and PHP. The platform will be tested on the newest web browser such as Chrome, Edge, Firefox, and Safari.

2.2 Architectural Description

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

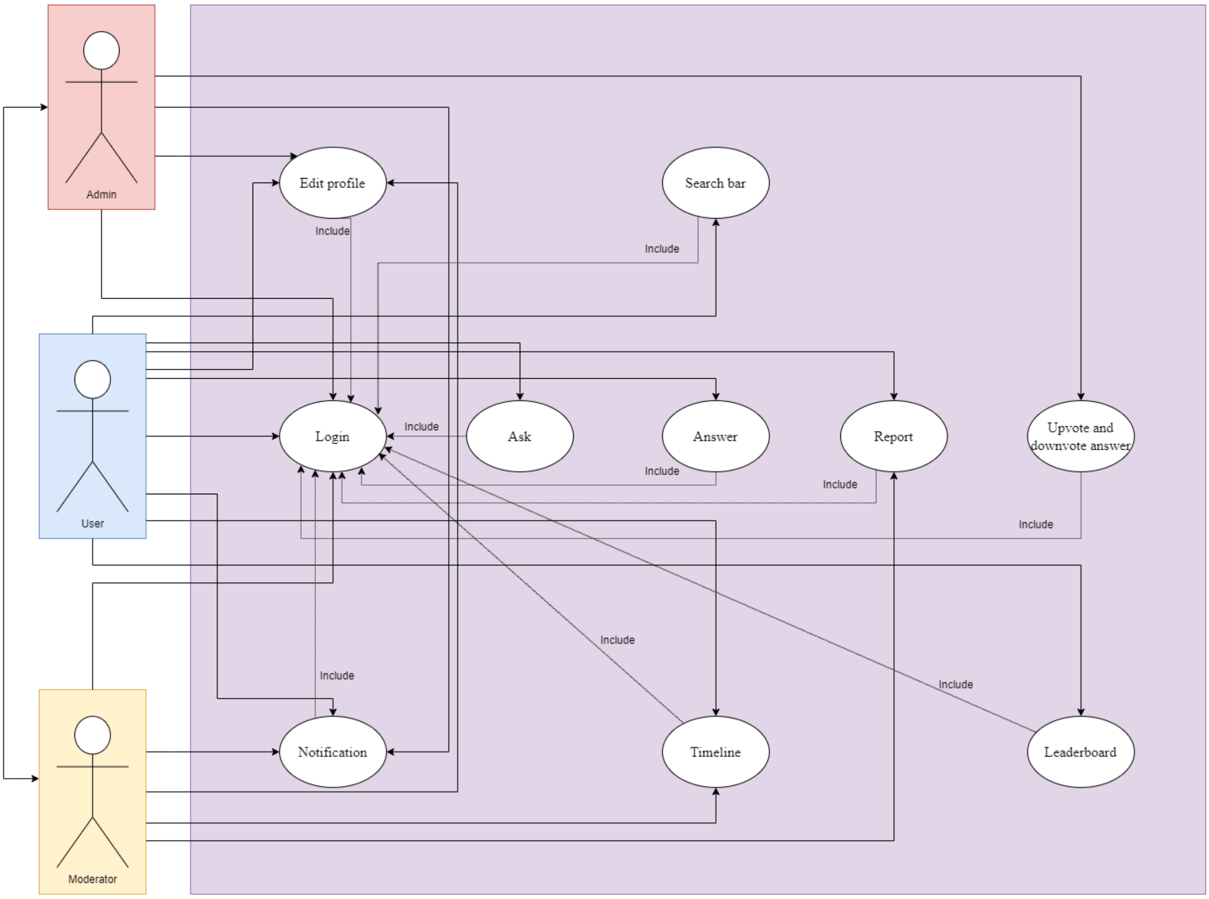
2.3 Component Description

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

No	Component Name	Detailed

3 Design

3.1 Use Case Realization



3.1.1 Login Use Case

This login feature will appear when a new user opens the application for the first time, this feature is on the login page before the user can access all the features on the application.

3.1.1.1 Class

No	Class Name Design	Class Type
1	User	Entity
2	Moderator	Entity
3	Admin	Entity

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***

3.1.2 Ask Use Case

Users may use this feature to ask a question that will appear on the timeline later.

3.1.2.1 Class

No	Class Name Design	Class Type *Class
1	User	Entity
2	Moderator	Entity
3	Question	Controller

3.1.2.2 Sequence Diagram

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

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 Answer Use Case

Users may use this feature to answer a question given on the timeline. They can answer by text, picture, or code. They can also choose whether they would like to answer the question anonymously or not.

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 *Class
1	User	Entity
2	Moderator	Entity
3	Answer	Controller

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 Upvote or Downvote Use Case

This feature lets users upvote or downvote an answer. An answer with the biggest score (upvote - downvote) will appear first. Users can only choose one, whether to upvote or downvote.

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 *Class
1	User	Entity
2	Answer	Controller

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 Report Use Case

This feature is used to report a question or answer that doesn't relate to artificial intelligence or contains harmful words.

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 *Class
1	User	Entity
2	Moderator	Entity
3	Question	Entity
4	Answer	Entity

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 Search Bar Use Case

This feature is used to search for a specific question, answer, topic, or other user profile.

3.1.6.1 Class

No	Class Name Design	Class Type *Class
1	Question	Entity
2	Answer	Entity

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 Timeline Use Case

This feature appears on the home page. It will show other questions'.

3.1.7.1 Class

No	Class Name Design	Class Type *Class
1	Question	Entity

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 Notification Use Case

This feature will notify users whenever there is a notification. The types of notification are: new answer on a question sent, a report sent by the moderator, and others.

3.1.8.1 Class

No	Class Name Design	Class Type *Class
1	User	Entity
2	Question	Controller
3	Answer	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***

3.1.9 Leaderboard Use Case

This feature will display users whose answers get many upvotes in rank format.

3.1.9.1 Class

No	Class Name Design	Class Type *Class
1	User	Entity

3.1.9.2 Sequence Diagram

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

3.1.9.3 Class Diagram

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

3.1.10 Edit Profile Use Case

This feature will let the user edit their profile. Whether that is their name, profile picture, e-mail address, or password. Users must first go to the profile page that contains a history of questions or answers and a setting icon.

3.1.10.1 Class

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

3.1.10.2 Sequence Diagram

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

3.1.10.3 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		