DPPL-xx

SOFTWARE DESIGN DESCRIPTION

ON TIME!2

for:

Software Engineering Laboratory

Prepared by:

Akmal Raafid Taufiqurrahman - 1301192218
Risyad Faisal Hadi - 130194232
Ditya Athallah - 1301194095

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 document is to present a detailed description of our application called “On Time!”. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, and the constraints under which the system must operate. This document is intended to be used as a reference for developing the initial version of the On time! application for the development team

1.2 Scope of the Problem

On time! is a reminder app with a mobile application method. It will allow users to set a reminder of an event and will be notified by a ringtone. So, that the users will not miss any event that has been marked in this application. On time! is a reminder app with a mobile application method. It will allow users to set a reminder of an event and will be notified by a ringtone. So, that the users will not miss any event that has been marked in this application.

1.3 Definitions and Terms

All definitions and abbreviations used in this document and their explanations

1.4 References

OT Documentation referenced by this document, at least SKPL
Books, Guides, other Documentation used in this document (rarely!).
-Insert the SKPL as a reference.

1.5 Systematic Discussion

Our document will describe a software design made with the object-oriented approach. This document will explain all the physical forms of the app Ontime! in detail, including the data's description, a library of the data, a physical decomposition of the module and a concise description of the system interface design. The document is aimed to describe the requirements of the software that was made.

2 Description of Global

2.1 Design Implementation Environment Design

Reminder System	Spesification
Operating System	Windows 10/Windows 11/ Linux
DBMS	MySQL
Development tools	Visual Studio Code and XAMPP
Filing System	Sistem harga, Sistem tanggal, dan Sistem lokasi, sistem kecenderungan dikunjungi
Bahasa Pemrograman	Html, Javascript, ReactJS, and PHP

2.2 Architectural Description

Architectural Description is the result of a set of practices for representing, communicating, and analyzing a software architecture (also known as architecture rendering) and the application of such practices by work products that represent the software architecture.

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.	User	The one whole used the web
2.	Login	Menu to access another menu
3.	Main page	Main menu that user can see everything on it
4.	Create event	Menu to create a new event
5.	See Upcoming Event	Menu to see upcoming event
6.	Customize the web	Menu for user to customize the web
7.	Link to other app	Menu to linked with another app
8.	See Information of the event	Menu for user to see the detailed information of the event

3 Design

3.1 Realization Use Case

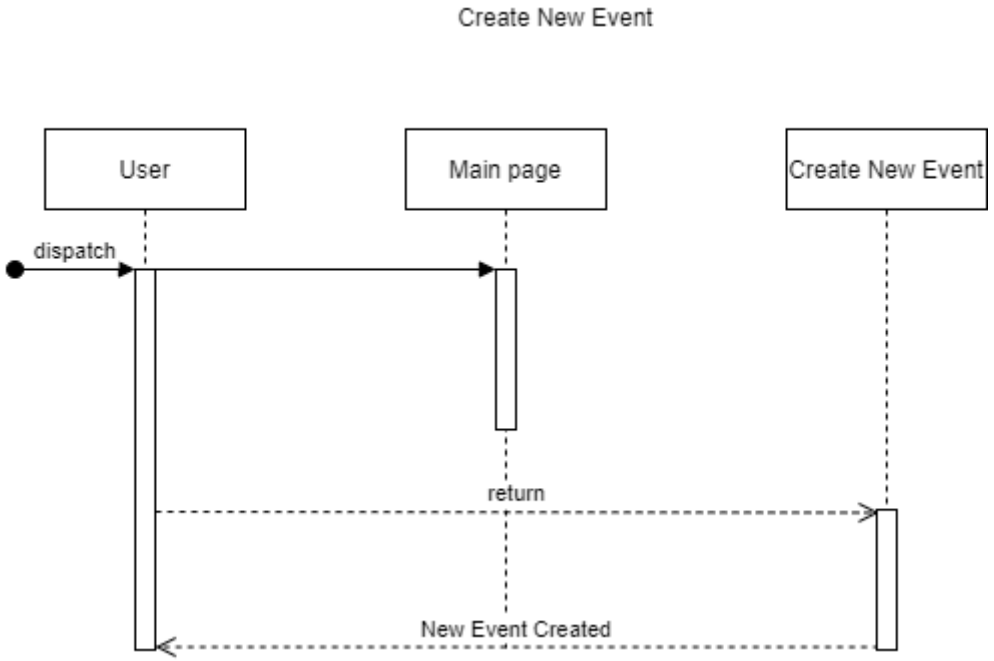
3.1.1 Create New event

For this use case, the user can create a new event for himself manually. So that they can customize what kind of event they want to make.

3.1.1.1 Class Identification

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*Create New event*	*Create New Event*

3.1.1.1.1 Sequence Diagram



3.1.1.1.1.1 Class Diagram

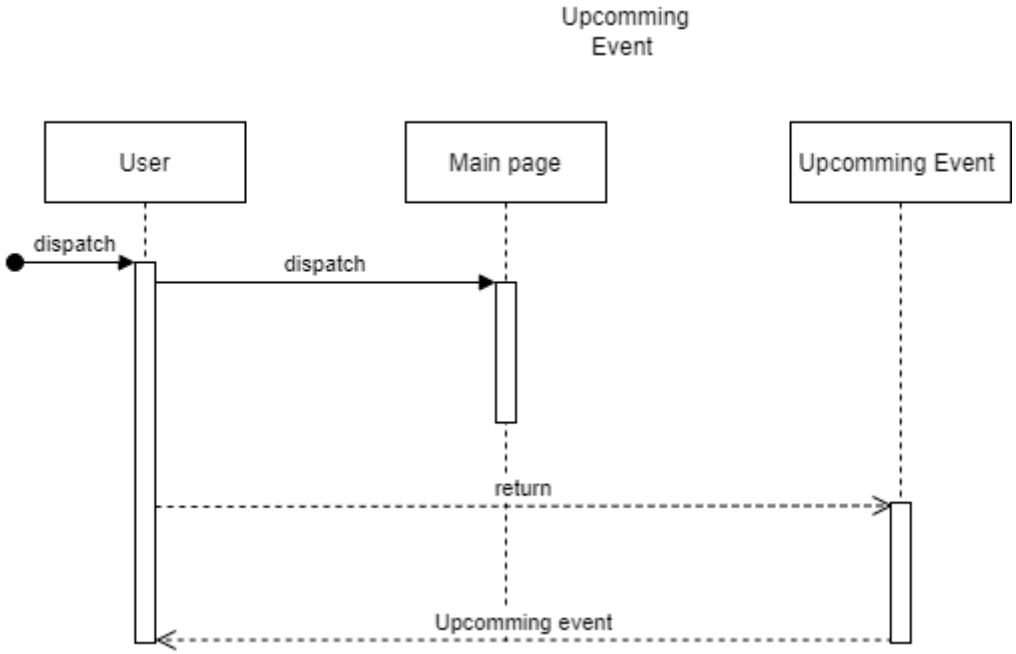
3.1.2 Upcoming Event

For this use case the user can see upcoming events that are made by the user itself or maybe from the event from the connected application.

3.1.2.1 Class Identification

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*Upcoming event*	*Upcoming event*

3.1.2.1.1 Sequence Diagram



3.1.2.1.1.1 Class Diagram

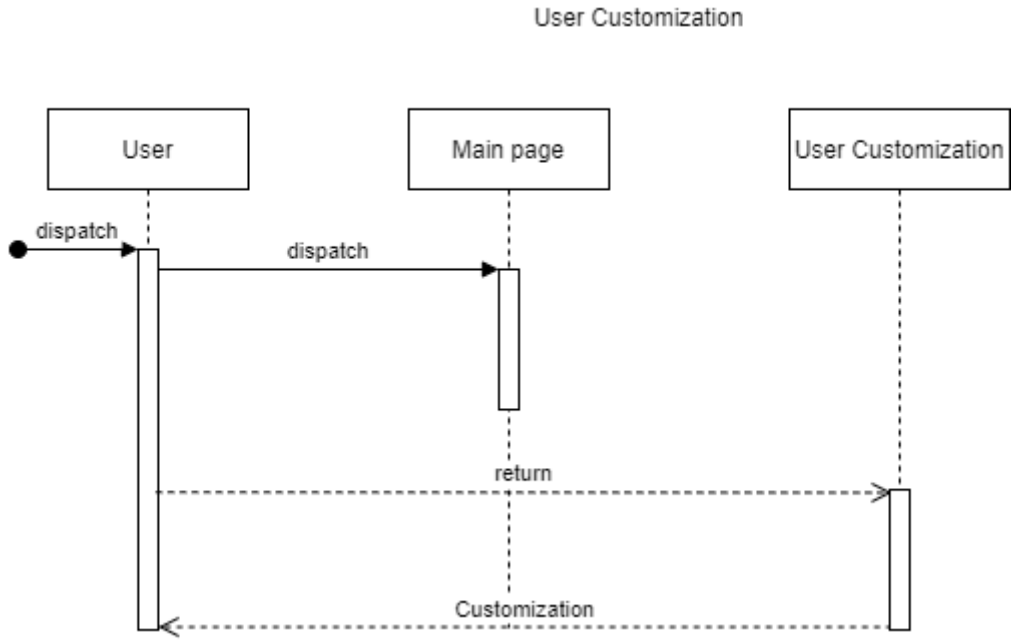
3.1.3 User Customization

Users can customize their reminder, ringtone or theme.

3.1.3.1 Class Identification

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*User Customization*	*User Customization*

3.1.3.1.1 Sequence Diagram



3.1.3.1.1.1 Class Diagram

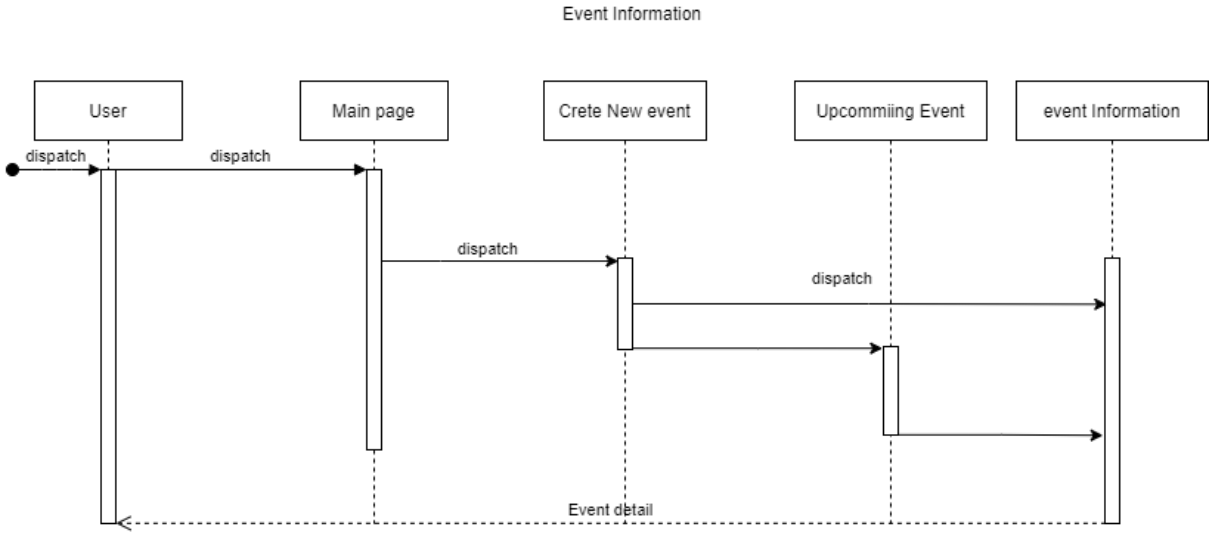
3.1.4 Event Information

In this use case, the user can see the information that has been set, from the name of the event, date, time etc.

3.1.4.1 Class Identification

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*User Customization*	*User Customization*

3.1.4.1.1 Sequence Diagram



3.1.4.1.1.1 Class Diagram

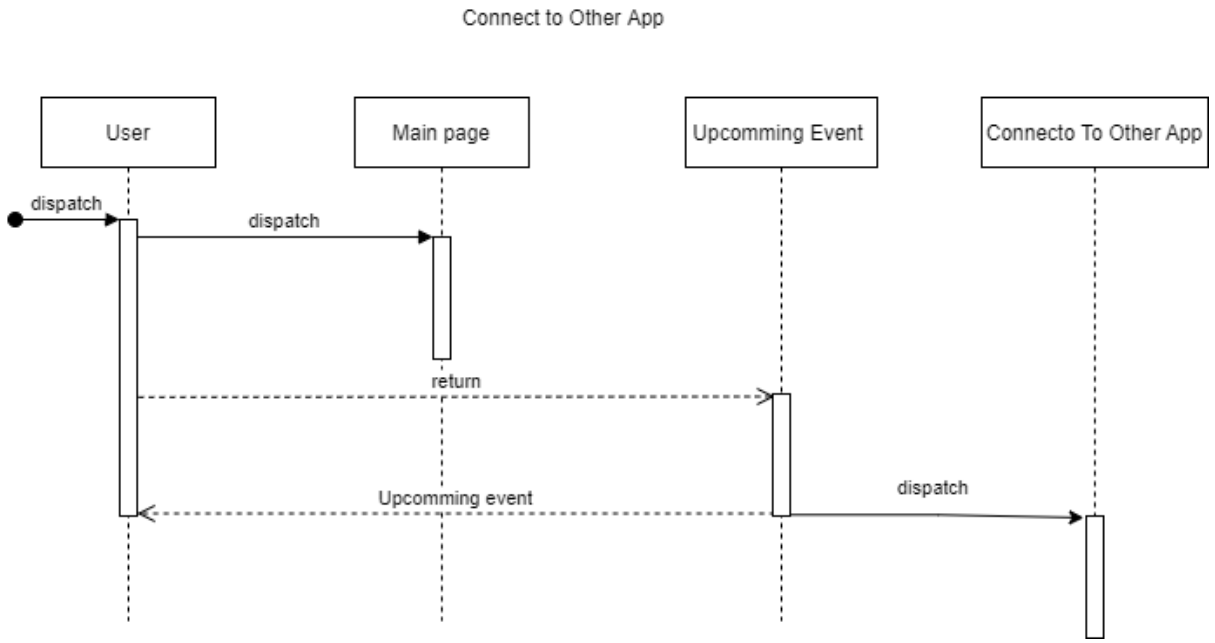
3.1.5 Connect To Other App

From this use case, the user can connect to other apps. For example LMS so that the timeline on LMS can be displayed and remembered in our app.

3.1.5.1 Class Identification

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*Upcoming Event*	*Upcoming Event*
4.	*Connect to Other App*	*Connect to Other App*

3.1.5.1.1 Sequence Diagram



3.1.5.1.1.1 Class Diagram

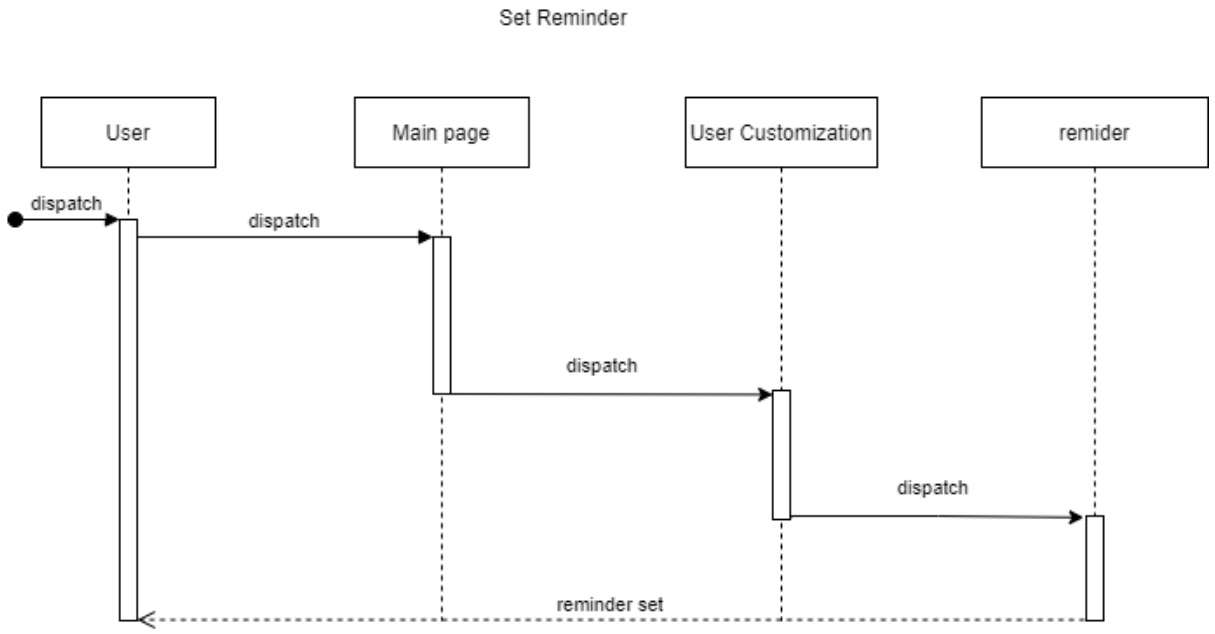
3.1.6 Reminder

The main feature, where the user will be notified if something is on.

3.1.6.1 Class Identification

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*User Customization*	*Upcoming Event*
4.	*reminder*	*reminder*

3.1.6.1.1 Sequence Diagram



3.1.6.1.1.1 Class Diagram

3.1.7 Change Theme

The user can change them to user customization.

3.1.7.1 Class Identification

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*User Customization*	*User Customization*
4.	*Change theme*	*Change theme*

3.1.7.1.1 Sequence Diagram

3.1.7.1.1.1 Class Diagram

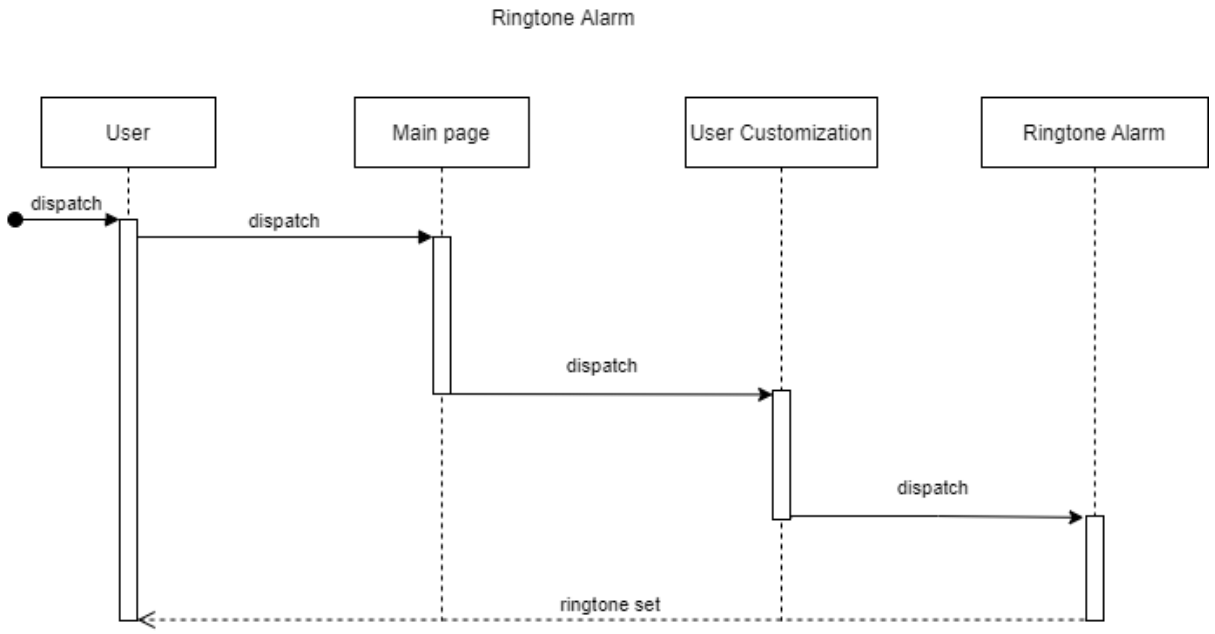
3.1.8 Ringtone Alarm

User can set ringtone

3.1.8.1 Class Identification

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*Upcoming Event*	*Upcoming Event*
4.	*Connect to Other App*	*Connect to Other App*

3.1.8.1.1 Sequence Diagram



3.1.8.1.1.1 Class Diagram

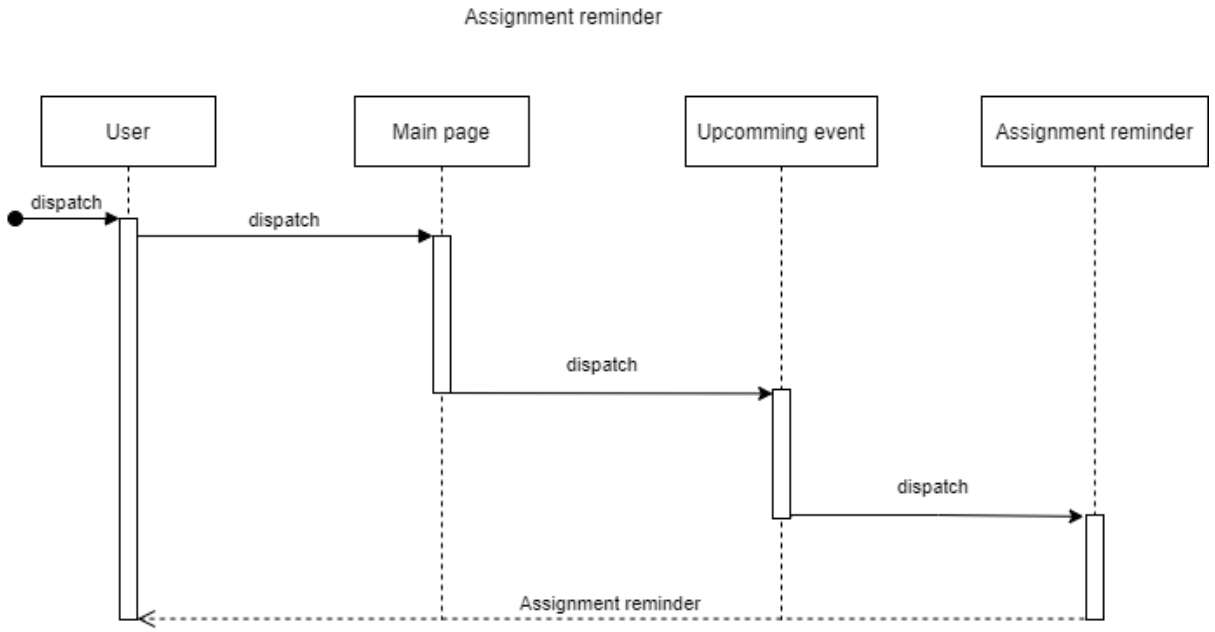
3.1.9 Assignment Reminder

For this use case the system will remind you for the assignment that has been assigned.

3.1.9.1 *Class Identification*

No	*Class Name*	*Detailed*
1.	*User*	*User*
2.	*Main page*	*Main page*
3.	*Upcoming Event*	*Upcoming Event*
4.	*Assignment reminder*	*Assignment reminder*

3.1.9.2 Sequence Diagram



3.1.9.3 Class Diagram

3.2 Design Detailed Classes

No	*Design Class*	*Name Related Analysis*
1.	*main*	*main*
2.	*Login*	*Login*
3.	*Create New Event*	*Create new Event*
4.	*Event Information*	*Event Information*
5.	*Upcoming Event*	*Upcoming Event*
6.	*User Customization*	*User Customization*
7.	*Assignment Reminder*	*Assignment Reminder*

3.2.1 Event Information

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

Name of Class : Main

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 : Login Page

Gambar

Id Objek	Type	Name	Description
Button1	*Button*	*OK*	*If clicked, User will be log in using gmail*
Button2	*Button*	*OK*	*If clicked, User will be login using igracias*

Interface : Main Page

Gambar

Id_Objek	Type	Name	Description
Button3	*Button*	*OK*	*If clicked, User will be directed to a assignment submission*
Button2	*Button*	*OK*	*If clicked, User can see other option on the side bar*

Interface : Detail list pag

Gambar

Id_Objek	Type	Name	Description
Button1	*Button*	*OK*	*If clicked, User will be log in using gmail*
Button2	*Button*	*OK*	*If clicked, User will be login using igracias*

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		