DPPL-Remindee

SOFTWARE DESIGN DOCUMENTS

Remindee

For:

Online Based Reminder System Software

Prepared by:

Naufal Edy Purwantono Scudetto (1301190197)

Muhammad Farhan Akbar (1301192246)

Muhammad Furqon Fahlevi (1301194214)

Hilal Ramadhan Utomo (1301194236

Informatics Study Program

Faculty of Informatics

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

LIST OF CHANGES

Revision	Description
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Written by								
Marked by								
Approved by								

LIST OF PAGE CHANGES

Page	Revision	Page	Revision

Table of Contents

1. Introduction

1.1 Document Writing Purpose

This document contains an explanation of the use and writing of a Software Design Description (DPPL) document with an Object-oriented design approach. This document will further use the term DPPL.

The description set forth in this document is used as a reference in writing the DPPL. This document was created to assist in making software specifications that will be developed with object-oriented design. In principle, the results of the analysis of software systems with this design are described as a set of processes that are organized hierarchically. These processes communicate with each other through a data flow path. This SKPL document was written to fulfil the assignment of the Software Design Analysis course.

This SKPL document contains software specifications regarding the owner of buying an online-based vacation package. The users of the software are the general public, with the aim of simplifying the process of buying and selling holiday packages from the conventional system to a computerized system. So that the owner's process of buying vacation packages can take place more effectively and easily.

1.2 Problem Scope

The problem scope with this SRS is:

1. This website is called "Remindee"
2. This software can create reminders, creating task lists, displaying a timer, notifies if there is a deadline coming up, calculating whether the user is doing their reminders and using AI to determine whether they are productive or not.
3. The target audience for this software is people in general who like to have reminders on what they are going through in life, but specifically for students who are more likely to encounter tasks and homework.

1.3 Terms and Definition

1. Admin

Admin is an object where the process of upgrading an account takes place, and also to moderate the upgrade requests coming in.

2. User

User is an object that signs up, has control of the account, uses the features provided by the application such as creating to d list, creating reminders, etc.

3. DPPL

A software representation a representation of a software design that is to be used for recording design information, addressing various design concerns, and communicating that information to the design's stakeholders.

4. SKPL

Is a specification of the software that will be developed.

5. DBMS

The Database Management System is an organizing system and a data maintenance system for the application

6. HTML

The Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser.

7. JavaScript

A Programming Language that is object based and is made up of components named "class". These classes are composed of methods that do work and returns the information related.

1.4 Reference

This document refers to the results of interviews and interviews and observations of owners of information related to various needs that include general data. The writing of this document is based on:

1. Jurnal Khatulistiwa Informatika, Vol.IV, No.2 Desember 2016 (Rancang Bangung Sistem Informasi Permintaan ATK Berbasis Intranet), Oleh Eka Wida Fridayanthie dan Tias Mahdiati.
2. Jurnal Spesifikasi Kebutuhan Perangkat Lunak (SKPL) (Sistem Informasi Ujian Harian SMK Negeri 2 Singosari), Oleh Lugas Anegah Bahalwan, Universitas Negeri Malang.
3. Jurnal TIKAR Volume 1. No. 2, Juli 2020 (Perancangan Sistem Aplikasi Pemesanan Makanan dan Minuman pada Cafetaria No Caffe di Tanjung Balai Karimun Menggunakan Bahasa Pemrograman PHP dan MYSQL), Oleh T.Bayu Kurniawan, Alumni dan Syarifuddin, Dosen Prodi Teknik Informatika, Universitas Karimun.

1.5 Systematic Discussion

This document contains a description of the software design that is made using an object-oriented approach. This document will explain the design description of the system, including data description, data dictionary, physical decomposition of the module, detailed description of the module and interface design of this system. This document describes all the physical forms of the application journey so that users can understand about the My Vacation application. So that a description is obtained according to the needs of the software made.

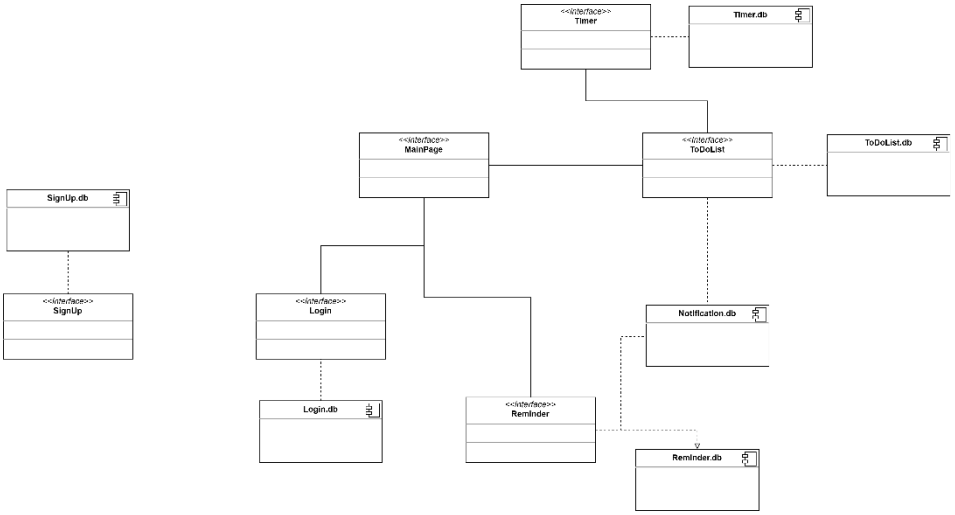
2 Description of Global Design

2.1 Implementation Environment Design

TIKET LIBURAN ONLINE	SPESIFIKASI
Browser	Any browser
DBMS	MySQL
Development tools	Visualparadigm online, draw.io, Figma, Visual Studio Code, Tailwind, NextJS, Postman, phpMyAdmin
Programming Language	Html, CSS, JavaScript, TypeScript

2.2 Architectural Description

This is the architectural description of Remindee application:



2.3 Component Description

No	Components	Explanation
1.	Login	Menu to login to gain access to more features
2.	Admin	Moderator of the website
3.	Free User	User of the website
4.	Premium User	User of the website
5.	Create Task List	Menu to create task lists
6.	Set Up Reminders	Menu to create reminders
7.	Unlimited Reminders	Feature to create unlimited reminders
8.	Notifications	Feature that notifies the user
9.	Upgrade	Menu to upgrade the account
10.	Accepting Upgrade	Menu to moderate accounts not upgraded yet
11.	Statistics	Menu to look at the statistics when using the website
12.	Timer	Menu to display a running timer

3 Detailed Design

In the detailed design chapter, it will be explained about use case realization, detailed class design, overall class diagram, algorithm/query, state chart diagram, interface design, and class persistence representation design.

3.1 Use Case Realization

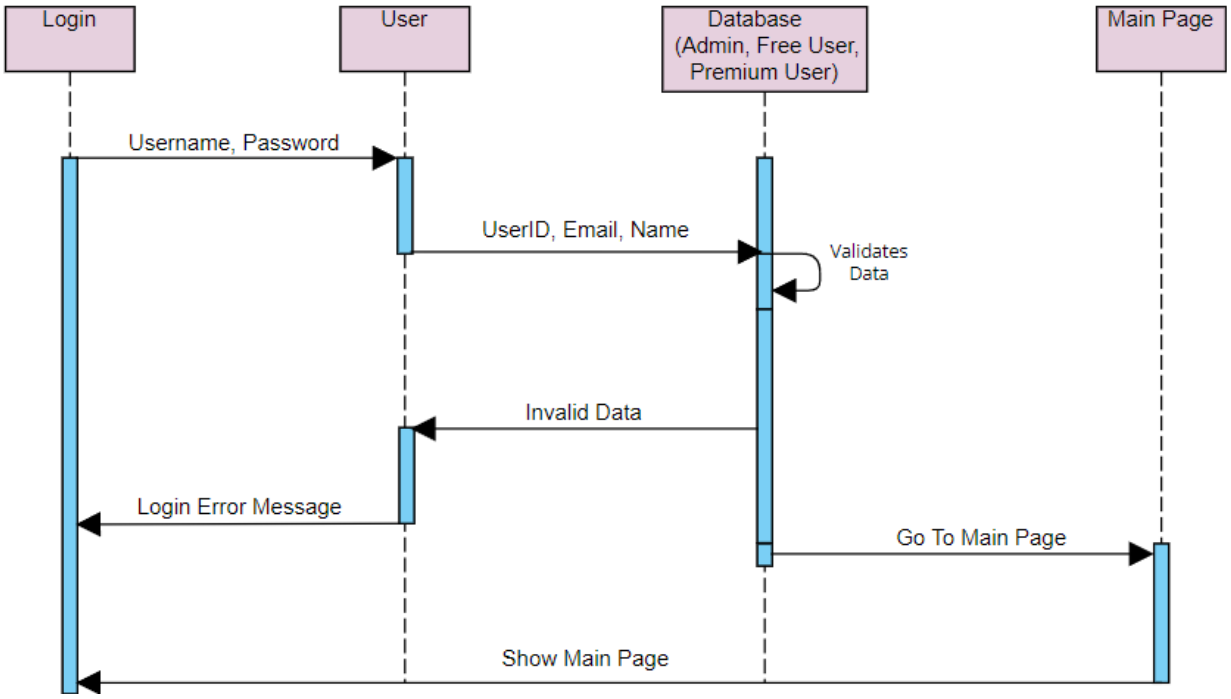
This section describes the realization of all use cases that have been designed in the software requirements specification document.

3.1.1 Use Case Login

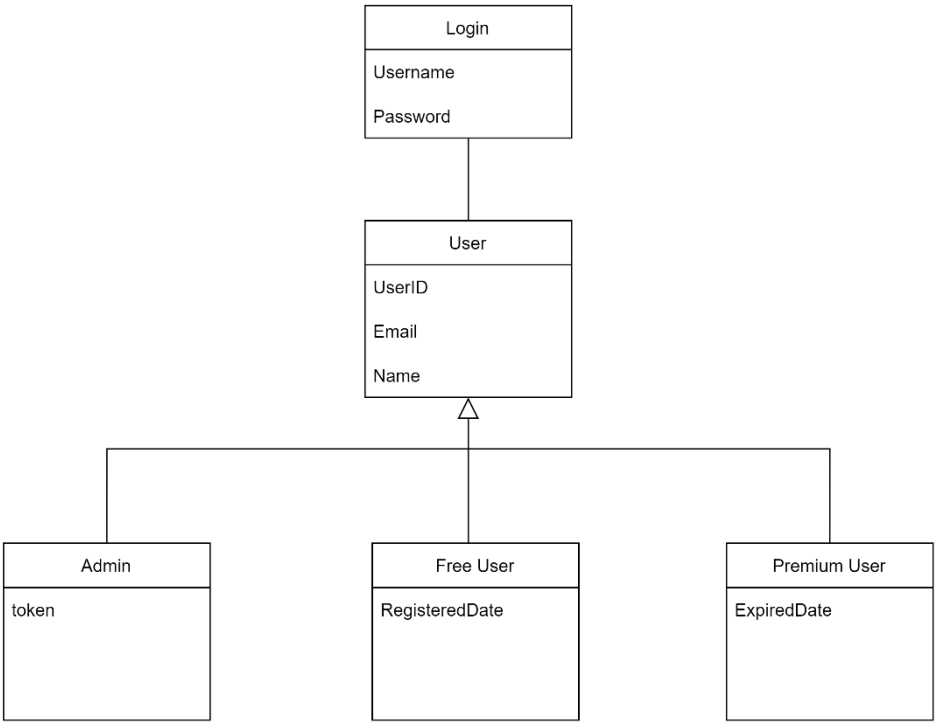
3.1.1.1 Class Identification

No	*Class Design Name*	*Class Type*
1	*Login*	*Login*
2	*User*	*User*
3	*Admin*	*Admin*
4	*Free User*	*Free User*
5	*Premium User*	*Premium User*

3.1.1.2 Sequence Diagram



3.1.1.3 Class Diagram

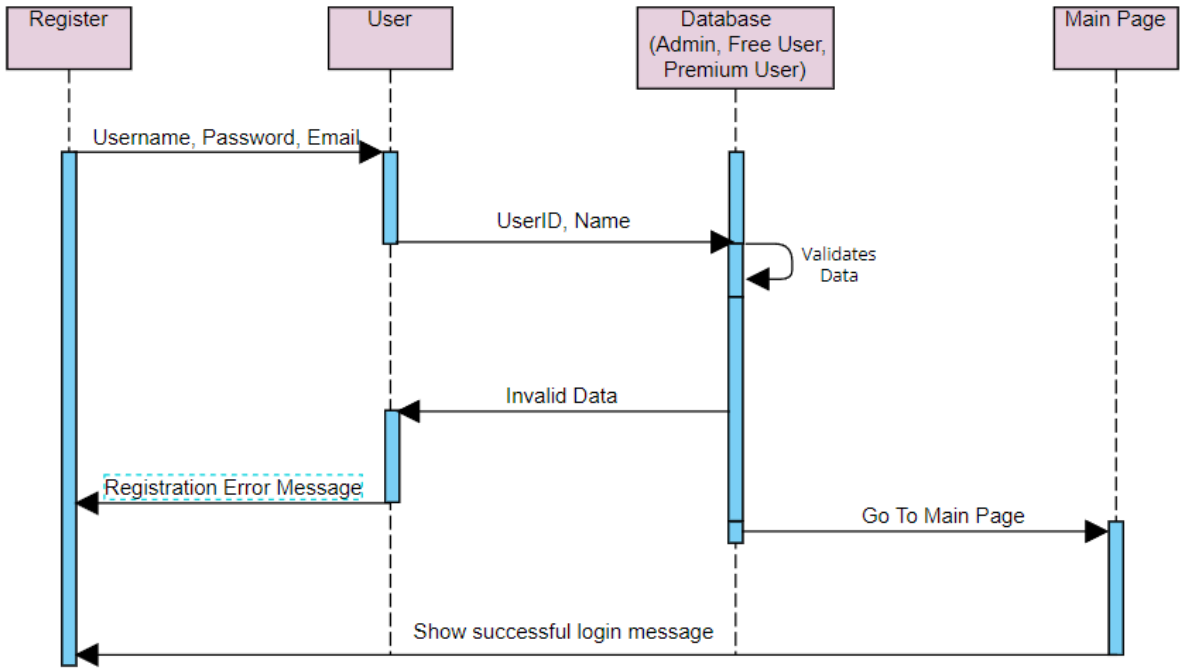


3.1.2 Use Case Sign Up

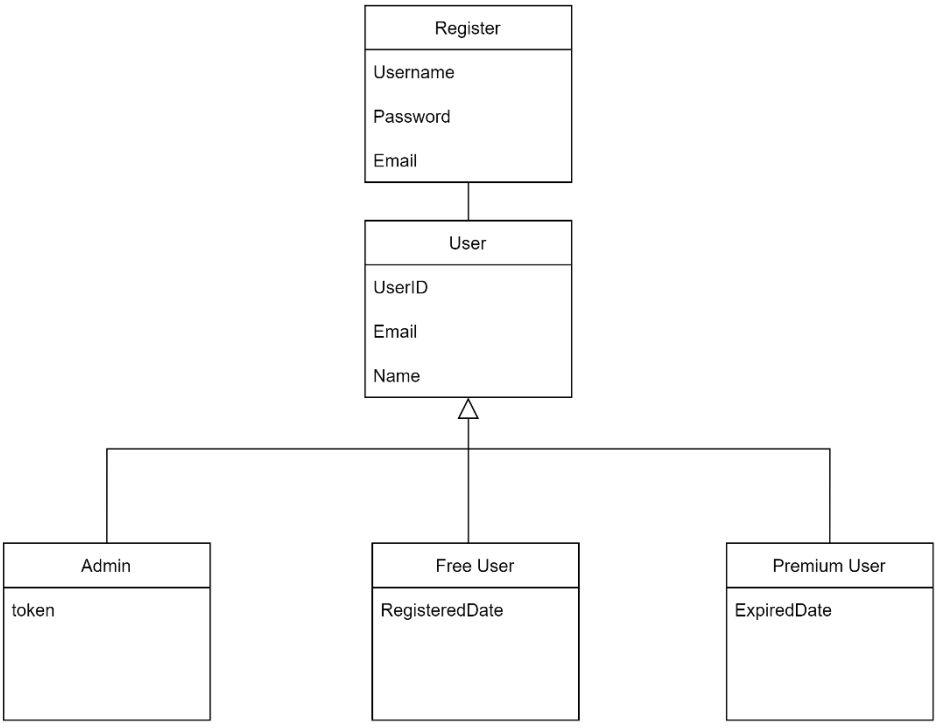
3.1.2.1 Class Identification

No	*Class Design Name*	*Class Type*
1	*Register*	*Register*
2	*User*	*User*
3	*Admin*	*Admin*
4	*Free User*	*Free User*
5	*Premium User*	*Premium User*

3.1.2.2 Sequence Diagram



3.1.2.3 Class Diagram



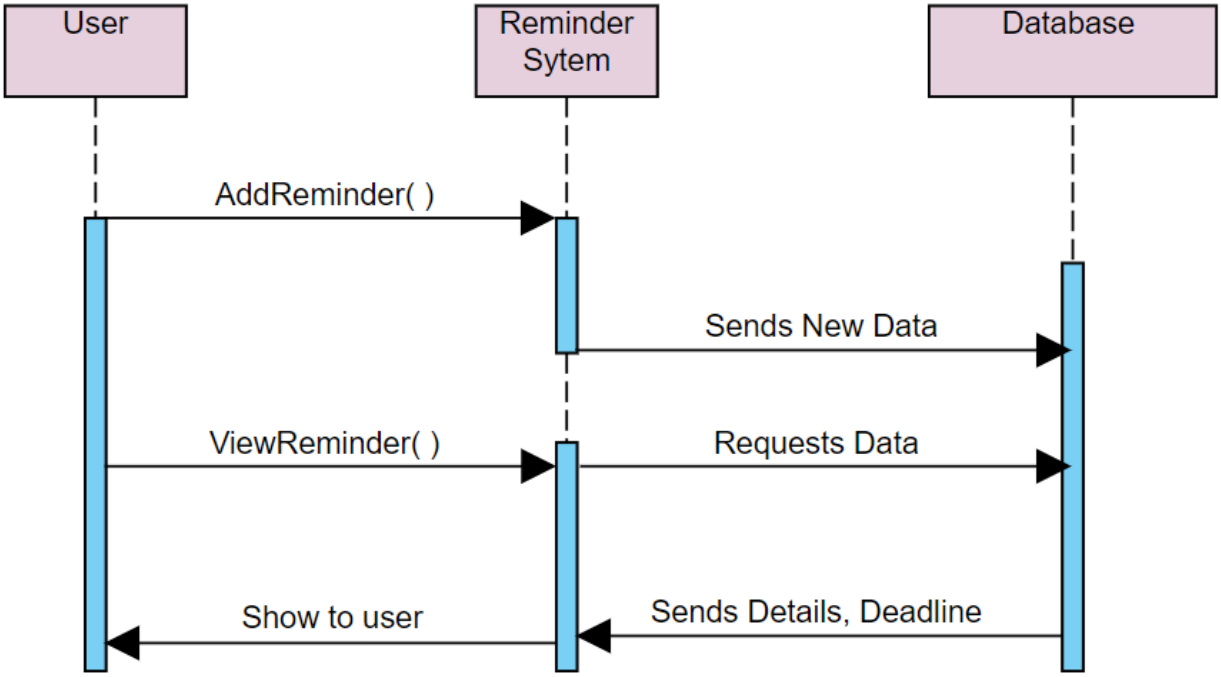
3.1.3 Use Case Set Up Reminder

3.1.3.1 Class Identification

No	*Class Design Name*	*Class Type*

1	*User*	*User*
2	*ReminderSystem*	*ReminderSystem*

3.1.3.2 Sequence Diagram



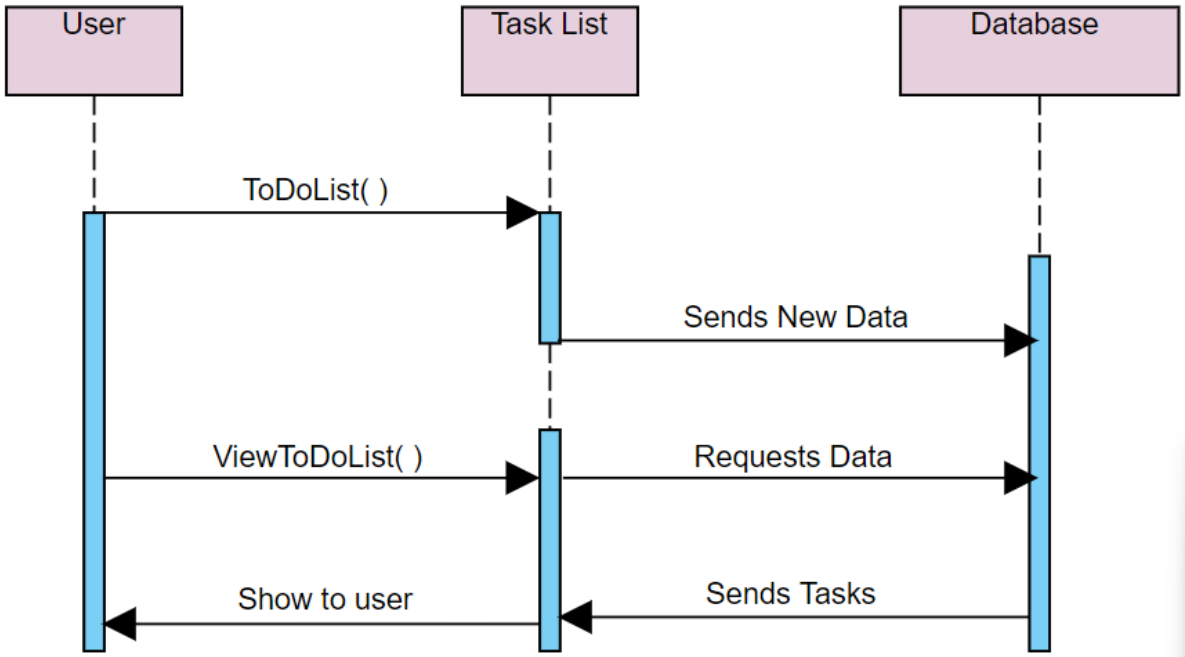
3.1.3.3 Class Diagram

3.1.4 Use Case Create Task List

3.1.4.1 Class Identification

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

3.1.4.2 Sequence Diagram



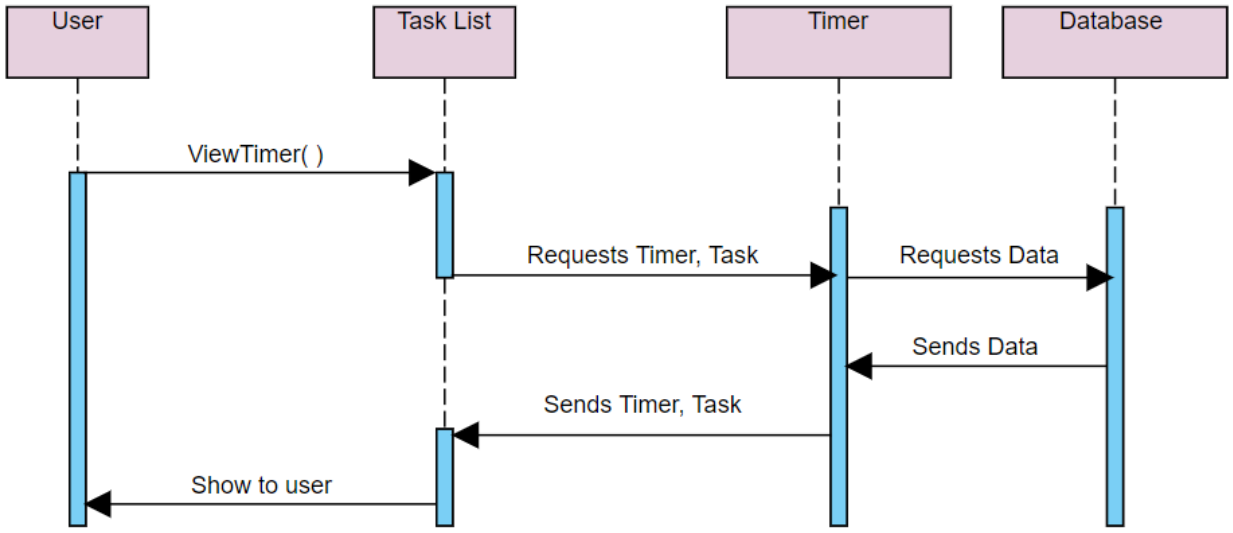
3.1.4.3 Class Diagram

3.1.5 Use Case Timer

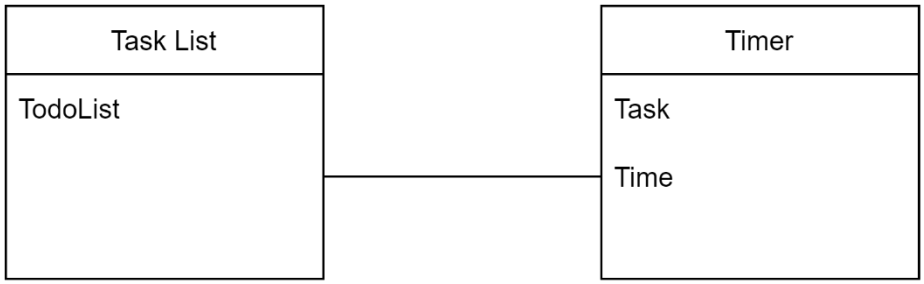
3.1.5.1 Class Identification

No	*Class Design Name*	*Class Type*
1	*Timer*	*Timer*
2	*TaskList*	*TaskList*

3.1.5.2 Sequence Diagram



3.1.5.3 Class Diagram

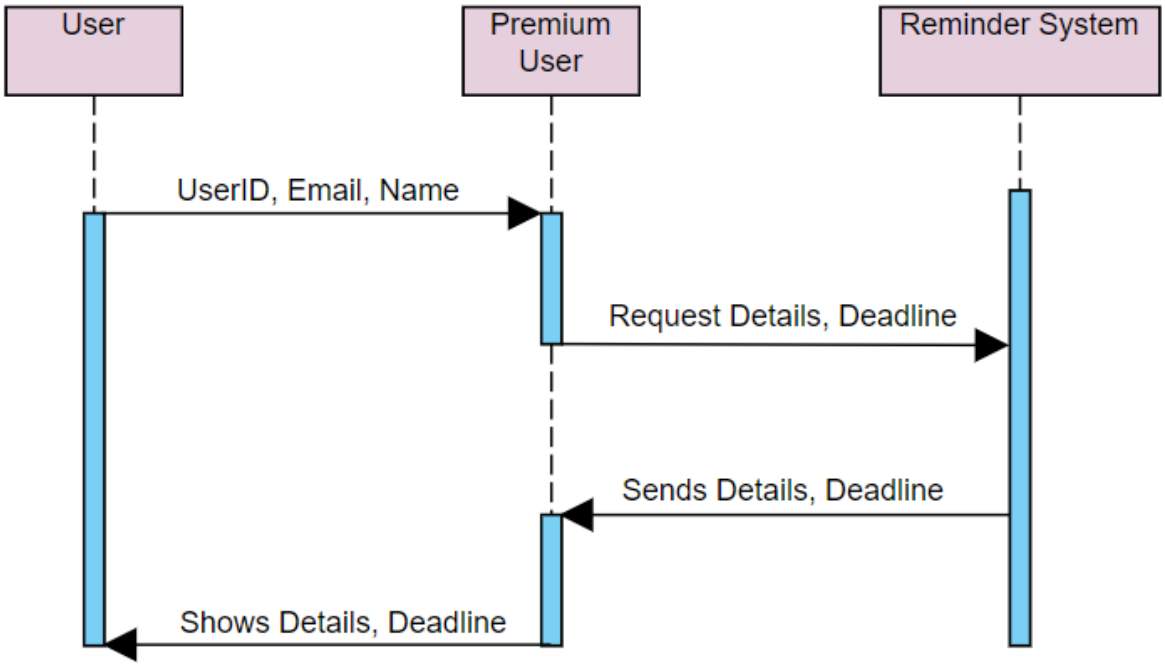


3.1.6 Use Case Unlimited Reminders

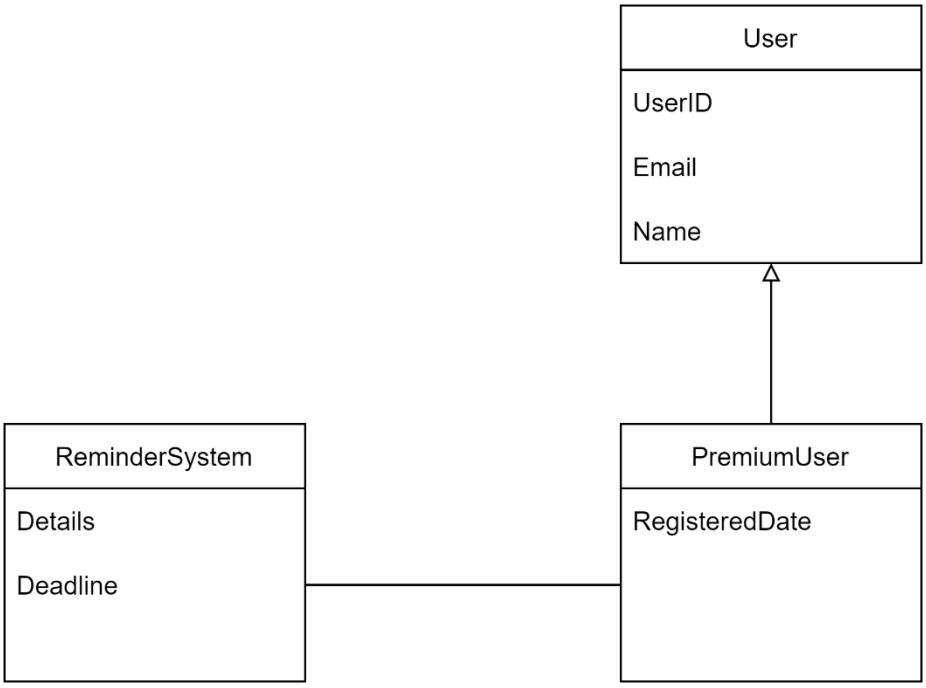
3.1.6.1 Class Identification

No	*Class Design Name*	*Class Type*
1	*User*	*User*
2	*ReminderSystem*	*ReminderSystem*
3	*PremiumUser*	*PremiumUser*

3.1.6.2 Sequence Diagram



3.1.6.3 Class Diagram



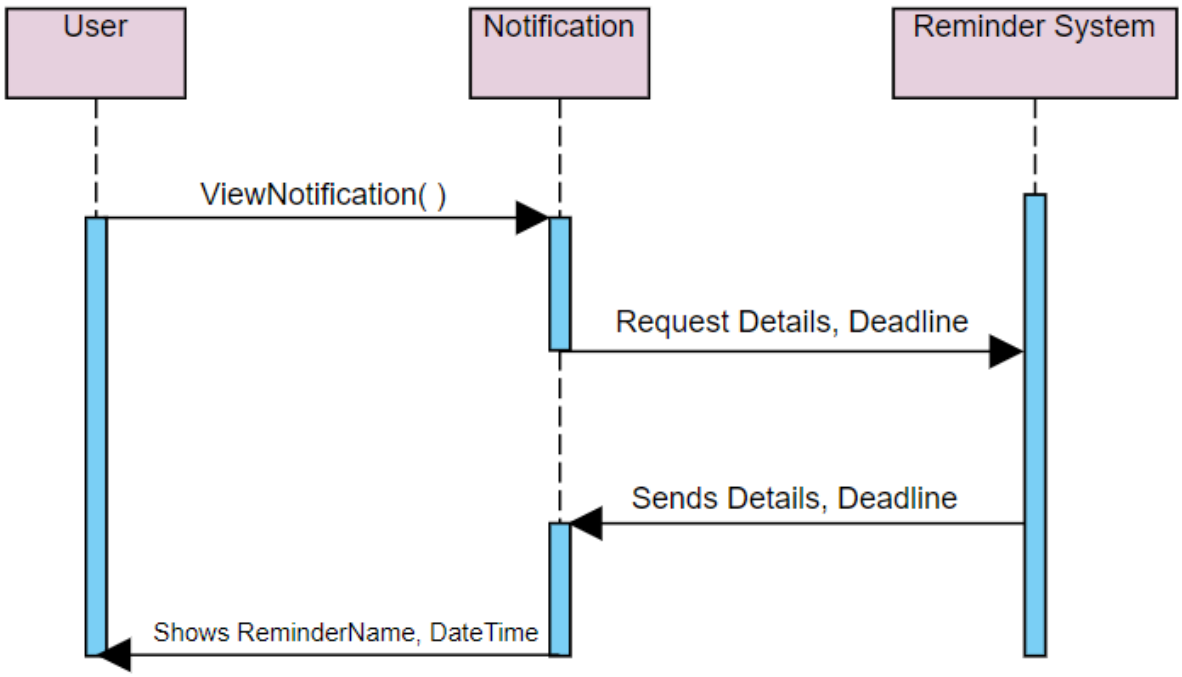
3.1.7 Use Case Notifications

3.1.7.1 Class Identification

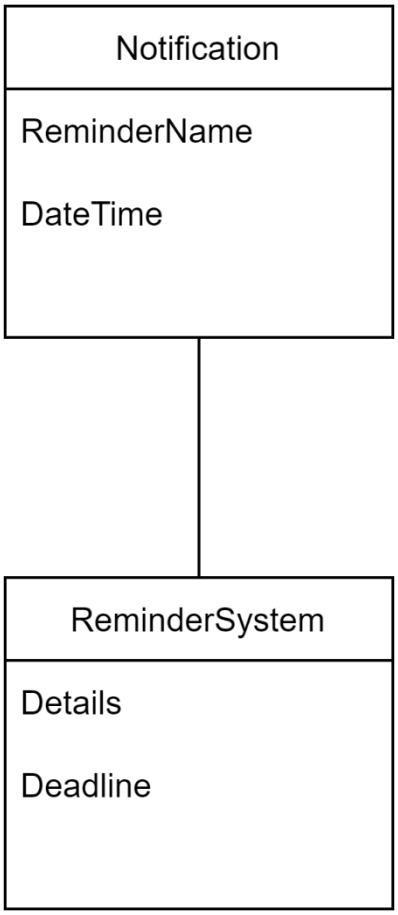
No	*Class Design Name*	*Class Type*

1	*Notifications*	*User*
2	*ReminderSystem*	*ReminderSystem*

3.1.7.2 Sequence Diagram



3.1.7.3 Class Diagram

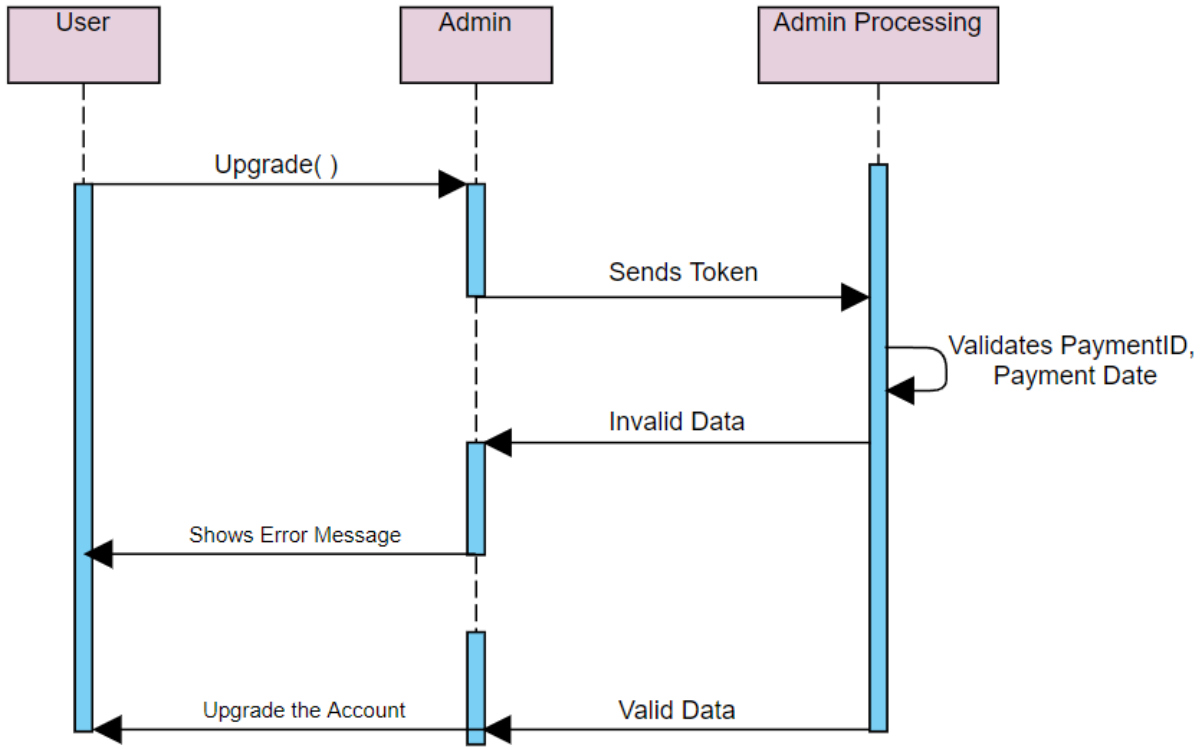


3.1.8 Use Case Accepting Upgrade

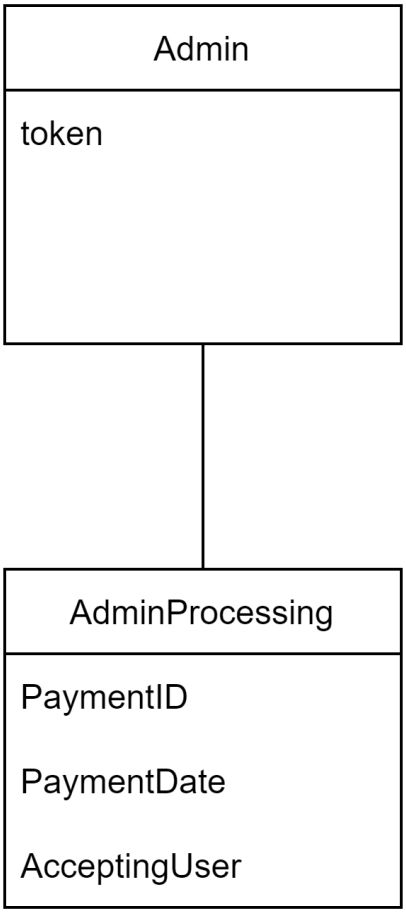
3.1.8.1 Class Identification

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

3.1.8.2 Sequence Diagram



3.1.8.3 Class Diagram

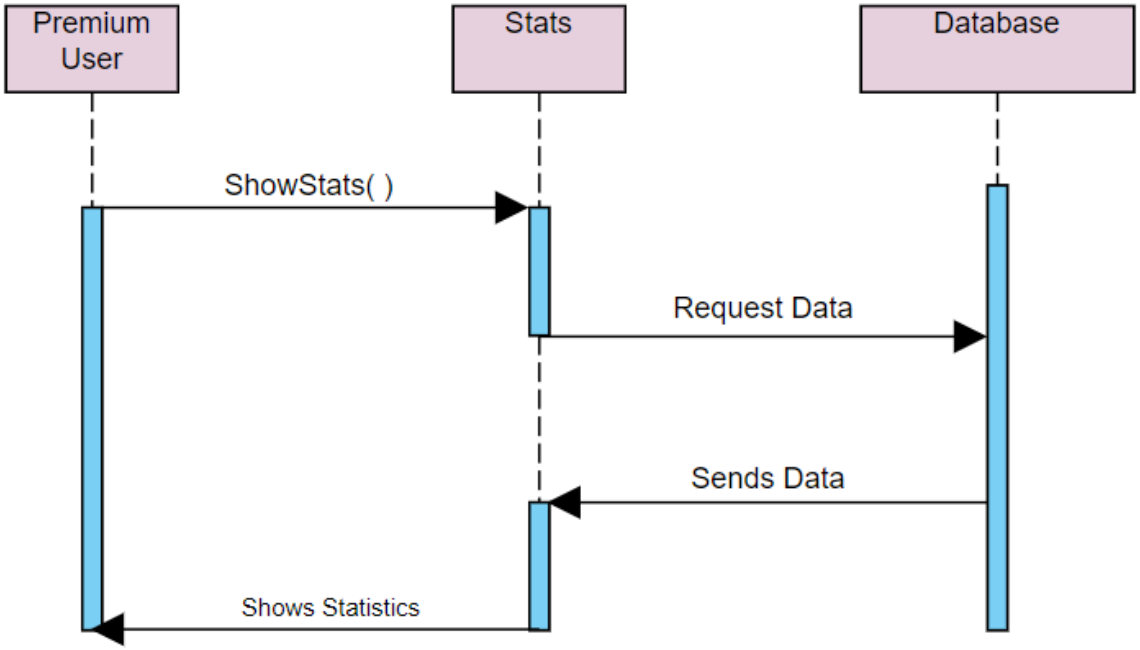


3.1.9 Use Case Statistics

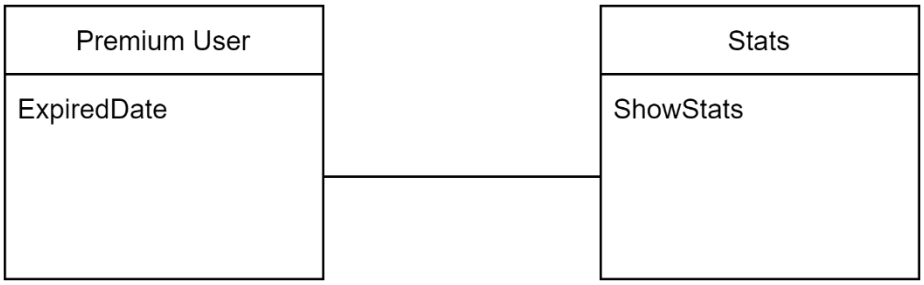
3.1.9.1 Class Identification

No	*Class Design Name*	*Class Type*
1	*PremiumUser*	*PremiumUser*
2	*Stats*	*Stats*

3.1.9.2 Sequence Diagram



3.1.9.3 Class Diagram



3.2 Class Detail Design

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

No	*Design Class Name*	*Related Analysis Class Name*
1	*Login*	*Login*
2	*Registration*	*Registration*
3	*User*	*User*
4	*Admin*	*Admin*
5	*PremiumUser*	*PremiumUser*
6	*FreeUser*	*FreeUser*
7	*ReminderSystem*	*ReminderSystem*
8	*TaskList*	*TaskList*

3.2.1 Class Login

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

Class Name : Login

Operation Name	*Visibility (private, public)*	*Description*
Signup()	*Public*	*Go to registration / sign-up page*
Login()	*Public*	*Login to the app*
Attribute Name	***Visibility (private, public)***	***Type***
Username	*Private*	*String*
Password	*Private*	*String*

3.2.2 Class Registration

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

Class Name : Registration

Operation Name	*Visibility (private, public)*	*Description*
Signup()	*Public*	*Register the user to the app database*
Attribute Name	***Visibility (private, public)***	***Type***
Username	*Private*	*String*
Password	*Private*	*String*
Email	*Private*	*String*

3.2.3 Class User

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

Class Name : User

Operation Name	*Visibility (private, public)*	*Description*
AddReminder()	*Public*	*Go to reminder page to add the reminder*
AddTask()	*Public*	*Go to Task List and see the task list or add task list*
ViewReminder()	*Public*	*Go to reminder page to view reminders*
AddToDoList()	*Public*	*Add a new to do list*
ViewToDoList()	*Public*	*View the to do list*
ViewNotification()	*Public*	*View the notifications sent*
Upgrade()	*Public*	*Request an account upgrade*
Attribute Name	***Visibility (private, public)***	***Type***
UserID	*Private*	*Int*

Email	*Private*	*String*
Username	*Private*	*String*
Password	*Private*	*String*
FullName	*Private*	*String*

3.2.4 Class Admin

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

Class Name : Admin

Operation Name	***Visibility (private, public)***	***Description***
AdminProcessing()	*Public*	*Process the request of premium access of user that login to our app*
Attribute Name	***Visibility (private, public)***	***Type***
Token	*Private*	*Int*

3.2.5 Class Premium User

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

Class Name : Premium User

Operation Name	***Visibility (private, public)***	***Description***
Stats()	*Public*	*Show statistics of the users to the screen*
Attribute Name	***Visibility (private, public)***	***Type***
ExpiredDate	*Private*	*DateTime*

3.2.6 Class Free User

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

Class Name : Free User

Operation Name	***Visibility (private, public)***	***Description***
Attribute Name	***Visibility (private, public)***	***Type***

RegisteredDate	*Private*	*DateTime*

3.2.7 Class ReminderSystem

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

Class Name : ReminderSystem

Operation Name	***Visibility (private, public)***	***Description***
Notification()	*Public*	*Notify the user about the reminder that has been set up before*
Attribute Name	***Visibility (private, public)***	***Type***
Details	*Private*	*String*
Deadline	*Private*	*DateTime*

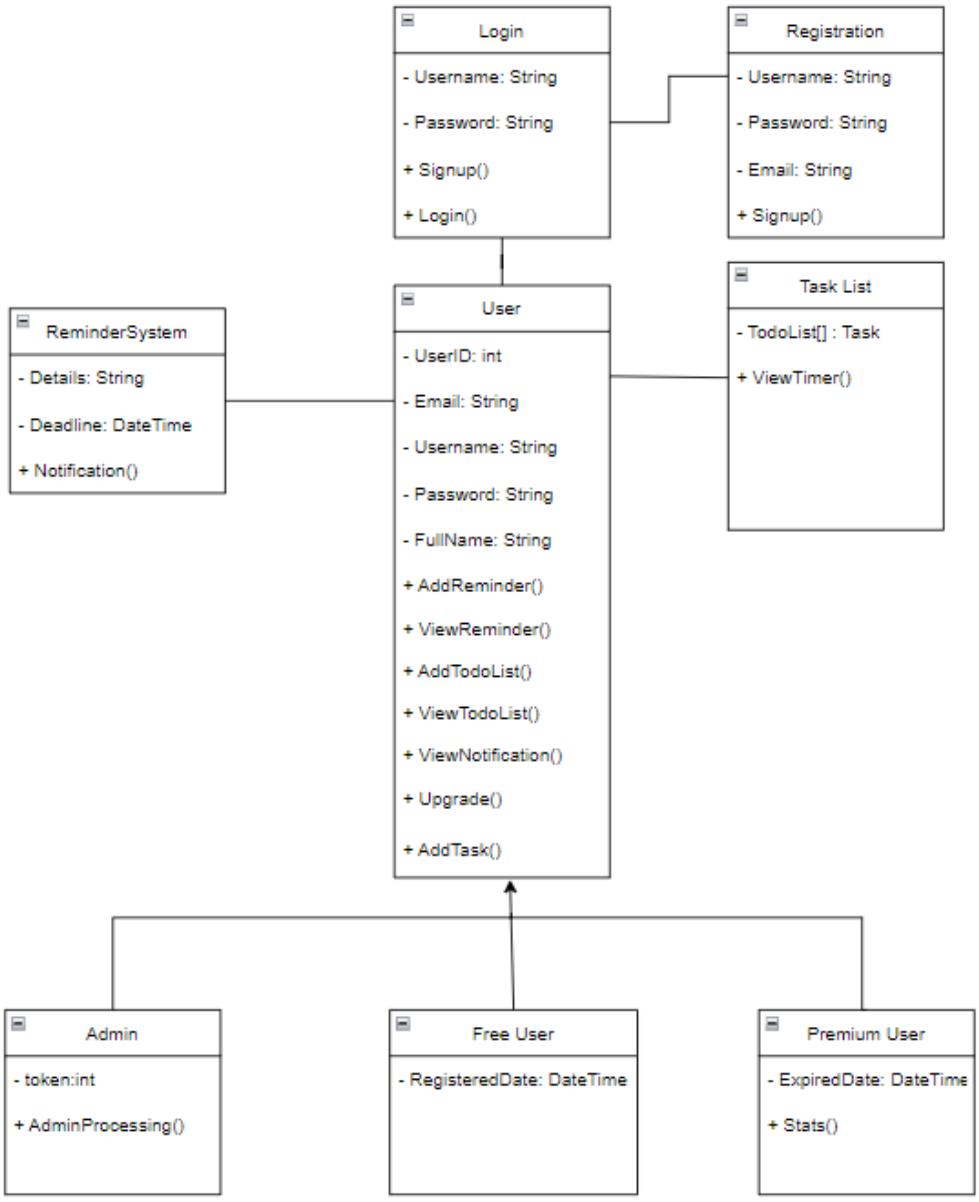
3.2.8 Class Task List

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

Class Name : Task List

Operation Name	***Visibility (private, public)***	***Description***
Timer()	*Public*	*Set the timer to the tasks*
Attribute Name	***Visibility (private, public)***	***Type***
ToDoList	*Private*	*Array of Task*

3.3 Overall Class Diagram



3.4 Algorithm/Query

3.4.1 Login Algorithm/Query

Class Name : Login
Method Name : login ()
Algorithm : (Algo-001)

input username
input password

if (SELECT * FROM Login WHERE (Login.username = username AND Login.password = password) != NULL) then

```
  return login_success
else
  return "username or password incorrect"
```

Query :

No Query	*Query*	*Description*
Q-001	SELECT * FROM Login WHERE (Login.username = username AND Login.password = password) != NULL	*Check the validity of username of password inputed*

3.4.2 Registration Algorithm/Query

Class Name : Registration
Method Name : Signup ()

Algorithm : (Algo-002)

```
input email
input username
input password

if (SELECT * FROM User WHERE (email NOT IN user.Email AND username NOT IN User.Username) ) then
  return register_account
else
  return "username and email already exist"
```

Query :

No Query	*Query*	*Description*
Q-002	SELECT * FROM User WHERE (email NOT IN user.Email AND username NOT IN User.Username)	*Check if the username and email already exist or not in the database*

3.4.3 Reminder Algorithm/Query

Class Name : User
Method Name : AddReminder()

Algorithm : (Algo-003)

```
input title
input date
input location
input description

reminder= new Reminder()
```

reminder.addTitle(title).addDate(date).addLoc(location).addDesc(description)

Reminder.createReminder()

3.4.4 Task Algorithm/Query

Class Name : User
Method Name : AddTask()

Algorithm : (Algo-004)

input title

input date

input description

input timer

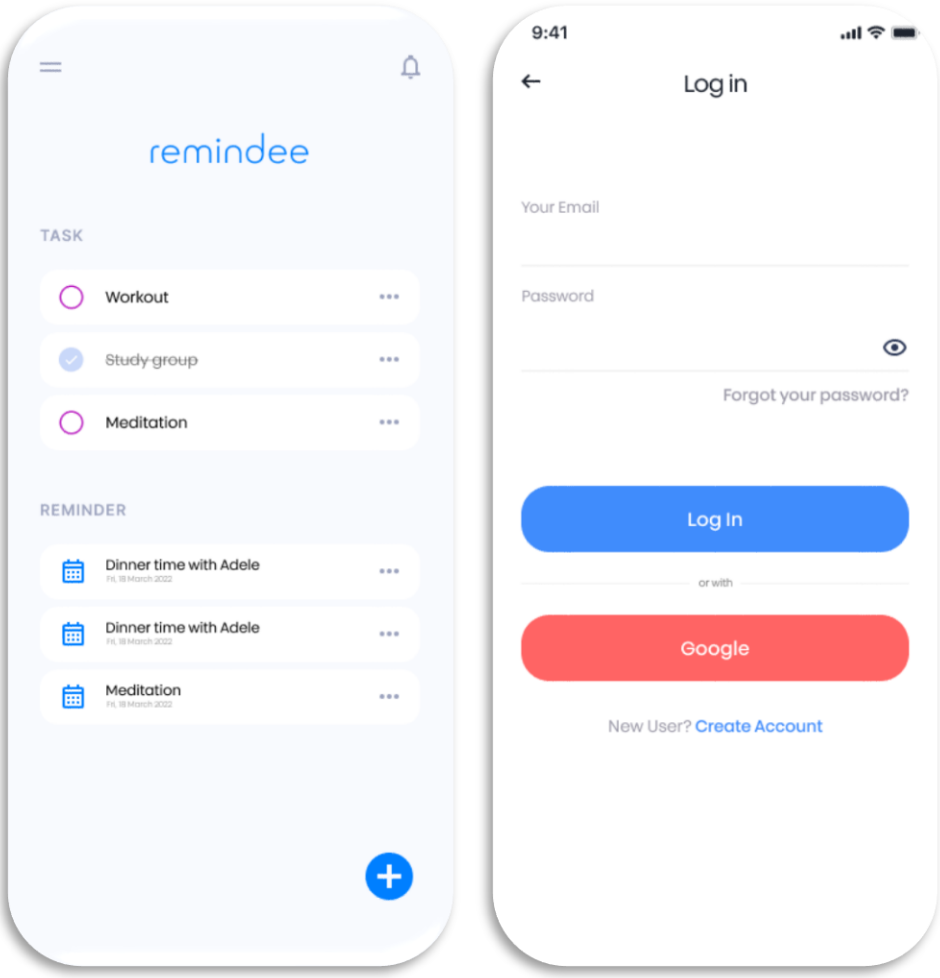
task= new Task()

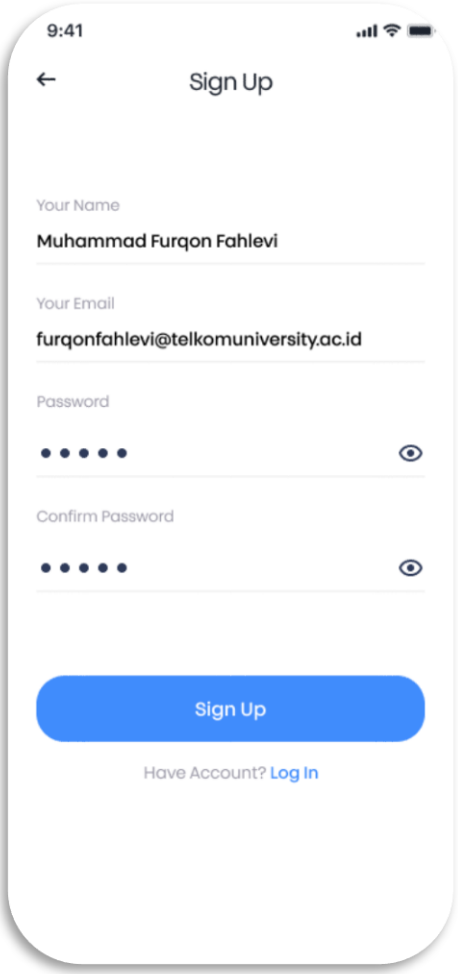
task.addTitle(title).addDate(date).addDesc(description).addTimer(timer)

task.createTask()

3.5 Interface Design

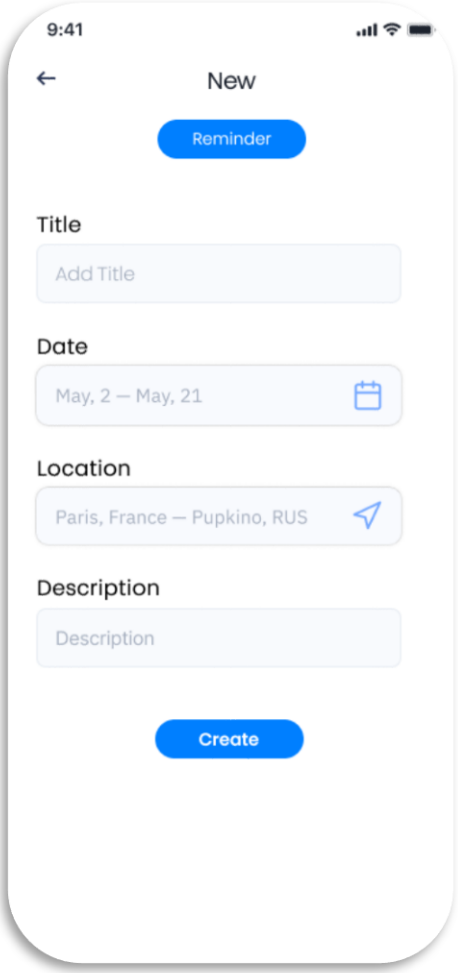
Interface : Main Page and login/signup





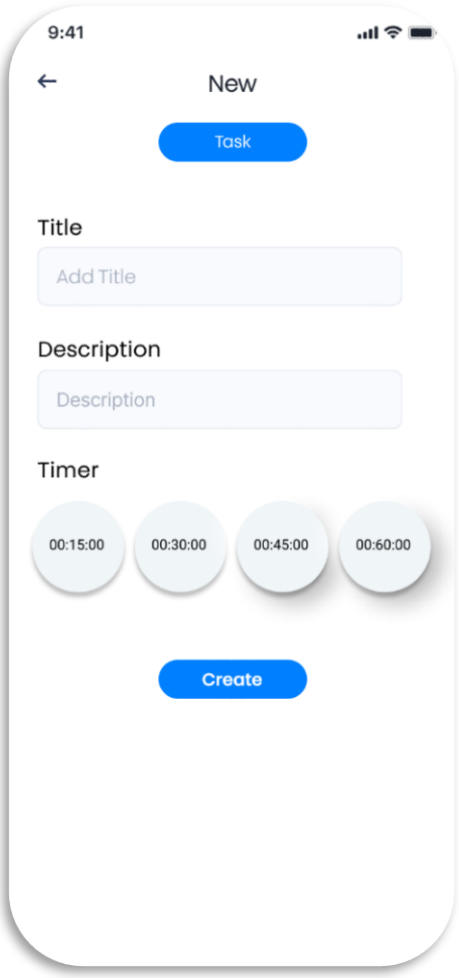
Object_Id	Type	Name	Description
BTN1	*Button*	*Hamburger menu*	*If click will show login page*
BTN2	*Button*	*Notification*	*To manage the notification settings*
BTN3	*Button*	*Add new*	*To add task or reminder*
BTN4	*Button*	*Detail*	*To show the detail of each task or reminder*
BTN5	*Button*	*Check button*	*To mark the task that has finished*
TXI1	*Text Input*	*Email login*	*Input email*
TXI2	*Text Input*	*Password login*	*Input password*
BTN6	*Button*	*Show login pass*	*Showing the input text of login password*
BTN7	*Button*	*Login*	*Click when finish input login email and password*
BTN8	*Button*	*Google login*	*Click if login with google*
BTN9	*Button*	*Create account*	*If click will create a new account and show sign up page*
TXI3	*Text Input*	*Full name*	*Input full name*
TXI4	*Text Input*	*Email signup*	*Input email*
TXI5	*Text Input*	*Password signup*	*Input password*
TXI6	*Text Input*	*Confirm pass signup*	*Input confirm password*
BTN10	*Button*	*Show signup pass*	*Showing the input text of login password*
BTN11	*Button*	*Signup*	*Click when finish input the signup detail*
BTN12	*Button*	*Login*	*If click will go back to the login page*

Interface: add new reminder/tas



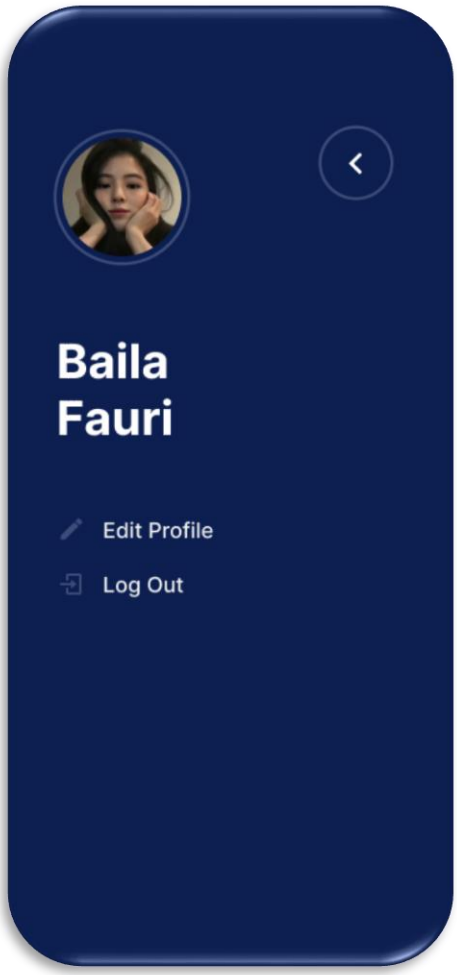
Object_Id	Type	Name	Description
BTN1	*Button*	*Back*	*If click will show main page*
TXI1	*Text Input*	*Title reminder*	*Input the title reminder name*
TXI2	*Text Input*	*Date reminder*	*Input date*
TXI3	*Text Input*	*Location reminder*	*Input location detail*
TXI4	*Button*	*Description reminder*	*Input description of the reminder*
BTN2	*Button*	*Create reminder*	*Click if all detail were finish*

Interface: add task



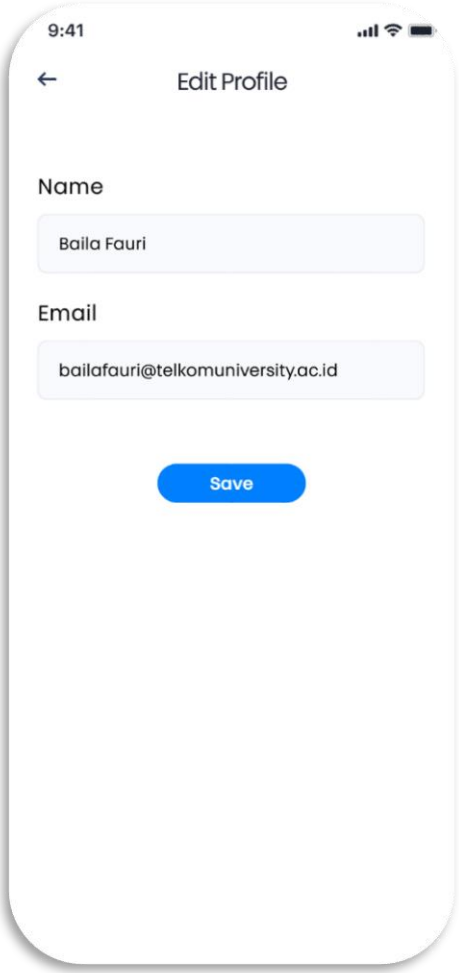
Object_Id	Type	Name	Description
BTN1	*Button*	*Back*	*If click will show main page*
TXI1	*Text Input*	*Title task*	*Input the title task name*
TXI2	*Text Input*	*Description task*	*Input description of the task*
BTN2	*Button*	*Timer task*	*Select the chosen timer for the task*
BTN3	*Button*	*Create reminder*	*Click if all detail were finish*

Interface: profile page



Object_Id	Type	Name	Description
BTN1	*Button*	*Back*	*If click will going back to main page*
BTN2	*Button*	*Edit profile*	*If click will go to edit profile page*
BTN3	*Button*	*Log out*	*If click will logging out current account*

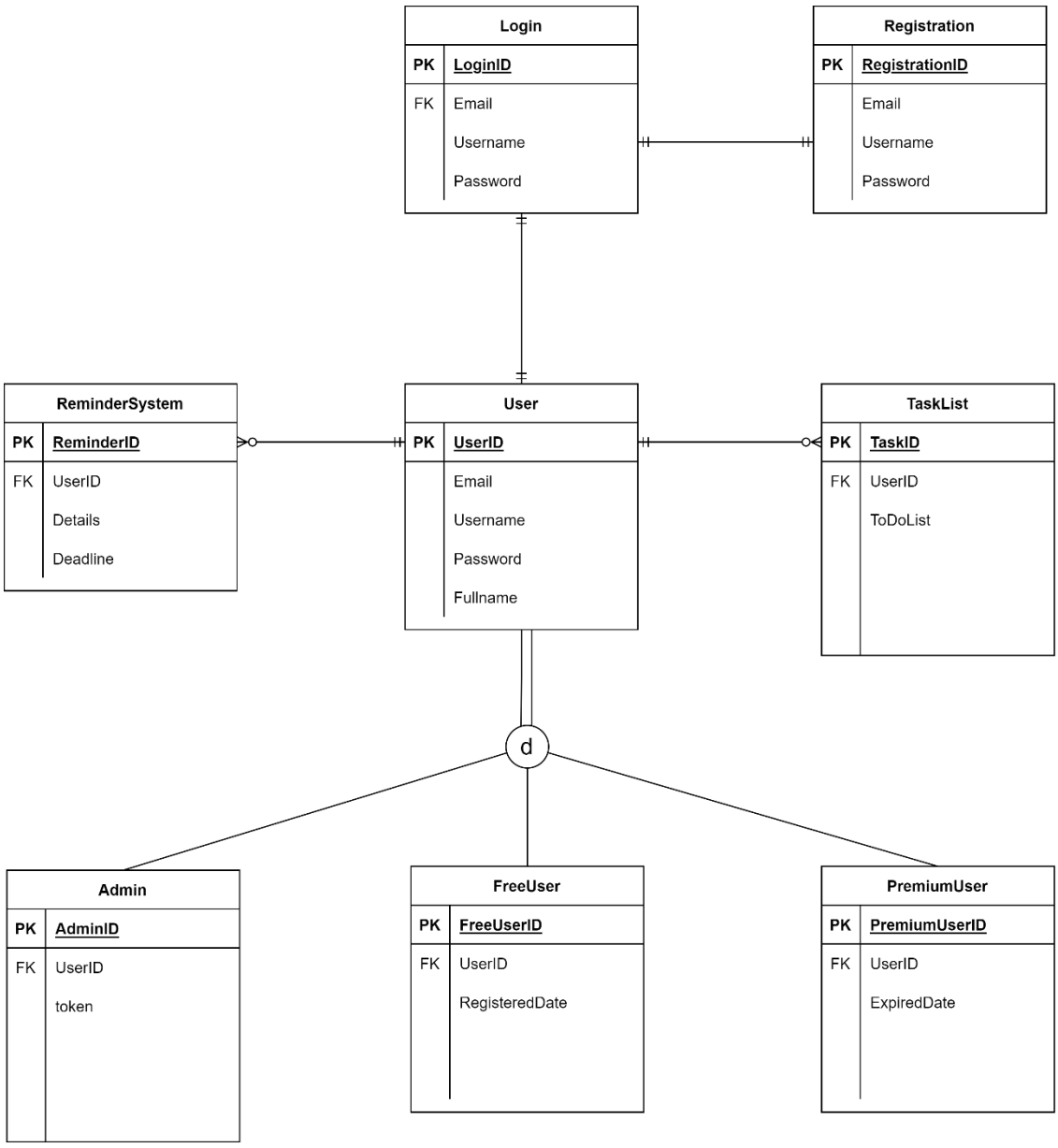
Interface: edit profile page



Object_Id	Type	Name	Description
BTN1	*Button*	*Back*	*If click will going back to profile page*
TXI1	*Text Input*	*Name*	*Input new name*
TXI2	*Text Input*	*Email*	*Input new email*
BTN2	*Button*	*Save*	*Save edited changes*

3.6 Class Persistence Representation Design

This part is filled with the basis data scheme and its traceability to the class entities.



4 Trace Matrix

Requirement	Related Use Case	Class
FR-01	Login	Login, User
FR-02	Sign Up	Login, User, Registration
FR-03	Set Up Reminders	User, ReminderSystem
FR-04	Create Task List	User, TaskList
FR-05	Timer	TaskList
FR-06	Unlimited Reminders	User, ReminderSystem
FR-07	Notifications	User, ReminderSystem
FR-08	Accepting Upgrade	Admin
FR-09	Statistics	User