

SDD-0002

SOFTWARE DESIGN DESCRIPTION

Human Resource Management for Practicum Laboratory

For:

Informatics Lab

Prepared By:

Lalu M. Riza Rizky	1301184521	Project Leader
Adrian Putra Perdana	1301170538	Analyst
Akmal Arij Santoso	1301174738	Analyst
Said Muhammad Firdaus	1301153619	Analyst
Muhammad Aufa Anasin	1301180497	Designer

Bachelor of Informatics - School of Computing

Telkom University

Telekomunikasi 1 Street, Dayeuhkolot, Bandung

Bachelor of Informatics School of Computing	Document Number		Page
	SDD-0002		
	Revision	B	11 May 2020

LIST OF CHANGES

Revision	Description
A	Changes on statechart diagram, algorithm and query, and component diagram for architectural description.
B	Changes on sequence diagram.
C	
D	
E	
F	
G	

INDEX Date	-	A	B	C	D	E	F	G
Written By	Lalu M. Riza R.	Lalu M. Riza R.	Lalu M. Riza R.					
Checked By								
Approved By								

LIST OF CHANGES PAGE

Page	Revision	Page	Revision
20 - 46	Added database entity for all sequence diagrams.		

Table of Contents

LIST OF CHANGES	2
LIST OF CHANGES PAGE	3
Table of Contents	3
INTRODUCTION	15
Document Writing Purpose	15
Problem Scope	15
Terms, Definition and Abbreviations	15
Reference	15
Systematic Discussion	16
Introduction	16
Description of Global Design	16
Detailed Design	16
Usability Matrix	16
Description of Global Design	17
Implementation Environment Design	17
Architectural Description	17
Component Description	18
Detailed Planning	20
Use Case Realization	20
Login	20
Class Identification	20
Sequence Diagram	20
Class Diagram	20
Create Account	21
Class identification	21
Sequence Diagram	21
Class Diagram	21
Authorize	22
Class Identification	22
Sequence Diagram	22
Class Diagram	22
Upload Document	23
Class Identification	23
Sequence Diagram	23
Class Diagram	23
Input Salary	24

Class Identification	24
Sequence Diagram	24
Class Diagram	24
View Assistant Salary	25
Class Identification	25
Sequence Diagram	25
Class Diagram	25
Input Ticketing	26
Class Identification	26
Sequence Diagram	26
Class Diagram	26
Input Assistant Schedule	26
Class Identification	26
Sequence Diagram	27
Class Diagram	27
Upgrade Account	27
Class Identification	27
Sequence Diagram	28
Class Diagram	28
Announce meeting or Recruitment	29
Class Identification	29
Sequence Diagram	29
Class Diagram	30
Edit Personal Account	30
Class Identification	30
Sequence Diagram	30
Class Diagram	31
View Assistant Schedule	31
Class Identification	31
Sequence Diagram	31
Class Diagram	32
View Recruitment Info Management	32
Class Identification	32
Sequence Diagram	32
Class Diagram	33
Delete Ticketing	33
Class Identification	33
Sequence Diagram	33
Class Diagram	34

Input Warning	34
Class Identification	34
Sequence Diagram	34
Class Diagram	35
View Ticketing	35
Class Identification	35
Sequence Diagram	35
Class Diagram	36
Input Recruitment Result	36
Class Identification	36
Sequence Diagram	36
Class Diagram	37
View Warning	37
Class Identification	37
Sequence Diagram	37
Class Diagram	38
Request Replacement	38
Class Identification	38
Sequence Diagram	38
Class Diagram	39
View Personal Account	39
Class Identification	39
Sequence Diagram	39
Class Diagram	40
Manage User Account	40
Class Identification	40
Sequence Diagram	40
Class Diagram	41
Deauthorize	41
Class Identification	41
Sequence Diagram	41
Class Diagram	42
Confirm Replacement	42
Class Identification	42
Sequence Diagram	43
Class Diagram	43
View Announcement	43
Class Identification	43
Sequence Diagram	44

Class Diagram	44
View and Confirm Attendance	44
Class Identification	44
Sequence Diagram	45
Class Diagram	45
Confirm Candidate Next Phase	45
Class Identification	45
Sequence Diagram	46
Class Diagram	46
Detailed Class Design	46
Class <User>	48
Class <Admin>	48
Class <Staff>	49
Class <Lecturer>	49
Class <Student>	49
Class <Manager>	49
Class <Candidate>	50
Class <PracticumAssistant>	50
Class <LabAssistant>	50
Class <WarningController>	51
Class <CandidateSelectionController>	51
Class <CommentController>	51
Class <TicketReplyController>	51
Class <RecruitmentController>	51
Class <DocumentController>	52
Class <SalaryController>	52
Class <AnnouncementController>	52
Class <CourseController>	52
Class <LecturerController>	53
Class <AssistantController>	53
Class <StudentController>	53
Class <StaffController>	53
Class <ManagerController>	54
Class <UserController>	54
Class <ScheduleController>	54
Class <TicketController>	55
Class <Controller>	55
Class <WarningModel>	55
Class <CandidateSelectionModel>	56

Class <CommentModel>	56
Class <TicketReplyModel>	56
Class <RecruitmentModel>	57
Class <DocumentModel>	57
Class <SalaryModel>	57
Class <AnnouncementModel>	58
Class <CourseModel>	58
Class <LecturerModel>	59
Class <AssistantModel>	59
Class <StudentModel>	59
Class <StaffModel>	59
Class <ManagerModel>	60
Class <UserModel>	60
Class <ScheduleModel>	61
Class <TicketModel>	61
Class <DataAccess>	62
Class <WarningRouter>	62
Class <CandidateSelectionRouter>	62
Class <CommentRouter>	62
Class <TicketReplyRouter>	63
Class <RecruitmentRouter>	63
Class <DocumentRouter>	63
Class <SalaryRouter>	63
Class <AnnouncementRouter>	64
Class <CourseRouter>	64
Class <LecturerRouter>	64
Class <AssistantRouter>	64
Class <StudentRouter>	65
Class <StaffRouter>	65
Class <ManagerRouter>	65
Class <UserRouter>	65
Class <ScheduleRouter>	66
Class <TicketRouter>	66
Class <APIRouter>	66
Overall Class Diagram	68
Algorithm / Query	69
Algorithm Class for User (Candidates, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff)	69
Class Algorithm Salary, Schedule	71

Statechart Diagram	72
Candidates	72
Practicum Assistant	73
Lecturer	74
Lab Assistant	75
Lab Staff	76
Lab Manager	77
Interface Design	78
Interface : Registration	78
Interface : Login	79
Interface : Homepage 4	80
Interface : Homepage 3	81
Interface : Announcement	82
Interface : View Schedule Form	83
Interface : View Schedule Table	85
Interface : Request Replacement	87
Interface : Authorization	88
Interface : Deauthorization	90
Interface : Input Ticketing	92
Interface : View Ticketing Table	93
Interface : View Ticketing Form	95
Interface : Salary	97
Interface : Upload Document	98
Interface : Profile	100
Interface : Recruitment Not Exist	102
Interface : Recruitment Selection Phase	103
Interface : Recruitment Exist Result	104
Class Persistence Representation Design	106
Usability Matrix	107
Attachments	111

List of Picture

Picture 1 Architectural Description	17
Picture 2 Login Sequence Diagram	20
Picture 3 Login Class Diagram	20
Picture 4 Create Account Sequence Diagram	21
Picture 5 Create Account Class Diagram	21
Picture 6 Authorize Sequence Diagram	22
Picture 7 Authorize Class Diagram	22
Picture 8 Upload Document Sequence Diagram	23
Picture 9 Upload Document Class Diagram	23
Picture 10 Input Salary Sequence Diagram	24
Picture 11 Input Salary Class Diagram	24
Picture 12 View Assistant Salary Sequence Diagram	25
Picture 13 View Assistant Class Diagram	25
Picture 14 Input Ticketing Sequence Diagram	26
Picture 15 Input Ticketing Class Diagram	26
Picture 16 Input Assistant Schedule Sequence Diagram	27
Picture 17 Input Assistant Schedule Class Diagram	27
Picture 18 Upgrade Account Sequence Diagram	28
Picture 19 Upgrade Account Class Diagram	28
Picture 20 Announce Meeting/Recruitment Sequence Diagram	29
Picture 21 Announcement Meeting/Recruitment Class Diagram	30
Picture 22 Edit Personal Account Sequence Diagram	30
Picture 23 Edit Personal Account Class Diagram	31
Picture 24 View Assistant Schedule Sequence Diagram	31
Picture 25 View Assistant Schedule Class Diagram	32
Picture 26 View Recruitment Info Management Sequence Diagram	32
Picture 27 View Recruitment Info Management Class Diagram	33
Picture 28 Delete Ticketing Sequence Diagram	33
Picture 29 Delete Ticketing Class Diagram	34
Picture 30 Input Warning Sequence Diagram	34
Picture 31 Input Warning Class Diagram	35
Picture 32 View Ticketing Sequence Diagram	35
Picture 33 View Ticketing Class Diagram	36
Picture 34 Input Recruitment Result Sequence Diagram	36
Picture 35 Input Recruitment Result Class Diagram	37
Picture 36 View Warning Sequence Diagram	37
Picture 37 View Warning Class Diagram	38
Picture 38 Request Replacement Sequence Diagram	38
Picture 39 Request Replacement Class Diagram	39
Picture 40 View Personal Account Sequence Diagram	39
Picture 41 View Personal Account Class Diagram	40
Picture 42 Manage User Account Sequence Diagram	40
Picture 43 Manage User Account Class Diagram	41
Picture 44 Deauthorize Sequence Diagram	41

Picture 45 Deauthorize Class Diagram	42
Picture 46 Confirm Replacement Sequence Diagram	43
Picture 47 Confirm Replacement Class Diagram	43
Picture 48 View Announcement Sequence Diagram	44
Picture 49 View Announcement Class Diagram	44
Picture 50 View and Confirm Attendance Sequence Diagram	45
Picture 51 View and Confirm Attendance Class Diagram	45
Picture 52 Confirm Candidate Next Phase Sequence Diagram	46
Picture 53 Confirm Candidate Next Phase Class Diagram	46
Picture 54 Overall Class Diagram	68
Picture 55 Statechart Diagram Candidates	72
Picture 56 Statechart Diagram Practicum Assistant	73
Picture 57 Statechart Lecturer	74
Picture 58 Statechart Lab Assistant	75
Picture 59 Statechart Lab Staff	76
Picture 60 Statechart Lab Manager	77
Picture 61 Registration Interface	78
Picture 62 Login Interface	79
Picture 63 Homepage 4 Interface	80
Picture 64 Homepage 3 Interface	81
Picture 65 Announcement Interface	82
Picture 66 View Schedule Form Interface	83
Picture 67 View Schedule Table Interface	85
Picture 68 Request Replacement Interface	87
Picture 69 Authorization Interface	88
Picture 70 Deauthorization Interface	90
Picture 71 Input Ticketing Interface	92
Picture 72 View Ticketing Table Interface	93
Picture 73 View Ticketing Form Interface	95
Picture 74 Salary Interface	97
Picture 75 Upload Document Interface	98
Picture 76 Profile Interface	100
Picture 77 Recruitment Not Exist Interface	102
Picture 78 Recruitment Selection Phase Interface	103
Picture 79 Recruitment Exist Result Interface	104
Picture 80 Class Persistence Representation	106

List of Table

Table 1 Component Description	18
Table 2 Login Class Identification	20
Table 3 Create Account Class Identification	21
Table 4 Authorize Class Identification	22
Table 5 Upload Document Class Identification	23
Table 6 Input Salary Class Identification	24
Table 7 View Assistant Salary Class Identification	25
Table 8 Input Ticketing Class Identification	26
Table 9 Input Assistant Schedule Class Identification	26
Table 10 Upgrade Account Class Identification	27
Table 11 Announce Meeting/Recruitment Class Identification	29
Table 12 Edit Personal Account Class Identification	30
Table 13 View Assistant Schedule Class Identification	31
Table 14 View Recruitment Info Management Class Identification	32
Table 15 Delete Ticketing Class Identification	33
Table 16 Input Warning Class Identification	34
Table 17 View Ticketing Class Identification	35
Table 18 Input Recruitment Result Class Identification	36
Table 19 View Warning Class Identification	37
Table 20 Request Replacement Class Identification	38
Table 21 View Personal Account Class Identification	39
Table 22 Manage User Account Class Identification	40
Table 23 Deauthorize Class Identification	41
Table 24 Confirm Replacement Class Identification	42
Table 25 View Announcement Class Identification	43
Table 26 View and Confirm Attendance Class Identification	44
Table 27 Confirm Candidate Next Phase Class Identification	45
Table 28 Detailed Class Design	46
Table 29 Class <User>	48
Table 30 Class <Admin>	48
Table 31 Class <Staff>	49
Table 32 Class <Lecturer>	49
Table 33 Class <Student>	49
Table 34 Class <Manager>	49
Table 35 Class <Candidate>	50
Table 36 Class <PracticumAssistant>	50
Table 37 Class <LabAsisstant>	50
Table 38 Class <WarningController>	51
Table 39 Class <CandidateSelectionController>	51
Table 40 Class <CommentController>	51
Table 41 Class <TicketReplyController>	51
Table 42 Class <RecruitmentController>	52
Table 43 Class <DocumentCotroller>	52
Table 44 Class <SalaryControler>	52

Table 45 Class <AnnouncementController>	52
Table 46 Class <CourseControllerer>	53
Table 47 Class <LecturerController>	53
Table 48 Class <AssistantControler>	53
Table 49 Class <StudentController>	53
Table 50 Class <StaffController>	54
Table 51 Class <ManagerController>	54
Table 52 Class <UserController>	54
Table 53 Class <ScheduleController>	55
Table 54 Class <TicketController>	55
Table 55 Class <Controller>	55
Table 56 Class <WarningModel>	56
Table 57 Class <CandidateSelectionModel>	56
Table 58 Class <CommentModel>	56
Table 59 Class <TicketReplyModel>	57
Table 60 Class <RecruitmentModel>	57
Table 61 Class <DocumentModel>	57
Table 62 Class <SalaryModel>	58
Table 63 Class <AnnouncementModel>	58
Table 64 Class <CourseModel>	58
Table 65 Class <LecturerModel>	59
Table 66 Class <AssistantModel>	59
Table 67 Class <StudentModel>	59
Table 68 Class <StaffModel>	60
Table 69 Class <ManagerModel>	60
Table 70 Class <UserModel>	60
Table 71 Class <ScheduleModel>	61
Table 72 Class <TicketModel>	61
Table 73 Class <DataAccess>	62
Table 74 Class <WarningRouter>	62
Table 75 Class <CandidateSelectionRouter>	62
Table 76 Class <CommentRouter>	63
Table 77 Class <TicketReplyRouter>	63
Table 78 Class <RecruitmentRouter>	63
Table 79 Class <DocumentRouter>	63
Table 80 Class <SalaryRouter>	64
Table 81 Class <AnnouncementRouter>	64
Table 82 Class <CourseRouter>	64
Table 83 Class <LecturerRouter>	64
Table 84 Class <AssistantRouter>	65
Table 85 Class <StudentRouter>	65
Table 86 Class <StaffRouter>	65
Table 87 Class <ManagerRouter>	65
Table 88 Class <UserRouter>	66
Table 89 Class <ScheduleRouter>	66
Table 90 Class <TicketRouter>	66
Table 91 Class <APIRouter>	67

Table 92 Class Login Operation Query	70
Table 93 Class Salary Operation Query	71
Table 94 Registration Interface Description	78
Table 95 Login Interface Description	79
Table 96 Homepage 4 Interface Description	80
Table 97 Homepage 3 Interface Description	81
Table 98 Announcement Interface Description	82
Table 99 View Schedule Form Interface Description	83
Table 100 View Schedule Table Interface Description	85
Table 101 Request Replacement Interface Description	87
Table 102 Authorization Interface Description	89
Table 103 Deauthorization Interface Description	90
Table 104 Input Ticketing Interface Description	92
Table 105 View Ticketing Table Interface Description	94
Table 106 View Ticketing Form Interface Description	95
Table 107 Salary Interface Description	97
Table 108 Upload Document Interface Description	98
Table 109 Profile Interface Description	100
Table 110 Recruitment Not Exist Interface Description	102
Table 111 Recruitment Selection Phase Interface Description	103
Table 112 Recruitment Exist Result Interface Description	104
Table 113 Usability Matrix	107

1. INTRODUCTION

1.1 Document Writing Purpose

This Document fills with the explanation of the software design description for the Human resource management for practicum laboratory. The purpose of writing this document is to describe every detail of the system that is going to be built. This document will be used as reference for the software building process and final evaluation in software build.

This document users are the software build team and stakeholders that are involved in this system. With the preparation of this SDD document, it is expected that software development will be more conceptualized and not cause ambiguity at the time of its development.

1.2 Problem Scope

The software Human Resource Management System for Practicum Laboratory that is developed in this paper discusses the human resource of the informatics practicum laboratory such as scheduling, announcement, meeting, salary. This system is hoped to be able to ease the human resource management of the lab easier.

1.3 Terms, Definition and Abbreviations

The following are some definitions, abbreviations, and acronyms contained in this document:

1. SDD : Software Design Description is a written description of a software product that a software designer writes in order to give a software development team overall guidance to the architecture of the software project.
2. SRS: Software Requirement Specification document is a document that specifies the requirement of the software that will be developed.
3. OO: Object Oriented is a programming paradigm that is based on object interaction between one and another.
4. DBMS: Abbreviation for "Database Management System", which is an organization system and database processing system on a computer.
5. JavaScript: High-level and multi-paradigm programming language that is used mainly for web development.
6. MySQL: An open source Relational DBMS developed by Oracle Corporation.
7. JSON: JavaScript Object Notation is a data object language that is universally used for interchanging data.
8. SQL: Structured Query Language is a language used to execute query, DDL, and DML from a database.
9. ORM: Object Relational Mapping is a method to convert data that makes virtual object databases based on the objects in OOP.

1.4 Reference

- Group 3 IF-39-03. (2017). DPPL Citra Mart Information System.
- Group 2 IF-41-INT02. (2019). SDD Web-Based Application of Assistant Recruitment in Telkom University
- Group 2 IF-42-INT. (2020). SRS Human Resource Management System For Practicum Laboratory

1.5 Systematic Discussion

This SDD document contains a description of the design of the library information system software which will be developed based on the SRS document. This SDD document will explain the details of the software design so that it can be implemented. This document outlines consists of four chapters with the following details:

1. Introduction

The introduction contains an explanation of the SDD document which includes the purpose of making this document, the scope of the problem solved by the software developed, definitions, naming and numbering rules, references, and document summaries.

2. Description of Global Design

The global design description contains the design of the software to be built including, design of the implementation environment, architectural description, and component description.

3. Detailed Design

The detailed design of this document contains the use case realization, class detail design, class diagram description, algorithm / query, statechart diagram, interface design, and class persistence representation design.

4. Usability Matrix

The compliance matrix contains functional matters contained in the SRS document.

2 Description of Global Design

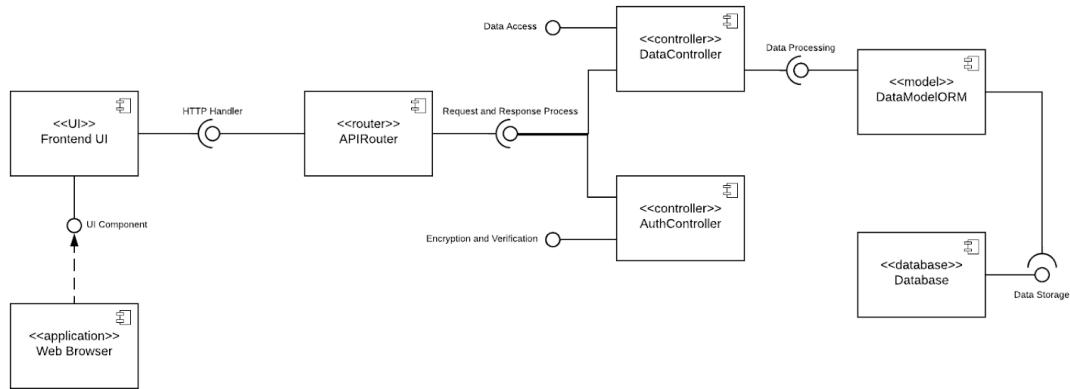
2.1 Implementation Environment Design

This system is implemented in the following environment:

- Operating system: Windows 10, Mac OS X, Ubuntu 19.10 or any Linux equivalent.
- Programming Language: Javascript, SQL
- Data Object Language: JSON
- Development Tool: Visual Studio Code, JetBrains WebStorm, Node.js, NPM, JetBrains DataGrip
- Backend Frameworks: Express.js, Prisma ORM
- Authentication Framework: JSON Web Token, OAuth2, Passport.js
- Frontend Framework: Nuxt.js
- DBMS: MySQL

2.2 Architectural Description

A picture of architecture or components that will be applied to the software "Human Resource Management for Practicum Laboratory" to facilitate developers in developing or implementing this software.



Picture 1 Architectural Description

2.3 Component Description

Table 1 Component Description

No	Component Name	Description
1	Admin	User in the Human resource management for practicum laboratory
2	User	User in the Human resource management for practicum laboratory
3	Lab Manager	User in the Human resource management for practicum laboratory
4	Lab Staff	User in the Human resource management for practicum laboratory
6	Lecturer	User in the Human resource management for practicum laboratory
5	Lab Assistant	User in the Human resource management for practicum laboratory
6	Practicum Assistant	User in the Human resource management for practicum laboratory
7	Student	User in the Human resource management for practicum laboratory
8	Candidate	User in the Human resource management for practicum laboratory
9	Create account	This function is used by candidates to make a recruitment account
10	Login	Used by every role in the software for logging in to system
11	Document upload	Candidate able to upload their documents for during recruitment session, document phase
12	View recruitment result	Candidates able to view their result for each recruitment phase
13	View schedule	Assistants able to check their schedules
14	Replacement request	Practicum assistants able to request replacements if they cannot attend
15	Input ticketing	Assistants able to input ticketing for complaints, feedback, and issues regarding practicum or internal issues
16	View announcement	Assistants and staffs can view an announcement given from others
17	Confirm attendance	Assistants are able to confirm their attendance based on roles
18	View attendance	Assistants able to view their attendance
19	View financial info	Assistants able to view their available honorary salaries to take and current salary
20	Manage schedule	Lab assistants and staffs able to manage schedules for assistants

21	Manage attendance	Lab assistants and staffs able to manage attendance (e.g. confirm attendance if there is no confirmation from assistants) for practicum assistants based on verified proof
22	Upgrade account	Lab assistants can upgrade candidates' accounts to practicum assistants. Lab staff can upgrade practicum assistants to lab assistant accounts. Both upgrades must be under the authorization of lab manager
23	Manage finance	Lab staffs can input values of available salary to take and estimated total salary
24	Authorize	Lab manager can authorize actions from staffs and ab assistants like recruitments and candidates acceptance
25	View ticketing	Every role can view their current ticketing process
26	Input warning	Lab managers can give warnings to practicum assistant and lab assistant, while lab assistant can only give warnings to practicum assistant.
27	Announce	Lab manager and lab assistants able to announce based on their respective roles to others.
28	Overview	Every role will have an overview based on their roles
29	Manage account	Lab staffs and assistants has rights to alter records of account, while others can only view or edit non institutional records
30	Deauthorize	Lab manager can deauthorize assistant accounts in case of fatal violations or warnings
31	Logout	Every role are able to logout from their account
32	Edit personal account	Menu to edit the personal info of the user
33	View personal account	Menu to view user's personal info
34	Assistant	User in the Human resource management for practicum laboratory
35	View Warning	Menu to see warnings on the user's account

3 Detailed Planning

3.1 Use Case Realization

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

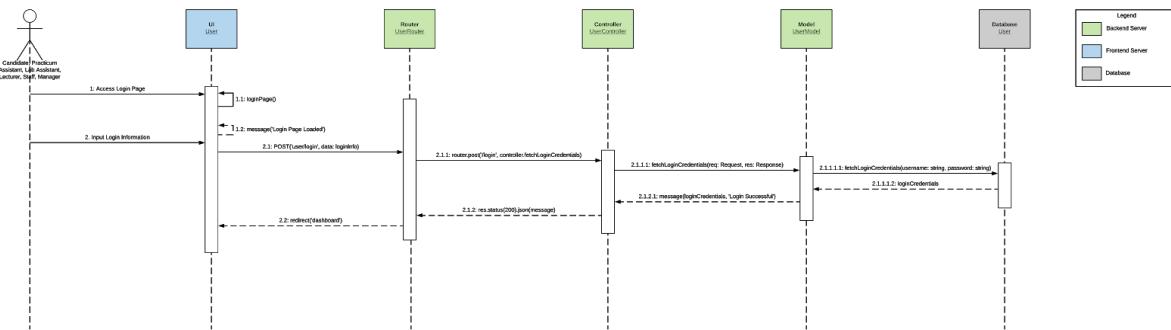
3.1.1 Login

3.1.1.A Class Identification

Table 2 Login Class Identification

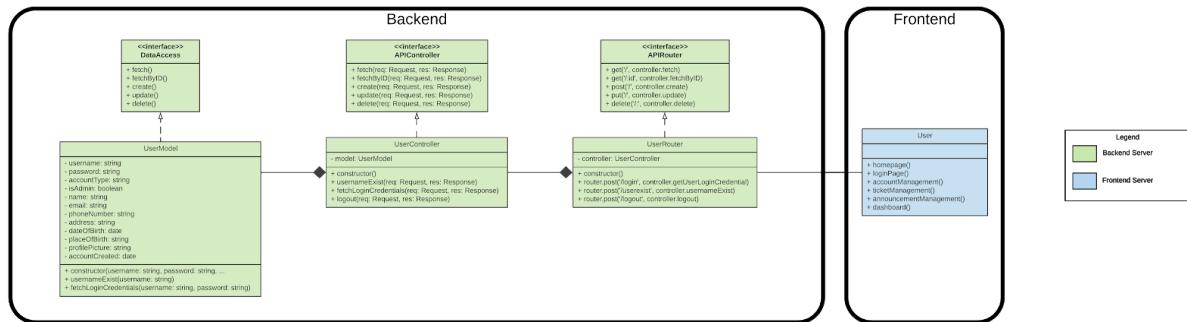
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	UserRouter	Router (Backend)
3	UserController	Controller (Backend)
4	UserModel	Model (Backend)
5	DataAccess	DataAccessInterface (Backend)
6	APIController	ControllerInterface (Backend)
7	APIRouter	RouterInterface (Backend)

3.1.1.B Sequence Diagram



Picture 2 Login Sequence Diagram

3.1.1.C Class Diagram



Picture 3 Login Class Diagram

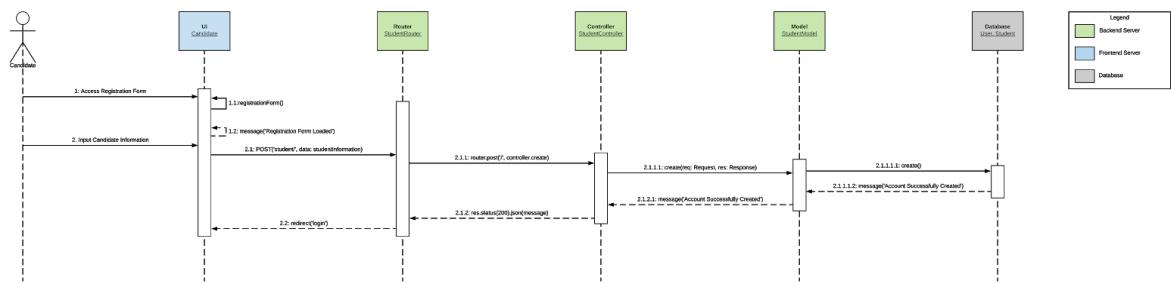
3.1.2 Create Account

3.1.2.A Class identification

Table 3 Create Account Class Identification

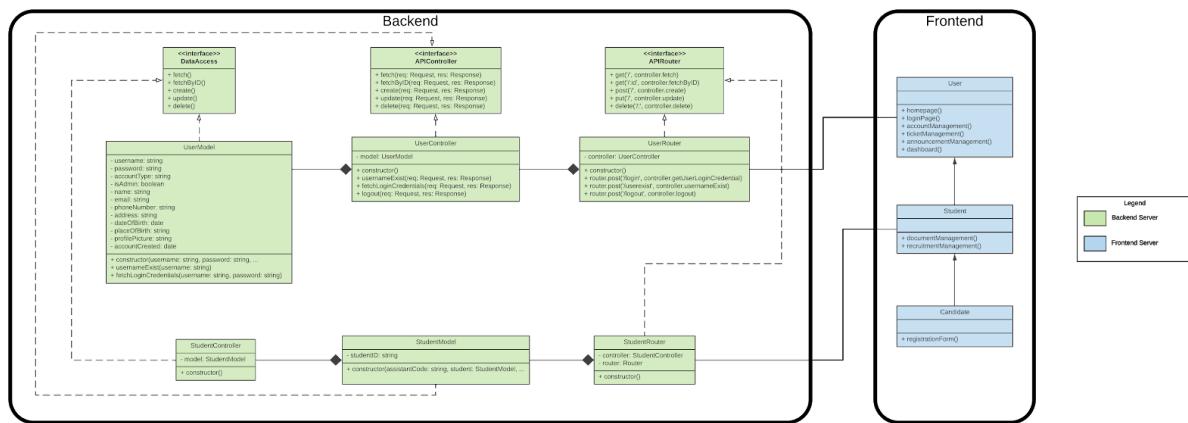
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	Candidate	UI (Frontend)
4	UserRouter	Router (Backend)
5	UserController	Controller (Backend)
6	UserModel	Model (Backend)
7	StudentController	Controller (Backend)
8	StudentModel	Model (Backend)
9	StudentRouter	Router (Backend)
10	DataAccess	DataAccessInterface (Backend)
11	APIController	ControllerInterface (Backend)
12	APIRouter	RouterInterface (Backend)

3.1.2.B Sequence Diagram



Picture 4 Create Account Sequence Diagram

3.1.2.C Class Diagram



Picture 5 Create Account Class Diagram

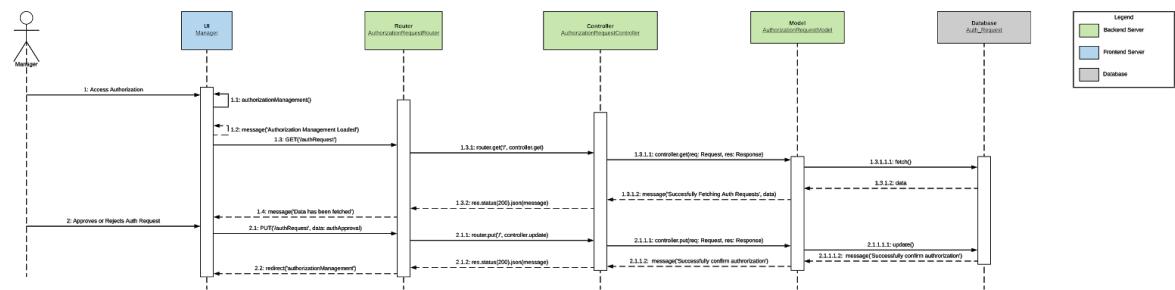
3.1.3 Authorize

3.1.3.A Class Identification

Table 4 Authorize Class Identification

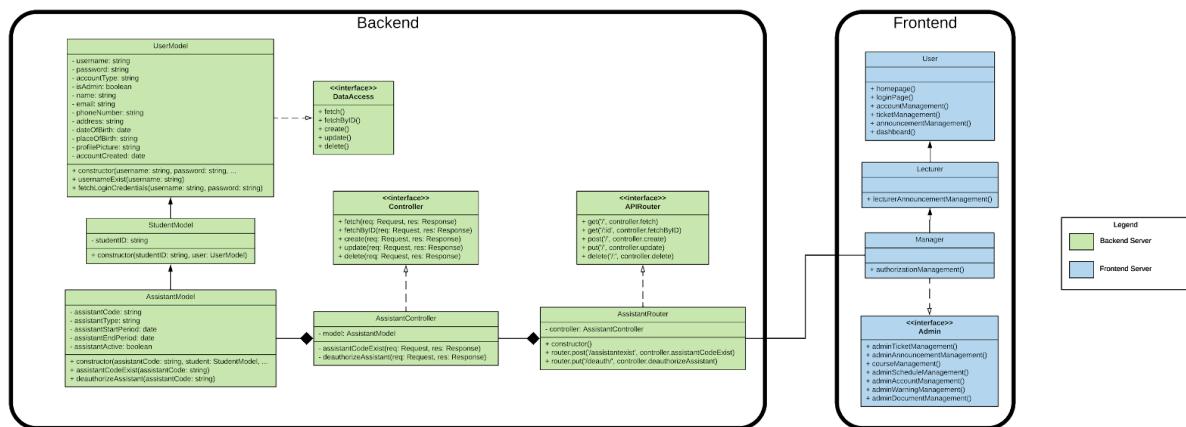
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Lecturer	UI (Frontend)
3	Manager	UI (Frontend)
4	Admin	UIInterface (Frontend)
5	AssistantRouter	Router (Backend)
6	AssistantController	Controller (Backend)
7	AssistantModel	Model (Backend)
8	StudentModel	Model (Backend)
9	UserModel	Model (Backend)
10	DataAccess	DataAccessInterface (Backend)
11	APIController	ControllerInterface (Backend)
12	APIRouter	RouterInterface (Backend)

3.1.3.B Sequence Diagram



Picture 6 Authorize Sequence Diagram

3.1.3.C Class Diagram



Picture 7 Authorize Class Diagram

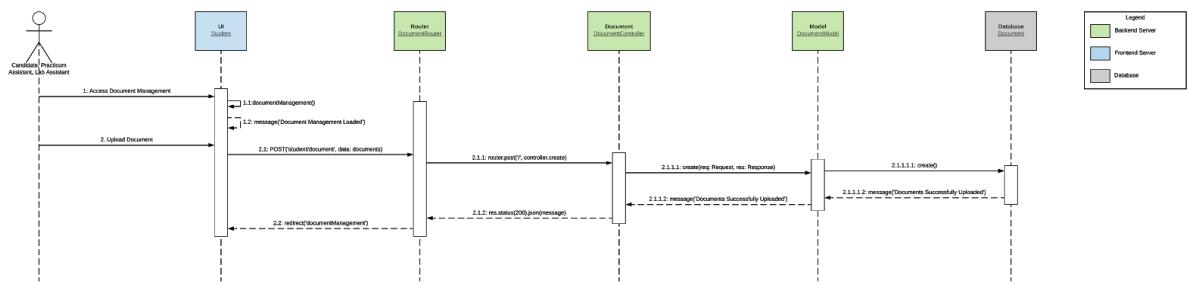
3.1.4 Upload Document

3.1.4.A Class Identification

Table 5 Upload Document Class Identification

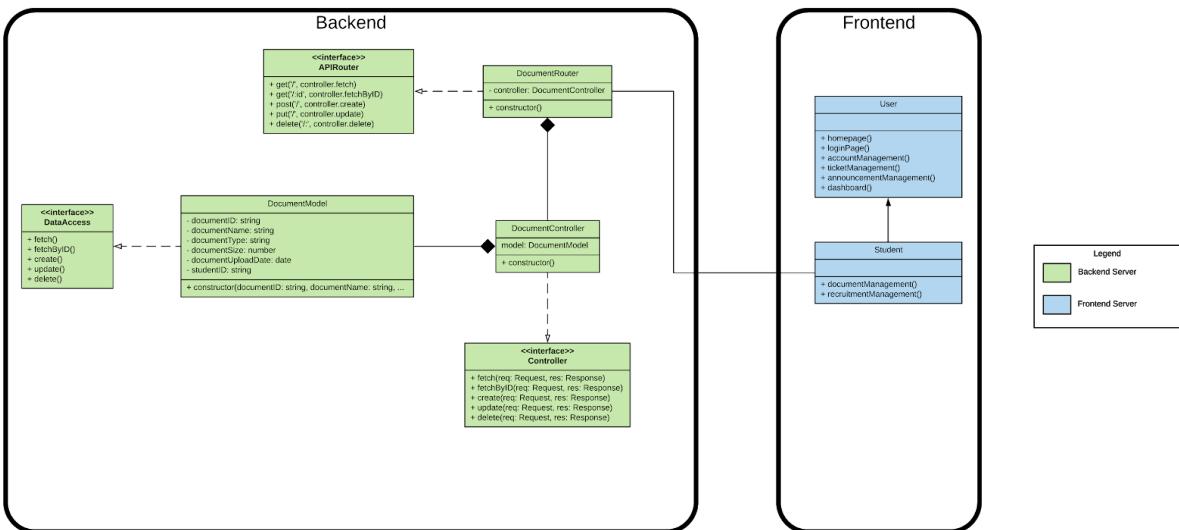
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	DocumentRouter	Router (Backend)
4	DocumentController	Controller (Backend)
5	DocumentModel	Model (Backend)
6	DataAccess	DataAccessInterface (Backend)
7	APIController	ControllerInterface (Backend)
8	APIRouter	RouterInterface (Backend)

3.1.4.B Sequence Diagram



Picture 8 Upload Document Sequence Diagram

3.1.4.C Class Diagram



Picture 9 Upload Document Class Diagram

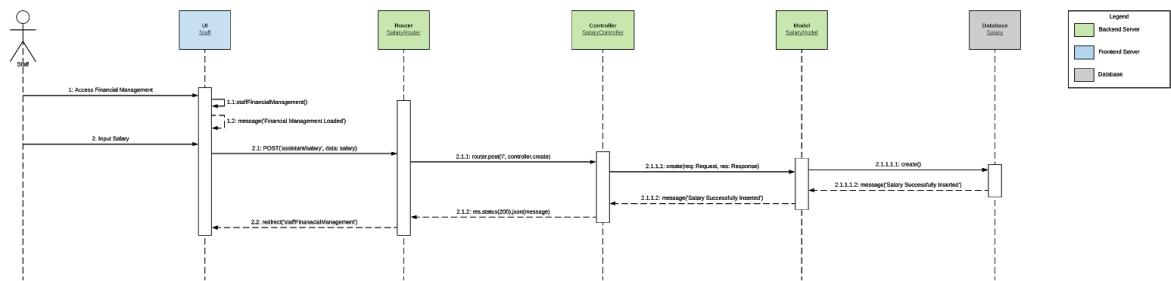
3.1.5 Input Salary

3.1.5.A Class Identification

Table 6 Input Salary Class Identification

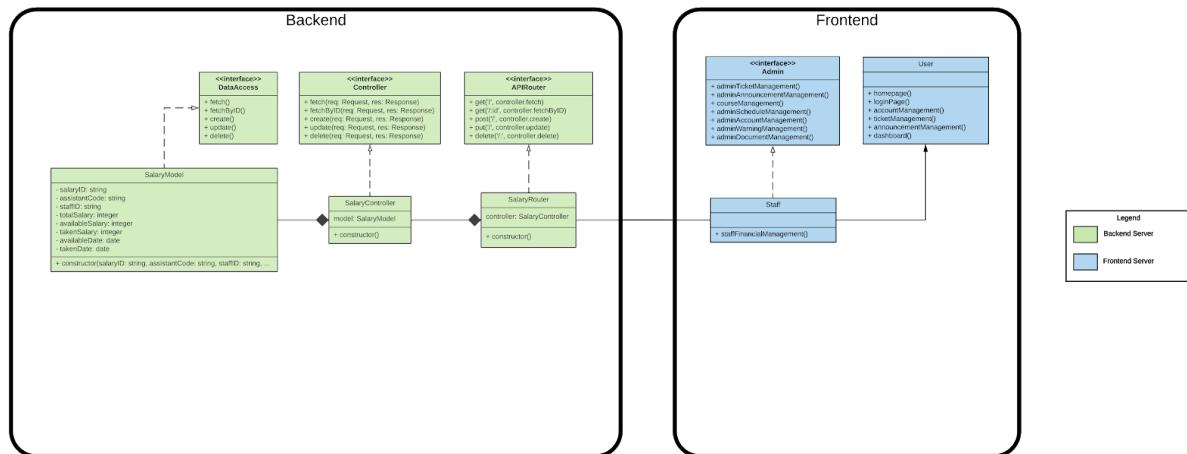
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Staff	UI (Frontend)
3	Admin	UIInterface (Frontend)
4	SalaryController	Controller (Backend)
5	SalaryModel	Model (Backend)
6	SalaryRouter	Router (Backend)
7	DataAccess	DataAccessInterface (Backend)
8	Controller	ControllerInterface (Backend)
9	APIRouter	RouterInterface (Backend)

3.1.5.B Sequence Diagram



Picture 10 Input Salary Sequence Diagram

3.1.5.C Class Diagram



Picture 11 Input Salary Class Diagram

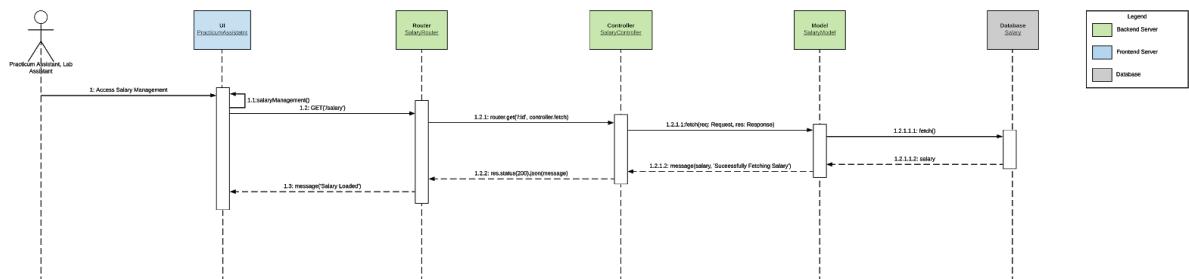
3.1.6 View Assistant Salary

3.1.6.A Class Identification

Table 7 View Assistant Salary Class Identification

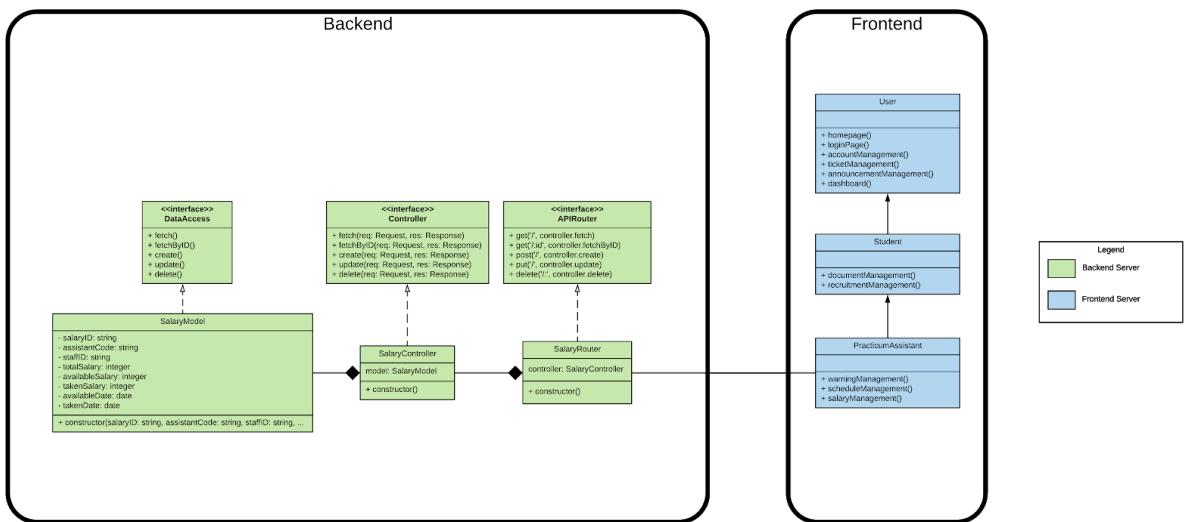
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	Practicum Assistant	UI (Frontend)
4	SalaryController	Controller (Backend)
5	SalaryModel	Controller (Backend)
6	SalaryRouter	Model (Backend)
7	DataAccess	DataAccessInterface (Backend)
8	Controller	ControllerInterface (Backend)
9	APIRouter	RouterInterface (Backend)

3.1.6.B Sequence Diagram



Picture 12 View Assistant Salary Sequence Diagram

3.1.6.C Class Diagram



Picture 13 View Assistant Salary Class Diagram

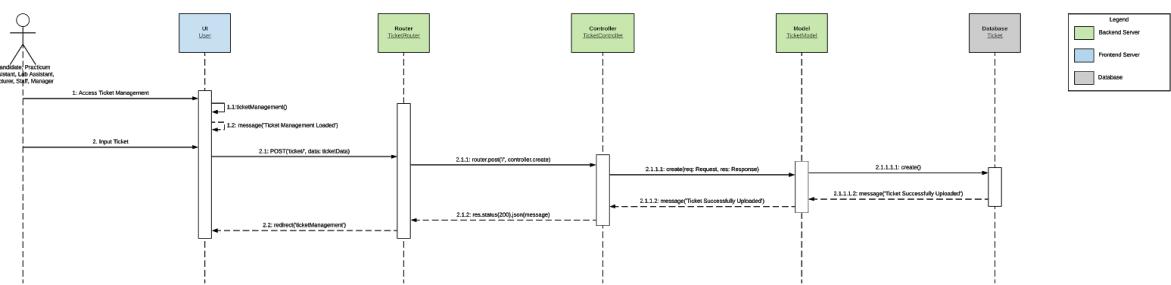
3.1.7 Input Ticketing

3.1.7.A Class Identification

Table 8 Input Ticketing Class Identification

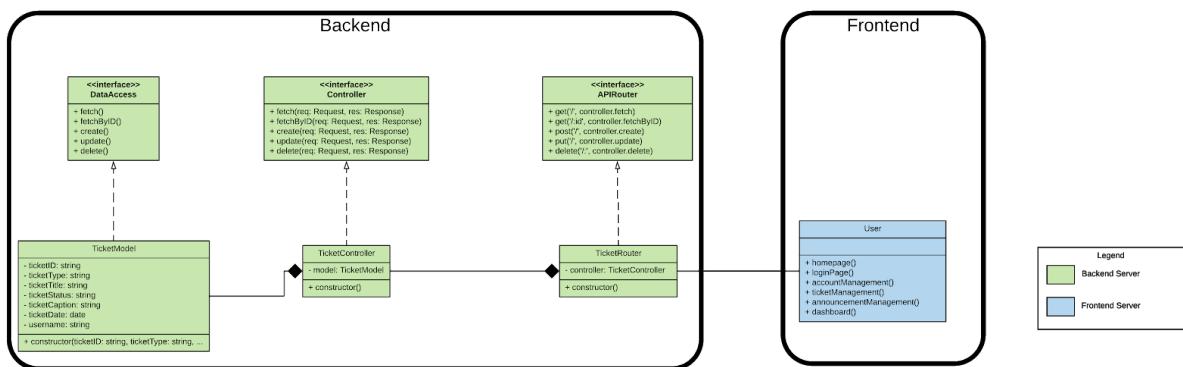
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	TicketRouter	Router (Backend)
3	TicketModel	Model (Backend)
4	TicketController	Controller (Backend)
5	DataAccess	DataAccessInterface (Backend)
6	Controller	ControllerInterface (Backend)
7	APIRouter	RouterInterface (Backend)

3.1.7.B Sequence Diagram



Picture 14 Input Ticketing Sequence Diagram

3.1.7.C Class Diagram



Picture 15 Input Ticketing Class Diagram

3.1.8 Input Assistant Schedule

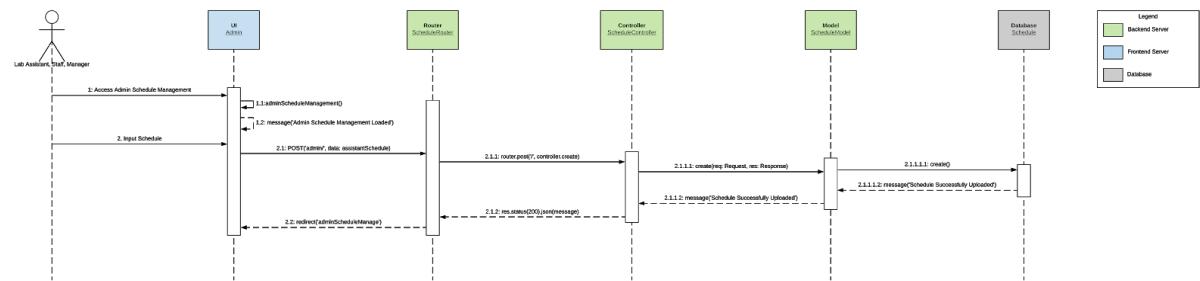
3.1.8.A Class Identification

Table 9 Input Assistant Schedule Class Identification

No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)

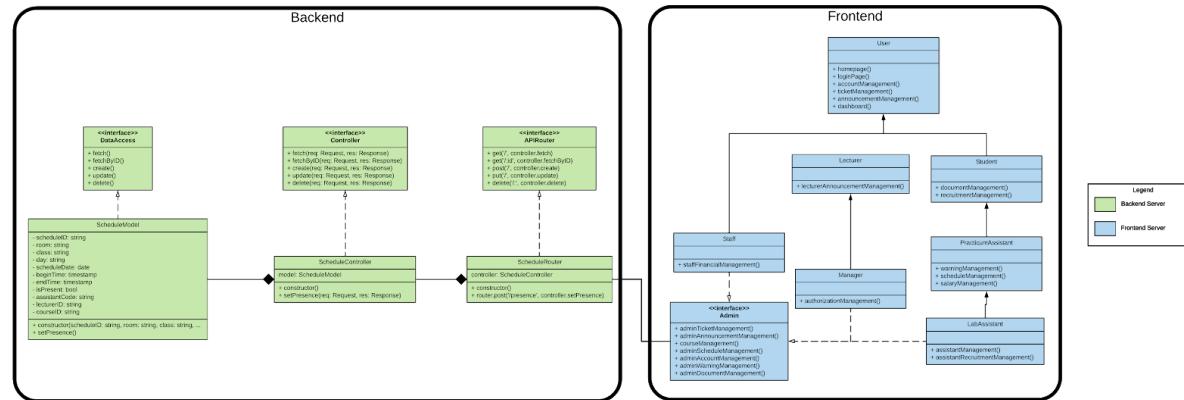
4	LabAssistant	UI (Frontend)
5	Lecturer	UI (Frontend)
6	Manager	UI (Frontend)
7	Staff	UI (Frontend)
8	Admin	UIInterface (Frontend)
9	ScheduleRouter	Router (Backend)
10	ScheduleModel	Model (Backend)
11	ScheduleController	Controller (Backend)
12	DataAccess	DataAccessInterface (Backend)
13	Controller	ControllerInterface (Backend)
14	APIRouter	RouterInterface (Backend)

3.1.8.B Sequence Diagram



Picture 16 Input Assistant Schedule Sequence Diagram

3.1.8.C Class Diagram



Picture 17 Input Assistant Schedule Class Diagram

3.1.9 Upgrade Account

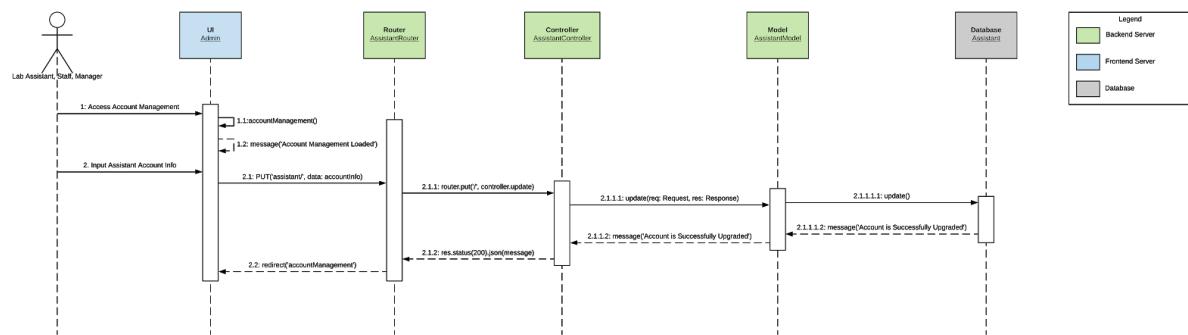
3.1.9.A Class Identification

Table 10 Upgrade Account Class Identification

No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	LabAssistant	UI (Frontend)

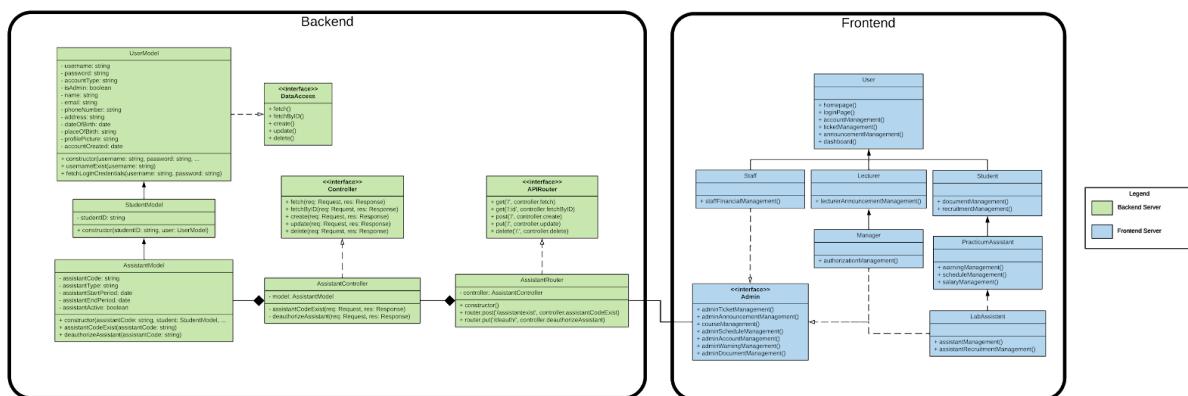
5	Lecturer	UI (Frontend)
6	Manager	UI (Frontend)
7	Staff	UI (Frontend)
8	Admin	UIInterface (Frontend)
9	AssistantRouter	Router (Backend)
10	AssistantModel	Model (Backend)
11	AssistantController	Controller (Backend)
12	StudentModel	Model (Backend)
13	UserModel	Model (Backend)
14	DataAccess	DataAccessInterface (Backend)
15	Controller	ControllerInterface (Backend)
16	APIRouter	RouterInterface (Backend)

3.1.9.B Sequence Diagram



Picture 18 Upgrade Account Sequence Diagram

3.1.9.C Class Diagram



Picture 19 Upgrade Account Class Diagram

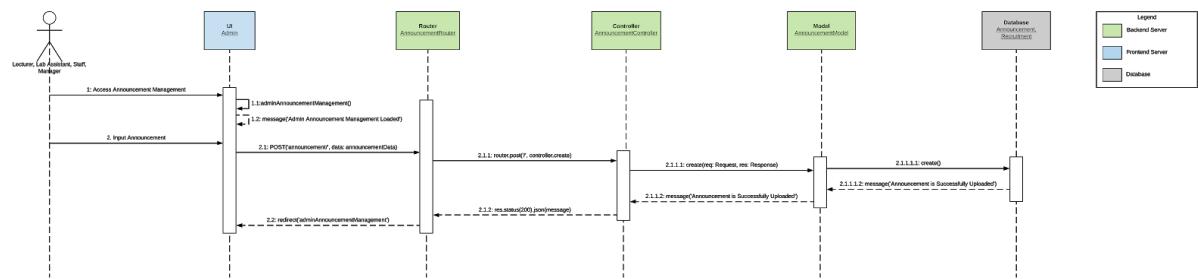
3.1.10 Announce meeting or Recruitment

3.1.10.A Class Identification

Table 11 Announce meeting or Recruitment Class Identification

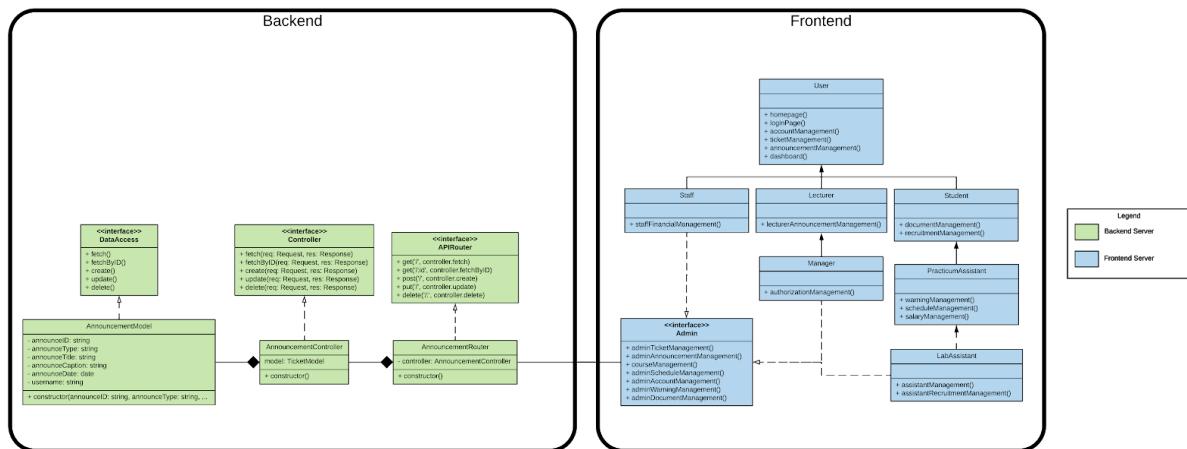
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	LabAssistant	UI (Frontend)
5	Lecturer	UI (Frontend)
6	Manager	UI (Frontend)
7	Staff	UI (Frontend)
8	Admin	UIInterface (Frontend)
9	AnnouncementRouter	Router (Backend)
10	AnnouncementModel	Model (Backend)
11	AnnouncementController	Controller (Backend)
12	DataAccess	DataAccessInterface (Backend)
13	Controller	ControllerInterface (Backend)
14	APIRouter	RouterInterface (Backend)

3.1.10.B Sequence Diagram



Picture 20 Announce Meeting or Recruitment Sequence Diagram

3.1.10.C Class Diagram



Picture 21 Announce Meeting or Recruitment Class Diagram

3.1.11 Edit Personal Account

3.1.11.A Class Identification

Table 12 Edit Personal Account Class Identification

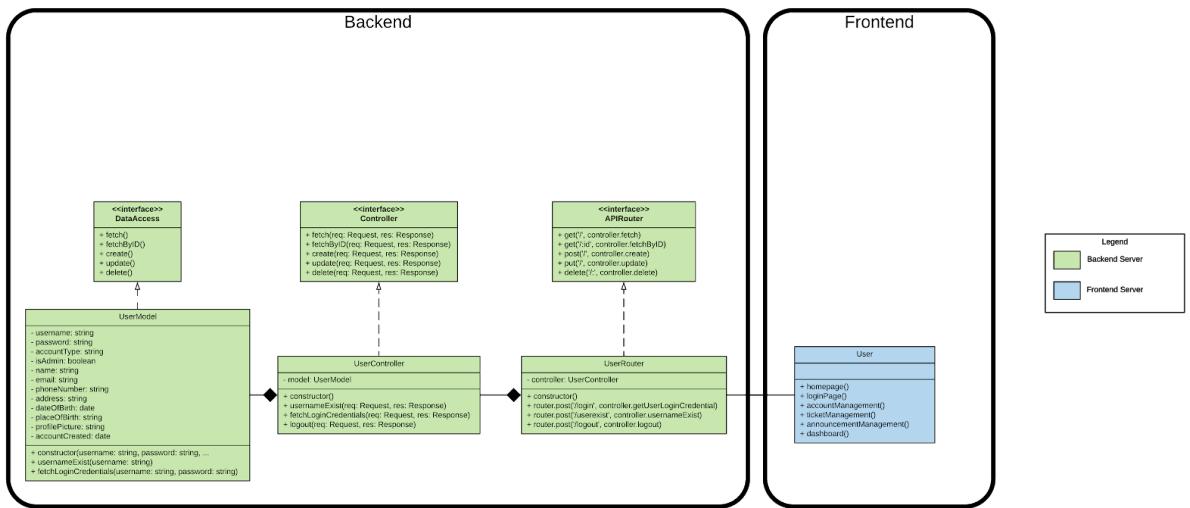
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	UserRouter	Router (Backend)
3	UserController	Controller (Backend)
4	UserModel	Model (Backend)
5	DataAccess	DataAccessInterface (Backend)
6	APIController	ControllerInterface (Backend)
7	APIRouter	RouterInterface (Backend)

3.1.11.B Sequence Diagram



Picture 22 Edit Personal Account Sequence Diagram

3.1.11.C Class Diagram



Picture 23 Edit Personal Account Class Diagram

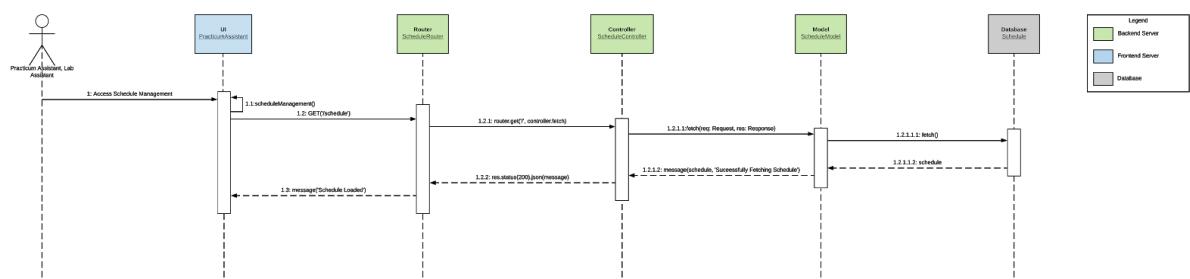
3.1.12 View Assistant Schedule

3.1.12.A Class Identification

Table 13 View Assistant Schedule Class Identification

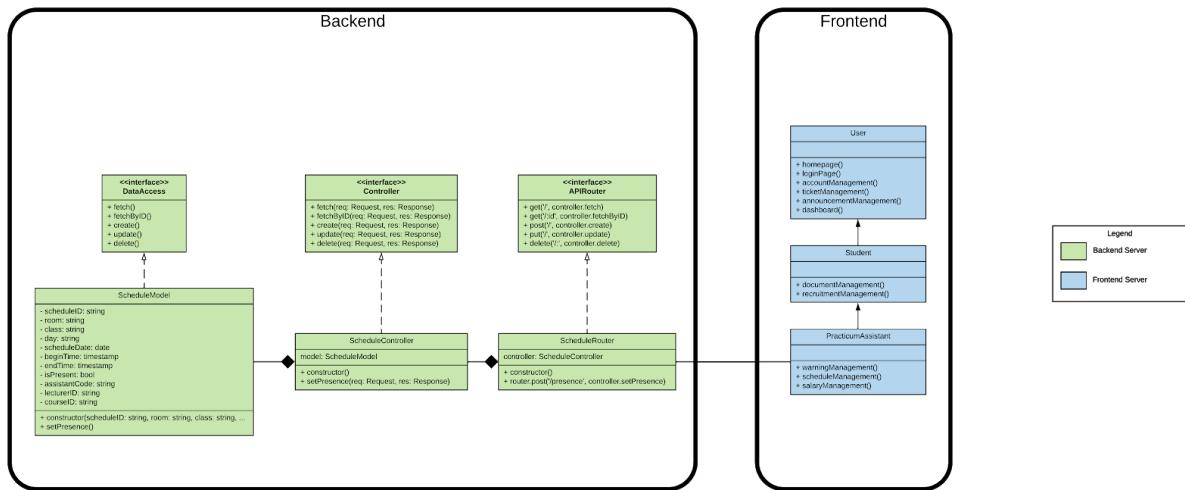
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	ScheduleRouter	Router (Backend)
5	ScheduleController	Controller (Backend)
6	ScheduleModel	Model (Backend)
7	DataAccess	DataAccessInterface (Backend)
8	APIController	ControllerInterface (Backend)
9	APIRouter	RouterInterface (Backend)

3.1.12.B Sequence Diagram



Picture 24 View Assistant Schedule Sequence Diagram

3.1.12.C Class Diagram



Picture 25 View Assistant Schedule Class Diagram

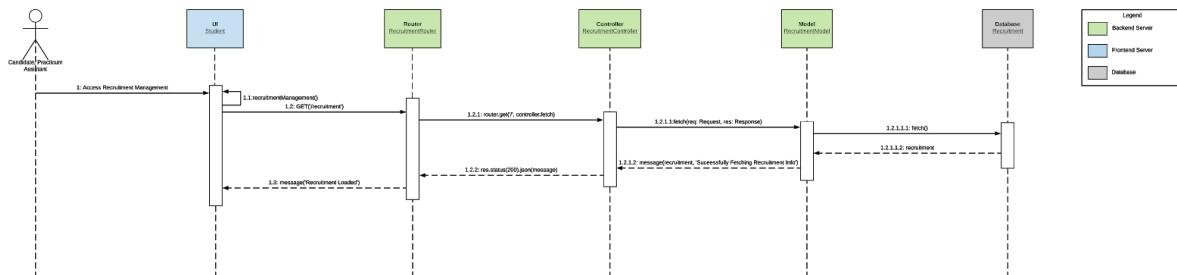
3.1.13 View Recruitment Info Management

3.1.13.A Class Identification

Table 14 View Recruitment Info Management Class Identification

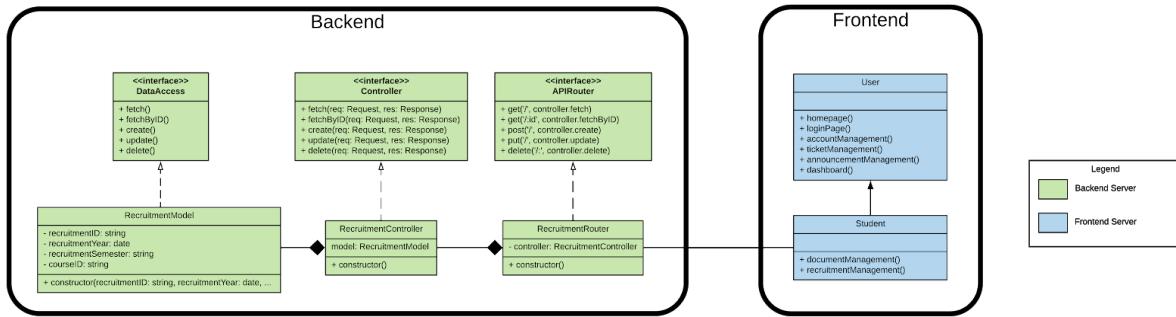
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	RecruitmentModel	Model (Backend)
4	RecruitmentController	Controller (Backend)
5	RecruitmentRouter	Router (Backend)
6	DataAccess	DataAccessInterface (Backend)
7	Controller	ControllerInterface (Backend)
8	APIRouter	RouterInterface (Backend)

3.1.13.B Sequence Diagram



Picture 26 View Recruitment Info Management Sequence Diagram

3.1.13.C Class Diagram



Picture 27 View Recruitment Info Management Class Diagram

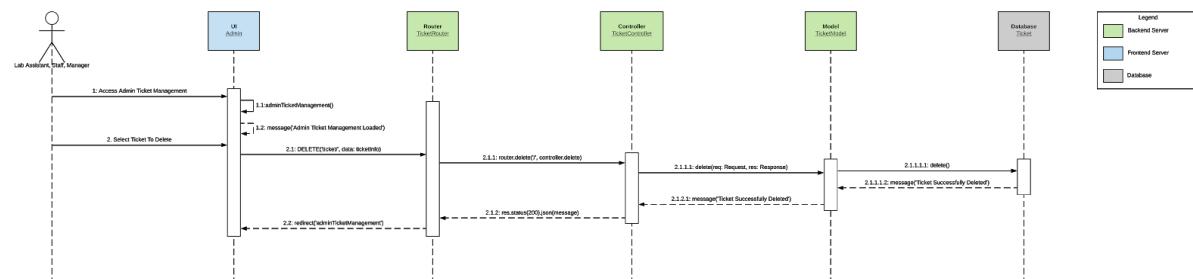
3.1.14 Delete Ticketing

3.1.14.A Class Identification

Table 15 Delete Ticketing Class Identification

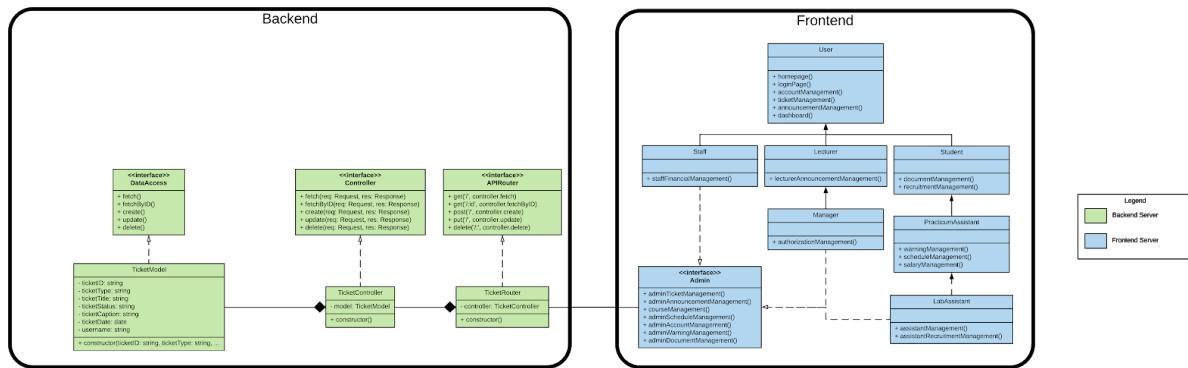
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	LabAssistant	UI (Frontend)
5	Lecturer	UI (Frontend)
6	Manager	UI (Frontend)
7	Staff	UI (Frontend)
8	Admin	UIInterface (Frontend)
9	TicketModel	Model (Backend)
10	TicketController	Controller (Backend)
11	TicketRouter	Router (Backend)
12	DataAccess	DataAccessInterface (Backend)
13	Controller	ControllerInterface (Backend)
14	APIRouter	RouterInterface (Backend)

3.1.14.B Sequence Diagram



Picture 28 Delete Ticketing Sequence Diagram

3.1.14.C Class Diagram



Picture 29 Delete Ticketing Class Diagram

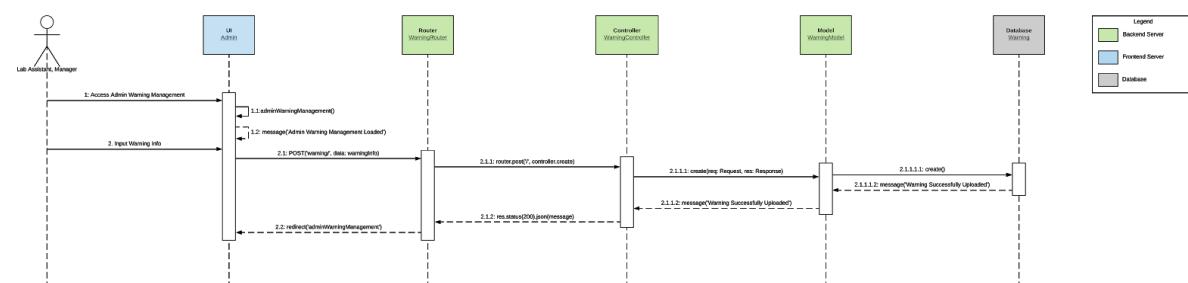
3.1.15 Input Warning

3.1.15.A Class Identification

Table 16 Input Warning Class Identification

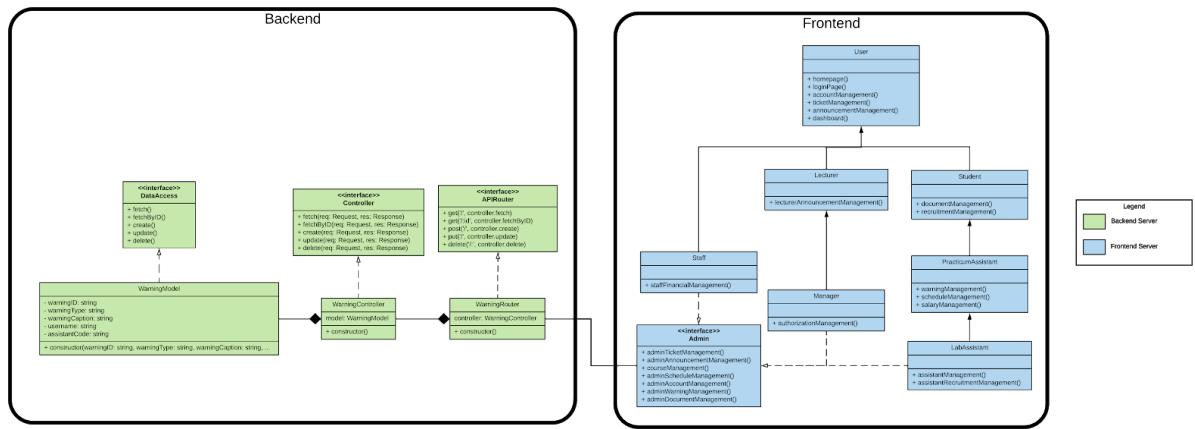
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	LabAssistant	UI (Frontend)
5	Lecturer	UI (Frontend)
6	Manager	UI (Frontend)
7	Staff	UI (Frontend)
8	Admin	UIInterface (Frontend)
9	WarningModel	Model (Backend)
10	WarningController	Controller (Backend)
11	WarningRouter	Router (Backend)
12	DataAccess	DataAccessInterface (Backend)
13	Controller	ControllerInterface (Backend)
14	APIRouter	RouterInterface (Backend)

3.1.15.B Sequence Diagram



Picture 30 Input Warning Sequence Diagram

3.1.15.C Class Diagram



Picture 31 Input Warning Class Diagram

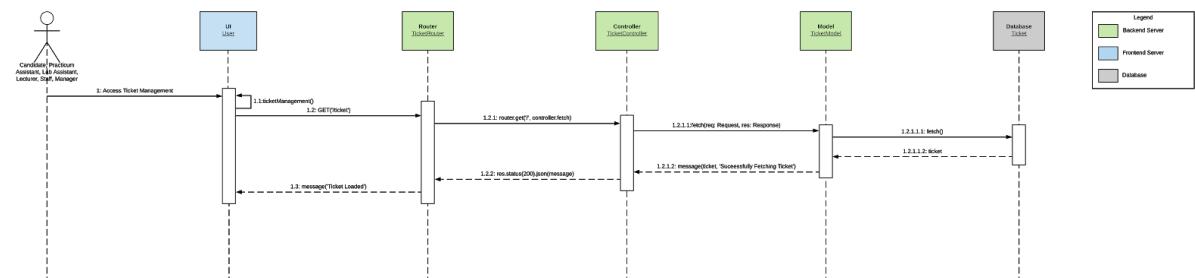
3.1.16 View Ticketing

3.1.16.A Class Identification

Table 17 View Ticketing Class Identification

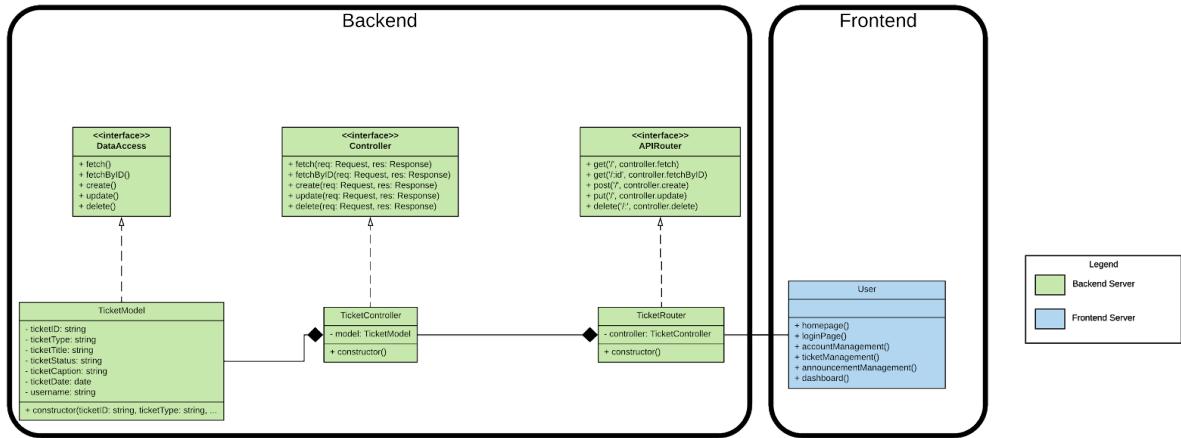
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	TicketModel	Model (Backend)
3	TicketController	Controller (Backend)
4	TicketRouter	Router (Backend)
5	DataAccess	DataAccessInterface (Backend)
6	Controller	ControllerInterface (Backend)
7	APIRouter	RouterInterface (Backend)

3.1.16.B Sequence Diagram



Picture 32 View Ticketing Sequence Diagram

3.1.16.C Class Diagram



Picture 33 View Ticketing Class Diagram

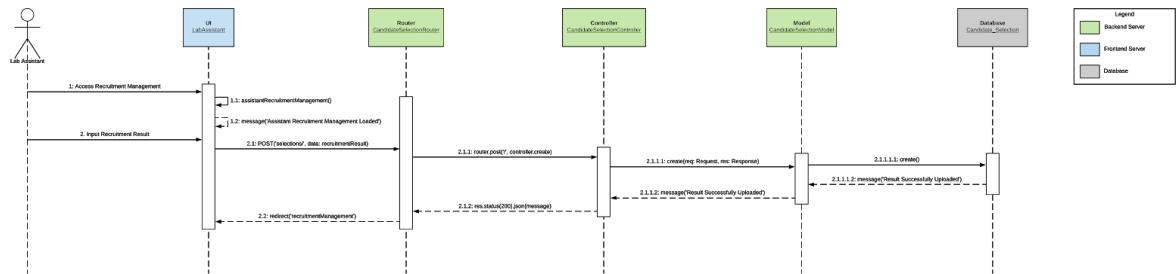
3.1.17 Input Recruitment Result

3.1.17.A Class Identification

Table 18 Input Recruitment Result Class Identification

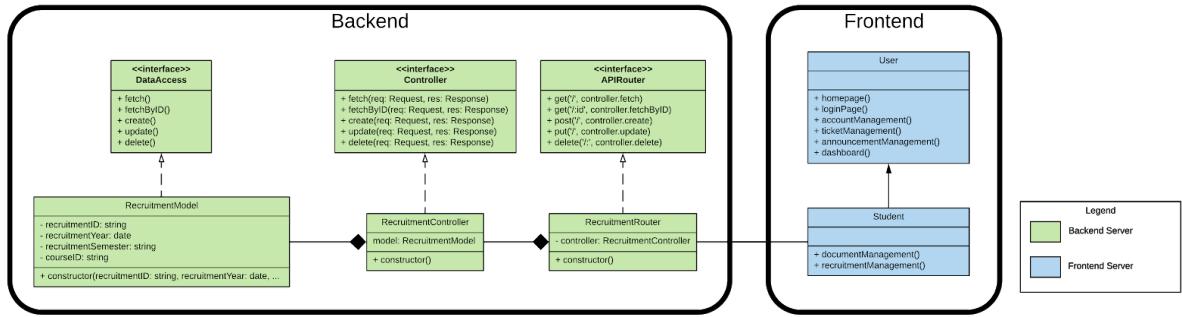
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	RecruitmentModel	Model (Backend)
4	RecruitmentController	Controller (Backend)
5	RecruitmentRouter	Router (Backend)
6	DataAccess	DataAccessInterface (Backend)
7	Controller	ControllerInterface (Backend)
8	APIRouter	RouterInterface (Backend)

3.1.17.B Sequence Diagram



Picture 34 Input Recruitment Result Sequence Diagram

3.1.17.C Class Diagram



Picture 35 Input Recruitment Result Class Diagram

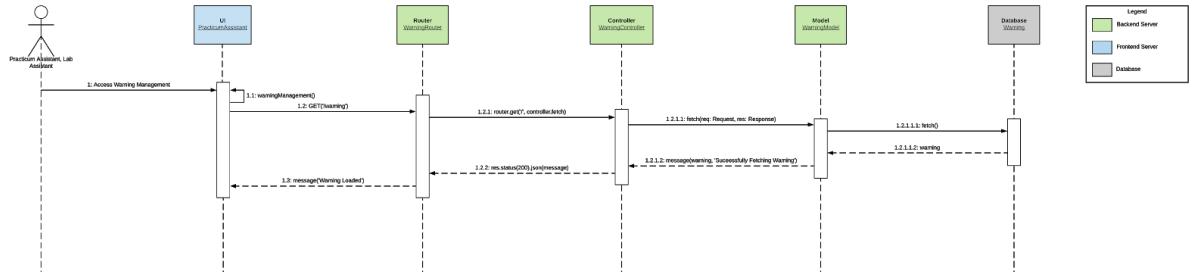
3.1.18 View Warning

3.1.18.A Class Identification

Table 19 View Warning Class Identification

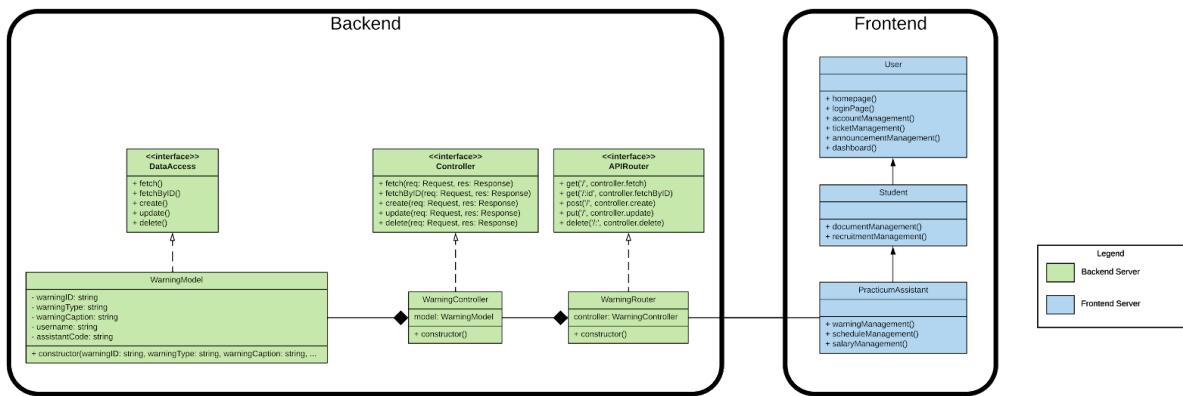
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	WarningModel	Model (Backend)
5	WarningController	Controller (Backend)
6	WarningRouter	Router (Backend)
7	DataAccess	DataAccessInterface (Backend)
8	Controller	ControllerInterface (Backend)
9	APIRouter	RouterInterface (Backend)

3.1.18.B Sequence Diagram



Picture 36 View Warning Sequence Diagram

3.1.18.C Class Diagram



Picture 37 View Warning Class Diagram

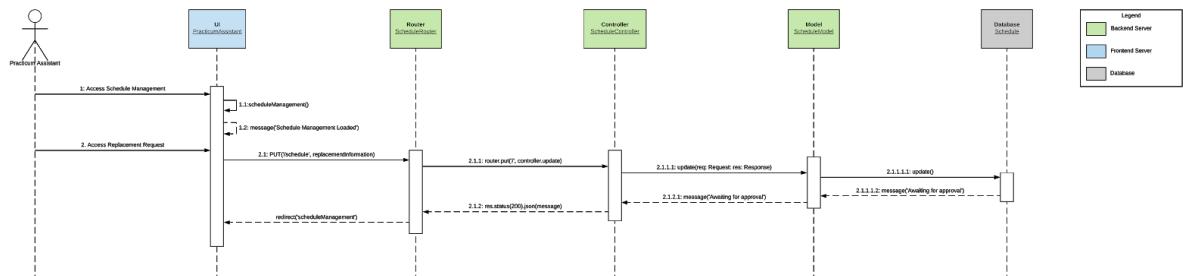
3.1.19 Request Replacement

3.1.19.A Class Identification

Table 20 Request Replacement Class Identification

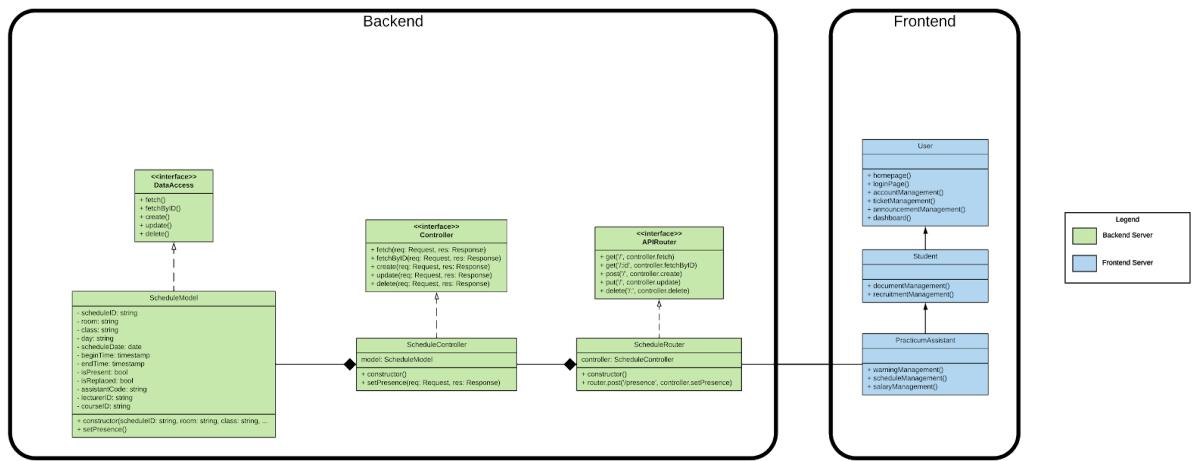
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	ScheduleRouter	Router (Backend)
5	ScheduleModel	Model (Backend)
6	ScheduleController	Controller (Backend)
7	DataAccess	DataAccessInterface (Backend)
8	Controller	ControllerInterface (Backend)
9	APIRouter	RouterInterface (Backend)

3.1.19.B Sequence Diagram



Picture 38 Request Replacement Sequence Diagram

3.1.19.C Class Diagram



Picture 39 Request Replacement Class Diagram

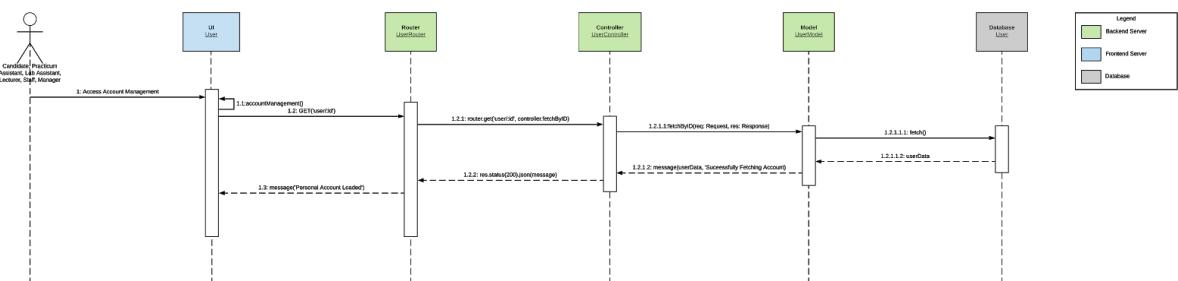
3.1.20 View Personal Account

3.1.20.A Class Identification

Table 21 View Personal Account Class Identification

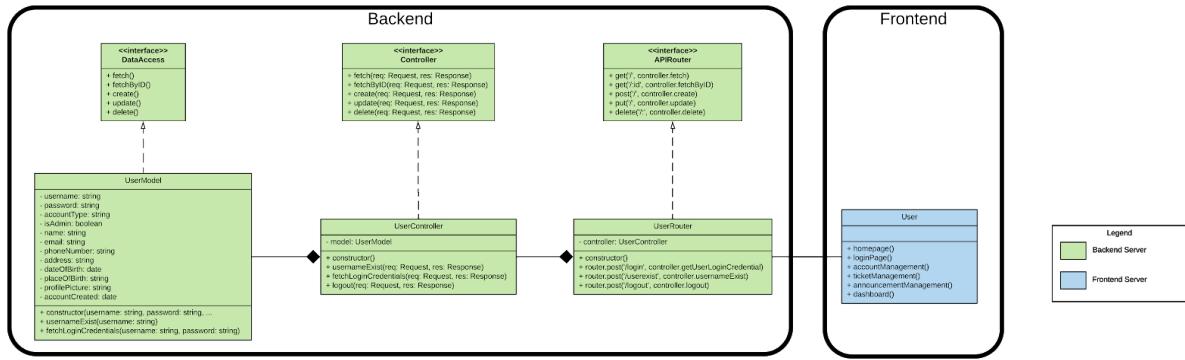
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	UserController	Controller (Backend)
3	UserModel	Model (Backend)
4	UserRouter	Router (Backend)
5	DataAccess	DataAccessInterface (Backend)
6	Controller	ControllerInterface (Backend)
7	APIRouter	RouterInterface (Backend)

3.1.20.B Sequence Diagram



Picture 40 View Personal Account Sequence Diagram

3.1.20.C Class Diagram



Picture 41 View Personal Account Class Diagram

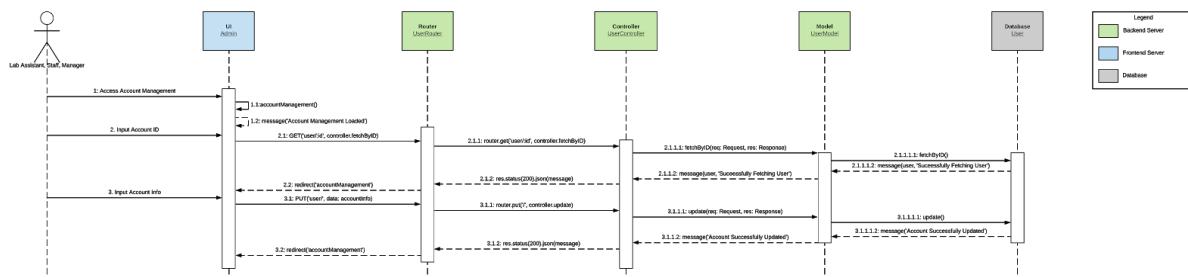
3.1.21 Manage User Account

3.1.21.A Class Identification

Table 22 Manage User Account Class Identification

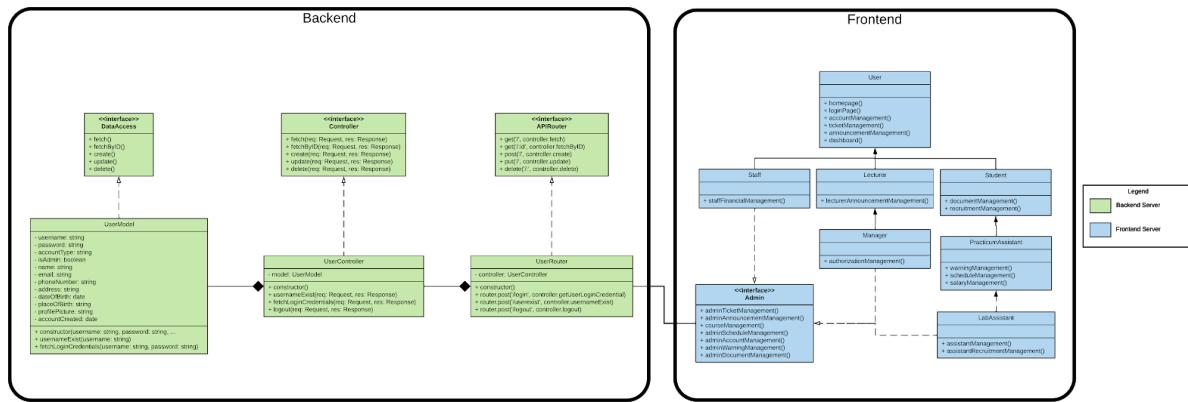
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	LabAssistant	UI (Frontend)
5	Lecturer	UI (Frontend)
6	Manager	UI (Frontend)
7	Staff	UI (Frontend)
8	Admin	UIInterface (Frontend)
9	UserController	Controller (Backend)
10	UserModel	Model (Backend)
11	UserRouter	Router (Backend)
12	DataAccess	DataAccessInterface (Backend)
13	Controller	ControllerInterface (Backend)
14	APIRouter	RouterInterface (Backend)

3.1.21.B Sequence Diagram



Picture 42 Manage User Account Sequence Diagram

3.1.21.C Class Diagram



Picture 43 Manage User Account Class Diagram

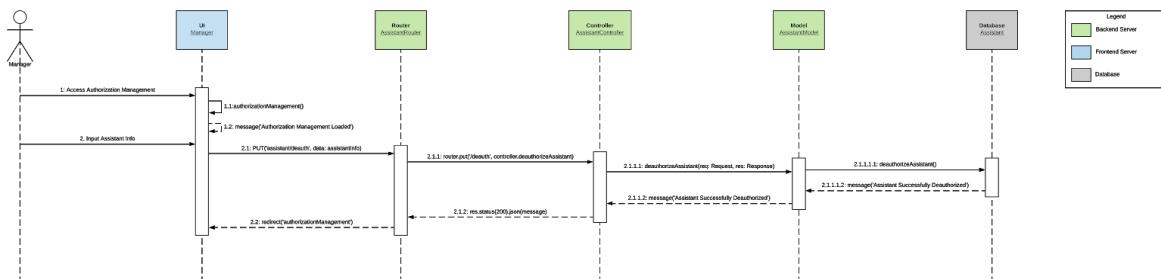
3.1.22 Deauthorize

3.1.22.A Class Identification

Table 23 Deauthorize Class Identification

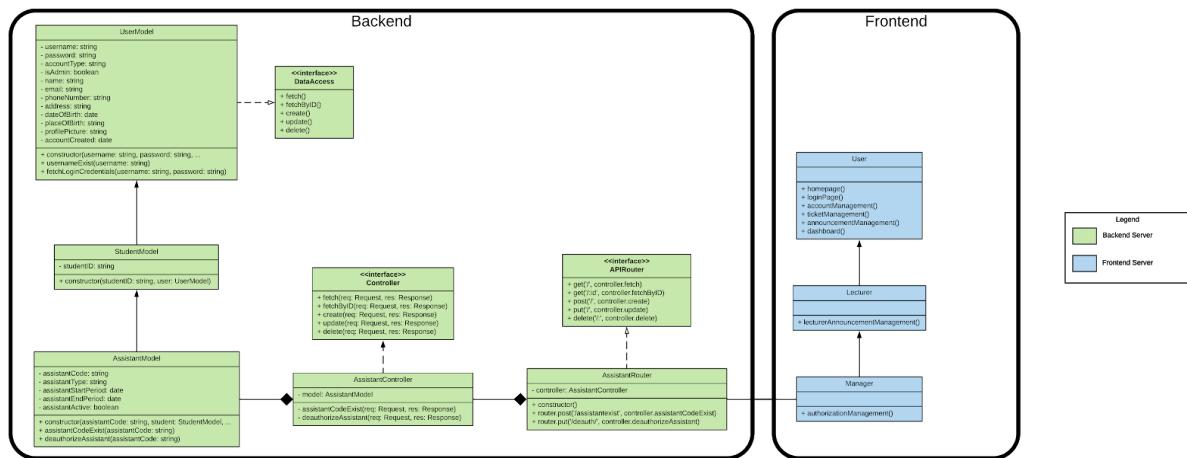
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Lecturer	UI (Frontend)
3	Manager	UI (Frontend)
4	AssistantRouter	Router (Backend)
5	AssistantModel	Model (Backend)
6	AssistantController	Controller (Backend)
7	UserModel	Model (Backend)
8	StudentModel	Model (Backend)
9	DataAccess	DataAccessInterface (Backend)
10	Controller	ControllerInterface (Backend)
11	APIRouter	RouterInterface (Backend)

3.1.22.B Sequence Diagram



Picture 44 Deauthorize Sequence Diagram

3.1.22.C Class Diagram



Picture 45 Deauthorize Class Diagram

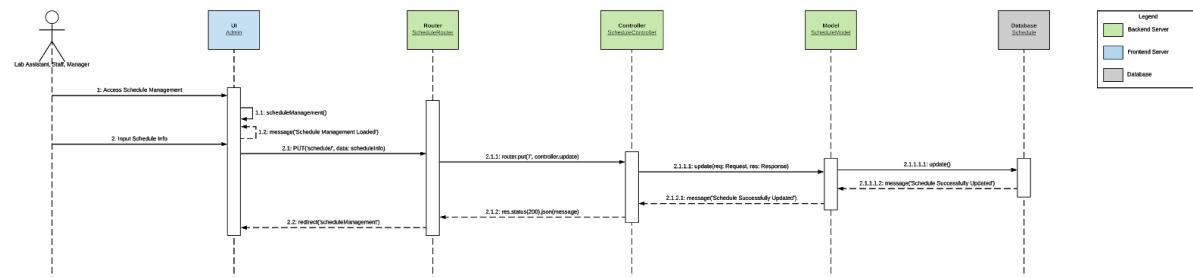
3.1.23 Confirm Replacement

3.1.23.A Class Identification

Table 24 Confirm Replacement Class Identification

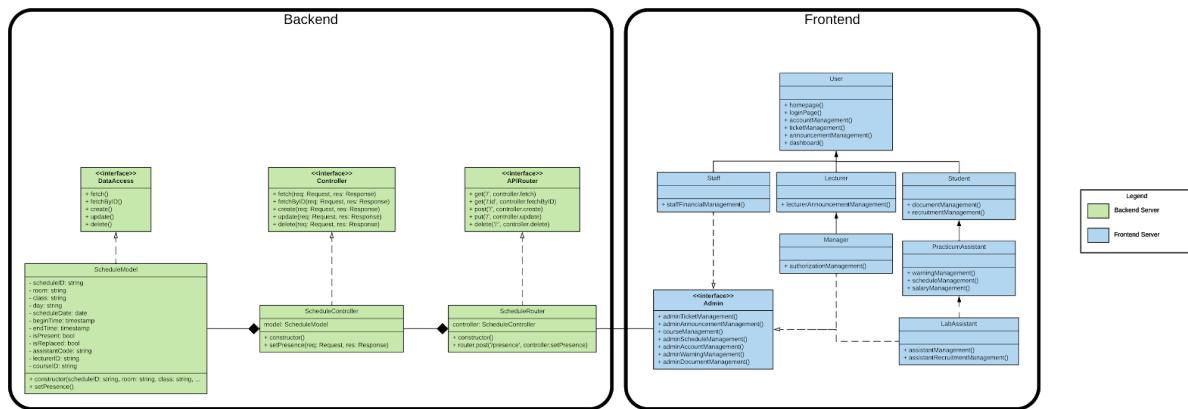
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	LabAssistant	UI (Frontend)
5	Lecturer	UI (Frontend)
6	Manager	UI (Frontend)
8	Staff	UI (Frontend)
9	Admin	UIInterface (Frontend)
10	ScheduleRouter	Router (Backend)
11	ScheduleModel	Model (Backend)
12	ScheduleController	Controller (Backend)
13	DataAccess	DataAccessInterface (Backend)
14	Controller	ControllerInterface (Backend)
15	APIRouter	RouterInterface (Backend)

3.1.23.B Sequence Diagram



Picture 46 Confirm Replacement Sequence Diagram

3.1.23.C Class Diagram



Picture 47 Confirm Replacement Class Diagram

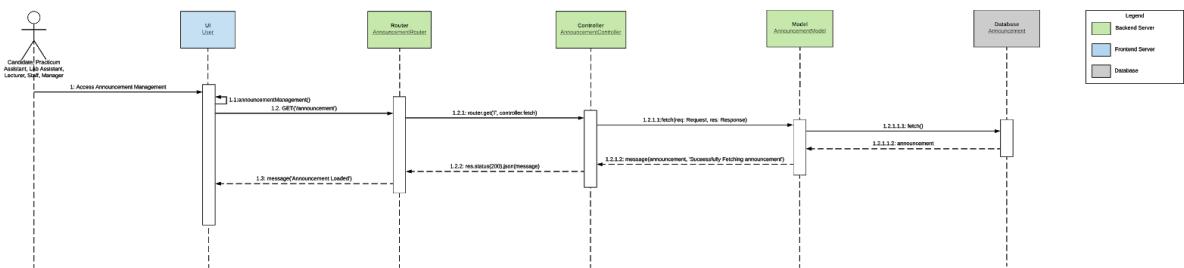
3.1.24 View Announcement

3.1.24.A Class Identification

Table 25 View Announcement Class Identification

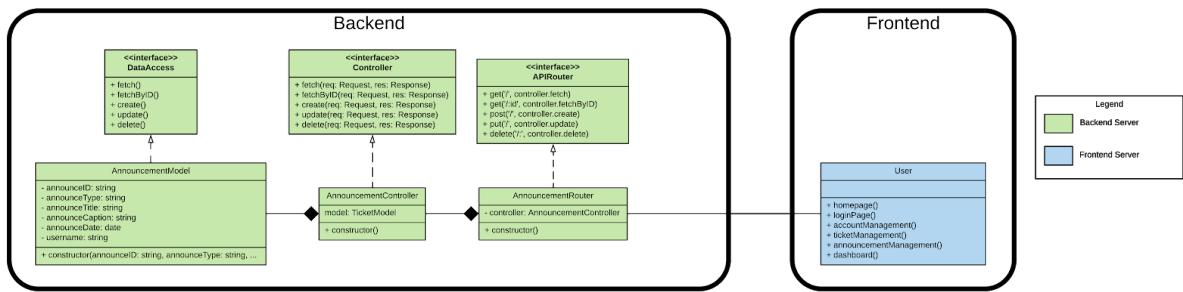
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	AnnouncementModel	Model (Backend)
3	AnnouncementRouter	Router (Backend)
4	AnnouncementController	Controller (Backend)
5	DataAccess	DataAccessInterface (Backend)
6	Controller	ControllerInterface (Backend)
7	APIRouter	RouterInterface (Backend)

3.1.24.B Sequence Diagram



Picture 48 View Announcement Sequence Diagram

3.1.24.C Class Diagram



Picture 49 View Announcement Class Diagram

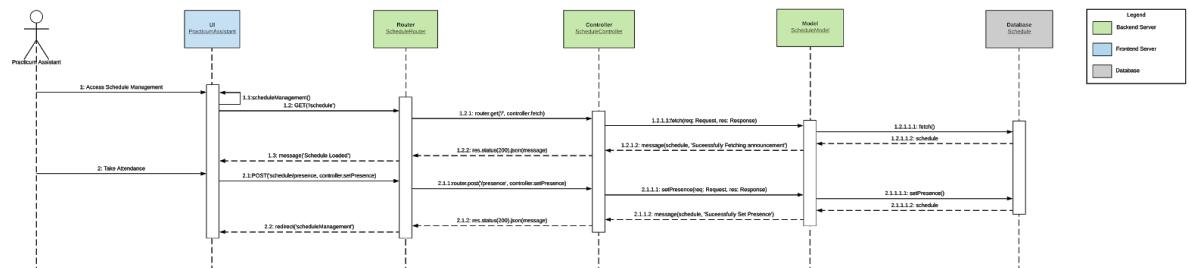
3.1.25 View and Confirm Attendance

3.1.25.A Class Identification

Table 26 View and Confirm Attendance Class Identification

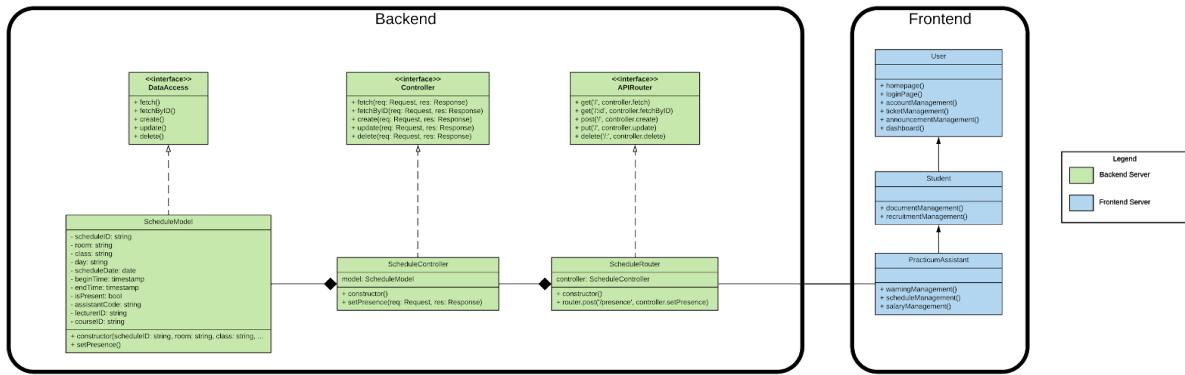
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	PracticumAssistant	UI (Frontend)
4	ScheduleModel	Model (Backend)
5	ScheduleController	Controller (Backend)
6	ScheduleRouter	Router (Backend)
7	DataAccess	DataAccessInterface (Backend)
8	Controller	ControllerInterface (Backend)
9	APIRouter	RouterInterface (Backend)

3.1.25.B Sequence Diagram



Picture 50 View and Confirm Attendance Sequence Diagram

3.1.25.C Class Diagram



Picture 51 View and Confirm Attendance Class Diagram

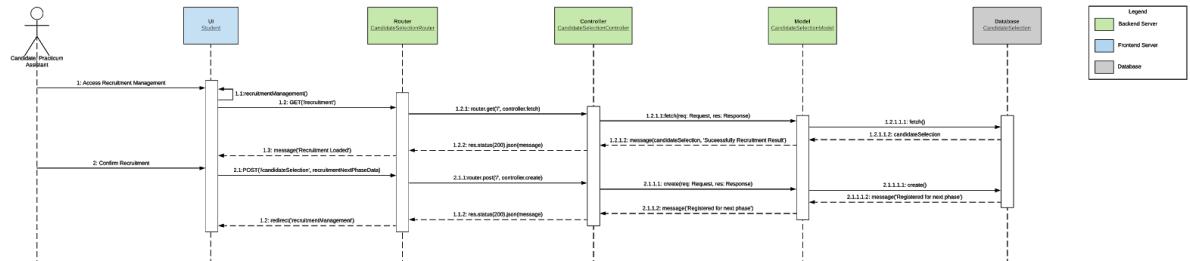
3.1.26 Confirm Candidate Next Phase

3.1.26.A Class Identification

Table 27 Confirm Candidate Next Phase Class Identification

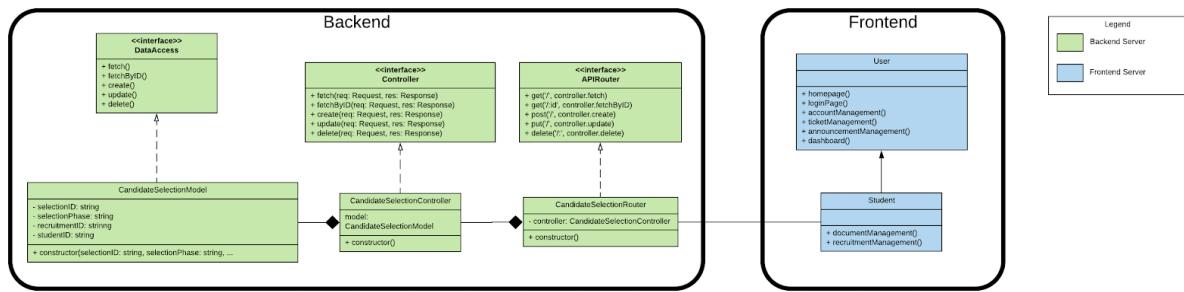
No	Design Class Name	Class Type
1	User	UI (Frontend)
2	Student	UI (Frontend)
3	CandidateSelectionRouter	Router (Backend)
4	CandidateSelectionModel	Model (Backend)
5	CandidateSelectionController	Controller (Backend)
6	DataAccess	DataAccessInterface (Backend)
7	Controller	ControllerInterface (Backend)
8	APIRouter	RouterInterface (Backend)

3.1.26.B Sequence Diagram



Picture 52 Confirm Candidate Next Phase Sequence Diagram

3.1.26.C Class Diagram



Picture 53 Confirm Candidate Next Phase Class Diagram

3.2 Detailed Class Design

Table 28 Detailed Class Design

No	Planning Class Name	Related Analysis Class Name
1	User	User
2	Admin	Admin
3	Staff	Staff
4	Lecturer	Lecturer
5	Manager	Manager
6	Student	Student
7	Candidate	Candidate
8	PracticumAssistant	PracticumAssistant
9	PracticumLab	PracticumLab
10	WarningController	WarningController
11	CandidateSelectionController	CandidateSelectionController
12	CommentController	CommentController
13	TicketReplyController	TicketReplyController
14	RecruitmentController	RecruitmentController
15	DocumentController	DocumentController
16	SalaryController	SalaryController
17	AnnouncementController	AnnouncementController
18	CourseController	CourseController
19	LectureController	LectureController
20	AssistantController	AssistantController
21	StudentController	StudentController
22	StaffController	StaffController
23	ManagerController	ManagerController
24	UserController	UserController
25	ScheduleController	ScheduleController
26	TicketController	TicketController
27	Controller	Controller
28	WarningModel	WarningModel
29	CandidateSelectionModel	CandidateSelectionModel
30	CommentModel	CommentModel

31	TicketReplyModel	TicketReplyModel
32	RecruitmentModel	RecruitmentModel
33	DocumentModel	DocumentModel
34	SalaryModel	SalaryModel
35	AnnouncementModel	AnnouncementModel
36	CourseModel	CourseModel
37	LectureModel	LectureModel
38	AssistantModel	AssistantModel
39	StudentModel	StudentModel
40	StaffModel	StaffModel
41	ManagerModel	ManagerModel
42	UserModel	UserModel
43	ScheduleModel	ScheduleModel
44	TicketModel	TicketModel
45	Model	Model
46	WarningRouter	WarningRouter
47	CandidateSelectionRouter	CandidateSelectionRouter
48	CommentRouter	CommentRouter
49	TicketReplyRouter	TicketReplyRouter
50	RecruitmentRouter	RecruitmentRouter
51	DocumentRouter	DocumentRouter
52	SalaryRouter	SalaryRouter
53	AnnouncementRouter	AnnouncementRouter
54	CourseRouter	CourseRouter
55	LectureRouter	LectureRouter
56	AssistantRouter	AssistantRouter
57	StudentRouter	StudentRouter
58	StaffRouter	StaffRouter
59	ManagerRouter	ManagerRouter
60	UserRouter	UserRouter
61	ScheduleRouter	ScheduleRouter
62	TicketRouter	TicketRouter
63	Router	Router

3.2.1 Class <User>

Class Name : User

Table 29 Class <User>

Operation Name	Visibility (private, public)	Description
homepage()	Public	User interface for homepage.
loginpage()	Public	User interface for login page.
accountManagement()	Public	User interface for personal account management.

ticketManagement()	Public	User interface for managing tickets.
announcementManagement()	Public	User interface for managing, in this case viewing and commenting announcements for users.
dashboard()	Public	User interface for accessing the features tied to the account role.
Attribute Name	Visibility (private, public)	Type

3.2.2 Class <Admin>

Class Name : Admin

Table 30 Class <Admin>

Operation Name	Visibility (private, public)	Description
adminTicketManagement()	Public	User interface to manage ticket (view and delete)
AdminAnnounceManagement()	Public	User interface to manage announcement (create) by admin
CourseManagement	Public	User interface to manage course
adminScheduleManagement()	Public	User interface to manage Schedule by admin
adminAccountManagement()	Public	User interface to manage account by admin
adminWarningManagement()	Public	User interface to manage Warning by admin
adminDocumentManagement()	Public	User interface to manage Document by admin
Attribute Name	Visibility (private, public)	Type

3.2.3 Class <Staff>

Class Name : Staff

Table 31 Class <Staff>

Operation Name	Visibility (private, public)	Description
staffFinancialManagement()	Public	User interface to manage assistant financial
Attribute Name	Visibility (private, public)	Type

3.2.4 Class <Lecturer>

Class Name : Lecturer

Table 32 Class <Lecturer>

Operation Name	Visibility (private, public)	Description
lecturerAnnouncementManagement()	Public	User interface to manage lecturers announcement
Attribute Name	Visibility (private, public)	Type

3.2.5 Class <Student>

Class Name : Student

Table 33 Class <Student>

Operation Name	Visibility (private, public)	Description
documentManagement()	Public	User interface to manage document
recruitmentManagement()	Public	User interface to manage recruitment
Attribute Name	Visibility (private, public)	Type

3.2.6 Class <Manager>

Class Name : Manager

Table 34 Class <Manager>

Operation Name	Visibility (private, public)	Description
authorizationManagement()	Public	User interface to handle manager authorization approval
Attribute Name	Visibility (private, public)	Type

--	--	--

3.2.7 Class <Candidate>

Class Name : Candidate

Table 35 Class <Candidate>

Operation Name	Visibility (private, public)	Description
registrationForm()	Public	User interface for candidate registration form
Attribute Name	Visibility (private, public)	Type

3.2.8 Class <PracticumAssistant>

Class Name : PracticumAssistant

Table 36 Class <Practicum Assistant>

Operation Name	Visibility (private, public)	Description
warningManagement()	Public	User interface for practicum assistant warning management
scheduleManagement()	Public	User interface for practicum assistant schedule management
salaryManagement()	Public	User interface for practicum assistant salary management
Attribute Name	Visibility (private, public)	Type

3.2.9 Class <LabAssistant>

Class Name : LabAssistant

Table 37 Class <Lab Assistant>

Operation Name	Visibility (private, public)	Description
assistantManagement()	Public	User interface for managing assistants
assistantRecruitmentManagement()	Public	User interface for managing recruitment on candidates or extending assistants
Attribute Name	Visibility (private, public)	Type

Prodi S1 Informatics Tel-U	SDD-0002	Page 50 dari 111
The template of this document and the information it holds belongs to Tel-U Informatics S1 Study Program and is confidential. Reproduction of this document is prohibited without the knowledge of the Tel-U Informatics Study Program.		

3.2.10 Class <WarningController>

Class Name : WarningController

Table 38 Class <WarningController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder warning controller
Attribute Name	Visibility (private, public)	Type
WarningModel	Public	Model

3.2.11 Class <CandidateSelectionController>

Class Name : CandidateSelectionController

Table 39 Class <CandidateSelectionController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Candidate Selection Controller
Attribute Name	Visibility (private, public)	Type
CandidateSelectionModel	Public	Model

3.2.12 Class <CommentController>

Class Name : CommentController

Table 40 Class <CommentController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Comment Controller
Attribute Name	Visibility (private, public)	Type
CommentModel	Public	Model

3.2.13 Class <TicketReplyController>

Class Name : TicketReplyController

Table 41 Class <TicketReplyController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Ticket Reply Controller
Attribute Name	Visibility (private, public)	Type
TicketReplyModel	Public	Model

3.2.14 Class <RecruitmentController>

Class Name : RecruitmentController

Table 42 Class <RecruitmentController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Recruitment Controller
Attribute Name	Visibility (private, public)	Type
RecruitmentModel	Public	Model

3.2.15 Class <DocumentController>

Class Name : DocumentController

Table 43 Class <DocumentController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Document Controller
Attribute Name	Visibility (private, public)	Type
DocumentModel	Public	Model

3.2.16 Class <SalaryController>

Class Name : SalaryController

Table 44 Class <SalaryController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder salary controller
Attribute Name	Visibility (private, public)	Type
SalaryModel	Public	Model

3.2.17 Class <AnnouncementController>

Class Name : AnnouncementController

Table 45 Class <AnnouncementController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder announcement controller
Attribute Name	Visibility (private, public)	Type
AnnouncementModel	Public	Model

3.2.18 Class <CourseController>

Class Name : CourseController

Table 46 Class <CourseController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder course controller
Attribute Name	Visibility (private, public)	Type
CourseController	Public	Model

3.2.19 Class <LecturerController>

Class Name : LecturerController

Table 47 Class <LecturerController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder lecturer controller
Attribute Name	Visibility (private, public)	Type
LecturerModel	Public	Model

3.2.20 Class <AssistantController>

Class Name : AssistantController

Table 48 Class <AssistantController>

Operation Name	Visibility (private, public)	Description
assistantCodeExist(req: Request, res: Response)	Public	Method for finding an assistantCode in the system
deauthorizeAssistant(req: Request, res: Response)	Public	Method to deauthorize assistant whom no longer in IFLAB
Attribute Name	Visibility (private, public)	Type
AssistantModel	Public	Model

3.2.21 Class <StudentController>

Class Name : StudentController

Table 49 Class <StudentController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder student controller
Attribute Name	Visibility	Type

	(private, public)	
StudentModel	Public	Model

3.2.22 Class <StaffController>

Class Name : StaffController

Table 50 Class <StaffController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder staff controller
Attribute Name	Visibility (private, public)	Type
staffModel	Public	Model

3.2.23 Class <ManagerController>

Class Name : ManagerController

Table 51 Class <ManagerController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder manage controller
Attribute Name	Visibility (private, public)	Type
ManagerModel	Public	Model

3.2.24 Class <UserController>

Class Name : UserController

Table 52 Class <UserController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder user controller
usernameExist(req: Request, res: Response)	Public	Method for finding an existing username in the system
fetchLoginCredentials(req: Request, res: Response)	Public	Method for getting loginInfo
logout(req: Request, res: Response)	Public	Method For logging out from the account
Attribute Name	Visibility (private, public)	Type
UserModel	Public	Model

3.2.25 Class <ScheduleController>

Class Name : ScheduleController

Table 53 Class <ScheduleController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder schedule control
setPresence(req: Request, res: Response)	Public	Method for setting presence in the schedule
Attribute Name	Visibility (private, public)	Type
ScheduleModel	Public	Model

3.2.26 Class <TicketController>

Class Name : TicketController

Table 54 Class <TicketController>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder ticket model
Attribute Name	Visibility (private, public)	Type
TicketModel	Public	Model

3.2.27 Class <Controller>

Class Name : Controller

Table 55 Class <Controller>

Operation Name	Visibility (private, public)	Description
fetch(req: Request, res: Response)	Public	Method used for handling requests and response on fetching data from the database.
fetchByID(req: Request, res: Response)	Public	Method used for handling requests and response on fetching data from database based on unique identifiers,
create(req: Request, res: Response)	Public	Method used for handling requests and response on creating data to the database.
update(req: Request, res: Response)	Public	Method used for handling requests and responses on updating data from the database.
delete(req: Request, res: Response)	Public	Method used for handling requests and responses on

		deleting data from the database
Attribute Name	Visibility (private, public)	Type

3.2.28 Class <WarningModel>

Class Name : WarningModel

Table 56 Class <WarningModel>

Operation Name	Visibility (private, public)	Description
constructor(warningID: string, warningType: string, warningCaption: string, ...)	Public	Construct object builder warning model
Attribute Name	Visibility (private, public)	Type
warningID	Public	String
warningType	Public	String
warningCaption	Public	String
username	Public	String
assistantCode	Public	String

3.2.29 Class <CandidateSelectionModel>

Class Name : CandidateSelectionModel

Table 57 Class <Candidate Selection Model>

Operation Name	Visibility (private, public)	Description
constructor(selectionID: string, selectionPhase: string, ...)	Public	Construct object builder candidate selection model
Attribute Name	Visibility (private, public)	Type
selectionID	Public	String
selectionPhase	Public	String
recruitmentID	Public	String
studentID	Public	String

3.2.30 Class <CommentModel>

Class Name : CommentModel

Table 58 Class <CommentModel>

Operation Name	Visibility (private, public)	Description
constructor(commentID: string, commentCaption: string, ...)	Public	Construct object builder comment model
Attribute Name	Visibility (private, public)	Type

commentID	Public	String
commentCaption	Public	String
commentDate	Public	date
announcementID	Public	String
username	Public	String

3.2.31 Class <TicketReplyModel>

Class Name : TicketReplyModel

Table 59 Class <TicketReplyModel>

Operation Name	Visibility (private, public)	Description
constructor(replyID: string, replyCaption: string, ...)	Public	Construct object builder ticket reply model
Attribute Name	Visibility (private, public)	Type
replyID	Public	String
replyCaption	Public	String
replyDate	Public	date
ticketID	Public	String
username	Public	String

3.2.32 Class <RecruitmentModel>

Class Name : RecruitmentModel

Table 60 Class <RecruitmentModel>

Operation Name	Visibility (private, public)	Description
constructor(replyID: string, replyCaption: string, ...)	Public	Construct object builder recruitment model
Attribute Name	Visibility (private, public)	Type
recruitmentID	Public	String
recruitmentYear	Public	String
recruitmentSemester	Public	date
courseID	Public	String

3.2.33 Class <DocumentModel>

Class Name : DocumentModel

Table 61 Class <DocumentModel>

Operation Name	Visibility (private, public)	Description
constructor(documentID: string, documentName: string, ...)	Public	Construct object builder document model
Attribute Name	Visibility (private, public)	Type
documentID	Public	string
documentName	Public	string

documentType	Public	string
documentSize	Public	number
documentUploadDate	Public	date

3.2.34 Class <SalaryModel>

Class Name : SalaryModel

Table 62 Class <SalaryModel>

Operation Name	Visibility (private, public)	Description
constructor(salaryID: string, assistantCode: string, staffID: string, ...)	Public	Construct object builder salary model
Attribute Name	Visibility (private, public)	Type
salaryID	Public	string
assistantCode	Public	string
staffID	Public	string
totalSalary	Public	integer
availableSalary	Public	integer
takenSalary	Public	integer
availableDate	Public	date
takenDate	Public	date

3.2.35 Class <AnnouncementModel>

Class Name : AnnouncementModel

Table 63 Class <AnnouncementModel>

Operation Name	Visibility (private, public)	Description
constructor(announceID: string, announceType: string, ...)	Public	Construct object builder announcement model
Attribute Name	Visibility (private, public)	Type
announceID	Public	string
announceType	Public	string
announceTitle	Public	string
announceCaption	Public	string
announceDate	Public	date
username	Public	string

3.2.36 Class <CourseModel>

Class Name : CourseModel

Table 64 Class <CourseModel>

Operation Name	Visibility (private, public)	Description
constructor(courseID: string, courseTitle: string, ...)	Public	Construct object builder course model

Attribute Name	Visibility (private, public)	Type
courseID	Public	string
courseTitle	Public	string
courseYear	Public	date
lecturerID	Public	string

3.2.37 Class <LecturerModel>

Class Name : LecturerModel

Table 65 Class <LecturerModel>

Operation Name	Visibility (private, public)	Description
constructor(lecturerID: string, user: UserModel, ...)	Public	Construct object builder lecturer model
Attribute Name	Visibility (private, public)	Type
lecturerID	Public	string
lecturerExpertise	Public	string

3.2.38 Class <AssistantModel>

Class Name : AssistantModel

Table 66 Class <AssistantModel>

Operation Name	Visibility (private, public)	Description
constructor(assistantCode: string, student: StudentModel, ...)	Public	Construct assistant model
assistantCodeExist(assistantCode: string)	Public	Checking an existing assistantCode
deauthorizeAssistant(assistantCode: string)	Public	Deauthorize assistants based on their assistant code.
Attribute Name	Visibility (private, public)	Type
assistantCode	Public	string
assistantType	Public	string
assistantStartPeriod	Public	date
assistantEndPeriod	Public	date
assistantActive	Public	boolean

3.2.39 Class <StudentModel>

Class Name : StudentModel

Table 67 Class <StudentModel>

Operation Name	Visibility (private, public)	Description
constructor(studentID: string, user: UserModel)	Public	Construct student model
Attribute Name	Visibility	Type

	(private, public)	
studentID	Public	string

3.2.40 Class <StaffModel>

Class Name : StaffModel

Table 68 Class <StaffModel>

Operation Name	Visibility (private, public)	Description
constructor(staffID: string, user: UserModel)	Public	Construct object builder staff model
Attribute Name	Visibility (private, public)	Type
staffID	Public	string

3.2.41 Class <ManagerModel>

Class Name : ManagerModel

Table 69 Class <ManagerModel>

Operation Name	Visibility (private, public)	Description
constructor(managerID: string, lecturer: LecturerModel, ...)	Public	Construct object builder manager model
Attribute Name	Visibility (private, public)	Type
managerID	Public	string
managerStartPeriod	Public	date
managerEndPeriod	Public	date

3.2.42 Class <UserModel>

Class Name : UserModel

Table 70 Class <UserModel>

Operation Name	Visibility (private, public)	Description
constructor(username: string, password: string, ...)	Public	Construct object builder user model
usernameExist(username: string)	Public	Checking if a username exists in the database.
fetchLoginCredentials(username: string, password: string)	Public	Generating login credentials in tokens for users if the login information is correct.
Attribute Name	Visibility (private, public)	Type
username	Public	string
password	Public	string
accountType	Public	string
isAdmin	Public	boolean
name	Public	string

email	Public	string
phoneNumber	Public	string
address	Public	string
dateOfBirth	Public	date
placeOfBirth	Public	string
profilePicture	Public	string
accountCreated	Public	date

3.2.43 Class <ScheduleModel>

Class Name : ScheduleModel

Table 71 Class <ScheduleModel>

Operation Name	Visibility (private, public)	Description
constructor(scheduleID: string, room: string, class: string, ...)	Public	Construct object builder for schedule model
setPresence()	Public	to set the presence on the schedule
Attribute Name	Visibility (private, public)	Type
scheduleID	Public	string
room	Public	string
class	Public	string
day	Public	string
scheduleDate	Public	date
beginTime	Public	timestamp
endTime	Public	timestamp
isPresent	Public	bool
isReplaced	Public	bool
assistantCode	Public	string
lecturerID	Public	string
courseID	Public	string

3.2.44 Class <TicketModel>

Class Name : TicketModel

Table 72 Class <TicketModel>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for ticket model
setPresence(req: Request, res: Response)	Public	Set assistant presence to the database.
Attribute Name	Visibility (private, public)	Type
ticketID	Public	string
ticketType	Public	string

ticketTitle	Public	string
ticketStatus	Public	string
ticketCaption	Public	string
ticketDate	Public	date
username	Public	string

3.2.45 Class <DataAccess>

Class Name : DataAccess

Table 73 Class <DataAccess>

Operation Name	Visibility (private, public)	Description
fetch()	Public	Fetching data from the database
fetchByID()	Public	Fetching data from the database based on the defined unique identifiers of the model..
create()	Public	Creates data based on the defined model.
update()	Public	Updates data based on the defined model.
delete()	Public	Deletes data based on the defined model.
Attribute Name	Visibility (private, public)	Type

3.2.46 Class <WarningRouter>

Class Name : WarningRouter

Table 74 Class <WarningRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for warning router.
Attribute Name	Visibility (private, public)	Type
WarningController	Public	Controller

3.2.47 Class <CandidateSelectionRouter>

Class Name : CandidateSelectionRouter

Table 75 Class <CandidateSelectionRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for candidate selection router.
Attribute Name	Visibility (private, public)	Type
CandidateSelectionController	Public	Controller

3.2.48 Class <CommentRouter>

Class Name : CommentRouter

Table 76 Class <CommentRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for comment router.
Attribute Name	Visibility (private, public)	Type
CommentController	Public	Controller

3.2.49 Class <TicketReplyRouter>

Class Name : TicketReplyRouter

Table 77 Class <TicketReplyRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for ticket reply
Attribute Name	Visibility (private, public)	Type
TicketReplyController	Public	Controller

3.2.50 Class <RecruitmentRouter>

Class Name : RecruitmentRouter

Table 78 Class <RecruitmentRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for recruitment
Attribute Name	Visibility (private, public)	Type
RecruitmentController	Public	Controller

3.2.51 Class <DocumentRouter>

Class Name : DocumentRouter

Table 79 Class <Document Router>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Document.
Attribute Name	Visibility (private, public)	Type
DocumentController	Public	Controller

3.2.52 Class <SalaryRouter>

Class Name : SalaryRouter

Table 80 Class <SalaryRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Salary.
Attribute Name	Visibility (private, public)	Type
SalaryController	Public	Controller

3.2.53 Class <AnnouncementRouter>

Class Name : AnnouncementRouter

Table 81 Class <AnnouncementRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Announcement
Attribute Name	Visibility (private, public)	Type
AnnouncementController	Public	Controller

3.2.54 Class <CourseRouter>

Class Name : CourseRouter

Table 82 Class <Courses Router>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Course
Attribute Name	Visibility (private, public)	Type
CourseController	Public	Controller

3.2.55 Class <LecturerRouter>

Class Name : LecturerRouter

Table 83 Class <LecturerRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder Lecturer
Attribute Name	Visibility (private, public)	Type
LecturerController	Public	Controller

3.2.56 Class <AssistantRouter>

Class Name : AssistantRouter

Table 84 Class <AssistantRouter>

Operation Name	Visibility (private, public)	Description
constructor()		Construct object builder Assistant
router.post('/assistantexist', controller.assistantCodeExist)	Public	URL Routing for receiving request and response on checking existing assistant code
router.put('/deauth/', controller.deauthorizeAssista nt)	Public	URL Routing for receiving request and response on assistant deauthorization
Attribute Name	Visibility (private, public)	Type
AssistantController	Public	Controller

3.2.57 Class <StudentRouter>

Class Name : StudentRouter

Table 85 Class <StudentRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder Student
Attribute Name	Visibility (private, public)	Type
StudentController	Public	Controller

3.2.58 Class <StaffRouter>

Class Name : StaffRouter

Table 86 Class <StaffRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder Staff
Attribute Name	Visibility (private, public)	Type
staffController	Public	Controller

3.2.59 Class <ManagerRouter>

Class Name : ManagerRouter

Table 87 Class <ManagerRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder Manage

Attribute Name	Visibility (private, public)	Type
ManagerModel	Public	Model

3.2.60 Class <UserRouter>

Class Name : UserRouter

Table 88 Class <UserRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder User
router.post('/login', controller.getUserLoginCredential)	Public	URL Routing for receiving request and response on login credentials.
router.post('/userexist', controller.usernameExist)	Public	URL Routing for receiving request and response on checking existing username.
router.post('/logout', controller.logout)	Public	URL Routing for receiving request and response on logging out.
Attribute Name	Visibility (private, public)	Type
UserController	Public	Controller

3.2.61 Class <ScheduleRouter>

Class Name : ScheduleRouter

Table 89 Class <ScheduleRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Schedule
router.post('/presence', controller.setPresence)	Public	URL Routing for receiving request and response on setting assistant presence.
Attribute Name	Visibility (private, public)	Type
ScheduleController	Public	Controller

3.2.62 Class <TicketRouter>

Class Name : TicketRouter

Table 90 Class <TicketRouter>

Operation Name	Visibility (private, public)	Description
constructor()	Public	Construct object builder for Ticket
Attribute Name	Visibility (private, public)	Type
TicketController	Public	Controller

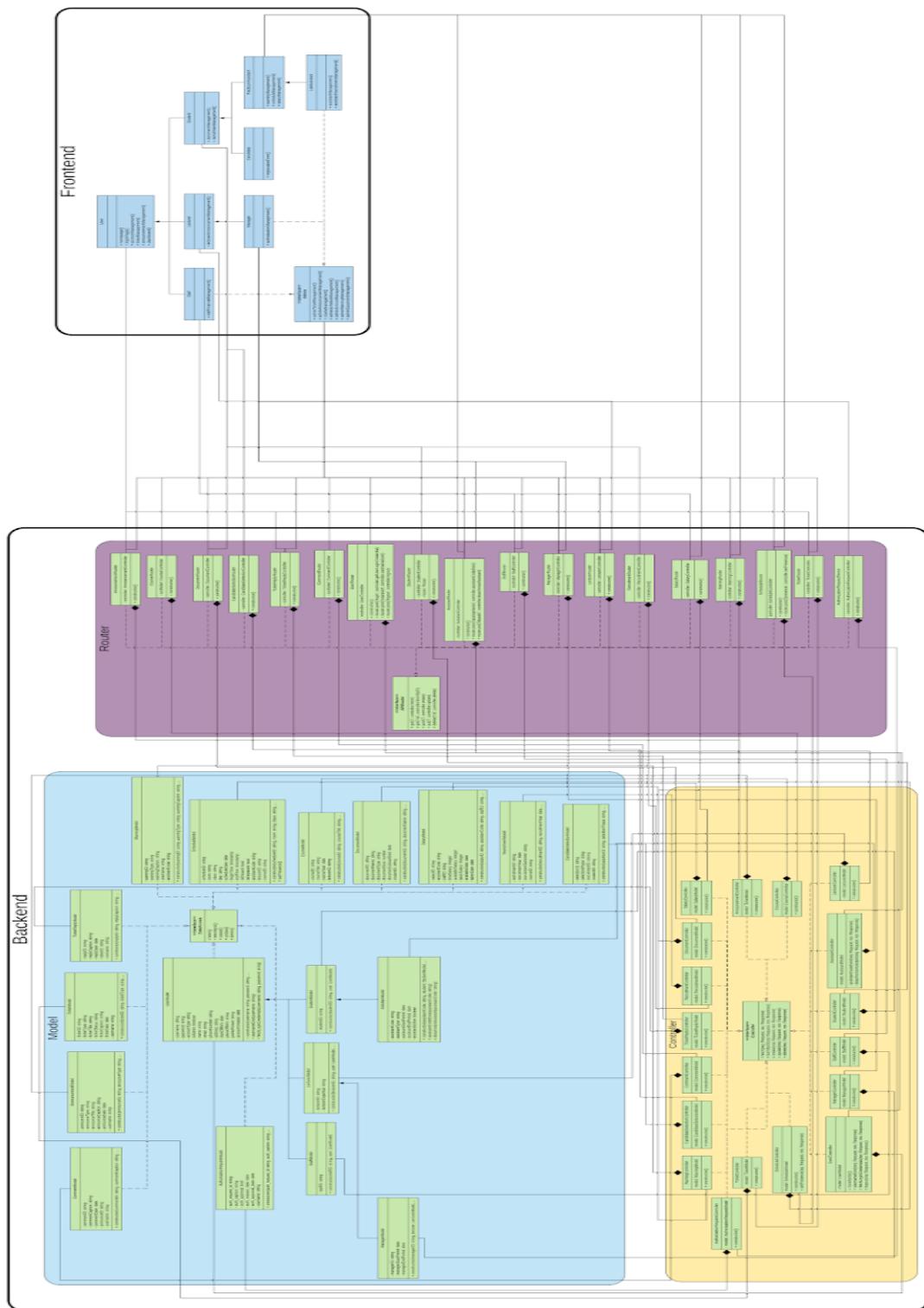
3.2.63 Class <APIRouter>

Class Name : APIRouter

Table 91 Class <APIRouter>

Operation Name	Visibility (private, public)	Description
get('/', controller.fetch)	Public	URL Routing for receiving requests and sending responses on fetching data from model.
get('/:id', controller.fetchByID)	Public	URL Routing for receiving requests and sending responses on fetching data from the model based on the unique identifier of the model.
post('/', controller.create)	Public	URL Routing for receiving requests and sending responses on sending data to model.
put('/', controller.update)	Public	URL Routing for receiving requests and sending responses on updating data to model.
delete('/:id', controller.delete)	Public	URL Routing for receiving requests and sending responses on deleting data to model.
Attribute Name	Visibility (private, public)	Type

3.3 Overall Class Diagram



Picture 54 Overall Class Diagram

3.4 Algorithm / Query

a. Algorithm Class for User (Candidates, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff)

Class Name : User (Candidates, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff)
Operation Name : Login, Create Account, Update Account, Delete Account
Algorithm : (Algo-001)

Login

```
input(username)
input(password)
userData = executeQuery('SELECT * FROM user WHERE username = ?', username).fetch()
if(validatePassword(password, userData.password))
    storeSession('accountType') = userData('accountType')
    storeSession('username') = userData('username')
    storeSession.tokenize()
    sendResponse('Login successful!')
else
    sendResponse('Incorrect username or password!')
endif
```

Create Account

```
data = getFormValue()
data.password = encryptPassword(data.password)
data.accountCreated = currentDate()
result = executeQuery('INSERT INTO user (username, password, accountType, isAdmin,
name, email, phoneNumber, address, dateOfBirth, placeOfBirth, profilePicture,
accountCreated) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)', data)
if(result.success())
    sendResponse('Account successfully created!')
else
    sendResponse('Failed to create account')
endif
```

Update Account

```
parameter(username)
data = getFormValue()
data.password = encryptPassword(data.password)
data.username = username
result = executeQuery('UPDATE user SET password=?, name=?, email=?,
phoneNumber=?, address=?, dateOfBirth=?, placeOfBirth=?, profilePicture=? WHERE
username=?', data)
if(result.success())
    sendResponse('Account successfully updated!')
else
    sendResponse('Failed to update account')
endif
```

Delete Account

```

parameter(username)
result = executeQuery('DELETE FROM user WHERE username=?', username)
if(result.success())
    sendResponse('Account successfully deleted!')
else
    sendResponse('Failed to delete account')
endif

```

Query :

Table 92 Login Operation Query

No Query	Query	Description
Q-001	<i>SELECT * FROM user WHERE username = ?</i>	Find if there exists a username based on the query execution username parameter.
Q-002	<i>INSERT INTO user (username, password, accountType, isAdmin, name, email, phoneNumber, address, dateOfBirth, placeOfBirth, profilePicture, accountCreated) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)</i>	Insert new user based on defined form data.
Q-003	<i>UPDATE user SET password=?, name=?, email=?, phoneNumber=?, address=?, dateOfBirth=?, placeOfBirth=?, profilePicture=? WHERE username=?</i>	Update user based on username parameter and fetched form data to be updated.
Q-004	<i>DELETE FROM user WHERE username=?</i>	Delete user based on query execution parameter of username.

b. Class Algorithm Salary, Schedule

Class Name : User(PracticumAssistant, LabAssistant, Staff, Admin)
 Operation Name: Input Schedule, View Schedule, Input Salary, View Salary
 Algorithm : (Algo-02)

Input Schedule

```
parameter(assistantCode)
data = getFormValue()
data.scheduleID = generateID('schedule')
data.isPresent = false
data.isReplaced = false
result = executeQuery('INSERT INTO schedule(scheduleID, room, class, day, scheduleDate,
beginTime, endTime, isPresent, isReplaced, assistantCode, lecturerID, courseID) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)', data)
if(result.success())
    sendResponse('Schedule successfully created!')
else
    sendResponse('Failed to create schedule')
endif
```

Input Salary

```
parameter(assistantCode)
data = getFormValue()
data.salaryID = generateID('salary')
result = executeQuery('INSERT INTO salary(salaryID, assistantCode, staffID, totalSalary,
availableSalary, takenSalary, availableDate, takenDate) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)', data)
if(result.success())
    sendResponse('Schedule successfully created!')
else
    sendResponse('Failed to create schedule')
endif
```

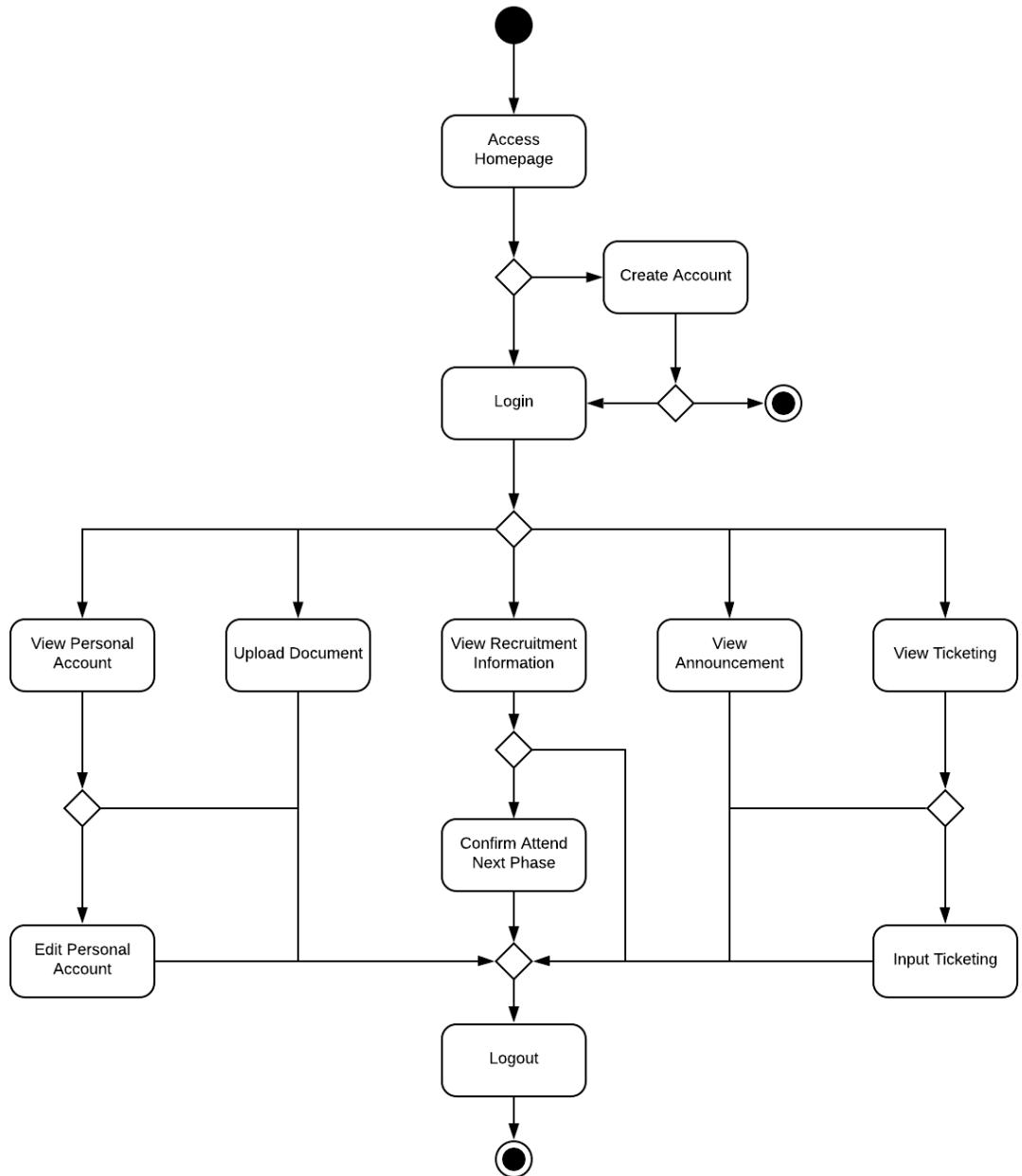
Query :

Table 93 Salary Operation Query

No Query	Query	Description
Q-005	<i>INSERT INTO schedule(scheduleID, room, class, day, scheduleDate, beginTime, endTime, isPresent, isReplaced, assistantCode, lecturerID, courseID) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)</i>	Insert schedule information of an assistant related to courseID, and lecturerID.
Q-006	<i>INSERT INTO salary(salaryID, assistantCode, staffID, totalSalary, availableSalary, takenSalary, availableDate, takenDate) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)</i>	Insert salary information of an assistant based on form data and assistantCode parameter.

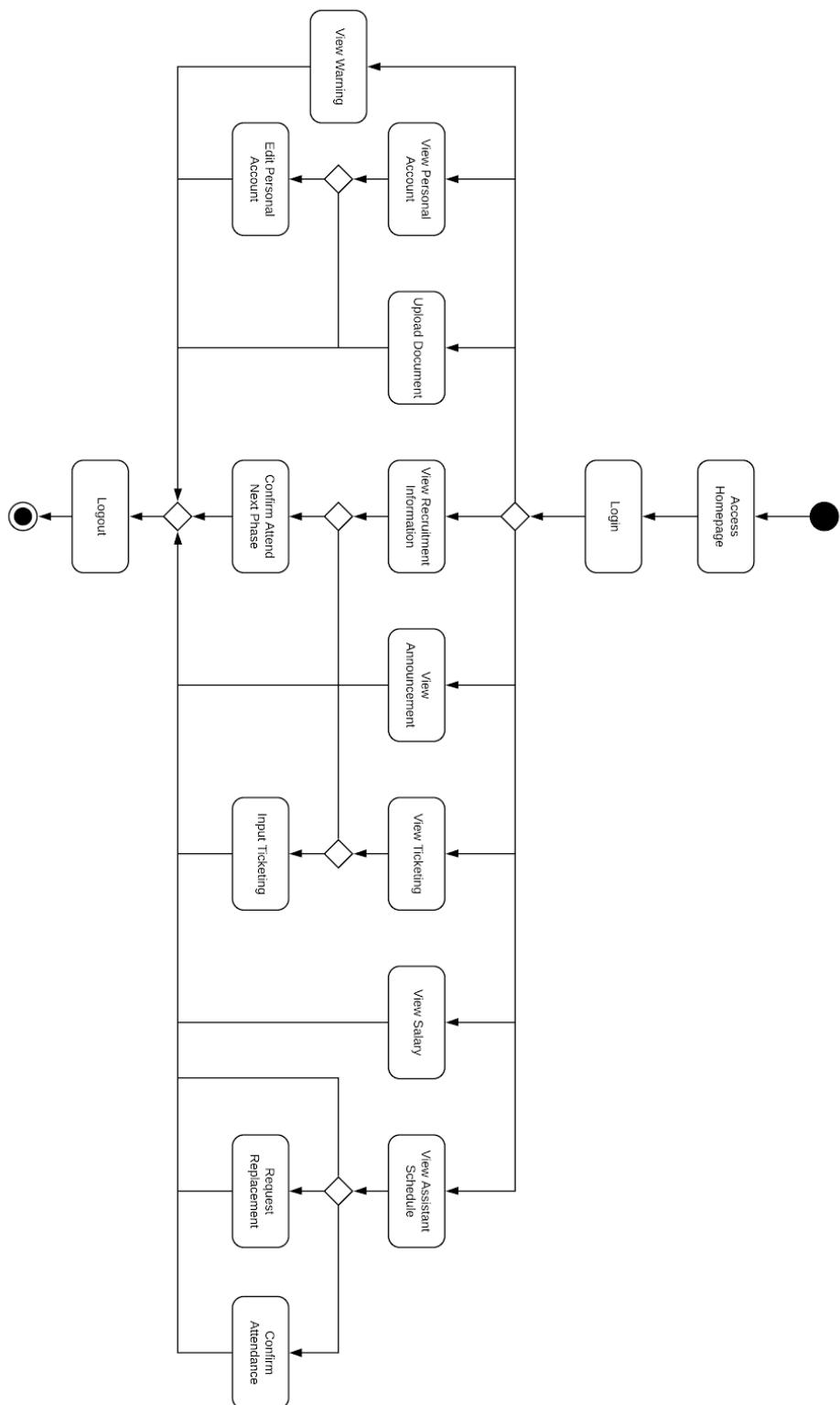
3.5 Statechart Diagram

3.5.1 Candidates



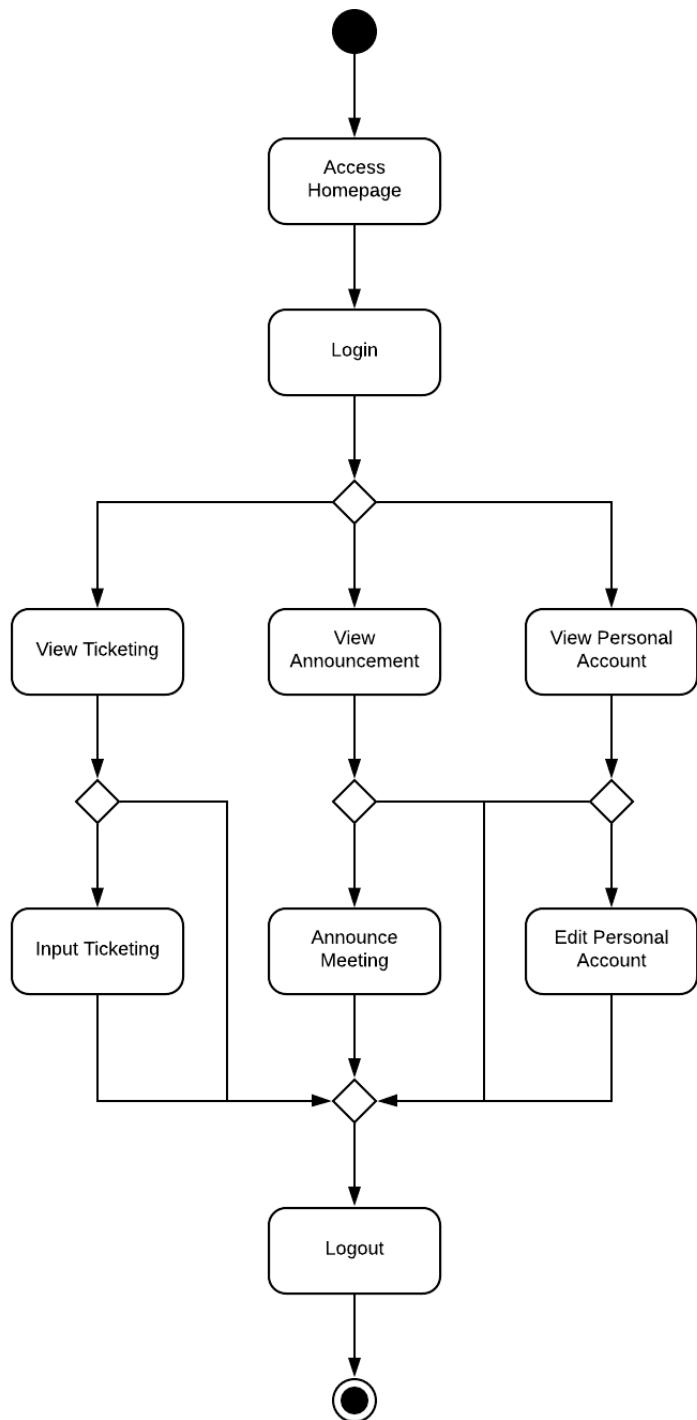
Picture 55 Statechart Diagram Candidates

3.5.2 Practicum Assistant



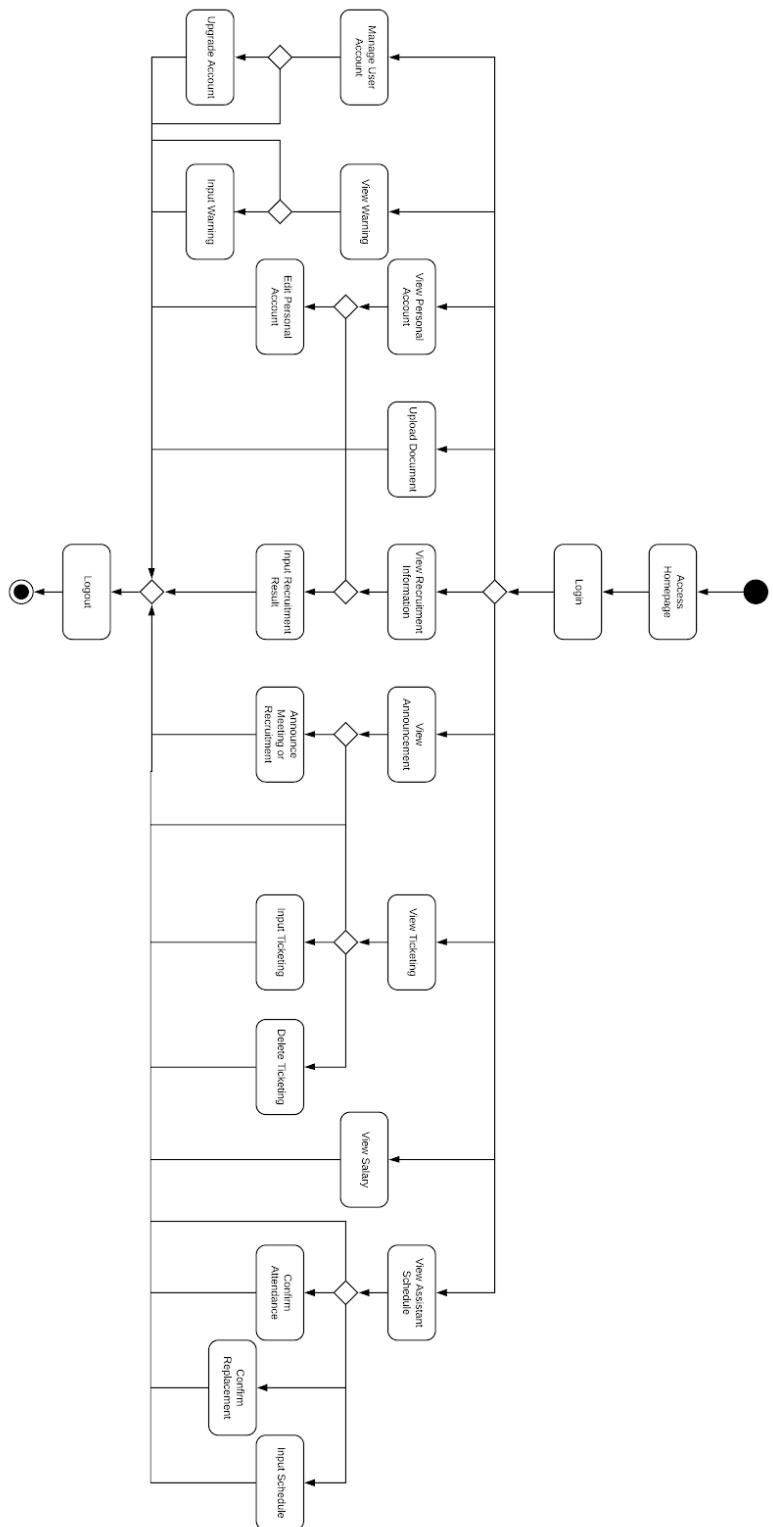
Picture 56 Statechart Diagram Practicum Assistant

3.5.3 Lecturer



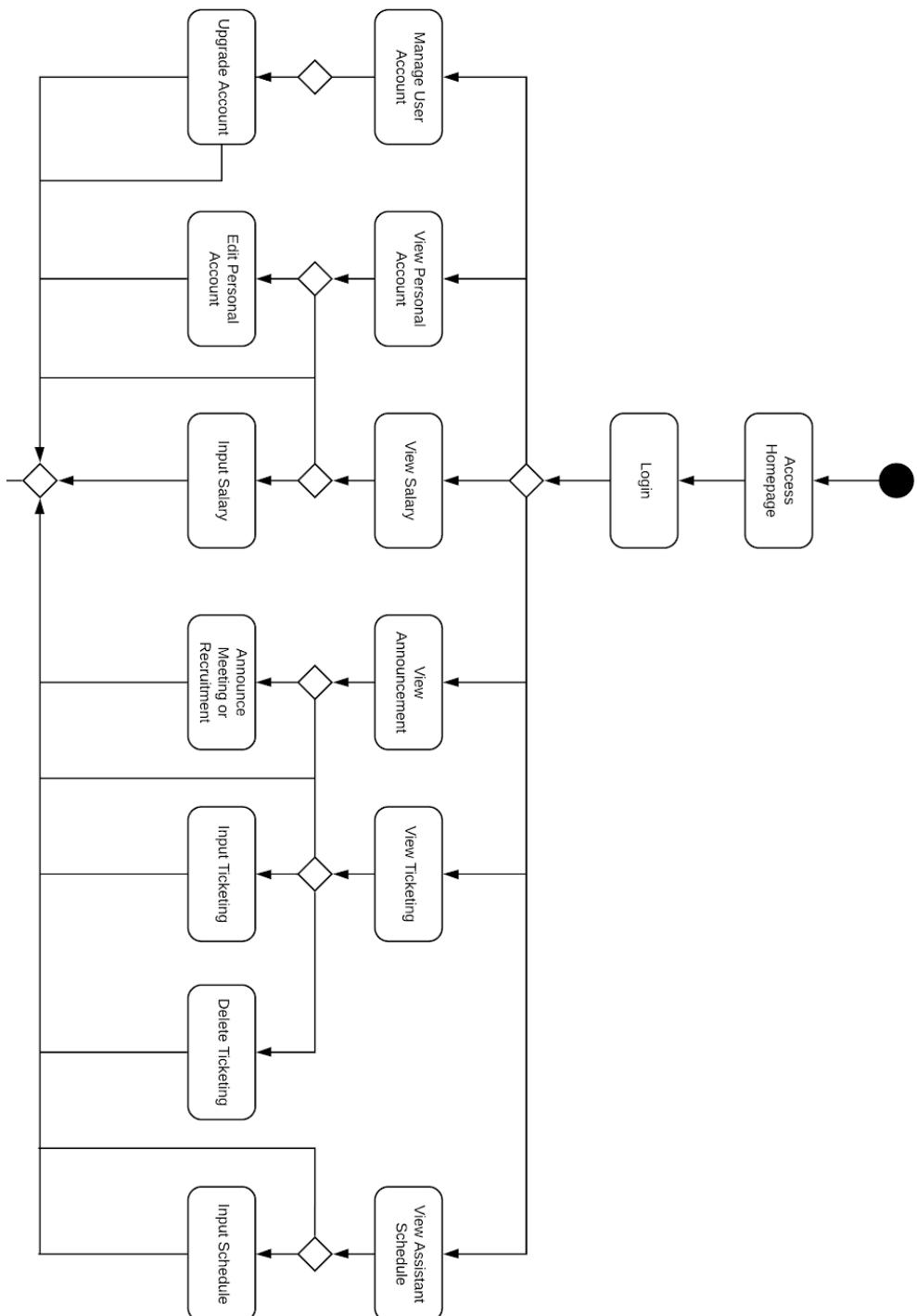
Picture 57 Statechart Diagram Lecturer

3.5.4 Lab Assistant



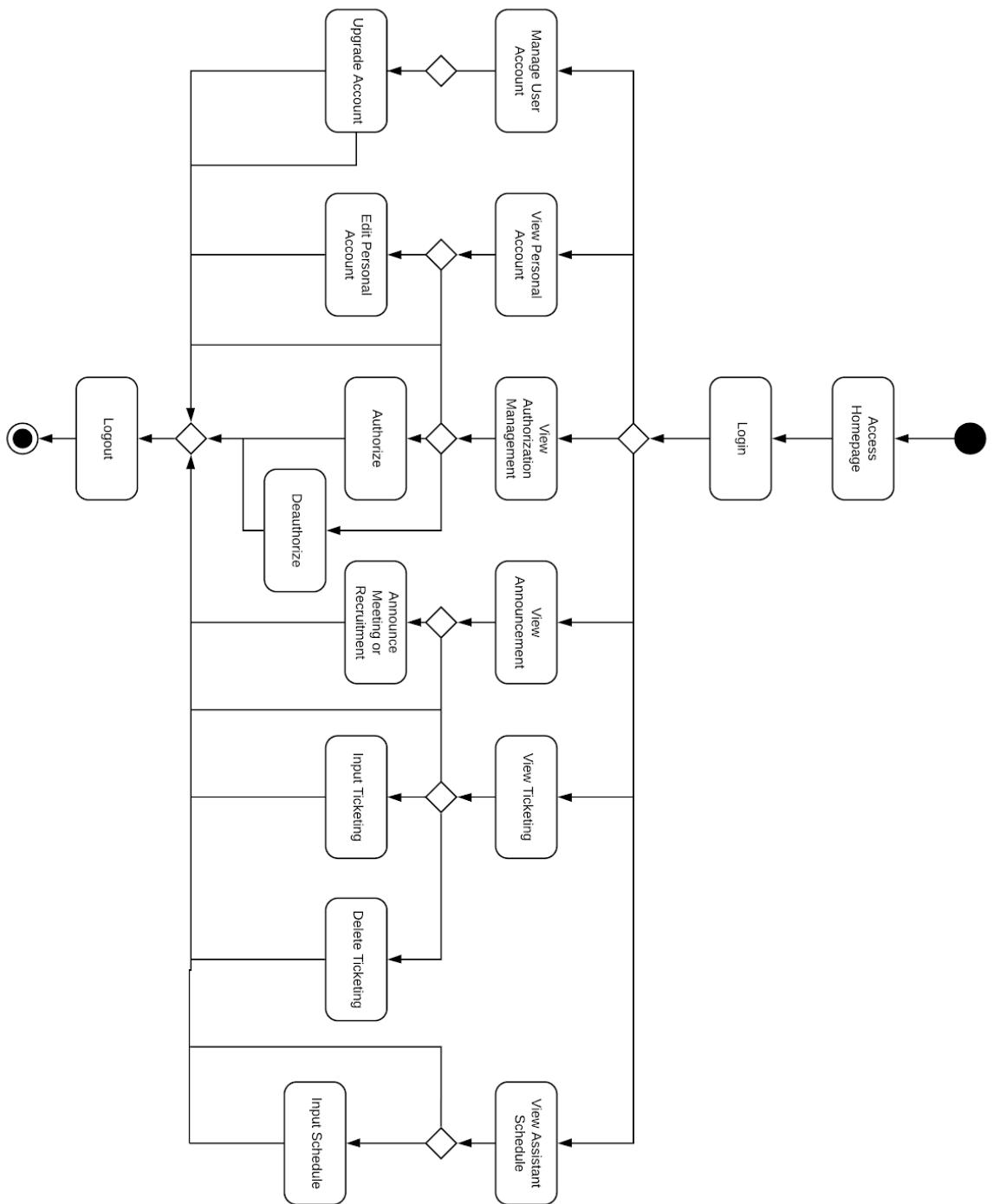
Picture 58 Statechart Diagram Lab Assistant

3.5.5 Lab Staff



Picture 59 Statechart Diagram Lab Staff

3.5.6 Lab Manager



Picture 60 Statechart Diagram Lab Manager

3.6 Interface Design

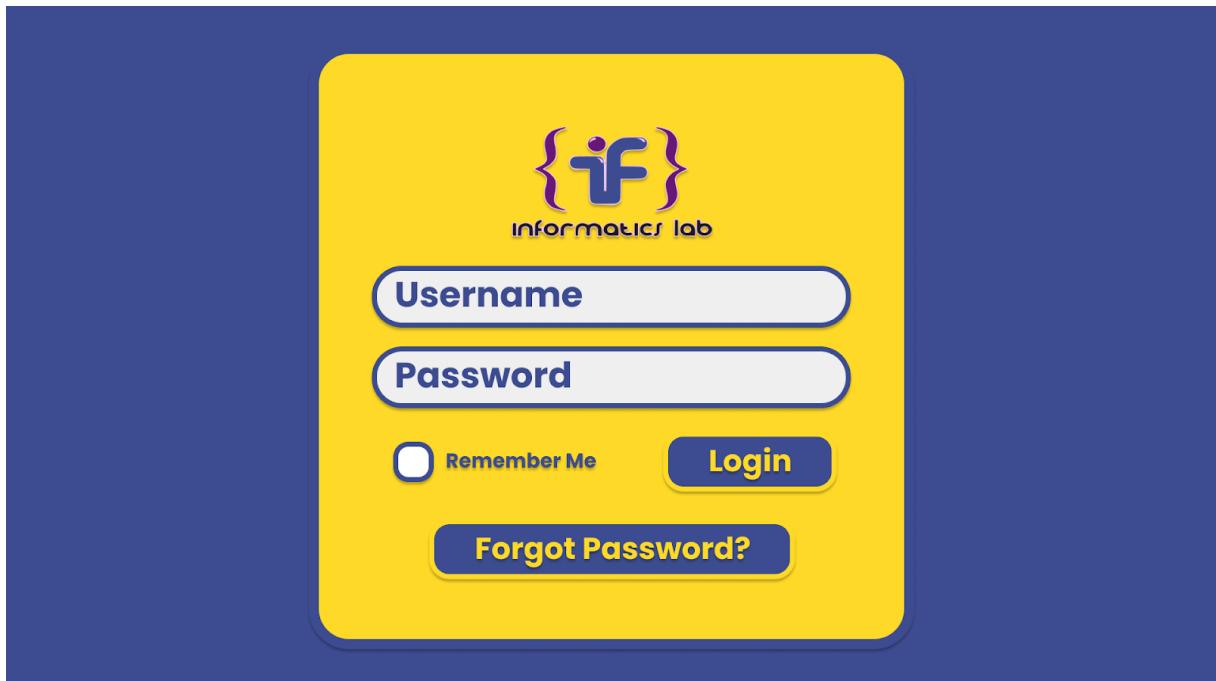
Interface : Registration

Picture 61 Registration Interface

Table 94 Registration Interface Description

ID_Object	Type	Name	Description
TXI1	Text Input	Username	Fill the text box with username.
BTN1	Button	Available	If clicked, it will show the username availability.
TXI2	Password Input	Password	Fill the text box with a password.
BTN2	Button	Show Password	If clicked, it will change the type of inputted password into text.
TXI3	Text Input	Name	Fill the text box with the full name.
TXI4	Text Input	E-Mail Address	Fill the textbox with an email address.
TXI5	Text Input	Phone Number	Fill text box with phone number
TXI6	Text Input	Address	Fill text box with address
TXI7	Text Input	Place of Birth	Fill text box with Place of Birth
DATE1	Date	Birth Date	Choose Birth Date
TXI8	Text Input	Student ID	Fill text box with Student ID
CHK1	Checkbox	Agreement	Box will be checked if clicked
BTN3	Button	Reset	If clicked, it will reset all text input
BTN4	Button	Register	If clicked, it will submit the inputted data and redirect to the next page.

Interface : Login

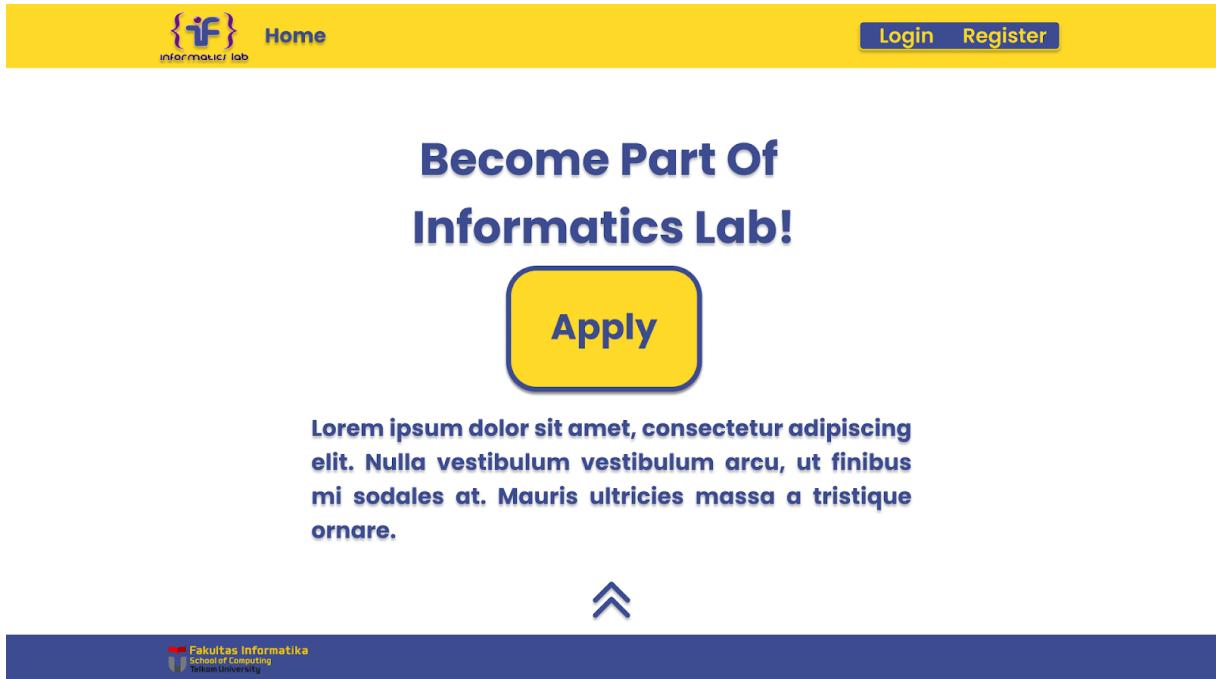


Picture 62 Login Interface

Table 95 Login Interface Description

ID_Object	Type	Name	Description
TXI1	Text Input	Username	fill the text box with username
TXI2	Password Input	Password	fill the box with password
CHK1	Checkbox	Remember Me	to keep the username and password in system
BTN1	Button	Login	to verified the username and password so the user can login with the inputted password
BTN2	Button	Forgot Password	to make the user proceed to forgot password page

Interface : Homepage 4



Picture 63 Homepage 4 Interface

Table 96 Homepage 4 Interface Description

ID_OBJECT	Type	Name	Description
BTN1	Button	Apply	to direct the user to registration page
BTN2	Button	Login	to direct user to the login page
BTN3	Button	Register	to direct user to the register page

Interface : Homepage 3



Assistant Benefits?

Placeholder

Salary!

TAK Credits!

Relations!

Many Hardskills and Softskills!

And Many More!

Picture 64 Homepage 3 Interface

Table 97 Homepage 3 Interface Description

ID_OBJECT	Type	Name	Description
IMG1	Placeholder	Placeholder1	to visualize what is the purpose that defined in the web
BTN1	Button	Login	to direct the user to login page
BTN2	Button	Register	to direct the user to the register page

Interface : Announcement

The screenshot shows a web-based application interface. At the top, there is a yellow header bar with the logo of 'Informatics lab' and the word 'Home'. To the right of the logo, there are links for 'Welcome, User', 'Dashboard', and 'Logout'. Below the header, on the left, is a sidebar with a dark blue background containing several navigation links: 'Announcement', 'Document', 'Profile', 'Schedule', 'Salary', 'Recruitment', and 'Ticketing'. A yellow box highlights the 'Current Date and Time' link, which shows 'Thu, 16/04/2020 - 11.24'. The main content area has a white background and displays three announcement cards. Each card has a dark blue header with the word 'Title' and a date 'Wed, 15 April 2020' followed by 'From Uploader'. The body of each card contains placeholder text: 'Lorem ipsum dolor sit amet consectetur adipisicing elit. Unde aliquid maiores nulla, blanditiis illum dolore amet esse nam quidem fugit quia iure praesentium libero tempora excepturi quaerat commodi saepe! Corporis. Doloremque aspernatur'. Below each card, there is a small text '0 Comments'. At the bottom of the page, there is a dark blue footer bar with the text 'Fakultas Informatika School of Computing Telkom University'.

Picture 65 Announcement Interface

Table 98 Announcement Interface Description

ID_Object	Type	Name	Description
BTNM01	Button Menu	Announcement	to direct the user to Announcement Page
BTNM02	Button Menu	Document	to direct the user to document page
BTNM03	Button Menu	Profile	to direct the user to profile page
BTNM04	Button Menu	Schedule	to direct the user to schedule page
BTNM05	Button Menu	Salary	to direct the user to salary page
BTNM06	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM07	Button Menu	Ticketing	to direct the user to ticketing page
BTN01	Button	Dashboard	to direct user to dashboard page

BTN02	Button	Logout	to direct user to logout page
TXD1	Text Dropdown	Filter	If clicked, it will show a dropdown with option to filter the announcement
TXD2	Text Dropdown	Sort	If clicked, it will show a dropdown with option to sort the announcement
SRC1	Search Combo	Search	Type desired keyword and click on arrow to search announcement that depends on the inputted keyword
TXF1	Text Field	Announcement Text	Shows announcement info
TXF2	Text Field	Date and Time	Shows current Date and Time

Interface : View Schedule Form

Fakultas Informatika
School of Computing
Telkom University

Picture 66 View Schedule Form Interface

Table 99 View Schedule Form Interface Description

ID_Object	Type	Name	Description
BTNM01	Button Menu	Announcement	to direct the user to Announcement Page

BTNM02	Button Menu	Document	to direct the user to document page
BTNM03	Button Menu	Profile	to direct the user to profile page
BTNM04	Button Menu	Schedule	to direct the user to view schedule form
BTNM05	Button Menu	Salary	to direct the user to salary page
BTNM06	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM07	Button Menu	Ticketing	to direct the user to ticketing page
TBTN01	Tab Button	Schedule Table	to direct user to View Schedule Table
TBTN02	Tab Button	Request Replacement Form	shows request replacement page (readonly)
BTN01	Button	Dashboard	to direct user to dashboard page
BTN02	Button	Logout	to direct user to logout page
TXF1	Text Field	Date and Time	Shows current Date and Time
TXI1	Text Input	Class	to show the class that set by system(read only)
TXI2	Text Input	Time	to show the time that set by system (read only)
TXI3	Text Input	Room	to show the room that set by system (read only)
TXD1	Text Dropdown	Practicum Date	to set the practicum date and the course subject practicum type
TXI4	Text Input	Arrived Time	to set the arrived time of the practicum assistant
TXI5	Text Input	Finished Time	to set the finish time of the practicum assistant
BTN03	Button	Submit	to submit the inputted data into database and show it in schedule table, and

			redirect to view schedule table
BTN04	Button	Reset	to clear all the text input

Interface : View Schedule Table

Picture 67 View Schedule Table Interface

Table 100 View Schedule Table Interface Description

BTNM08	Button Menu	Announcement	to direct the user to Announcement Page
BTNM09	Button Menu	Document	to direct the user to document page
BTNM10	Button Menu	Profile	to direct the user to profile page
BTNM11	Button Menu	Schedule	to direct the user to view schedule form
BTNM12	Button Menu	Salary	to direct the user to salary page
BTNM13	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM14	Button	Ticketing	to direct the user to ticketing page

	Menu		
TBTN03	Tab Button	Schedule Table	to direct user to View Schedule Table
TBTN04	Tab Button	Request Replacement Form	shows request replacement page (readonly)
BTN03	Button	Dashboard	to direct user to dashboard page
BTN04	Button	Logout	to direct user to logout page
TXF2	Text Field	Date and Time	Shows current Date and Time
TXD2	Text Dropdown	Filter	If clicked, it will show a dropdown with option to filter the data
TXD3	Text Dropdown	Sort	If clicked, it will show a dropdown with option to sort the data
SRC1	Search Combo	Search	Type desired keyword and click on arrow to search for specific data that depends on the inputted keyword
TBL01	Table	Schedule Table	Shows Schedule data that is saved in the system
ACB1	Action Button	Request Replacement	To direct user to request replacement page if user wants to change schedule
PPB01	Previous Page Button	Previous Page	To direct user to the page before current page
NPB01	Next Page Button	Next Page	To direct user to the page after current page
PNB01	Page Number Button	Page Number	To direct user to the page that is desired to access

Interface : Request Replacement

Picture 68 Request Replacement Interface

Table 101 Request Replacement Interface Description

ID_Object	Type	Name	Description
BTNM15	Button Menu	Announcement	to direct the user to Announcement Page
BTNM16	Button Menu	Document	to direct the user to document page
BTNM17	Button Menu	Profile	to direct the user to profile page
BTNM18	Button Menu	Schedule	to direct the user to view schedule form
BTNM19	Button Menu	Salary	to direct the user to salary page
BTNM20	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM21	Button Menu	Ticketing	to direct the user to ticketing page
TBTN05	Tab Button	Schedule Table	to direct user to View Schedule Table
TBTN06	Tab Button	Request	to direct user to Request Replacement

		Replacement Form Menu	form(readonly)
BTN05	Button	Dashboard	to direct user to dashboard page
BTN06	Button	Logout	to direct user to logout page
TXF3	Text Field	Date and Time	Shows current Date and Time
TXI6	Text Input	Class	to show the class that set by system(read only)
TXI7	Text Input	Time	to show the time that set by system (read only)
TXD4	Text Dropdown	Practicum Date	to set the practicum date and the course subject practicum type
TXI8	Text Input	Description	Fill text box with description
BTN07	Button	Submit	to submit the inputted data into database
BTN08	Button	Reset	to clear all the text input

Interface : Authorization

The screenshot shows a web-based application interface for managing authorization requests. At the top, there's a yellow header bar with the logo 'iF' for Informatika lab, a 'Home' link, and navigation links for 'Welcome, Manager', 'Dashboard', and 'Logout'. Below the header, on the left, is a sidebar menu with options: 'Announcement', 'Admin', 'Authorization' (which is currently selected and highlighted in blue), 'Profile', and 'Ticketing'. A yellow box displays the 'Current Date and Time' as 'Thu, 16/04/2020 - 11.24'. The main content area has a dark blue header with tabs for 'Authorization' and 'Deauthorization'. Below this is a search/filter bar with dropdowns for 'Filter by' (set to 'Any') and 'Sort by' (set to 'Latest'), and a 'Search' input field. The main data area is a table with columns: 'No.', 'Action', 'Status', 'Auth. Date', and 'Action' (containing checkboxes). The table contains four rows of data:

No.	Action	Status	Auth. Date	Action
1	Test Phase Recruitment	Wait Appr..	N/A	<input checked="" type="checkbox"/> <input type="checkbox"/>
2	Document Phase Recruitment Res..	Done	Wed, 15/04/2020	<input checked="" type="checkbox"/> Approved
3	Warning for assistant SLR violation..	Done	Tue, 14/04/2020	<input type="checkbox"/> Rejected
4	Allow admin for lab assistant inter..	Done	Tue, 14/04/2020	<input checked="" type="checkbox"/> Approved

At the bottom right of the table, there are navigation icons for a grid, a list, and two arrows. The footer of the page includes the university logo and the text 'Fakultas Informatika Universitas Telkom'.

Picture 69 Authorization Interface

Table 102 Authorization Interface Description

ID_Object	Type	Name	Description
BTNM01	Button Menu	Announcement	to direct the user to Announcement Page
BTNM02	Button Menu	Admin	to direct the user to admin page
BTNM03	Button Menu	Authorization	to direct user to authorization page
BTNM04	Button Menu	Profile	to direct the user to profile page
BTNM05	Button Menu	Ticketing	to direct the user to ticketing page
BTN01	Button	Dashboard	to direct user to dashboard page
BTN02	Button	Logout	to direct user to logout page
TXD1	Text Dropdown	Filter	If clicked, it will show a dropdown with option to filter the data
TXD2	Text Dropdown	Sort	If clicked, it will show a dropdown with option to sort the data
SRC1	Search Combo	Search	Type desired keyword and click on arrow to search for specific data that depends on the inputted keyword
TBTN01	Tab Button	Authorization	to direct user to authorization menu
TBTN02	Tab Button	Deauthorization	to direct user to deauthorization menu
TBL01	Table	Authorization Data	Shows Authorization data that is saved in the system
ACB01	Action Button	Approve	Gives an action to approve the data
ACB02	Action Button	Reject	Gives an action to reject the data
PPB01	Previous Page Button	Previous Page	To direct user to the page before current page
NPB01	Next Page Button	Next Page	To direct user to the page after current page

PNB01	Page Number Button	Page Number	To direct user to the page that is desired to access
-------	--------------------	-------------	--

Interface : Deauthorization

The screenshot shows a web-based application interface for managing user deauthorization. At the top, there is a yellow header bar with the logo 'if' (Informatics Lab) and links for 'Home', 'Welcome, Manager', 'Dashboard', and 'Logout'. Below the header, on the left, is a sidebar menu with options: 'Announcement', 'Admin', 'Authorization', 'Profile', and 'Ticketing'. A yellow box highlights the 'Current Date and Time' as 'Thu, 16/04/2020 - 11.24'. On the right, there is a main content area titled 'Authorization' with a sub-tab 'Deauthorization' selected. Below this are filtering and sorting options ('Filter by Any', 'Sort by Latest', 'Search'). A table lists four users:

No.	Student ID	Name	Code	Status	Reason	Action
1	1301180001	User	USR	Active	N/A	<input type="radio"/> Deauth
2	1301180002	User X	UEX	Active	N/A	<input type="radio"/> Deauth
3	1301180003	User Y	SEY	Active	N/A	<input type="radio"/> Deauth
4	1301180004	User Z	ERZ	Deauthorized	Violation..	<input checked="" type="radio"/> Deauth

At the bottom of the table are navigation icons: a left arrow, a bolded number 1, a right arrow, and a double-right arrow.



Picture 70 Deauthorization Interface

Table 103 Deauthorization Interface Description

ID_Object	Type	Name	Description
BTNM08	Button Menu	Announcement	to direct the user to Announcement Page
BTNM09	Button Menu	Admin	to direct the user to admin page
BTNM10	Button Menu	Authorization	to direct user to authorization page
BTNM11	Button Menu	Profile	to direct the user to profile page
BTNM12	Button Menu	Ticketing	to direct the user to ticketing page
BTN03	Button	Dashboard	to direct user to dashboard page
BTN04	Button	Logout	to direct user to logout page

TXD3	Text Dropdown	Filter	If clicked, it will show a dropdown with option to filter the data
TXD4	Text Dropdown	Sort	If clicked, it will show a dropdown with option to sort the data
SRC2	Search Combo	Search	Type desired keyword and click on arrow to search for specific data that depends on the inputted keyword
TBTN03	Tab Button	Authorization	to direct user to authorization menu
TBTN04	Tab Button	Deauthorization	to direct user to deauthorization menu
TBL02	Table	Authorization Data	Shows Authorization data that is saved in the system
ACB03	Action Button	Approve	Gives an action to approve the data
ACB04	Action Button	Reject	Gives an action to reject the data
PPB02	Previous Page Button	Previous Page	To direct user to the page before current page
NPB02	Next Page Button	Next Page	To direct user to the page after current page
PNB02	Page Number Button	Page Number	To direct user to the page that is desired to access

Interface : Input Ticketing

Picture 71 Input Ticketing Interface

Table 104 Input Ticketing Interface Description

ID_Object	Type	Name	Description
TBTN01	Tab Button	Input Ticket	to direct the user to input ticket form
TBTN02	Tab Button	View Active Ticket	to direct user to active ticket tab(readonly)
BTNM01	Button Menu	Announcement	to direct the user to Announcement Page
BTNM02	Button Menu	Document	to direct the user to document page
BTNM03	Button Menu	Profile	to direct the user to profile page
BTNM04	Button Menu	Schedule	to direct the user to schedule page
BTNM05	Button Menu	Salary	to direct the user to salary page
BTNM06	Button Menu	Recruitment	to direct the user to Recruitment page

BTNM07	Button Menu	Ticketing	to direct the user to ticketing page
BTN01	Button	Dashboard	to direct user to dashboard page
BTN02	Button	Logout	to direct user to logout page
TXF1	Text Field	Date and Time	Shows current Date and Time
BTN03	Button	Submit	to submit the inputted data into database and show it in view active ticket table, and redirect into view ticketing table
BTN04	Button	Reset	to clear all the text input
TXI01	Text Input	Title	to input the title to the ticketing database or something
TXD01	Text Dropdown	Type	to make the user select the type of the ticketing

Interface : View Ticketing Table

Picture 72 View Ticketing Table Interface

Table 105 View Ticketing Table Interface Description

ID_OBJECT	Type	Name	Description
BTNM8	Button Menu	Announcement	to direct the user to Announcement Page
BTNM9	Button Menu	Document	to direct the user to document page
BTNM10	Button Menu	Profile	to direct the user to profile page
BTNM11	Button Menu	Schedule	to direct the user to schedule page
BTNM12	Button Menu	Salary	to direct the user to salary page
BTNM13	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM14	Button Menu	Ticketing	to direct the user to ticketing page
BTN5	Button	Dashboard	to direct user to dashboard page
BTN6	Button	Logout	to direct user to logout page
TXF2	Text Field	Date and Time	Shows current Date and Time
TXD2	Text Dropdown	Filter	If clicked, it will show a dropdown with option to filter the data
TXD3	Text Dropdown	Sort	If clicked, it will show a dropdown with option to sort the data
SRC1	Search Combo	Search	Type desired keyword and click on arrow to search for specific data that depends on the inputted keyword
TBL01	Table	Authorization Data	Shows Authorization data that is saved in the system
ACB1	Action Button	View Ticket Form	Gives an action to direct into View Ticket Form depends on the chosen ticket
TBTN03	Tab Button	Input Ticket	to direct the user to input ticket form
TBTN04	Tab Button	View Active Ticket	to direct user to active ticket tab(readonly)
PPB01	Previous Page Button	Previous Page	To direct user to the page before current page
NPB01	Next Page	Next Page	To direct user to the page after current

	Button		page
PNB01	Page Number Button	Page Number	To direct user to the page that is desired to access

Interface : View Ticketing Form

Picture 73 View Ticketing Form Interface

Table 106 View Ticketing Form Interface Description

ID_OBJECT	Type	Name	Description
BTNM15	Button Menu	Announcement	to direct the user to Announcement Page
BTNM16	Button Menu	Document	to direct the user to document page
BTNM17	Button Menu	Profile	to direct the user to profile page
BTNM18	Button Menu	Schedule	to direct the user to schedule page
BTNM19	Button Menu	Salary	to direct the user to salary page
BTNM20	Button Menu	Recruitment	to direct the user to

			Recruitment page
BTNM21	Button Menu	Ticketing	to direct the user to ticketing page
BTN7	Button	Dashboard	to direct user to dashboard page
BTN8	Button	Logout	to direct user to logout page
TXF3	Text Field	Date and Time	Shows current Date and Time
BTN9	Button	Reply	to give option to the user make reply to the message
TXF4	Text Field	Title	to show the imputed title from input ticket page (read only)
TXF5	Text Field	Type	to show the input type from input ticket page (read only)
TXF6	Text Field	Caption	to show the imputed caption from input ticket page (read only)
TXF7	Text Field	Replies	to show the imputed replies from input ticket page (read only)
TXF8	Text Field	Date and Time	Shows current Date and Time
TBTN05	Tab Button	Input Ticket	to direct the user to input ticket form
TBTN06	Tab Button	View Active Ticket	to direct user to active ticket tab

Interface : Salary

The screenshot shows a web-based application interface for managing salaries. At the top, there is a yellow header bar with the logo of the Faculty of Informatics (iF) and a 'Home' button. To the right of the header are links for 'Welcome, User', 'Dashboard', and 'Logout'. Below the header, on the left, is a sidebar menu with links for 'Announcement', 'Document', 'Profile', 'Schedule', 'Salary', 'Recruitment', and 'Ticketing'. A yellow box highlights the 'Current Date and Time' which shows 'Thu, 16/04/2020 - 11.24'. The main content area features a table titled 'Salary' with columns for 'No.', 'Date', 'Available Salary', 'Taken Salary', and 'Total Salary'. Two rows of data are visible: one for Monday, 09/03/2020, and another for Wednesday, 15/04/2020. At the bottom of the table, summary boxes show 'Total Available Salary' (Rp. 250.000,00,-) and 'Total Taken Salary' (Rp. 50.000,00,-). Navigation buttons for page 1 of 2 are also present.

Picture 74 Salary Interface

Table 107 Salary Interface Description

ID_OBJE CT	Type	Name	Description
BTNM01	Button Menu	Announcement	to direct the user to Announcement Page
BTNM02	Button Menu	Document	to direct the user to document page
BTNM03	Button Menu	Profile	to direct the user to profile page
BTNM04	Button Menu	Schedule	to direct the user to schedule page
BTNM05	Button Menu	Salary	to direct the user to salary page
BTNM06	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM07	Button Menu	Ticketing	to direct the user to ticketing page

BTN01	Button	Dashboard	to direct user to dashboard page
BTN02	Button	Logout	to direct user to logout page
TBL01	Table	Salary Table	to show date, taken, available, and total salary for
TXD01	Text Dropdown	Filter by	to make the user select what based are they looking for
TXD02	Text Dropdown	Sort by	to make the user select what sort-based are they looking
SRC01	Search Combo	Search	to make the user search what are they looking for at the page

Interface : Upload Document

Picture 75 Upload Document Interface

Table 108 Upload Document Interface Description

ID_OBJECT	Type	Name	Description
BTNM01	Button Menu	Announcement	to direct the user to Announcement Page
BTNM02	Button Menu	Document	to direct the user to document page

BTNM03	Button Menu	Profile	to direct the user to profile page
BTNM04	Button Menu	Schedule	to direct the user to schedule page
BTNM05	Button Menu	Salary	to direct the user to salary page
BTNM06	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM07	Button Menu	Ticketing	to direct the user to ticketing page
BTN01	Button	Dashboard	to direct user to dashboard page
BTN02	Button	Logout	to direct user to logout page
TBL01	Table	List Uploaded	to show, edit, and delete the data that inputted to the table
TXD02	Text Dropdown	Filter By	to show the user the list for the document that they want to upload
TXD03	Text Dropdown	Sort By	to show the user the list for the document that they want to upload
FINPUT01	Input File	Upload File	to make user upload the file that they wanted to

Interface : Profile

The screenshot shows a web-based application interface for managing a user profile. At the top, there is a yellow header bar with the 'iF' logo and 'Home' link on the left, and 'Welcome, User' with 'Dashboard' and 'Logout' links on the right. Below the header, on the left, is a vertical navigation menu with links: 'Announcement', 'Document', 'Profile' (which is highlighted in blue), 'Schedule', 'Salary', 'Recruitment', and 'Ticketing'. A yellow box highlights the 'Profile' link. To its right is a circular profile icon with a pencil and paper symbol. Next to the icon is the text 'User - My Name' and 'Practicum Assistant ~ USR ~ 1301180000' with the date 'Bojongsong, 01/01/1999'. Below this is a yellow 'Edit Profile' button. To the right of the profile icon is a section titled 'Account Management' containing fields for 'Old Password' (with a masked input), 'New Password' (with a masked input and a 'Show' button), 'Address' (input field), 'E-Mail Address' (input field), 'Phone Number' (input field), and buttons for 'Reset' and 'Update'. At the bottom of the page is a dark blue footer bar with the 'Fakultas Informatika' logo.

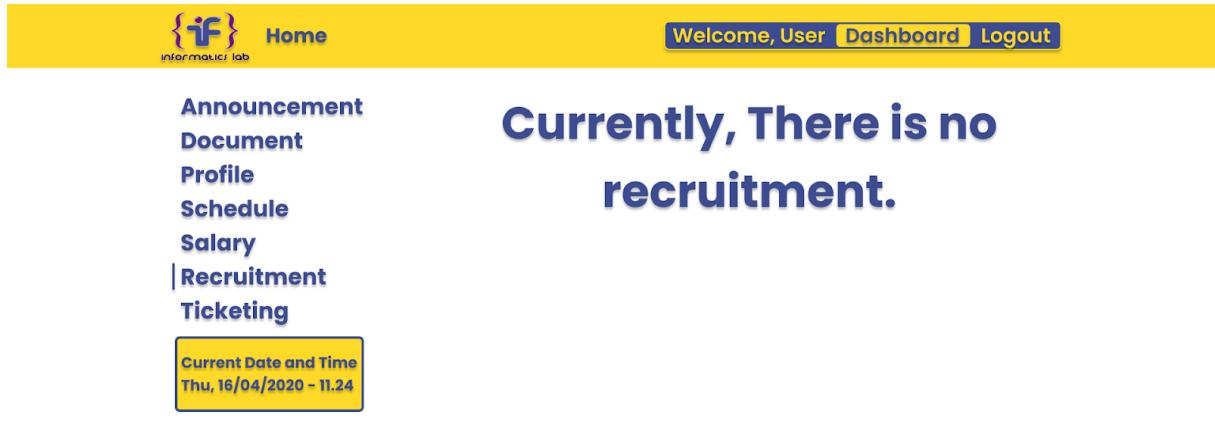
Picture 76 Profile Interface

Table 109 Profile Interface Description

ID_OBJECT	Type	Name	Description
BTNM01	Button Menu	Announcement	to direct the user to Announcement Page
BTNM02	Button Menu	Document	to direct the user to document page
BTNM03	Button Menu	Profile	to direct the user to profile page
BTNM04	Button Menu	Schedule	to direct the user to schedule page
BTNM05	Button Menu	Salary	to direct the user to salary page
BTNM06	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM07	Button Menu	Ticketing	to direct the user to ticketing page

BTN01	Button	Dashboard	to direct user to dashboard page
BTN02	Button	Logout	to direct user to logout page
BTN03	Button	Reset	to set all text input becomes null
BTN04	Button	Update	to update the data based on the user input in text input
BTN05	Button	Edit Profile	to make the user edit their profile on the text
TXI1	Text Input	Old Password	to make user input their current password
TXI2	Text Input	New Password	to make user input their new password
TXI3	Text Input	Address	to make user input their address
TXI4	Text Input	Email Address	to make user input their email address
TXI5	Text Input	Phone Number	to make user input their phone number
FINPUT01	File Input	Profile Upload	to make the user change the profile photos based what their inputted

Interface : Recruitment Not Exist



Picture 77 Recruitment Not Exist Interface

Table 110 Recruitment Not Exist Interface Description

ID_Object	Type	Name	Description
BTNM01	Button Menu	Announcement	to direct the user to Announcement Page
BTNM02	Button Menu	Document	to direct the user to document page
BTNM03	Button Menu	Profile	to direct the user to profile page
BTNM04	Button Menu	Schedule	to direct the user to schedule page
BTNM05	Button Menu	Salary	to direct the user to salary page
BTNM06	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM07	Button Menu	Ticketing	to direct the user to ticketing page
BTN01	Button	Dashboard	to direct user to dashboard page
BTN02	Button	Logout	to direct user to logout page

TXTB1	Text Box	Date and Time	Show current date and time
TXF1	Text Field	Announcement Text	Show announcement text

Interface : Recruitment Selection Phase

Picture 78 Recruitment Selection Phase Interface

Table 111 Recruitment Selection Phase Interface Description

ID_Object	Type	Name	Description
BTNM08	Button Menu	Announcement	to direct the user to Announcement Page
BTNM09	Button Menu	Document	to direct the user to document page
BTNM10	Button Menu	Profile	to direct the user to profile page
BTNM11	Button Menu	Schedule	to direct the user to schedule page
BTNM12	Button Menu	Salary	to direct the user to salary page
BTNM13	Button Menu	Recruitment	to direct the user to

			Recruitment page
BTNM14	Button Menu	Ticketing	to direct the user to ticketing page
BTN03	Button	Dashboard	to direct user to dashboard page
BTN04	Button	Logout	to direct user to logout page
TXTB2	Text Box	Date and Time	Show current date and time
CARD01	Cards	Card Phase	to display the selection phase based on the cards

Interface : Recruitment Exist Result

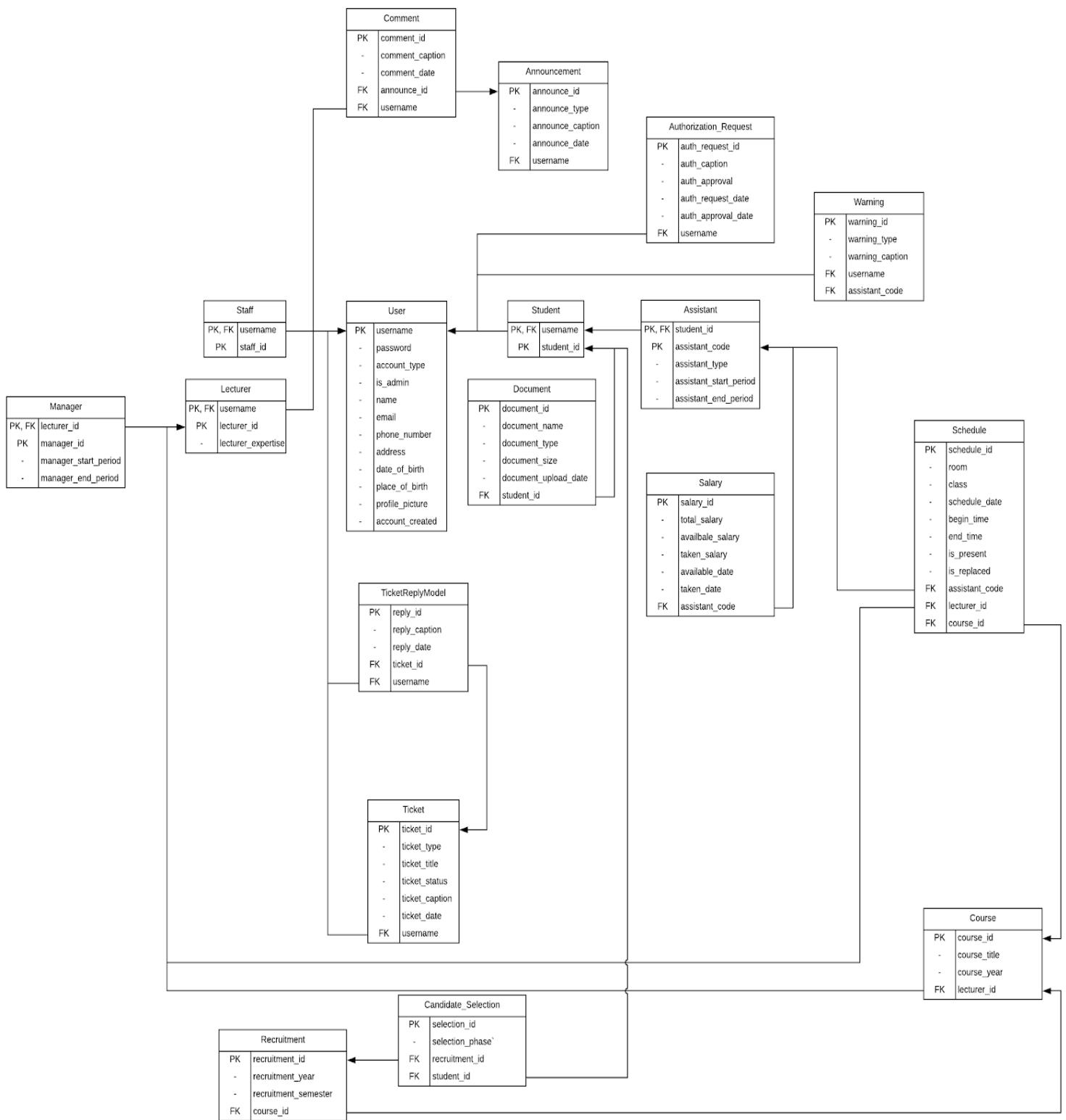
Picture 79 Recruitment Exist Result Interface

Table 112 Recruitment Exist Result Interface Description

ID_Object	Type	Name	Description
BTNM15	Button Menu	Announcement	to direct the user to Announcement Page

BTNM16	Button Menu	Document	to direct the user to document page
BTNM17	Button Menu	Profile	to direct the user to profile page
BTNM18	Button Menu	Schedule	to direct the user to schedule page
BTNM19	Button Menu	Salary	to direct the user to salary page
BTNM20	Button Menu	Recruitment	to direct the user to Recruitment page
BTNM21	Button Menu	Ticketing	to direct the user to ticketing page
BTN05	Button	Dashboard	to direct user to dashboard page
BTN06	Button	Logout	to direct user to logout page
TXTB2	Text Box	Date and Time	Show current date and time
TBL01	Table	Document Phase Result	to show the user that the result of the Phase
TXD02	Text Dropdown	Filter By	to show the user what they want to see by using content that available in filter
TXD03	Text Dropdown	Sort By	o show the user what they want to see by using sorting method that available in short by

3.7 Class Persistence Representation Design



Picture 80 Class Persistence Representation

4 Usability Matrix

Table 113 Usability Matrix

Requirement	Related Use Case	Class
F-01	Login	User (Candidates, Practicum Assistant, Lab Assistant, Manager, Staff, Lecturer), UserRouter, UserController, UserModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-02	Create account	User , Student, Candidates, UserRouter, UserController, UserModel, StudentRouter, StudentController, StudentModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-03	Authorize	User, Lecturer, Manager, AuthorizationRequestRouter, AuthorizationRequestController, AuthorizationRequestModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-04	Upload document	User , Student, DocumentRouter, DocumentController, DocumentModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-05	Input salary	Staff, User , <<interface>>Admin, SalaryRouter, SalaryController, SalaryModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-06	View assistant salary	User , Student, PracticumAssistant, SalaryRouter, SalaryController, SalaryModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-07	Input ticketing	User (Candidates, Practicum Assistant, Lab Assistant, Manager, Staff, Lecturer), TicketRouter, TicketController, TicketModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-08	Input assistant schedule	User, Student, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff,

		<<interface>>Admin, ScheduleRouter, ScheduleController, ScheduleModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-09	Upgrade account	User, Student, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff, <<interface>>Admin, AssistantRouter, AssistantController, AssistantModel, StudentModel, UserModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-10	Announce meeting or recruitment	User, Student, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff, <<interface>>Admin, AnnouncementRouter, AnnouncementController, AnnouncementModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-11	Edit personal account	User, UserRouter, UserController, UserModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-12	View assistant schedule	User, Student, PracticumAssistant, ScheduleRouter, ScheduleController, ScheduleModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-13	View recruitment info management	User, Student, RecruitmentRouter, RecruitmentController, RecruitmentModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-14	Delete ticketing	User, Student, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff, <<interface>>Admin, TicketRouter, TicketController, TicketModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-15	Input warning	User, Student, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff, <<interface>>Admin, WarningRouter, WarningController, WarningModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess

F-16	View ticketing	User, TicketRouter, TicketController, TicketModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-17	Input recruitment result	User, Student, RecruitmentRouter, RecruitmentController, RecruitmentModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-18	View warning	User, Student, PracticumAssistant, WarningRouter, WarningController, WarningModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-19	Request replacement	User, Student, PracticumAssistant, ScheduleRouter, ScheduleController, ScheduleModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-20	View personal account	User, UserRouter, UserController, UserModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-21	Manage user account	User, Student, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff, <<interface>>Admin, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-22	Deauthorize	User, Lecturer, Manager, AssistantRouter, AssistantController, AssistantModel, StudentModel, UserModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-23	Confirm replacement	User, Student, PracticumAssistant, LabAssistant, Lecturer, Manager, Staff, <<interface>>Admin, ScheduleRouter, ScheduleController, ScheduleModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-24	View announcement	User, AnnouncementRouter, AnnouncementController, AnnouncementModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-25	View and confirm attendance	User, Student, PracticumAssistant, ScheduleRouter, ScheduleController,

		ScheduleModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess
F-26	Confirm next phase attendance	User, Student, CandidateSelectionRouter, CandidateSelectionController, CandidateSelectionModel, <<interface>> APIRouter, <<interface>>APIController, <<interface>>DataAccess

Attachments

- A. Full Resolution Complete Class Diagram

<https://www.lucidchart.com/publicSegments/view/d5756656-d24f-496c-85fc-f2dd85cb3af4/image.png>