SOFTWARE DESIGN DOCUMENT

**(SDD)**

**For**

**DRIVING SCHOOL MANAGEMENT SYSTEM**

**Prepared for:**

**MUHAMAD IDAHAM BIN UMAR ONG**

**Generated By:**

**BETA TECH**

HII WEI KAI CB 14155

ZAFIRAH BINTI ZAILANI CB 13117

WILLIAM SOON SIIK YANG CB 14154

MUHAMAD HAFIZI BIN AZIZ CB 13057

AL QASTHOLANI BIN KAREBOLLAH CB 13019

**October 2015**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Approval List Table** | | | | | | |
| Index | 01 | 02 | 03 | 04 | 05 | 06 |
| Writed by:  Name: | Date: | Date: | Date: | Date: | Date: | Date: |
| Verified by:  Quality Manager  Name:  Muhamad Hafizi B. Aziz | Date: | Date: | Date: | Date: | Date: | Date: |
| Check by:  Configuration Manager  Name:  Al Qastholani B. Karebollah | Date | Date: | Date: | Date: | Date: | Date: |
| Approved by:  System Developer Manager  Name:  Hii Wei Kai | Date | Date: | Date: | Date: | Date: | Date: |
| Approved by:  System Analyst Manager  Name:  William Soon Siik Yang | Date | Date | Date | Date | Date | Date |
| Authenticated by:  Project Manager  Name:  Zafirah Binti Zailani | Date | Date: | Date: | Date: | Date: | Date: |

|  |  |
| --- | --- |
| **Revision History** | |
| **Revision** | **Description** |
| 01 |  |
| 02 |  |
| 03 |  |
| 04 |  |
| 05 |  |
| 06 |  |

Contents

[1. SCOPE 1](#_Toc437599539)

[1.1. Identification 1](#_Toc437599540)

[1.2. Overview of the System 1](#_Toc437599541)

[1.3. Overview of the Document 2](#_Toc437599542)

[2. REFERENCED DOCUMENTS 3](#_Toc437599543)

[2.1. Overview of the Documents 3](#_Toc437599544)

[2.2. Reference Documents 4](#_Toc437599545)

[3. PRELIMINARY DESIGN 5](#_Toc437599546)

[3.1. System Overview 5](#_Toc437599547)

[3.1.1 System Architecture 5](#_Toc437599548)

[3.1.2. System States and Modes 11](#_Toc437599549)

[3.2. DSMS Design Description 14](#_Toc437599550)

[3.2.1 Student Management 14](#_Toc437599551)

[3.2.2 Trainer Management 15](#_Toc437599552)

[3.2.3 Class Management 16](#_Toc437599553)

[3.2.4 Staff Management 17](#_Toc437599554)

[3.2.5 Vehicle Management 18](#_Toc437599555)

[4. DETAILED DESIGN 19](#_Toc437599556)

[4.1 Student Management 19](#_Toc437599557)

[4.1.1 studentProfile\_model 19](#_Toc437599558)

[4.1.2 studentProfile 24](#_Toc437599559)

[4.1.3 index (studentProfile) 28](#_Toc437599560)

[4.1.4 edit (studentProfile) 31](#_Toc437599561)

[4.1.5 picture (studentProfile) 33](#_Toc437599562)

[4.1.6 studentAttendance\_model 34](#_Toc437599563)

[4.1.7 studentAttendance 37](#_Toc437599564)

[4.1.8 index (studentAttendance) 40](#_Toc437599565)

[4.1.9 studentAssesment\_model 42](#_Toc437599566)

[4.1.10 studentAssesment 44](#_Toc437599567)

[4.1.11 index (studentAssesment) 46](#_Toc437599568)

[4.2 Trainer Management 48](#_Toc437599569)

[4.2.1 trainerAttendence Controllers Class 48](#_Toc437599570)

[4.2.2 trainerAttendence\_model Model Class 51](#_Toc437599571)

[4.2.3 index (trainerAttendence) View Class 53](#_Toc437599572)

[4.2.4 trainerProfile Controllers Class 55](#_Toc437599573)

[4.2.5 trainerProfile\_model Model Class 59](#_Toc437599574)

[4.2.6 index (trainerAttendence) View Class 62](#_Toc437599575)

[4.2.7 edit (trainerProfile) View Class 63](#_Toc437599576)

[4.2.8 picture (trainerProfile) View Class 66](#_Toc437599577)

[4.2.9 trainerStudentAssesment Controllers Class 67](#_Toc437599578)

[4.2.10 trainerStudentAssesment \_model Model Class 69](#_Toc437599579)

[4.2.11 index (trainerStudentAssesment) View Class 72](#_Toc437599580)

[4.2.12 trainerStudentList Controllers Class 74](#_Toc437599581)

[4.2.13 trainerStudentList\_model Model Class 76](#_Toc437599582)

[4.2.14 index (trainerStduentList) View Class 78](#_Toc437599583)

[4.2.15 view (trainerStudentList) View Class 79](#_Toc437599584)

[4.3 Classes Management 83](#_Toc437599585)

[4.3.1 classReserve Controllers Class 83](#_Toc437599586)

[4.3.2 classReserve\_model Model Class 86](#_Toc437599587)

[4.3.3 index (classReserve) View Class 88](#_Toc437599588)

[4.3.4 staffClassApprove Controllers Class 90](#_Toc437599589)

[4.3.5 staffClassApprove\_model Model Class 93](#_Toc437599590)

[4.3.6 index (staffClassApprove) View Class 96](#_Toc437599591)

[4.3.7 edit (staffClassApprove) View Class 97](#_Toc437599592)

[4.4 Staff Management 100](#_Toc437599593)

[4.4.1 staffAttendance 100](#_Toc437599594)

[4.4.2 staffAttendance\_model 103](#_Toc437599595)

[4.4.3 index (staffAttendance) 105](#_Toc437599596)

[4.4.4 staffProfile 107](#_Toc437599597)

[4.4.5 staffProfile\_model 112](#_Toc437599598)

[4.4.6 edit (staffProfile) 117](#_Toc437599599)

[4.4.7 index (staffProfile) 119](#_Toc437599600)

[4.4.8 picture (staffProfile) 123](#_Toc437599601)

[4.4.9 userProfile 124](#_Toc437599602)

[4.4.10 userProfile\_model 129](#_Toc437599603)

[4.4.11 AddUser (userProfile) 133](#_Toc437599604)

[4.4.12 index (userProfile) 137](#_Toc437599605)

[4.4.13 studentEdit (userProfile) 138](#_Toc437599606)

[4.4.14 studentView (userProfile) 142](#_Toc437599607)

[4.4.15 trainerEdit (userProfile) 145](#_Toc437599608)

[4.4.16 trainerView (userProfile) 149](#_Toc437599609)

[4.5 Vehicle Management 152](#_Toc437599610)

[4.5.1 vehicleReverse Controller Class 152](#_Toc437599611)

[4.5.2 vehicleReverse\_model Model Class 156](#_Toc437599612)

[4.5.3 index (vehicleReverse) View Class 160](#_Toc437599613)

[4.5.4 view (vehicleReverse) View Class 162](#_Toc437599614)

[4.5.5 book (vehicleReverse) View Class 164](#_Toc437599615)

[4.5.6 vehicleReverseStaff Controller Class 166](#_Toc437599616)

[4.5.7 vehicleReverseStaff\_model Model Class 172](#_Toc437599617)

[4.5.8 index (vehicleReverseStaff) View Class 176](#_Toc437599618)

[4.5.9 addnew (vehicleReverseStaff) View Class 178](#_Toc437599619)

[4.5.10 view (vehicleReverseStaff) View Class 181](#_Toc437599620)

[4.5.11 recordEdit (vehicleReverseStaff) View Class 184](#_Toc437599621)

[4.5.12 edit (vehicleReverseStaff) View Class 186](#_Toc437599622)

[5. NOTES 188](#_Toc437599623)

[6. CD 189](#_Toc437599624)

**List of Figure**

Figure 3.1 System Design Overview………………………………………………………. 5

Figure 3.2 Static Organization DSMS……………………………………………………... 6

Figure 3.3 Package Views of DSMS……………………………………………………… 6

Figure 3.4 Package/Subsystem Interface………………………………………………….. 10

Figure 3.5 State Diagram in DSMS specific for Login………………………………….… 10

Figure 3.6 State Diagram in DSMS specific for Student Management…………………… 11

Figure 3.7 State Diagram in DSMS specific for Trainer Management……………………. 11

Figure 3.8 State Diagram in DSMS specific for Class Management……………… ……... 12

Figure 3.9 State Diagram in DSMS specific for Staff Management………………………. 12

Figure 3.10 State Diagram in DSMS specific for Vehicle Management……………………. 12

Figure 3.11 Visibility of Student Management……………………………………………… 13

Figure 3.12 Visibility of Trainer Management……………………………………………… 14

Figure 3.13 Visibility of Class Management………………………………………………… 15

Figure 3.14 Visibility of Staff Management………………………………………… ……… 16

Figure 3.15 Visibility of Vehicle Management……………………………………………… 17

# SCOPE

## Identification

System name : Driving School Management System

Abbreviation : DSMS

System ID No.: BTech-01-2015

## Overview of the System

The Driving School System is a system developed to manage the registration of students, keep records of student, user profile and the facilities details of the driving school. But, most of driving schools in Malaysia still use the traditional method using a paper form such as manual registration, a manual arrangement for driving schedule. But in certain driving schools in Malaysia have used a computerized system to manage their operation, but the system is limited which is can be accessed only by administration and staff and this system cannot be accessed by students.

So, this system must be upgraded to web based development, so it can be used by all driving schools in Malaysia and can be accessed by all scopes such as staff, student and instructor. Besides, existing student also can see their instructor profile, insert their attendance, and see their assessment status through this system.

This system provides the details regarding the user of the profile and the details of facilities in the driving school such as the classes and the vehicle. This system consists of three different interfaces which is students, instructors and staff has their respective interface. Staff can control and see all data of students, instructor and facilities, while the instructor and student have their own limited access in the system module.

**Purpose of the System:**

The main purpose of the system is to ease the management of the driving school in term of manage all the data systematically and efficiently. The DSMS brings advantages to the 3 main users of the system which are Student, Staff and also Trainer on the following functions:

1. Helps staff to easily add the student with the trainer and show the details of student to the trainer as well as modify the details of student and trainer.
2. To keep the facilities records, students and trainers information and allow to edit according to the different level of accessibility on different users.
3. Help student to view the assessment and training time on the system.
4. The system attendance taking is also able to take the student and staff attendance record and review and manage by staff.
5. Trainer can book the school car through the system if the car is available and record will be saved and reviewable.

.

## Overview of the Document

This paragraph summarizes the purpose and contents of this Software Design Document (SDD). It specifies the requirement aspects related to the Driving School Management System.

In general, this SDD is divided into 5 sections as the following:

|  |  |
| --- | --- |
| **Chapter 1** | Describes the scope identification, system overview and the document overview |
| **Chapter 2** | Referenced documents, government documents and non-government documents |
| **Chapter 3** | Describes the preliminary design for the DSMS. The preliminary design will identify DSMS, the description of each DSMS design, the characteristics of each DSMS and the traceability of requirements set forth in SRS and IRS. |
| **Chapter 4** | Describes the detail design |
| **Chapter 5** | Notes and abbreviations |

# 2. REFERENCED DOCUMENTS

This section lists the document number and document name referenced in this document. Any discrepancies of this document in describing the software development process should be covered by the documents listed in this chapter. The following documents were referred as the basis for this SDD preparation.

Copies of specification, standards, drawings and publication requested by suppliers in contact with the specified supplying functions may be obtained by contacting the agency or directly through the contracting office.

## 2.1. Overview of the Documents

The purpose of this software design document is to provide the overall description of the system and design models that are developed in order used to satisfy the requirements. This document presents all detailed diagrams that represent the functionality of the system included how should be the system work like from the system’s point of view to provide the necessary interface to the uses. Other than that, this document is useful explain to the user since sometimes understanding will be different between client and developer due to the different fields of knowledge. Below are the list of standard that been used:

1. IEEE Recommend Practice for Software Requirements Specifications

<http://ieeexplore.iee.org/stamp/stamp.jsp?arnumber=00720574&tag=1>

1. IEEE Std 1016-1998, “IEEE Recommended Practice for Software Design Descriptions”, 1998 Edition, IEEE 1998. Retrieved from <http://standards.ieee.org/findstds/standard/1016-1998.html>
2. Software Requirements Specification (SRS) Military Standard, version 1.0 (5 December 1994)
3. Software Design Engineering Education Requirements, Retrieved from  
   <http://study.com/articles/Software_Design_Engineering_Educational_Requirements_for_Becoming_a_Design_Engineer.html> 1st April 2015, 1130pm.
4. Bell, Donald. "UML basics: The component diagram." *IBM Global Services* (2004). Retrieved from

<http://www.softwareresearch.net/fileadmin/src/docs/teaching/WS13/SE/UML_basics-_The_component_diagram.pdf>

1. Somenzi, Fabio. "CUDD: Colorado university decision diagram package." Retrieved from <https://hal.inria.fr/inria-00121468/document> (1996).
2. Bourque, P., & Fairley, R. E. (2014). Guide to the Software Engineering Body of Knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press.
3. Gamma, Erich, et al. Design Patterns: elements of reusable object-oriented software. Pearson Education, 1994. Retrieved from <https://books.google.com/books?hl=en&lr=&id=6oHuKQe3TjQC&oi=fnd&pg=PT2&dq=state+diagram+in+software+design+document+&ots=lOmKGTbONH&sig=N2-6-TehnjoGC5o6VNvP2xFGgw4>
4. Hua, C. H. E. N. “MVC Design Pattern. “Computer Knowledge and Technology (Academic Exchange) 14 (2007): 051. Retrieved from <http://en.cnki.com.cn/Article_en/CJFDTOTAL-DNZS200714051.htm>
5. Leff, Avraham, and James T. Rayfield. “Web-application development using the model/view/controller design pattern.” Enterprise Distributed Object Computing Conference, 2001. EDOC’01. Proceedings. Fifth IEEE International. IEEE, 2001. Retrieved from <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=950428>

## 2.2. Reference Documents

This section consists of the contractual documents and non-contractual documents.

1. Beta Tech Sdn Bhd, BTech-01-2015, Software Requirement Specification (SRS) Document for Driving School Management System (2015)
2. Request For Proposal (RFP) of Driving School Management System (BCS2333-01)(2015)
3. Scott W.Ambler (2003). UML 2 Sequence Diagramming Guideline, Agile Modelling. Retrieved from [http://agilemodeling.com/style/sequenceDiagram.htm 1st May,2015](http://agilemodeling.com/style/sequenceDiagram.htm%201st%20May,2015)

# 3. PRELIMINARY DESIGN

## 3.1. System Overview

Figure 1 shows the relationship between Staff, Trainer and Student with Driving School Management System.

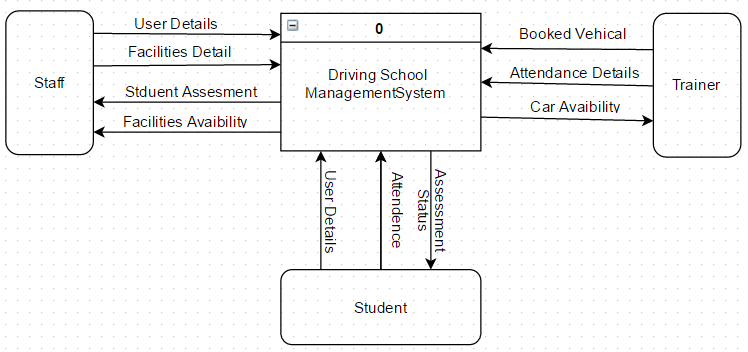


Figure 3.1 System Design Overview

### 3.1.1 System Architecture

This paragraph identifies the internal organizational structure of the system. The relationship among system subsystem will be described.

#### 3.1.1.1 Static Organization

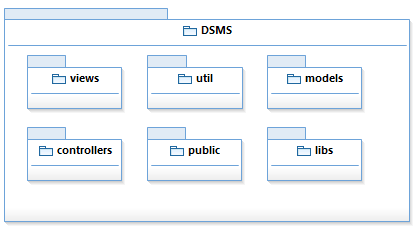


Figure 3.2: Static Organization of DSMS

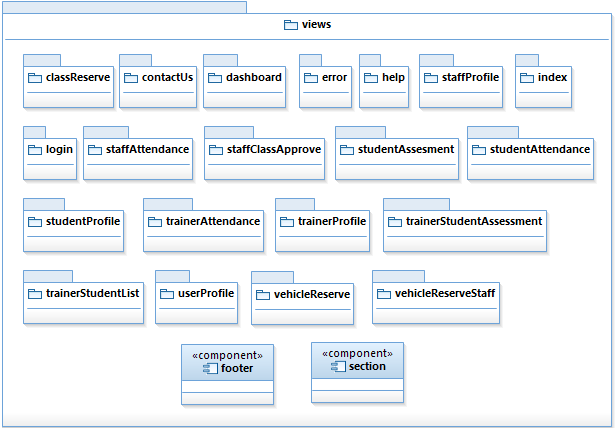


Figure 3.3: Package View of DSMS

This section describes the detail for each subsystem/package.

* 1. **Student Management**

This package responsibility is to manage the information of student, the attendance, and user profile and student assessment. The package provides the means of student to see and update their certain details and add their attendance.

1. studentProfile\_Model Model class
2. studentProfile Controller class
3. index (studentProfile) View class
4. edit (studentProfile) View class
5. picture (studentProfile) View class
6. studentAttendace\_Model class Model
7. studentAttendance class Controller
8. index (studentAttendance) View class
9. studentAssesment\_Model class Model
10. studentAssesment class Controller
11. index (studentAssesment) View class
    1. **Trainer Management**

This sub system responsibility is to manage the information of trainer, trainer attendance, the student assessment and the trainer student list. The sub system provides the means of trainer to see and update their certain details and add their attendance. The subsystem also allows the entity to view the student profile that the trainer teach and update the student assessment status.

1. trainerAttendence Controller Class
2. trainerAttendence\_model Model Class
3. index (trainerAttendence) View Class
4. trainerProfile Controller Class
5. trainerProfile\_model Model Class
6. index (trainerProfile) View Class
7. edit (trainerProfile) View Class
8. picture (trainerProfile) View Class
9. trainerStudentAssessment Controller Class
10. trainerStudentAssessment\_model Model Class
11. index (trainerStudentAssessment) View Class
12. trainerStudentList Controll Class
13. trainerStudentList\_model Model Class
14. index (trainerStudentList) View Class
15. view (trainerStudentList) View Class
    1. **Class Management**

This package is to manage the three main classes in the driving school which are hall talks, lab and on road class. This package provides the means to add new booking class, update booking class and delete booking as well as displays booked class lists. The package helps staff and trainer to manage the classes in term of making appointment and manage appointment.

1. classReserve Controller Class
2. classReserve \_model Model Class
3. index (classReserve) View Class
4. staffClassApprove Controller Class
5. staffClassApprove\_model Model Class
6. index (staffClassApprove) View Class
7. edit (staffClassApprove) View Class
   1. **Staff Management**

The role of this subsystem is to manage function such as attendance, car, search user and classes. This package consists of the following classes or unit:

1. staffProfile\_Model class
2. staffProfile Controller class
3. index (staffProfile) View class
4. edit (staffProfile) View class
5. picture (staffProfile) View class
6. staffAttendace\_Model class Model
7. staffAttendace class Controller
8. index (staffAttendace) View class
9. userProfile\_Model class Model
10. userProfile class Controller
11. index (userProfile) View class
12. AddUser (userProfile) View class
13. studentEdit (userProfile) View class
14. studentView (userProfile) View class
15. trainerEdit (userProfile) View class
16. trainerView (userProfile) View class
    1. **Vehicle Management**

The role of this subsystem is to manage the car in driving school management. All the related interaction about car will handle under this subsystem. This package allow trainee to view a list of vehicles with its details include maintenance record, and so on in order to book the vehicle for the training purpose. The package also allows staff to update the detail and maintenance record once they send the car for the service and approve the booking from the trainer. Lastly, this package consists of the following classes or units.

1. vehicleReverse Controller Class
2. vehicleReverse\_model Model Class
3. index (vehicleReverse) View Class
4. view (vehicleReverse) View Class
5. book (vehicleReverse) View Class
6. vehicleReserveStaff Controller Class
7. vehicleReserveStaff\_model Model Class
8. index (vehicleReserveStaff) View Class
9. addnew (vehicleReserveStaff) View Class
10. view (vehicleReserveStaff) View Class
11. recordEdit (vehicleReserveStaff) View Class
12. edit (vehicleReserveStaff) View Class

#### 3.1.1.2. Dynamic Organization

Not Applicable

#### 3.1.1.3. DSMS Interfaces

Figure 3.4 shows all the interaction between package / subsystem and non-development external subsystem/unit. Further description for external package describe in section 3.1.1.4

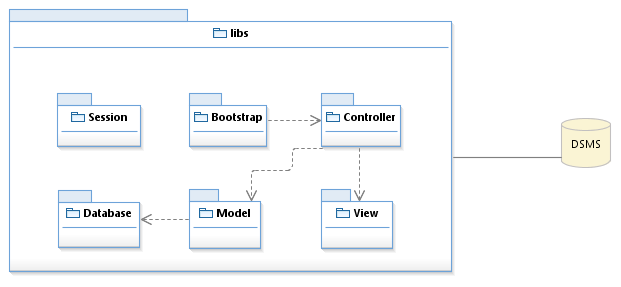


Figure 3.4: Package/Subsystem Interfaces

#### 3.1.1.4. External Interfaces

Not Applicable

### 3.1.2. System States and Modes

This section describes states diagrams for DBMS.



Figure 3.5 State Diagram in DBMS specific for Login

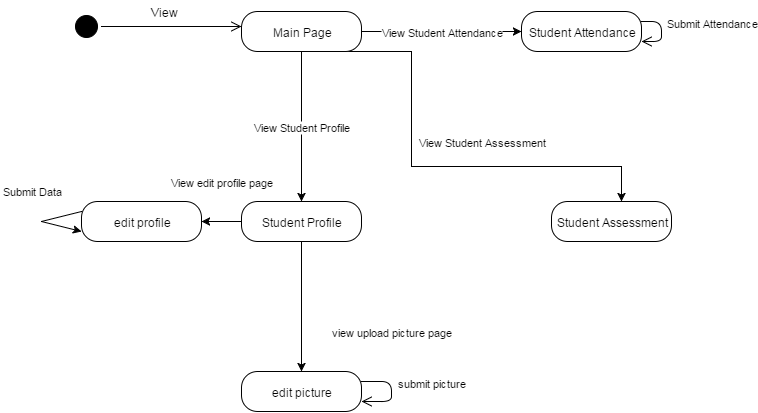


Figure 3.6: State Diagram in DBMS specific for student management

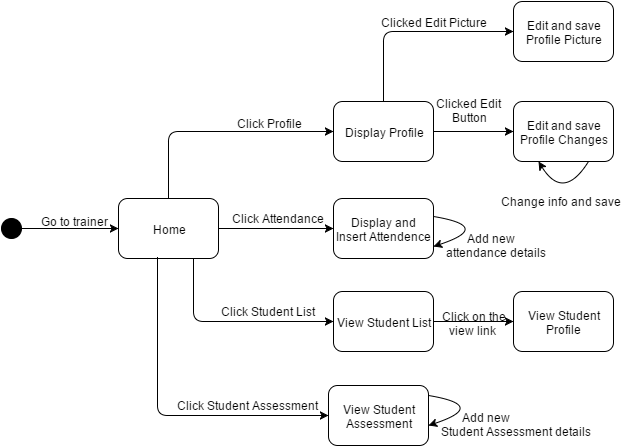


Figure 3.7: State Diagram in DSMS specific for Trainer Management

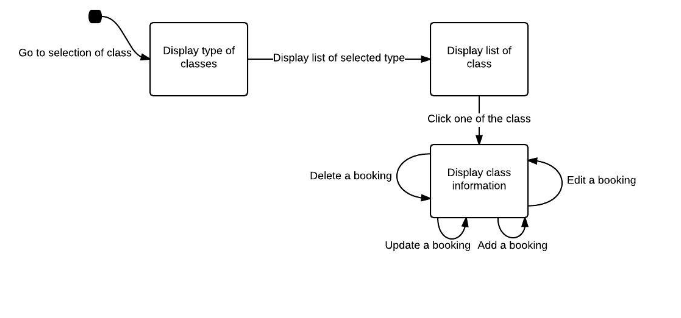


Figure 3.8: State Diagram in DSMS specific for Class Management

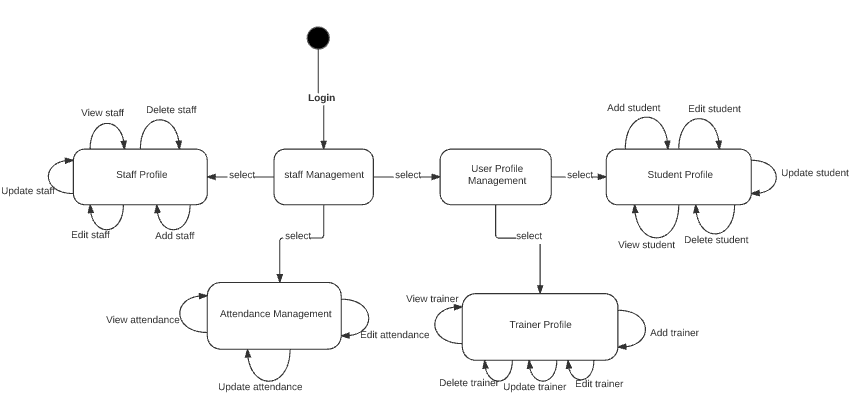


Figure 3.9: State Diagram in DSMS specific for Staff Management

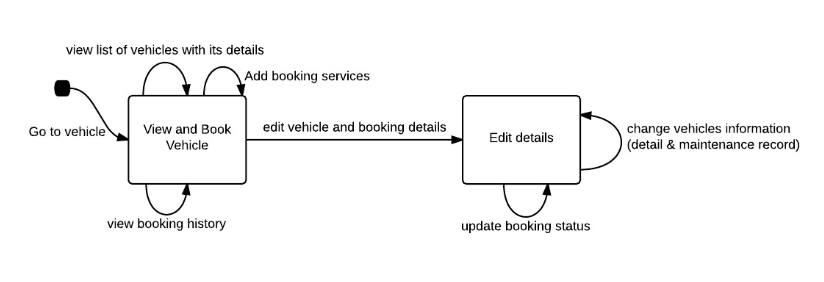


Figure 3.10 State Diagram in DSMS specific for Vehicle Management

## 3.2. DSMS Design Description

### 3.2.1 Student Management

Figure 3.11 shows the visibility of student management relationship between each subsystem

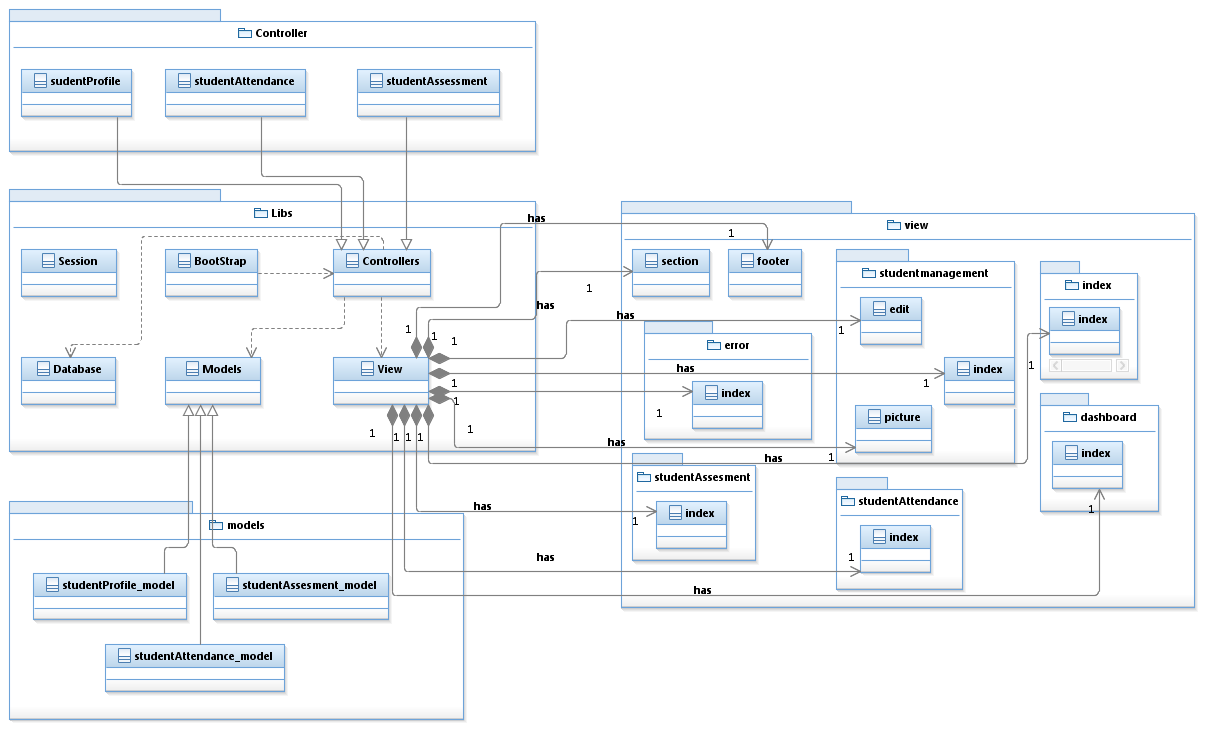


Figure 3.11: Visibility of Student Management

### 3.2.2 Trainer Management

Figure 3.12 show the visibility of the trainer subsystem and the relationship between each of the component related to the trainer subsystem.

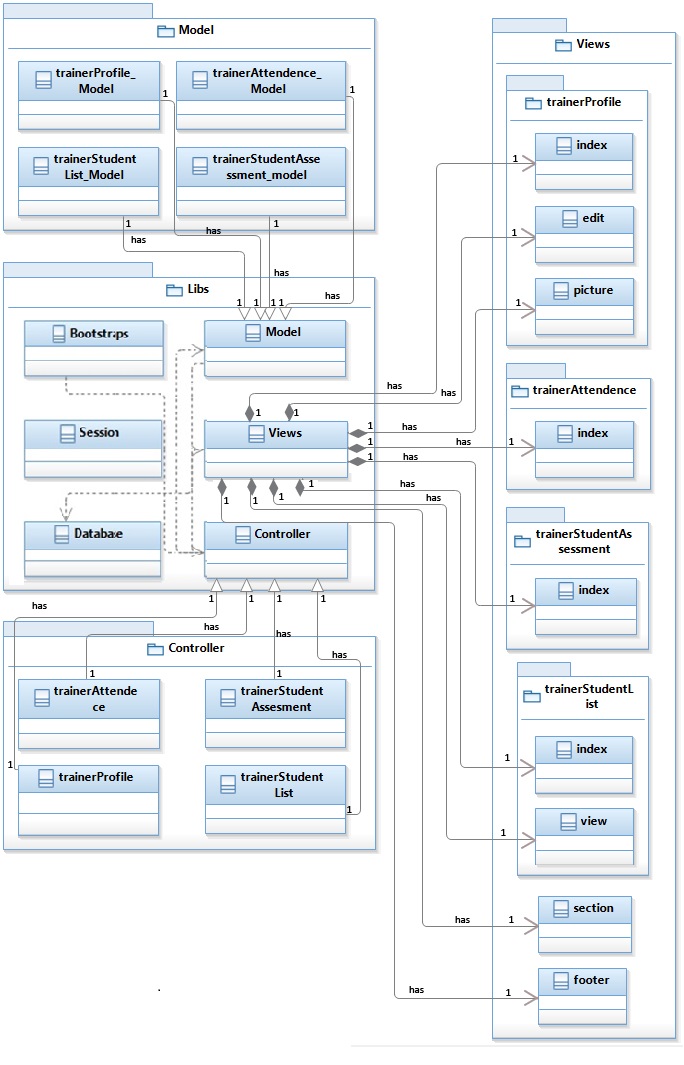


Figure 3.12: Visibility of Trainer Management

### 3.2.3 Class Management

Figure 3.13 shows package of classes that consists of classReserve controller, staffClassApprove controller, classReserve\_model Model, staffClassApprove\_model Model, index (classReserve) View, index (staffClassApprove) View and edit View.

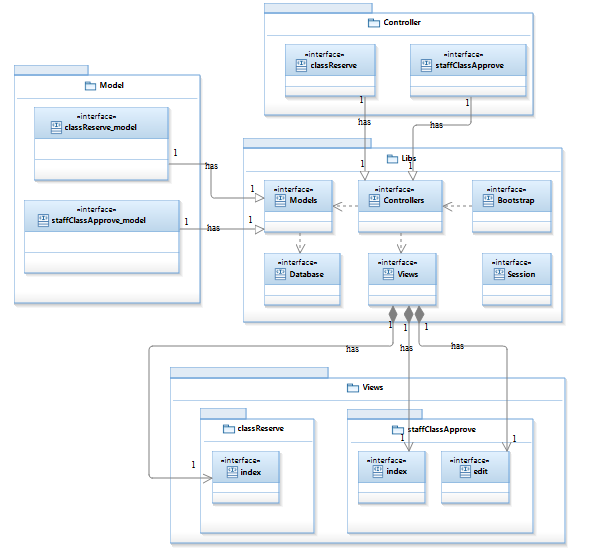


Figure 3.13: Visibility of classes subsystem

### 3.2.4 Staff Management

Figure 3.14 shows package of Staff Management that consists of staffAttendance View, staffAttendance Controller, staffAttendance \_model, userProfile View, userProfile Controller, userProfile \_model, staffProfile View, staffProfile Controller, staffProfile \_model.

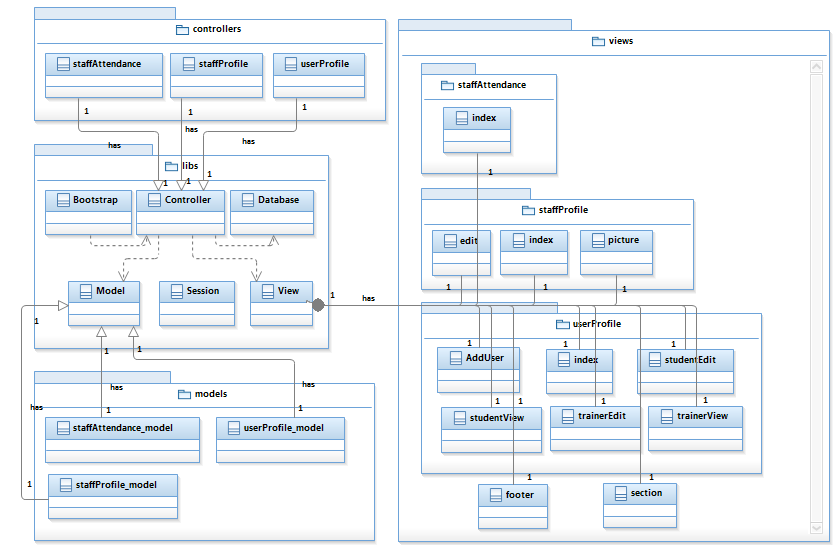


Figure 3.14: Visibility of staff Management

### 3.2.5 Vehicle Management

The Vehicle Subsystemis responsible for all interaction involving car. All interactions with any car will be handled through here. The car subsystem provides a list of all vehicles with its detail (plate number, road tax, brand, maintenance record, etc) and availability (booking history). This subsystem will handle two types of user which are trainer and staff of the system. Trainer is allow come to here to book the car for the training purpose. Besides, staffs are allowed only to edit the maintenance record of the car and approve the booking status. Both the car detail and booking record will each be responsible for utilizing the appropriate and necessary method within the database to insert, update, and display all the corresponding data on the system.

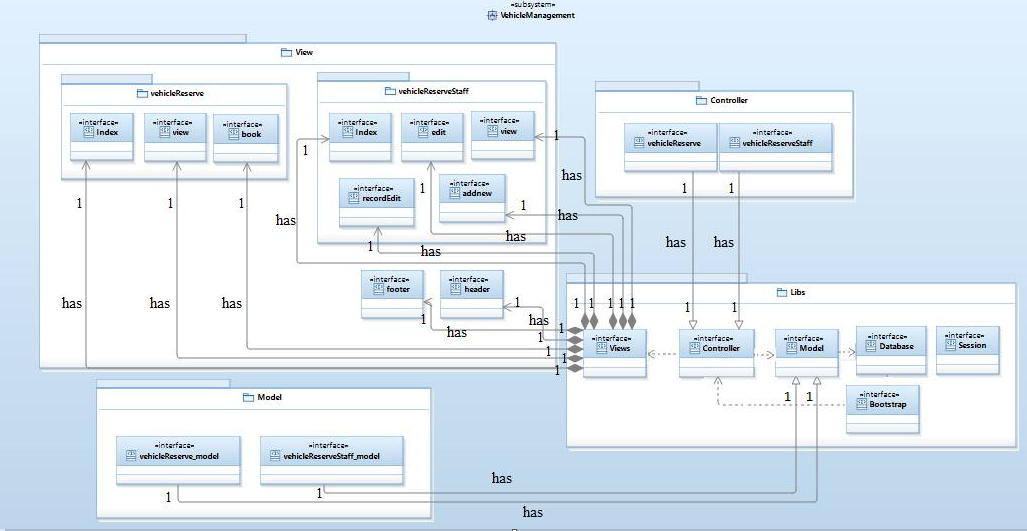


Figure 3.15: Visibility of Vehicle Management

# 4. DETAILED DESIGN

This section divided into the following paragraphs and subparagraphs to describe the detailed design.

## 4.1 Student Management

This package consists of:

1. studentProfile\_Model class Model
2. studentProfile class Controller
3. index class View (studentProfile)
4. edit class View (studentProfile)
5. picture class View (studentProfile)
6. studentAttendace\_Model class Model
7. studentAttendance class Controller
8. index class View (studentAttendance)
9. studentAssesment\_Model class Model
10. studentAssesment class Controller
11. index class View (studentAssesment)

### 4.1.1 studentProfile\_model

The purpose of this class is to create object from another classes into it. It acts as a platform to let controller class to update and supply data for view class.

#### 4.1.1.1 Class Model studentProfile\_Model design

This subparagraph specifies the design of studentProfile\_model

1. **Input/Output data elements**

List of input and output data elements:

Input : data, id, password, address, gender, religion, phoneNum, email, picture

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data

|  |  |
| --- | --- |
| Name | data |
| Description | To pass data using array |
| Data Type | Array |
| Precision/resolution | - |

Table 2: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Student log in id |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | Student address |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | Student Gender |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for picture

|  |  |
| --- | --- |
| Name | picture |
| Description | Student profile picture |
| Data Type | Binary Large OBject |
| Precision/resolution | - |

Table 6: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | Student can change their password |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | Student phone number |
| Data Type | Integer |
| Precision/resolution | - |

Table 8: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | Student religion |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | Student email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : Model class

Responsibility : To insert and update data for picture and profile student

Attributes : data, id, password, address, gender, religion, phonenum, email, picture

Methods: \_\_construct(),studentProfileEdit(),

studentProfileEdited(id),editSave(data),editPicture(data)

**1. studentProfileEdit()**

Responsibility : to view data for student profile

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

SELECT \* from full WHERE id = id

END

**2.studentProfileEdit(id)**

Responsibility : to display certain student profile

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

SELECT \* from full WHERE id = id

END

**3.editsave (data)**

Responsibility : to update data

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

UPDATE full

SET `password` = :password,`address` = :address,`gender` = :gender,`religion` = :religion, `phoneNum` = :phoneNum, `email` = :email

WHERE id = :id

END

**4.editPicture (data)**

Responsibility : to display certain student profile

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

SELECT \* from full WHERE id = id

END

**5.\_\_construct ()**

Responsibility : act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm: None

### 4.1.2 studentProfile

The purpose of this class is to check and send the data to model class

#### 4.1.2.1 Class Controller studentProfile design

This subparagraph specifies the design of studentProfile

1. **Input/Output data elements**

List of input and output data elements:

Input : data, id, password, address, gender, religion, phonenum, email, imgData

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data

|  |  |
| --- | --- |
| Name | data |
| Description | To pass data using array |
| Data Type | Array |
| Precision/resolution | - |

Table 2: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Student log in id |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | Student address |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | Student Gender |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for imgData

|  |  |
| --- | --- |
| Name | imgData |
| Description | Where picture has been upload |
| Data Type | Image |
| Precision/resolution | - |

Table 6: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | Student can change their password |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for phonenum

|  |  |
| --- | --- |
| Name | phonenum |
| Description | Student phone number |
| Data Type | Integer |
| Precision/resolution | - |

Table 8: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | Student religion |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | Student email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : Controller class

Responsibility : To send data into model class

Attributes :studentName, studentIC, studentAddress, studentPicture, studentDOB, studentTrainer

Methods : \_\_construct(), index(), edit(id), picture(id), editsave(id), editpicture(id)

**1. \_\_construct()**

Responsibility : act as parent consctuctor

Input Parameter : None

Output Parameter : None

Algorithm : None

**2.index()**

Responsibility : to render or view the student profile

Input Parameter : None

Output Parameter : None

Algorithm : None

**3.edit(id)**

Responsibility : to render or view the student profile

Input Parameter : None

Output Parameter : None

Algorithm : None

**4.picture(id)**

Responsibility : to render picture page

Input Parameter : None

Output Parameter : None

Algorithm : None

**5.editSave(id)**

Responsibility : to send data to model class

Input Parameter : None

Output Parameter : None

Algorithm : None

**6.editPicture(id)**

Responsibility : to send image data to model class

Input Parameter : None

Output Parameter : None

Algorithm : None

### 4.1.3 index (studentProfile)

The purpose of this class is to view the data from database

#### 4.1.3.1 Class view index design (studentProfile)

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : picture, id, name, ic, role, gender, address, dob, religion, race, phonenum, email

1. **Local data elements**

Table 1: Local Data Definition for picture

|  |  |
| --- | --- |
| Name | picture |
| Description | Display profile picture of the student |
| Data Type | Jpeg |
| Precision/resolution | Weight = 190, height = 190 |

Table 2: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Id of the student |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | Student name |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for ic

|  |  |
| --- | --- |
| Name | ic |
| Description | Student phone number |
| Data Type | Integer |
| Precision/resolution | - |

Table 5: Local Data Definition for role

|  |  |
| --- | --- |
| Name | role |
| Description | To display role of the user |
| Data Type | String |
| Precision/resolution | - |

Table 6: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | Student gender |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | Student address |
| Data Type | String |
| Precision/resolution | - |

Table 8: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | Student date of birth |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | Religion |
| Description | Student religion |
| Data Type | String |
| Precision/resolution | - |

Table 10: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | Student races |
| Data Type | String |
| Precision/resolution | - |

Table 11: Local Data Definition for phonenum

|  |  |
| --- | --- |
| Name | phonenum |
| Description | Student phone number |
| Data Type | Integer |
| Precision/resolution | - |

Table 12: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | Student email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data and picture of student

Attributes : picture, id, name, ic, role, gender, address, dob, religion, race, phonenum, email

Methods **:** None

### 4.1.4 edit (studentProfile)

The purpose of this class is to view and update the data.

#### 4.1.4.1 Class view edit design (studentProfile)

This subparagraph specifies the design of edit

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : password, address, gender, religion, phonenum, email

1. **Local data elements**

Table 1: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | Display profile picture of the student |
| Data Type | Jpeg |
| Precision/resolution | Weight = 190, height = 190 |

Table 2: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | Student gender |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | Student address |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | Religion |
| Description | Student religion |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for phonenum

|  |  |
| --- | --- |
| Name | phonenum |
| Description | Student phone number |
| Data Type | Integer |
| Precision/resolution | - |

Table 6: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | Student email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data of student profile

Attributes : password, address, gender, religion, phonenum, email

Methods **:** None

### 4.1.5 picture (studentProfile)

The purpose of this class is to insert or update profile picture

#### 4.1.5.1 Class view picture design (studentProfile)

This subparagraph specifies the design of picture

1. **Input/Output data elements**

List of input and output data elements:

Input : file

Output : None

1. **Local data elements**

Table 1: Local Data Definition for file

|  |  |
| --- | --- |
| Name | file |
| Description | Image to upload |
| Data Type | Jpeg |
| Precision/resolution | 65535 bytes |

1. **Algorithms**

Class Type : View Class

Responsibility : To insert or update new profile picture

Attributes : picture

Methods **:** post

### 4.1.6 studentAttendance\_model

The purpose of this class is to insert or update attendance of student

#### 4.1.6.1 Class studentAttendance\_model design

This subparagraph specifies the design of student attendance model

1. **Input/Output data elements**

List of input and output data elements:

Input : data, id, name, class, date, timein, timeout

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data

|  |  |
| --- | --- |
| Name | data |
| Description | Array of data |
| Data Type | array |
| Precision/resolution | **-** |

Table 2: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Id of student |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | Student name |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for class

|  |  |
| --- | --- |
| Name | class |
| Description | Class that student want to attend |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Date when student to attend |
| Data Type | String |
| Precision/resolution | - |

Table 6: Local Data Definition for timein

|  |  |
| --- | --- |
| Name | timein |
| Description | Time student attend |
| Data Type | time |
| Precision/resolution | - |

Table 7: Local Data Definition for timeout

|  |  |
| --- | --- |
| Name | timeout |
| Description | Time when student finish the class |
| Data Type | time |
| Precision/resolution | - |

1. **Algorithms**

Class Type : Model Class

Responsibility : To insert student attendance data

Attributes : data, id, name, class, date, timein, timeout

Methods **:** \_\_construct(), AttendanceList(), create(data)

**1.\_\_construct ()**

Responsibility : act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm: None

**2.AttendanceList()**

Responsibility : to display all attendance of this student

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

('SELECT \* FROM studentattendance WHERE id = :id',

array('id' => $\_SESSION['id']));

END

**3.create(data)**

Responsibility : to display all attendance of this student

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

('INSERT INTO studentattendance(`id`, `name`,`class`,`date`,`timeIn`,`timeOut`)

VALUES (:id, :name, :class ,:date,:timeIn,:timeOut)');

END

### 4.1.7 studentAttendance

The purpose of this class is to check and send the data to model class

#### 4.1.7.1 Class Controller studentAttendance design

This subparagraph specifies the design of studentAttendance

1. **Input/Output data elements**

List of input and output data elements:

Input : data, id, name, class, date, timein, timeout

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data

|  |  |
| --- | --- |
| Name | data |
| Description | Array of data |
| Data Type | array |
| Precision/resolution | **-** |

Table 2: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Id of student |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | Student name |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for class

|  |  |
| --- | --- |
| Name | class |
| Description | Class that student want to attend |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Date when student to attend |
| Data Type | String |
| Precision/resolution | - |

Table 6: Local Data Definition for timein

|  |  |
| --- | --- |
| Name | timein |
| Description | Time student attend |
| Data Type | time |
| Precision/resolution | - |

Table 7: Local Data Definition for timeout

|  |  |
| --- | --- |
| Name | timeout |
| Description | Time when student finish the class |
| Data Type | time |
| Precision/resolution | - |

1. **Algorithms**

Class Type : Contoller Class

Responsibility : To insert student attendance data

Attributes : data, id, name, class, date, timein, timeout

Methods **:** \_\_construct(), index(), create()

**1.\_\_construct ()**

Responsibility : act as parent constructor and also to check whether the student has been log in or not. If not it will directly into login page

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

if ($logged == false) {

Session::destroy();

header('location: ../login');

exit;

}

END

**2.index()**

Responsibility : to display all attendance of this student

Input Parameter : None

Output Parameter : None

Algorithm: None

**3.create()**

Responsibility : to send data of student using array to the model

Input Parameter : None

Output Parameter : None

Algorithm: None

### 4.1.8 index (studentAttendance)

The purpose of this class is to view the data from database

#### 4.1.3.1 Class view index design (studentAttendance)

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : id, name, class, date, timein, timeout

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Student id |
| Data Type | String |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | Name |
| Description | Name of the student |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for class

|  |  |
| --- | --- |
| Name | class |
| Description | Type of class student want to attend |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Date of the student attend |
| Data Type | date |
| Precision/resolution | - |

Table 5: Local Data Definition for timein

|  |  |
| --- | --- |
| Name | timein |
| Description | Time student enter the class |
| Data Type | time |
| Precision/resolution | - |

Table 6: Local Data Definition for timeout

|  |  |
| --- | --- |
| Name | timeout |
| Description | Time student leave the class |
| Data Type | time |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data and picture of student

Attributes : id, name, class, date, timein, timeout

Methods **:** None

### 4.1.9 studentAssesment\_model

The purpose of this class is to insert or update attendance of student

#### 4.1.9.1 Class studentAssesment\_model design

This subparagraph specifies the design of student assesment model

1. **Input/Output data elements**

List of input and output data elements:

Input : id

Output : None

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | To display the student assessment by student id |
| Data Type | String |
| Precision/resolution | **-** |

1. **Algorithms**

Class Type : Model Class

Responsibility : To retrieve student assessment

Attributes : bil, id, trainerId, assessment, status, date

Methods **:** \_\_construct(), AssesmentList()

**1.\_\_construct ()**

Responsibility : act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm: None

**2.AssesmentList()**

Responsibility : to display all assesment of this student

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

('SELECT \* FROM studentassessment WHERE id = :id',

array('id' => $\_SESSION['id']));

END

### 4.1.10 studentAssesment

The purpose of this class is to check and send the data to model class

#### 4.1.10.1 Class Controller studentAssesment design

This subparagraph specifies the design of studentAssesment

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : None

1. **Local data elements**
2. **Algorithms**

Class Type : Contoller Class

Responsibility : To render index page to display the data

Attributes : None

Methods **:** \_\_construct(), index()

**1.\_\_construct ()**

Responsibility : act as parent constructor and also to check whether the student has been log in or not. If not it will directly into login page

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

if ($logged == false) {

Session::destroy();

header('location: ../login');

exit;

}

END

**2.index()**

Responsibility : to display the index page. Without this function index page not working

Input Parameter : None

Output Parameter : None

Algorithm: None

### 4.1.11 index (studentAssesment)

The purpose of this class is to view the data from database

#### 4.1.11.1 Class view index design (studentAssesmesnt)

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : date, assessment, status

1. **Local data elements**

Table 1: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | To display of the date that student attend the class |
| Data Type | date |
| Precision/resolution | - |

Table 2: Local Data Definition for assessment

|  |  |
| --- | --- |
| Name | assessment |
| Description | To display the assessment |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for status

|  |  |
| --- | --- |
| Name | status |
| Description | To display the status of the assessment of the student whether pass or fail |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view date, assessment and status of the student

Attributes : date, assessment, status

Methods **:** None

## 4.2 Trainer Management

This package consists of:

1. trainerAttendence Controller Class
2. trainerAttendence\_model Model Class
3. index (trainerAttendence) View Class
4. trainerProfile Controller Class
5. trainerProfile\_model Model Class
6. index (trainerProfile) View Class
7. edit (trainerProfile) View Class
8. picture (trainerProfile) View Class
9. trainerStudentAssessment Controller Class
10. trainerStudentAssessment\_model Model Class
11. index (trainerStudentAssessment) View Class
12. trainerStudentList Controll Class
13. trainerStudentList\_model Model Class
14. index (trainerStudentList) View Class
15. view (trainerStudentList) View Class

### 4.2.1 trainerAttendence Controllers Class

The purpose of this class is to control which model to be use and check for data correctness.

#### 4.2.1.1 Class trainerAttendece Design

This subparagraph specifies the design of trainerAttendence

1. **Input/Output data elements**

List of input and output data elements:

Input : none

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), index(), create(),

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view to the model

Input Parameter : None

Output Parameter : Render to the trainerAttendece index page

Algorithm :

BEGIN

$this->view->trainerAttendanceList = $this->model-> trainerAttendanceList ();

$this->view->render('trainerAttendance/index');

END

**3.create()**

Responsibility : To create the data in array format

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$this->model->create($data);

END

### 4.2.2 trainerAttendence\_model Model Class

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store/retrieve/delete the selected data from/to the database

#### 4.2.2.1 Class trainerAttendence\_model Design

This subparagraph specifies the design of trainerAttendence\_model

1. **Input/Output data elements**

List of input and output data elements:

Input : $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database

Attributes : None

Methods : construct(),trainerAttendanceList(), create()

**1. construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

**2. trainerAttendenceList()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM trainerattendance WHERE id = :id ORDER BY date ', array('id' => $\_SESSION['id']));

END

**3.create()**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('INSERT INTO trainerattendance’),

$sth->execute(array)

END

### 4.2.3 index (trainerAttendence) View Class

The purpose of this class is to display and get input from the user

#### 4.2.3.1 Class index Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : id,name,date,timeIn,timeOut

Output : date,timeIn,timeOut

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Get the user id from the logged session |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | Get the user name from the logged session |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Get the input of date from the user |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for timeIn

|  |  |
| --- | --- |
| Name | timeIn |
| Description | Get the input of time in from the user |
| Data Type | time |
| Precision/resolution | - |

Table 5: Local Data Definition for timeOut

|  |  |
| --- | --- |
| Name | timeOut |
| Description | Get the input of time out from the user |
| Data Type | time |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display and get input from the user

Attributes : None

Methods : None

### 4.2.4 trainerProfile Controllers Class

The purpose of this class is to control which model to be use and check for data correctness.

#### 4.2.4.1 Class trainerProfile Design

This subparagraph specifies the design of trainerAttendence

1. **Input/Output data elements**

List of input and output data elements:

Input : $id,$data

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for id array

|  |  |
| --- | --- |
| Name | id |
| Description | The id use as parameter |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), index(), edit($id), picture($id), editSave($id), editPicture($id)

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view class to the model class

Input Parameter : None

Output Parameter : Render to the trainerProfile index page

Algorithm :

BEGIN

$this->view->trainerProfileList = $this->model->trainerProfileList();

$this->view->render('trainerProfile/index');

END

**3.edit($id)**

Responsibility : To render from view class to the model class

Input Parameter : $id

Output Parameter : Render to the trainerProfile edit page

Algorithm :

BEGIN

$this->view->trainerProfileEdit = $this->model->trainerProfileEdit($id);

$this->view->render('trainerProfile/edit');

END

**4.picture($id)**

Responsibility : To render from view class to the model class

Input Parameter : $id

Output Parameter : Render to the trainerProfile picture

Algorithm :

BEGIN

$this->view->trainerProfileEdit = $this->model->trainerProfileEdit($id);

$this->view->render('trainerProfile/picture');

END

**5.editSave($id)**

Responsibility : To control the data from view class to the model class

Input Parameter : $id

Output Parameter : $data

Algorithm :

BEGIN

$this->model->editSave($data);

END

**6.editPicture($id)**

Responsibility : To control the data from view class to the model class

Input Parameter : $id

Output Parameter : $data

Algorithm :

BEGIN

$this->model->editPicture($data);

END

### 4.2.5 trainerProfile\_model Model Class

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store/retrieve/delete the selected data from/to the database

#### 4.2.5.1 Class trainerProfile\_model Design

This subparagraph specifies the design of trainerProfile\_model

1. **Input/Output data elements**

List of input and output data elements:

Input : $id, $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for id array

|  |  |
| --- | --- |
| Name | id |
| Description | The id use as parameter |
| Data Type | array |
| Precision/resolution | - |

Table 2: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database

Attributes : None

Methods : construct(),trainerProfileList(),trainerProfileEdit($id), editSave($id), editPicture($data)

**1. construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

**2. trainerProfileList()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE id = :id', array('id' => $\_SESSION['id']));

END

**3. trainerProfileEdit($id)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : $id

Output Parameter : Return data to view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE id = :id', array(':id' => $id));

END

**4.editSave($data)**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('UPDATE full $data),

$sth->execute(array)

END

**5.editPicture($data)**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('UPDATE full $data),

$sth->execute(array)

END

### 4.2.6 index (trainerAttendence) View Class

The purpose of this class is to display and get input from the user

#### 4.2.6.1 Class index Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : $value

1. **Local data elements**

Table 1: Local Data Definition for value

|  |  |
| --- | --- |
| Name | value |
| Description | The attribute that are stored in the database in array form |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display output to the user

Attributes :id, name, ic, role, gender, address, dob, religion, race, phoneNum, email

Methods : None

### 4.2.7 edit (trainerProfile) View Class

The purpose of this class is to display and get input from the user

#### 4.2.7.1 Class edit Design

This subparagraph specifies the design of edit

1. **Input/Output data elements**

List of input and output data elements:

Input : password, gender, address, dob, religion, race, phoneNum, email

Output : None

1. **Local data elements**

Table 1: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | Get the user password |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | Get the user gender |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | Get the user address |
| Data Type | varchar |
| Precision/resolution | - |

Table 4: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | Get the user date of born |
| Data Type | date |
| Precision/resolution | - |

Table 5: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | Get the user religion |
| Data Type | varchar |
| Precision/resolution | - |

Table 6: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | Get the user race |
| Data Type | varchar |
| Precision/resolution | - |

Table 7: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | Get the user phone number |
| Data Type | integer |
| Precision/resolution | - |

Table 8: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | Get the user email |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To get input from the user

Attributes : password, gender, address, dob, religion, race, phoneNum, email

Methods : None

### 4.2.8 picture (trainerProfile) View Class

The purpose of this class is to get input from the user

#### 4.2.8.1 Class picture Design

This subparagraph specifies the design of picture

1. **Input/Output data elements**

List of input and output data elements:

Input : inputFile

Output : None

1. **Local data elements**

Table 1: Local Data Definition for inputFile

|  |  |
| --- | --- |
| Name | inputFile |
| Description | Get the user picture |
| Data Type | blob |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To get input from the user

Attributes : inputFile

Methods : None

### 4.2.9 trainerStudentAssesment Controllers Class

The purpose of this class is to control which model to be use and check for data correctness.

#### 4.2.9.1 Class t trainerStudentAssesment Design

This subparagraph specifies the design of trainerStudentAssesment

1. **Input/Output data elements**

List of input and output data elements:

Input : none

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), index(), create()

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view to the model

Input Parameter : None

Output Parameter : Render to the trainerStudentAssessment index page

Algorithm :

BEGIN

$this->view->trainerStudentAssessmentList = $this->model->trainerStudentAssessmentList();

$this->view->studentAssessment = $this->model->studentAssessment();

$thisv->iew->render('trainerStudentAssessment/index');

END

**3.create()**

Responsibility : To create the data in array format

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$this->model->create($data);

END

### 4.2.10 trainerStudentAssesment \_model Model Class

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store/retrieve/delete the selected data from/to the database

#### 4.2.10.1 Class trainerStudentAssesment \_model Design

This subparagraph specifies the design of trainerStudentAssesment \_model

1. **Input/Output data elements**

List of input and output data elements:

Input : $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database

Attributes : None

Methods :construct(),trainerStudentAssessmentList(), studentAssessment(), create($data), delete($bil))

**1. construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

**2. trainerStudentAssessmentList()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE trainerId = :id ORDER BY id ', array('id' => $\_SESSION['id']));

END

**3. trainerStudentAssessmentList()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM studentassessment WHERE trainerId = :id ORDER BY id ', array('id' => $\_SESSION['id'])); END

**4.create($data)**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('INSERT INTO studentassessment’),

$sth->execute(array)

END

**4.delete($bil)**

Responsibility : To delete the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$result = $this->db->select('SELECT bil FROM studentassessment WHERE bil = :bil', array(':bil' => $bil));

$this->db->delete('studentassessment', "bil = '$bil'");

END

### 4.2.11 index (trainerStudentAssesment) View Class

The purpose of this class is to display and get input from the user

#### 4.2.11.1 Class index Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : id,date,assessment,status

Output : id,date,assessment,status

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Get the user id from the logged session and the student Id from the user input |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Get the input of date from the user |
| Data Type | date |
| Precision/resolution | - |

Table 3: Local Data Definition for assessment

|  |  |
| --- | --- |
| Name | assessment |
| Description | Get the input of assessment from the user |
| Data Type | varchar |
| Precision/resolution | - |

Table 4: Local Data Definition for status

|  |  |
| --- | --- |
| Name | status |
| Description | Get the input of the assessment status from the user |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display and get input from the user

Attributes : id,date,assessment,status

Methods : None

### 4.2.12 trainerStudentList Controllers Class

The purpose of this class is to control which model to be use and check for data correctness.

#### 4.2.12.1 Class trainerProfile Design

This subparagraph specifies the design of trainerStudentList

1. **Input/Output data elements**

List of input and output data elements:

Input : $id

Output : None

1. **Local data elements**

Table 1: Local Data Definition for id array

|  |  |
| --- | --- |
| Name | id |
| Description | The id use as parameter |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), index(), view($id)

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view class to the model class

Input Parameter : None

Output Parameter : Render to the trainerStudentList index page

Algorithm :

BEGIN

$this->view->trainerStudentList = $this->model-trainerStudentList();

$this->view->render('trainerStudentList/index');

END

**3.view($id)**

Responsibility : To render from view class to the model class

Input Parameter : $id

Output Parameter : Render to the trainerStudentList view page

Algorithm :

BEGIN

$this->view->trainerStudentListView = $this->model->trainerStudentListView($id);

$this->view->render('trainerStudentList/view');

END

### 4.2.13 trainerStudentList\_model Model Class

The purpose of this class is to check the data handling according to specific rules. This class also responsible to retrieve the selected data from/to the database

#### 4.2.13.1 Class trainerStudentList\_model Design

This subparagraph specifies the design of trainerStudentList\_model

1. **Input/Output data elements**

List of input and output data elements:

Input : $id

Output : None

1. **Local data elements**

Table 1: Local Data Definition for id array

|  |  |
| --- | --- |
| Name | id |
| Description | The id use as parameter |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database

Attributes : None

Methods : construct(),trainerStudentList(),trainerStudentListView($id)

**1. construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

**2. trainerStudentList()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE trainerId = :id ', array('id' => $\_SESSION['id']));

END

**3. trainerStudentListView($id)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : $id

Output Parameter : Return data to view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE id = :id', array(':id' => $id));

END

### 4.2.14 index (trainerStduentList) View Class

The purpose of this class is to display and get input from the user

#### 4.2.14.1 Class index Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : $value

1. **Local data elements**

Table 1: Local Data Definition for value

|  |  |
| --- | --- |
| Name | value |
| Description | The attribute that are stored in the database in array form |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display output to the user

Attributes : id,name

Methods : None

### 4.2.15 view (trainerStudentList) View Class

The purpose of this class is to display the output to the user

#### 4.2.15.1 Class view Design

This subparagraph specifies the design of view

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : id, name, role, ic, password, gender, address, dob, religion, race, phoneNum, email

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Display the student id |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | Display the student name |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for role

|  |  |
| --- | --- |
| Name | role |
| Description | Display the student role |
| Data Type | varchar |
| Precision/resolution | - |

Table 4: Local Data Definition for ic

|  |  |
| --- | --- |
| Name | ic |
| Description | Display the student national identification number |
| Data Type | integer |
| Precision/resolution | - |

Table 5: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | Display the student password |
| Data Type | varchar |
| Precision/resolution | - |

Table 6: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | Display the student gender |
| Data Type | varchar |
| Precision/resolution | - |

Table 7: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | Display the student address |
| Data Type | varchar |
| Precision/resolution | - |

Table 8: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | Display the student date of born |
| Data Type | date |
| Precision/resolution | - |

Table 9: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | Display the student religion |
| Data Type | varchar |
| Precision/resolution | - |

Table 10: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | Display the student race |
| Data Type | varchar |
| Precision/resolution | - |

Table 11: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | Display the student phone number |
| Data Type | integer |
| Precision/resolution | - |

Table 12: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | Display the student email |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility : To display the output to the user

Attributes : id, name, role, ic, password, gender, address, dob, religion, race, phoneNum, email

Methods : None

## 4.3 Classes Management

This package consists of:

1. classReserve Controller Class
2. classReserve \_model Model Class
3. index (classReserve) View Class
4. staffClassApprove Controller Class
5. staffClassApprove\_model Model Class
6. index (staffClassApprove) View Class
7. edit (staffClassApprove) View Class

### 4.3.1 classReserve Controllers Class

The purpose of this class is to control which model to be use and check for data correctness.

#### 4.3.1.1 Class classReserve Design

This subparagraph specifies the design of classReserve

1. **Input/Output data elements**

List of input and output data elements:

Input : none

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct (), index (), create (),

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view to the model

Input Parameter : None

Output Parameter : Render to the classReserveindex page

Algorithm :

BEGIN

$this->view->classReserveList = $this->model->classReserveList();

$this->view->render('classReserve/index');

END

**3.create()**

Responsibility : To create the data in array format

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$this->model->create($data);

END

### 4.3.2 classReserve\_model Model Class

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store/retrieve/delete the selected data from/to the database

#### 4.3.2.1 Class classReserve\_model Design

This subparagraph specifies the design of classReserve\_model

1. **Input/Output data elements**

List of input and output data elements:

Input : $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database

Attributes : None

Methods : construct (),classReserveList (), create (),

**1. construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

**2. classReserveList ()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM classes WHERE TrainerId = :TrainerId', array('TrainerId' => $\_SESSION['id']));

END

**3.create($data)**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : none

Algorithm :

BEGIN

$this->db->insert('classes', array($data,

'status'=>"pending"));

END

### 4.3.3 index (classReserve) View Class

The purpose of this class is to display and get input from the user

#### 4.3.3.1 Class index Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : TrainerId, classType,date,timeIn,timeOut, purpose

Output : classType ,date,timeIn,timeOut, status

1. **Local data elements**

Table 1: Local Data Definition for TrainerId

|  |  |
| --- | --- |
| Name | TrainerId |
| Description | Get the user id from the logged session |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for classType

|  |  |
| --- | --- |
| Name | classType |
| Description | Get the class type from the user |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Get the input of date from the user |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for timeIn

|  |  |
| --- | --- |
| Name | timeIn |
| Description | Get the input of time in from the user |
| Data Type | time |
| Precision/resolution | - |

Table 5: Local Data Definition for timeOut

|  |  |
| --- | --- |
| Name | timeOut |
| Description | Get the input of time out from the user |
| Data Type | time |
| Precision/resolution | - |

Table 6: Local Data Definition for purpose

|  |  |
| --- | --- |
| Name | purpose |
| Description | Get the input of purpose from the user |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display and get input from the user

Attributes : None

Methods : None

### 4.3.4 staffClassApprove Controllers Class

The purpose of this class is to control which model to be use and check for data correctness.

#### 4.3.4.1 Class staffClassApprove Design

This subparagraph specifies the design of staffClassApprove

1. **Input/Output data elements**

List of input and output data elements:

Input : $id

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for id array

|  |  |
| --- | --- |
| Name | id |
| Description | The id use as parameter |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), index(), edit($id), editSave($id)

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view class to the model class

Input Parameter : None

Output Parameter : Render to the staffClassApprove index page

Algorithm :

BEGIN

$this->view->staffClassApproveList = $this->model->staffClassApproveList();

$this->view->render('staffClassApprove/index');

END

**3.edit($id)**

Responsibility : To render from view class to the model class

Input Parameter : $id

Output Parameter : Render to the staffClassApprove edit page

Algorithm :

BEGIN

$this->view->staffClassApproveEdit = $this->model->staffClassApproveEdit($id);

$this->view->render('staffClassApprove/edit');

END

**4.editSave($id)**

Responsibility : To control the data from view class to the model class

Input Parameter : $id

Output Parameter : $data

Algorithm :

BEGIN

$this->model->editSave($data);

END

### 4.3.5 staffClassApprove\_model Model Class

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store/retrieve/delete the selected data from/to the database

#### 4.3.5.1 Class staffClassApprove\_model Design

This subparagraph specifies the design of staffClassApprove\_model

1. **Input/Output data elements**

List of input and output data elements:

Input : $id, $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for id array

|  |  |
| --- | --- |
| Name | id |
| Description | The id use as parameter |
| Data Type | array |
| Precision/resolution | - |

Table 2: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database

Attributes : None

Methods : construct(), staffClassApproveList(),staffClassApproveEdit($id), editSave($data)

**1. construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

**2. staffClassApproveList()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM classes');

END

**3. staffClassApproveEdit ($id)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : $id

Output Parameter : Return data to view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM classes WHERE id = :id', array(':id' => $id));

END

**4.editSave($data)**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('UPDATE classes $data),

$sth->execute(array)

END

### 4.3.6 index (staffClassApprove) View Class

The purpose of this class is to display and get input from the user

#### 4.3.6.1 Class index Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : $value

1. **Local data elements**

Table 1: Local Data Definition for value

|  |  |
| --- | --- |
| Name | value |
| Description | The attribute that are stored in the database in array form |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display output to the user

Attributes : None

Methods : None

### 4.3.7 edit (staffClassApprove) View Class

The purpose of this class is to display and get input from the user

#### 4.3.7.1 Class edit Design

This subparagraph specifies the design of edit

1. **Input/Output data elements**

List of input and output data elements:

Input : trainerId, classType, date, timeIn, timeOut, purpose, status

Output : None

1. **Local data elements**

Table 1: Local Data Definition for trainerId

|  |  |
| --- | --- |
| Name | trainerId |
| Description | Get the user Id |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for classType

|  |  |
| --- | --- |
| Name | classType |
| Description | Get the type of class from user |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Get the date from user |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for timeIn

|  |  |
| --- | --- |
| Name | timeIn |
| Description | Get the time in from user |
| Data Type | date |
| Precision/resolution | - |

Table 5: Local Data Definition for timeOut

|  |  |
| --- | --- |
| Name | timeOut |
| Description | Get the time out from user |
| Data Type | date |
| Precision/resolution | - |

Table 6: Local Data Definition for purpose

|  |  |
| --- | --- |
| Name | purpose |
| Description | Get the purpose from user |
| Data Type | varchar |
| Precision/resolution | - |

Table 7: Local Data Definition for status

|  |  |
| --- | --- |
| Name | status |
| Description | Get the status user |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To get input from the user

Attributes : None

Methods : None

## 4.4 Staff Management

This package consists of:

1. staffProfile\_Model class
2. staffProfile Controller class
3. index (staffProfile) View class
4. edit (staffProfile) View class
5. picture (staffProfile) View class
6. staffAttendace\_Model class Model
7. staffAttendace class Controller
8. index (staffAttendace) View class
9. userProfile\_Model class Model
10. userProfile class Controller
11. index (userProfile) View class
12. AddUser (userProfile) View class
13. studentEdit (userProfile) View class
14. studentView (userProfile) View class
15. trainerEdit (userProfile) View class
16. trainerView (userProfile) View class

### 4.4.1 staffAttendance

The purpose of this class is to check and send the data to model class

#### 4.4.1.1 Class Controller staffAttendance design

This subparagraph specifies the design of staffProfile

1. **Input/Output data elements**

List of input and output data elements:

Input : none

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), index(),

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view to the model

Input Parameter : None

Output Parameter : Render to the trainerAttendance and studentAttendance index page

Algorithm :

BEGIN

$this->view->AttendanceList = $this->model->AttendanceList();

$this->view->AttendanceList1 = $this->model->AttendanceList1();

$this->view->render('staffAttendance/index');

END

### 4.4.2 staffAttendance\_model

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store/retrieve/delete the selected data from/to the database

#### 4.4.2.1 Class staffAttendance\_model Design

This subparagraph specifies the design of staffAttendance\_model

1. **Input/Output data elements**

List of input and output data elements:

Input : $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database

Attributes : None

Methods : construct(),AttendanceList(),AttendanceList1()

**1. construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

**2. AttendanceList ()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM studentattendance');

END

**3.** **AttendanceList1 ()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM trainerattendance END

### 4.4.3 index (staffAttendance)

The purpose of this class is to display and get input from the user

#### 4.4.3.1 Class index Design

1. **Input/Output data elements**

List of input and output data elements:

Input : id,name,date,timeIn,timeOut

Output : date,timeIn,timeOut

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Get the user id from the logged session |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | Get the user name from the logged session |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Get the input of date from the user |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for timeIn

|  |  |
| --- | --- |
| Name | timeIn |
| Description | Get the input of time in from the user |
| Data Type | time |
| Precision/resolution | - |

Table 5: Local Data Definition for timeOut

|  |  |
| --- | --- |
| Name | timeOut |
| Description | Get the input of time out from the user |
| Data Type | time |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display and get input from the user

Attributes : None

Methods : None

### 4.4.4 staffProfile

The purpose of this class is to check and send the data to model class

#### 4.4.4.1 Class Controller staffProfile design

This subparagraph specifies the design of staffProfile

1. **Input/Output data elements**

List of input and output data elements:

Input : data, id, password, address, gender, religion, phoneNum, email, imgData

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data

|  |  |
| --- | --- |
| Name | data |
| Description | To pass data using array |
| Data Type | Array |
| Precision/resolution | - |

Table 2: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | staff log in id |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | staff address |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | staff Gender |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for imgData

|  |  |
| --- | --- |
| Name | imgData |
| Description | Where picture has been upload |
| Data Type | Image |
| Precision/resolution | - |

Table 6: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | staff can change their password |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for phonenum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | staff phone number |
| Data Type | Integer |
| Precision/resolution | - |

Table 8: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | staff religion |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | staff email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : Controller class

Responsibility : To send data into model class

Attributes :staffName, staffIC, staffAddress, staffPicture, staffDOB, staffTrainer

Methods : \_\_construct(), index(), edit(id), picture(id), editsave(id), editpicture(id)

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view class to the model class

Input Parameter : None

Output Parameter : Render to the trainerProfile index page

Algorithm :

BEGIN

$this->view->staffProfileEdit = $this->model->staffProfileEdit();

$this->view->render('staffProfile/index');

END

**3.edit($id)**

Responsibility : To render from view class to the model class

Input Parameter : $id

Output Parameter : Render to the trainerProfile edit page

Algorithm :

BEGIN

$this->view->staffProfileEdited = $this->model->staffProfileEdited($id);

$this->view->render('staffProfile/edit');

END

**4.picture($id)**

Responsibility : To render from view class to the model class

Input Parameter : $id

Output Parameter : Render to the trainerProfile picture

Algorithm :

BEGIN

$this->view->staffProfileEdited = $this->model->staffProfileEdited($id);

$this->view->render('staffProfile/picture');

END

**5.editSave($id)**

Responsibility : To control the data from view class to the model class

Input Parameter : $id

Output Parameter : $data

Algorithm :

BEGIN

$this->model->editSave($data);

END

**6.editPicture($id)**

Responsibility : To control the data from view class to the model class

Input Parameter : $id

Output Parameter : $data

Algorithm :

BEGIN

$this->model->editPicture($data);

END

### 4.4.5 staffProfile\_model

The purpose of this class is to create object from another classes into it. It acts as a platform to let controller class to update and supply data for view class.

#### 4.4.5.1 Class Model staffProfile\_Model design

This subparagraph specifies the design of staffProfile\_Model

1. **Input/Output data elements**

List of input and output data elements:

Input : data, id, password, address, gender, religion, phonenum, email, picture

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data

|  |  |
| --- | --- |
| Name | data |
| Description | To pass data using array |
| Data Type | Array |
| Precision/resolution | - |

Table 2: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | Student log in id |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | Student address |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | Student Gender |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for picture

|  |  |
| --- | --- |
| Name | picture |
| Description | Student profile picture |
| Data Type | Binary Large OBject |
| Precision/resolution | - |

Table 6: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | Student can change their password |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for phonenum

|  |  |
| --- | --- |
| Name | phonenum |
| Description | Student phone number |
| Data Type | Integer |
| Precision/resolution | - |

Table 8: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | Student religion |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | Student email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : Model class

Responsibility : To insert and update data for picture and profile student

Attributes : data, id, password, address, gender, religion, phonenum, email, picture

Methods: \_\_construct(),studentProfileEdit(),

studentProfileEdited(id),editSave(data),editPicture(data)

**1. studentProfileEdit()**

Responsibility : to view data for student profile

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

SELECT \* from full WHERE id = id

END

**2.studentProfileEdit(id)**

Responsibility : to display certain student profile

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

SELECT \* from full WHERE id = id

END

**3.editsave (data)**

Responsibility : to update data

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

UPDATE full

SET `password` = :password,`address` = :address,`gender` = :gender,`religion` = :religion, `phoneNum` = :phoneNum, `email` = :email

WHERE id = :id

END

**4.editPicture (data)**

Responsibility : to display certain student profile

Input Parameter : None

Output Parameter : None

Algorithm:

BEGIN

SELECT \* from full WHERE id = id

END

**5.\_\_construct ()**

Responsibility : act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm: None

### 4.4.6 edit (staffProfile)

The purpose of this class is to view and update the data.

#### 4.4.6.1 Class view edit design (staffProfile)

This subparagraph specifies the design of edit

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : password, address, gender, religion, phonenum, email

1. **Local data elements**

Table 1: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | Staff password |
| Data Type | string |
| Precision/resolution | none |

Table 2: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | Student gender |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | Student address |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | Religion |
| Description | Student religion |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for phonenum

|  |  |
| --- | --- |
| Name | phonenum |
| Description | Student phone number |
| Data Type | Integer |
| Precision/resolution | - |

Table 6: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | Student email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data of student profile

Attributes : password, address, gender, religion, phonenum, email

Methods **:** None

### 4.4.7 index (staffProfile)

The purpose of this class is to view the data from database

#### 4.4.7.1 Class view index design (staffProfile)

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : picture, id, name, ic, role, gender, address, dob, religion, race, phonenum, email

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | User id |
| Data Type | String |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | user name |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | user password |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for role

|  |  |
| --- | --- |
| Name | role |
| Description | user role |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for ic

|  |  |
| --- | --- |
| Name | ic |
| Description | User ic |
| Data Type | Integer |
| Precision/resolution | - |

Table 6: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | user gender |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | user address |
| Data Type | String |
| Precision/resolution | - |

Table 8: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | user dob |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | user religion |
| Data Type | String |
| Precision/resolution | - |

Table 10: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | user race |
| Data Type | String |
| Precision/resolution | - |

Table 11: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | user phoneNum |
| Data Type | Integer |
| Precision/resolution | - |

Table 12: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | user email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data and picture of student

Attributes : picture, id, name, ic, role, gender, address, dob, religion, race, phonenum, email

Methods **:** None

### 4.4.8 picture (staffProfile)

The purpose of this class is to insert or update profile picture

#### 4.4.8.1 Class view picture design (staffProfile)

This subparagraph specifies the design of picture

1. **Input/Output data elements**

List of input and output data elements:

Input : file

Output : None

1. **Local data elements**

Table 1: Local Data Definition for file

|  |  |
| --- | --- |
| Name | file |
| Description | Image to upload |
| Data Type | Jpeg |
| Precision/resolution | 65535 bytes |

1. **Algorithms**

Class Type : View Class

Responsibility : To insert or update new profile picture

Attributes : picture

Methods **:** post

### 4.4.9 userProfile

The purpose of this class is to check and send the data to model class

#### 4.4.9.1 Class userProfile \_model Design

This subparagraph specifies the design of userProfile

1. **Input/Output data elements**

List of input and output data elements:

Input : $id

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

Table 2: Local Data Definition for id array

|  |  |
| --- | --- |
| Name | data |
| Description | The id use as parameter |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), AddUser($id), index(), studentEdit($id), studentview($id), trainerEdit($id), trainerView($id), create(), editsave($id),

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view to the model

Input Parameter : None

Output Parameter : Render to the userProfile index page

Algorithm :

BEGIN

$this->view->userProfileEdit = $this->model->userProfileEdit();

$this->view->render('userProfile/index');

END

**3. AddUser($id)**

Responsibility : To render from view to the model

Input Parameter : $id

Output Parameter : Render to the userProfile addUser page

Algorithm :

BEGIN

$this->view->userProfileEdited = $this->model->userProfileEdited($id);

$this->view->render('userProfile/AddUser');

END

**4.** **studentEdit($id)**

Responsibility : To render from view to the model

Input Parameter : $id

Output Parameter : Render to the userProfile studentEdit page

Algorithm :

BEGIN

$this->view->userProfileEdited = $this->model->userProfileEdited($id);

$this->view->render('userProfile/studentEdit');

END

**5.** **studentview($id)**

Responsibility : To render from view to the model

Input Parameter : $id

Output Parameter : Render to the userProfile studentView page

Algorithm :

BEGIN

$this->view->userProfileEdited = $this->model->userProfileEdited($id);

$this->view->AttendanceList = $this->model->AttendanceList();

$this->view->render('userProfile/studentView');

END

**6.** **trainerEdit($id)**

Responsibility : To render from view to the model

Input Parameter : $id

Output Parameter : Render to the userProfile trainerEdit page

Algorithm :

BEGIN

$this->view->userProfileEdited = $this->model->userProfileEdited($id);

$this->view->render('userProfile/trainerEdit');

END

**7.** **trainerView($id)**

Responsibility : To render from view to the model

Input Parameter : $id

Output Parameter : Render to the userProfile trainerView page

Algorithm :

BEGIN

$this->view->userProfileEdited = $this->model->userProfileEdited($id);

$this->view->AttendanceList1 = $this->model->AttendanceList1();

$this->view->render('userProfile/trainerView');

END

**8.** **create()**

Responsibility : To create the data in array format

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$this->model->create($data);

END

**9.editSave($data)**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$this->model->editSave($data);

END

### 4.4.10 userProfile\_model

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store/retrieve/delete the selected data from/to the database

#### 4.4.10.1 Class userProfile\_model Design

This subparagraph specifies the design of userProfile\_model

1. **Input/Output data elements**

List of input and output data elements:

Input : $id, $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for id array

|  |  |
| --- | --- |
| Name | id |
| Description | The id use as parameter |
| Data Type | array |
| Precision/resolution | - |

Table 2: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database

Attributes : None

Methods : construct(),userProfileEdit(),userProfileEdited($id), create($data), editSave($data), AttendanceList(),AttendanceList1()

**1. construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

**2. userProfileEdit ()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE id = :id', array('id' => $\_SESSION['id']));

END

**3. userProfileEdited ($id)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : $id

Output Parameter : Return data to view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE id = :id', array(':id' => $id));

END

**4.** **create ($data)**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('INSERT INTO full

(`id`,`name`,`password`,`role`,`ic`,`gender`,`address`,`dob`,`religion`,`race`,`phoneNum`,`email`)

VALUES (:id, :name, :password,:role,:ic,:gender,:address, :dob, :religion, :race, :phoneNum,:email)')

END

**5.** **editSave ($data)**

Responsibility : To store the data in array format

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare

('UPDATE full SET `name` = :name,`password` = :password,`role` = :role,`ic` = :ic,`gender` = :gender,`address` = :address,`dob` = :dob,`religion` = :religion,`race` = :race,

`phoneNum` = :phoneNum, `email` = :email,`trainerId` = :trainerId

WHERE id = :id' );

END

**6.** **AttendanceList ($data)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE role = "student" ',

array('role' => $\_SESSION['role']));

END

**7.** **AttendanceList1 ($data)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM full WHERE role = "Trainer" ',

array('role' => $\_SESSION['role']));

END

### 4.4.11 AddUser (userProfile)

The purpose of this class is to display and get input from the user

#### 4.4.11.1 Class AddUser Design

This subparagraph specifies the design of edit

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : id, name, password, role, ic, gender, address, dob, religion, race, phoneNum, email

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | User id |
| Data Type | String |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | user name |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | user password |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for role

|  |  |
| --- | --- |
| Name | role |
| Description | user role |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for ic

|  |  |
| --- | --- |
| Name | ic |
| Description | User ic |
| Data Type | Integer |
| Precision/resolution | - |

Table 6: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | user gender |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | user address |
| Data Type | String |
| Precision/resolution | - |

Table 8: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | user dob |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | user religion |
| Data Type | String |
| Precision/resolution | - |

Table 10: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | user race |
| Data Type | String |
| Precision/resolution | - |

Table 11: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | user phoneNum |
| Data Type | Integer |
| Precision/resolution | - |

Table 12: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | user email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data of Add User

Attributes : id, name, password, role, ic, gender, address, dob, religion, race, phoneNum, email

Methods **:** None

### 4.4.12 index (userProfile)

The purpose of this class is to display and get input from the user

#### 4.4.12.1 Class index Design

This subparagraph specifies the design of edit

1. **Input/Output data elements**

List of input and output data elements:

Input : id

Output : None

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | User id |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data of Add User, trainer View, Student View

Attributes : id

Methods **:** None

### 4.4.13 studentEdit (userProfile)

The purpose of this class is to manage student

#### 4.4.13.1 Class studentEdit Design

This subparagraph specifies the design of edit

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : id, name, password, role, ic, gender, address, dob, religion, race, phoneNum, email, trainerid

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | User id |
| Data Type | String |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | user name |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | user password |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for role

|  |  |
| --- | --- |
| Name | role |
| Description | user role |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for ic

|  |  |
| --- | --- |
| Name | ic |
| Description | User ic |
| Data Type | Integer |
| Precision/resolution | - |

Table 6: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | user gender |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | user address |
| Data Type | String |
| Precision/resolution | - |

Table 8: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | user dob |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | user religion |
| Data Type | String |
| Precision/resolution | - |

Table 10: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | user race |
| Data Type | String |
| Precision/resolution | - |

Table 11: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | user phoneNum |
| Data Type | Integer |
| Precision/resolution | - |

Table 12: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | user email |
| Data Type | String |
| Precision/resolution | - |

Table 13: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | user email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data of student edit

Attributes : id, name, password, role, ic, gender, address, dob, religion, race, phoneNum, email, trainer id

Methods **:** None

### 4.4.14 studentView (userProfile)

The purpose of this class is to view student

#### 4.4.14.1 Class studentView Design

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : picture, id, name, ic, role, gender, address, dob, religion, race, phonenum, email

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | User id |
| Data Type | String |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | user name |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | user password |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for role

|  |  |
| --- | --- |
| Name | role |
| Description | user role |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for ic

|  |  |
| --- | --- |
| Name | ic |
| Description | User ic |
| Data Type | Integer |
| Precision/resolution | - |

Table 6: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | user gender |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | user address |
| Data Type | String |
| Precision/resolution | - |

Table 8: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | user dob |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | user religion |
| Data Type | String |
| Precision/resolution | - |

Table 10: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | user race |
| Data Type | String |
| Precision/resolution | - |

Table 11: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | user phoneNum |
| Data Type | Integer |
| Precision/resolution | - |

Table 12: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | user email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data and picture of student

Attributes : picture, id, name, ic, role, gender, address, dob, religion, race, phonenum, email

Methods **:** None

### 4.4.15 trainerEdit (userProfile)

The purpose of this class is to manage trainer

#### 4.4.15.1 Class trainerEdit Design

This subparagraph specifies the design of edit

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : id, name, password, role, ic, gender, address, dob, religion, race, phoneNum, email, trainerid

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | trainer id |
| Data Type | String |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | trainer name |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | trainer password |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for role

|  |  |
| --- | --- |
| Name | role |
| Description | trainer role |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for ic

|  |  |
| --- | --- |
| Name | ic |
| Description | trainer ic |
| Data Type | Integer |
| Precision/resolution | - |

Table 6: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | trainer gender |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | trainer address |
| Data Type | String |
| Precision/resolution | - |

Table 8: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | trainer dob |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | trainer religion |
| Data Type | String |
| Precision/resolution | - |

Table 10: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | trainer race |
| Data Type | String |
| Precision/resolution | - |

Table 11: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | trainer phoneNum |
| Data Type | Integer |
| Precision/resolution | - |

Table 12: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | trainer email |
| Data Type | String |
| Precision/resolution | - |

Table 13: Local Data Definition for trainerid

|  |  |
| --- | --- |
| Name | trainerid |
| Description | trainer trainerid |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data of student edit

Attributes : id, name, password, role, ic, gender, address, dob, religion, race, phoneNum, email, trainer id

Methods **:** None

### 4.4.16 trainerView (userProfile)

The purpose of this class is to manage student

#### 4.4.16.1 Class userManagementProfile \_model Design

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : picture, id, name, ic, role, gender, address, dob, religion, race, phonenum, email

1. **Local data elements**

Table 1: Local Data Definition for id

|  |  |
| --- | --- |
| Name | id |
| Description | User id |
| Data Type | String |
| Precision/resolution | - |

Table 2: Local Data Definition for name

|  |  |
| --- | --- |
| Name | name |
| Description | user name |
| Data Type | String |
| Precision/resolution | - |

Table 3: Local Data Definition for password

|  |  |
| --- | --- |
| Name | password |
| Description | user password |
| Data Type | String |
| Precision/resolution | - |

Table 4: Local Data Definition for role

|  |  |
| --- | --- |
| Name | role |
| Description | user role |
| Data Type | String |
| Precision/resolution | - |

Table 5: Local Data Definition for ic

|  |  |
| --- | --- |
| Name | ic |
| Description | User ic |
| Data Type | Integer |
| Precision/resolution | - |

Table 6: Local Data Definition for gender

|  |  |
| --- | --- |
| Name | gender |
| Description | user gender |
| Data Type | String |
| Precision/resolution | - |

Table 7: Local Data Definition for address

|  |  |
| --- | --- |
| Name | address |
| Description | user address |
| Data Type | String |
| Precision/resolution | - |

Table 8: Local Data Definition for dob

|  |  |
| --- | --- |
| Name | dob |
| Description | user dob |
| Data Type | String |
| Precision/resolution | - |

Table 9: Local Data Definition for religion

|  |  |
| --- | --- |
| Name | religion |
| Description | user religion |
| Data Type | String |
| Precision/resolution | - |

Table 10: Local Data Definition for race

|  |  |
| --- | --- |
| Name | race |
| Description | user race |
| Data Type | String |
| Precision/resolution | - |

Table 11: Local Data Definition for phoneNum

|  |  |
| --- | --- |
| Name | phoneNum |
| Description | user phoneNum |
| Data Type | Integer |
| Precision/resolution | - |

Table 12: Local Data Definition for email

|  |  |
| --- | --- |
| Name | email |
| Description | user email |
| Data Type | String |
| Precision/resolution | - |

1. **Algorithms**

Class Type : View Class

Responsibility : To view data and picture of student

Attributes : picture, id, name, ic, role, gender, address, dob, religion, race, phonenum, email

Methods **:** None

## 4.5 Vehicle Management

This package consists of:

1. vehicleReverse Controller Class
2. vehicleReserve\_model Model Class
3. index (vehicleReserve) View Class
4. view (vehicleReserve) View Class
5. book (vehicleReserve) View Class
6. vehicleReserveStaff Controller Class
7. vehicleReserveStaff\_model Model Class
8. index (vehicleReserveStaff) View Class
9. addnew (vehicleReserveStaff) View Class
10. view (vehicleReserveStaff) View Class
11. recordEdit (vehicleReserveStaff) View Class
12. edit (vehicleReserveStaff) View Class

### 4.5.1 vehicleReverse Controller Class

The purpose of this class is to control which model to be use and check for data correctness, only proceed to the next step if data is correct.

#### 4.5.1.1 Class vehicleReverse Design

This subparagraph specifies the design of vehicleReserve

1. **Input/output data elements**

List of input and output data elements:

Input : $carPlate, $data

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | The carPlate is use as parameter |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), index(), viewVehicleDetail($carPlate), vehicleBooking($carPlate), create(), vehicleBookingFunction()

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view to the model

Input Parameter : None

Output Parameter : Render to the staffReserve index page

Algorithm :

BEGIN

$this->view->vehicleList = $this->model->vehicleList();

$this->view->vehicleListStaff = $this->model->vehicleListStaff();

$this->view->render('vehicleReserve/index');

END

**3.** **viewVehicleDetail ($carPlate)**

Responsibility : To render from view class to model class.

Input Parameter : $carPlate

Output Parameter : Render to the vehicleReserve view page.

Algorithm :

BEGIN

$this->view->vehicleListView = $this->model->vehicleListView($carPlate);

$this->view->render('vehicleReserve/view');

END

**4.** **vehicleBooking ($carPlate)**

Responsibility : To render from view class to model class.

Input Parameter : $carPlate

Output Parameter : Render to the vehicleReserve book page.

Algorithm :

BEGIN

$this->view->vehicleBookingView = $this->model->vehicleBookingVehicleView($carPlate)

$this->view->render('vehicleReserve/book');

END

**5.create()**

Responsibility : To create the data in array format

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$this->model->create($data);

END

**6.** **vehicleBookingFunction ()**

Responsibility : To control the data from view class to the model class

Input Parameter : None

Output Parameter : $data

Algorithm :

BEGIN

$this->model->vehicleBookingFunction($data);

END

### 4.5.2 vehicleReverse\_model Model Class

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store and display the data from/to the database.

#### 4.5.2.1 Class vehicleReserve\_model Model Design

This subparagraph specifies the design of vehicleReserve\_model.

1. **Input/output data elements**

List of input and output data elements:

Input : $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database.

Attributes : None

Methods : construct(), vehicleList(), vehicleListView($carPlate), vehicleBookingView($carPlate), vehicleListStaff(), vehicleBookingFunction($data)

1. **construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

1. **vehicleList ()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM vehicle');

END

1. **vehicleListView ($carPlate)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : $carPlate

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM vehicle WHERE carPlate = :carPlate', array('carPlate' => $carPlate));

END

1. **vehicleBookingView ($carPlate)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : $carPlate

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM vehicle WHERE carPlate = :carPlate', array('carPlate' => $carPlate));

END

1. **vehicleListStaff ()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM vehiclebooking');

END

1. **vehicleBookingFunction ($data)**

Responsibility : To store the vehicle booking data in the database

Input Parameter : $data

Output Parameter : None

Algorithm :

BEGIN

$this->db->insert('vehiclebooking', array(

'carPlate' => $data['carPlate'],

'brand'=> $data['brand'],

'date' => $data['date'],

'timeIn' => $data['timeIn'],

'timeOut' => $data['timeOut'],

'status'=> $data['status'],

'trainerId' => $data['trainerId']

));

END

### 4.5.3 index (vehicleReverse) View Class

The purpose of this class is to display and get input from the user.

##### 4.5.3.1 Class index Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : carPlate, brand, date, timeIn, timeOut, trainerId, status.

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | Get the vehicle plate number from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for brand

|  |  |
| --- | --- |
| Name | brand |
| Description | Get the brand of vehicle from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Get the date of booking from the user in the database |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for timeIn

|  |  |
| --- | --- |
| Name | timeIn |
| Description | Get the input of time in from the user in the database |
| Data Type | time |
| Precision/resolution | - |

Table 5: Local Data Definition for timeOut

|  |  |
| --- | --- |
| Name | timeOut |
| Description | Get the input of time out from the user in the database |
| Data Type | time |
| Precision/resolution | - |

Table 6: Local Data Definition for trainerId

|  |  |
| --- | --- |
| Name | trainerId |
| Description | Get the trainer’s id from the logged session |
| Data Type | varchar |
| Precision/resolution | - |

Table 7: Local Data Definition for status

|  |  |
| --- | --- |
| Name | status |
| Description | Get the booking status from the staff in the database |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display output to the user.

Attributes : carPlate, brand, date, timeIn, timeOut, trainerId, status.

Methods : None

### 4.5.4 view (vehicleReverse) View Class

The purpose of this class is to display and get input from the user.

##### 4.5.4.1 Class view Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : carPlate, brand, lastServices, roadTax, typeOfCar.

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | Display vehicle plate number from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for brand

|  |  |
| --- | --- |
| Name | brand |
| Description | Display the brand of vehicle from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for lastServices

|  |  |
| --- | --- |
| Name | lastServices |
| Description | Display the date of last services from database |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for roadTax

|  |  |
| --- | --- |
| Name | roadTax |
| Description | Display the road tax of the vehicle from the database |
| Data Type | date |
| Precision/resolution | - |

Table 5: Local Data Definition for typeOfCar

|  |  |
| --- | --- |
| Name | typeOfCar |
| Description | Display type of vehicle from the database. |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display output to the user.

Attributes : carPlate, brand, lastServices, roadTax, typeOfCar.

Methods : None

### 4.5.5 book (vehicleReverse) View Class

The purpose of this class is to let user to book the vehicle.

##### 4.5.5.1 Class view Design

This subparagraph specifies the design of book

1. **Input/Output data elements**

List of input and output data elements:

Input : date, timeIn, timeOut

Output : carPlate, brand, date, timeIn, timeOut.

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | Display vehicle plate number from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for brand

|  |  |
| --- | --- |
| Name | brand |
| Description | Display the brand of vehicle from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Get the date of vehicle needed from the user |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for timeIn

|  |  |
| --- | --- |
| Name | timeIn |
| Description | Get the time start to use the vehicle from the user |
| Data Type | time |
| Precision/resolution | - |

Table 5: Local Data Definition for timeOut

|  |  |
| --- | --- |
| Name | timeOut |
| Description | Get the return time of vehicle from the user. |
| Data Type | time |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: Get the required data from the user and display it.

Attributes : carPlate, brand, date, timeIn, timeOut.

Methods : None

### 4.5.6 vehicleReverseStaff Controller Class

The purpose of this class is to control which model to be use and check for data correctness, only proceed to the next step if data is correct.

##### 4.5.6.1 Class vehicleReverseStaff Design

This subparagraph specifies the design of vehicleReserveStaff

1. **Input/output data elements**

List of input and output data elements:

Input : $carPlate, $data, $Id

Output : $data

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | The carPlate is use as parameter |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to be create and to be send to the model |
| Data Type | array |
| Precision/resolution | - |

Table 2: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | Id |
| Description | The Id is use as parameter |
| Data Type | Varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Controller class

Responsibility: Determine which model to be used and check for data correctness

Attributes : None

Methods : construct(), index(), viewVehicleRecord($carPlate), viewVehicleMaintenanceRecord($Id), editBookingRecord($Id), editVehicleMaintenanceRecord($Id), newVehicle (),create(),editSaveVehicleBookingRecord($Id), ditSaveVehicleMaintenanceRecord($Id), insertNewVehicle($Id),

**1. construct()**

Responsibility : To check the user login status and role

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$logged = Session::get('loggedIn'),

$role = Session::get('role'),

If false, destroy.

END

**2.index()**

Responsibility : To render from view to the model

Input Parameter : None

Output Parameter : Render to the vehicleReserveStaff index page

Algorithm :

BEGIN

$this->view->vehicleBookingDetail = $this->model->vehicleBookingDetail();

$this->view->vehicleListStaff = $this->model->vehicleListStaff();

$this->view->render('vehicleReserveStaff/index');

END

**3.** **viewVehicleRecord ($carPlate)**

Responsibility : To render from view to the model.

Input Parameter : $carPlate

Output Parameter : Render to the vehicleReserveStaff view page.

Algorithm :

BEGIN

$this->view->vehicleListViewStaff = $this->model->vehicleListViewStaff($carPlate);

$this->view->render('vehicleReserveStaff/view');

END

**4.** **viewVehicleMaintenanceRecord ($Id)**

Responsibility : To render from view class to model class.

Input Parameter : $Id

Output Parameter : Render to the vehicleReserveStaff recordEdit page.

Algorithm :

BEGIN

$this->view->vehicleListViewStaff = $this->model->vehicleListViewStaff($Id);

$this->view->render('vehicleReserveStaff/recordEdit');

END

**5.** **editBookingRecord ($Id)**

Responsibility : To render from view class to model class.

Input Parameter : $Id

Output Parameter : Render to the vehicleReserveStaff edit page.

Algorithm :

BEGIN

$this->view->vehicleBookingViewStaff = $this->model->vehicleBookingViewStaff($Id);

$this->view->render('vehicleReserveStaff/edit');

END

**6.** **editVehicleMaintenanceRecord ($Id)**

Responsibility : To render from view class to model class.

Input Parameter : $Id

Output Parameter : Render to the vehicleReserveStaff recordEdit page.

Algorithm :

BEGIN

$this->view->vehicleListViewStaff = $this->model->vehicleListViewStaff($Id);

$this->view->render('vehicleReserveStaff/recordEdit');

END

**7.** **newVehicle ()**

Responsibility : To render from view class to model class.

Input Parameter : None

Output Parameter : Render to the vehicleReserveStaff addnew page.

Algorithm :

BEGIN

$this->view->vehicleBookingDetail = $this->model->vehicleBookingDetail();

$this->view->render('vehicleReserveStaff/addnew');

END

**8.create()**

Responsibility : To create the data in array format

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

$this->model->create($data);

END

**9.** **editSaveVehicleBookingRecord($Id)**

Responsibility : To control the data from view class to the model class

Input Parameter : $Id

Output Parameter : $data

Algorithm :

BEGIN

$this->model->editSaveVehicleBookingRecord($data);

END

**10.** **editSaveVehicleMaintenanceRecord($Id)**

Responsibility : To control the data from view class to the model class

Input Parameter : $Id

Output Parameter : $data

Algorithm :

BEGIN

$this->model->editSaveVehicleMaintenanceRecord($data);

END

**11. insertNewVehicle ($Id)**

Responsibility : To control the data from view class to the model class

Input Parameter : $Id

Output Parameter : $data

Algorithm :

BEGIN

$this->model->insertNewVehicle($data);

END

### 4.5.7 vehicleReverseStaff\_model Model Class

The purpose of this class is to check the data handling according to specific rules. This class also responsible to store and display the data from/to the database.

##### 4.5.7.1 Class vehicleReserveStaff\_model Model Design

This subparagraph specifies the design of vehicleReserveStaff\_model.

1. **Input/output data elements**

List of input and output data elements:

Input : $data

Output : None

1. **Local data elements**

Table 1: Local Data Definition for data array

|  |  |
| --- | --- |
| Name | data |
| Description | The data to store in the database |
| Data Type | array |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : Model class

Responsibility: To store the data to the database.

Attributes : None

Methods : construct(), vehicleListStaff(), vehicleListViewStaff($carPlate), vehicleBookingViewStaff($Id),vehicleBookingDetail(),editSaveVehicleBookingRecord($data), editSaveVehicleMaintenanceRecord($data), insertNewVehicle($data),

1. **construct()**

Responsibility : Act as parent constructor

Input Parameter : None

Output Parameter : None

Algorithm :

BEGIN

parent::\_\_construct();

END

1. **vehicleListStaff** ()

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM vehiclebooking');

END

1. **vehicleListViewStaff ($carPlate)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : $carPlate

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM vehicle WHERE carPlate = :carPlate', array('carPlate' => $carPlate));

END

1. **vehicleBookingViewStaff ($Id)**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : $Id

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM vehiclebooking WHERE Id=:Id', array('Id'=>$Id));

END

1. **vehicleBookingDetail()**

Responsibility : To render the data from the selected table to be view by user

Input Parameter : None

Output Parameter : Return data to be view by user

Algorithm :

BEGIN

return $this->db->select('SELECT \* FROM vehicle');

END

1. **editSaveVehicleBookingRecord ($data)**

Responsibility : To store the vehicle edited booking data in the database

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('UPDATE vehiclebooking

SET `carPlate`=:carPlate,`brand`=:brand,`date`=:date, `timeIn`=:timeIn,`timeOut`=:timeOut,`trainerId`=:trainerId,`status`=:status WHERE `Id`=:Id');

END

1. **editSaveVehicleMaintenanceRecord ($data)**

Responsibility : To store the vehicle edited maintenance record in the database

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('UPDATE vehicle

SET `carPlate`=:carPlate,`brand`=:brand,`shopName`=:shopName, `typeOfService`=:typeOfService,`totalInvoice`=:totalInvoice WHERE `Id`=:Id');

END

1. **insertNewVehicle ($data)**

Responsibility : To store the record of new vehicle in the database

Input Parameter : $data

Output Parameter : $sth

Algorithm :

BEGIN

$sth = $this->db->prepare('INSERT vehicle

SET `carPlate`=:carPlate,`brand`=:brand,`lastServices`=:lastServices, `roadTax`=:roadTax,`shopName`=:shopName, `totalInvoice`=:totalInvoice, `typeOfService`=:typeOfService,`typeOfCar`=:typeOfCar');

END

### 4.5.8 index (vehicleReverseStaff) View Class

The purpose of this class is to display and get input from the user.

##### 4.5.8.1 Class index Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : carPlate, brand, date, timeIn, timeOut, trainerId, status.

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | Display the vehicle plate number from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for brand

|  |  |
| --- | --- |
| Name | brand |
| Description | Display the brand of vehicle from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Display the date of booking from the user in the database |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for timeIn

|  |  |
| --- | --- |
| Name | timeIn |
| Description | Display the input of time in from the user in the database |
| Data Type | time |
| Precision/resolution | - |

Table 5: Local Data Definition for timeOut

|  |  |
| --- | --- |
| Name | timeOut |
| Description | Display the input of time out from the user in the database |
| Data Type | time |
| Precision/resolution | - |

Table 6: Local Data Definition for trainerId

|  |  |
| --- | --- |
| Name | trainerId |
| Description | Display the trainer’s id from the logged session |
| Data Type | varchar |
| Precision/resolution | - |

Table 7: Local Data Definition for status

|  |  |
| --- | --- |
| Name | status |
| Description | Display the booking status from the staff in the database |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display output to the user.

Attributes : None

Methods : None

### 4.5.9 addnew (vehicleReverseStaff) View Class

The purpose of this class is to display and get input from the user.

##### 4.5.9.1 Class addnew Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : carPlate, brand, lastServices, roadTax, shopName, totalInvoice, typeOfCar, typeOfServices

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | Get the vehicle plate number from the user |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for brand

|  |  |
| --- | --- |
| Name | brand |
| Description | Get the brand of vehicle from the user |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for lastServices

|  |  |
| --- | --- |
| Name | lastServices |
| Description | Get the date of last services from the user |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for roadTax

|  |  |
| --- | --- |
| Name | roadTax |
| Description | Get the road tax of the vehicle from the user |
| Data Type | date |
| Precision/resolution | - |

Table 5: Local Data Definition for shopName

|  |  |
| --- | --- |
| Name | shopName |
| Description | Get the service place from the user. |
| Data Type | varchar |
| Precision/resolution | - |

Table 6: Local Data Definition for totalInvoice

|  |  |
| --- | --- |
| Name | totalInvoice |
| Description | Get the total invoice from the user. |
| Data Type | float |
| Precision/resolution | - |

Table 7: Local Data Definition for typeOfCar

|  |  |
| --- | --- |
| Name | typeOfCar |
| Description | Get the type of vehicle from the user. |
| Data Type | varchar |
| Precision/resolution | - |

Table 8: Local Data Definition for typeOfServices

|  |  |
| --- | --- |
| Name | typeOfServices |
| Description | Get the type of service of the vehicle from the user. |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: Get all the required input from the user.

Attributes :None

Methods : None

### 4.5.10 view (vehicleReverseStaff) View Class

The purpose of this class is to display and get input from the user.

##### 4.5.10.1 Class view Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : None

Output : carPlate, brand, lastServices, roadTax, shopName, totalInvoice, typeOfCar, typeOfServices

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | Display the vehicle plate number from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for brand

|  |  |
| --- | --- |
| Name | brand |
| Description | Display the brand of vehicle from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for lastServices

|  |  |
| --- | --- |
| Name | lastServices |
| Description | Display the date of last services from the database |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for roadTax

|  |  |
| --- | --- |
| Name | roadTax |
| Description | Display the road tax of the vehicle from the database |
| Data Type | date |
| Precision/resolution | - |

Table 5: Local Data Definition for shopName

|  |  |
| --- | --- |
| Name | shopName |
| Description | Display the service place from the database. |
| Data Type | varchar |
| Precision/resolution | - |

Table 6: Local Data Definition for totalInvoice

|  |  |
| --- | --- |
| Name | totalInvoice |
| Description | Display the total invoice from the database. |
| Data Type | float |
| Precision/resolution | - |

Table 7: Local Data Definition for typeOfCar

|  |  |
| --- | --- |
| Name | typeOfCar |
| Description | Display the type of vehicle from the database. |
| Data Type | varchar |
| Precision/resolution | - |

Table 8: Local Data Definition for typeOfServices

|  |  |
| --- | --- |
| Name | typeOfServices |
| Description | Display the type of service of the vehicle from the database. |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility : To display output to the user.

Attributes : None

Methods : None

### 4.5.11 recordEdit (vehicleReverseStaff) View Class

The purpose of this class is to display and get input from the user.

##### 4.5.11.1 Class recordEdit Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : shopName, typeOfServices, totalInvoice

Output : carPlate, brand, shopName, totalInvoice, typeOfServices

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | Display the vehicle plate number from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for brand

|  |  |
| --- | --- |
| Name | brand |
| Description | Display the brand of vehicle from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for shopName

|  |  |
| --- | --- |
| Name | shopName |
| Description | Get the service place from the user. |
| Data Type | varchar |
| Precision/resolution | - |

Table 4: Local Data Definition for totalInvoice

|  |  |
| --- | --- |
| Name | totalInvoice |
| Description | Get the total invoice from the user. |
| Data Type | float |
| Precision/resolution | - |

Table 5: Local Data Definition for typeOfServices

|  |  |
| --- | --- |
| Name | typeOfServices |
| Description | Get the type of service of the vehicle from the user. |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility : Get the required input from the user.

Attributes : None

Methods : None

### 4.5.12 edit (vehicleReverseStaff) View Class

The purpose of this class is to display and get input from the user.

##### 4.5.12.1 Class edit Design

This subparagraph specifies the design of index

1. **Input/Output data elements**

List of input and output data elements:

Input : date, timeIn, timeOut, trainerId, status.

Output : carPlate, brand

1. **Local data elements**

Table 1: Local Data Definition for carPlate

|  |  |
| --- | --- |
| Name | carPlate |
| Description | Display the vehicle plate number from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 2: Local Data Definition for brand

|  |  |
| --- | --- |
| Name | brand |
| Description | Display the brand of vehicle from the database |
| Data Type | varchar |
| Precision/resolution | - |

Table 3: Local Data Definition for date

|  |  |
| --- | --- |
| Name | date |
| Description | Get the date of booking from the user |
| Data Type | date |
| Precision/resolution | - |

Table 4: Local Data Definition for timeIn

|  |  |
| --- | --- |
| Name | timeIn |
| Description | Get the input of time in from the user |
| Data Type | time |
| Precision/resolution | - |

Table 5: Local Data Definition for timeOut

|  |  |
| --- | --- |
| Name | timeOut |
| Description | Get the input of time out from the user |
| Data Type | time |
| Precision/resolution | - |

Table 6: Local Data Definition for trainerId

|  |  |
| --- | --- |
| Name | trainerId |
| Description | Get the trainer’s id from the logged session |
| Data Type | varchar |
| Precision/resolution | - |

Table 7: Local Data Definition for status

|  |  |
| --- | --- |
| Name | status |
| Description | Get the booking status from the staff. |
| Data Type | varchar |
| Precision/resolution | - |

1. **Algorithms**

This section states the purpose and describes in detail the algorithms of this class

Class Type : View class

Responsibility: To display output to the user.

Attributes : None

Methods : None

# 5. NOTES

Abbreviation used:

* BTech Beta Technology Sdn Bhd
* DSMS Driving School Management System
* SDD Software Development Document
* SDP Software Development Plan
* SRS Software Requirement Specification

# 6. CD