CHAPTER 4

RESULTS AND DISCUSSION

4.1. System Design

System design is the process of defining the components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theoryto product development.

4.1.1. System Design Objective

1. The application of the web-based Fr. Peter Secondary School Management System can manage staffs and students personal information (profiling), subjects and teachers, boarding student accommodation, class advisory, creating class schedules or timetables, and manage other sections.
2. The web-based system can perform accounting and financial records processes more effectively especially the payment of student's school fee.
3. The new system could manage student enrollment and registration processes in a well-organized manner and more effectively.
4. The web-based school management system that can generate quickly a correct and an accurate student transcript, class timetable, and student school fee ledger so every student can have access to them anytime.

4.1.2. Proposed System Overview

The proposed system for Fr. Peter Secondary School Management System is particulary a web-based systemthat usesthe concept of client-server, in whichthe client uses a browser while PHP and MySql are used as the server. This is done online so other users or clients can access these applications through using the internet.

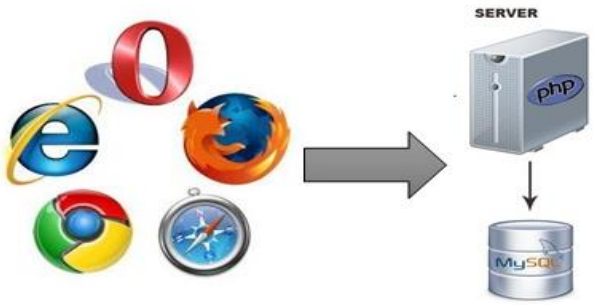


Figure 4.1 Client – Server Scheme

4.1.3. Proposed Design Procedure

This design includes the use case diagrams, activity diagrams, class diagrams, and deployment diagrams in order to build a good system. The designed process was broken down into several sections which can be formed into a single unified system component.

4.1.3.1. Use Case Diagram

Use case describes an interaction between one or more actors with information systems that will be made. The use case is used to determine what function is in an information system and who is entitled to use those functions. The following is the use case diagram of the proposed web-based school management system at Fr. Peter Secondary School.

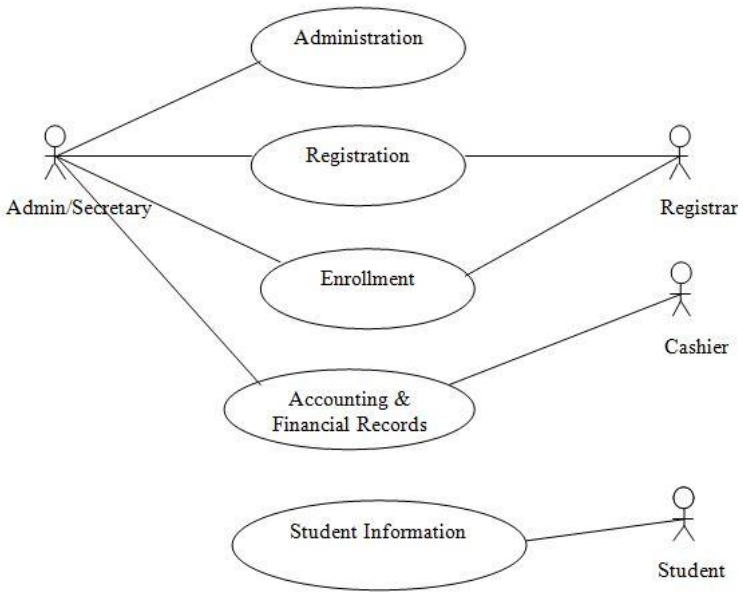


Figure 4.2 Use case diagram of Fr. Peter Secondary School Management System the Web-based

4.1.3.2. Scenario Use Case

Use case scenarios are used to make it easier to analyze scenarios that will be used in subsequent phases by assessing these scenarios, the following use case scenarios proposed.

1. Scenario Use Case Administration.

The use case scenario Administration in the proposed Fr. Peter Secondary School system is as follows:

1. Name of the use case : Administration
2. The main actors : Admin/ Secretary
3. Purpose : To manage administration activities like managing staffs' profiles, subjects and subject allocations, create class schedules, boarding students accommodations, create class advisory, and manage other sections such as registration, enrollment, finance, and the whole system.

Table 4.1 Scenario Administration Use Case

Admin/Secretary	System
	1. Displays the Login page
2. Fill in the User Name and Password and click Login	
	3. Displays the Menu page
4. Select "Administrative"	
	5. Displays the "Sub-

	Menus"
6. Select one of the sub-menus such as "Teacher Subjects"	
	7. Displays the empty form with options "Add" and Delete"
8. Click "Add" to add a new teacher with the subject	
	10. Displays an empty form
11. Fill all necessary information and click "Save"	
	12. Displays the word "Successful" with Green color background
AlternativeScenario	
	13. If one of the field is empty, the system only highlights those empty fields with a Red background color
9. Re-fill all the fields as required	

1. Scenario Use Case Registration

The use case scenario Registration in the proposed Fr. Peter Secondary School Management System is as follows:

1. Name of the use case : Registration
2. Main actor : Registrar/Admin
3. Purpose : Perform registration process of the school

Tabel 4.2 Scenario Registration Use Case

Registrar/Admin	System
	1. Display the Login page
2. Fill in the User Name and Password and click Login	
	3. Display menu page
4. Choose "Registration"	
	5. Shows sub-menus
6. Select "Students"	
	7. Displays the empty form with options "Add" and Delete"
8. Click "Add" to register a new student	
	9. Display the registration form
10. Fill the registration and click "Save"	
	11. Displays the word "Successful" with Green color background
AlternativeScenario	
	12. If one of the field is empty, the system only highlights those empty fields with a Red background color
10. Re-fill the form completely	

2. Scenario Use Case Enrollment

The use case scenario Enrollment in the proposed Fr. Peter Secondary Management System is as follows:

1. Name of the use case : Enrollment

2. Main actor : Registrar/Admin

3. Purpose : Perform enrollment process of the school

Tabel 4.3 Scenario Enrollment Use Case

Registrar/Admin	System
	1. Display Login page
2. Fill in the User Name and Password and click Login	
	3. Display menu page
4. Select "Enrollment" and click it	
	5. Displays the empty form with options "Add" and Delete"
6. Choose "Add" to enroll a new student	
	7. Display form
8. Fill in the form and click "Save"	
	9. Displays the word "Successful" with Green color background
AlternativeScenario	
	10. If one of the field is empty, the system only highlights those empty fields with a Red background color
11. Re-fill all the fields	

3. Scenario Use Case Accounting &Financial Records

The use case scenario Financial Records in the proposed Fr. Peter Secondary Management System is as follows:

1. Name of the use case : Accounting &Financial Records
2. Main actor : Cashier/Admin
3. Purpose : Perform all accounting and financial tasks and produce student school fee ledger.

Tabel 4.4 Scenario Accounts &Financial Records Use Case

Cashier/Admin	System
	1. Shows the Login page
2. Fill the User Name and Password and click Login	
	3. Display the Menu Page
4. Select Cashier	
	5. Displays Sub-menu lists
6. Choose “Payment”	
	7. Shows the payment form
8. Fill in the form and click “Save”	
	9. Displays the word “Successful” with Green color background
AlternativeScenario	
	10. If one of the field is empty, the system only highlights those empty fields with a Red background color
11. Re-fill the form	

4. Scenario Use Case Student Information

The use case scenario Students Informationin the proposed Fr. Peter Secondary School System is as follows:

1. Name of the use case : Student Information

2. Main actor : Student

3. Purpose : For student to see and print his/her transcript, class timetable, and school fee ledger

Tabel 4.5 Scenario Student Information Use Case

Student	System
	1. Displaythe Login page
2. Enter Student ID Number as "User Name" and fill in the Password and click "Login"	
	3. Display the Manu
4. If student want to print his/her Transcript clicks "Grade List"	
	7. Displays the transcript
8. Click "Print" to download the PDF file	
AlternativeScenario	
	9. If id number or password entered is wrong, then the message "Wrong User Name or Password" will be displayed when trying to Login
10. Fill in the right or correct user name and password to login	

Activity Diagram of the Proposed New System

Activity diagram below shows in detail the logical data flow without considering the physical environment in which the data flows. Here is the proposed activity on the application of Fr. Peter Secondary School Management System.

1. Activity Diagram Administration

This section describes the workflow of the proposed Administration system in the proposed new system of Fr. Peter Secondary School.

The Activity diagram is as follows:

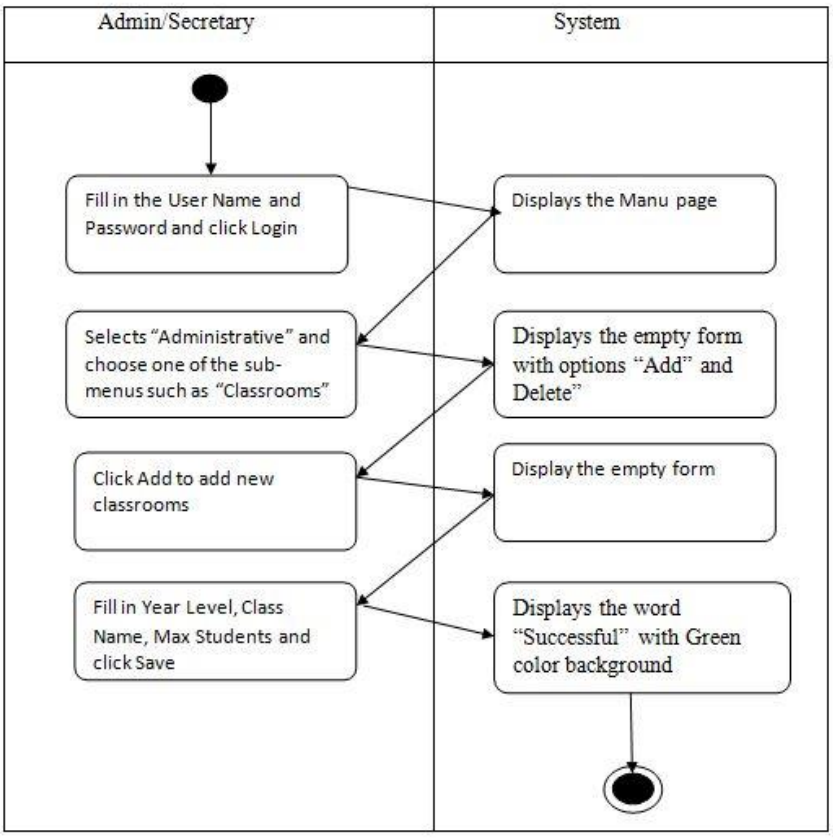


Figure 4.3 Activity Diagram Administration Proposed

2. Activity Diagram Registration

This section describes the workflow of the proposed Registration system in the proposed new system.

The Activity diagram is as follows:

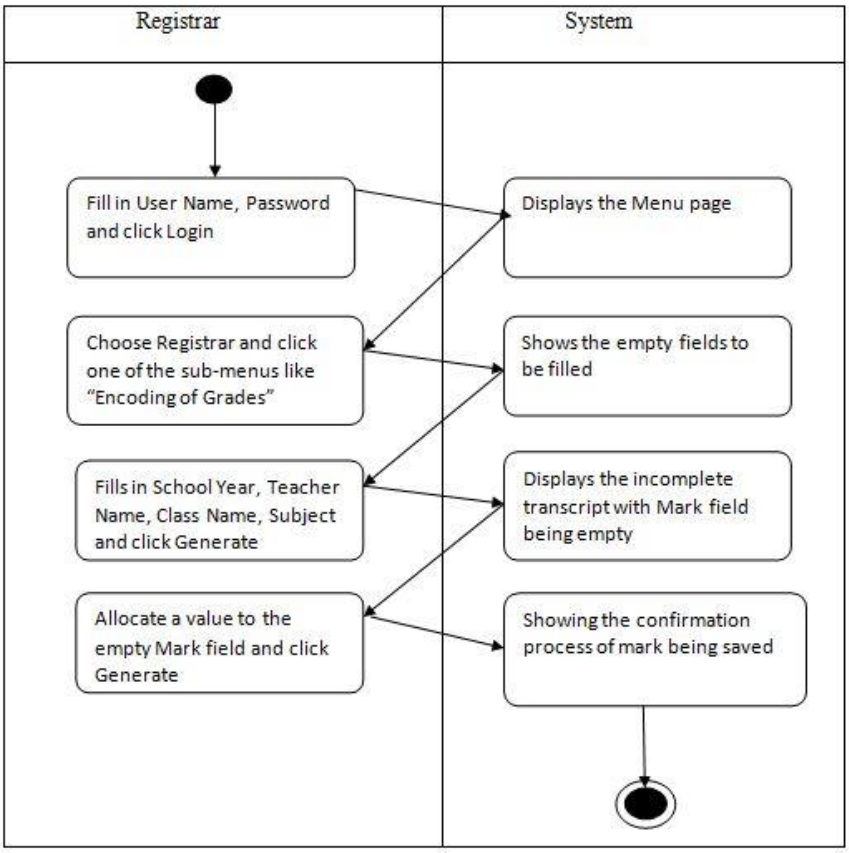


Figure 4.4 Activity Diagram Registration Proposed

3. Activity Diagram Enrollment

This section describes the workflow of the proposed Enrollment system in the proposed new system.

The Activity diagram is as follows:

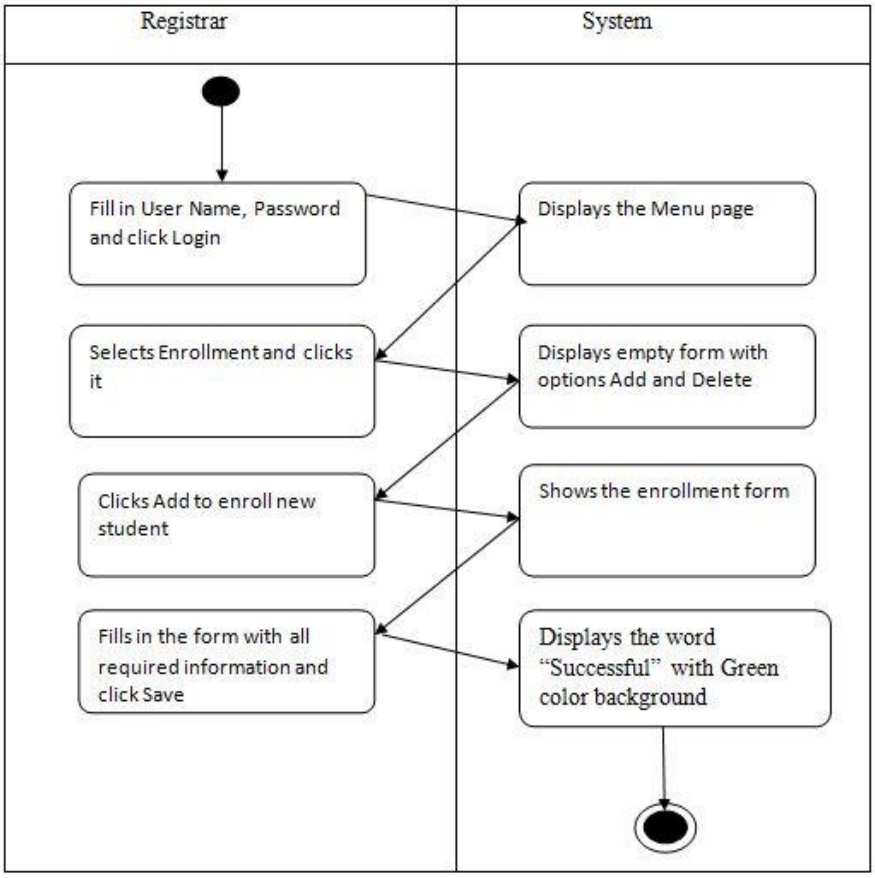


Figure 4.5 Activity Diagram Enrollment Proposed

4. Activity Diagram Accounting &Financial Records

This section describes the workflow of the proposed Financial Records system in the proposed new system of Fr. Peter Secondary School.

The Activity diagram is as follows:

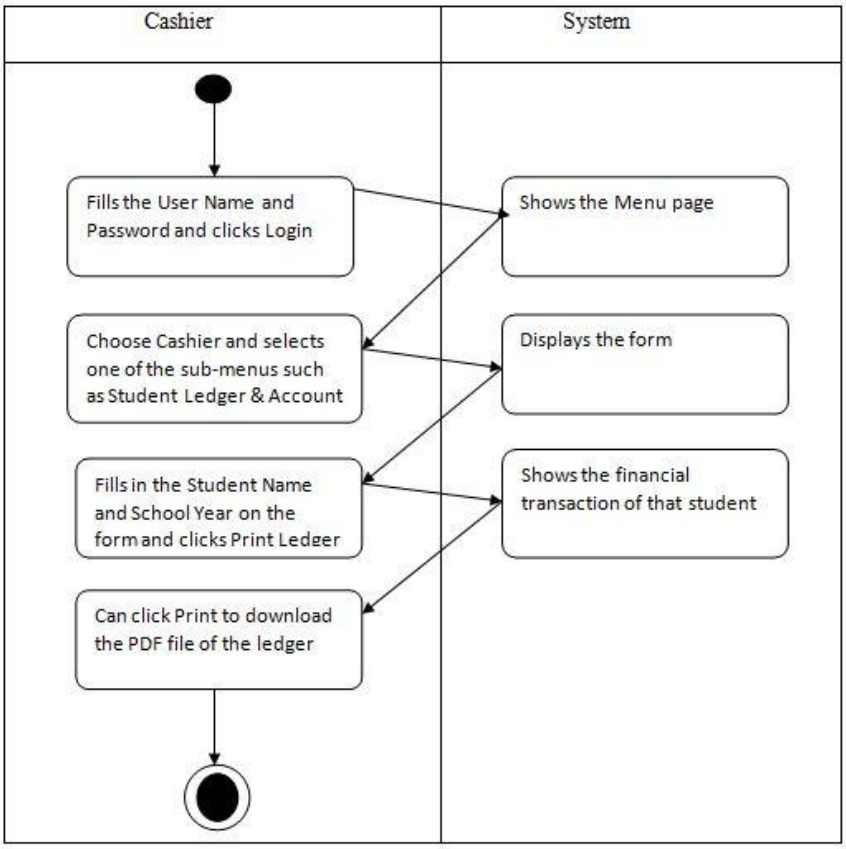


Figure 4.6 Activity Diagram Accounting &Financial Records Proposed

5. Activity Diagram Student Information

This section describes the workflow of the proposed Student Information in the proposed new system.

The Activity diagram is as follows:

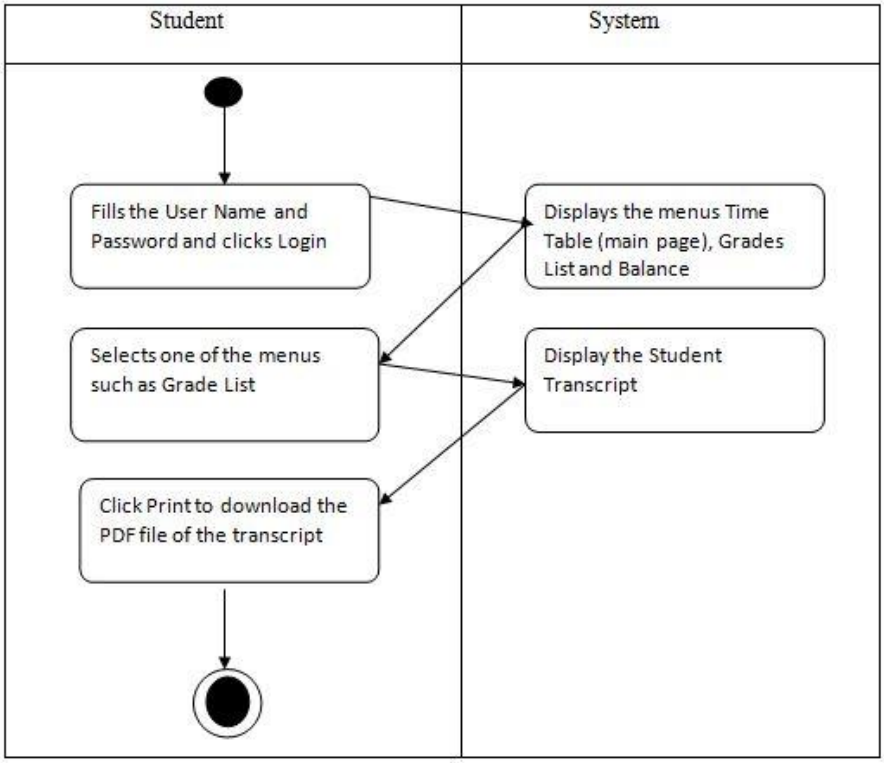


Figure 4.7 Activity Diagram of the proposed Student Information

4.1.3.3. Sequence Diagram

A sequence diagram is an interaction diagram. It deals with some sequences, which are the sequence of messages flowing from one object to another. To see the interaction between objects, the following describe the sequence diagram of each identified use cases.

The control objectsnamed *"LoginControl, TimetableControl, RegistrationControl, EnrollmentControl, PaymentControl and GradeControl"*manages the activities involved in each use case respectively as shown in the figures.

1. Sequence Diagram Administration (Create Class Timetable)

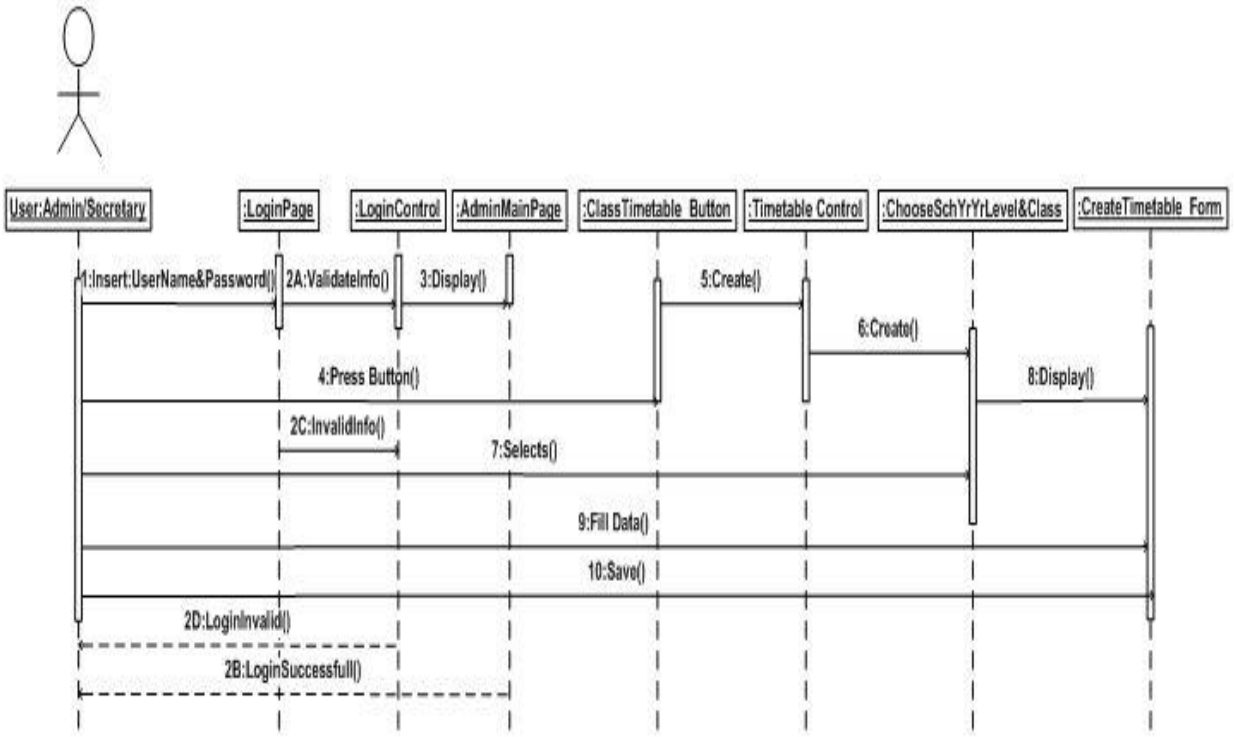


Figure 4.8 Sequence Diagram Administration (Create Class Timetable)

2. Sequence Diagram Registration

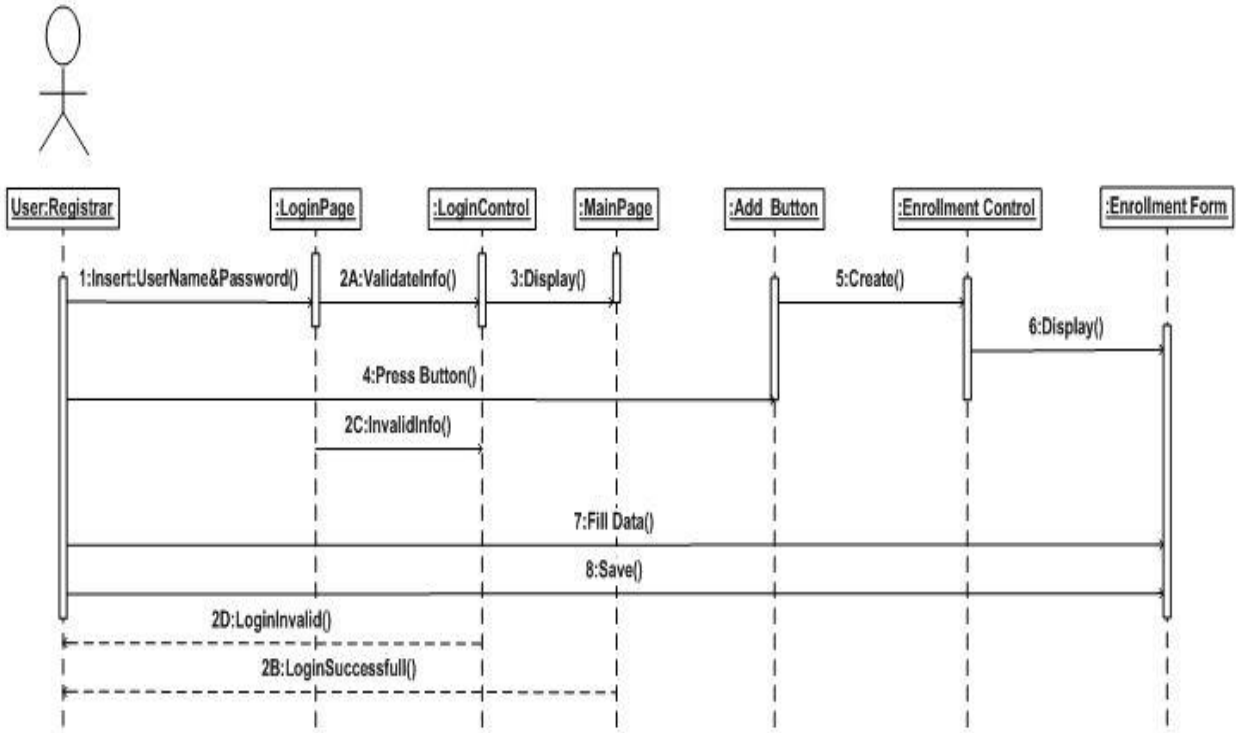


Figure 4.9 Sequence Diagram Registration

3. Sequence Diagram Enrollment

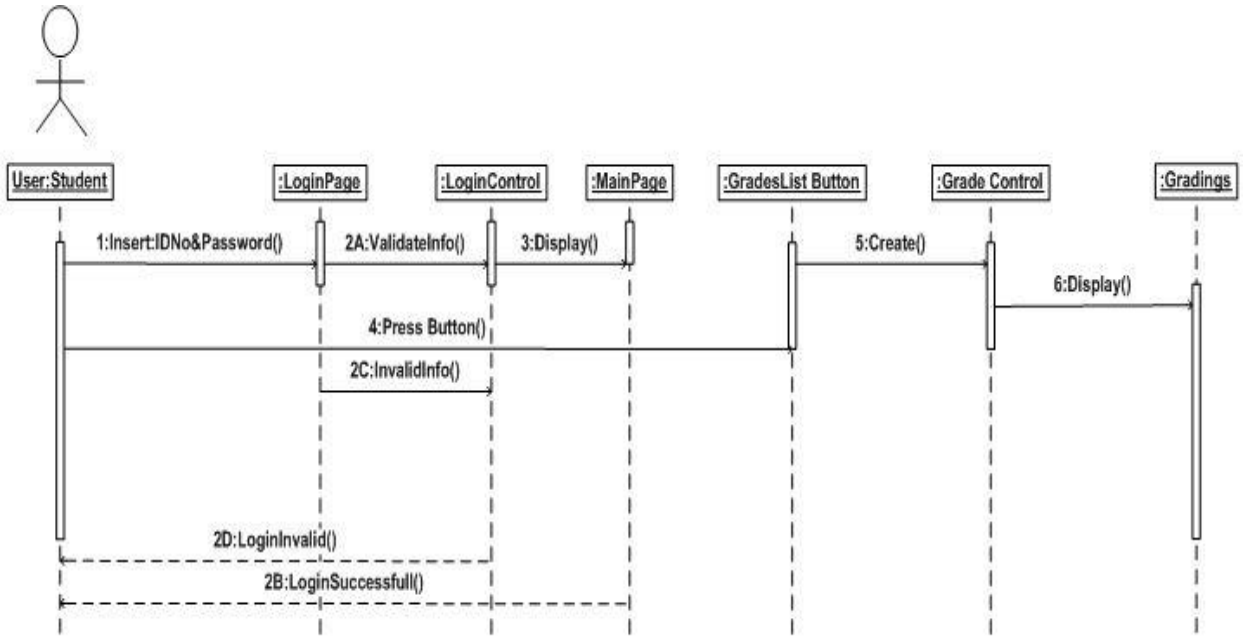


Figure 4.10 Sequence Diagram Enrollment

4. Sequence Diagram Accounting & Financial Records

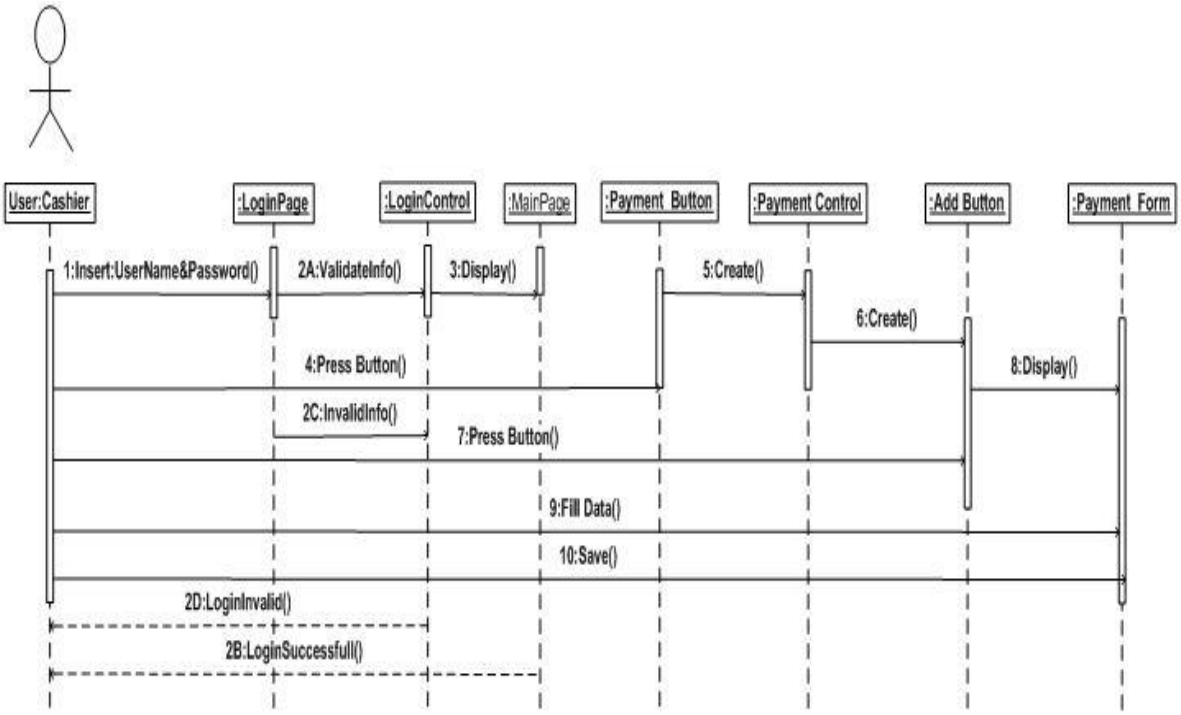


Figure 4.11 Sequence Diagram Accounting & Financial Records

5. Sequence Diagram Student Information (View Student Grades)

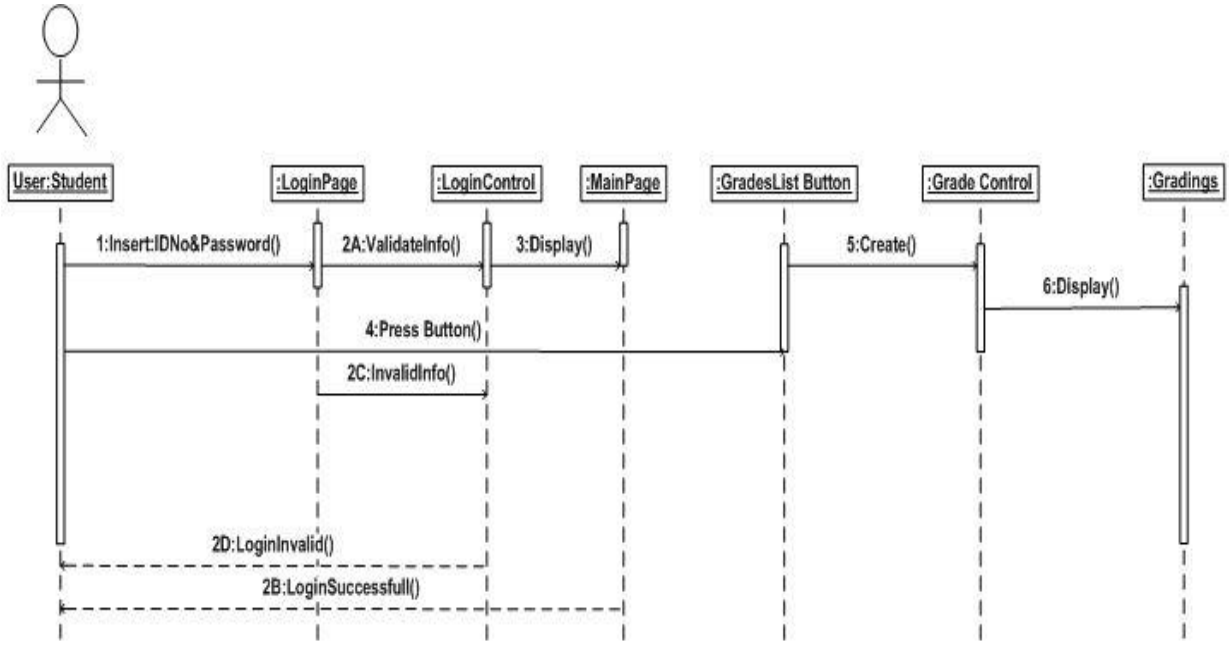


Figure 4.12 Sequence Diagram Student Information (View Student Grades)

4.1.3.4. Class Diagram

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations and collaboration. Class diagrams basically represent the object oriented view of a system which is static in nature.

Here is the Class Diagram of the designed system.

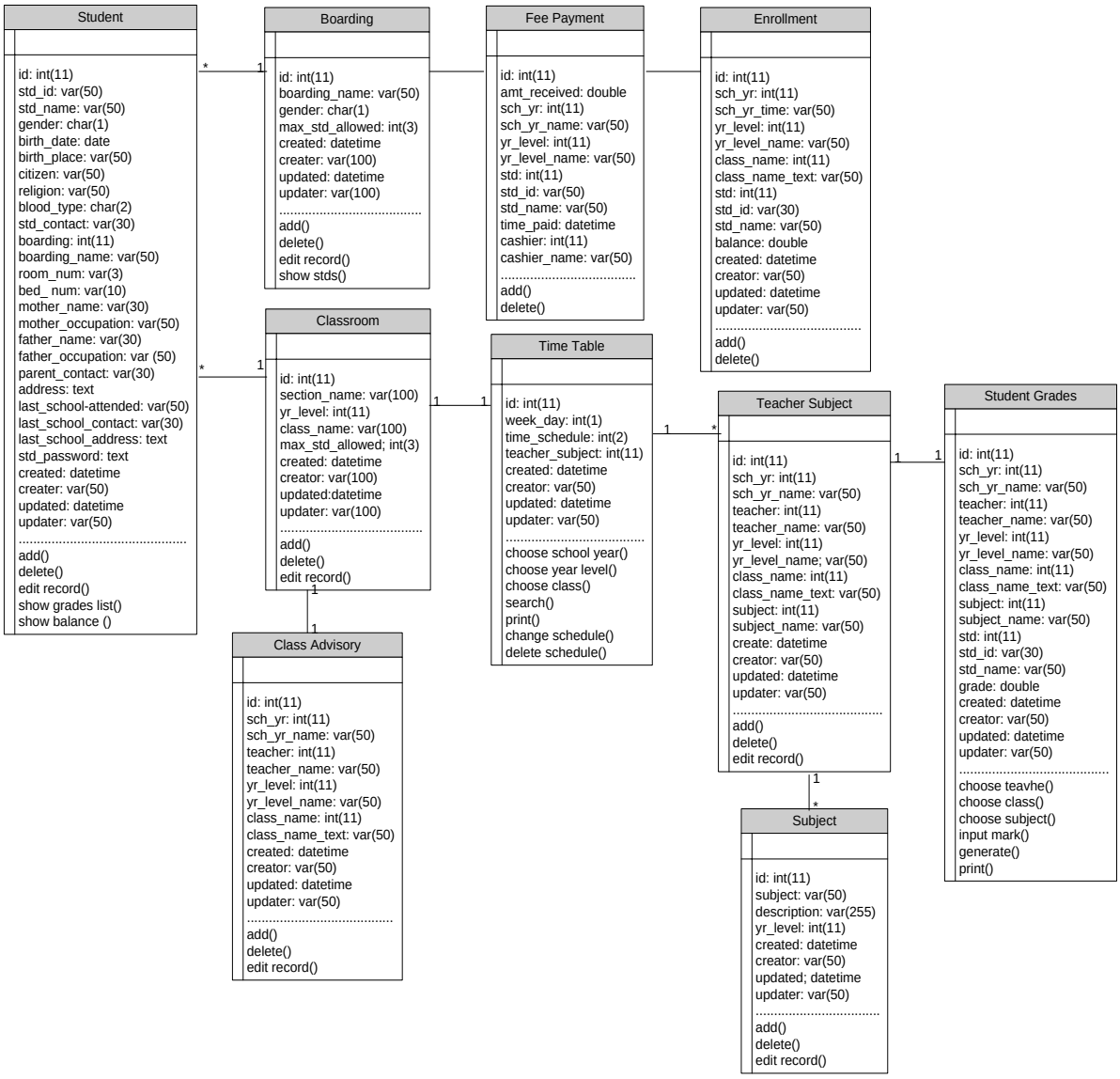


Figure 4.13 Class Diagram

4.1.3.5. Deployment Diagram

Deployment diagrams are a set of nodes and their relationships. These nodes are physical entities where the components are deployed.

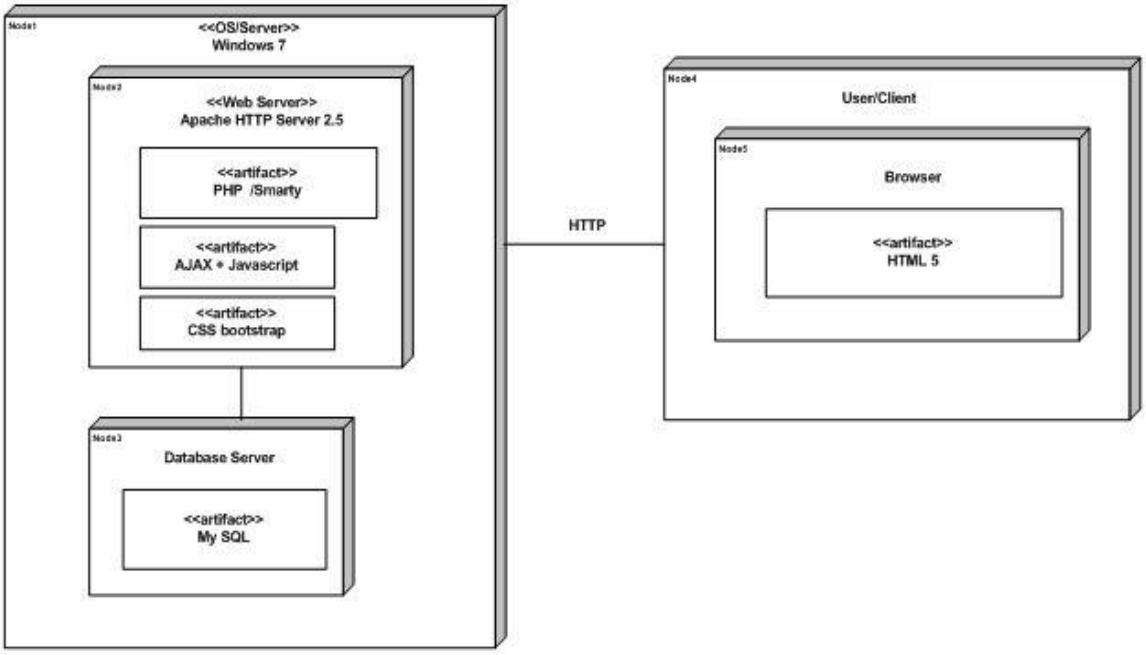


Figure 4.14 Deployment Diagram

4.1.3.6. Network Architecture of the Proposed System

The proposed system is expected to replace the existing manual system by an automated system in all facets. It is mainly based on the system Analysis document (chapter 3).

The architecture used for the system is a 3 tier Client/Server Architecture where a client can use Internet browsers to access the applications provided by the

system within the local area network of the school or any where using the Internet. Figure 4.14 below shows the network architecture of the proposed system.

The data tier maintains the applications data such as student data, teacher data, timetable data, etc. It stores these data in a relational database management system (RDBMS).

The middle tier (web/application server) implements the business logic, controller logic and presentation logic to control the interaction between the application's clients and data. The controller logic processes client requests such as requests to view student's grades, to register/ enroll students or to retrieve data from the database. Business rules enforced by the business logic dictate how clients can and cannot access application data and how applications process data.

A web server is a program that runs on a network server (computer) to respond to HTTP requests. The most commonly used web servers are Internet Information Server (IIS) and Apache. The web server used in this system is Apache. HTTP is used to transfer data across an Intranet or the Internet. It is the standard protocol for moving data across the internet.

The client tier is the applications user interface containing data entry forms and client side applications. It displays data to the user. Users interact directly with the application through user interface. The client tier interacts with the web/application server to make requests and to retrieve data from the database. It then displays to the user the data retrieved from the server.

The Network Architecture of the Proposed System is shown below:

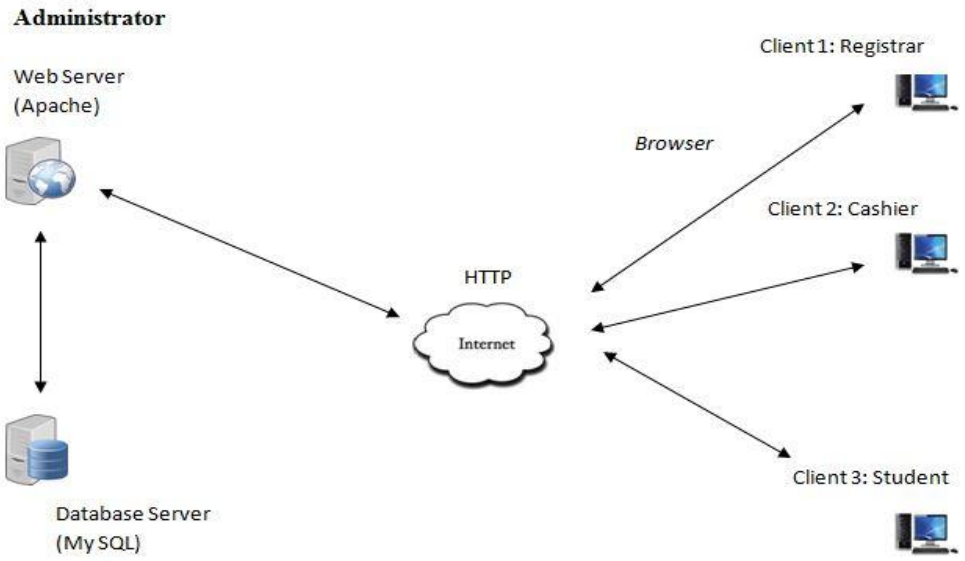


Figure 4.15 Network Architecture

4.2. Implementation

The implementation of this prototype application of Fr. Peter Secondary School Management System is done using PHP and Smarty, MySQL as the database and through using Apache HTTP Server (XAMPP) as the web server. This application is implemented on Windows 7 Operating System.

The purpose of this implementation is to make it possible for the users or clients to have access to the developed system. That is, the users will see whether the newly build system satisfies or meet their requirements or not. So from user's judgment and

feedbacks, this system could be (or not) further re-build in order to satisfy the users' needs. .

4.2.1. Implementation Limits

The limit on the implementation of Fr. Peter Secondary School Management System includes:

1. Implementation of this application is only accessible by four users: admin/secretary, registrar, cashier and student.
2. Implementation of the application is only built for Fr. Peter Secondary School.

4.2.2. Software Implementation

The application can be installed on a PC or laptop with Windows 7 operating system. The required softwares to be used in the PC/laptop for the implementation of this system is as follows:

1. Operating System -Windows 7
2. XAMPP-2.5 for Internet and creating local database and data storage.
3. Smarty
4. PHP Storm, Notepad ++, Macromedia Dreamweaver 8, to write coding.
5. Mozilla Firefox as a browser.

4.2.3. Hardware Implentation

In order to to run the designed application,it needs hardware to rely on. The hardware requirements are as follows:

1. Processor: Intel (R) Core(TM) i3 CPU

2. Memory: 2GB RAM

3. Hard drive: 500GB

4.2.4. Database Implementation (SQL syntax)

Database Implementation taken is based on the design of the database that was created for this system. The following table structure was implemented in the database with their respective indexes:

1. "tbl_class_advisory"

```
CREATE TABLE IF NOT EXISTS `tbl_class_advisory` (
  `Id` int(11) NOT NULL,
  `SchoolYear` int(11) DEFAULT NULL,
  `SchoolYearName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Teacher` int(11) DEFAULT NULL,
  `TeacherName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `YearLevel` int(11) DEFAULT NULL,
  `YearLevelName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `ClassName` int(11) DEFAULT NULL,
  `ClassNameText` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_class_advisory`
ALTER TABLE `tbl_class_advisory`
ADD PRIMARY KEY (`Id`),
ADD KEY `FK1` (`SchoolYear`,`SchoolYearName`),
ADD KEY `FK2` (`Teacher`,`TeacherName`),
ADD KEY `FK3` (`YearLevel`,`YearLevelName`),
ADD KEY `FK4` (`ClassName`,`ClassNameText`);
```

2. “tbl_enrollment”

```
CREATE TABLE IF NOT EXISTS `tbl_enrollment` (
  `Id` int(11) NOT NULL,
  `SchoolYear` int(11) DEFAULT NULL,
  `SchoolYearName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `YearLevel` int(11) DEFAULT NULL,
  `YearLevelName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `ClassName` int(11) DEFAULT NULL,
  `ClassNameText` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Student` int(11) DEFAULT NULL,
  `StudentId` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
  `StudentName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Balance` double DEFAULT NULL,
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_enrollment`
ALTER TABLE `tbl_enrollment`
ADD PRIMARY KEY (`Id`),
ADD KEY `FK1` (`SchoolYear`,`SchoolYearName`),
ADD KEY `FK3` (`YearLevel`,`YearLevelName`),
ADD KEY `FK4` (`ClassName`,`ClassNameText`),
ADD KEY `FK5` (`StudentId`);
```

3. “tbl_fee_paid”

```
CREATE TABLE IF NOT EXISTS `tbl_fee_paid` (
  `Id` int(11) unsigned NOT NULL,
  `AmountReceived` double DEFAULT '0',
  `Balance` double DEFAULT '0',
  `SchoolYear` int(11) DEFAULT NULL,
  `SchoolYearName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `YearLevel` int(11) DEFAULT NULL,
  `YearLevelName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Student` int(11) DEFAULT NULL,
  `StudentId` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `StudentName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `TimePaid` datetime DEFAULT NULL,
  `Cashier` int(11) DEFAULT NULL,
  `CashierName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_fee_paid`

ALTER TABLE `tbl_fee_paid`

ADD PRIMARY KEY (`Id`);
```

4. “tbl_log_user”

```
CREATE TABLE IF NOT EXISTS `tbl_log_user` (
  `id_log` int(10) unsigned NOT NULL,
  `user_id` int(10) unsigned DEFAULT NULL,
  `time_login` datetime DEFAULT NULL,
  `time_logout` datetime DEFAULT NULL,
  `user_ip` varchar(100) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_log_user`

ALTER TABLE `tbl_log_user`

  ADD PRIMARY KEY (`id_log`);
```

5. “tbl_md_boarding”

```
CREATE TABLE IF NOT EXISTS `tbl_md_boarding` (
  `Id` int(11) NOT NULL,
  `BoardingName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Gender` char(1) COLLATE latin1_general_ci DEFAULT NULL,
  `MaxStudentsAllowed` int(3) DEFAULT NULL,
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(100) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_md_boarding`

ALTER TABLE `tbl_md_boarding`

ADD PRIMARY KEY (`Id`),

ADD KEY `Id` (`Id`);
```

6. “tbl_md_fee”

```
CREATE TABLE IF NOT EXISTS `tbl_md_fee` (
  `Id` int(11) unsigned NOT NULL,
  `FeeName` varchar(50) COLLATE latin1_general_ci DEFAULT '',
  `Description` varchar(255) COLLATE latin1_general_ci DEFAULT '',
  `Amount` double DEFAULT NULL,
  `YearLevel` int(11) DEFAULT NULL,
  `YearLevelName` varchar(3) COLLATE latin1_general_ci DEFAULT NULL,
  `IsBoarding` int(1) DEFAULT '0',
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
```

```
Indexes for table `tbl_md_fee`
ALTER TABLE `tbl_md_fee`
  ADD PRIMARY KEY (`Id`);
```

7. "tbl_md_school_year"

```
CREATE TABLE IF NOT EXISTS `tbl_md_school_year` (
  `Id` int(11) NOT NULL,
  `SchoolYear` varchar(15) COLLATE latin1_general_ci DEFAULT NULL,
  `IsActive` int(1) DEFAULT '0',
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_md_school_year`
ALTER TABLE `tbl_md_school_year`
  ADD PRIMARY KEY (`Id`),
  ADD KEY `Id` (`Id`,`SchoolYear`);
```

8. "tbl_md_sections"

```
CREATE TABLE IF NOT EXISTS `tbl_md_sections` (
  `Id` int(11) NOT NULL,
  `SectionName` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
  `YearLevel` int(11) DEFAULT NULL,
  `ClassName` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
  `MaxStudentsAllowed` int(3) DEFAULT NULL,
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
```

```
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(100) COLLATE latin1_general_ci DEFAULT
NULL
)
Indexes for table `tbl_md_sections`

ALTER TABLE `tbl_md_sections`

  ADD PRIMARY KEY (`Id`),

  ADD KEY `FK_01` (`YearLevel`),

  ADD KEY `Id` (`Id`,`ClassName`);
```

9. "tble_md_students"

```
    CREATE TABLE IF NOT EXISTS `tbl_md_students` (
      `Id` int(11) unsigned NOT NULL,
      `StudentId` varchar(30) COLLATE latin1_general_ci
DEFAULT NULL,
      `StudentName` varchar(50) COLLATE latin1_general_ci
DEFAULT NULL,
      `Gender` char(1) COLLATE latin1_general_ci DEFAULT
NULL,
      `BirthDate` date DEFAULT NULL,
      `BirthPlace` varchar(50) COLLATE latin1_general_ci
DEFAULT NULL,
      `Citizen` varchar(50) COLLATE latin1_general_ci
DEFAULT NULL,
      `Religion` varchar(50) COLLATE latin1_general_ci
DEFAULT NULL,
      `BloodType` char(2) COLLATE latin1_general_ci
DEFAULT NULL,
      `StudentContact` varchar(30) COLLATE latin1_general_ci
DEFAULT NULL,
      `Boarding` int(11) DEFAULT NULL,
      `BoardingName` varchar(50) COLLATE latin1_general_ci
DEFAULT NULL,
      `RoomNumber` varchar(3) COLLATE latin1_general_ci
DEFAULT NULL,
```

```
    `BedNumber` varchar(10) COLLATE latin1_general_ci DEFAULT NULL,
    `MotherName` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
    `MotherOccupation` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
    `FatherName` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
    `FatherOccupation` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
    `ParentContact` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
    `Address` text COLLATE latin1_general_ci,
    `LastSchoolAttend` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
    `LastSchoolContact` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
    `LastSchoolAddress` text COLLATE latin1_general_ci,
    `StudentPassword` text COLLATE latin1_general_ci,
    `Created` datetime DEFAULT NULL,
    `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
    `Updated` datetime DEFAULT NULL,
    `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_md_students`

ALTER TABLE `tbl_md_students`

 ADD PRIMARY KEY (`Id`);
```

10. “tbl_md_subjects”

```
CREATE TABLE IF NOT EXISTS `tbl_md_subjects` (
 `Id` int(11) NOT NULL,
 `Subject` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
 `Description` varchar(255) COLLATE latin1_general_ci DEFAULT NULL,
 `YearLevel` int(11) DEFAULT NULL,
 `Created` datetime DEFAULT NULL,
```

```
  `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_md_subjects`

ALTER TABLE `tbl_md_subjects`

  ADD PRIMARY KEY (`Id`);
```

11.“tbl_md_year_level”

```
CREATE TABLE IF NOT EXISTS `tbl_md_year_level` (
  `Id` int(11) NOT NULL,
  `YearLevel` varchar(3) COLLATE latin1_general_ci DEFAULT NULL,
  `Description` text COLLATE latin1_general_ci,
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(100) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_md_year_level`

ALTER TABLE `tbl_md_year_level`

  ADD PRIMARY KEY (`Id`),

  ADD KEY `Id` (`Id`,`YearLevel`);
```

12. “tbl_parameter_settings”

```
CREATE TABLE IF NOT EXISTS `tbl_parameter_settings` (
  `ParameterName` varchar(25) COLLATE latin1_general_ci NOT NULL,
  `ParameterValue` text COLLATE latin1_general_ci NOT NULL,
  `Description` text COLLATE latin1_general_ci NOT NULL
)
Indexes for table `tbl_parameter_settings`

ALTER TABLE `tbl_parameter_settings`

 ADD PRIMARY KEY (`ParameterName`);
```

13. “tbl_sessions”

```
CREATE TABLE IF NOT EXISTS `tbl_sessions` (
  `session_id` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `session_user_id` float NOT NULL,
  `session_start` float NOT NULL,
  `session_time` float NOT NULL,
  `session_ip` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `session_page` float NOT NULL,
  `session_logged_in` float NOT NULL,
  `session_admin` float NOT NULL
)
```

14. “tbl_student_grades”

```
CREATE TABLE IF NOT EXISTS `tbl_student_grades` (
  `Id` int(11) NOT NULL,
  `SchoolYear` int(11) DEFAULT NULL,
  `SchoolYearName` varchar(50) COLLATE latin1_general_ci DEFAULT
NULL,
  `Teacher` int(11) DEFAULT NULL,
  `TeacherName` varchar(50) COLLATE latin1_general_ci DEFAULT
NULL,
  `YearLevel` int(11) DEFAULT NULL,
  `YearLevelName` varchar(50) COLLATE latin1_general_ci DEFAULT
NULL,
  `ClassName` int(11) DEFAULT NULL,
  `ClassNameText` varchar(50) COLLATE latin1_general_ci DEFAULT
```

```
NULL,
  `Subject` int(11) DEFAULT NULL,
  `SubjectName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Student` int(11) DEFAULT NULL,
  `StudentId` varchar(30) COLLATE latin1_general_ci DEFAULT NULL,
  `StudentName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Grade` double DEFAULT NULL,
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_student_grades`

ALTER TABLE `tbl_student_grades`

  ADD PRIMARY KEY (`Id`),

  ADD KEY `FK1` (`SchoolYear`,`SchoolYearName`),

  ADD KEY `FK2` (`Teacher`,`TeacherName`),

  ADD KEY `FK4` (`ClassName`,`ClassNameText`);
```

15. "tbl_teacher_subject"

```
CREATE TABLE IF NOT EXISTS `tbl_teacher_subject` (
  `Id` int(11) NOT NULL,
  `SchoolYear` int(11) DEFAULT NULL,
  `SchoolYearName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Teacher` int(11) DEFAULT NULL,
  `TeacherName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `YearLevel` int(11) DEFAULT NULL,
  `YearLevelName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `ClassName` int(11) DEFAULT NULL,
  `ClassNameText` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Subject` int(11) DEFAULT NULL,
```

```
  `SubjectName` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_teacher_subject`

ALTER TABLE `tbl_teacher_subject`

  ADD PRIMARY KEY (`Id`),

  ADD KEY `FK1` (`SchoolYear`,`SchoolYearName`),

  ADD KEY `FK2` (`Teacher`,`TeacherName`),

  ADD KEY `FK3` (`YearLevel`,`YearLevelName`),

  ADD KEY `FK4` (`ClassName`,`ClassNameText`);
```

16. "tbl_time_table"

```
CREATE TABLE IF NOT EXISTS `tbl_time_table` (
  `Id` int(11) unsigned NOT NULL,
  `WeekDay` int(1) DEFAULT NULL,
  `TimeSchedule` int(2) DEFAULT NULL,
  `TeacherSubject` int(11) DEFAULT NULL,
  `Created` datetime DEFAULT NULL,
  `Creator` varchar(50) COLLATE latin1_general_ci DEFAULT NULL,
  `Updated` datetime DEFAULT NULL,
  `Updater` varchar(50) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_time_table`

ALTER TABLE `tbl_time_table`

  ADD PRIMARY KEY (`Id`);
```

17. “tbl_user”

```
CREATE TABLE IF NOT EXISTS `tbl_user` (
  `user_id` int(11) NOT NULL,
  `BranchCode` varchar(20) COLLATE latin1_general_ci NOT NULL,
  `user_active` float NOT NULL,
  `username` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `user_password` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `user_session_time` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `user_session_page` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `user_lastvisit` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `user_level` int(2) NOT NULL,
  `user_timezone` float NOT NULL,
  `apply` datetime NOT NULL,
  `NRP` varchar(20) COLLATE latin1_general_ci DEFAULT NULL,
  `name` varchar(50) COLLATE latin1_general_ci NOT NULL,
  `birthday` datetime NOT NULL,
  `telp` varchar(20) COLLATE latin1_general_ci NOT NULL,
  `hp` varchar(20) COLLATE latin1_general_ci NOT NULL,
  `email` varchar(50) COLLATE latin1_general_ci NOT NULL,
  `address` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `status_online` int(1) NOT NULL,
  `last_time_updated` datetime DEFAULT NULL,
  `time_login` datetime NOT NULL,
  `time_logout` datetime NOT NULL,
  `created` datetime DEFAULT NULL,
  `creator` varchar(100) COLLATE latin1_general_ci DEFAULT NULL,
  `updated` datetime DEFAULT NULL,
  `updater` varchar(100) COLLATE latin1_general_ci DEFAULT NULL
)
Indexes for table `tbl_user`

ALTER TABLE `tbl_user`

  ADD PRIMARY KEY (`user_id`),

  ADD KEY `user_id` (`user_id`,`name`);
```

4.3. Interface Implementation

Implementation of Fr. Peter Secondary School Management Web-based System is only assessable by four (4) users, that is; The Admin/Secretary, Registrar, Cashier, and Students in Fr. Peter Secondary School.

4.3.1. Following is the interface implementation of user: Admin/Secretary

1. Login

When opening this application, this is the login page displayed. So the Admin must type in the User Name and Password to log in (click log in).

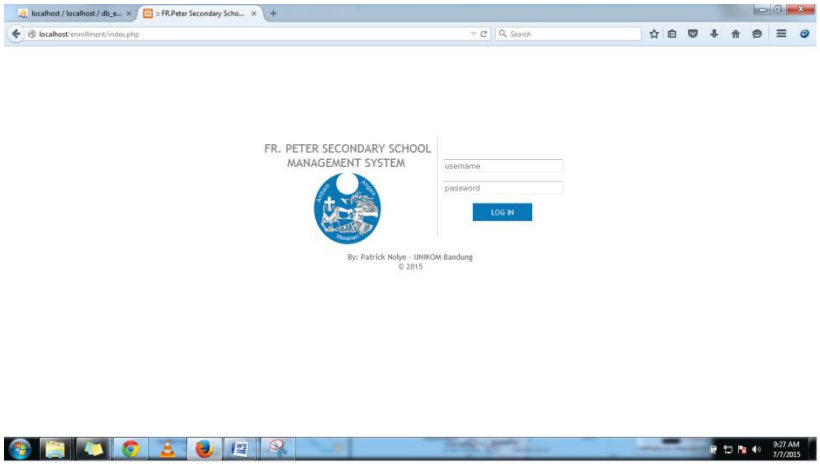


Figure 4.16 Login Page-Admin/Secretary

2. Home-Admin

Once logged in successfully, the main menu is the ”USER” page.

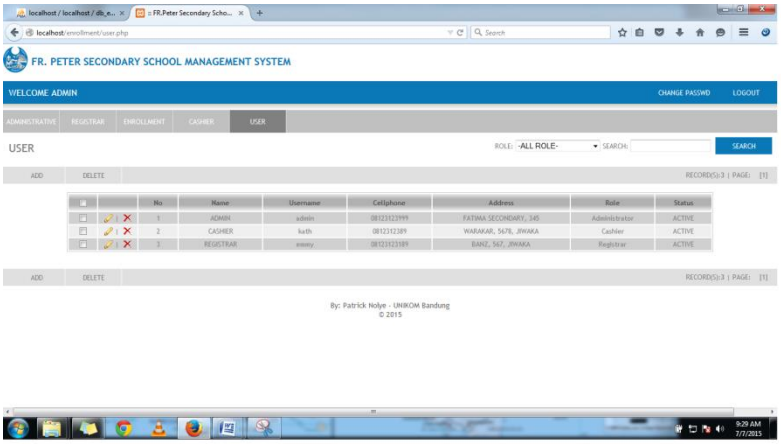


Figure 4.17 Main Page-Admin

3. The Sub-Menus of Administrative

The admin can access all the menus. Anyway, particularly to take care of its function, may select Administrative menu. There are nine (9) sub-menus in Administrative

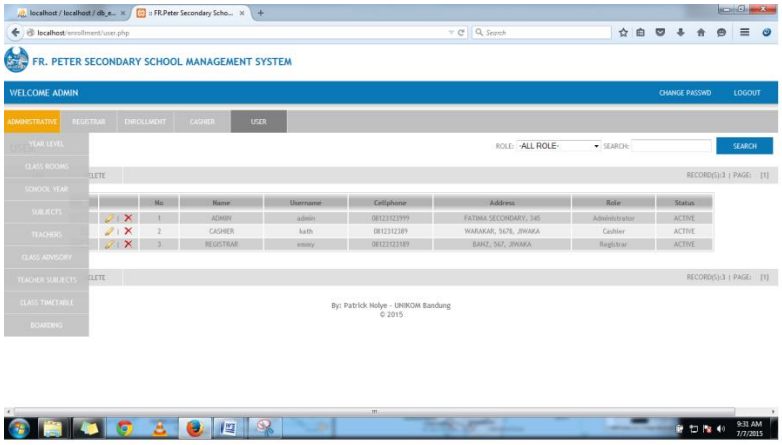


Figure 4.18 Sub-Menus-Administrative

4. Year Level Menu

In this menu, year levels or grades level are created.

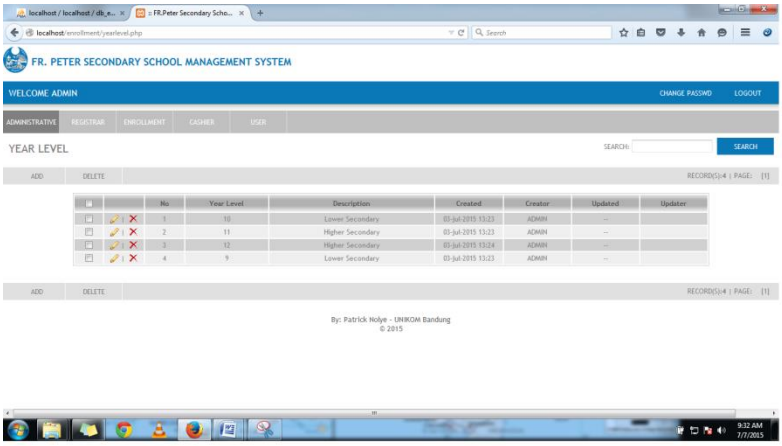


Figure 4.19 Year Level Menu

4.1 Adding New Year Level Menu

To add a New Year Level, click Add (as shown above) and fill in the required information and save it.

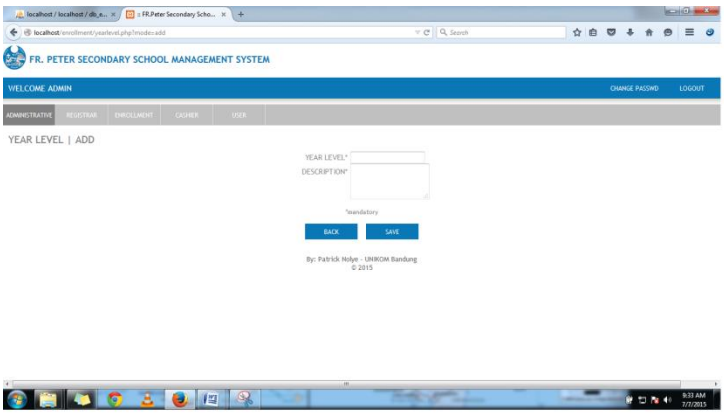


Figure 4.20 Add New Year Level Menu

5. Classrooms Menu

The Admin uses this menu to create classrooms for each grade.

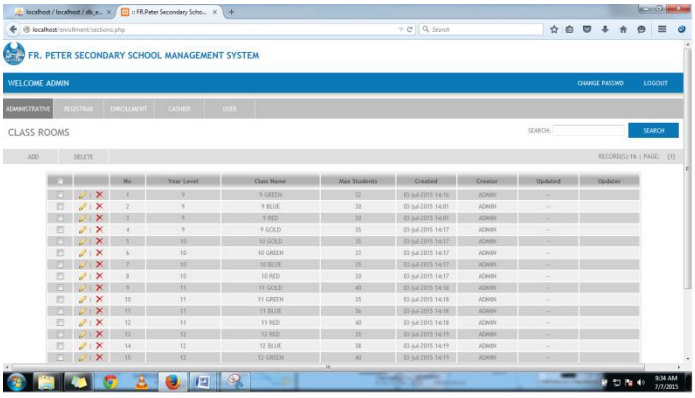


Figure 4.21 Classrooms Menu

5.1 Add New Classroom Menu

The Secretary clicks Add and fills in the form displayed to add a new classroom.

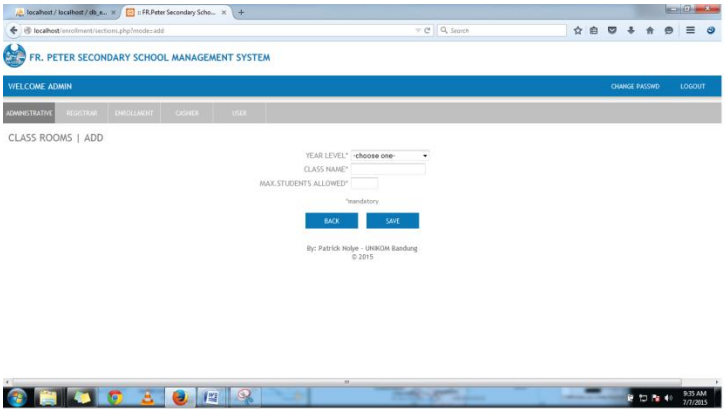


Figure 4.22 Add New Classrooms Menu

6. School Year Menu

In this menu, the Admin add School Year and only activates the current academic year.

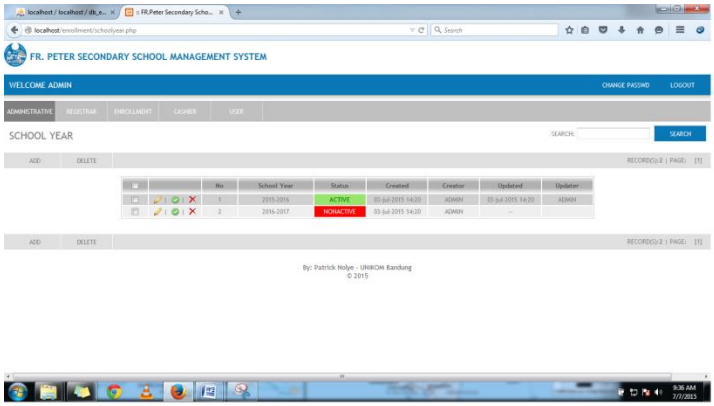


Figure 4.23 School Year Menu

6.1 Add New School Year Menu

In order to add new school year, secretary clicks the Add button and fills in the form as shown below.

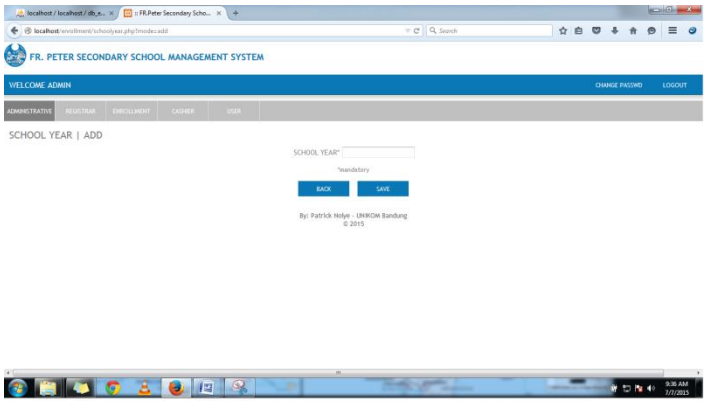


Figure 4.24 Add New School Year Menu

7. Subject Menu

In this menu, the Admin creates Subjects for every grades or year levels.

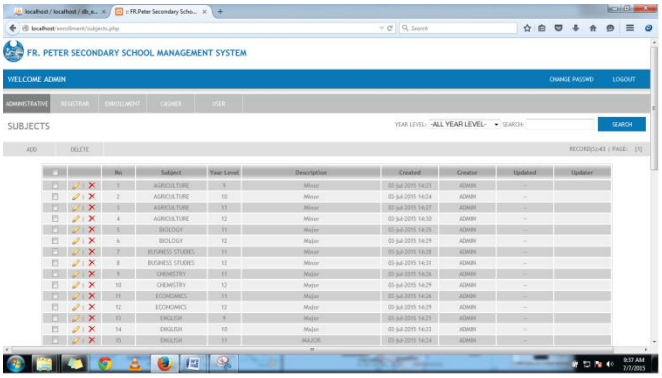


Figure 4.25 Subjects Menu

7.1 Add New Subject Menu

In this menu secretariat Prodi can print the data dossier of each semester based courses that diampuh by lecturer.

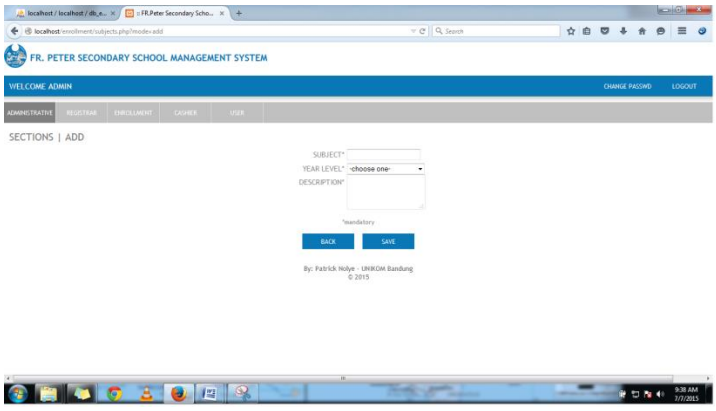


Figure 4.26 Add New Subject Menu

8. Teachers Menu

This menu deals with adding of teachers to the school system.

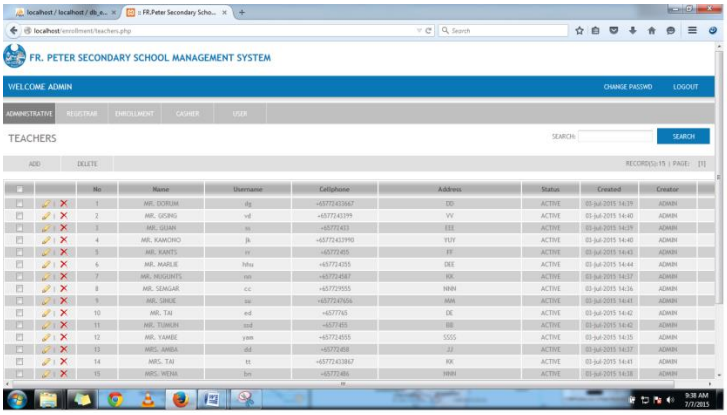


Figure 4.27 Teachers Menu

8.1 Add New Teacher Menu

Click Add on the teachers menu above to add a new teacher. The Admin make sure fills the form correctly.

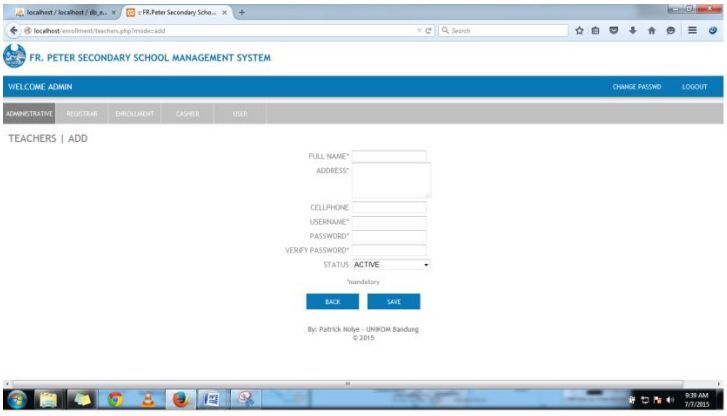


Figure 4.28 Add New Teacher Menu

9. Class Advisory Menu

In this menu, the Secretary allocates teachers to each class and grade as class advisory (class patron/matron)

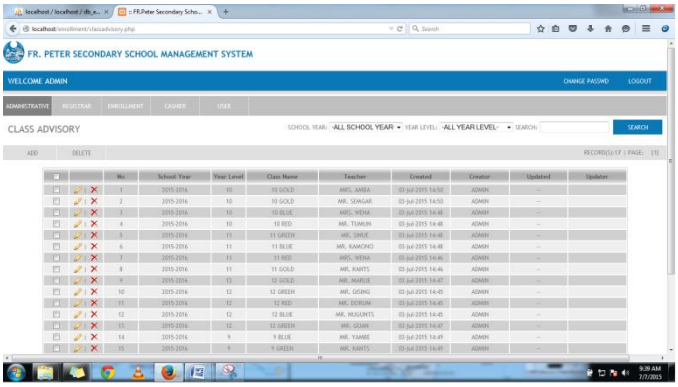


Figure 4.29 Class Advisory Menu

9.1 Add New Class Advisory Menu

This menu is used to add new class advisory by the Admin.

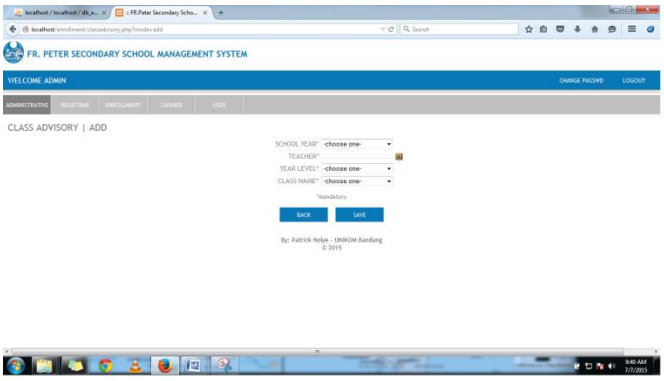


Figure 4.30 Add New Class Advisory Menu

10. Teacher Subject Menu

The Secretary uses this menu to allocate subjects to respective teachers and also the classroom where they will teach those subjects.

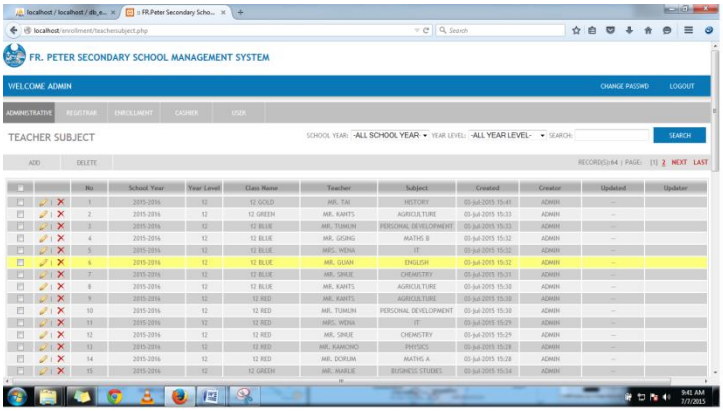


Figure 4.31 Teacher Subject Menu

10.1 Add New Teacher Subject Menu

The menu provides the form to be filled by the Admin in order to allocate subjects to new teachers and assigned them in their respective classrooms.

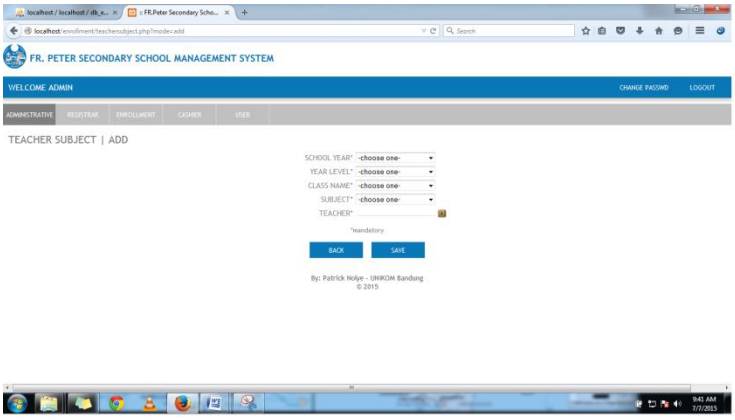


Figure 4.32 Add New Teacher Subject Menu

11. Timetable Menu

When the Admin clicks on this menu, an empty form is shown in order to be filled and create that particular timetable for which school year, year level and class.

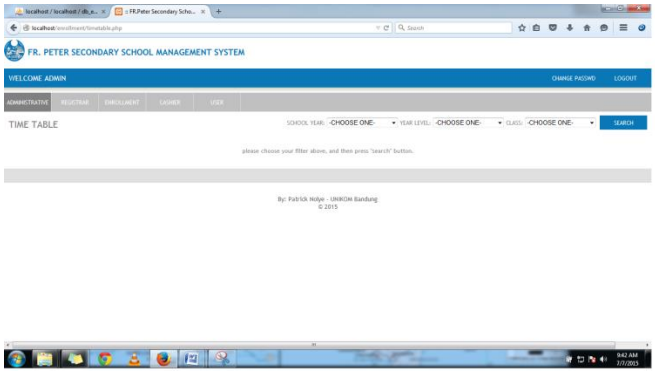


Figure 4.33 Timetable Menu

11.1 Creating Class Time Table Menu

This menu is used by the Secretary to create the class timetable. Clicks add in each time period to add a class/subject

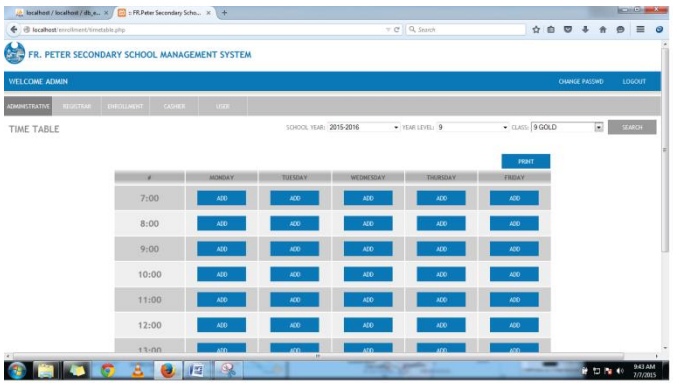


Figure 4.34 Creating Class Timetable Menu

11.2 Created Class Timetable Menu

If the Admin fills in the form and if a class timetable was already created, the created timetable will appear so the Admin can just edit the timetable.

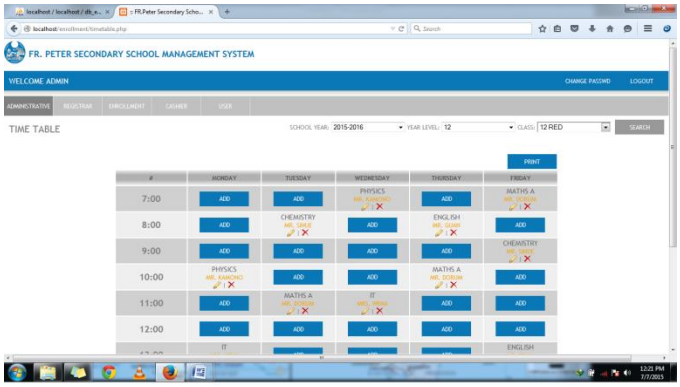


Figure 4.35 Created Class Timetable Menu

12. Boarding Menu

This menu is used to create boarding dormitories and rooms for both genders (male/female) respectively

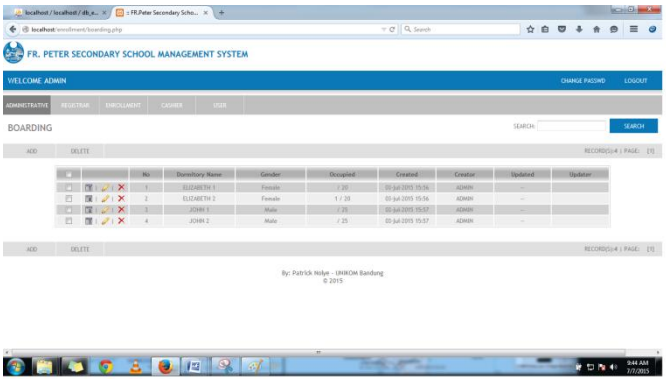


Figure 4.36 Boarding Menu

12.1 New Boarding-Dormitory Menu

This menu provides the form for the Admin to add a new dormitory with the number of space available to a particular gender(male/female).

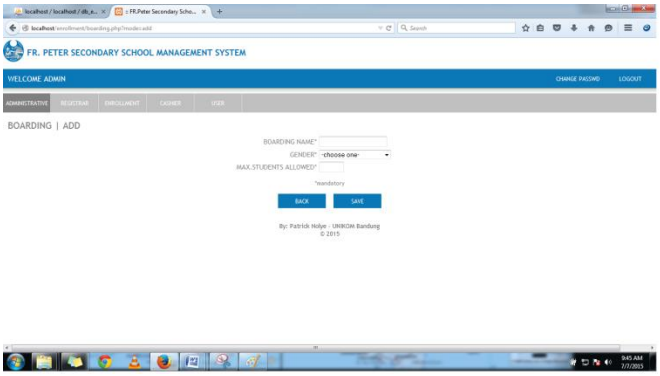


Figure 4.37 New Boarding-Dormitory Menu

13. User Menu

Only the Admin/Secretary takes care of the user menu. That is, allocate a user name and password to various roles/positions like Registrar and Cashier.

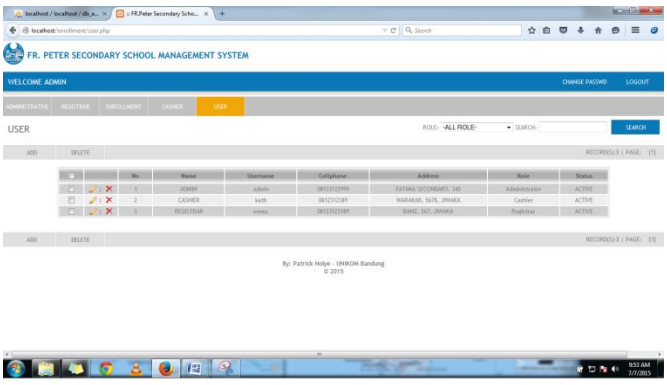


Figure 4.38 User Menu

13.1 Add New User Menu

The Admin uses this menu to create and add new users to the system. In this system, only three (3) users are added; Admin (itself), Cashier and Registrar.

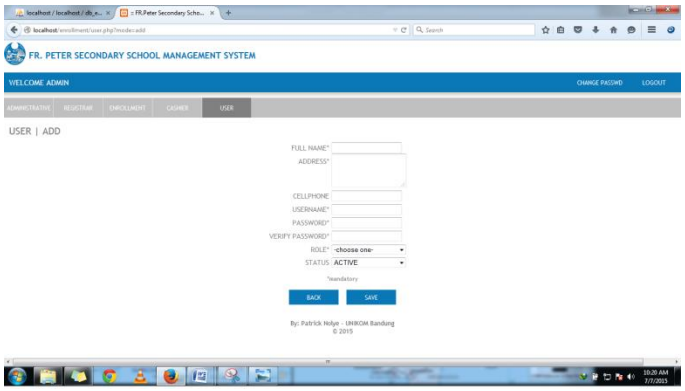


Figure 4.39 Add New User Men

4.3.2. The Following is the Interface Implementation of the User: Registrar

1. Registrar Login Page

When the Registrar opens the application, this is the login page displayed. So the Registrar fills in the user name and password to log in.

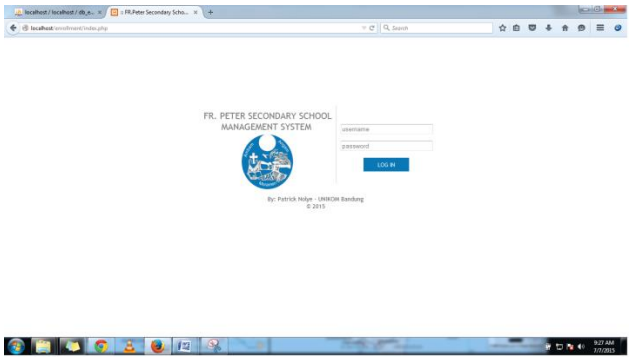


Figure 4.40 Registrar Login Page Menu

2. Registrar Home Page

If successfully logged in, this is the home page displayed. There are only two (2) menus whereby the registrar takes care of: Registrar and Enrollment menus.

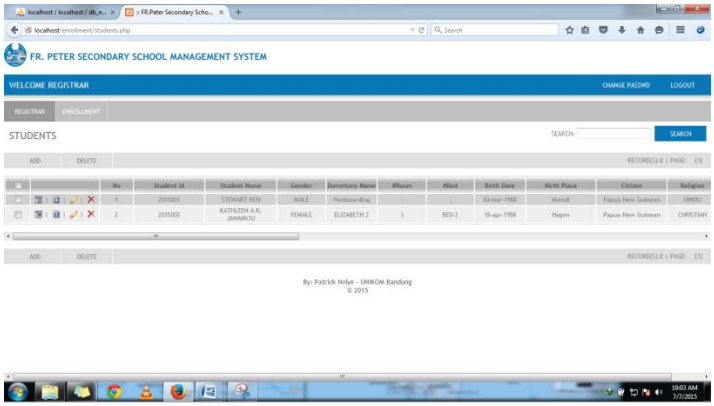


Figure 4.41 Registrar Home Page Menu

3. Registrar Sub-Menus

There are three (3) sub-menus in registrar menu; Students, Encoding of Grades and View Student Grades.

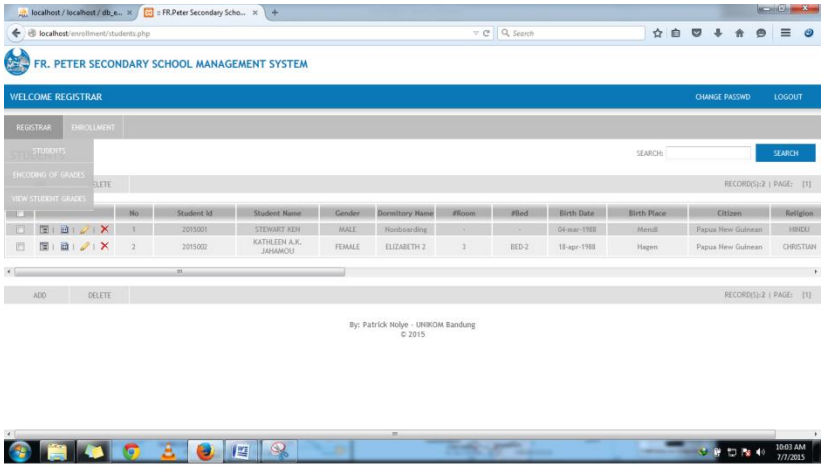


Figure 4.42 Registrar Sub-Menu

3.1 Students Menu

The Registrar uses this menu to register new students.

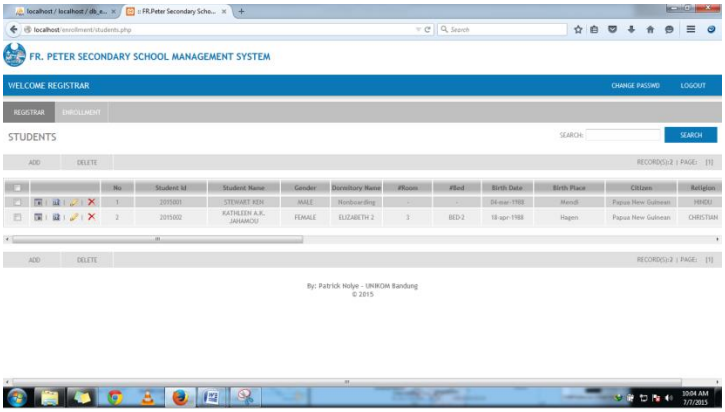


Figure 4.43 Students Menu

3.1.1 New Student Registration Menu

In order to register a new student, the Registrar clicks add and the form as shown below is displayed. The Registrar fills up all the necessary information of the student to register him/her.

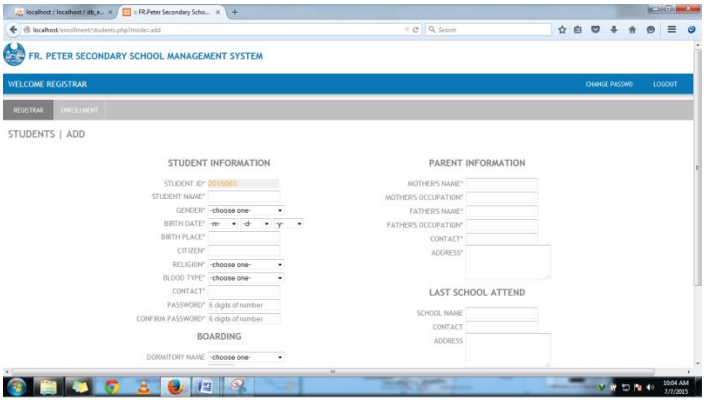


Figure 4.44 New Student Registration Menu

3.2 Encoding of Grades Menu

The Registrar uses this menu to encode grades of each subject to the student according to their classroom and subjects studied.

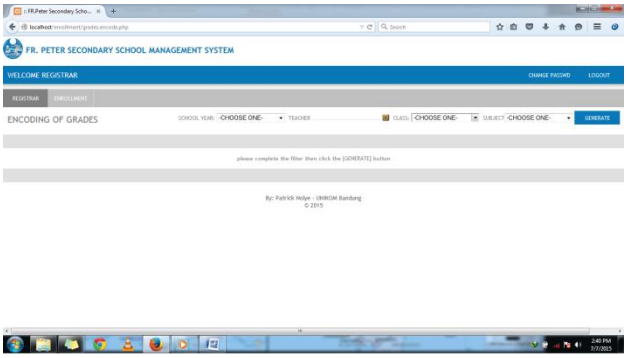


Figure 4.45 Encoding of Grades Menu

3.2.1 Confirm Student Grades

Registrar can confirm student grade using the options provide by clicking Show Grades List.

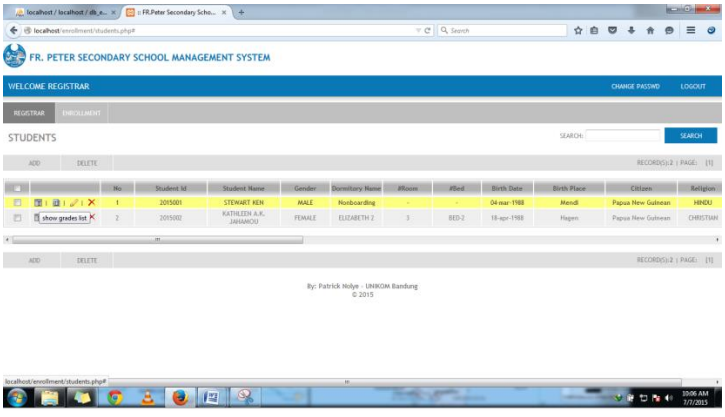


Figure 4.46 Student Grade

3.2.2 View Student Grade

When the Registrar clicks Show Grades List, the system shows the grades or marks of that particular student only.

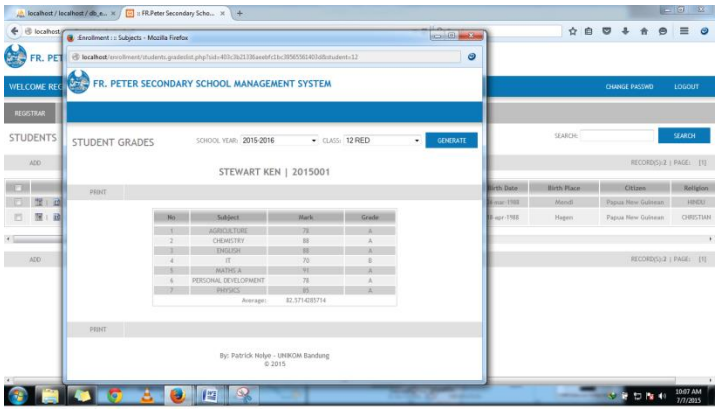


Figure 4.47 View Student Grades

3.3 View Student Grades Menu

The Registrar uses this menu to view and print a student grade. Just input the school year, class name and the student name and clicks generate.

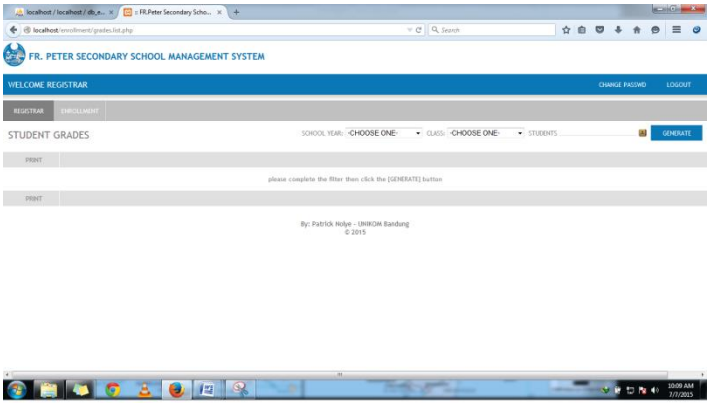


Figure 4.48 View Student Grades Menu

3.3.1 Student Grade

The student grade and marks is shown after the Registrar has click generate

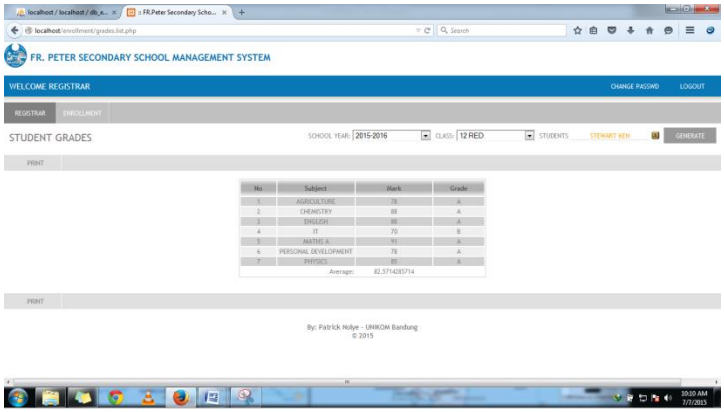


Figure 4.49 Student Grades

4. Enrollment Menu

The Registrar uses this menu to enroll those registered students only.

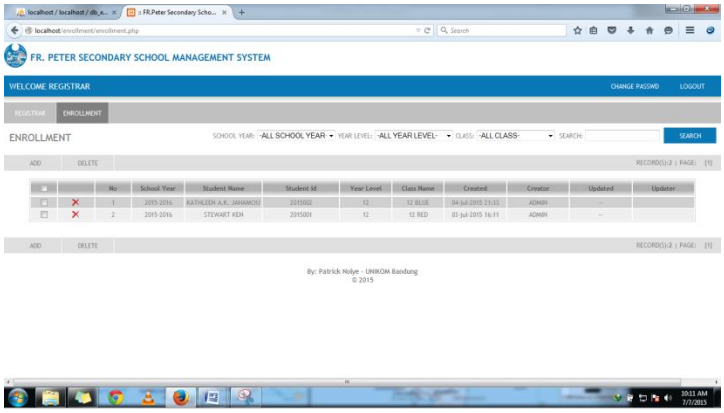


Figure 4.50 Enrollment Menu

4.1 New Student Enrollment Menu

To enroll a new student, the Registrar clicks add and the enrollment form shown below is provided. The registrar fills this form and enrolls the new student.

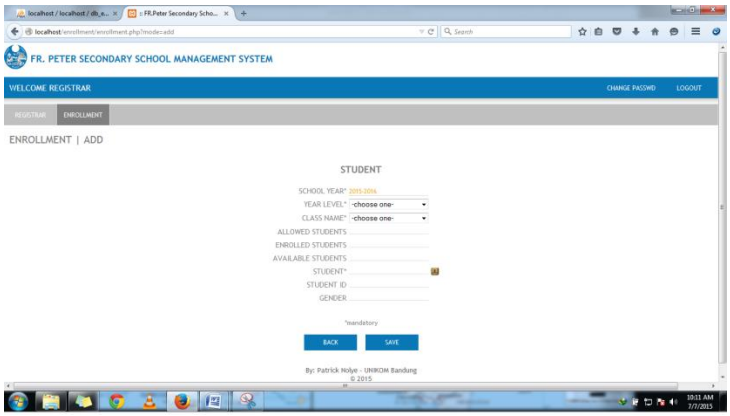


Figure 4.51 New Student Enrollment Menu

4.3.3. Following is the interface implementation of user: Cashier

1. Cashier Login Page Menu

When the Cashier opens this application, this is the login page menu that is shown. So the Cashier type in the user name and password to log in successfully.

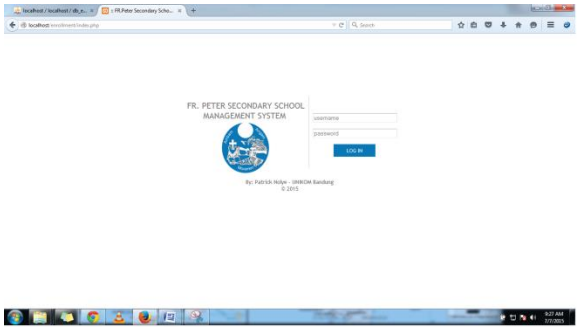


Figure 4.52 Cashier Login Page Menu

2. Cashier Home Page Menu

When the Cashier has successfully log in, the home page will directly shows the payment list if there are records of students' payments.

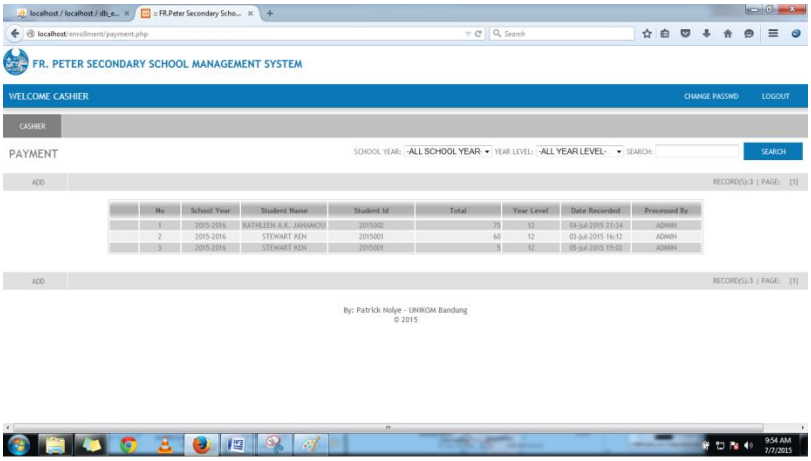


Figure 4.53 Cashier Home Page Menu

3. Cashier Sub-Menus

There are three sub-menus: Fees, Payment, Student Ledger & Account

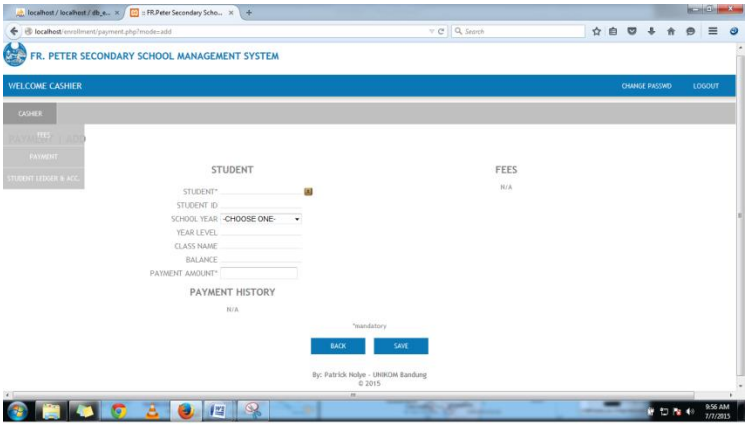


Figure 4.54 Cashier Sub-Menus

4. Fee Manu

The Cashier uses this menu to allocate different amount of school fees to the respective levels of grades and the boarding/non-boarding fees.

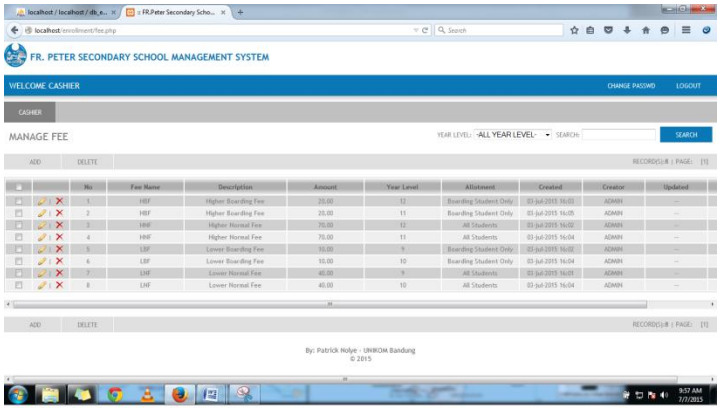


Figure 4.55 Fee Menu

4.1. Add New Fee Menu

In this menu, the Cashier adds a new fee.

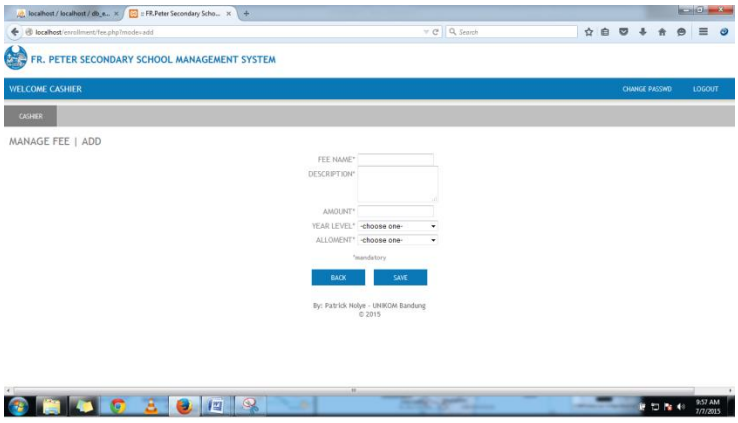


Figure 4.56 Add New Fee Menu

4.2. Payment Menu

This menu is used purposely for school fee payment of students.

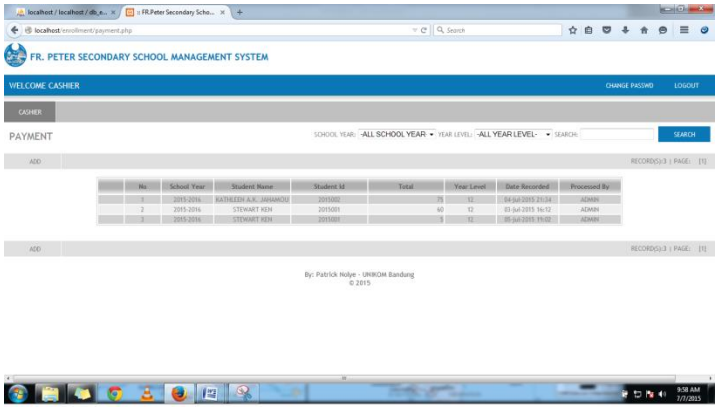


Figure 4.57 Payment Menu

4.3 New Payment Menu

The Cashier uses this menu to add new students to the payment lists by filling in the amount that was paid by that particular student.

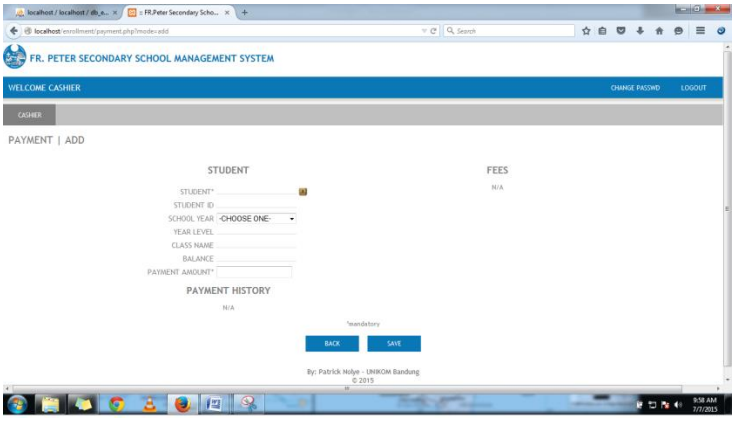


Figure 4.58 New Payment Menu

4.4 Student Ledger & Account Menu

The Cashier fills in the student name and school year so the system provides the financial record of that student

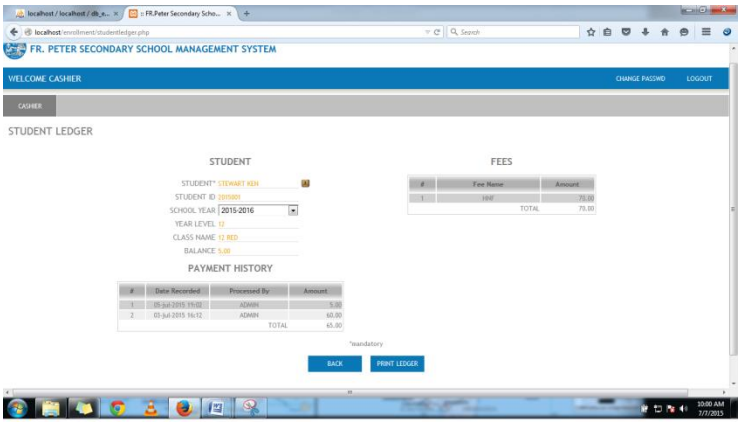


Figure 4.59 Student Ledger & Account Menu

4.5 Print the Ledger

The Cashier just clicks print and downloads the pdf file of the ledger.

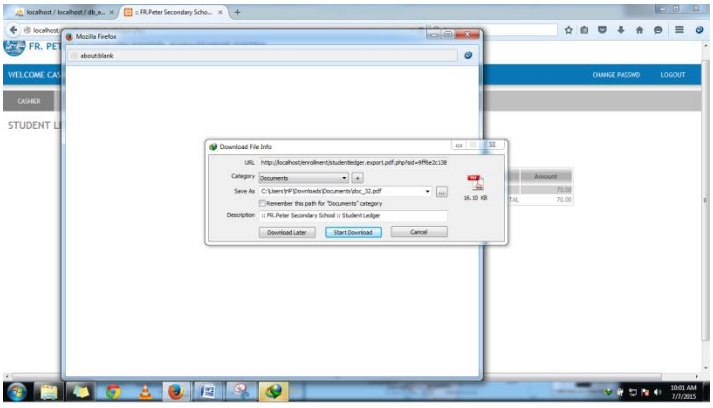


Figure 4.60 Print the Ledger

4.3.4. Following is the interface implementation of user: Student

1. Student Login Page Menu

When opening the application, the login menu is displayed. The student has to use his/her Student Id Number as user name and the Password that was given at the registration process.

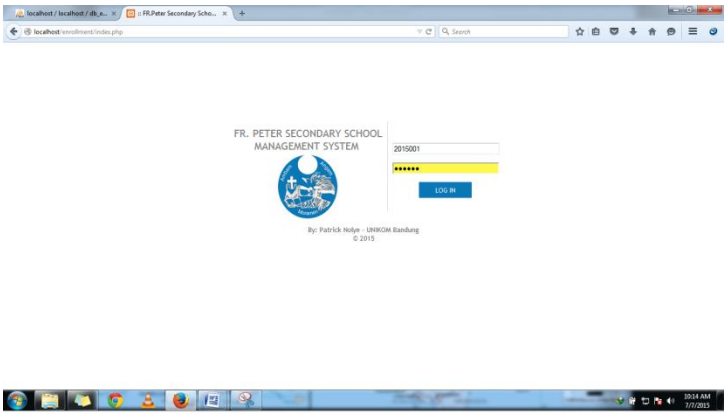


Figure 4.61 Student Login Page Menu

2. Student Main Page Menu

After the student has successfully log in, the system displays the three main menus: Time Table, Grades List and Balance. But the time table is shown directly after the log in as main page.

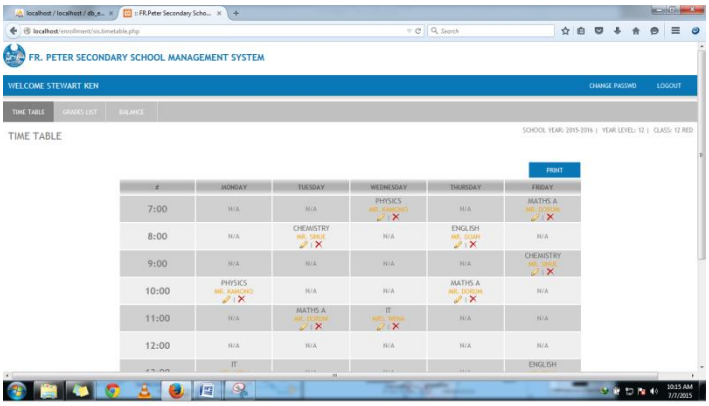


Figure 4.62 Student Main Page Menu

3. Print Timetable

Student clicks print to download the pdf file of the timetable for his/her class

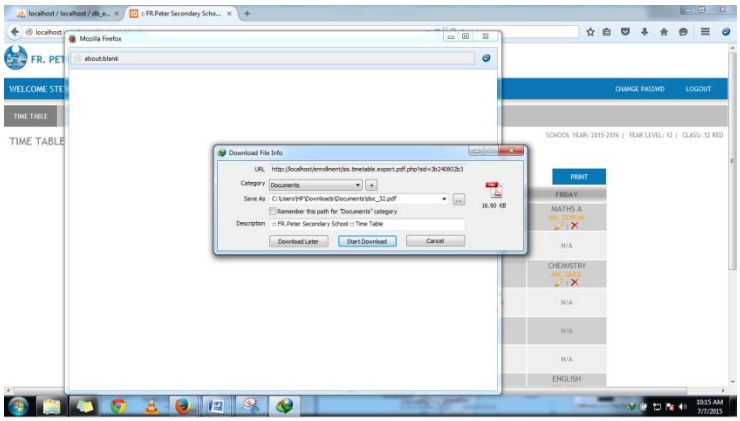


Figure 4.63 Print Timetable

4. View Student Grades Menu

Student just clicks Grades List and views his/her transcript.



Figure 4.64 View Student Grades Menu

5. Print Student Grades

The student clicks print to download the pdf transcript file.

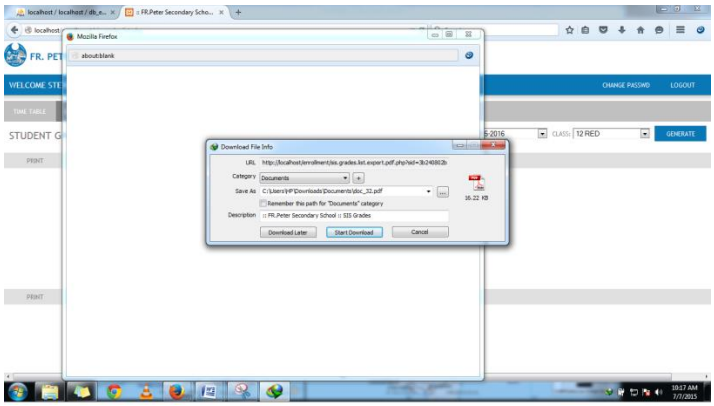


Figure 4.65 Print Student Grades

6. View Ledger

Student views the ledger by just clicking Balance.

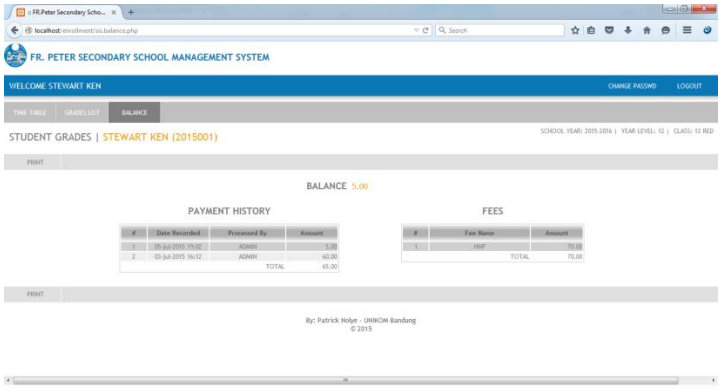


Figure 4.66 View Ledger

7. Print Ledgers

Student clicks print to download the pdf file of his/her school fee ledger.

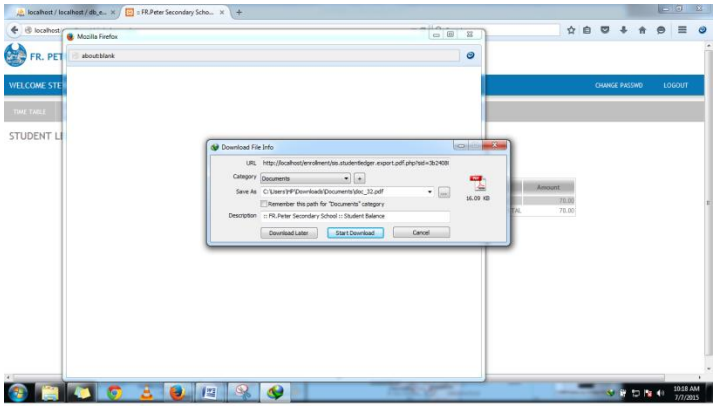


Figure 4.67 Print Ledger

4.4. Testing

In this research, the tests that were performed on the system mainly based on functionally testing. The method used in this testing is Black Box Testing which focuses on the functionality of the system.

4.4.1. Testing plan

The system test plan that will be tested is *BlackBox* testing method which is presented in the following tables below.

Here are the results of the testing of Applications of Fr. Peter Secondary School Management System -Web-Based.

Tabel 4.6 Admin Page Testing Plan

No	The Component Tested	Submenu	Test ItemScenario	Testing Results
1	Display Admin/Secretary Login Page		Displays the login form to the user Admin	Black Box
2	Administrative Menu	Year Level	Showing year level page	Black Box
			Displaying form to Add Year Level	Black Box
		Classrooms	Displaying Classroom Page	Black Box
			Displaying form to Add Classroom	Black Box
		School Year	Displays School Year Page	Black Box
			Showing form to add	Black Box

			School Year	
		Subjects	Shows Subject Page	Black Box
			Showing form to add Subject	Black Box
		Teachers	Display Teacher Page	Black Box
			Display form to add Teacher	Black Box
		Class Advisory	Shows Class Advisory page	Black Box
			Display form to add Class Advisory	Black Box
		Teacher Subjects	Showing Teacher Subject page	Black Box
			Display form to add Teacher Subject	Black Box
		Class Timetable	Showing class timetable page	Black Box
			Showing form to create class timetable	Black Box
		Boarding	Display Boarding Page	Black Box
			Display form to add Boarding	Black Box
3	User Menu		Showing User Page	Black Box
			Show form to add User	Black Box

Table 4.7 Registrar Testing Plan Page

No	The Component Tested	Submenu	Test ItemScenario	Testing Results
1	Display Registrar Login Page		Displays the login form to the user Registrar	Black Box
2	Registrar Menu	Students	Shows Student Page	Black Box
			Shows form to register Student	Black Box
		Encoding of Grades	Shows Encoding of Grade Page	Black Box
			Shows the form to allocate mark on the Subject	Black Box
		View Student Grades	Display the View Student Grade Page	Black Box
			Display the Student Grade or Transcript	Black Box
3	Enrollment Menu		Display the Enrollment page	Black Box
			Display form to enroll student	Black Box

Tabel 4.8 Cashier Testing Plan Page

NO	The Component Tested	Submenu	Test ItemScenario	Testing Results
1	Display Cashier Login Page		Displays the login form to the user Cashier	Black Box
2	Cashier Menu	Fees	Shows the Fee Page	Black Box
			Show form to add Fees	Black Box

		Payment	Display the Payment Page	Black Box
			Show form to add Payment	Black Box
		Student Ledger & Account	Display form to print Ledger	Black Box

Tabel 4.9 StudentTesting Plan Page

NO	The Component Tested	Submenu	Test ItemScenario	Testing Results
1	Display Student Login Page		Displays the login form to the user Student	Black Box
2	Student Menu	Timetable	Display Timetable	Black Box
		Grades List	Display the form to Generate marks	Black Box
			Display the Grades	Black Box
		Balance	Show the form to produce ledger	Black Box
			Display financial transactions	Black Box

4.4.2. Cases and Test Results

Here are casse to test the software that has been designed using the method which is based on the blackbox testing plans tables above.

Tabel 4.10 Cases and Test Results for Admin/Secretary section

No	Menu	Test Cases	Test ItemScenario	Expected Results	Testing Results
1	Secretary/Admin	Display user Login	Choosing a login form to the user	When opening the application, the login form will be displayed to the user	[**X**] Accepted [ ] Denied
2	Administrative Menu	Display Year Level	Selecting a Form to add Year Level	When choosing Add, the form will be displayed to add year level	[**X**] Accepted [ ] Denied
		Display Classrooms	Choosing a form to add Classrooms	When selecting the Add option, a form will be shown to add classrooms	[**X**] Accepted [ ] Denied
		Display School Year	Choosing a form to add School Year	When selecting the option Add, a form will be displayed to add school year.	[**X**] Accepted [ ] Denied
		Display Subjects	Select a form to add subject	When selecting the option Add, a form will be displayed to add subject.	[**X**] Accepted [ ] Denied
		Display Teachers	Select a form to add teachers	When selecting the option Add, a form will be displayed to add teachers	[**X**] Accepted [ ] Denied
		Display Class Advisory	Select a form to add class advisory	When selecting the option Add, a form will be displayed to add class advisory	[**X**] Accepted [ ] Denied
		Display Teacher Subjects	Selecting a form to add teacher subjects	When selecting the option Add, a form will be displayed to allocate subjects to teacher	[**X**] Accepted [ ] Denied

		Display Class Timetable	Select a form to create class timetable	When choosing School Year, Year Level and Class on the form displayed, an empty class timetable will be shown	[**X**] Accepted [ ] Denied
		Display Boarding	Select a form to add dormitories	When selecting the option Add, a form will be displayed to add dormitories	[**X**] Accepted [ ] Denied
3	User Menu	Display Users	Selecting a form to add users	When selecting the option Add, a form will be displayed to add user	[**X**] Accepted [ ] Denied

Tabel 4.11 Cases and Test Results for Registrar section

No	Menu	Test Cases	Test ItemScenario	Expected Results	Testing Results
1	Registrar	Display user Login	Choosing a login form to the user	When opening the application, a user login form will be shown	[**X**] Accepted [ ] Denied
2	Registrar Menu	Display registered Students	Selecting a form to register student	When selecting the option Add, the registration form will be displayed to register student	[**X**] Accepted [ ] Denied
		Display Encoding of Grades	Selecting a form to encode or allocate grades	When choosing School Year, Teacher, Class and Subject, a form will be displayed to add mark to the given subject	[**X**] Accepted [ ] Denied

		Display Student Grades	Selecting a form to display grades	When selecting choosing School Year, Class and Student, the student's transcript or grades will be shown	[**X**] Accepted [ ] Denied
3	Enrollment Menu	Display enrolled students	Selecting a form to enroll students	When selecting the option Add, the enrollment form will be displayed to enroll student	[**X**] Accepted [ ] Denied

Tabel 4.12 Cases and Test Results for Cashier section

No	Menu	Test Cases	Test ItemScenario	Expected Results	Testing Results
1	Cashier	Display user Login	Choosing a login form to the user	When opening the application, a user login form will be shown	[**X**] Accepted [ ] Denied
2	Cashier Menu	Display Fees	Selecting a form to add fees	When selecting the option Add, a form will be displayed to add the types of fees and their respective amounts	[**X**] Accepted [ ] Denied
		Display School Fee Payments	Selecting a form to pay school fee	When selecting the option Add, a form will be displayed to do school fee payment	[**X**] Accepted [ ] Denied
		Display Student Ledger and Accounts	Selecting a form to produce school fee ledger	When choosing Student and School Year, the financial transaction will be displayed.	[**X**] Accepted [ ] Denied

Tabel 4.13 Cases and Test Results for Student section

No	Menu	Test Cases	Test ItemScenario	Expected Results	Testing Results
1	Student	Display user Login	Choosing a login form to the user	When opening the application, a user login form will be shown	[**X**] Accepted [ ] Denied
2	Student Menu	Display Class Timetable	Selecting a menu to see and download the timetable	When selecting the timetable menu, the timetable will be displayed and can be downloaded as PDF file	[**X**] Accepted [ ] Denied
		Display Student Grade	Selecting a form to see and download transcript	When choosing School Year and Class, the transcript will be displayed and can be downloaded as PDF file	[**X**] Accepted [ ] Denied

		Display School Fee Balance	Selecting a form to produce school fee ledger	When choosing School Year, the ledger will be displayed and can be downloaded as a PDF file	[**X**] Accepted [ ] Denied

4.4.3. Conclusion of Testing Results

Based on the test results with sample test cases above, it can be concluded that this software is free of syntax errors and can functionally produce the results as expected by the different users.