UNIVERSITY OF SINDH JAMSHORO

Design and Implementation of a Comprehensive School Management System

Thesis submitted for the degree of

BS (Computer Science)

By

Jay Kumar

2K11/CSM/25

Supervisor

Dr. Hyder Ali Nizamani

2014

UNIVERSITY OF SINDH JAMSHORO

CERTIFICATE

This is to certify that **Mr. Jay Kumar** has carried out Project entitled "**Design and Implementation of a Comprehensive School Management System"** during the academic year **2014**, under the supervision of **Dr. Hyder Ali Nizamani** in partial fulfillment for the degree of **BS (Computer Science).**

Dr. Hyder Ali Nizamani Assistant Professor	Prof. Dr. Muhammad Saleem Chandio Director

ACKNOWLEDGMENT

I got this opportunity to express my deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation and helped me towards the successful completion of this project work.

First, I would like to express my sincere gratitude to my project guide **Dr. Hyder Ali Nizamani,** Assistant Professor, Institute of Mathematics and Computer Science, for his insightful advice, motivating suggestions, invaluable guidance, help and lots of moral support in successful completion of this Project and also for his constant encouragement and advice throughout my BS (Computer Science) program.

Finally my deepest gratitude goes to my parents who have given me much needed comfort, support, encouragement and inspiration for completing this project.

COPYRIGHT

DECLARATION

The content of this submission is my original work and has not been submitted, in whole or in part, for a degree at this or any other university. Nor does it contain, to the best of my knowledge and belief, any material published or written by another person, except as acknowledgment in the text.

Jay Kumar

CONTENTS

ABSTRACT

The proposed system is design and implementation of comprehensive school management system in a private school, The Guardian School, Hirabad Hyderabad. The purposed system will have certain features i.e., student record keeping, fees challan, printing certificates of student, employee record keeping, payroll and event notification via SMS.

Firstly, all requirements and documents related with system are collected. Then every activity is observed for the implementation in system. At last, developed system is given to real user for quality assurance.

CHAPTER 1: INTRODUCTION

In this chapter, we describe the purpose and need of the developed system and its features.

We develop the School Management System (ScMS) which provides the complete features that a school needs. This ScMS is superficially divided into two modules.

First module is related with the students that provides complete features from the admissions management to school leaving. ScMS provides the admissions and enrolment that include the personal information, parent information, educational background of the family, educational record of student and enrollment in a specific class of a campus. The second feature is related with managing fee payment that include generating challans and entry of paid. Another feature is monthly and yearly attendance of the students. ScMS also provides the examination management feature which includes maintaining records of different terms, class tests and printing marks sheets. It is very important for most of the schools to keep records of those students who have won position, award or trofee in competitions or events organized by the school. The ScMS provides functionality to keep record of the events held and those students who won any positions.

Second module is related with employees record management and accounts. It provides hiring of employee and record keeping of their information regarding personal, qualification, salary and allowances. ScMS provides the monthly salary distribution and maintaining accounts of the school. The main feature a school owner needs that system shall generate different statistical reports e.g., how much school is earning and strength of the students studying in different classes of a campus.

This developed system is a complete system for school management because it provides all functionality that is to needed by school. It covers many working areas which include working of administration, admissions, teachers, examinations, accountant and human resource manager.

The requirements of this ScMS are gathered from The Guardian School, Hirabad, Hyderabad. The requirements gathered from administration area of the school, they provided different files and reports. The analysis of requirements begin from management of student record with monthly fees and academic and personal information. Such information was kept in school register which was maintained on

yearly basis. The next requirement was maintaining records of current and ex-employees.

Use case modeling is used to represent the system behavior, where every part of system is designed to understand working of the system. Class diagram is used to model the relations between those objects that constitutes the system.

ScMS is developed on open source and free technologies that affects cost of the system. Interface and logical controllers are developed in Java which is very powerful, free, portable and open source programming language. MySQL database is used as database server because it is free and having feature that suitably fulfill our requirement of a database management system. Java also provide a feature to communicate with database but the ScMS is having many entities, it is very difficult to maintain them, therefore, an object relational mapping framework *hibernate* is used to maintain all entities and object accessed from database. Jasper reports is used to create and design reports of the ScMS because changing and designing report is very easy as compare to the API provided by Java.

We choose the incremental model approach to develop ScMS, hence the validation and quality assurance are done by users. The main users of ScMs are the administrator, human resource manager, principal and teacher.

CHAPTER 2: BACKGROUND

In this chapter, we describe the use case modeling, entity relation modeling, class diagram, hibernate, jasper report, tools and technologies used to develop ScMS.

1.1 Use Case Diagram

A use case is a technique to model the behavior of a system and description of actions that is perform by actors in system. An actor is an idealization of an external person, process, or thing interacting with a system. For instance in Figure 2.1, shopkeeper and customer are two actors, shopkeeper purchase products from customer (wholesaler), that purchased product is sold to customer. Shopkeeper manages credit and debit of account of customers.

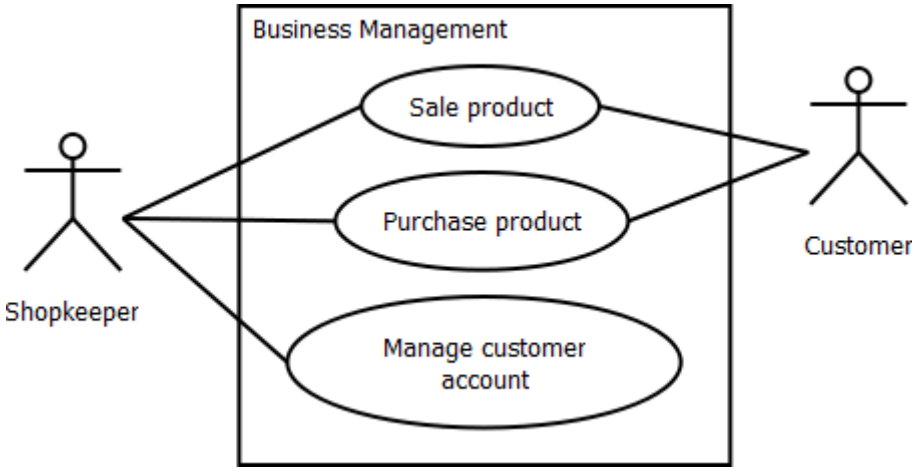


Figure 2.1: Business Management

Name: Business Management System

Actor: Shopkeeper, Customer.

Main scenario:

1. The shopkeeper purchases the product from customer.
2. The shopkeeper sale the product to customer.
3. The shopkeeper manages all customer accounts like sale, purchase and expense accounts.

1.2 Class Diagram

A class diagram describes the types of objects in the system and the various kinds of static relationships that exist among them. Class diagrams also show the properties

and operations of a class and the constraints that apply to the way objects are connected. The UML uses the term feature as a general term that covers properties and operations of a class.

1.2.1 Class

In the real world, you'll often find many individual objects all of the same kind. There may be thousands of other cars in existence. Each car was built from the same set of blueprints and therefore contains the same components e.g., in Figure 2.2 engine, doors. In object-oriented terms, we say that your car is an instance of the class of objects known as cars. A class is the blueprint from which individual objects are created. The Figure 2.2 shows the legal notation that can be used for class.

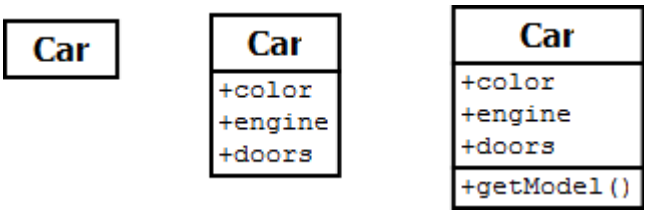


Figure 2.2: Legal Class Notations

1.2.2 Attribute

A named slot within a classifier that describes a range of values that instance of the classifier may hold. In Figure 2.2 car have attribute of color and doors.

1.2.3 Method

A method can be represented in a class diagram by placing them inside the bottom section of the class box. It is important that the method has its visibility characteristic, proper name, parentheses for parameters and return type. For instance in Figure 2.2 getModel() represents method of the Car class.

1.2.4 Association

Association is a relationship where all object have their own lifecycle and there is no owner. Let’s take an example of Teacher and Student shown in Figure 2.3.

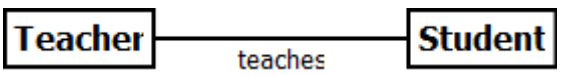


Figure 2.3: Association

Multiple students can associate with single teacher and single student can associate with multiple teachers but there is no ownership between the objects and both have their own lifecycle. Both can create and delete independently.

1.2.5 Aggregation

Aggregation is a specialize form of association where all object have their own lifecycle but there is ownership and child object cannot belongs to another parent object. Let's take an example of department and teacher shown in Figure 2.4. A single teacher cannot belong to multiple departments, but if we delete the department teacher object will not destroy. We can think about "has-a" relationship.

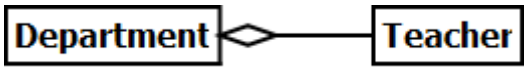


Figure 2.4: Aggregation

1.2.6 Composition

Composition is again specialize form of aggregation and we can call this as a "death" relationship. It is a strong type of Aggregation. Child object does not have their lifecycle and if parent object deletes all child objects will also be deleted. Let's take again an example of relationship between house and rooms as shown in Figure 2.5. House can contain multiple rooms there is no independent life of room and any room cannot belongs to two different house if we delete the house room will automatically delete.

Figure 2.5: Composition

1.3 Data Modeling

A technique used to analyze and model the data in organizations using an Entity Relationship Diagram (ERD).

1.3.1 Entity type

A class of entities with the same attributes.

1.3.2 ERD Development Process

Following are the main activities to create a data model

- Identify the entities.
- Determine the attributes for each entity.
- Select the primary key for each entity.
- Establish the relationships between the entities.
- Draw an entity model.
- Test the relationships and the keys.

1.3.3 Relationship

The term relationships or relationship set represents the number of entity sets associated with the relationship. An association between two or more entities that is of particular interest. Relationships are distinguished by their degree, mapping cardinality and existence.

One-to-One Relationship

A one-to-one relationship in a relational database occurs when one parent record or field has either zero or one child record only. Figure 2.6 shows that student has only one picture.

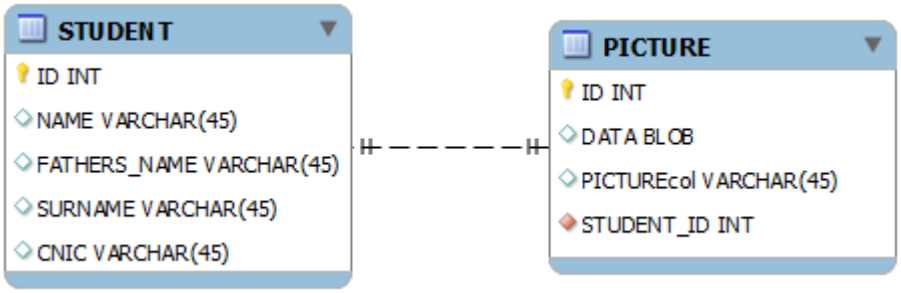


Figure 2.6: One to one relationship

One-to-Many Relationship

In relational databases, a one-to-many relationship occurs when a parent record in one table can potentially reference several child records in another table. In a one-to-many relationship, the parent is not required to have child records; therefore, the one-to-many relationship allows zero child records, a single child record or multiple child records. The important thing is that the child cannot have more than one parent record. Figure 2.7 describe that student can have one or more than one books.

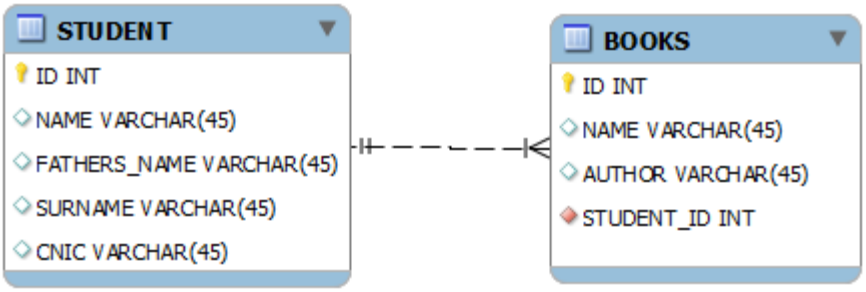


Figure 2.7: One to many relationship

1.3.4 Key Attributes

Certain attributes identify particular facts within an entity, these are known as KEY attributes. The different types of KEY attribute are:

- Primary Key
- Composite Primary Key
- Foreign Key

Primary Key: An attribute whose value can uniquely identify a complete record (one row of data) within an entity.

Composite Primary Key: A primary key that consists of two or more attribute within an entity.

Foreign Key: A copy of a primary key that exists in another entity for the purpose of forming a relationship between the entities involved

1.4 Hibernate ORM

Object relation mapping is a programming technique that creates the relation of object that reflects with entities in relational database.

Hibernate is an object relational mapping library for the Java language. It is a framework for mapping an object oriented domain model to a traditional relational database. Hibernate solves objection-relation impedance mismatch problems by replacing direct persistence related database accesses with high-level object handling functions.

1.5 Jasper Report (iReport)

Java provides printing API to generate report from application, but if a report is designed than it is very difficult to modify it because printing API report is generated by putting text pixel by pixel.

Jasper Report is an open source java reporting engine, is Java based and doesn't have its own expression syntax. As Jasper Reports is a Java class library, and is not meant for end users, but rather is targeted towards Java developers who need to add reporting capabilities to their applications. Jasper report provides a very user friendly environment to design and test report.

1.6 Tools And Technologies

In this section we describe the tools and technologies used in this project.

1.6.1 User Interface

User interface is developed in Swing Java API. We used NetBeans IDE to developing software such as IDE helps programmer to quickly develop the programs.

1.6.2 Hardware Requirements

- P-IV or faster Processor.
- Minimum 512MB hard disk space.
- 512 MB of RAM or more.
- Display capable of showing 16 bit colors or more.
- GSM device

1.6.3 Installation Requirements

- MySQL Server Version 5.5.34
- JDK 1.7
- Java Communications API
- Penguin Fonts

1.6.4 Java Communication API (JCA)

JPA provides the communication of java application with any other device e.g., GSM device, finger print reader etc. The following files are the core of JCA and they are very important to be placed at defined path in your system for a proper operation:

1. **comm.jar**

   JAVAHOME/jdk/lib

JAVAHOME/jdk/jre/lib/ext

2. **win32com.dll**

   JAVAHOME/jdk/bin

   JAVAHOME/jdk/jre/bin

   c:/windows/system32

3. **javax.comm.properties**

   JAVAHOME/jdk/lib

   JAVAHOME/jdk/jre/lib

1.7 Definitions, Acronyms, and Abbreviations

SQL: Structured Query Language, used to communicate with database.

GSM: Global System for Mobile Communication, for the communication of cell phone.

SIM: Subscriber Identity Module, that connects a cell phone with a specified network.

SMS: Short Message Service, provides sending of a message over GSM network.

AT Command: The command set consists of a series of short text strings which combine together to produce complete commands for operations such as dialing, hanging up, and changing the parameters of the connection.

CHAPTER 3: PROBLEM STATEMENT

School Management System has become the most prominent feature of school management systems in most countries around the world. In Pakistan most of the schools are not having computerized school management system because of lack of awareness. An administration of a school describes their problems which are described in following sections.

2.1 Admission, Enrollment and Fees Management

First and basic problem is admission of students and keeping all information of student like personal information, admission class etc. then, enrollment of student in class and allocation of campus. Figure 3.1 describes that administrator collects information about student and registers in system then enroll student in a class of a campus.

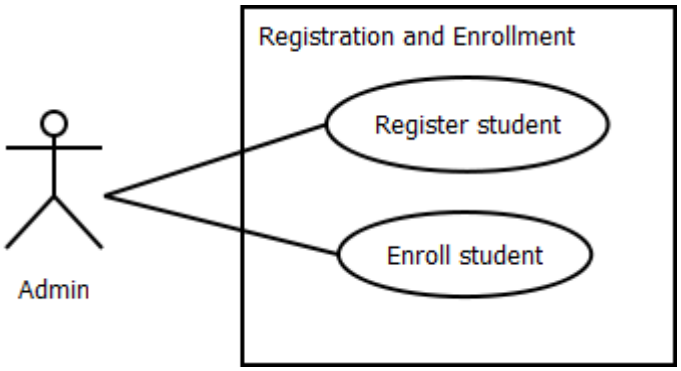


Figure 3.1: Student admission and enrollment

Name: Admission and Enrollment

Actor: Administrator

Main Scenario:

1. Student gives all information to administrator.
2. Administrator registers student information.
3. Administrator enroll student.

School contains two categories of students (refer with fees), one those who pay monthly fee and others pay of two months fee at a time. Major problem is to manage the fee of student studying in difference class of different campuses. Another software product is used by school; the task of it is only to print challan. Due to difficult to

handle huge amount of challans, system needed time to complete printing process. As you know, every school contain discounted students (student is facilitated with little discount in monthly fee) or free student who have not to pay fee, these students were also a serious problem for maintaining fee. At the end of month there is manually report of paid and defaulter students. In Figure 3.2 administrator generates challan and student collect it, then challan is paid into bank. Bank send the school copy to school, on behalf of that copy that challan is entered in system as shown in Figure 3.3.

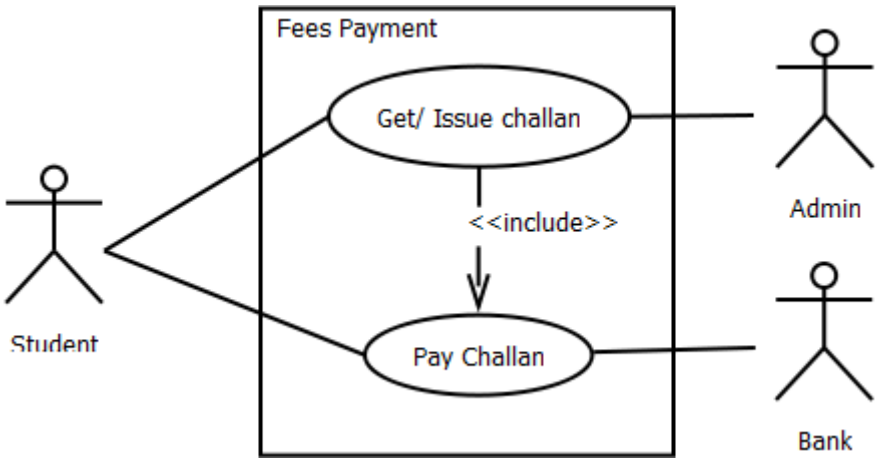


Figure 3.2: Student fees payment

Name: Fees Challan Issue and Pay

Actors: Student, Bank, School Administrator

Main Scenario:

1. School administrator generates challan of month.
2. School administrator gives challan to student.
3. Student pays challan in bank.

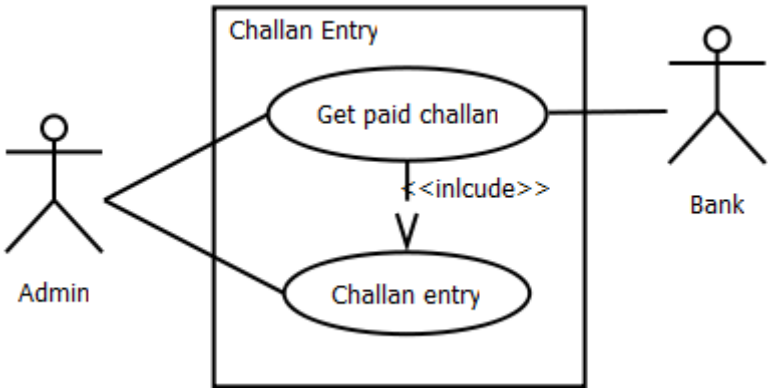


Figure 3.3: Challan entry

Name: Challan Entry

Actors: School Administrator, Bank

Main Scenario:

1. School administrator gets paid challan from bank.
2. School administrator makes entry of challan in system.
3. School administrator than make list of defaulter students who have not paid fee.

2.2 Event Notification

It is very important the parent or student should aware of monthly or yearly activities of school, that's why short message service is commonly, used which is provided by cell phone networks. As shown in Figure 3.4 there are many types of notification can be sent by administrator e.g., emergency, examination, holidays etc.

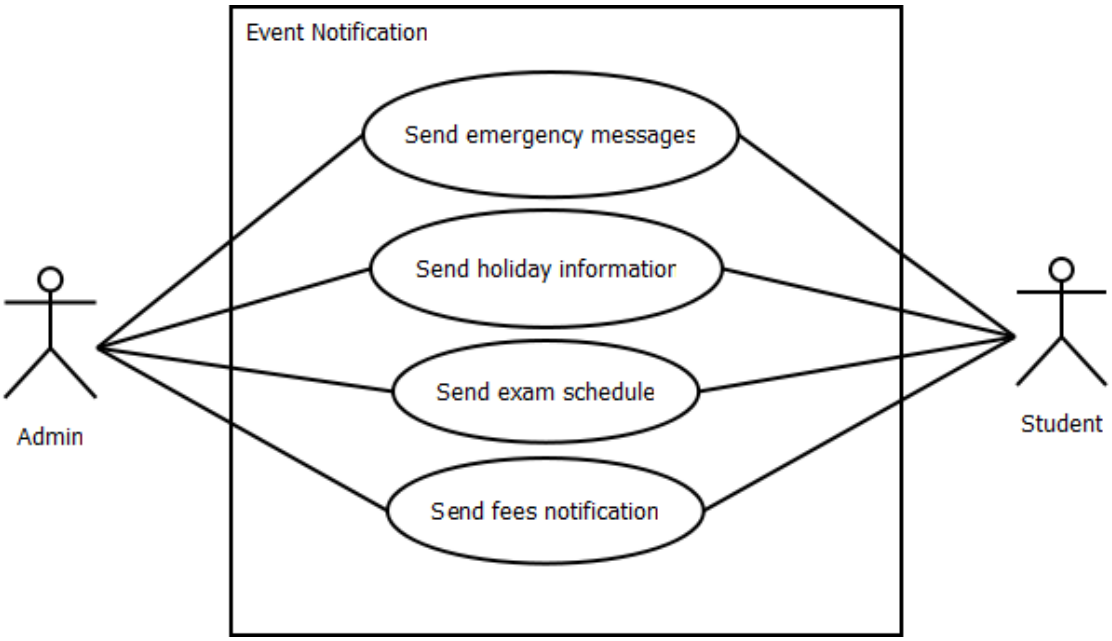


Figure 3.4: Event notification

Name: Event Notification to Student / Parent via SMS

Actors: Administrator, Student

Main Scenario:

1. The administrator send information about school activities, urgent information, holiday information, complains, parent meetings, examination schedule, fees notification and many more information through short message service to mobile number of parent or student.

2.3 Employee Record Keeping and Payroll

Another considerable problem, school have staff working at different level i.e., Administrator, teacher, clerk, pion, guard etc. Their record is to be maintained be past and present information like appointment, service time, qualification, promotion etc. Every month, it is manually made receipt of monthly increments or bonus and deductions with salary. A salary receipt (pay slip) is given with bank cheque to employee; cheque is being cashed from bank as shown in Figure 3.5. There manual system is working fine but it might also possibilities of leakage or any human error which can effect to staff salary. There is need of a system that can handle these problems.

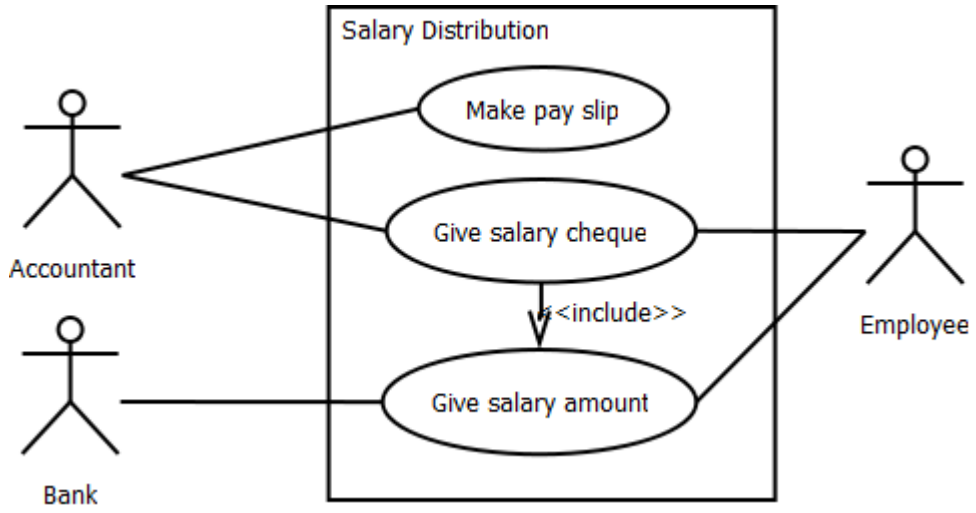


Figure 3.5: Salary distribution

Name: Salary Distribution

Actors: Accountant, Employee, Bank

Main Scenario:

1. Accountant make the monthly salary slip in which all deduction, bonus and allowances are calculated.
2. Accountant makes cheque of that amount that driven after calculation and gives it employee**.**
3. Employee gives cheque to bank and receive salary amount.

CHAPTER 4: DESIGN AND IMPLEMENTATION

In this chapter, we describe relationship between entities using class diagram and user interface of system.

3.1 Class Diagram

The Figure 4.1 shows the analysis class diagram of ScMS.

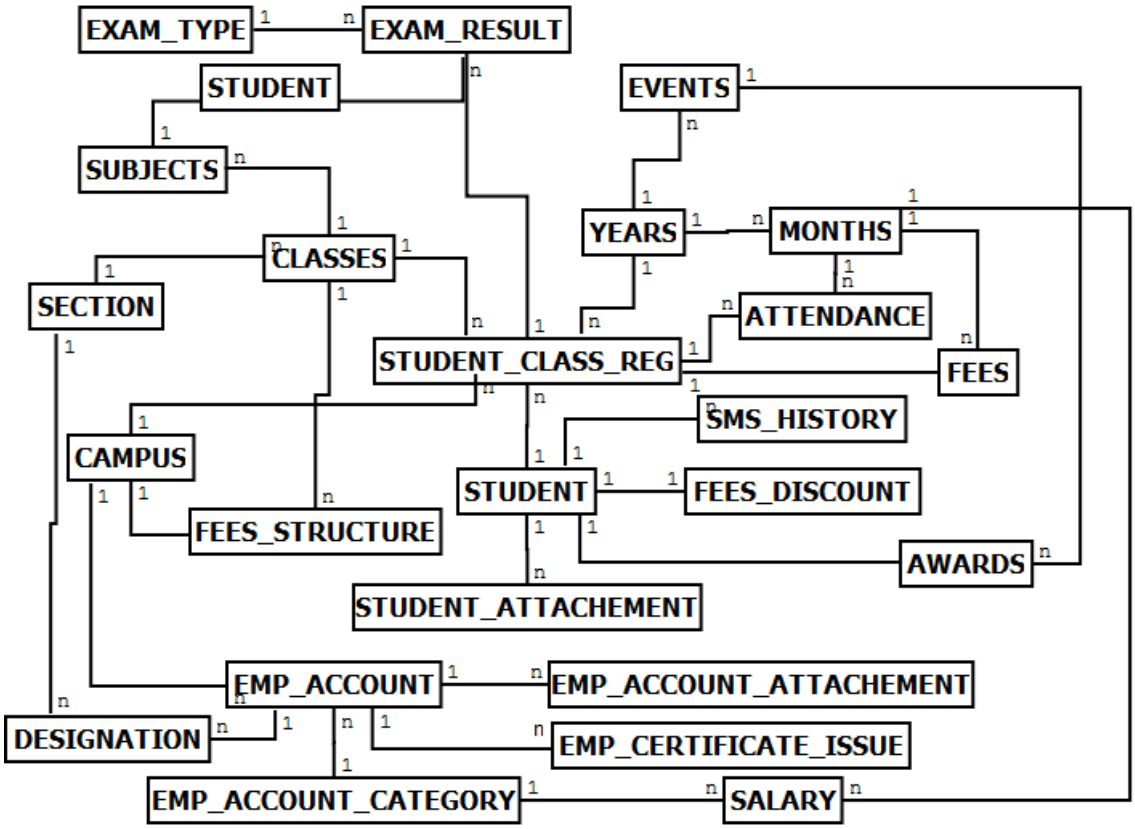


Figure 4.1: Analysis class diagram

3.1.1 Admission and Registration

A school might have many braches that is called here as **Campus**. In a campus many section are made due to different age category of students that's why every **Section** contain **Classes** as shown in Figure 4.2.

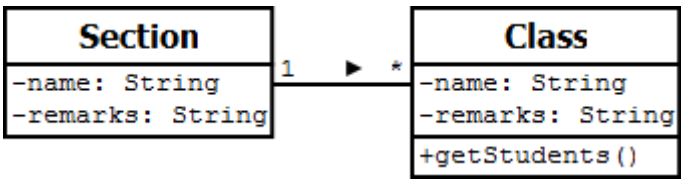


Figure 4.2: Section and class

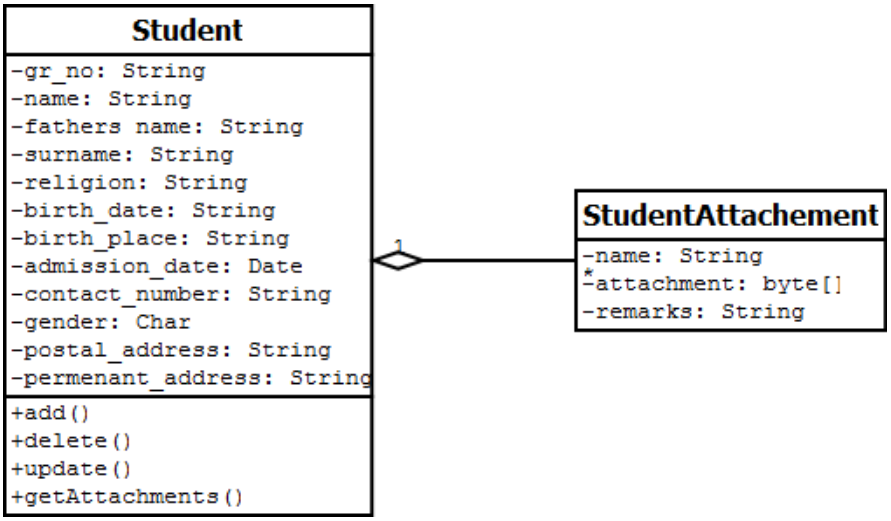


Figure 4.3: Student and attachments

Student class contains personal, parental, educational information, including picture or any other type of attachment is stored as shown in Figure 4.3. The enrollment of student is related with class of a campus and year in which student is going to attend as shown in Figure 4.4.

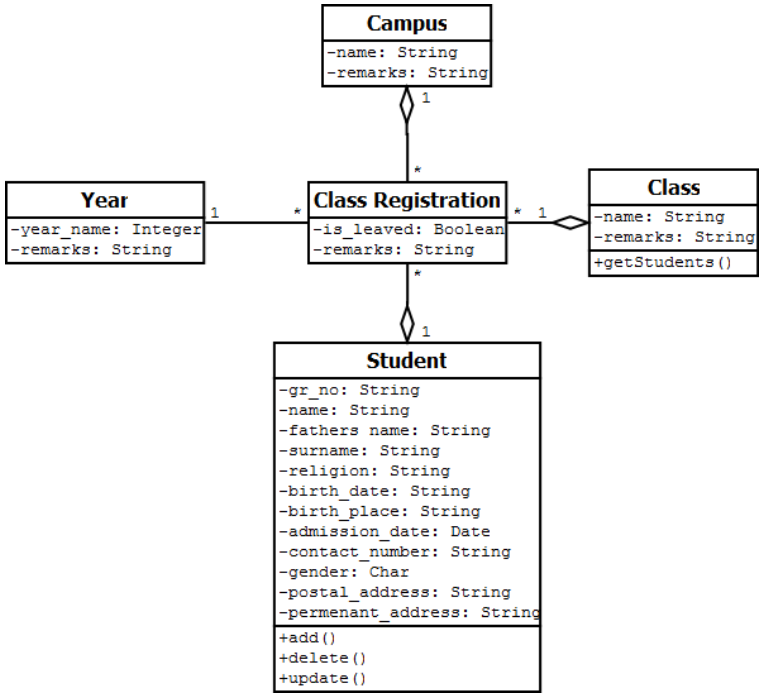


Figure 4.4: Student enrollment

3.1.2 Fees

If we talk about fees management of school there might be many questions arise like in which pattern fees structure is designed, how student pay monthly fees, how discount in fee of student is managed every month etc. Start from basic entity **Years**

having twelve **Months**, student will pay fees with a reference from enrollment in specific class and year which is **StudentClassReg** as shown in Figure 4.5.

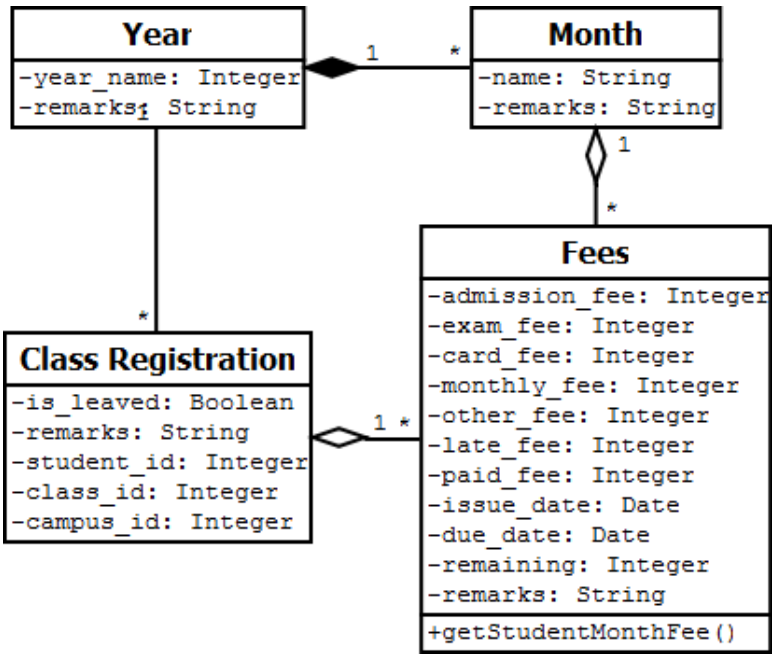


Figure 4.5: Fees generation and payment

FeesStrucure is related with fees amount of specific **Class** of **Campus** as shown in Figure 4.6.

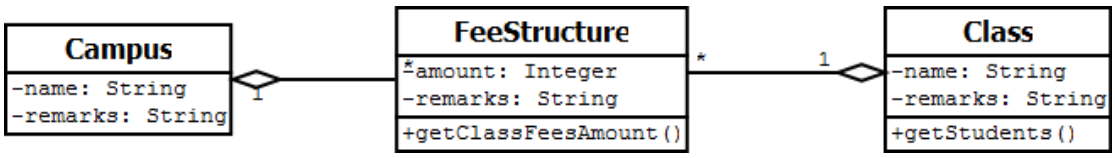


Figure 4.6: Fees Structure

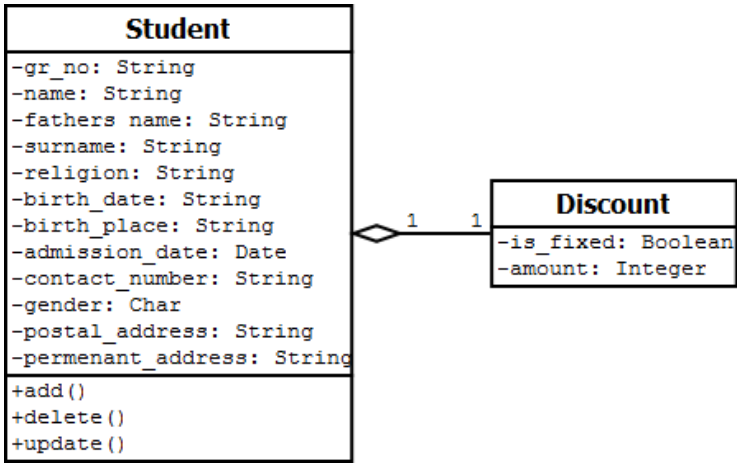


Figure 4.6: Student discount

In some cases school might face those students who are not able to pay fee as defined in fees structure. The amount that is to be deducted from their fees is stored in **Discount.**

3.1.3 Examination

As discussed above that classes are categorized by section, every class must have its own course scheme. **Subject** class stores the course scheme of every class as shown in Figure 4.8.

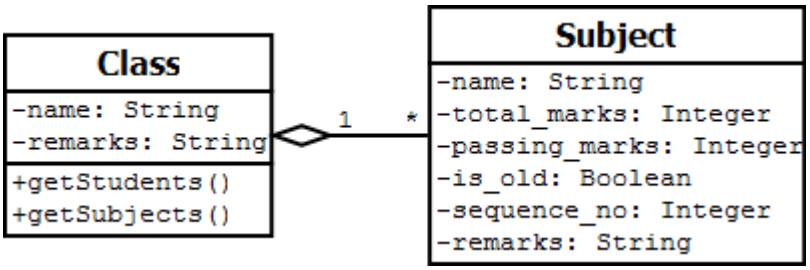


Figure 4.8: Class and subject (course scheme)

ExamType means terms or examination taken by school from enrolled student. The **ExamResult** refers the result of every enrolled student appearing in paper of examination of class.

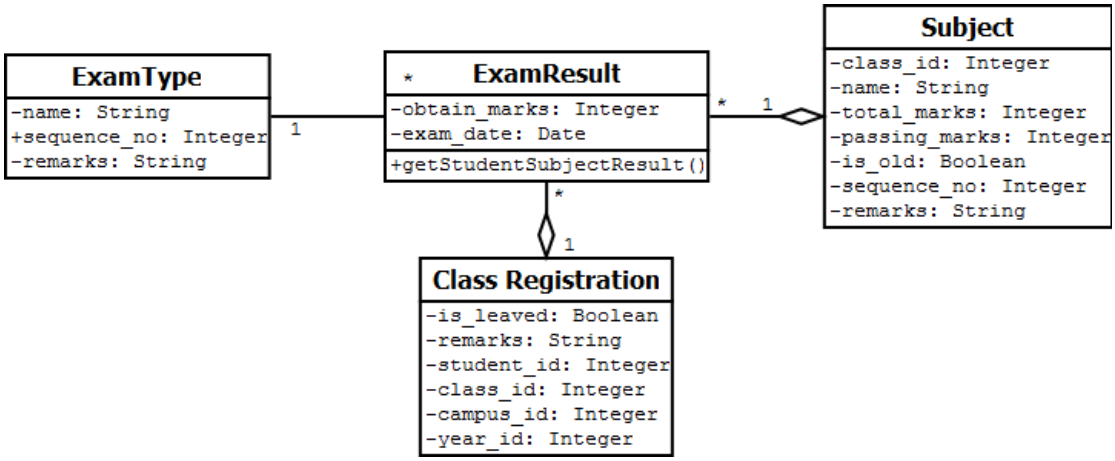


Figure 4.9: Examination result

3.1.4 Attendence

Attendence of enrolled student is taken monthly with total days school is open, present, absent and leaves of enrolled student.

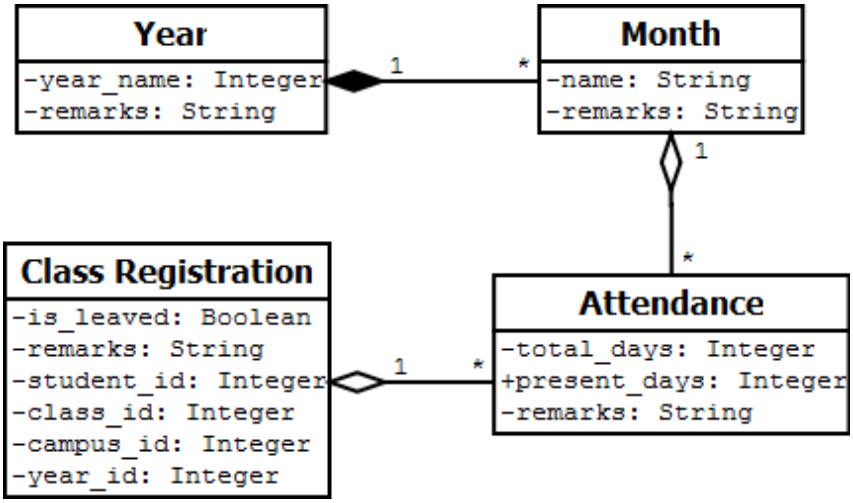


Figure 4.10: Attendence

3.1.5 Event Notification

There are many types of notification or information from school administration given to parent of student by a phone call or writing a note on school diary. Here information or any notification is sent by SMS to cell phone number of parent. Figure 4.11 shows SMS is sent to student, the message and sending date is stored in **SMSHistroy** class.

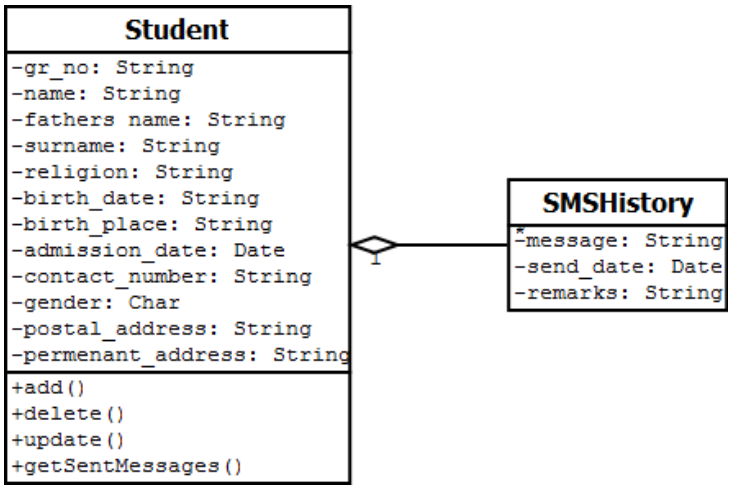


Figure 4.11: Event notification

3.1.6 Employee Record Keeping and Salary Distribution

EmpAccount contains personal and qualification information of employee. Employee is hired in any particular campus and his duty is assigned in any section where employee works at particular designation. As Figure 4.12 also shows that pictures and other documents are kept in **EmpAccountAttachment** class. The school administration issues different certificate to any employee like experience, leaving etc.

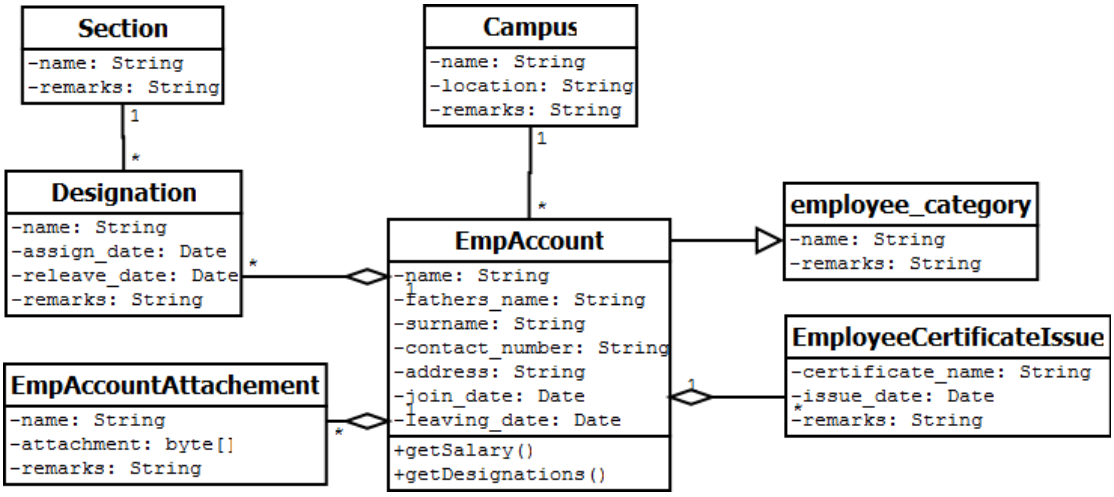


Figure 4.12: Employee record keeping

The class **EmployeeCertificateIssued** is related with certificate given to employee. EmpAccount also contains accounts like utility bills, expense account etc; that's why **EmployeeCategory** is used to categorize the employees and accounts. **Salary** is given to employee monthly including deduction and bonus amount as shown in Figure 4.13.

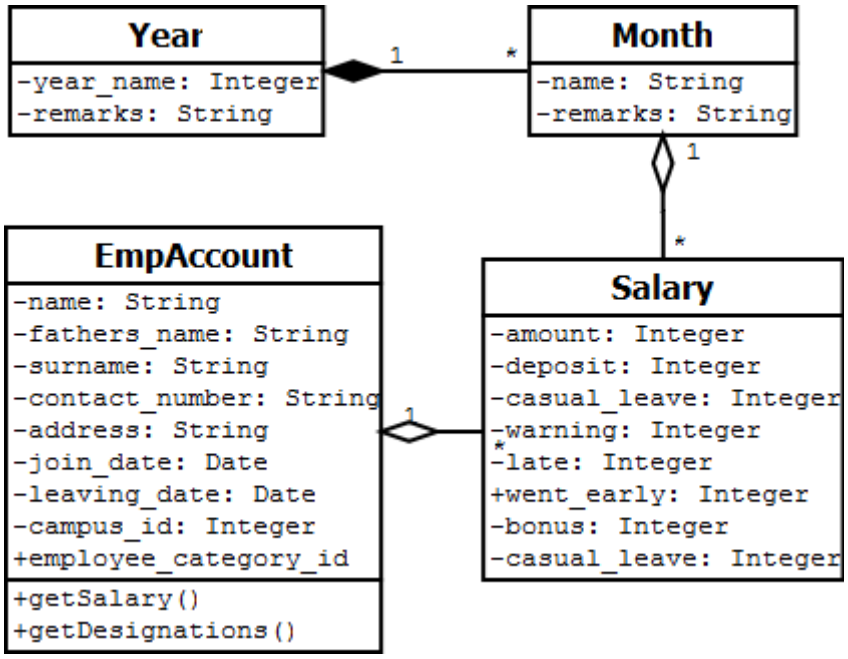


Figure 4.13: Salary distribution

3.2 SMS Sending System Architecture

The application is connected with a GSM device or mobile that accepts AT Commands. Application sends commands of sending SMS to students or staff via passing parameters of text and mobile number (stored in database) in commands.

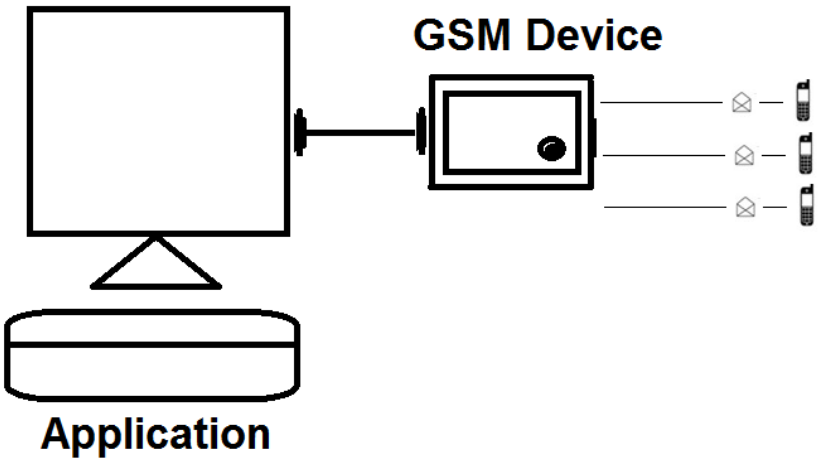


Figure 4.14: SMS sending system architecture

3.3 User Interface

The interface of application is designed according to user requirement and which is very user friendly. Below every interface is defined, how it works and what are features in it.

3.3.1 Login and Database Configuration Wizard

When application start login frame appears, as no user account is created press configuration button at bottom corner of left side, a database configuration frame appears where database driver is already defined.

Figure 4.15: Login Frame

MySQL server IP address, database server username and password has to be set then press the next button, the database server configuration will be set and database will be imported on server. The user frame will be appear to create the user of all privileges.

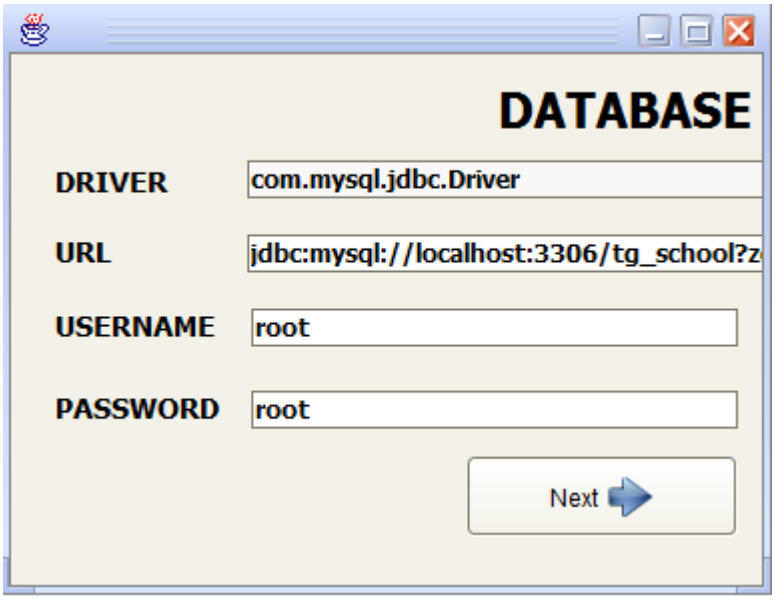


Figure 4.16: Database configuration frame

3.3.2 Users and Privileges

User accounts creation, deletion and updating can be done through this frame. Accessibility of different features can be provided to different user account that is privileges of different actions.

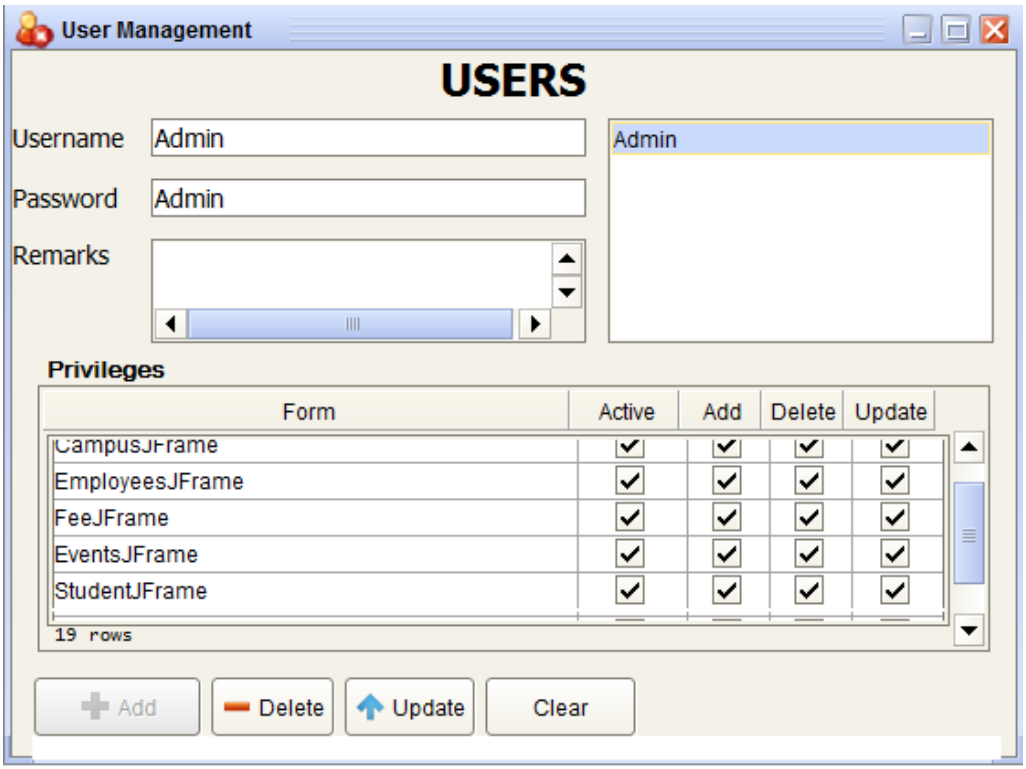


Figure 4.17: User Frame

3.3.3 Section and Class

The classes are categorized by section. This frame allow user to add, update, delete any section, class and subject. When selecting any section the class under that section appears next to list of class. Every class contains different course scheme that having different subject (course scheme) and marks. So if a class is select, subject of class appears in table. Subjects can be added and updated, however if any subject is changed in course scheme of a class, then previous subject is selected as 'old' and new subject is to be added, because previous students who were in that class (class of which course scheme is changed) in previous year had studied and passed examination in that subject, so previous record will not be changed.

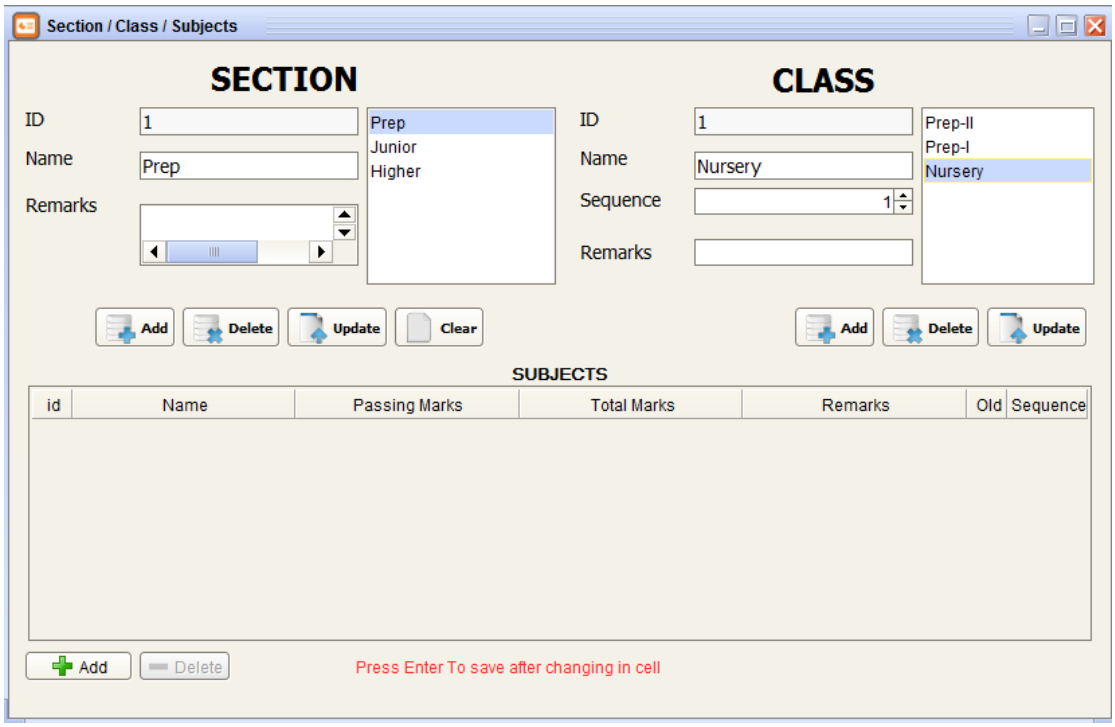


Figure 4.18: Section, class and subject frame

3.3.4 Registration and Search

The Figure 4.19 shows that registration of students needs the following fields of information of student. The student record can be deleted or updated, also generating various reports, leaving certificate, character certificate shown in Figure 4.20, fees information shown in Figure 4.21. 'Family' section contains the information about mother and father with their education and profession. 'Attachment' allow user to attach any file, certificate, picture with record of student. 'Classes' contain all class which student has been studying in school. 'Examination' has information about all

exams and class test which student has given. ‘Certificates’ show the table that which certificates school has issued to student.

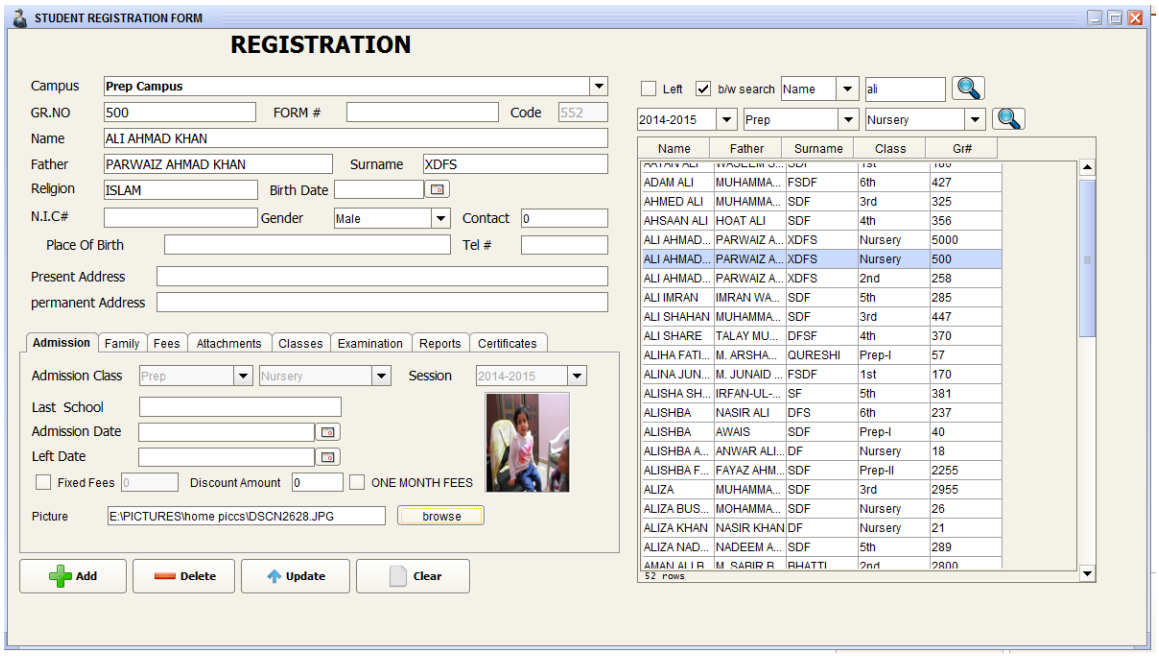
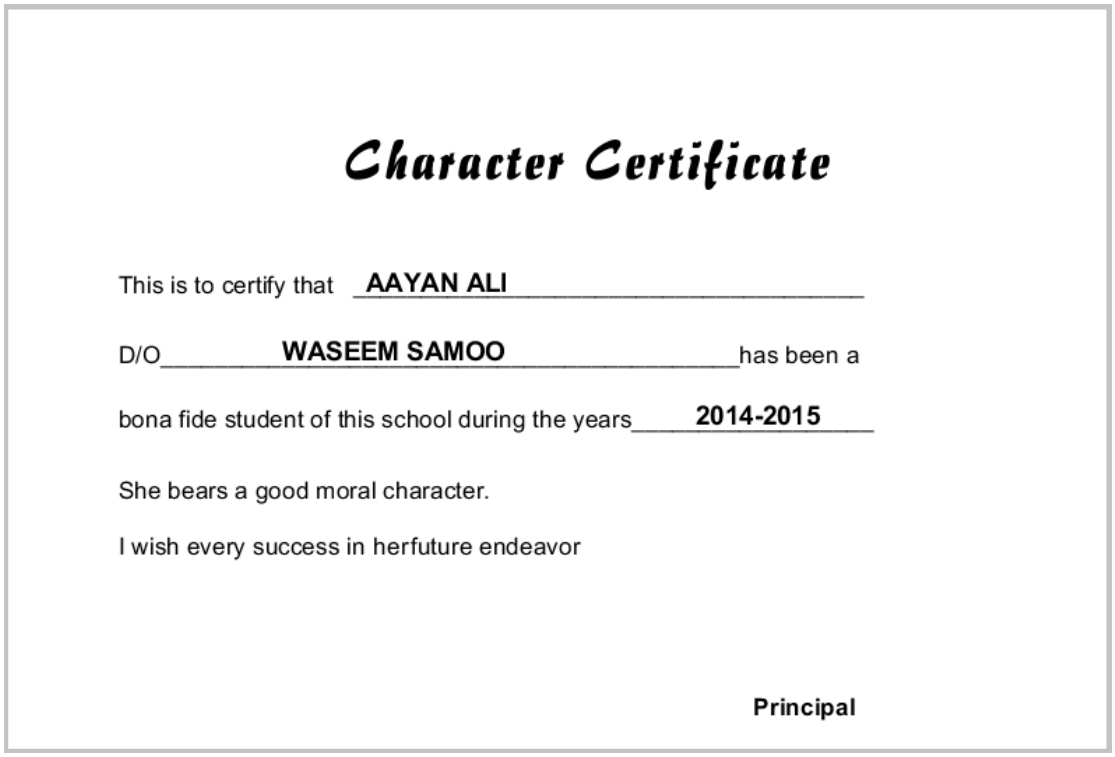


Figure 4.19: Student registration and search engine frame

At the right side, user can search student with multiple filters. Student can be search by name, father’s name, general registration number, class and session or year. ‘Between Search’ option provide user that text user type is some alphabets or whole text should match with field.

Usually it is observed that every school provide some discount or concession to some student due to special reasons, that discount is given here with record of student after that whenever monthly fees is generated of that student, the discounted amount will be deducted from fees. ‘Fixed Fee’ is option if student has paid a fixed amount that not concern with uniform fee structure of class. If zero is input of fixed fee then the student will be free; means student who has not to pay any fee to school.



Character Certificate

This is to certify that AAYAN ALI

D/O WASEEM SAMOO has been a

bona fide student of this school during the years 2014-2015

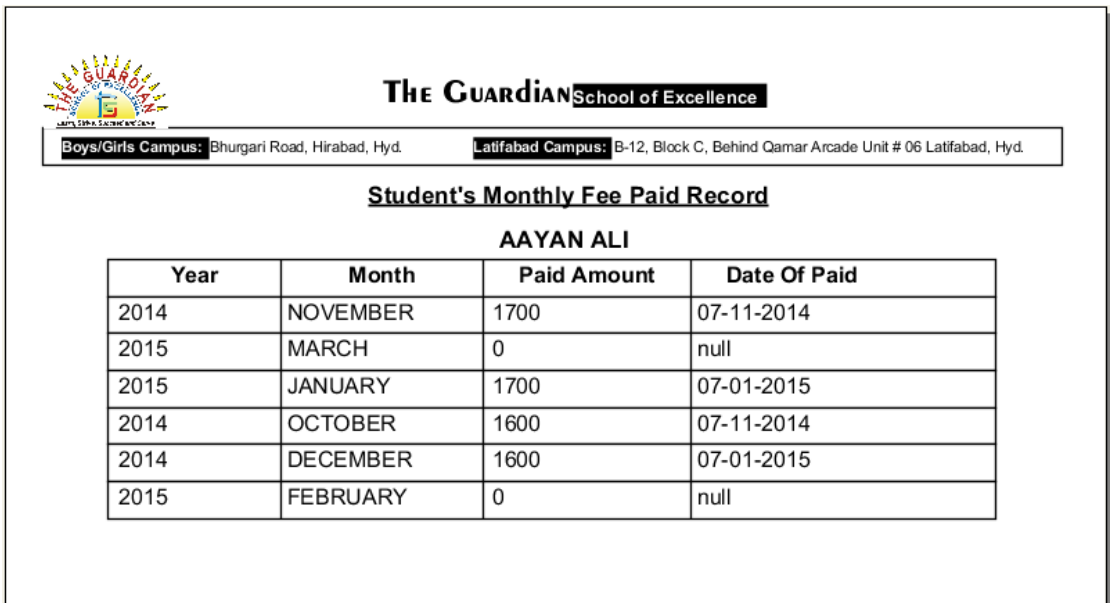
She bears a good moral character.

I wish every success in herfuture endeavor

Principal

Figure 4.20: Character certificate

The Figure 4.20 shows the character certificate that is issued by the principal or authorized user by the principal.



THE GUARDIAN School of Excellence

Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

Student's Monthly Fee Paid Record

AAYAN ALI

Year	Month	Paid Amount	Date Of Paid
2014	NOVEMBER	1700	07-11-2014
2015	MARCH	0	null
2015	JANUARY	1700	07-01-2015
2014	OCTOBER	1600	07-11-2014
2014	DECEMBER	1600	07-01-2015
2015	FEBRUARY	0	null

Figure 4.21: Fees payment history

This report in Figure 4.21 is used to investigate a single student either it is paid or not under studying period.

3.3.5 Fees Generation and Challan Payment

The Figure 4.22 shows fees generation and fees payment frame. Session and class is selected for appearing the students in table except free student; after selecting student

rows, ‘generate challan’ for generating fees of month of selected year from list. Fees will be generated according to class fee structure set by user, if any student information is set with fee deduction or fixed, the amount will be deducted or fixed amount will be adjusted with fees if ‘Apply Discount & Fixed Fee’ option is selected.

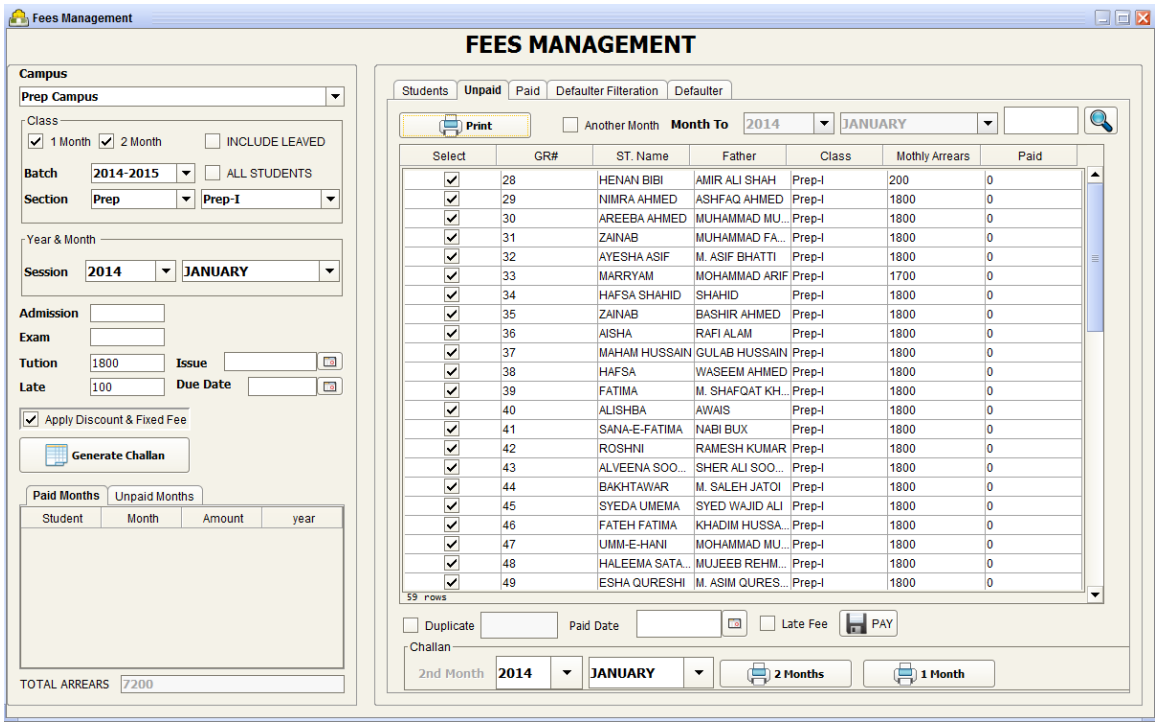
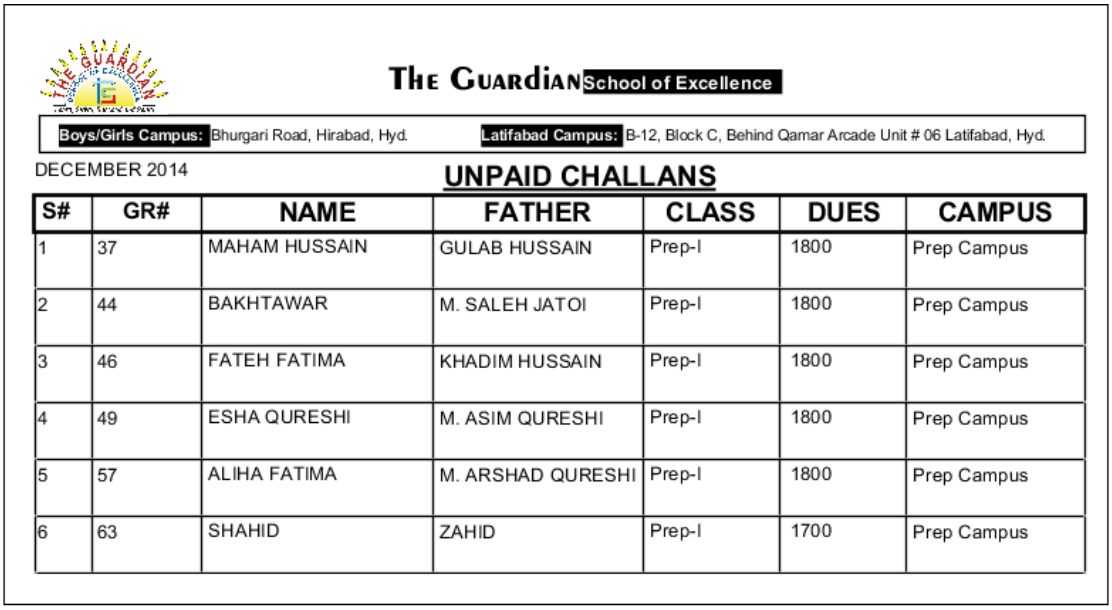


Figure 4.22: Student fees management frame

The role of issue and due date is displayed on printing of monthly challan. Student is divided into two categories according to fees payment pattern, one month student category that pays monthly fee and other are two month student category that pay fee of once in two months.

To pay generated fees will require paid date, if fees is paid after due date check the ‘late fee’ option then fee will be paid with sum of late fee penalty. Defaulter students are those who have any arrears, report of those students are also generated through this frame as shown in Figure 4.25.



The Guardian School of Excellence

Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

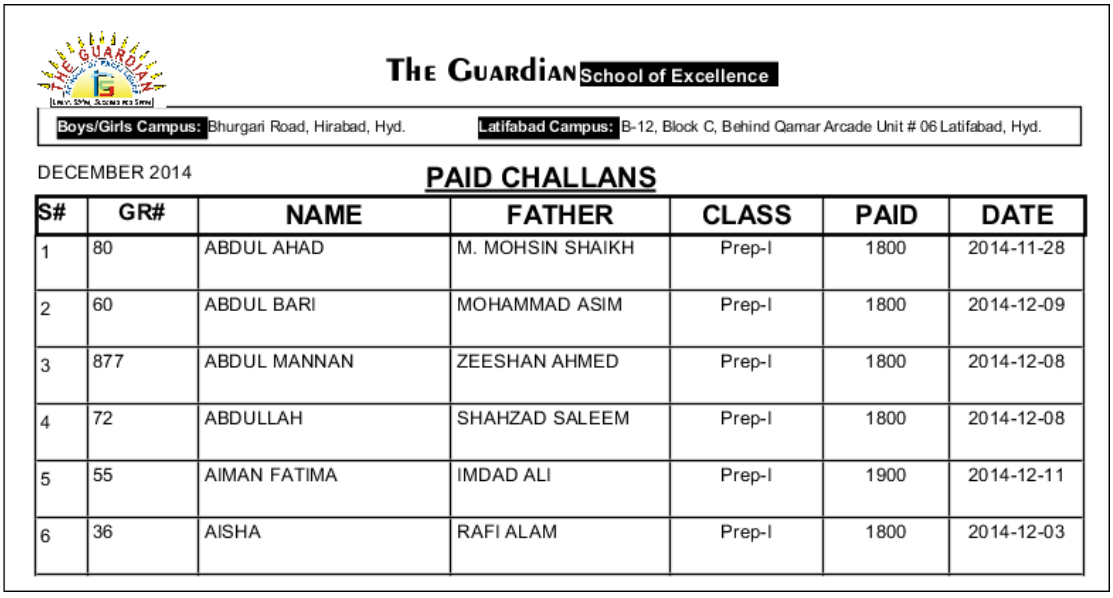
DECEMBER 2014

UNPAID CHALLANS

S#	GR#	NAME	FATHER	CLASS	DUES	CAMPUS
1	37	MAHAM HUSSAIN	GULAB HUSSAIN	Prep-I	1800	Prep Campus
2	44	BAKHTAWAR	M. SALEH JATOI	Prep-I	1800	Prep Campus
3	46	FATEH FATIMA	KHADIM HUSSAIN	Prep-I	1800	Prep Campus
4	49	ESHA QURESHI	M. ASIM QURESHI	Prep-I	1800	Prep Campus
5	57	ALIHA FATIMA	M. ARSHAD QURESHI	Prep-I	1800	Prep Campus
6	63	SHAHID	ZAHID	Prep-I	1700	Prep Campus

Figure 4.23: Unpaid challans of month report

This report in Figure 4.23 is generated every month by administrator, shows student name and class who had not paid their fees.



The Guardian School of Excellence

Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

DECEMBER 2014

PAID CHALLANS

S#	GR#	NAME	FATHER	CLASS	PAID	DATE
1	80	ABDUL AHAD	M. MOHSIN SHAIKH	Prep-I	1800	2014-11-28
2	60	ABDUL BARI	MOHAMMAD ASIM	Prep-I	1800	2014-12-09
3	877	ABDUL MANNAN	ZEESHAN AHMED	Prep-I	1800	2014-12-08
4	72	ABDULLAH	SHAHZAD SALEEM	Prep-I	1800	2014-12-08
5	55	AIMAN FATIMA	IMDAD ALI	Prep-I	1900	2014-12-11
6	36	AISHA	RAFI ALAM	Prep-I	1800	2014-12-03

Figure 4.24: Paid challans of month report

The Figure 4.24 show monthly report of students who had paid their fees. This report is generated with different filters e.g., class, section, campus etc. The Figure 4.25 shows a report of student and total dues which students have to pay.

The Guardian School of Excellence

Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

DEFAULTER STUDENTS

S#	Gr#	NAME	FATHER	CLASS	DUES	CAMPUS
1	1	AIZAZ HUSSAIN	SHAHZAD HUSSAIN	10th	4000	Girls Campus
2	2	FAISAL SULEMAN	ARSALAN	Nursery	3600	Girls Campus
3	3	MUHAMMAD SUBIYAL SHAH	ABDULLAH SHAH	Nursery	3600	Girls Campus
4	4	MOHAMMAD HUSSAIN	SAIFULLAH	Nursery	3600	Girls Campus
5	5	DUAA ABRO	M. MOOSA	10th	6000	Girls Campus
6	6	AHMED	M. AAMIR	Nursery	3600	Girls Campus

Figure 4.25: Defaulter student report

3.3.6 Attendence

The Figure 4.26 shows when select the campus, session (year) and section, class appears at list. All active student of that session under selected class appear at table with present, leave days, if user want to view inactive then 'include left' should be selected. Total days represent total days on which school is open.

STUDENT ATTENDENCE

STUDENT ATTENDENCE

Campus: Prep Campus

Session: 2014-2015 Section: Prep

Year: 2014 Month: JANUARY

Class: Nursery, Prep-I, Prep-II

INCLUDE LEFT Total Days: 22

Class	St. Name	Father	Present Days	Leave
Prep-I	ABDUL AHAD	M. MOHSIN SHAIKH	21	
Prep-I	ABDUL BARI	MOHAMMAD ASIM	12	
Prep-I	ABDUL MANNAN	ZEESHAN AHMED	20	
Prep-I	ABDULLAH	SHAHZAD SALEEM	20	
Prep-I	AIMAN FATIMA	IMDAD ALI	12	
Prep-I	AISHA	RAFI ALAM	17	
Prep-I	ALIHA FATIMA	M. ARSHAD QURE...	12	
Prep-I	ALISHBA	AWAIS	17	
Prep-I	ALVEENA SOOMRO	SHER ALI SOOMRO	20	

Save Press Enter after changing in cell

Figure 4.26: Student attendance frame

3.3.7 Examination

Examination depends on term and course scheme. Course scheme is defined when adding a class at Figure 4.18 and exam terms of class is added to database.

The Figure 4.27 shows examination of student taken by school. When selecting any class students will appear at table with subjects of that class. Marks of that subject is input to save the marks, if any student is absent in any paper then 'A' is input of field. All marks report is generated though this frame. Only those subjects will appear which are currently enabled in course scheme when selecting a class. The button 'Print Sheet' prints the ledger report of selected term of class.

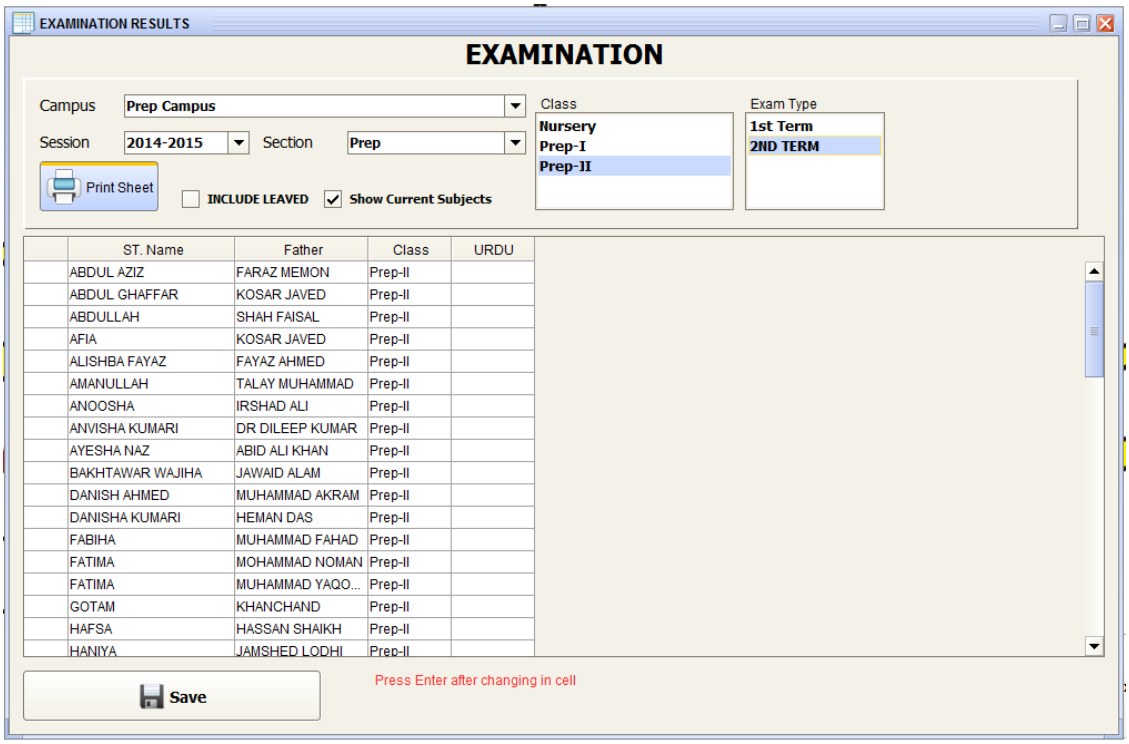


Figure 4.27: Student examination record frame

3.3.8 SMS Sending

The Figure 4.28 shows SMS (short text message) send to student. When GSM device is attached with computer, a COM Port (communication port) is assigned. Connection should be established to that port. To send message, students should be added from class to sending list. The configuration of sending should be

- baud rate 9600
- databits 8
- stopbits 1
- parity 'none'
- SMS sending delay time (in millisecond) depends on device.

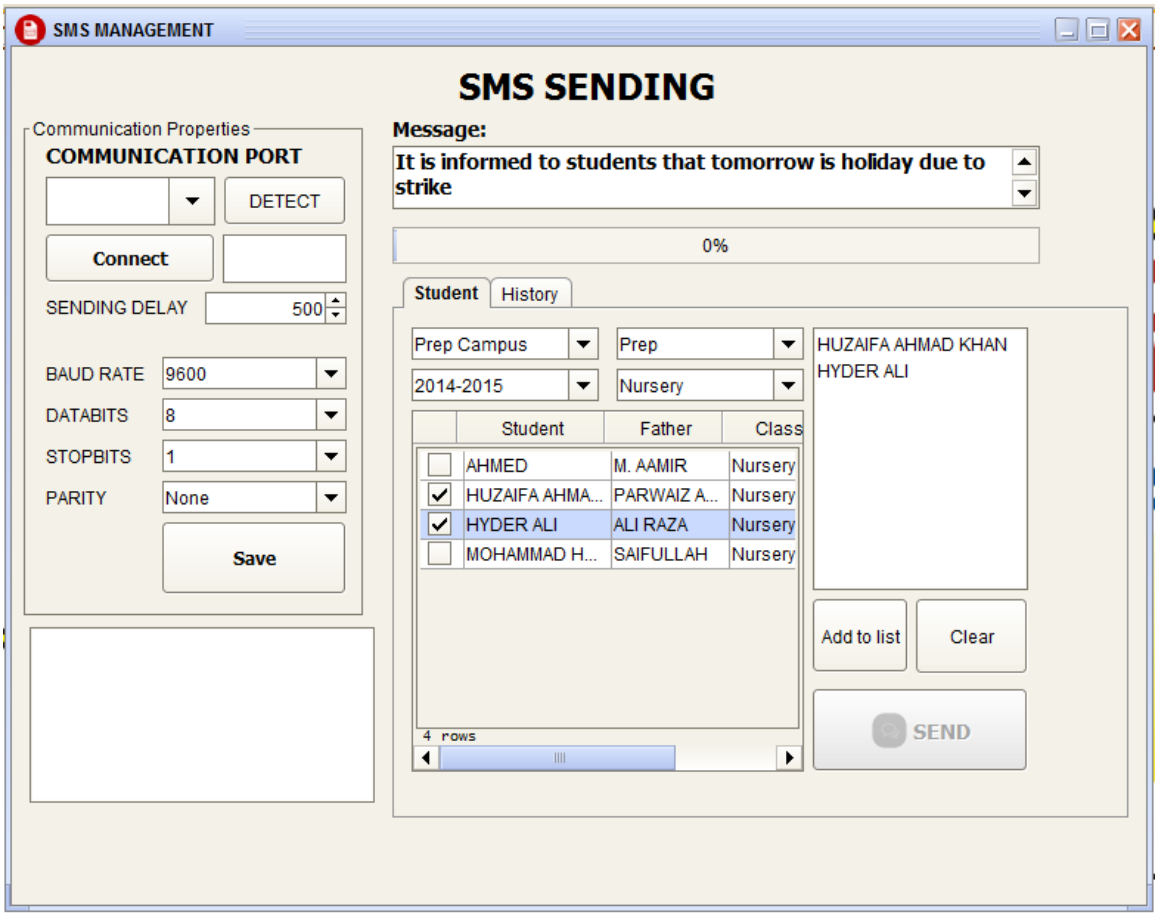


Figure 4.28: SMS sending frame

3.3.9 Statistical Chart

The Figure 4.29 shows the statistical position strength of student of different class studying at different campus and session.

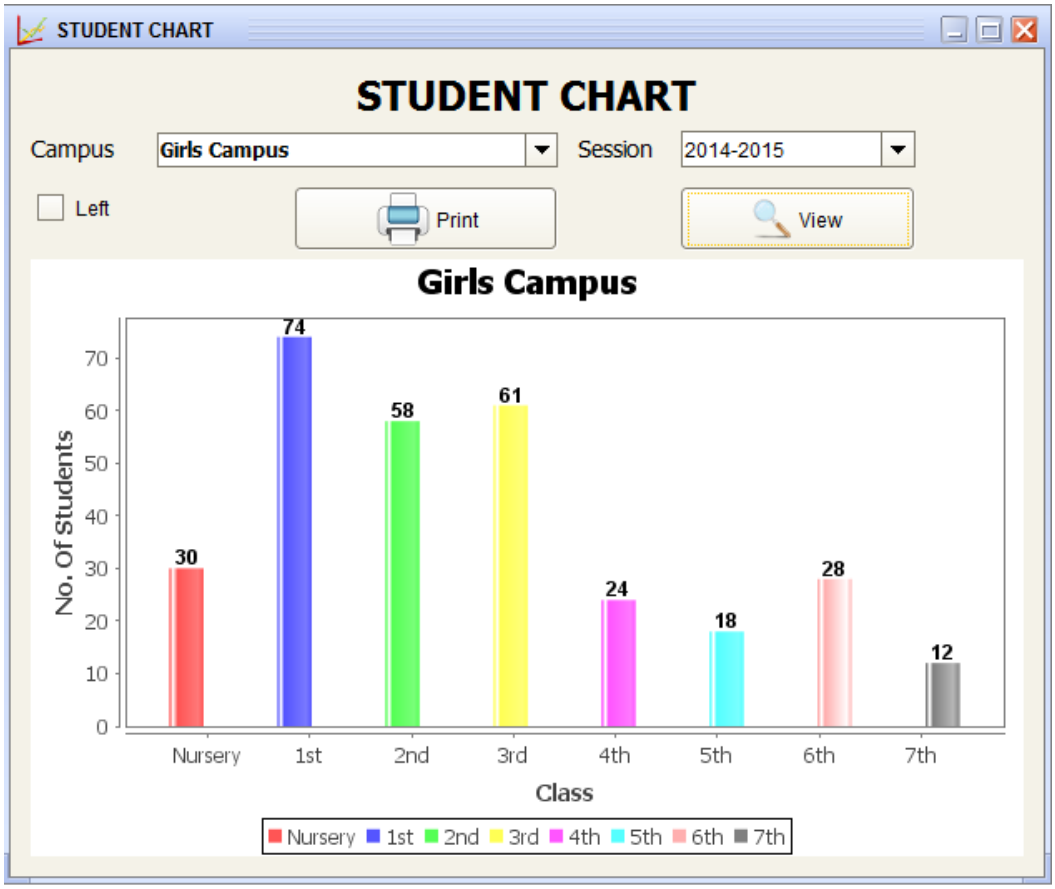


Figure 4.29: Class strength statistical chart frame

3.3.10 Employee Registration

Employee is registered according to related category (i.e., teacher, administration etc.). Personal information, picture, designation and salary information of employee is saved. The frame in Figure 4.30 shows the employee frame in which we handle the employee records and can search by any field. The ‘order button’ at right side of searching button is for custom ordering of employee, that sequential order of employee records will appear on view frames and generated reports. ‘Reports’ contains appointment letter, salary offer letter, experience letter shown in and employee information report shown in Appendix A.

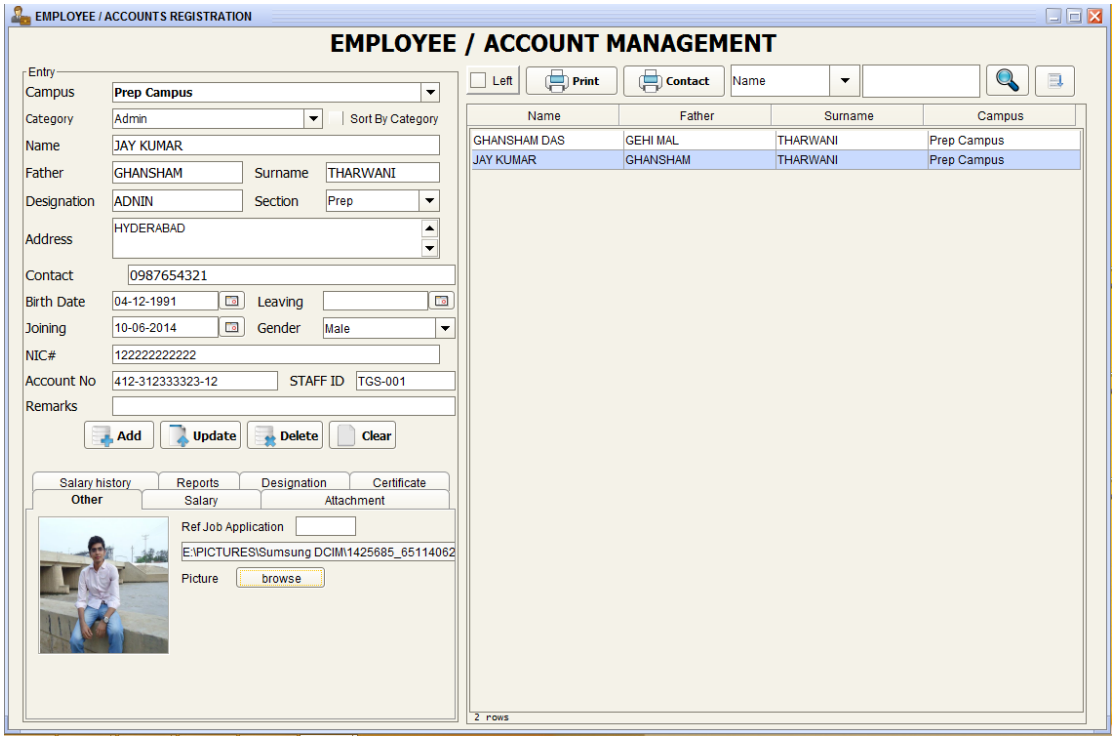


Figure 4.30: Employee registration and search engine frame

Two reports are generated by provided buttons above table, first 'print' button provide employee information of selected category and campus shown in Figure 4.31, however other 'contact' button print report of employee with their contact numbers. Employee record can be search by name, father's name, surname and staff identity number. This employee registration can also be used for keeping record of expense accounts like utility bills.

The Guardian School of Excellence

Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

Current

EMPLOYEES / ACCOUNTS

Prep Campus

S#	S.ID.	NAME	FATHER	SURNAME	JOINING	DESIGNATION	PAY	REMARKS
1	null	SUMAIRA	SHOUKAT ALI	ARAIN	null	SNR TEACHER	5750	
2	null	SABEEN	SHOUKAT ALI	MUGHAL	null	SNR TEACHER	6000	
3		ALEENA GUL	INAYAT RASOOL	QURESHI	null	SNR TEACHER	6500	
4	null	RIFAT NAZ	KHURSHEED HUSSAIN	SIDDIQUI	null	SNR TEACHER	6500	
5	null	ZAREEN ASLAM	MOHAMMAD ASLAM	ABC	null	TEACHER	6800	
6	null	IQRA JAVAID	MOHAMMAD JAVAID	SHEIKH	null	SNR TEACHER	6600	
7	null	NAZIA	ABDUL AZIZ	MEMON	null	FIXTURE TEACHER	5500	
8		KULSOOM	MUHAMMAD YAQOOB	ABC	null	SUPERVISION	2000	

Figure 4.31: Report of record of all employee and accounts

3.3.11 Salary

Salary is monthly paid to employees with deduction and bonus of extra timing. To generate pay slip select the employee or account record from table and select month of a year then press button ‘pay’. When you select any month, employees and accounts with paid amount is shown in table as you can see in Figure 4.32. ‘Determined Salary’ option is used when we are entering monthly amount of any account. There is report available for bank about salary amount and pay slip, on the basis of that report bank will transfer defined amount to employees’ account as shown in Figure 4.33.

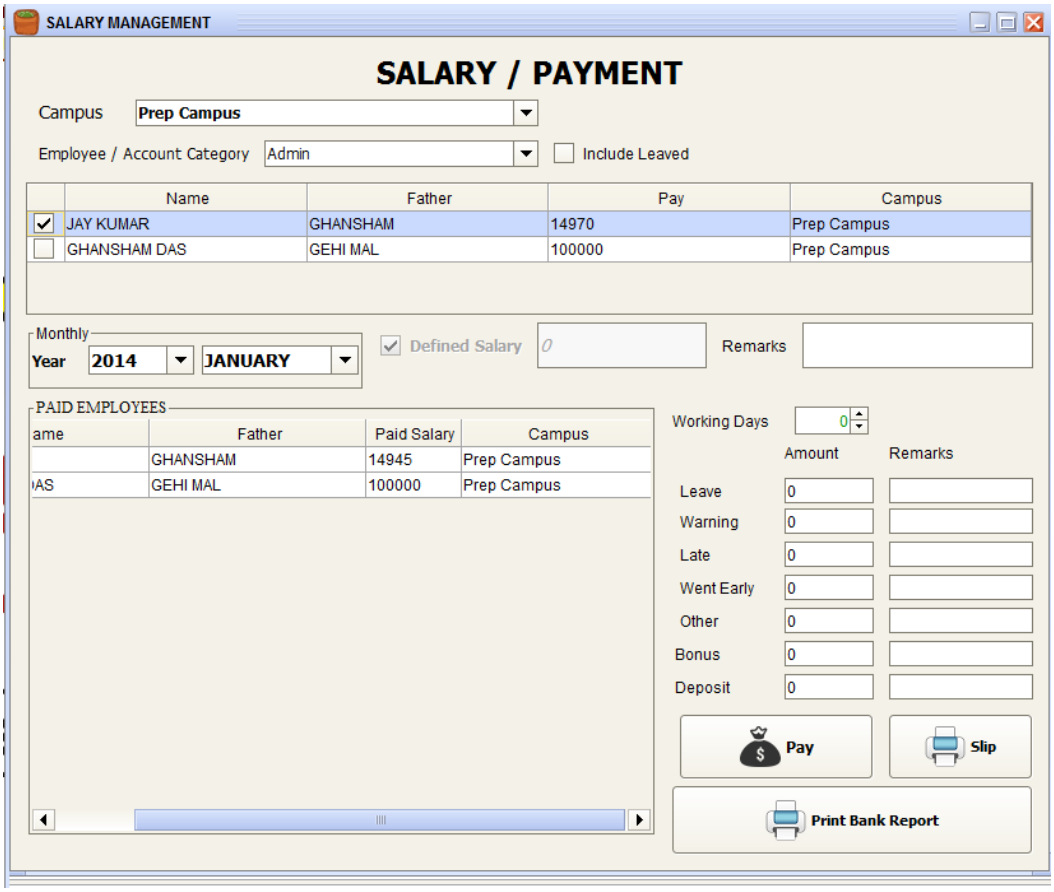


Figure 4.32: Employee payroll frame

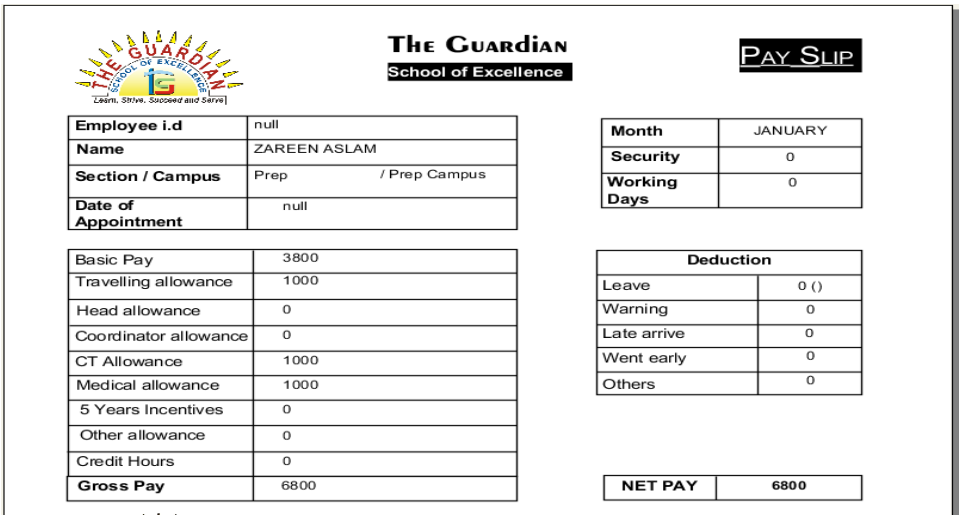


Figure 4.33: Pay slip

3.3.12 Reports

Different types of reports are available. 'Print' button generated report of selected campus, session and class, if option 'one month' is selected then record of student appear on report who pay monthly fee and vice versa with 'two month' option. If 'discounted / fixed' option is selected then discounted student record appears. 'Left' is for those who left the school.

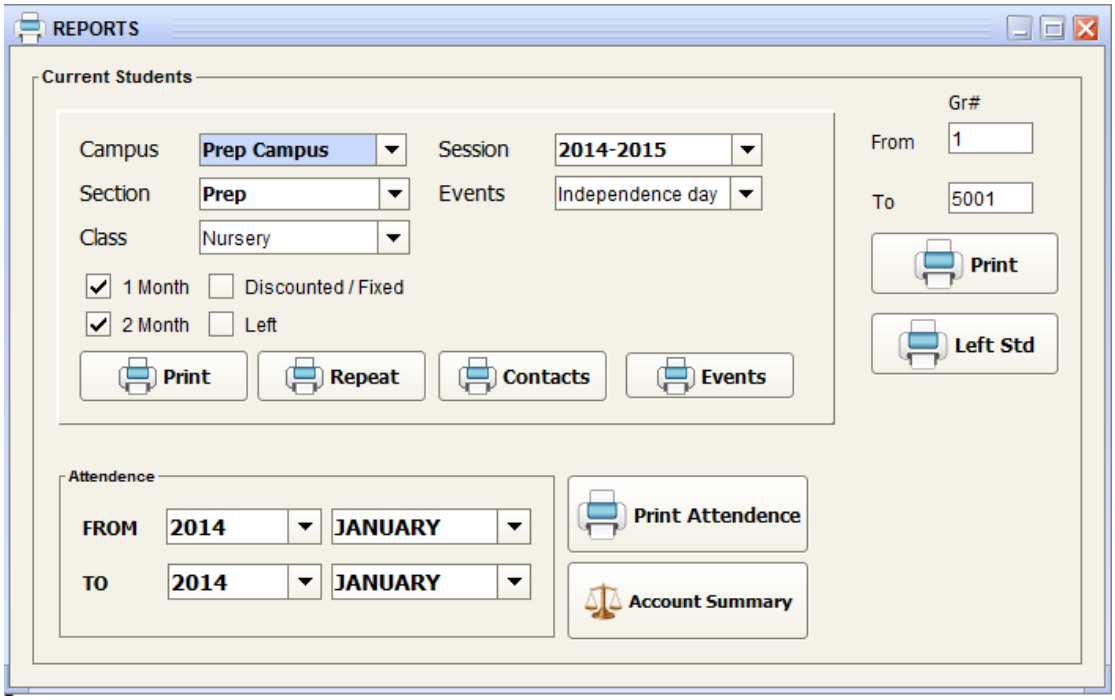
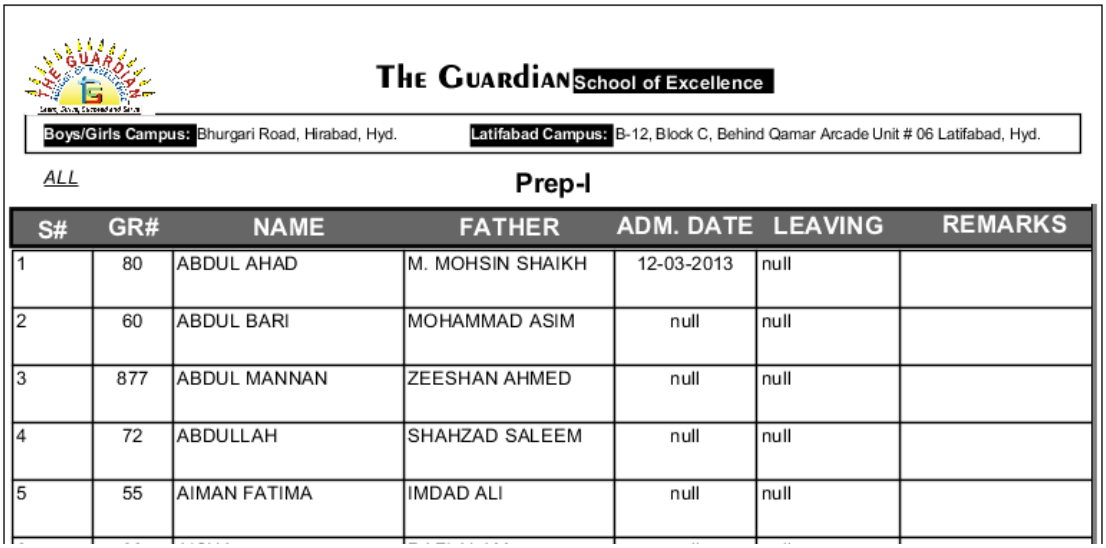


Figure 4.34: Repots frame

'Repeat' generate report of those student who are repeating in selected session. 'Contact' report of student's name, father's name, class and contact number. At the

right side ‘from’ caption shows which first general registration number (GR.No) and ‘to’ show last general registration number.



The Guardian School of Excellence

Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

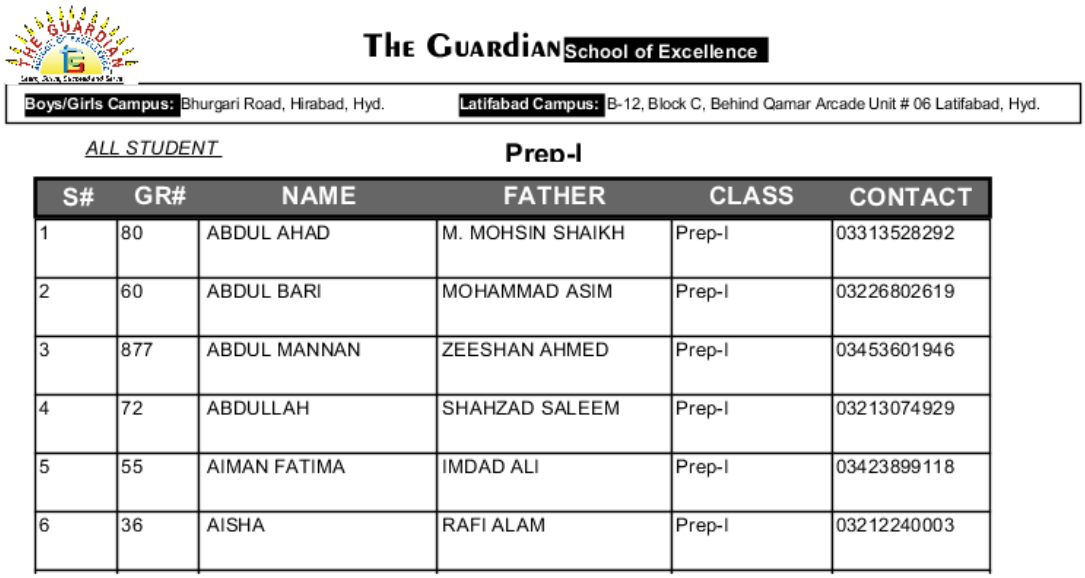
ALL

Prep-I

S#	GR#	NAME	FATHER	ADM. DATE	LEAVING	REMARKS
1	80	ABDUL AHAD	M. MOHSIN SHAIKH	12-03-2013	null	
2	60	ABDUL BARI	MOHAMMAD ASIM	null	null	
3	877	ABDUL MANNAN	ZEESHAN AHMED	null	null	
4	72	ABDULLAH	SHAHZAD SALEEM	null	null	
5	55	AIMAN FATIMA	IMDAD ALI	null	null	

Figure 4.35: Class student report

‘Print Attendance’ generates report of attendance of student of selected class from month to month. ‘Account summary’ shows how much students have paid fees, salary given to employees, expenses have spent.



The Guardian School of Excellence

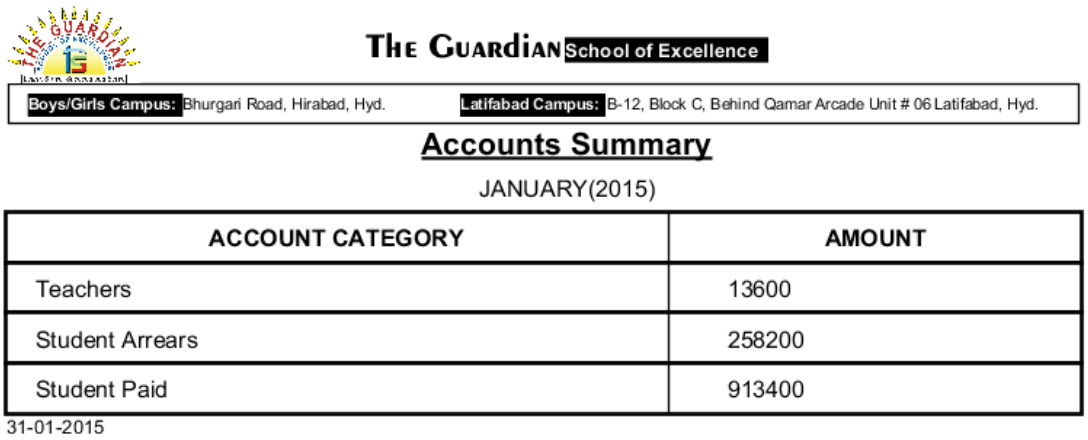
Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

ALL STUDENT

Prep-I

S#	GR#	NAME	FATHER	CLASS	CONTACT
1	80	ABDUL AHAD	M. MOHSIN SHAIKH	Prep-I	03313528292
2	60	ABDUL BARI	MOHAMMAD ASIM	Prep-I	03226802619
3	877	ABDUL MANNAN	ZEESHAN AHMED	Prep-I	03453601946
4	72	ABDULLAH	SHAHZAD SALEEM	Prep-I	03213074929
5	55	AIMAN FATIMA	IMDAD ALI	Prep-I	03423899118
6	36	AISHA	RAFI ALAM	Prep-I	03212240003

Figure 4.36: Contact number report



THE GUARDIAN School of Excellence

Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

Accounts Summary

JANUARY(2015)

ACCOUNT CATEGORY	AMOUNT
Teachers	13600
Student Arrears	258200
Student Paid	913400

31-01-2015

Figure 4.37: Account summary report

Further user interface and reports are shown in appendix A where details and role of interfaces are defined there.

CHAPTER 5: SUMMARY AND FUTURE WORK

In this chapter, we summarize our work and describe the implementation of future work.

5.1 Summary

We developed a complete featured system for a school. It can manage students' records with maintaining monthly fee that includes generating and printing monthly challan. Monthly attendance of the students is managed with present, absent and leave days. Examination terms and marks of every subject in which student appears are also supported. ScMs is provides record keeping of students who participate or won any competition organized by the school. ScMS contains feature of keeping employee information including personal and professional information, promotion and maintain employee accounts with monthly salary distribution. ScMS provide a notification system to send any type of information to parents or students via SMS.

5.2 Future Work

In the future, time tabling support will be integrated in which class and examination time table will be managed. A school calendar will also be included because there are many celebration days and holidays at which there might be any type of activity in school. In this regard, calendar will help to remind such events. The current system only shows statistical bar chart of strength of the students, but in future there will be different charts of strength of students, fees recovery, monthly attendance of employees and annually expenses of school. Daily attendance of employee will be maintained with biometric (finger print) that includes the entry and exit timing. At the end of month total late time will be calculated.

BIBLIOGRAPHY

[1] Java JDK 1.7, URL: http://java.com/en/download/index.jsp. 12 Dec. 2014

[2] Jasper Report, URL: http://sourceforge.net/projects/ireport/files/iReport/iReport-5.1. 12 Dec. 2014

[3] MySQL, URL: http://www.mysql.com/ , 12 Dec. 2014

[4] Netbeans IDE, URL: https://netbeans.org/downloads/ , 12 Dec. 2014

[5] Hibernate ORM, URL: http://hibernate.org/ , 12 Dec. 2014

[6] Java Communication API, URL: http://www.oracle.com/technetwork/java/index-jsp-141752.html , 12 Dec. 2014

[7] Fowler, Martin. UML distilled: a brief guide to the standard object modeling language. Addison-Wesley Professional, 2004.

[8] Coronel, Carlos, Steven Morris, and Peter Rob. Database systems: design, implementation, and management. Cengage Learning, 2009.

APPENDIX A: SNAPSHOTS

In this section, we will discuss those snapshots of application which helps to set initial setup data for school.

Application Interface

The Figure A.1 shows frame of school configuration, in which we can set name, contact, email and logo of school, which affects the title of application main form. As you know there are schools having different grading system in examination, so when exam sheet report is generated the grade will appear there. Every school has at least one campus where there must be on going classes, so campus can be added by its name and location as shown in Figure A.2.

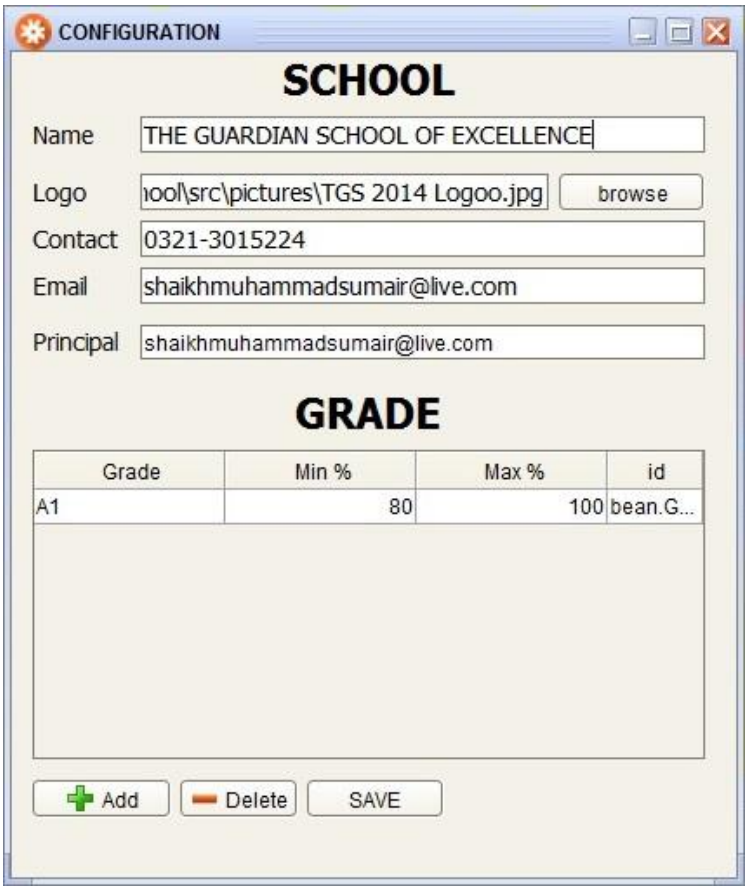


Figure A.1: School configuration frame

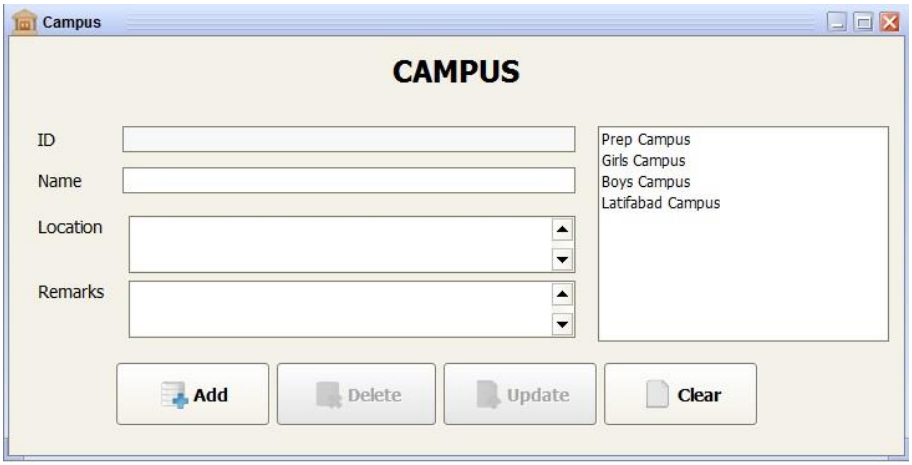


Figure A.2: Campus frame

As we discussed in chapter 4, students are enrolled, fees and salary are maintained with yearly; so whenever a year starts, a year is been added as shown in Figure A.3. It is essential to categorize existing staff so that account and activity regarding employee having same designation can be parallel maintained as shown in Figure A.4.

Figure A.3: Year or session frame

Figure A.5 shows frame of adding examination terms or types taken by school.

In school, student leaves the school in two situations, first student leaves during classes or without completing examination of current class and second vice versa. So when we set leaving date in student frame shown in Figure 4.19, the record of that student will be not shown in fees generation frame and also that student will not be shift to next class.

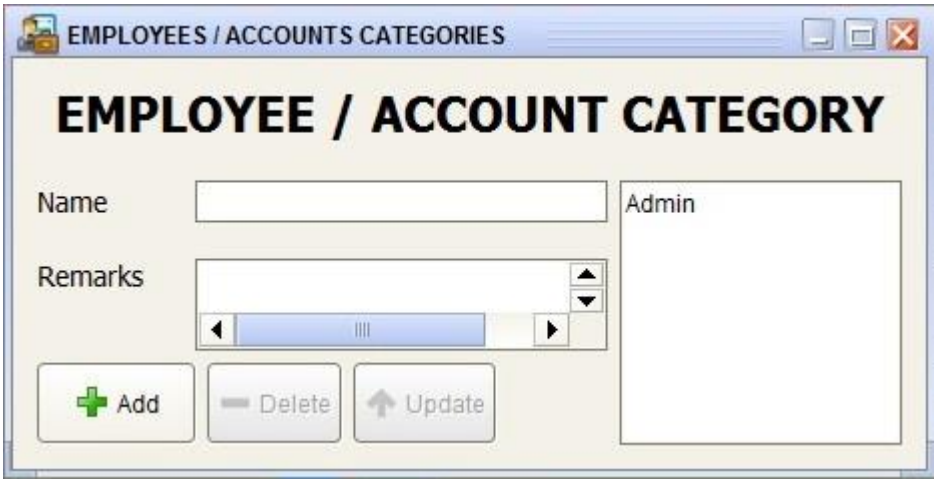


Figure A.4: Employee category frame

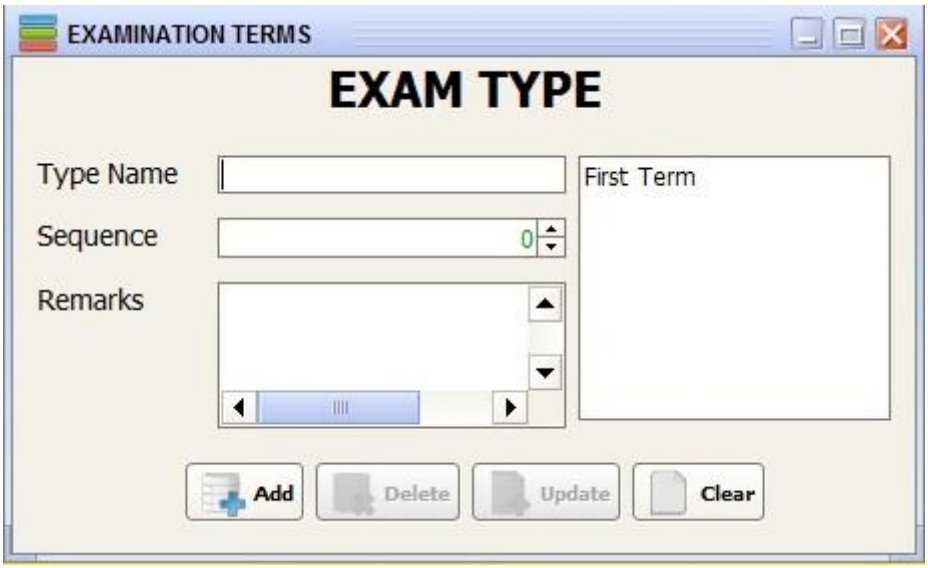


Figure A.5: Exam type frame

The frame class shifting in Figure A.6 is used to shift the student from lower to higher class every year. In ‘From’ area, students appear who have not left school, in other words student record does not contain date of leaving.

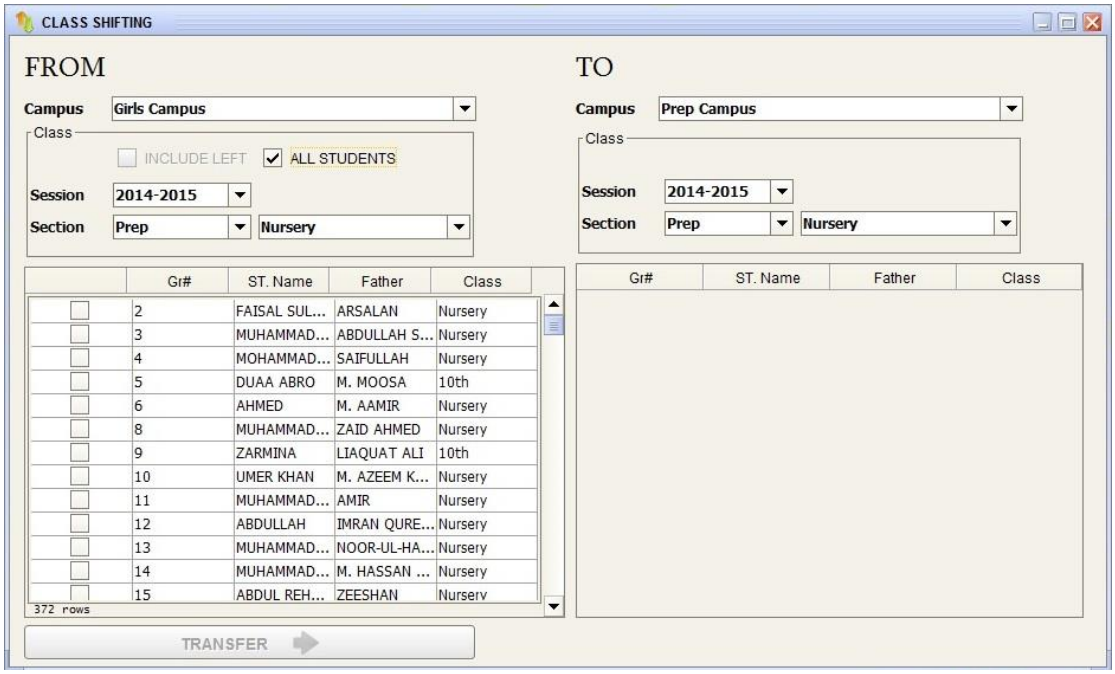
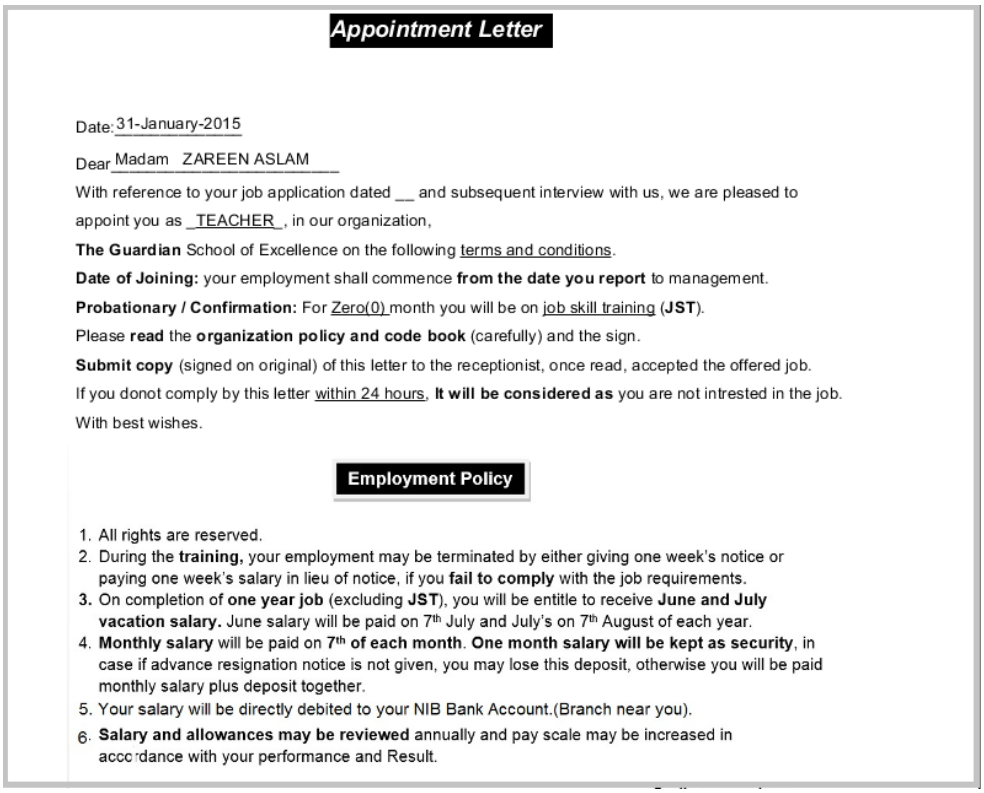


Figure A.6: Class shifting frame

Reports

The Figure shows appointment letter of an employee that is generated when an employee is hired by school after all process of interview and document verification.



Appointment Letter

Date: 31-January-2015

Dear Madam ZAREEN ASLAM

With reference to your job application dated __ and subsequent interview with us, we are pleased to appoint you as _TEACHER_, in our organization,

The Guardian School of Excellence on the following terms and conditions.

Date of Joining: your employment shall commence **from the date you report** to management.

Probationary / Confirmation: For Zero(0) month you will be on job skill training (**JST**).

Please **read** the **organization policy and code book** (carefully) and the sign.

Submit copy (signed on original) of this letter to the receptionist, once read, accepted the offered job.

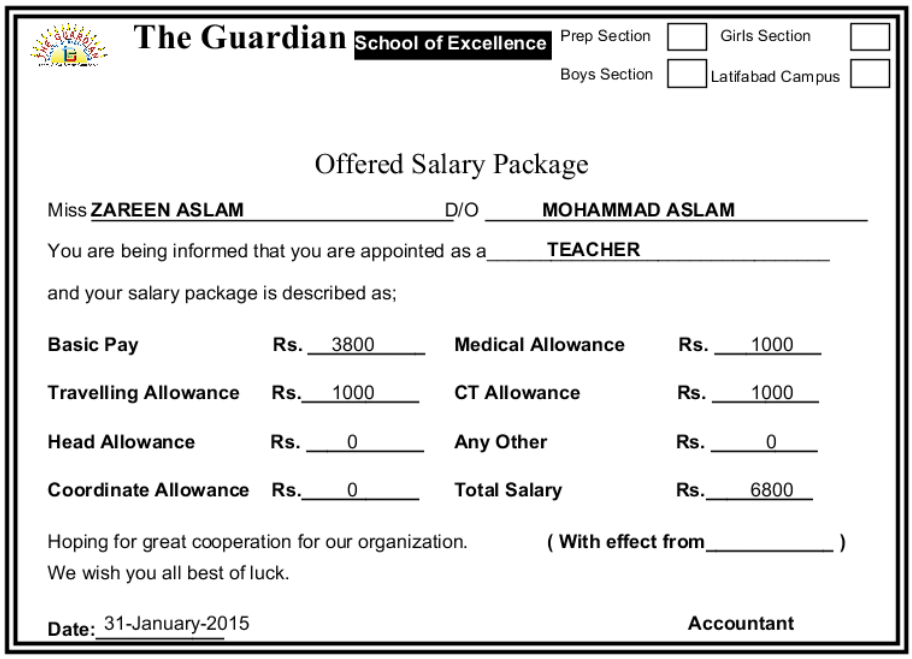
If you donot comply by this letter within 24 hours, **It will be considered as** you are not intrested in the job.

With best wishes.

Employment Policy

1. All rights are reserved.
2. During the **training,** your employment may be terminated by either giving one week's notice or paying one week's salary in lieu of notice, if you **fail to comply** with the job requirements.
3. On completion of **one year job** (excluding **JST**), you will be entitle to receive **June and July vacation salary.** June salary will be paid on 7th July and July's on 7th August of each year.
4. **Monthly salary** will be paid on **7th of each month**. **One month salary will be kept as security**, in case if advance resignation notice is not given, you may lose this deposit, otherwise you will be paid monthly salary plus deposit together.
5. Your salary will be directly debited to your NIB Bank Account.(Branch near you).
6. **Salary and allowances may be reviewed** annually and pay scale may be increased in accordance with your performance and Result.

Figure A.7: Appointment letter



The Guardian School of Excellence

Prep Section ☐ Girls Section ☐

Boys Section ☐ Latifabad Campus ☐

Offered Salary Package

Miss **ZAREEN ASLAM** D/O **MOHAMMAD ASLAM**

You are being informed that you are appointed as a **TEACHER**

and your salary package is described as;

Basic Pay	**Rs.** 3800	**Medical Allowance**	**Rs.** 1000
Travelling Allowance	**Rs.** 1000	**CT Allowance**	**Rs.** 1000
Head Allowance	**Rs.** 0	**Any Other**	**Rs.** 0
Coordinate Allowance	**Rs.** 0	**Total Salary**	**Rs.** 6800

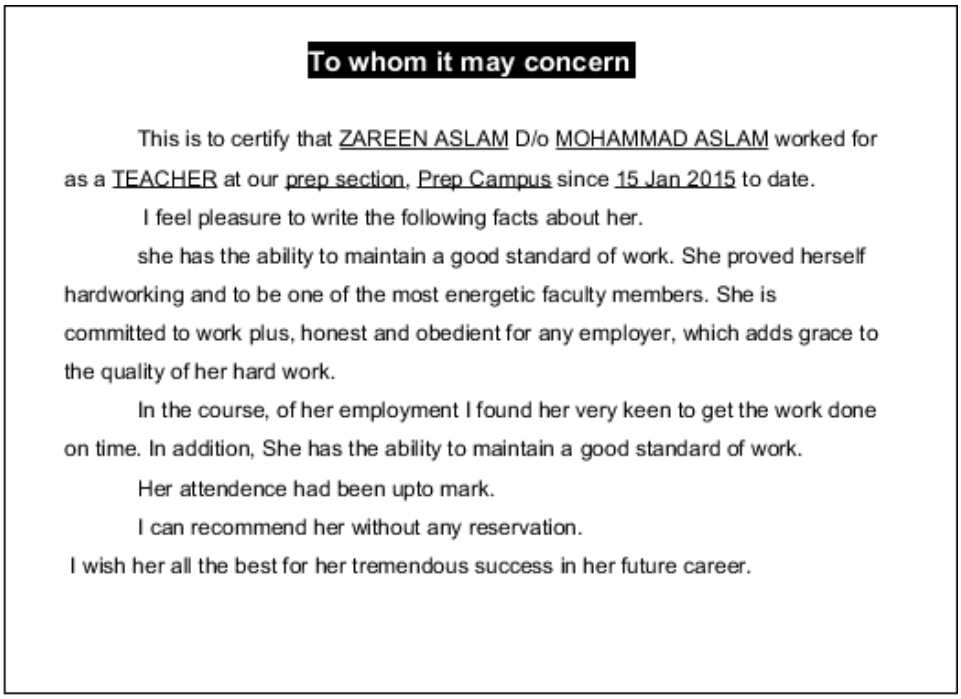
Hoping for great cooperation for our organization. **(With effect from_____)**

We wish you all best of luck.

Date: 31-January-2015 **Accountant**

Figure A.8: Salary offer slip

The Figure A.8 shows salary offer slip, given to employee with appointment letter.



To whom it may concern

This is to certify that ZAREEN ASLAM D/o MOHAMMAD ASLAM worked for as a TEACHER at our prep section, Prep Campus since 15 Jan 2015 to date.

I feel pleasure to write the following facts about her.

she has the ability to maintain a good standard of work. She proved herself hardworking and to be one of the most energetic faculty members. She is committed to work plus, honest and obedient for any employer, which adds grace to the quality of her hard work.

In the course, of her employment I found her very keen to get the work done on time. In addition, She has the ability to maintain a good standard of work.

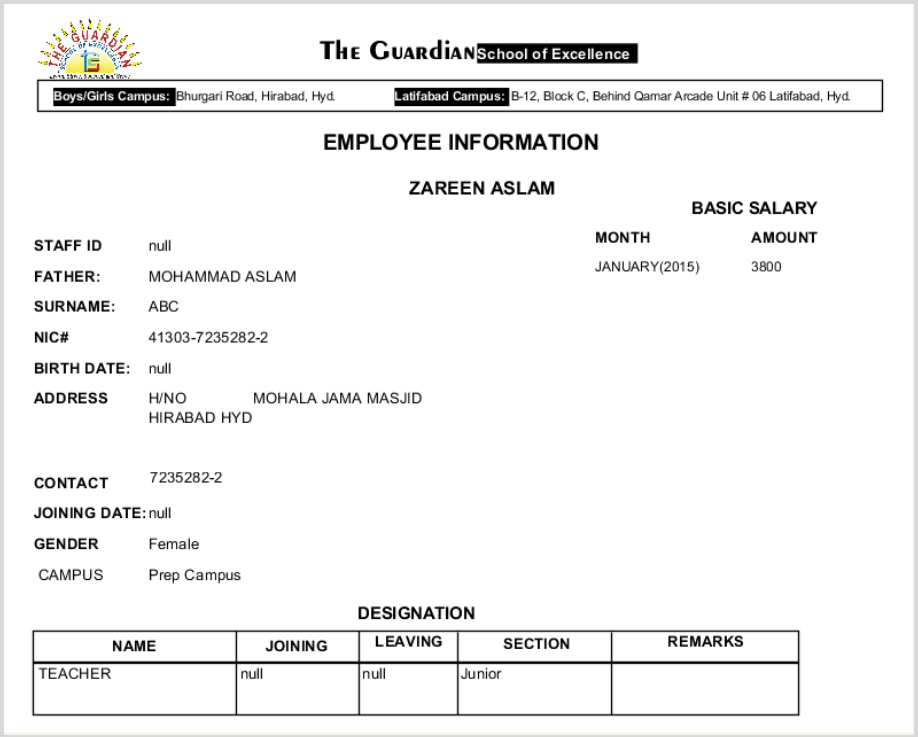
Her attendence had been upto mark.

I can recommend her without any reservation.

I wish her all the best for her tremendous success in her future career.

Figure A.9: Experience letter

The Figure A.9 shows report of experience letter of an employee.



THE GUARDIAN School of Excellence

Boys/Girls Campus: Bhurgari Road, Hirabad, Hyd. Latifabad Campus: B-12, Block C, Behind Qamar Arcade Unit # 06 Latifabad, Hyd.

EMPLOYEE INFORMATION

ZAREEN ASLAM

BASIC SALARY

MONTH	AMOUNT
JANUARY(2015)	3800

STAFF ID null
FATHER: MOHAMMAD ASLAM
SURNAME: ABC
NIC# 41303-7235282-2
BIRTH DATE: null
ADDRESS H/NO MOHALA JAMA MASJID HIRABAD HYD
CONTACT 7235282-2
JOINING DATE: null
GENDER Female
CAMPUS Prep Campus

DESIGNATION

NAME	JOINING	LEAVING	SECTION	REMARKS
TEACHER	null	null	Junior	

Figure A.10: Employee information report

The Figure A.10 shows report of all description of a single employee. The Figure A.11 shows the frame of communication port. We connect our operating system with mobile which support AT commands via Bluetooth. The operating system assign a communication port to that connectivity. That assigned port will be set when we send SMS to students.

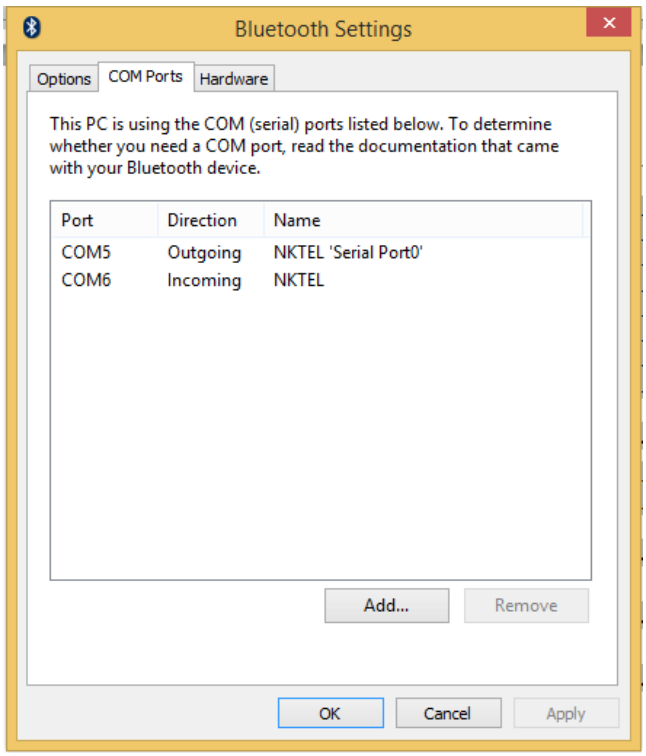


Figure A.11: COM Port Frame

APPENDIX B: CODE

SMS_CONFIG.java

```
package sms;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.comm.SerialPort;

/**
 * @author JAY KUMAR
 */
public class SMS_CONFIG {

    public     static     String[]     BAUD_RATES_KEY     =
{"600","2400","4800","9600","7200","12000","14400","19200
","38400","57600","115200","230400"};
    public      static      int[]      BAUD_RATES      =
{600,2400,4800,9600,7200,12000,14400,19200,38400,57600,11
5200,230400};

    public     static     String[]     DATABITS_KEY     =
{"5","6","7","8"};
    public     static     int[]     DATABITS_VALUES     =
{SerialPort.DATABITS_5,SerialPort.DATABITS_6,SerialPort.D
ATABITS_7,SerialPort.DATABITS_8};

    public     static     String[]     STOPBITS_KEY     =
{"1","1_5","2"};
    public     static     int[]     STOPBITS_VALUES     =
{SerialPort.STOPBITS_1,SerialPort.STOPBITS_1_5,SerialPort
.STOPBITS_2};

    public      static      String[]      PARITY_KEY      =
{"Even","Mark","None","Odd","Space"};
    public     static     int[]      PARITY_VALUES      =
{SerialPort.PARITY_EVEN,SerialPort.PARITY_MARK,SerialPort
.PARITY_NONE,SerialPort.PARITY_ODD,SerialPort.PARITY_SPAC
E};

    public static int getBaudRate(String s){
        return get(s, BAUD_RATES_KEY, BAUD_RATES);
    }

    public static int getDatabits(String s){
        return get(s, DATABITS_KEY, DATABITS_VALUES);
    }

    public static int getStopbits(String s){
        return get(s, STOPBITS_KEY, STOPBITS_VALUES);
    }

    public static int getParity(String s){
        return get(s, PARITY_KEY, PARITY_VALUES);
    }
```

```
    public static String getBaudRate(int s){
        return get(s, BAUD_RATES_KEY, BAUD_RATES);
    }

    public static String getDatabits(int s){
        return get(s, DATABITS_KEY, DATABITS_VALUES);
    }

    public static String getStopbits(int s){
        return get(s, STOPBITS_KEY, STOPBITS_VALUES);
    }

    public static String getParity(int s){
        return get(s, PARITY_KEY, PARITY_VALUES);
    }


    public static int get(String s,String[] keys,int[]
values){
        for(int i=0;i<keys.length;i++){
            if(keys[i].equals(s)){
                return values[i];
            }
        }
        try {
            throw new Exception("Item not found in
Array...!");
        } catch (Exception ex) {

Logger.getLogger(SMS_CONFIG.class.getName()).log(Level.SE
VERE, null, ex);
        }
        return -200;
    }

    public static String get(int v,String[] keys,int[]
values){
        for(int i=0;i<keys.length;i++){
            if(values[i]== v){
                return keys[i];
            }
        }
        try {
            throw new Exception("Item not found in
Array...!");
        } catch (Exception ex) {

Logger.getLogger(SMS_CONFIG.class.getName()).log(Level.SE
VERE, null, ex);
        }
        return null;
    }

}
```

Connect.java

```
package sms;
import java.io.*;
import java.util.Enumeration;
import java.util.Properties;
import java.util.TooManyListenersException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.comm.CommPortIdentifier;
import javax.comm.SerialPort;
import javax.comm.SerialPortEvent;
import javax.comm.SerialPortEventListener;

public class Connect implements  SerialPortEventListener
{
    public static int SMS_SENT = 1;
    public static int SMS_SENDING_FAILED = 0;

    public    static    String    SMS_CONFIG_FILE_NAME    =
"smsconfig.properties";
    public static String BAUD_RATE_KEY = "BAUD_RATE";
    public    static    String    SMS_DELAY_TIME_KEY    =
"SMS_DELAY_TIME";
    public static String DATABIT_KEY = "DATABIT";
    public static String STOPBIT_KEY = "STOPBIT";
    public static String PARITY_KEY = "PARITY";

    String line1 ;
    String line2 ;
    int index=0;

    CommPortIdentifier portId;
    String messageString = "Dont consider this sms, just
for testing....:-D";
    char ch = '"';
    String dest ="";
    String s = "";
    InputStream inputStream;

    SerialPort serialPort;
    OutputStream outputStream;

    public  Connect(String  port,Properties  props)throws
Exception
    {
            portId                                          =
CommPortIdentifier.getPortIdentifier(port);

            serialPort          =          (SerialPort)
portId.open("Sms_GSM", 0);
            System.out.println("SMS       Sending....Port
Found");

            inputStream = serialPort.getInputStream();
            outputStream = serialPort.getOutputStream();
```

```
            serialPort.notifyOnDataAvailable(true);

            serialPort.setSerialPortParams(

SMS_CONFIG.getBaudRate((String)props.get(BAUD_RATE_KEY)),

SMS_CONFIG.getDatabits((String)props.get(DATABIT_KEY)),

SMS_CONFIG.getStopbits((String)props.get(STOPBIT_KEY)),

SMS_CONFIG.getParity((String)props.get(PARITY_KEY)));
            System.out.println("setted    serial    port
parameters");
            recieve();
    }


    public void addEventListener(SerialPort serial){

        try{
            serialPort.addEventListener(this);
        }
        catch (TooManyListenersException e){
            System.out.println("Exception    in    Adding
Listener" + e);
        }
    }

    @Override
    public void serialEvent(SerialPortEvent event)
    {

        BufferedReader  input  =  new  BufferedReader(new
InputStreamReader(inputStream));
        index = 0;

if(event.getEventType()==SerialPortEvent.DATA_AVAILABLE){
            try {
                while((s=input.readLine())!=null){
                    System.out.println("line : "+s);
                    if(s.contains("CMTI")){
                        new Thread(){
                            public void run(){

if((index=s.indexOf(','))!=-1){

s=s.substring(index+1);
                                    index                =
Integer.parseInt(s.trim());

System.out.println("Index:"+index);
                                    try {

outputStream.write("AT+CMGF=1\r\n".getBytes());

outputStream.write(("AT+CMGR="+index+"\r\n").getBytes());
```

```
outputStream.flush();


Thread.sleep(200);
                                        }   catch   (Exception
ex) {

ex.printStackTrace();
                                        }

                                    }
                                }
                            }.start();
                        }
                        if(s.contains("+CMGR:")){
                                String          msg          =
(s+"\n"+input.readLine());
                                System.out.println("Message:
"+msg);
                                String          from         =
msg.split(",")[1];
                                String        message        =
msg.substring(msg.lastIndexOf("\"")+1);
//                                                          new
MessageHandler(from.replaceAll("\"",           ""),message
).start();
//                                deleteMessage(index);
                        }

                    }

                } catch (IOException ex) {
                    ex.printStackTrace();}
            }

        }

    synchronized  public  int  send(String  phoneNo,String
message){
            dest = ch + phoneNo + ch;
            messageString = message;
            System.out.println("To: "+dest);
            System.out.println("Message: "+message);

            line1 = "AT+CMGF=1\n";
            line2 = "AT+CMGS=" + dest + "\n";

            //String line1 = "AT+CREG=2";
            //String line2 = "AT+CGREG?";
            try
            {
                Thread.sleep(400);
                System.out.println("now sending...");

                outputStream.write(line1.getBytes());
```

```
            outputStream.write(line2.getBytes());
            outputStream.write(messageString.getBytes());
            outputStream.write(26);
            outputStream.flush();
            Thread.sleep(400);
            return SMS_SENT;
        }
        catch (Exception e)
        {
            System.out.println("Error writing message " +
e);
        }
        return SMS_SENDING_FAILED;
    }
    public void contactEntries(int index){
        String command = "AT+CPBR="+index+"\r\n";
        try{
            outputStream.write(command.getBytes());
            outputStream.flush();
            byte[] b = new byte[20];
            inputStream.read(b , 0,20);
            String s = new String(b,0,b.length);
            System.out.println("read       :       "+s+":
length"+s.trim().length() );
        }catch (Exception e){
            System.out.println("Error writing message " +
e);
        }

    }

    public void recieve(){
        try{
            String s = "AT+CNMI=1,1,0,0,0\r";
            String source = "AT+CPMS=\"SM\"\r\n";
            outputStream.write(s.getBytes());
            outputStream.write(source.getBytes());
            outputStream.flush();
            addEventListener(serialPort);
        }catch(Exception ex){ex.printStackTrace();}
   }
    synchronized public void deleteMessage(int index){
        String source = "AT+CMGD="+index+"\r\n";
        try {
            outputStream.write(source.getBytes());
            outputStream.flush();
        } catch (IOException ex) {

Logger.getLogger(Connect.class.getName()).log(Level.SEVER
E, null, ex);
        }

    }

    public void close(){
        if(serialPort!=null){
```

```
            serialPort.close();
        }
        serialPort = null;
    }


    // ALL STATIC METHODS

    public     static     Enumeration<CommPortIdentifier>
detectPorts(){
        Enumeration<CommPortIdentifier>        e        =
CommPortIdentifier.getPortIdentifiers();
        System.out.println("Detecting Port..!");
        return e;
    }

    public  static  Properties  prepareProperties(String
baudRate,  String  databits,  String  stopBits,  String
parity,String delayTime){
        Properties props = new Properties();
        props.put(BAUD_RATE_KEY, baudRate);
        props.put(DATABIT_KEY, databits);
        props.put(STOPBIT_KEY, stopBits);
        props.put(PARITY_KEY, parity);
        props.put(SMS_DELAY_TIME_KEY, delayTime);
        return props;
    }

    public    static    void    saveProperties(Properties
props)throws Exception{
        File f = new File(SMS_CONFIG_FILE_NAME);
        if(!f.exists()){
            f.createNewFile();
        }
        FileOutputStream out = new FileOutputStream(f);
        props.store(out, null);
        out.close();
    }

    public   static   Properties   getProperties()throws
Exception{
        Properties props = null;
        File f = new File(SMS_CONFIG_FILE_NAME);
        if(!f.exists()){
            props= prepareProperties(

SMS_CONFIG.getBaudRate(SMS_CONFIG.BAUD_RATES[3]),

SMS_CONFIG.getDatabits(SMS_CONFIG.DATABITS_VALUES[3]),

SMS_CONFIG.getStopbits(SMS_CONFIG.STOPBITS_VALUES[0]),

SMS_CONFIG.getParity(SMS_CONFIG.PARITY_VALUES[2]),
                    "400" );
            saveProperties(props);
        }else{
```

```
            props = new Properties();
        }
        props.load( new FileInputStream(f));
        return props;
    }
}
```