



Zagazig University
Faculty of Computers and Information
Computer Science Department



School Management System



Supervisor
DR. Mahmoud Ismail
Decision support Department



Zagazig University
Faculty of Computers and Information
Computer Science Department

Graduation Project documentation

School Management System

Supervisor
DR. Mahmoud Ismail
Decision support Department

Abstract:

School application is a web application which based on ADF Technology of Oracle, this app provides services to help everyone in school to perform his work easily without any wasting for time or high cost, this app can be used by teacher, Student, Parent, and management staff.

There are a lot of schools, and often public schools that lack commercial support, and the lack of possibilities to be leading to a lack of ability to buy or doing a school software system in order to help them in business administration, exams, monitor grades, students evaluate, communicate with parents.... etc.

This documentation describes our graduation project in four chapters:

- In Chapter one you will find a brief introduction about the project
- Chapter two discuss the application Methodology of the System
- In Chapter three you will find System Requirement Study and analysis
- In chapter Four you will find System Design
- In chapter Five you will Find the Implementation and information about ADF Technology, and how we use it in our project
- Conclusion and future work will be found at the end of the document

Contents List

List of Tables/Figures	1
Acknowledgements	2
Chapter 1:	3
Introduction:.....	3
Chapter 2	5
Methodology:	5
Section 1: Main work	5
Section 2: App Languages and API.....	6
Chapter 3:	7
Section 1 System Requirement Study.....	7
Section 2 system analysis.....	10
Section 2 : System Design	20
Chapter 4:	30
Section 1: Log in page	30
Section 1: log in as admin	31
Section 1 : log in as teacher	43
Section 1 : log in as Student.....	47
Section 1: log in as parent	50
Chapter 5: Implementation	54
Section 1 : Oracle ADF Overview	55
Section 2: Oracle Database:.....	67
Section 3 : Technologies used to build Software.....	70
Section 4 : Implementation code.....	71
Conclusion:.....	81
Future work :.....	82
References	83

List of Tables/Figures

- Figure 1: Level 0 of data flow **Error! Bookmark not defined.**
- Figure 2: Level 1 of data flow **Error! Bookmark not defined.**
- Figure 3 : Physical ERD **Error! Bookmark not defined.**
- Figure 4 : Use case **Error! Bookmark not defined.**
- Figure 5 Log in use case **Error! Bookmark not defined.**
- Figure 6: Class Diagram **Error! Bookmark not defined.**
- Figure 7: Sequence diagram for Admin **Error! Bookmark not defined.**
- Figure 8: Sequence diagram for student **Error! Bookmark not defined.**
- Figure 9: Sequence diagram for teacher..... **Error! Bookmark not defined.**
- Figure 10: Sequence diagram for parent.... **Error! Bookmark not defined.**
- Figure 11: activity diagram for Admin **Error! Bookmark not defined.**
- Figure 12: activity diagram for student **Error! Bookmark not defined.**
- Figure 13: activity diagram for teacher **Error! Bookmark not defined.**
- Figure 14: activity diagram for parent..... **Error! Bookmark not defined.**
- Figure 15:Log in Page **Error! Bookmark not defined.**
- Figure 16:Log in as admin **Error! Bookmark not defined.**
- Figure 17: Main menu of admin **Error! Bookmark not defined.**
- Figure 18: create new account for user..... **Error! Bookmark not defined.**
- Figure 19: admin operations **Error! Bookmark not defined.**
- Figure 20: define stages and patches **Error! Bookmark not defined.**
- Figure 21: define classes **Error! Bookmark not defined.**
- Figure 22: define scheduale **Error! Bookmark not defined.**
- Figure 23: define parent for student **Error! Bookmark not defined.**
- Figure 24: define student settings **Error! Bookmark not defined.**
- Figure 25: define teacher and his subjects. **Error! Bookmark not defined.**
- Figure 26: define test types..... **Error! Bookmark not defined.**
- Figure 27: List of classes and students **Error! Bookmark not defined.**
- Figure 28:add and defune comlaint **Error! Bookmark not defined.**
- Figure 29:Security **Error! Bookmark not defined.**
- Figure 30:Security **Error! Bookmark not defined.**
- Figure 31:Log in as teacher **Error! Bookmark not defined.**

Figure 32:Teacher home page.....	Error! Bookmark not defined.
Figure 33:view scheduale.....	Error! Bookmark not defined.
Figure 34:add an exam.....	Error! Bookmark not defined.
Figure 35:add student evaluation	Error! Bookmark not defined.
Figure 36:Log in as student	Error! Bookmark not defined.
Figure 37:view student home page.....	Error! Bookmark not defined.
Figure 38:View student scheduale	Error! Bookmark not defined.
Figure 39:View student score.....	Error! Bookmark not defined.
Figure 40:Log in as parent.....	Error! Bookmark not defined.
Figure 41:parent home page.....	Error! Bookmark not defined.
Figure 42:Parent view evaluation	Error! Bookmark not defined.
Figure 43:Parent view complaint	Error! Bookmark not defined.
Figure 44:JDeveloper Platform.....	Error! Bookmark not defined.
Figure 45:Orcale desktop layer	Error! Bookmark not defined.
Figure 46:User Interface Development	Error! Bookmark not defined.
Figure 47:Table in toad	Error! Bookmark not defined.
Figure 48:Orcale ADF APP	Error! Bookmark not defined.
Figure 49:Model Of business Component..	Error! Bookmark not defined.
Figure 50:Entity Object	Error! Bookmark not defined.
Figure 51:View Objects	Error! Bookmark not defined.
Figure 52: AM APP Model	Error! Bookmark not defined.
Figure 53: Association	Error! Bookmark not defined.
Figure 54: View Links.....	Error! Bookmark not defined.
Figure 55: Web pages.....	Error! Bookmark not defined.
Figure 56: Admin pages.....	Error! Bookmark not defined.
Figure 57: Student, parent, teacher pages .	Error! Bookmark not defined.
Figure 58: Task Flow	Error! Bookmark not defined.
Figure 59: DB Connection	Error! Bookmark not defined.
Figure 60: Data Control.....	Error! Bookmark not defined.

Acknowledgements

This project could not be done without **Dr. Mahmoud Ismail** who not only served as our supervisor but also encouraged and challenged us throughout our academic program. He and the other department staff, **Dr. Ahmed Raft, Dr. Ahmed Salah, Dr. Hiba El-Fique**, and all other department staff guided us through the learning process, never accepting less than my best efforts. We thank them all.

we would like to express my sincere thanks to our Head of Department.
Prof. Ibrahim El Hanaway

We also thankful to **Mr. said Mohamed Mahmoud Ibrahim** the Manager of school, who we make interview with him, and he give us more information which help us to build our system.

Finally, we would like to thank our **parents** the cause of all beautiful things

Chapter1

Introduction:

The use of computers in the processes of learning and education is one of the newest areas that stormed the computer is known that teachers are always searching for ways to help them perform educational and jobs in order to access better education Sometimes using color images and sometimes uses stereoscopic formats are also used blackboards, books and some simple devices in recent years appeared some modern devices such as recording devices and a microscope and a telescope and devices rear-projection, films and educational devices film projectors and educational television and other despite the multiplicity of these means and diversity, all means serve a specific target. this means complex in installed and used in some cases it is also more expensive than It has led to the reluctance of many schools to purchase and use.

In recent years the use of computers began in the processes of learning and education in advanced computer countries is not just a way educational, but is a several tools in one way being the new functions is unable to be achieved by any other method It provides a two-way interactive learning environment

The computer is a gateway or approach in the field of teaching and learning of different subjects of study With the development of computer hardware and theories of learning and teaching and the development of this portal has become a phenomenon and its implications for the justification and their effects in the processes of learning and education, so we make the system which provide each works of business administration, monitor grades ...etc.

This system is called

" School Management System "

School Management System is a web application in general aims to:

- 1- Make sure that of the staff of management department do his works easily.
- 2- Help Students to their grades, exams, and schedule.
- 3- Parents can know the level of their Students and can communicate with their students easily.
- 4- Teachers can perform their works such as monitor grade, set exams, set degrees, view his schedule ...etc.
- 5- Saving time and effort for Members of system.
- 6- Prevent manual errors, and decrease effort and fatigue of the staff.
- 7- Decrease The high cost of printing papers ...etc.
- 8- keep up with the times.

Chapter2

Methodology

2.1 Main Work

We can divide our application's main work into main four parts:

A- Learning ADF developing basics (took Month)

Before developing our application, we needed to read about ADF because knowledge about ADF developing was not sufficient enough to start developing the project.

B- Building database of the system: (Took a week)

- a. We started in reading about database and know how to convert your ideas for data in a diagram is called ERD
- b. Converting the ER diagram into actual data base
- c. Using toad IDE to deal with tables and define their features

C- Server developing Using Java language: (took a week)

D- Connect ADF with Database: (took a week)

E- Developing the rest of our application services (took 2 months)

F- Application final testing and interface (took one week)

2.1 Applications languages and API's used

Languages used:

- 1- Java language

Applications used:

- 1- J Developer IDE on, Windows 8.1 copy
- 2- Toad, Windows 8.1 copy
- 3- Fire Fox browser

Chapter3

System Requirement Study and analysis

3.1 System Requirement Study

administrator:

The administrator has all the rights to access the system. He is the one who has all rights to view the applicant details, modify those details. The administrator also keeps a track of the file status of the applicants.

Student:

The student can show his own details. The Student has rights to interaction with the school management system with giving the online examination as well as student can show the result. Student can see his absenteeism as well as they have the right to collect document. Student can check the remainder if found.

Teacher:

Teacher can show the responsible task provided by the administrator. He can view his schedule flow and other related tasks. Teacher can assess the exam paper and also can check the paper. He can generate his daily report to the administrator and fulfill the attendance of student. Apart from that he can see his batch with its progress and evaluate student progress. Teacher can also be sending the global note to his desired student.

parent:

Parent can show the report provided by the administrator and teacher. He can view student time table flow other related task. He can view student evaluation and student result.

3.1.1 Hardware Requirement Study

Development technologies	Oracle ADF, Oracle database
Development tools	J Developer
Application server	Web logic server
Database	Oracle 11g
Operating system	Windows 8.1
Web browser	Google chrome, Mozilla Firefox
Hardware	P-IV or +, 4 GB RAM, 80 GB HDD

3.1.2 Constraints

3.1.2.1 Regulatory Policies

- The length of the project is 5 months which a limited amount of time.
- The project developers are beginners and will take time to understand about the technology.
- The users of the system are given permission to be a part of the system by administrator,

3.1.2.2 Reliability Requirements

- The system should be reliable enough so that the data found in the database system is consistent at any point.
- The system should be able to handle loads of requests from different users around the world at the same time.

3.1.2.3 Criticality of the Application

- The system is a web- application and so fails to work if there is no Internet connection. The system might not work if the Internet connection slows down.
- The system stops working in case if the database server or the application server stops working.
- The system might give erroneous output if it fails to connect to the database server.

3.1.2.4 Safety and Security Considerations

- Each applicant is given a login account through which he can view his own information and also modify and save it. He has the rights to access only his own information.
- The administrator has rights through which he can access and manage whole system.

3 -System Requirement Study and analysis

3.2 system analysis

3.2.1 Study of the Current System

The current school management system (SMS) deals with maintaining a physical contact with the school management classes. for filling all the details and the documentation work. The management doesn't need to visit the school. and collect the assignment and submitting his/her documents directly.

According to the current system, the management has to fill in the forms manually, go to the account management school., and submit him the form. The applicant needs to visit the school portal now and then in order to get his work accomplished. The admin also has to manage all the users. He needs to maintain records of all the users, their activity status, submission methods and installation details on paper. The Manual process is more error prone and also slow. Moreover, Students in the school can interface his/her work area only. But if an online application is available then they can communicate whole system. Thus a simulation of this entire process can be a boon to the applicants as well as the admin.

3.2.2 Problems and Weakness of Current System

- The present system has certain major disadvantages. A few to be listed can be excessive paperwork, time consuming process flow, laborious work environment for employees, difficulty to access historical data and all these problems lead to inefficient working of government sector causing dissatisfaction in the general public.

- Apart from the above stated problems there is lack of transparency in the existing system. This being one of the major drawbacks in the system needs special attention.
- The problem stated above have certain deep rooted problems like time consuming process flow for which may need to change the structure of the process flow in certain cases so that the system output can become faster.
- The following listed are the problems or weaknesses of the current system:
- So much time consume in preparing registers which is having replicated data
- It is difficult to prepare report for decision making.
- Attendance related module is not there.
- Reporting and appraisal of the performance are not there.

3.2.3 Requirement of New System

- Registration details of the applicant.
- Login details of the applicant.
- Personal details of the applicant.
- Information of all the members of the applicant's group.
- Educational and employment information
- All information and rules regarding the e-forms must follow.
- Certain legal details of the Student and teacher
- Answers to the questionnaire for exam.
- Communication with whole system.

3.2.4 Feasibility Study

The aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the system or not. A feasibility study is carried out from following different aspects.

3.2.4 Feasibility Study

The aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the system or not. A feasibility study is carried out from following different aspects.

3.2.5 Operational Feasibility:

The system has been developed for any user who wants to use this system. We have given a demo of our project and the users found the system friendly and easy to use. The interoperability with the existing system is also checked after uploading the website. So they may face certain problems in using the user interface. So keeping this consideration in mind we have provided field for each and every field on the forms. The administrator also may be non-technical, so the user interface is designed in such a way that it gets comfortable for the non-technical person to operate easily.

3.2.6 Technical Feasibility:

It determines if the system can be implemented using the current technology. This system has been developed by using Oracle ADF

3.2.7 Economical Feasibility:

The company being a well-to-do company didn't have any problem in buying any software that was required in developing the application. The software's we used were readily available. So as such we didn't face any economical constrains.

3.2.8 Implementation Feasibility:

This project can easily be made available online without much consideration of the hardware and software. The only required thing at the applicant's side is the Internet connection and a web browser, which are a no difficult issue these days. A database server and application server are required to set up at the admin side. After setting up the project online, even the administrator can access the system from anywhere.

3.2.9 Requirements Validation:

Requirement Validation examines the specification to ensure that all system requirements have been stated unambiguously; those inconsistencies, errors have been detected and corrected and the work products conform to the standard.

- Source of the requirements are identified. Final statement of requirement has been examined by original source.
- Requirements related to main requirements are found.
- Requirements are testable.
- Requirements are clearly stated and are not misinterpreted.
- All sources of requirements are covered to get maximum requirement.
- All methods of finding requirements are applied.
- Requirements are not duplicated and each of them gives distinct idea of processes within project.
- Requirement associated with system performance, behavioral and operational characteristics are clearly stated.
- Requirements are being discussed with the client in order to remove the misinterpretations if they exist.
- Each requirement is being analyzed to prove its feasibility for the current system.

3.2.10 data flow diagram:

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes and how it gets stored.

Level 0:

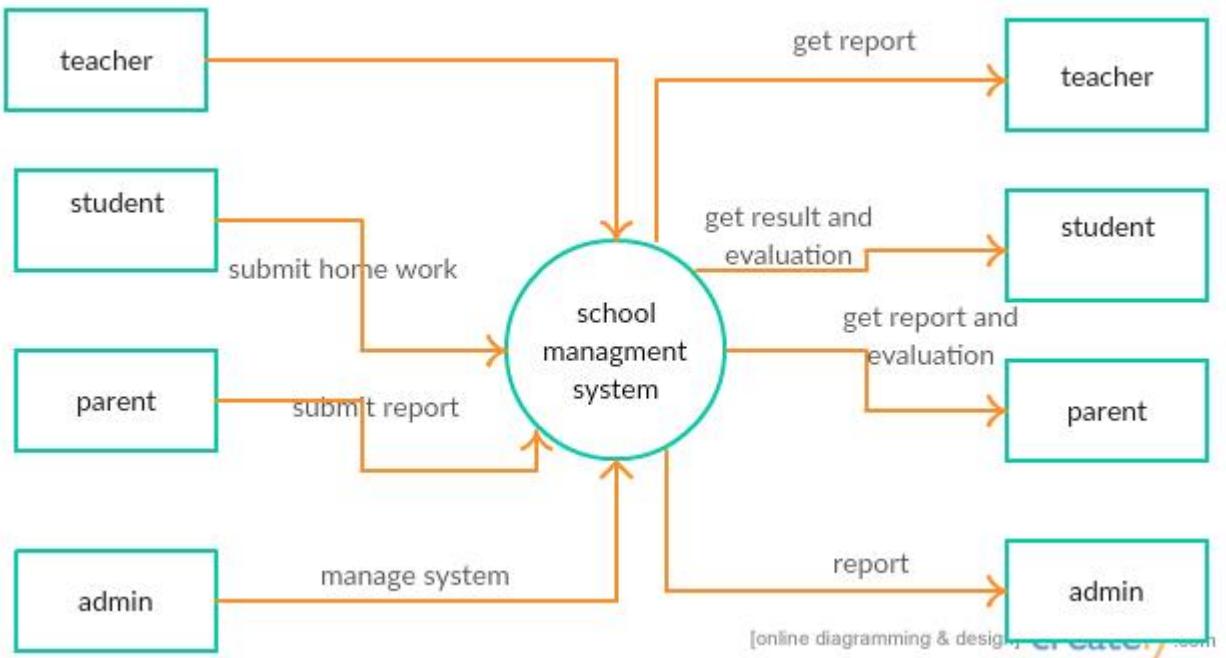


Figure 1 level 0 data flow

Level 1 :

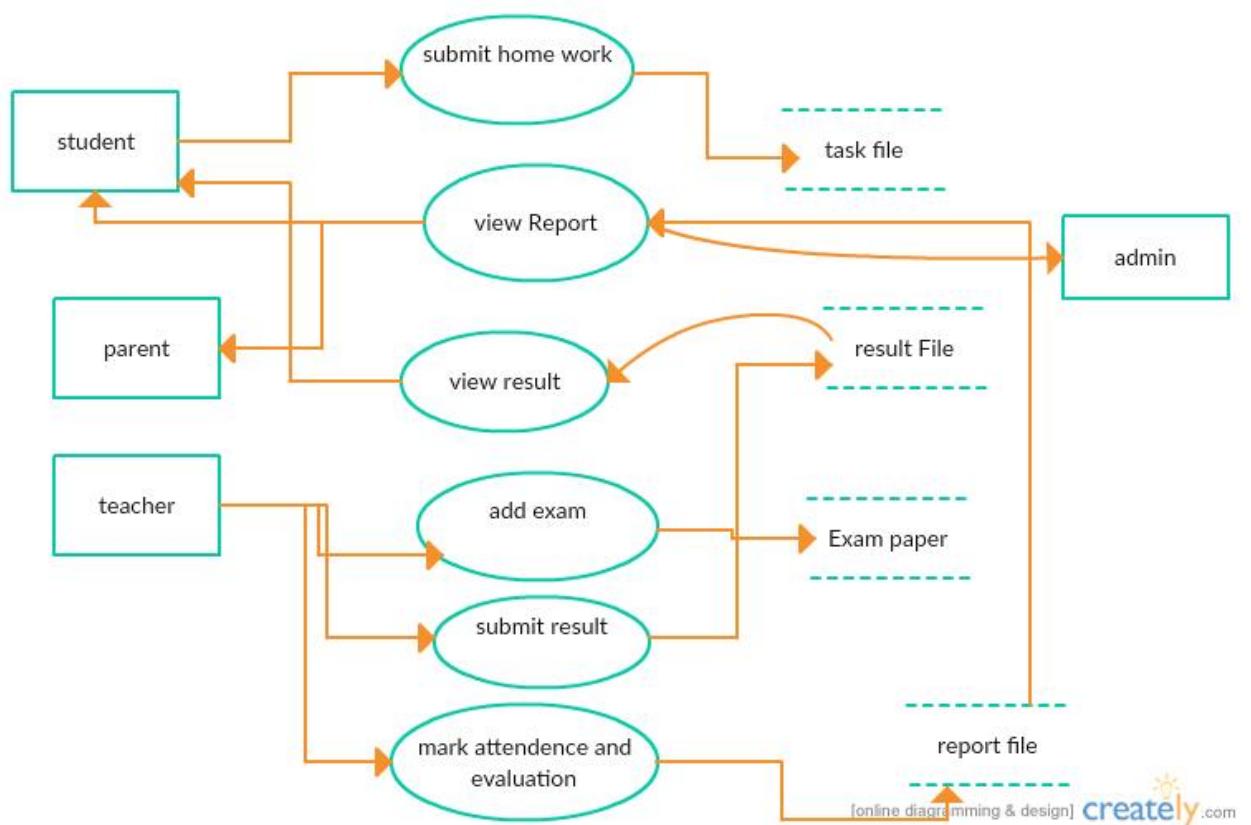


Figure 2 level1 data flow

ERD:

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. It does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

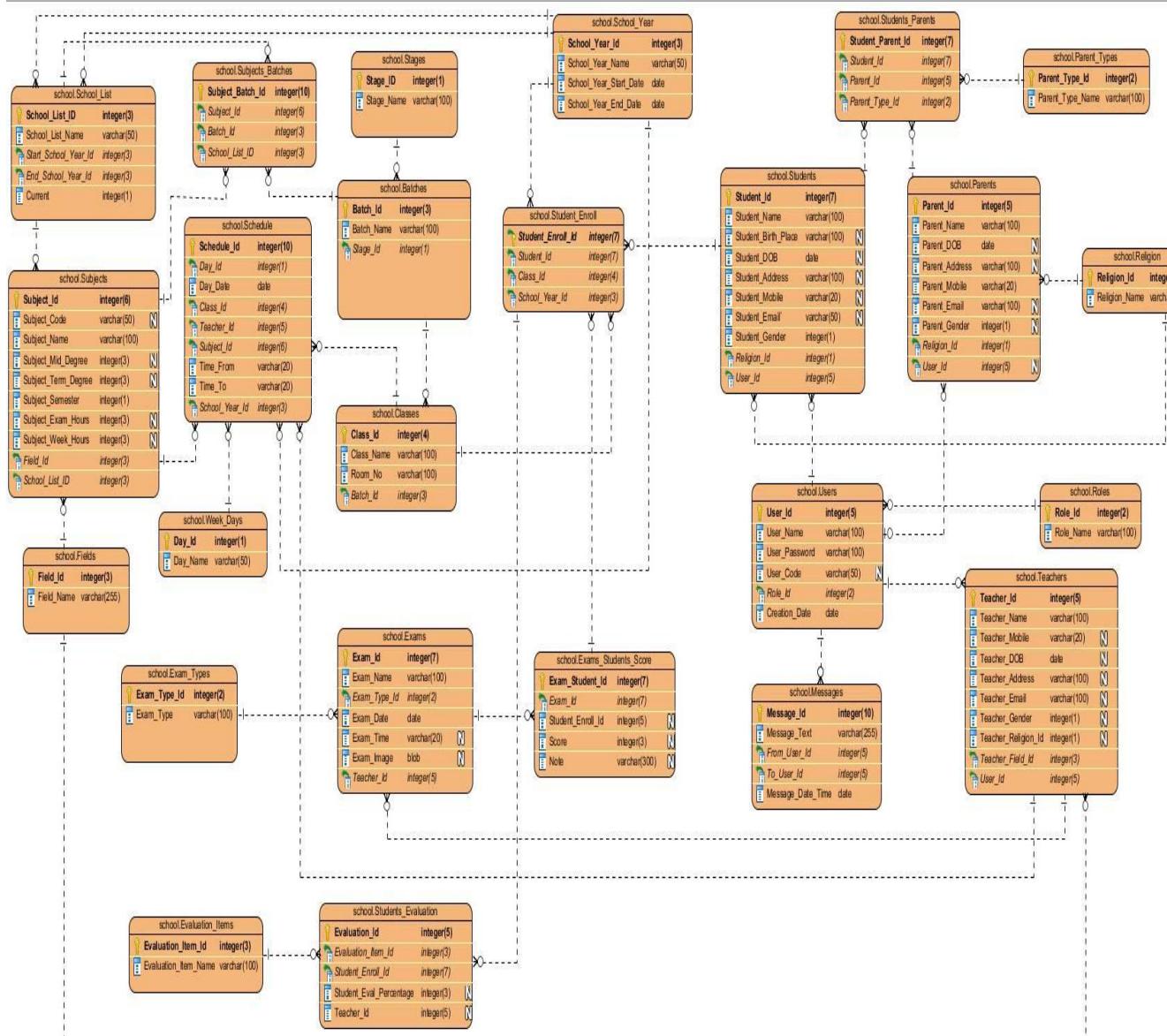


Figure 3 physical ERD

3.2.11 Use Case Diagram:

is a UML (unified model language) behaviour part representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

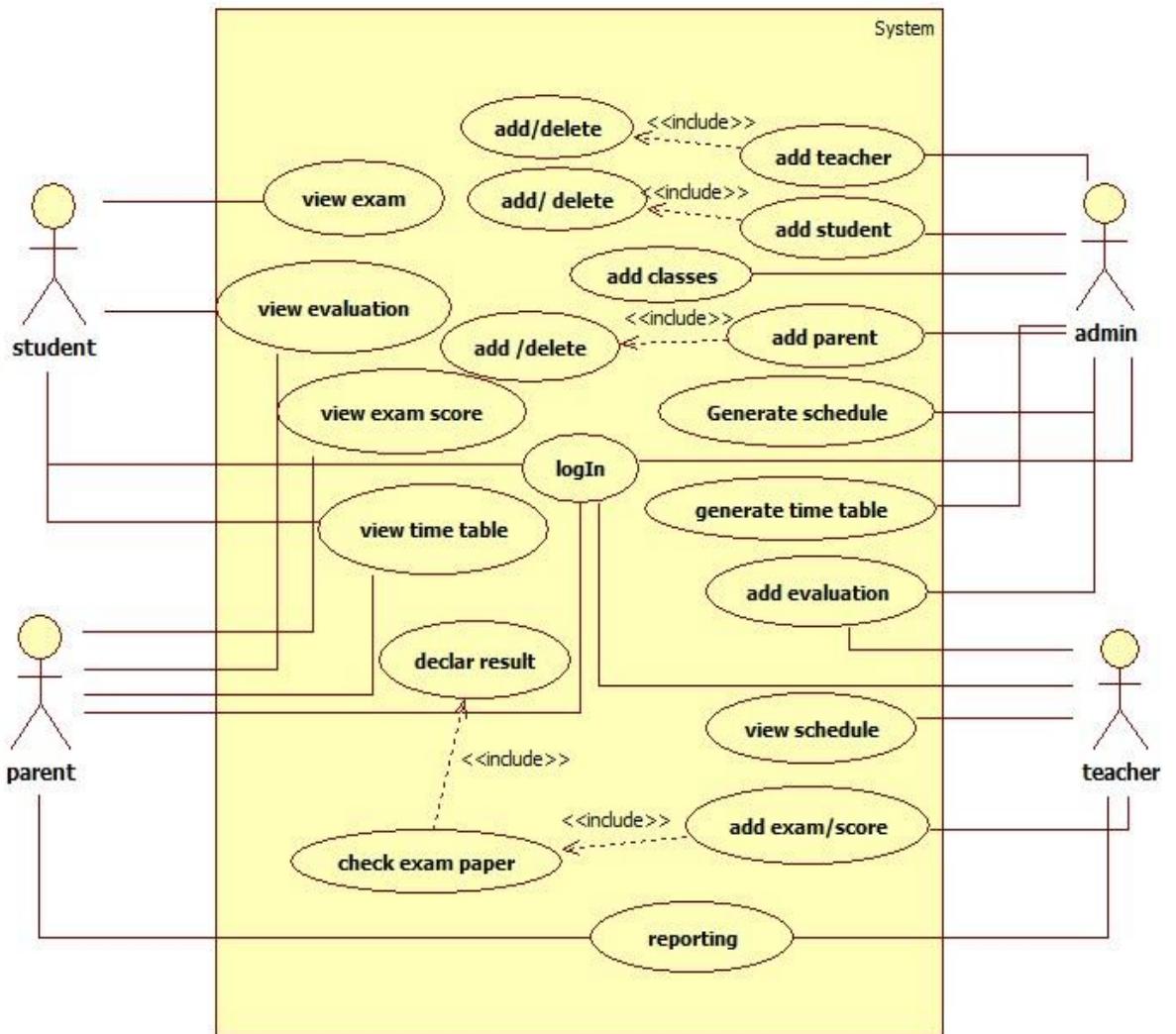


Figure 4 use case

Use Case Diagram:

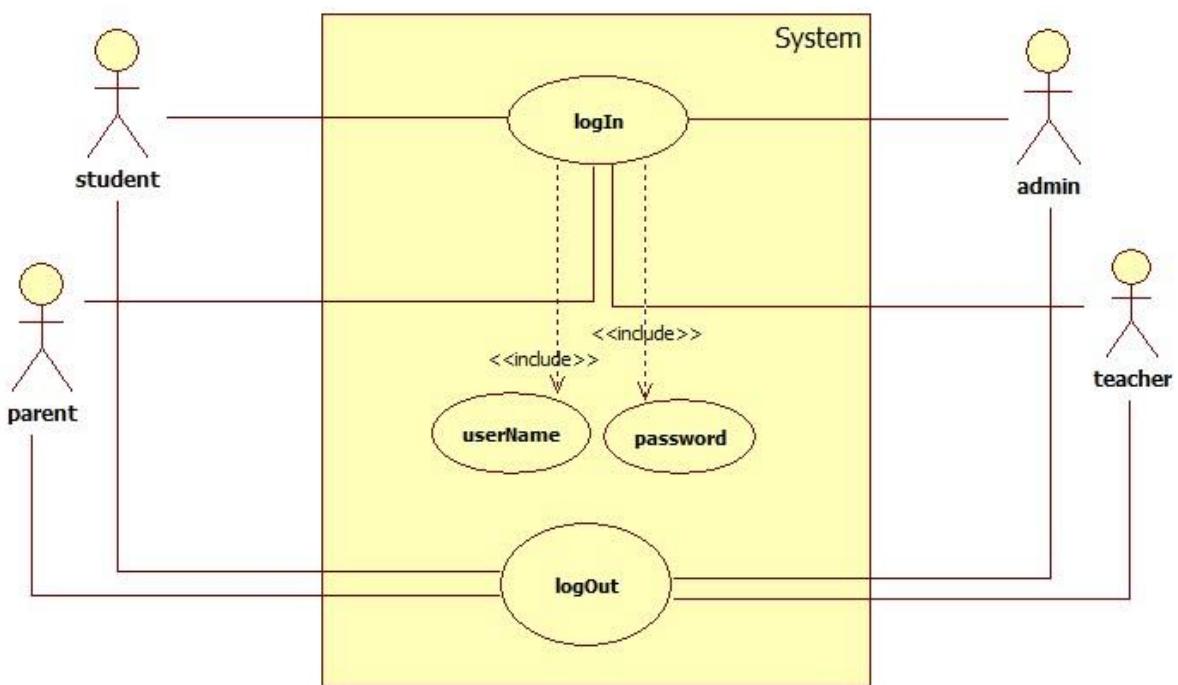


Figure 5 log in use case

3.2.12 class Diagram:

a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

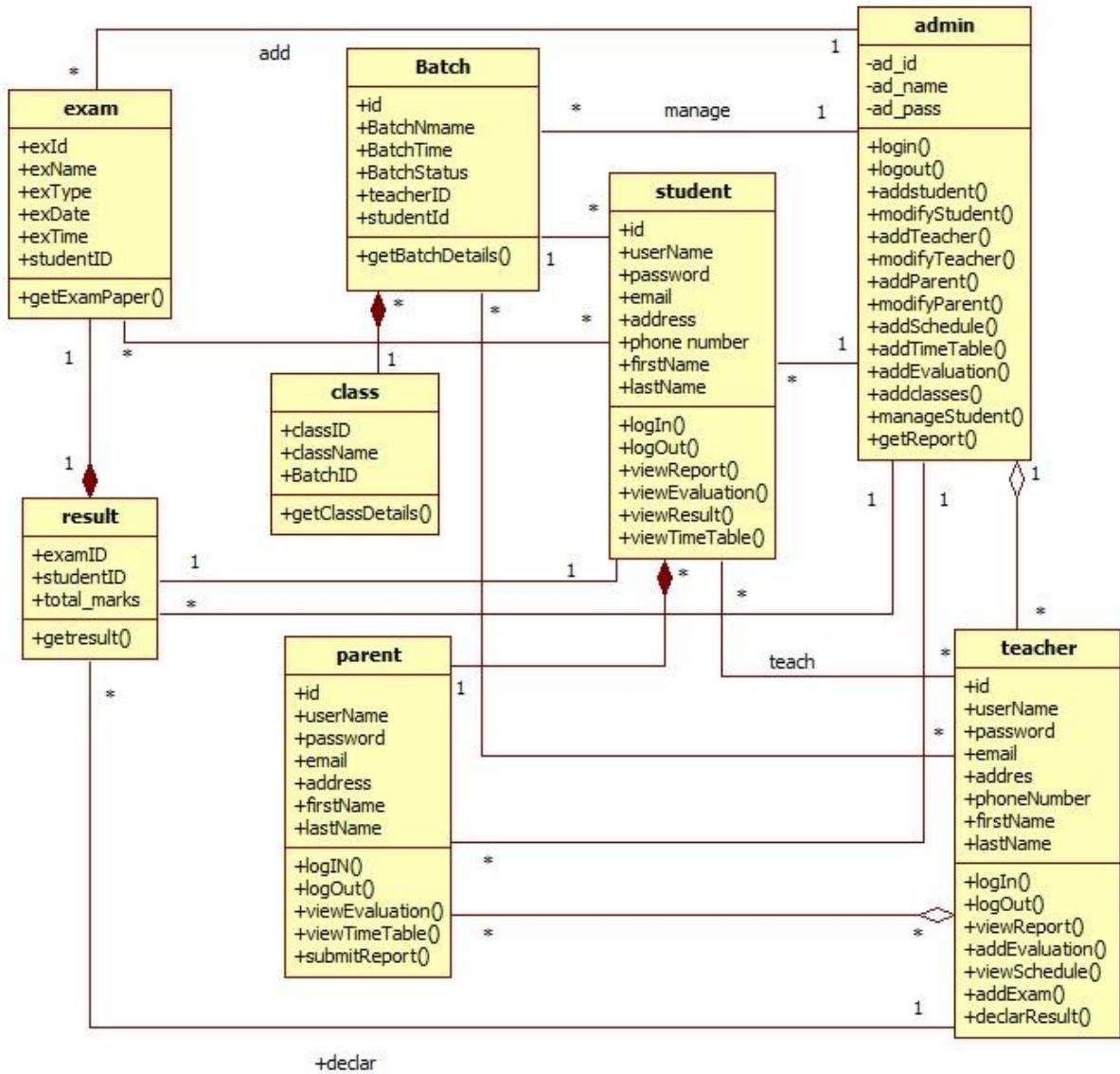


Figure 6 class diagram

(3-p2) System Design

3p2.1 definition of the System Design:

System design is the process of defining the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running

system designing phase commences after the system analysis phase is completed. It's appropriate to mention that the output or the specifications taken through the phase of system analysis become an input in the system design phase which in turn leads to workout based on the user defined estimations.

3p2.2 sequence diagram:

is an interaction diagram that shows how objects operate with one another and in what order? It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

3p2.2.1 sequence diagram for admin:

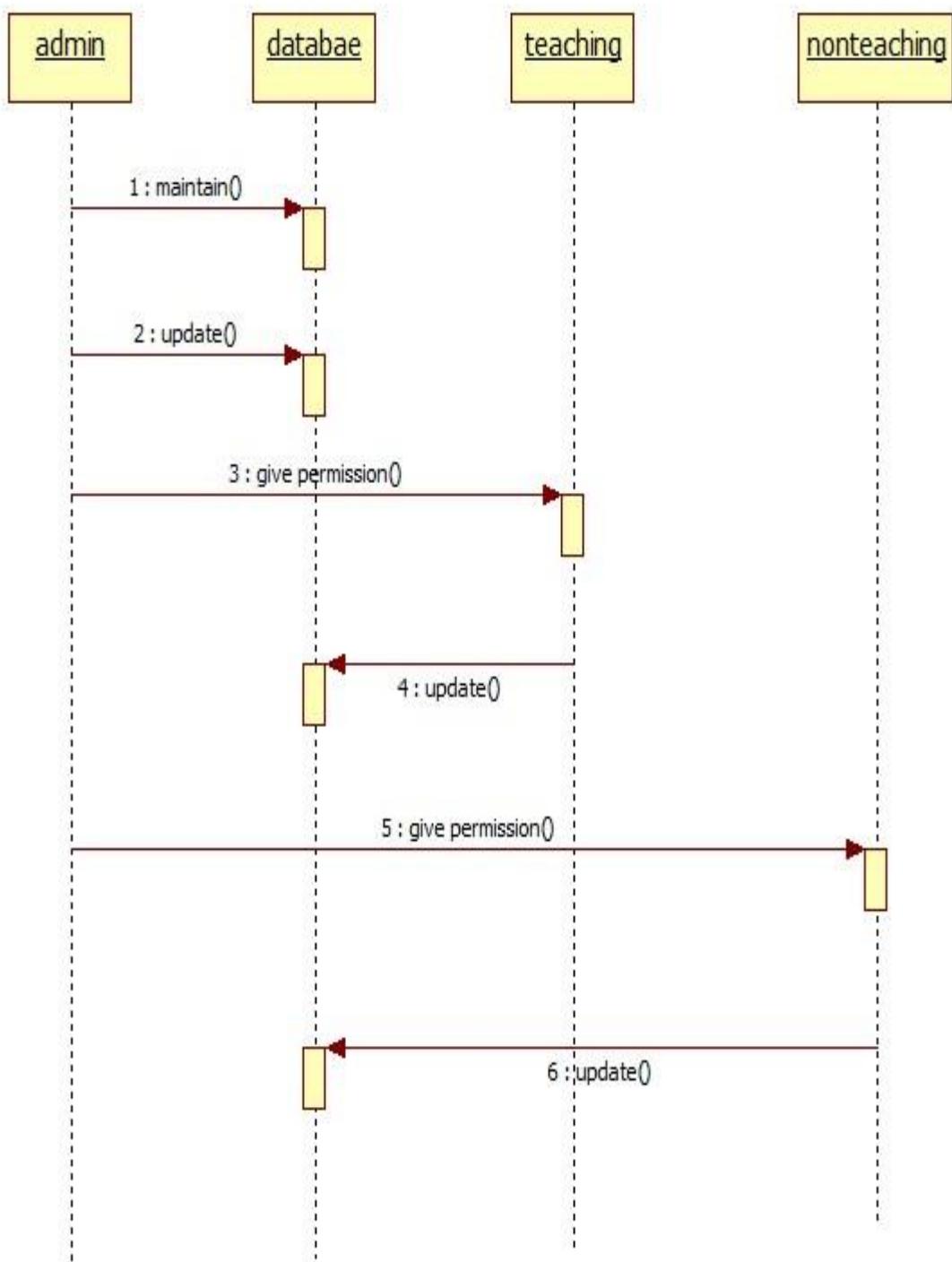


Figure 7 sequence diagram for admin

3p2.2.2 sequence diagram for student:

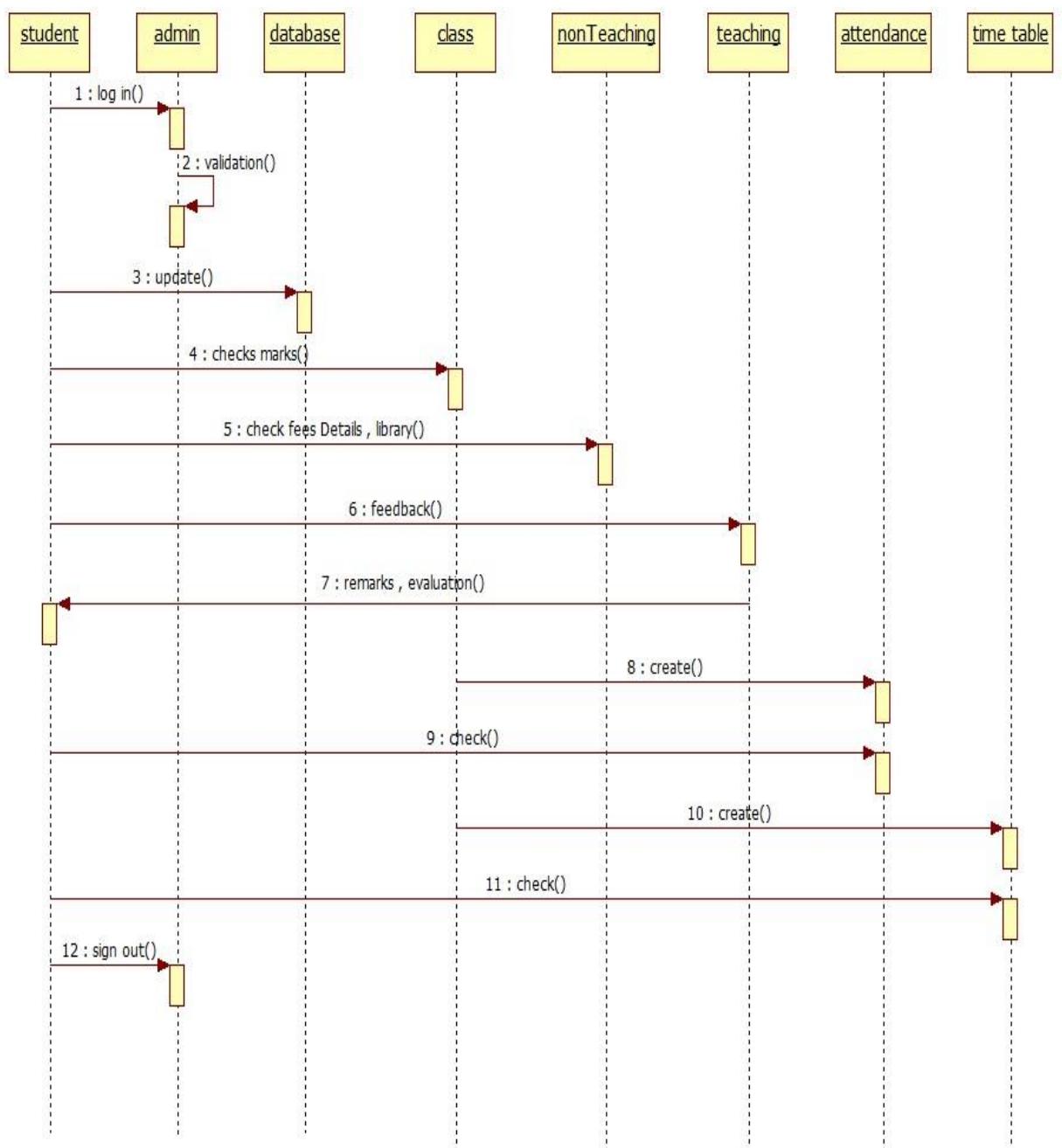


Figure 8 sequence diagram for student

3p2.2.3 sequence diagram for teacher:

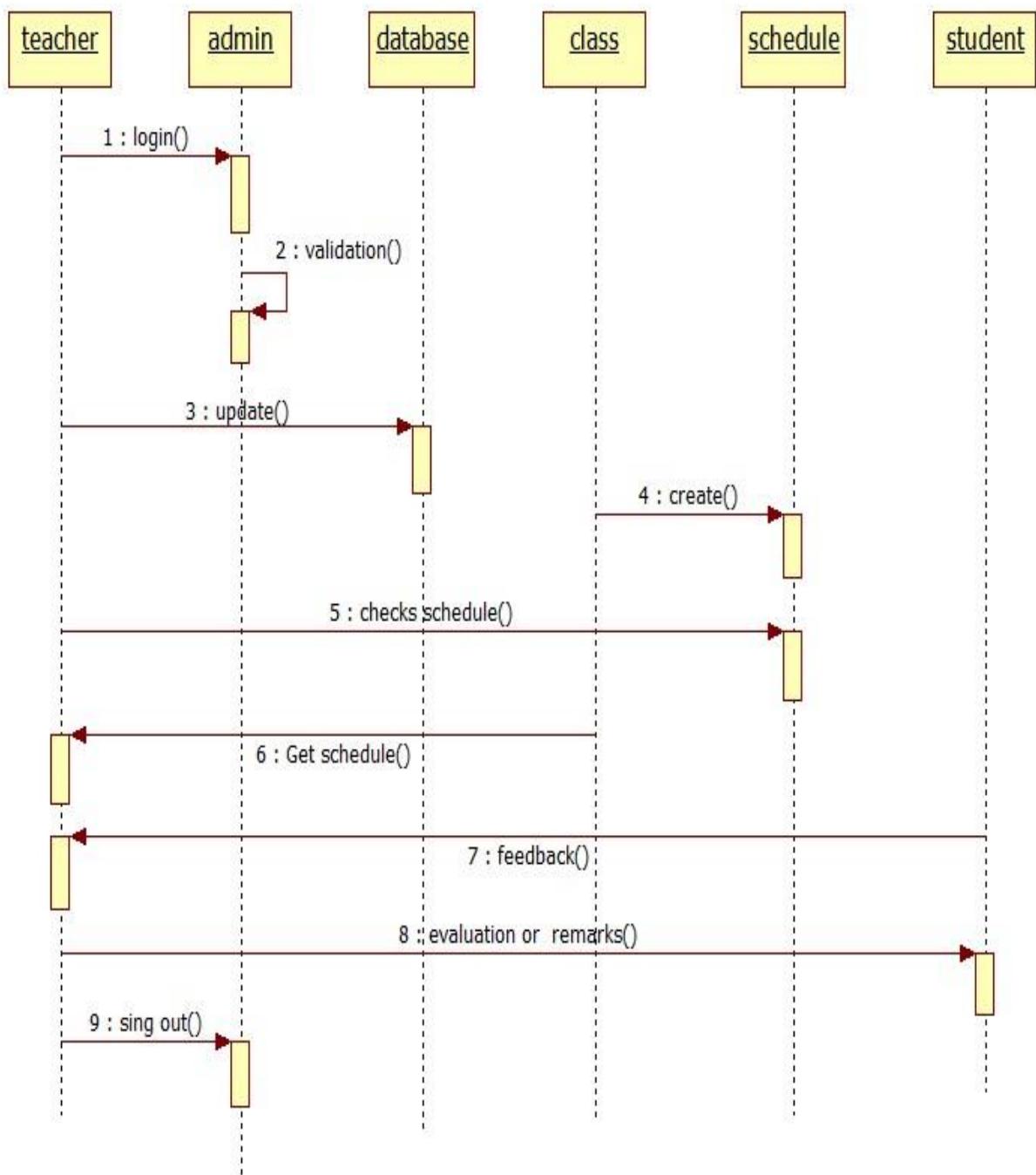


Figure 9 sequence diagram for teacher

3p2.2.4 sequence diagram for parent:

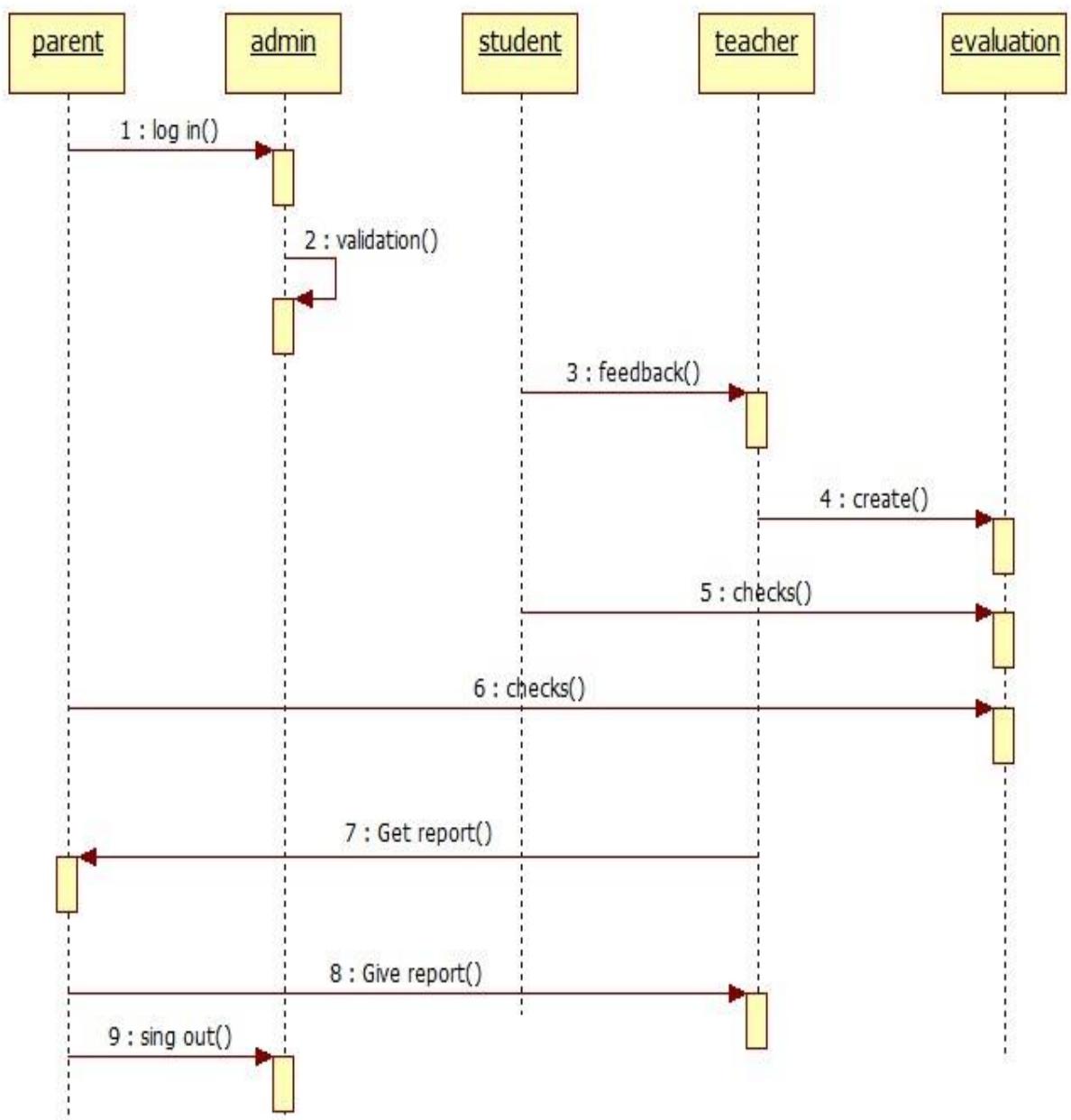


Figure 10 sequence diagram for parent

3p2.3 Activity Diagrams:

are graphical representations of workflows of stepwise activities and actions [1] with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows).

Activity diagrams show the overall flow of control.

Activity diagrams are constructed from a limited number of shapes, connected with arrows

The most important shape types:

rounded rectangles represent actions;

diamonds represent decisions;

bars represent the start (split) or end (join) of concurrent activities;

a black circle represents the start (initial state) of the workflow;

an encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen.

Activity diagrams may be regarded as a form of flowchart.

Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases;

the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

3p2.3.1 Activity Diagram for admin:

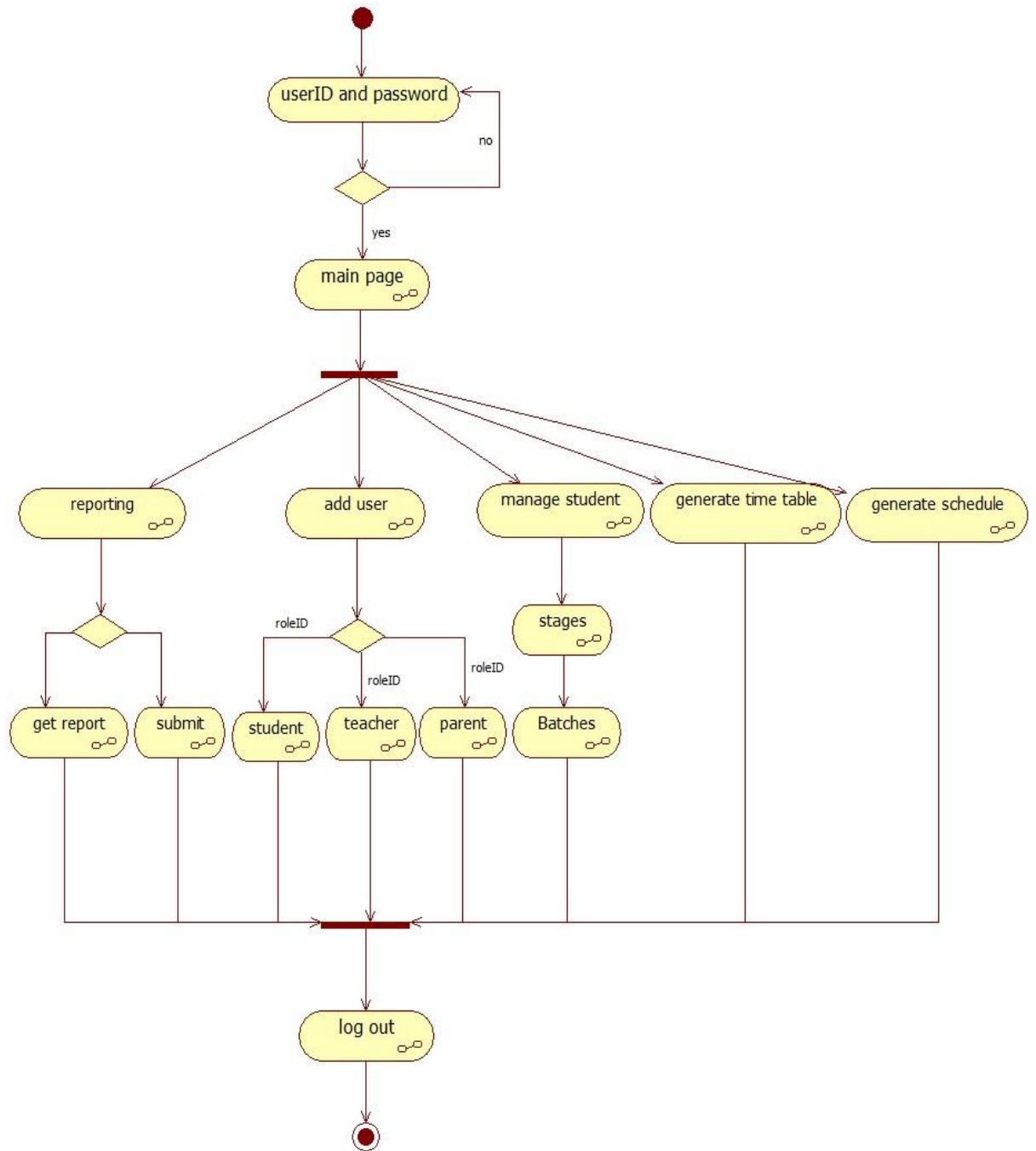


Figure 11 activity diagram for administrator

3p2.3.2 Activity Diagram for student:

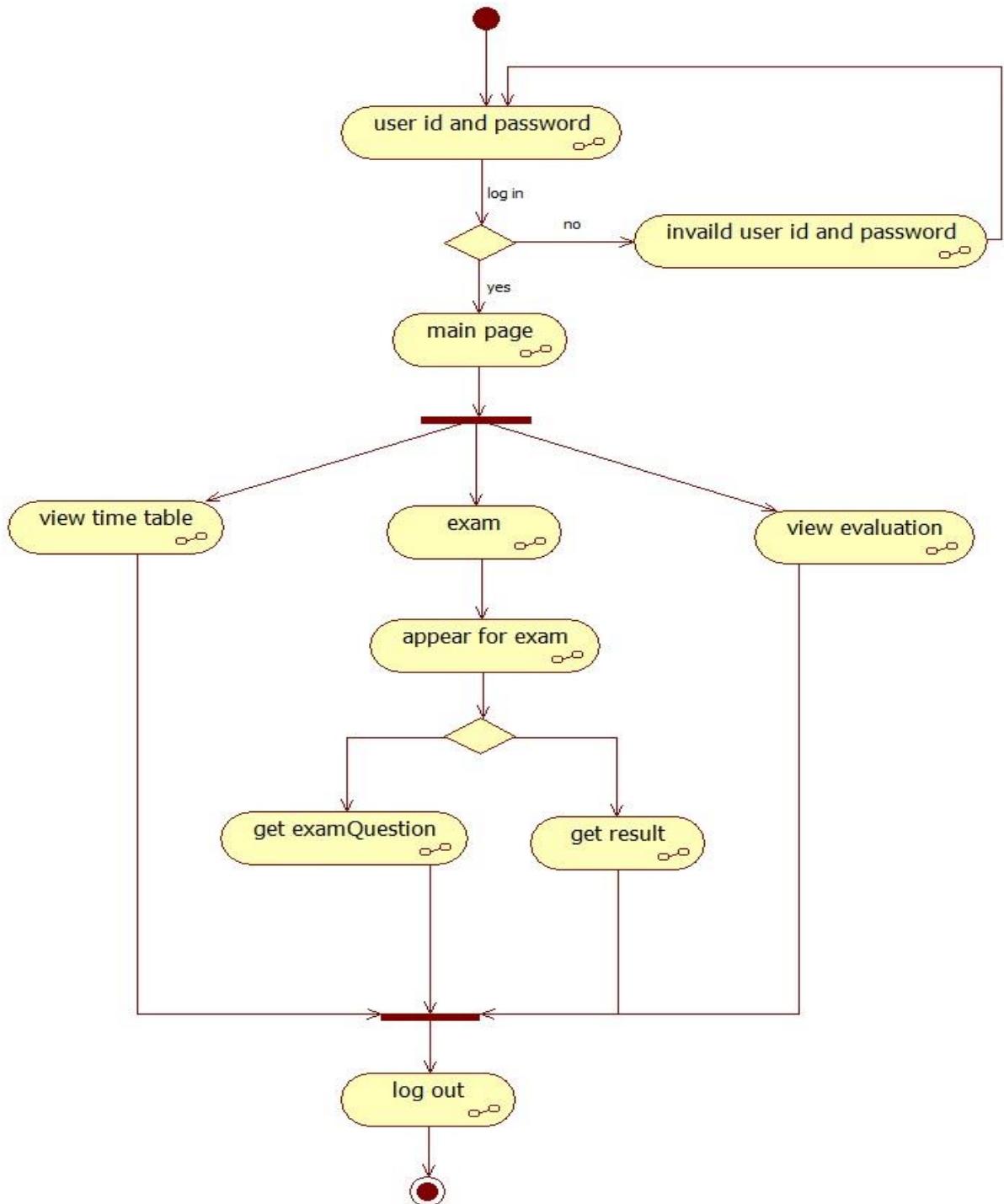


Figure 12 activity diagram for student

3p2.3.3 Activity Diagram for teacher:

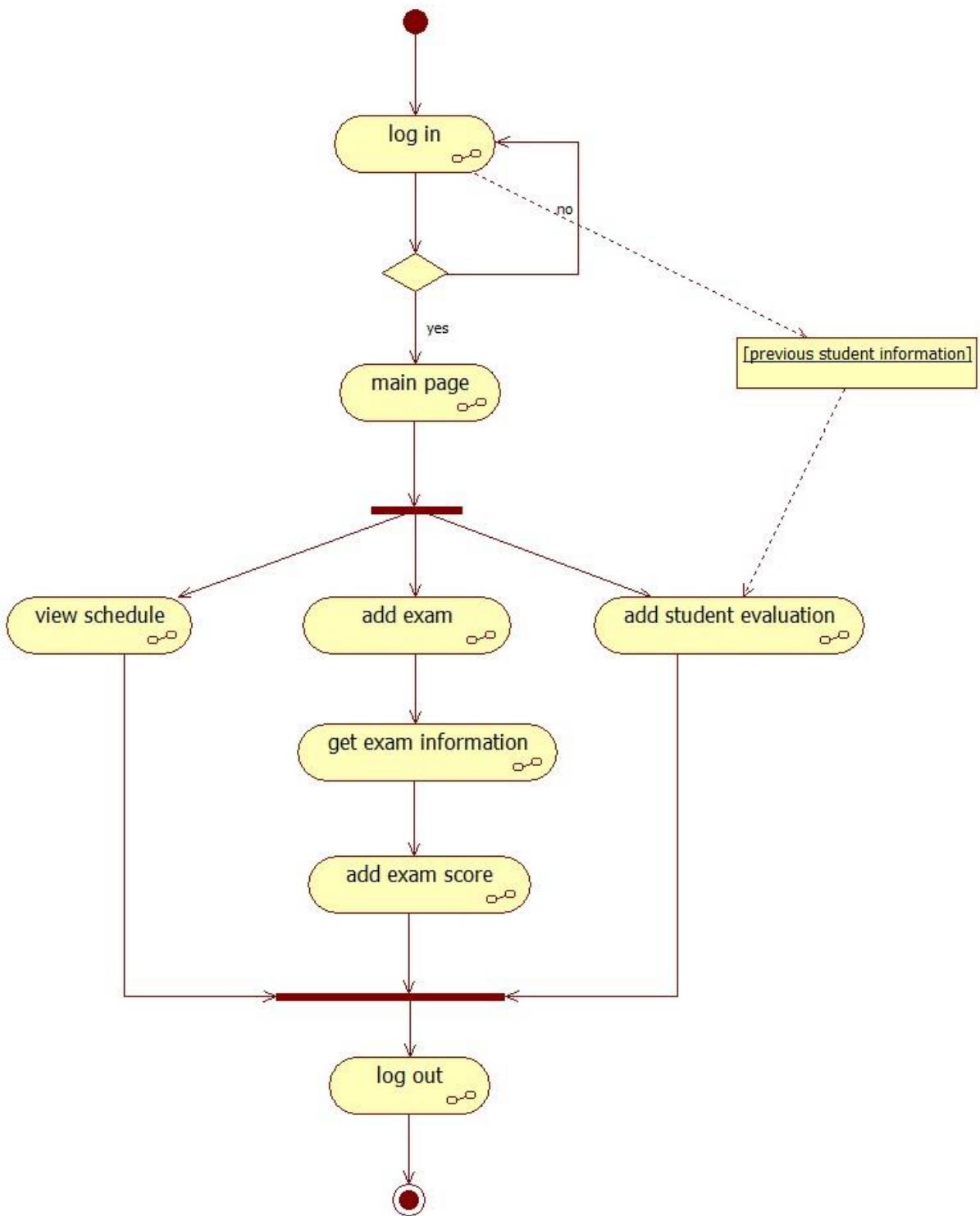


Figure 13 activity diagram for teacher

3p2.3.4 Activity Diagram for parent:

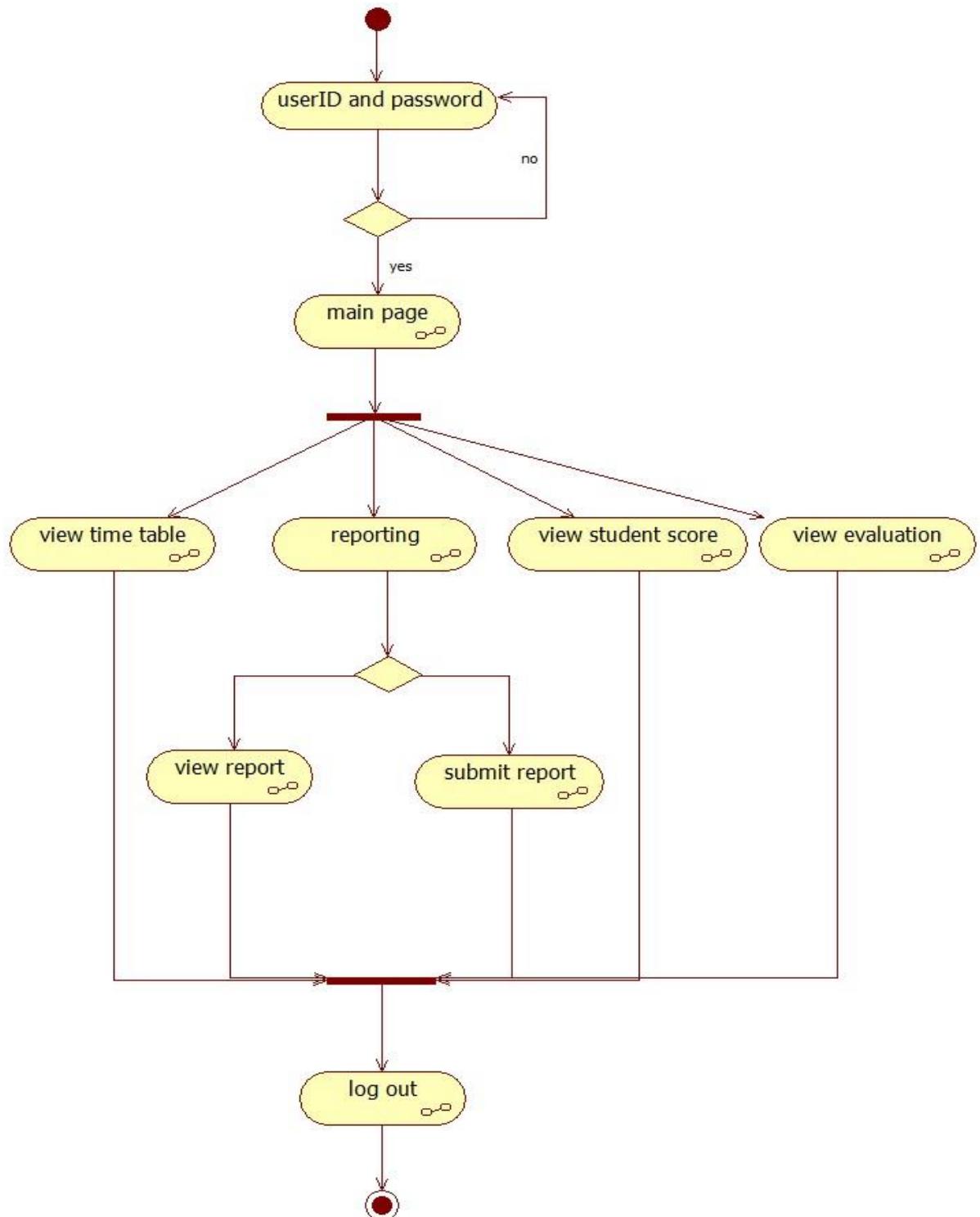


Figure 14 activity diagram for parent

Chapter4

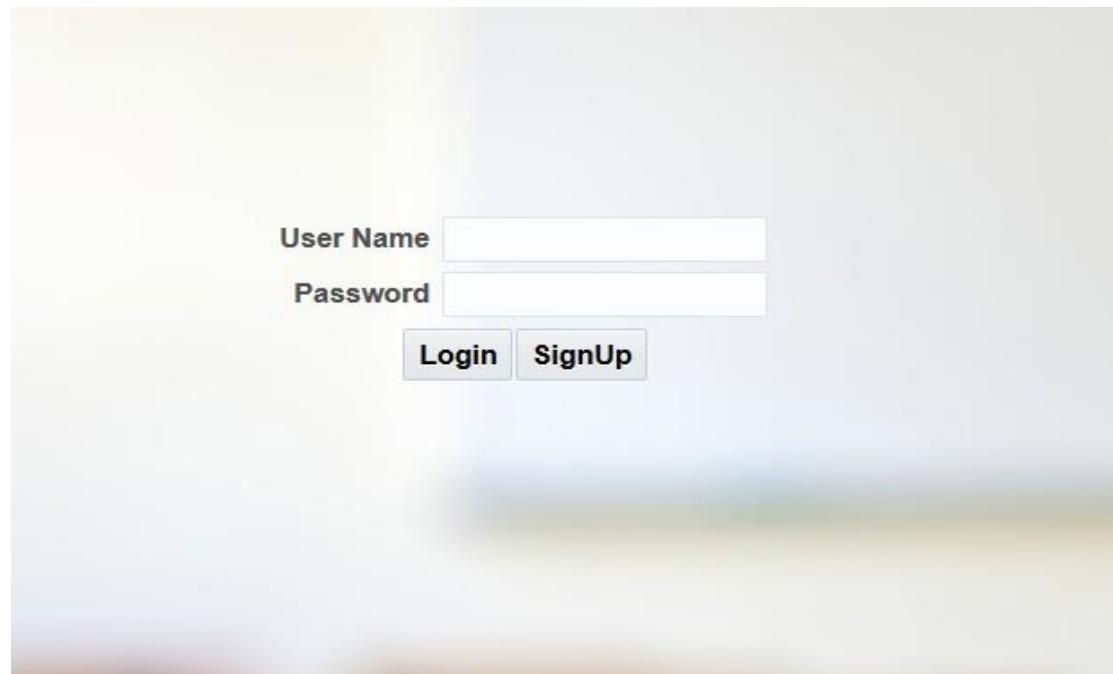
4.System Design

- This chapter discuss the system design and how to use system.

4.1 Login page

Which you can login as admin, teacher, student, and parent

Through your user name and password, you can login to the



system.

Figure 15 Log in page

4.1.1 Log in as admin:

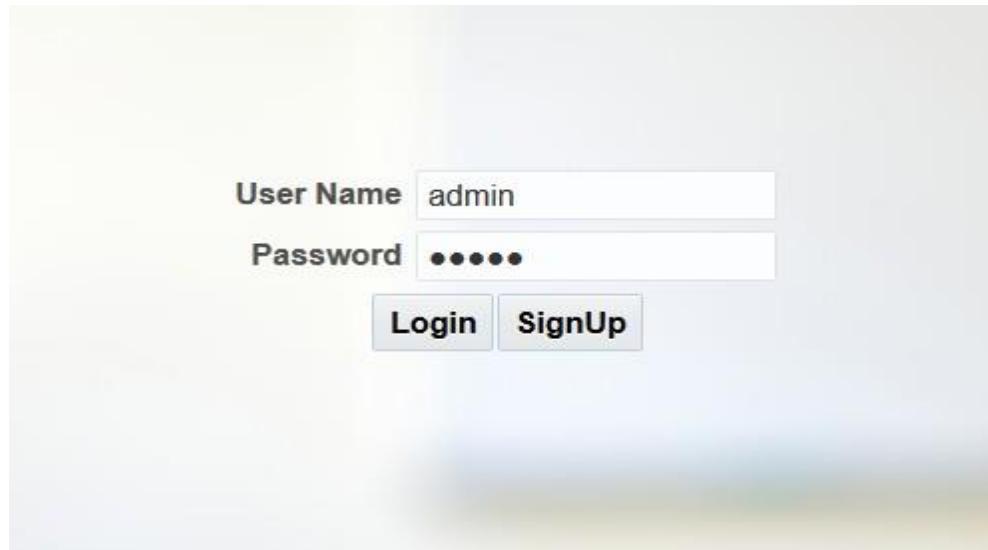


Figure 16 log in as admin

Now you can login as admin which has all privileges in this system.

[3 – Admin home page](#)

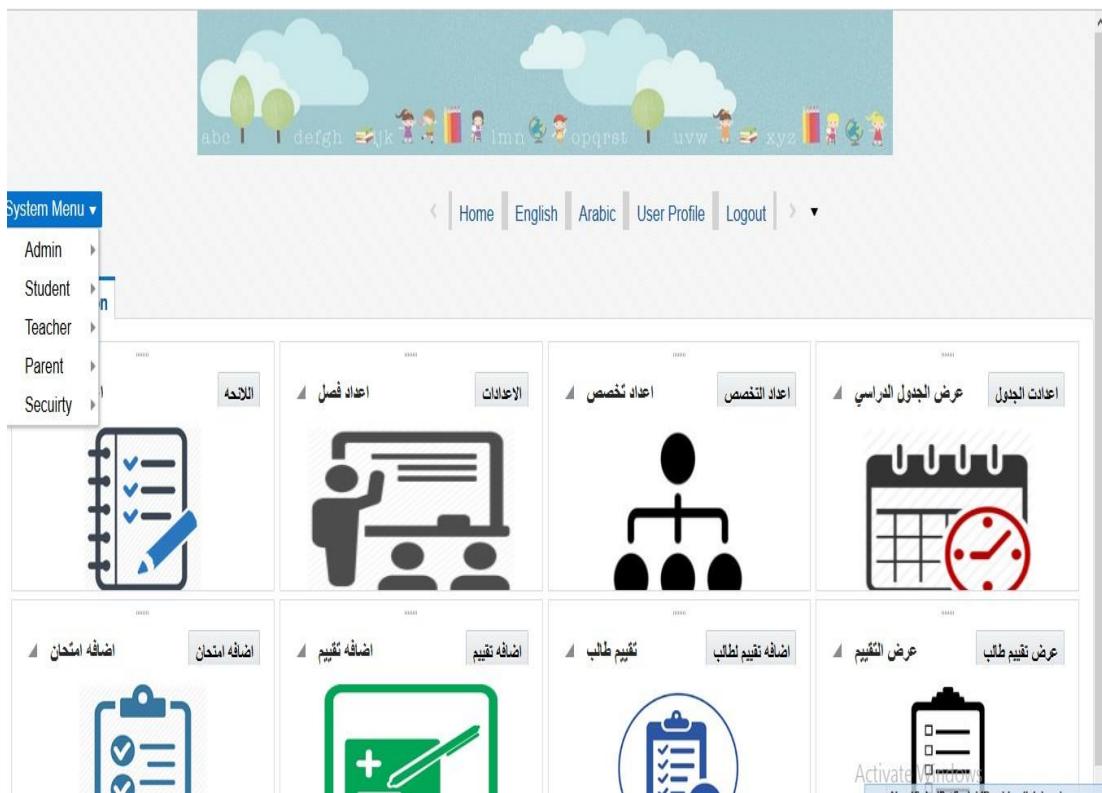


Figure 17 main menu of admin

- Through admin system menu can see all entities in the system.
- And through admin menu we will see all operations available

for admin:

1. Define the school list
2. Define Stages and its patches.
3. Define classes.
4. Define Fields.
5. Define the schedule.
6. Define parents for Students
7. Define Students setting
8. Define teachers and its subjects
9. Define Tests Types.
10. List of classes and Students.
11. List of teachers and subjects.
12. Add and define complaints.
13. Security:
 - a. Define rules of the system.

b. Create new account for user.

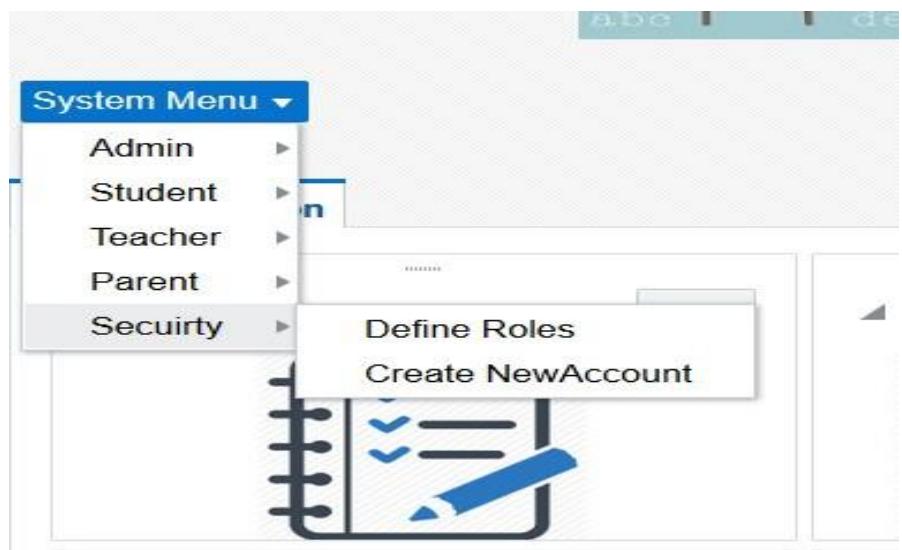


Figure 18 create new account for user



Figure 19 Admin operations

Admin operations:

3.1 Define the school list.

- Which admin can add, remove and edit school list.
- Admin can define subjects and fields for the school list.

3.2 Define Stages and its patches.

- Admin can add, delete and edit stage.
- Admin can add, delete and edit patch to Stage.

The screenshot shows a web-based administrative interface. At the top, there is a navigation bar with links for Home, English, Arabic, User Profile, and Logout. Below the navigation, the word "Stages" is displayed in bold. There are two tabs: "Search" and "Show Stages", with "Show Stages" being active. Below the tabs, there is a toolbar with buttons for View (dropdown), ADD, Edit, Delete, and Detach. A table follows, with columns for StageId and StageName. The data rows are: StageId 1, StageName Primary; StageId 2, StageName Preparatory; StageId 3, StageName Secondary. The second part of the screenshot shows a similar table structure for Batches. The toolbar buttons are Add, Edit, Delete, Save, and Detach. The table has columns for BatchId and BatchName. The data rows are: BatchId 11, BatchName First Primary; BatchId 12, BatchName Secand Primary; BatchId 13, BatchName Third Primary; BatchId 15, BatchName Fifth Primary.

StageId	StageName
1	Primary
2	Preparatory
3	Secondary

BatchId	BatchName
11	First Primary
12	Secand Primary
13	Third Primary
15	Fifth Primary

Figure 20 define stages and patches

3.3 Define classes.

- define classes for each patch.
- Admin can add, remove and edit classes for each patch.

System Menu ▾

Home English Arabic User Profile Logout

BatchesView

BatchId 11

BatchName First Primary

StageId 1

First Previous Next Last

Classes

View	Add	Edit	Delete	Save	Cancel	Detach
ClassId	ClassName	RoomNo	BatchId			
101	1-1	11	11			
102	1-2	12	11			

Figure 21 define classes

3.5 Define the schedule.

- Admin can define and edit the schedule.

The screenshot shows a web-based application interface for defining schedules. At the top, there's a decorative banner with letters and icons. Below it is a navigation bar with links for Home, English, Arabic, User Profile, and Logout. A 'System Menu' dropdown is also present.

The main area is titled 'Schedule' and contains a 'Search' section with fields for DayDate, TeacherId, and an 'Advanced' button. Below the search is a table with columns: ScheduleId, DayId, DayDate, ClassId, TeacherId, SubjectId, TimeFrom, TimeTo, and SchoolYearId. Two rows of data are visible:

ScheduleId	DayId	DayDate	ClassId	TeacherId	SubjectId	TimeFrom	TimeTo	SchoolYearId
1	Saturday	9/20/2014	1-1	Magdy	Arabic 1	8 : 00	10 : 00	2015-2016
2	Saturday	9/20/2014	1-1	Adel	English 1	10 : 30	11 : 00	2015-2016

Figure 22 define schedule

3.6 Define parents for Students

- Admin can add parents and its data for parents of students.

The screenshot shows a 'Parents' section of the application. It has a search interface with 'Add', 'Edit', and 'Delete' buttons. Below is a table with columns: ParentName, ParentDob, ParentAddress, ParentMobile, ParentEmail, ParentGender, ReligionId, and UserId. Two rows of data are visible:

ParentName	ParentDob	ParentAddress	ParentMobile	ParentEmail	ParentGender	ReligionId	UserId
Hassan	1/8/1970	Zagazig	Zagzig	hassan3432@...	Male	Muslim	5 Hassan
Salem	6/8/1966	Zagazig	Zagzig	salem2308@y...	Male	Muslim	6 Salem

At the bottom, a copyright notice reads 'All Right reserved'.

Figure 23 define parent for student

3.7 Define Students setting

- Admin can add, delete and edit student.
- Then can add parents for each student such Father, mother, uncle and brother.

The screenshot shows a web-based application interface for managing student settings. At the top, there is a navigation bar with links for System Menu, Home, English, Arabic, User Profile, and Logout. Below the navigation bar, the title "Student Setting" is displayed, along with "Cancel" and "Save" buttons. There are two main tabs: "Add Student" (which is currently selected) and "Parent View".

Add Student Tab:

StudentId	StudentName	StudentBirthPlace	StudentDob	StudentAddress	StudentMobile	StudentEmail	StudentGender	ReligionId	UserId
2	Mahmoud	Zagazig	6/4/2003	Zagazig	0118796544	mahmoud/2	Male	1	/
1	Hossam	Zagazig	6/12/2008	zagazig	0116576589	Hossam233	Male	1	4

Parent View Tab:

StudentParentId	StudentId	ParentId	ParentTypeId
103	1	Hassan	Father

Figure 24 define student settings

3.8 Define teachers and his subjects

- Admin can add, remove and edit teacher data.
- admin can add field and add teacher to this field.

System Menu ▾

Home English Arabic User Profile Logout ▾

Fields

Save Cancel

View ▾	Add New Filed	Delete Filed	Detach
FieldId	FieldName		
4	Science		
1	Arabic		
2	Math		
3	English		

Teacher Subject

View ▾	Add New Teacher	Delete Teacher	Detach						
TeacherId	TeacherName	TeacherMobile	TeacherDob	TeacherAddress	TeacherEmail	TeacherGender	TeacherReligion	TeacherFieldId	UserId
1	Magdy	0114785699	8/18/1970	Zagazig	Magdy23@c	Male	1	1	2

Figure 25 define teacher and his subjects

3.9 Define Tests Types.

- Admin can add, remove and edit type of tests.

The screenshot shows a web-based application with a decorative header featuring stylized trees and letters (abc, defgh, lmn, opqrst, uvw, xyz) along with small icons of children and books.

Navigation and User Options:

- System Menu ▾
- Home | English | Arabic | User Profile | Logout | ▾

Section Title:

أنواع الامتحانات

Search and Filter Options:

- Search
- ExamType
- Advanced
- Search | Reset

Action Buttons:

View ▾ Create New Delete Save Cancel Detach

Data Table:

ExamTypeId	ExamType
2	paper
3	oral
21	on Inetnet
41	Lab

Figure 26 define test types

3.10 List of classes and Students

- List of classes in the right
- Table of students which belongs to the classes in the left



System Menu ▾

Home English Arabic User Profile Logout ▾

101 1-1 11	View ▾	Detach
110 5-2 20		
111 6-1 21	StudentEnrollId	SchoolYearId
112 6-2 22	101	1
102 1-2 12		
103 2-1 13		
104 2-2 14		
105 3-1 15		
106 3-2 16		
107 4-1 17	View ▾	Detach
108 4-2 18		
109 5-1 19	StudentId	StudentName
113 1-1 23	1	Hossam
114 1-2 24		Zagazig
115 2-1 25		6/12/2008
116 2-2 26		zagazig
		01165765890
		Hossam23...
		1

Figure 27 list of classes and students

3.12 Add and define complaints.

- Admin can add complaints of students which have problems.

System Menu ▾

Home English Arabic User Profile Logout ▾

Complaints

Search **Advanced**

View ▾	Add New	Delete	Save	Cancel	Detach					
ComplaintId	Type	ComplaintFrom	EmployeeName	ComplaintTo	ParentName	Aboutstudent	StudentName	ComplaintsDate	Note	Status
12	Warning	1 Magdy C	Magdy	142 Salem	Salem	2 Mahmou	Mahmoud	10/17/2014	Mahmoud is	مُحتج

All Right reserved

Figure 28 add and define complaints

3.13 Security:

- A. Define rules of the system.
- B. Create new account for user.

The screenshot shows a web-based application interface. At the top, there is a decorative banner featuring stylized clouds and small illustrations of children and books. Below this is a navigation bar with links: 'System Menu ▾', 'Home', 'English', 'Arabic', 'User Profile', 'Logout', and a dropdown menu icon. The main content area is titled 'Roles'. It contains a search bar with a magnifying glass icon and an 'Advanced' link. Below the search bar is a table with the following data:

RoleId	RoleName
0	Admin
1	Teacher
2	Student
3	Parent

Figure 29 Security

View ▾ Add New Delete Save Cancel Detach						
UserId	UserName	UserPassword	UserCode	RoleId	CreationDate	
6	Salem	*****	Salem	Parent	6/28/2016	
7	Mahmoud	*****	mahmoud	Student	6/28/2016	
1	admin	*****	Demo	Admin	5/12/2016	
2	Magdy	*****	Magdy	Teacher	6/24/2016	
3	Adel	*****	mohamed	Teacher	6/18/2016	
4	Hussam	*****	hussam	Student	6/28/2016	
5	Hassan	*****	hassan	Parent	6/28/2016	

Figure 30 Security

4.1.2 Log in as Teacher:

Such that

User Name

Password

Figure 31 Log in as teacher

5 – Teacher home page

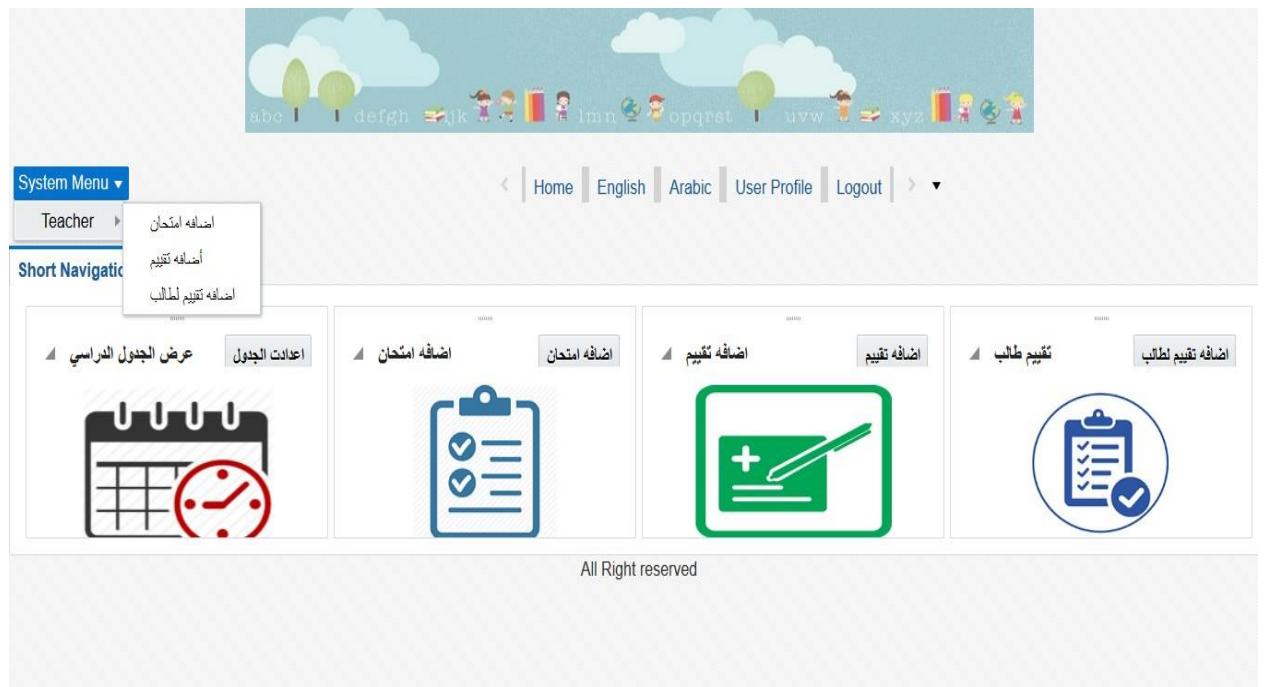


Figure 32 Teacher home page

Teacher operations that can do:

- 1- View Schedule
- 2- Add Exam
- 3- Add Exam Score
- 4- Add Student Evaluation

5.1 View Schedule

- Teacher can view schedule and search in it.

ScheduleId	DayId	DayDate	ClassId	TeacherId	SubjectId	TimeFrom	TimeTo	SchoolYearId
1	Saturday	9/20/2014	1-1	Magdy	Arabic 1	8:00	10:00	2015-2016
2	Saturday	9/20/2014	1-1	Adel	English 1	10:30	11:00	2015-2016

Figure 33 view schedule

5.2 Add Exam

- Teacher can add exams to its subjects

And define the exams data.

System Menu ▾

Home English Arabic User Profile Logout ▾

Arabic

View ▾ Detach

SubjectId	SubjectCode	SubjectName	SubjectMidDegree	SubjectTermDegree	SubjectSemester	SubjectExamHour	SubjectWeekHour	FieldId	SchoolListId
2	par2	Arabic 2	40	100	1	3	8	1	El_Farouk ...
1	par1	Arabic 1	40	100	1	3	8	1	El_Farouk ...

ExamId 122

* ExamName Arabic Quiz 2

* ExamTypeId paper

* ExamDate 11/29/2014

ExamTime 8 : 00

SubjectId 1

ExamQuestion

* TeacherId Magdy

Previous Next Add New Delete Save Cancel

All Right reserved

Activate Windows
Go to PC settings to activate Windows.

Figure 34 add an exam

5.4 Add Student Evaluation

- Teacher can add evaluation for students about them

Attend, solve tasks and exams

StudentEnrollViewEx

▶ Search Advanced

View ▾ Detach

ClassId	SchoolYearId	StudentEnrollId	StudentId
1-1	2015-09-19 08:00:00.0 2016-05-13 14:00:00.0	101	Hossam
1-2	2015-09-19 08:00:00.0 2016-05-13 14:00:00.0	102	Mahmoud

StudentsEvaluationView1

View ▾ **ADD New** Delete Save Cancel

EvaluationId	EvaluationItemId	StudentEnrollId	StudentEvalPerc	TeacherId
65	solve task	102	85	Magdy
66	attend	102	95	Magdy

Figure 35 add student evaluation

4.1.3 Log in as Student:

Such that

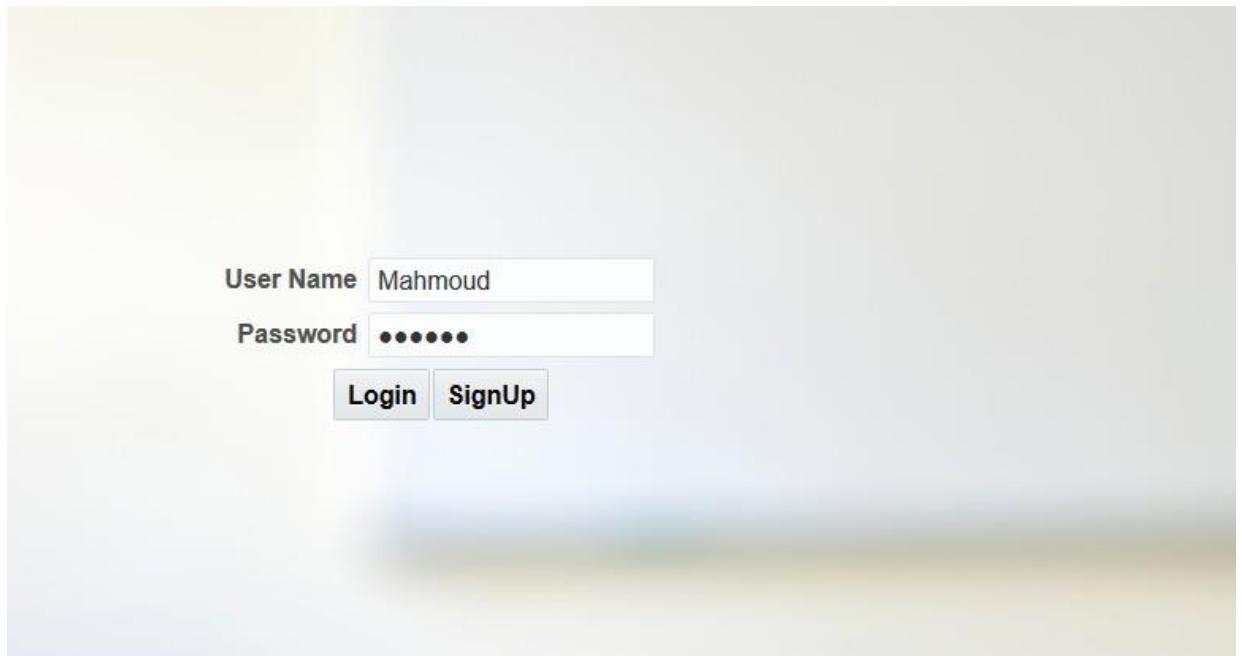


Figure 36 Log in as Student

7 - Student Home Page.

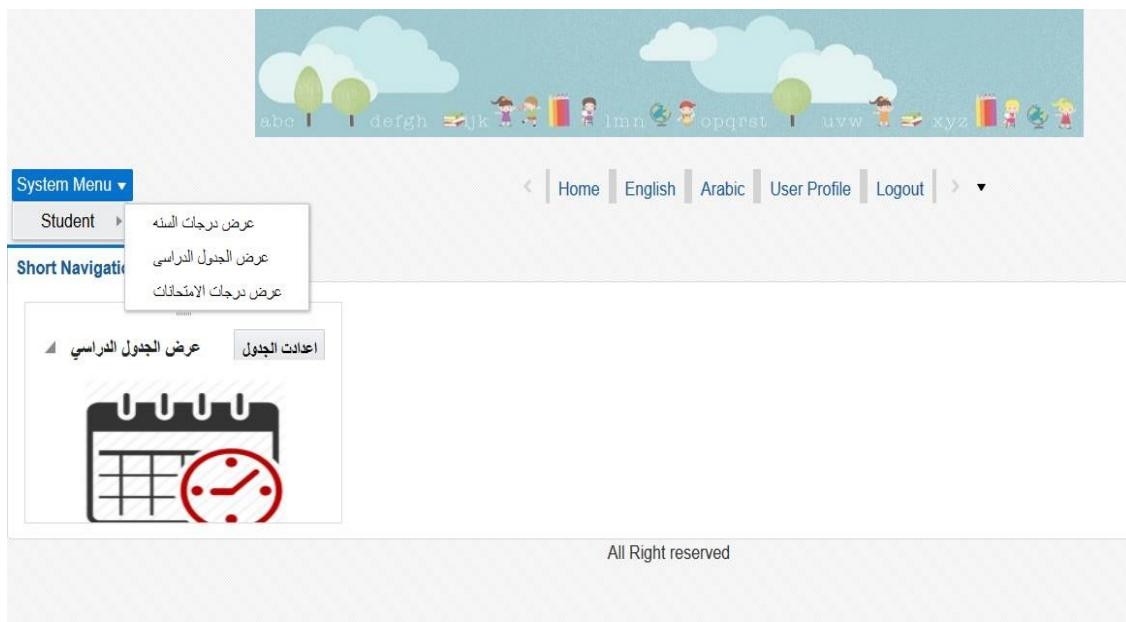


Figure 37 Student home page

Student operations that have:

- 1 - View Schedule
- 2 - View Exams Score
- 3 - View the score

7.1 View Schedule

- Student can view the schedule.

Schedule

Search Advanced

View	Add New	Delete	Save	Cancel	Detach			
ScheduleId	DayId	DayDate	ClassId	TeacherId	SubjectId	TimeFrom	TimeTo	SchoolYearId
1	Saturday	9/20/2014	1-1	Magdy	Arabic 1	8:00	10:00	2015-2016
2	Saturday	9/20/2014	1-1	Adel	English 1	10:30	11:00	2015-2016

Figure 38 view student schedule

7.2 View Exams Score

- Student can view its score in subjects that have exams in it.

The screenshot shows a web-based application for managing student enrollment. At the top, there is a decorative banner with clouds, trees, and children's icons. Below the banner, the navigation bar includes 'System Menu ▾', 'Home', 'English', 'Arabic', 'User Profile', 'Logout', and a dropdown menu. The main content area is titled 'StudentEnroll'. It displays two tables of data:

View			
StudentEnrollId	StudentId	ClassId	SchoolYearId
102	Mahmoud	1-2	2015-2016

View				
ExamStudentId	ExamId	StudentEnrollId	Score	Note
102	English Quiz	102	15	

Figure 39 view student score

4.1.4 Log in as Parent:

Such that

The screenshot shows a login form with a blurred background. The form fields are:

- User Name: Hassan
- Password: *****
- Buttons: Login and SignUp

Figure 40 Log in as parent

9 - Parent home Page



Figure 41parent home page

- Parent operations that have
 - 1 - View Schedule
 - 2 - View Evaluation
 - 3 - View Complaints

9.1 View Schedule

9.2 View Evaluation

- Parent can view evaluation of its student.

Figure 42 Parent view evaluation of its student.

9.3 - View Complaints

- Parent can view the complaints of its students
If there are.

The screenshot shows a web-based application interface. At the top, there is a decorative banner featuring stylized clouds, trees, and colorful icons representing letters and children. Below the banner, a navigation bar includes links for 'System Menu ▾', 'Home', 'English', 'Arabic', 'User Profile', and 'Logout'. A search bar with a magnifying glass icon and the text 'Search' is positioned above a table. The table has a header row with columns labeled: ComplaintId, Type, ComplaintFrom, EmployeeName, ComplaintTo, ParentName, Aboutstudent, StudentName, ComplaintsDate, Note, and Status. The status column contains Arabic text. A single row of data is visible in the table, corresponding to the header. At the bottom of the page, the text 'All Right reserved' is centered.

ComplaintId	Type	ComplaintFrom	EmployeeName	ComplaintTo	ParentName	Aboutstudent	StudentName	ComplaintsDate	Note	Status
12	Warning	1	Magdy	142	Salem	2	Mahmoud	10/17/2014	Mahmoud i...	مُتّوّجه

Figure 43 Parent view complaints

Chapter 5

5.Implementation

5.1 Oracle ADF Overview:

- Introduction
- Oracle ADF –Making Java EE Development Simpler
- THE ORACLE ADF ARCHITECTURE
- The Business Services Layer
- The Controller Layer
- The View Layer
- The Model Layer
- Productivity with Choice
- Declarative Customization and Personalization
- Integrated Security
- ORACLE ADF BENEFITS
- Visual and Declarative Java EE Development
- Benefits of Oracle ADF over Other Frameworks

5.2 Oracle Database:

- Introduction
- Schema
- Tables
- Oracle Database Features
- Oracle Database Application Development
- Oracle Database editions
- Oracle Database Versions

5.3 Technologies used to build Software:

5.4 Implementation code:

5.1.1 Introduction:

Java EE is a standard, robust, scalable, and secure platform that forms the basis for many of today's enterprise applications. Java EE provides a set of building multi-tier applications using the Java TM language. In the past, there was a direct correlation between the robust nature of an application to the complexity required to achieve it. However, with the advent of the Oracle ADF framework, you are able to provide the implementation of extremely rich Java EE applications, adhering to standard patterns and practices with greatly reduced effort.

Additionally, the increased need for organizations to build composite applications that utilize Service Oriented Architecture (SOA) principles has forced developers to create applications that are extremely agile. Implementing these best practices in agile applications usually involves writing a significant amount of infrastructure code, adding another obstacle for developers building their first Java EE application.

In addition to providing robust, performant, and maintainable applications; Oracle Application Development Framework also provides the best of breed infrastructure code to implement agile SOA based applications thereby removing the effort involved in an organization “rolling their own” and allowing a team to jump right versus building an infrastructure.

5.1.2 Oracle ADF –Making Java EE Development Simpler:

Oracle Application Development Framework (Oracle ADF) is an innovative, yet mature Java EE development framework available from Oracle and directly supported and enabled by the award winning development environment, Oracle J Developer 11g.

Oracle ADF simplifies Java EE development by minimizing the need to write code that implements the application's infrastructure allowing the developers to focus on the features of the actual application. Oracle ADF

provides these infrastructure implementations as part of the framework. To recognize a set of runtime services is not enough, Oracle ADF is also focused on the development experience to

provide a visual and declarative approach to Java EE development through the Oracle J Developer 11g development tool.

Oracle ADF implements the Model-View-Controller design pattern and offers an integrated solution that covers all the layers of this architecture with solution to such areas as: Object/Relational mapping, data persistence, reusable controller layer, rich Web user interface framework, data binding to UI, security and customization. Extending beyond the core Web based MVC approach, ADF also integrates with the Oracle SOA and Web Center Portal frameworks simplifying the creation of complete composite applications.

For example, Oracle ADF makes it easy to develop agile applications that expose data as services by coupling a service interface to the built-in business services in ADF. This separation of business service implementation details is performed in Oracle ADF via metadata. Use of this metadata-driven architecture enables application developers to focus on the business logic and user experience, rather than the details of how services are accessed.

Creating the user experience is as simple as dragging-and-dropping the desired business services onto a visual page designer and indicating what type of component should represent that data. In the example illustrated, we are able to take a database table exposed as a business service, and request J Developer to render the data as a table simply by dragging-and-dropping the control on the page and responding to the automatic popup by indicating a table as the desired rendering component. Oracle ADF takes care of the rest.

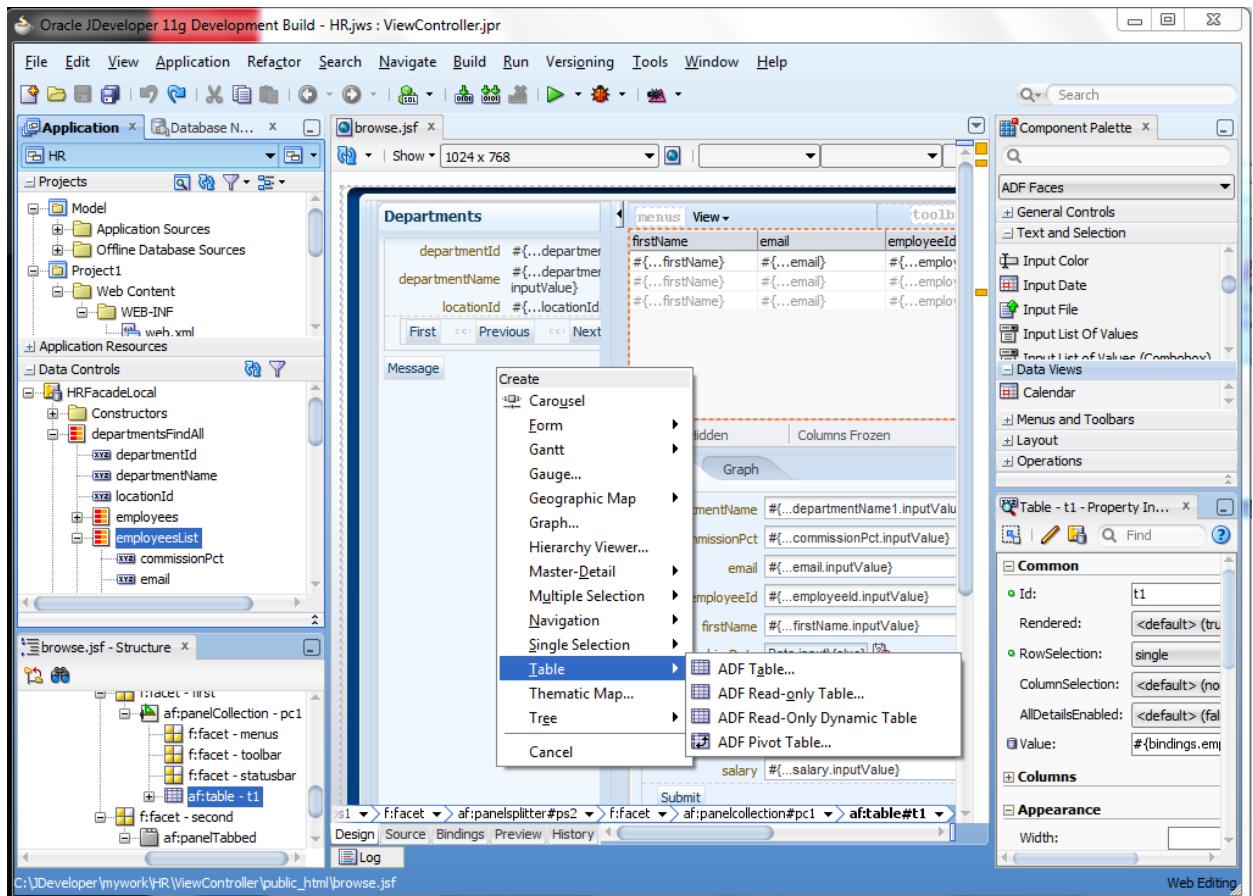


Figure 44 J Developer platform

Oracle ADF stores the implementation details of these services in metadata in the ADF Model layer. This enables developers to exchange services without modifying the user interface, making the application extremely agile. Additionally, the developer creating the user interface does not need to bother with business service access details. Instead they can focus on developing the application interface and interaction logic.

5.1.3 THE ORACLE ADF ARCHITECTURE:

Oracle ADF is based on the Model-View-Controller (MVC) design pattern. An MVC application is separated into: 1) a model layer that handles interaction with data-sources and runs the business logic, 2) a view layer that handles the application user interface, and 3) a controller

that manages the application flow and acts as the interface between the Model and the View layers.

Separating applications into these three layers simplifies maintenance and reuse of components across applications. The independence of each layer from the others results in a loosely coupled, Service Oriented Architecture (SOA).

Oracle ADF implements MVC and further separates the model layer from the business services to enable service-oriented development of applications. The Oracle ADF architecture is based on four layers:

- **The Business Services layer:**
Provides access to data from various sources and handles business logic.
- **The Model layer:**
Provides an abstraction layer on top of the Business Services layer, enabling the View and Controller layers to work with different implementations of Business Services in a consistent way.
- **The Controller layer:**
Provides a mechanism to control the flow of the Web application.
- **The View layer:**
Provides the user interface of the application

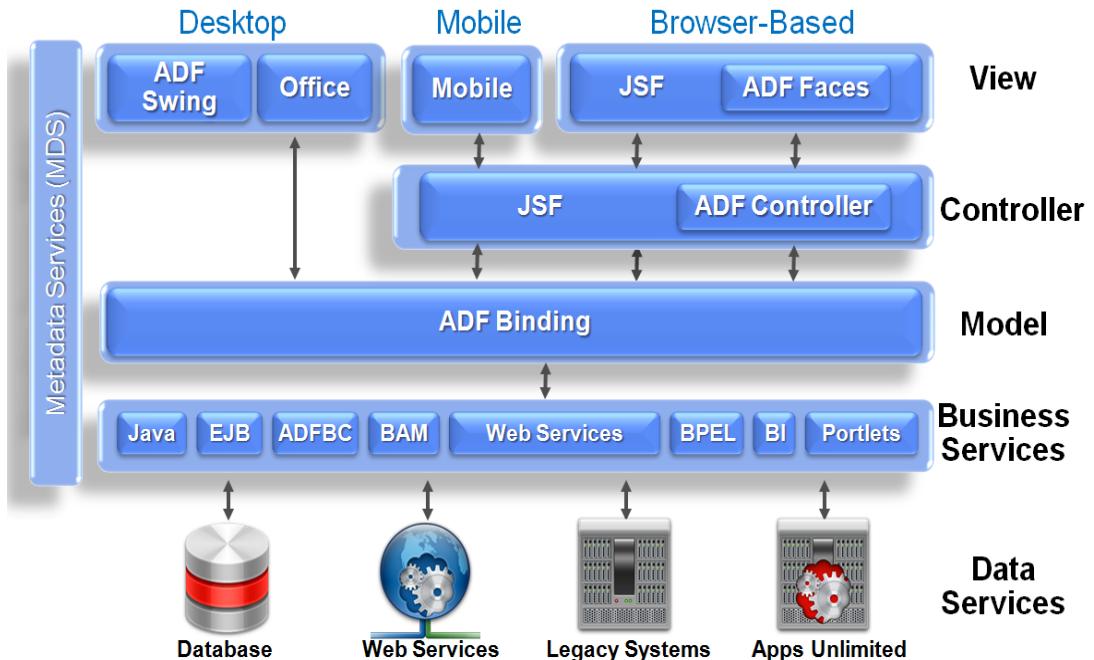


Figure 45 oracle desktop layers

Oracle ADF lets developers choose the technology they prefer to use when implementing each of the layers. The diagram above shows the various options available for developers when building Oracle ADF applications. The glue that integrates the various components of Java EE applications and makes development so flexible is the Oracle ADF model layer. EJB, Web Services, JavaBeans, JPA/Eclipse Link/Top Link objects and many others can all be used as Business Services for the Oracle ADF Model. View layers can include Web based interfaces implemented with JSF, Desktop Swing applications and MS Office front ends, as well as interfaces for mobile devices.

5.1.4 THE Business Services Layer:

The Business Services layer manages interaction with a data persistence layer. It provides such services as data persistence, object/relational mapping, transaction management, and business logic execution.

The Business Services layer in Oracle ADF can be implemented in any of the following options: As simple Java classes, EJB, Web services, JPA objects, and Oracle ADF Business Components. In addition, data can be consumed directly from files (XML or CSV) as well as REST.

5.1.5 THE Controller layer:

The controller layer manages the applications flow and handles user input. For example, when you click a Search button on a page, the controller determines what action to perform (do a search) and where to navigate to (the results page).

There are two controller options for web-based applications in JDeveloper: the standard JSF controller or the ADF Controller which extends the JSF controller functionality. Whichever controller you use, you will typically design your application flow by laying out pages and navigation rules on a diagram.

With the ADF controller you can break your application's flow into smaller, reusable task flows; include non-visual components such as method calls and decision points in your flow; and create "page fragment" flows that run inside a region of a single containing page. This approach encourages maximum reusability for user interface fragments and simplified integration into portals and mash up applications.

5.1.6 THE View Layer:

The View layer represents the user interface of the application.

Oracle ADF support multi-channel access to your business services allowing you to reuse your business services and access them from a Web client, a client-server swing desktop based application, Microsoft Excel spreadsheets, or a mobile device such as a smart-phone.

For Web based interface Oracle ADF offers a rich set of over a 150 Ajax enabled JSF components that simplified the creation of dynamic and appealing user interfaces.

5.1.7 THE Model Layer:

The model layer connects the business services to the objects that use them in the other layers. Oracle ADF provides a model layer implementation that sits on top of business services, providing a single

interface that can be used to access any type of business service. The model layer consists of two components, data controls and data bindings, which utilize metadata files to define the interface. Data controls abstract the business service implementation details from clients. Data bindings expose data control methods and attributes to UI components, providing a clean separation of the view and model. Due to the metadata architecture of the model layer, developers get the same development experience when binding any type of Business Service layer implementation to the View and Controller layers.

5.1.8 Productivity with choice:

Developers can choose different technologies to implement each of Oracle ADF ‘s layers, and still get the same productive development experience. For example, the same gestures and methods would be used to create an ADF Swing application based on ADF Business Components as would be used to create a ADF Faces application based on Enterprise JavaBeans. In addition to the choice of technologies implemented, developers can choose their development style (declarative, visual, or by coding), allowing developers with different skills and ways of working to cooperate on the same project.

5.1.9 Declarative Customization and Personalization:

A unique aspect of Oracle ADF is the ability to tailor the resulting application to specific user requirements. A layer called MDS – Metadata Services – allows developers to customize XML definitions that Oracle ADF relies on and use different versions for different users.

The first level of customization is enabled for end users at runtime – users can personalize the interface of their application - for example re-order

fields in a table, expand specific accordions etc. – and MDS will persist those changes for them.

The second level of customization, called seeded customization, allows developers to implement modifications in business rules, page flows, page layouts and other items that ADF persist in XML format. These modifications will be applied at runtime based on the user that logs into the application. This layered approach to customization allows an organization to use one base application to serve different customer needs.

5.1.10 Integrated Security:

Oracle ADF comes with built in security implementation across the layers. Developers can define users and roles and assign various authorizations to them. Privileges can be assigned at the business service layer, the controller layer or the UI layer. This integrated security framework across the layers eliminate the need to replicate security mechanisms at each layer.

For example, if a field in a business service is defined as non-updateable to some users, the UI layer will automatically switch the display of this field to be in read only mode in all screens without the developer needing to code this check at the UI layer.

5.1.11 Oracle ADF Benefits:

Visual and Declarative Java EE Development

A critical aspect of making a development framework useful is having a development tool that simplifies the creation of applications using this framework.

Oracle offers visual and declarative tools for each layer of Oracle ADF. These tools, which are integrated into the JDeveloper IDE, benefit Java developers even if they don't use the runtime features of Oracle ADF.

5.1.12 Business Services Development:

Oracle JDeveloper includes a variety of ways to construct business services including: EJB/JPA, web services, simple Java objects, and ADF

BC, among others. “Productivity with Choice” is a cornerstone to this approach. When generating these, it is possible to make use of a wizard-driven approach to generate Business Services that provide Java interfaces to these tables. With simply a right-click these interfaces can then be exposed as web services, including SDO based web services. Additionally, keeping with the theme of being visual and declarative, it is also possible to accomplish the same thing via visual modeling to generate these interfaces.

Oracle ADF Business Components is a framework focused on creating objects, which implement the Business Services layer on top of a data source, in a more declarative way. It provides out-of-the-box services such as transaction management, resource pooling, locking, declarative validation rules, translation, and object-relational mapping. Oracle ADF BC should feel familiar to developers with a background in 4gl declarative database driven development, offering such features as defining java objects based on SQL, declarative definition of validation rules, and pre-defined events where code can be injected into the business service life cycle. ADF BC development is done through declarative dialogs and property inspectors. With built-in implementation of common JAVA EE design patterns in the framework, the performance and scalability of the application is assured.

5.1.13 User Interface Development:

Visual and declarative development features of the View and Controller layers of an application are plentiful in Oracle JDeveloper:

- ADF Faces - a large set of over a 150 UI components built on top of the standard JSF APIs that leverage the latest technologies — including partial page rendering and Ajax — to provide a rich, interactive user interface.
- A page flow modeler for the ADF controller and the standard JSF framework page flow, providing visual page flow modeling using simple drag and drop of components onto a diagram.
- A visual editor for JSP, JSF, HTML, Swing, and Wireless based user interfaces, allowing WYSIWYG development for all types of components.

- Declarative development tools for adding components to the user interface, including the creation of declarative components, a property inspector, extensible component palette, and data control palette.
- Reusability features – several features for maximizing reusability, including the creation of task flows, ADF Libraries, and declarative components.

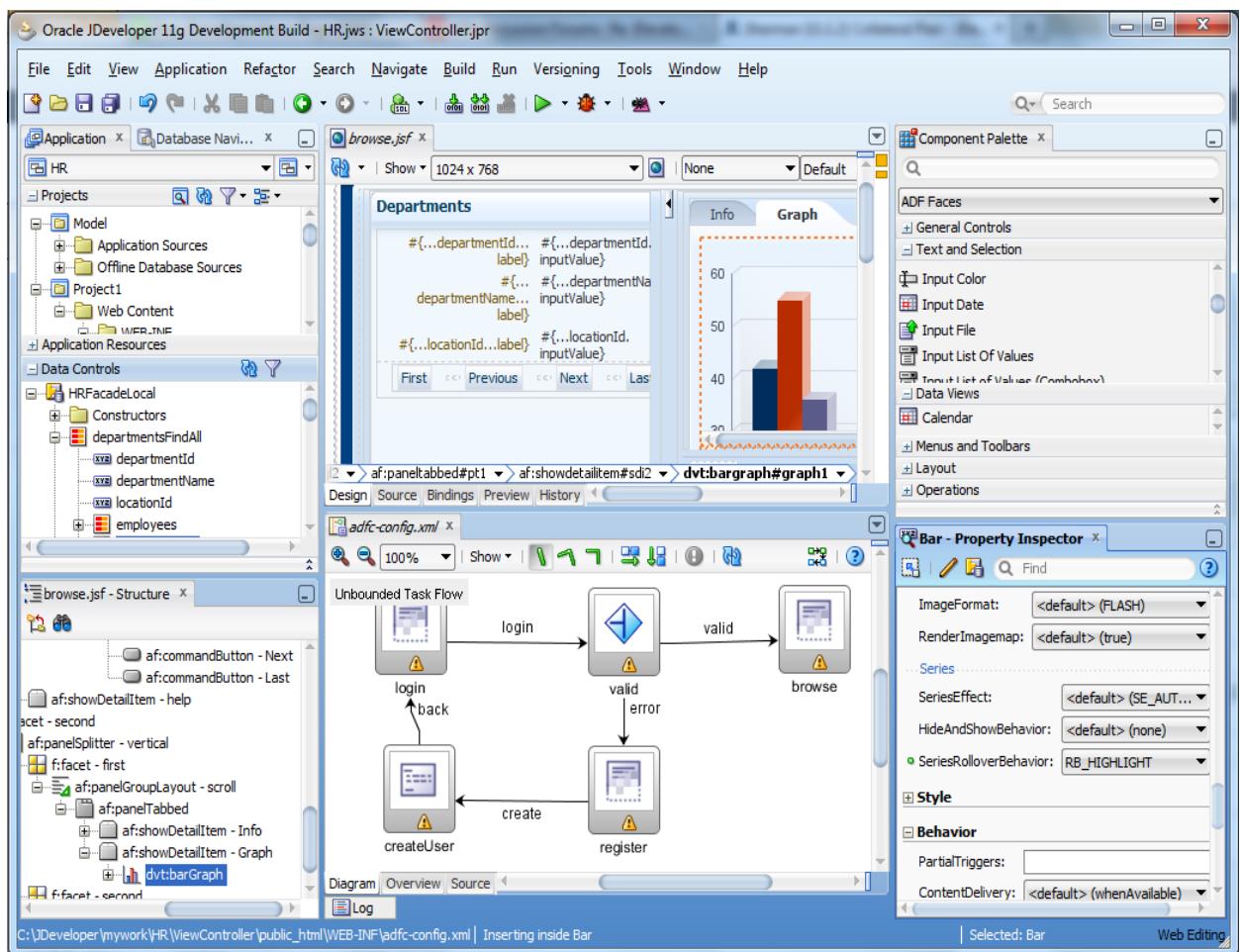


Figure 46 User Interface Development

The visual and declarative development tools are synchronized in the JDeveloper IDE so that the visual editor, property inspector, and modelers are synchronized with the source code at all times. Thus, developers can choose their development style – drag and drop, declaratively define properties, or edit source code directly.

Binding Business Services Components to the User Interface

Oracle JDeveloper provides a very easy way to bind components from the Business Services layer to your Controller and View layers using an innovative binding layer approach. The Data Control Palette provides a view into the Business Services layer. Developers can simply drag-and-drop data objects and bind them to their user interface implementation.

The same mechanism enables an easy binding of controller actions to methods defined in the Business Services layer. All using purely visual and declarative gestures.

5.1.14 Benefits of Oracle ADF over Other Frameworks:

The key characteristics of Oracle ADF that makes it unique among other Java EE frameworks are:

- [End-to-End Solution](#) – Oracle ADF doesn't focus on just one layer of the Java EE architecture. ADF provides an integrated and complete solution for every Java EE layer from the view layer and data-bindings, through the business services and data access; as well as support for every development life-cycle phase from inception through support.

- [Development Environment](#) – Many of the other Java EE frameworks lack strong integrated support by development tools. Oracle JDeveloper provides visual aids and a declarative approach to minimize the need to write framework code, making it a perfect tool for building Oracle ADF-based applications. This declarative development approach also reduces the learning curve for developers familiar with 4GL-style tools. Developers who wish to use another IDE, such as Eclipse, are able to do

so with built-in features provided in Oracle Enterprise Pack for Eclipse packaging and relying on ADF's support for the Java EE standards.

- Platform Independence – Other frameworks lock developers into a specific software vendor. The Oracle ADF runtime, however, can be installed on various Java EE compliant application servers and business services can connect to any SQL-92 compliant database.

- Technology Choice – Developers have preferences for the way they implement different layers of an application. Oracle ADF supports multiple technologies for each of the layers of the application and doesn't enforce a specific technology or a specific development style on the developer.

- Technology Commitment - It is important to note that Oracle ADF is the technology choice for the Oracle next generation set of enterprise -- applications – Oracle Fusion Applications - and is in continuous use for internal development

purposes. The product is used to develop Portal applications, wireless applications, and web applications, and therefore provides a committed, supported, and consistent technology stack.

- Metadata-Driven – All layers of the Oracle ADF framework offer declarative options for development, configured from XML metadata, while accommodating custom coding wherever necessary. You can choose to use all or part of the framework in the applications you build, making the application components much more reusable and flexible. The use of metadata also enables rules for data bound fields to be specified at the model layer. Labels, validation, and tooltip properties can be specified in the metadata for ADF data bindings - those properties are utilized independent of the user interface implementation.

- Declarative Customization – Oracle ADF provides a unique solution that allows an organization to use a single base application and customize it to fit the requirements of different users. Oracle ADF works in conjunction with a Metadata Services (MDS) layer that provides for application customization via two different implementation layers: The first, would be “seeded customization” which refers to an application wide customization that would be in effect for anyone accessing the application for a particular group. The second is “user customization” often referred to as “personalization” in which the end user designates

customizations to their personal experience that are then persisted via the MDS repository.

- [Enhanced Reusability](#) – JDeveloper + ADF provides support for superior reusability features including: JSF tinplating, reusable task flows, task flow tinplating, reusable business services, ADF libraries, JSF fragment based regions, and much, much more.
 - [Source availability](#) - Oracle provides the source code for the ADF framework to customers with a support license. Having the source available can help developers understand the underlying mechanisms of the framework and debug problems in their applications.
 - [Support](#) - Oracle ADF is an official Oracle product and as such is serviced by the Oracle Support organization. This provides around the clock support from an established organization.
 - [Training](#) - Oracle University offers regular instructor lead courses on Oracle ADF and JDeveloper.
-

5.2 Oracle Database:

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structure.

A **schema** is a collection of database objects. A schema is owned by a database user and has the same name as that user. Schema objects are the logical structures that directly refer to the database's data. Schema objects include structures like tables, views, and indexes. (There is no relationship between a table space and a schema. Objects in the same schema can be in different table spaces, and a table space can hold objects from different schemas.)

Some of the most common schema objects are defined in the following section.

[Tables](#) are the basic unit of data storage in an Oracle database. Database tables hold all user-accessible data. Each table has columns and rows. A table that has an employee database, for example, can have a column called employee number, and each row in that column is an employee's number.

[Indexes](#) are optional structures associated with tables. Indexes can be created to increase the performance of data retrieval. Just as the index in this manual helps you quickly locate specific information, an Oracle index provides an access path to table data.

When processing a request, Oracle can use some or all of the available indexes to locate the requested rows efficiently. Indexes are useful when applications frequently query a table for a range of rows (for example, all employees with a salary greater than 1000 dollars) or a specific row.

Indexes are created on one or more columns of a table. After it is created, an index is automatically maintained and used by Oracle. Changes to table data (such as adding new rows, updating rows, or deleting rows) are automatically incorporated into all relevant indexes with complete transparency to the users.

[Views](#) are customized presentations of data in one or more tables or other views. A view can also be considered a stored query. Views do not actually contain data. Rather, they derive their data from the tables on which they are based, referred to as the base tables of the views.

5.2.1 Oracle Database Features:

- Scalability and Performance Features
 - Manageability Features
 - Database Backup and Recovery Features
 - High Availability Features
 - Business Intelligence Features
-
- Content Management Features
 - Security Features
 - Data Integrity and Triggers
 - Information Integration Features
-

5.2.2 Oracle Database Application Development:

Oracle SQL

SQL (pronounced SEQUEL) is the programming language that defines and manipulates the database. SQL databases are relational databases, which means that data is stored in a set of simple relations.

PL/SQL

PL/SQL is Oracle's procedural language extension to SQL. PL/SQL combines the ease and flexibility of SQL with the procedural functionality of a structured programming language, such as IF ... THEN, WHILE, and LOOP.

5.2.3 Oracle Database Editions:

Express Edition

Is an entry-level, small-footprint database based on the Oracle Database 11g Release 2 code base. It is free to develop, deploy and distribute; fast to download; and simple to administer.

Enterprise Edition

Delivers scalability, security and reliability on a choice of clustered or single-servers running Windows, Linux and UNIX. It provides comprehensive features to easily manage the most demanding transaction processing, business intelligence and content management applications.

5.2.4 Oracle Database Versions:

- Oracle Database 10g.
 - Oracle Database 11g.
 - Oracle Database 12c.
-

5.3 Technologies used to build Software:

- Oracle ADF to application and UI.
- Oracle Database 11g

IDE:

- JDeveloper 12c
- Toad for oracle

5.4 Implementation code:

Implementation

Tables in toad is created in school schema

Img	Table	Schema	Tablespace	Last Analyzed	Num Rows
	BATCHES	SCHOOL	USERS	6/28/2016 11:15:58 PM	12
	CLASSES	SCHOOL	USERS	6/28/2016 11:15:21 PM	18
	COMPLAINTS	SCHOOL	USERS	6/28/2016 11:15:58 PM	
	EVALUATION_ITEMS	SCHOOL	USERS	6/28/2016 11:15:59 PM	3
	EXAM_TYPES	SCHOOL	USERS	6/28/2016 11:15:59 PM	4
	EXAMS	SCHOOL	USERS	6/29/2016 10:24:17 PM	3
	EXAMS_STUDENTS_SCORE	SCHOOL	USERS	6/29/2016 10:24:20 PM	4
	FIELDS	SCHOOL	USERS	6/28/2016 11:15:59 PM	4
	PARENT_TYPES	SCHOOL	USERS	6/26/2016 6:54:30 AM	6
	PARENTS	SCHOOL	USERS	6/28/2016 11:15:59 PM	2
	PS_TXN	SCHOOL	USERS	6/29/2016 10:24:20 PM	3
	RELIGION	SCHOOL	USERS	6/28/2016 11:16:00 PM	3
	ROLES	SCHOOL	USERS	6/28/2016 11:16:00 PM	4
	SCHEDULE	SCHOOL	USERS	6/28/2016 11:16:00 PM	2
	SCHOOL_LIST	SCHOOL	USERS	6/28/2016 11:16:02 PM	2
	SCHOOL_YEAR	SCHOOL	USERS	6/28/2016 11:16:02 PM	2
	STAGES	SCHOOL	USERS	6/28/2016 11:16:19 PM	3
	STUDENT_ENROLL	SCHOOL	USERS	6/29/2016 10:24:21 PM	2
	STUDENT_EXAM_SCORE	SCHOOL	USERS	6/29/2016 10:25:30 PM	
	STUDENTS	SCHOOL	USERS	6/28/2016 11:16:02 PM	2
	STUDENTS_EVALUATION	SCHOOL	USERS	6/29/2016 10:24:21 PM	4
	STUDENTS_PARENTS	SCHOOL	USERS	6/28/2016 11:16:02 PM	2
	SUBJECTS	SCHOOL	USERS	6/28/2016 11:16:03 PM	5
	SUBJECTS_BATCHES	SCHOOL	USERS	6/28/2016 11:16:03 PM	
	TEACHERS	SCHOOL	USERS	6/28/2016 11:16:03 PM	2
	USERS	SCHOOL	USERS	6/28/2016 11:16:03 PM	7
	WEEK_DAYS	SCHOOL	USERS	6/26/2016 6:54:31 AM	7

Figure 47 tables in toad which create school schema

- Oracle ADF Application



Figure 48 Oracle ADF APP

- Model -business component.



Figure 49 Model of business component

- Entity Object – which have structure of tables.

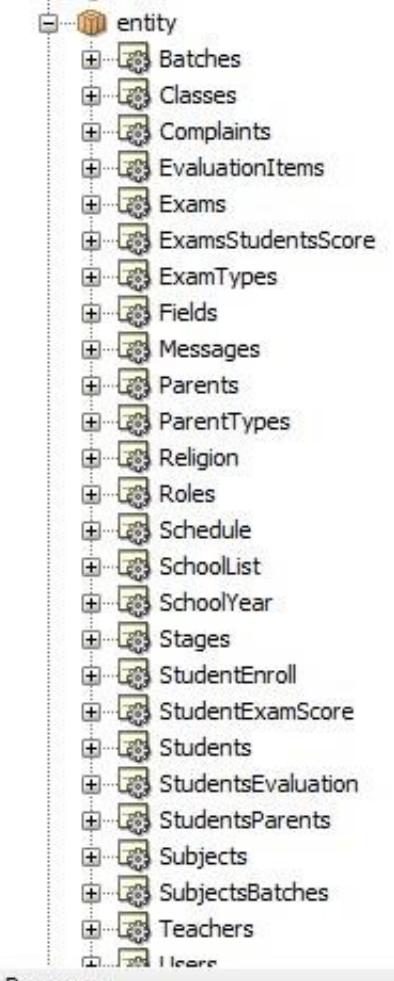


Figure 50 Entity Object

- View Objects – which have database queries

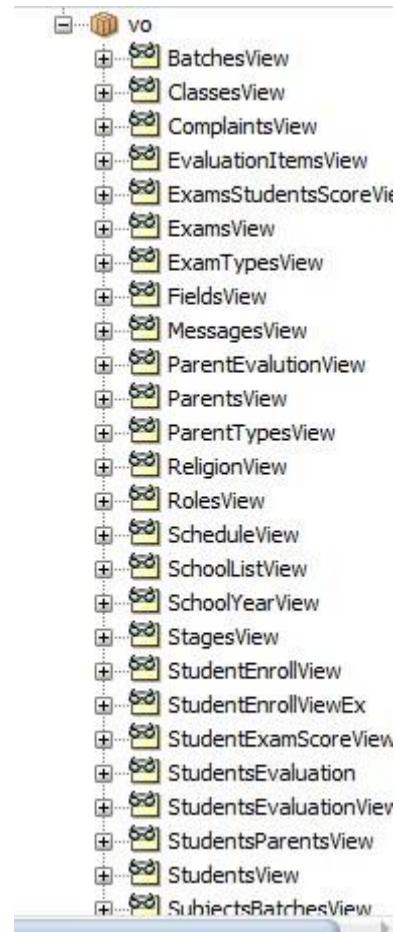


Figure 51 View Objects

- AM Application Module: which responsible about DB transaction.

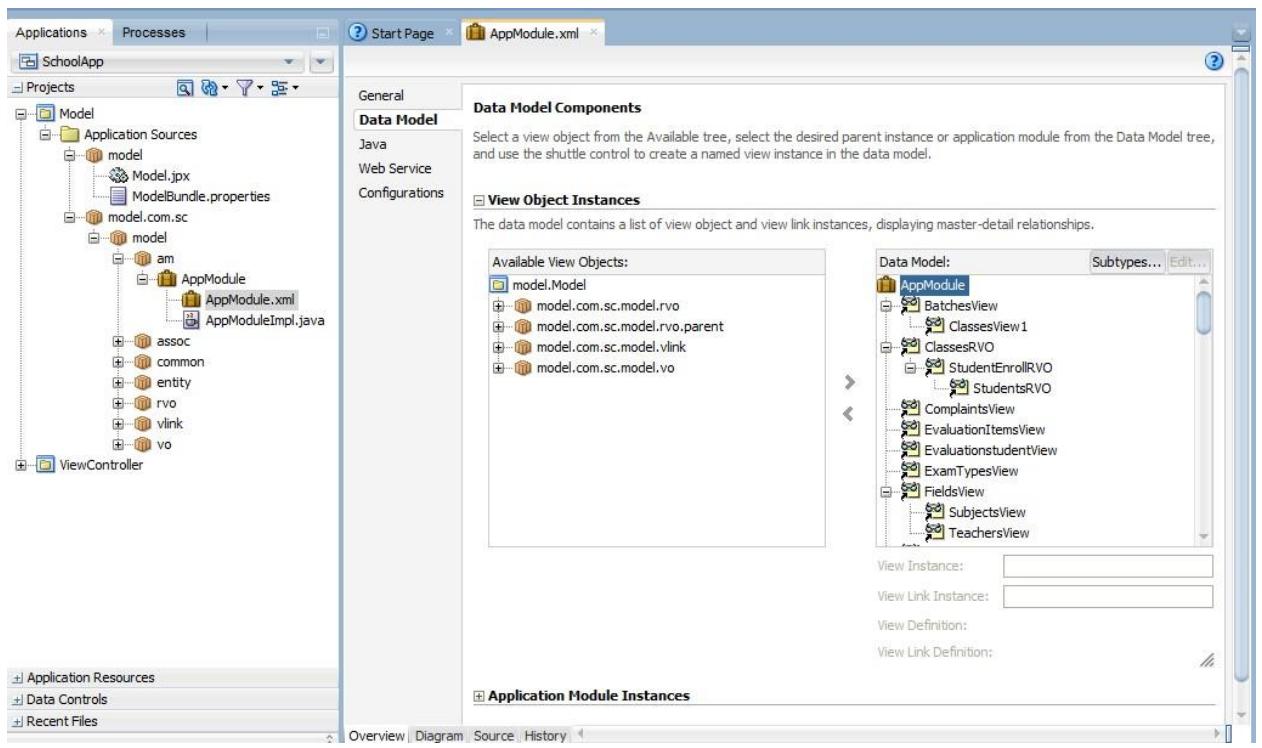


Figure 52 AM APP Model

- Association – has the relations between tables PK – PK.

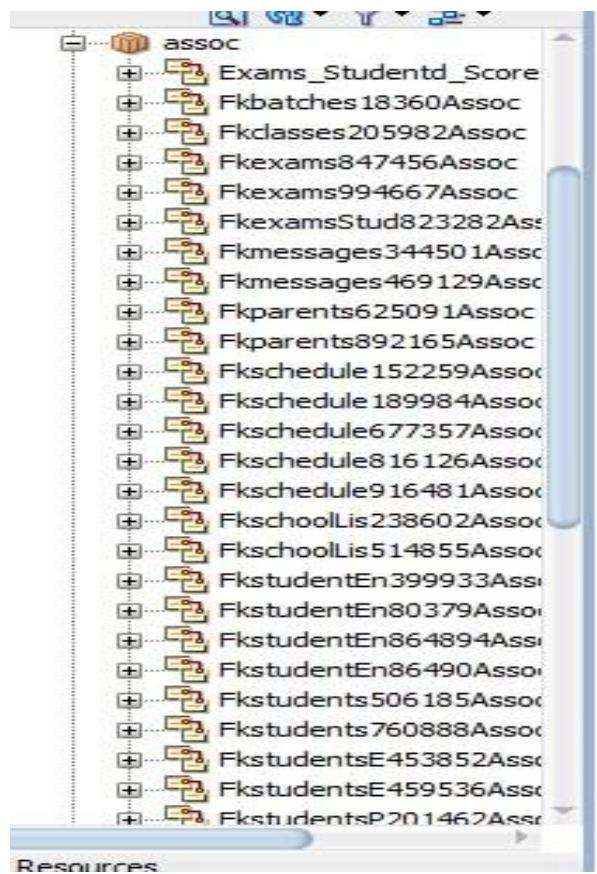


Figure 53 Association

- View Link: which responsible for master and details

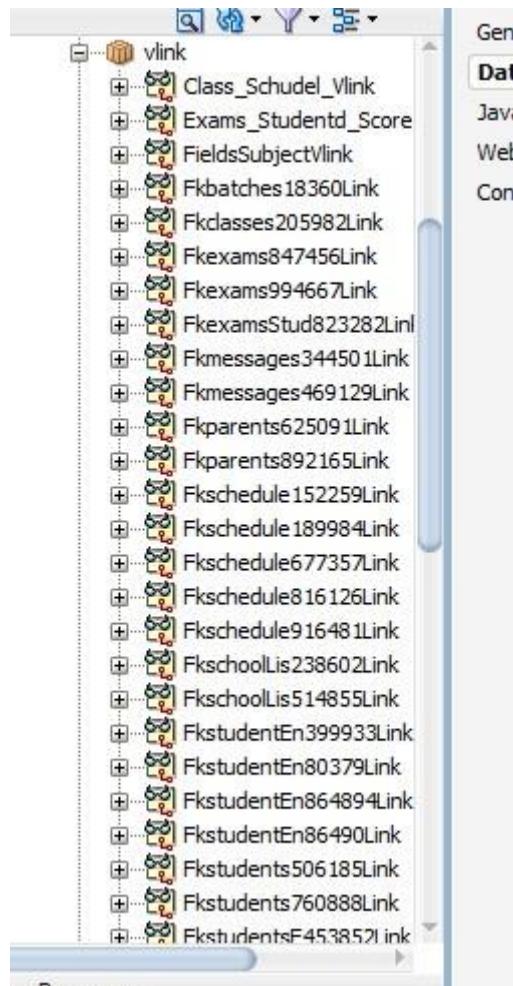


Figure 54 View Links

- Web Pages:

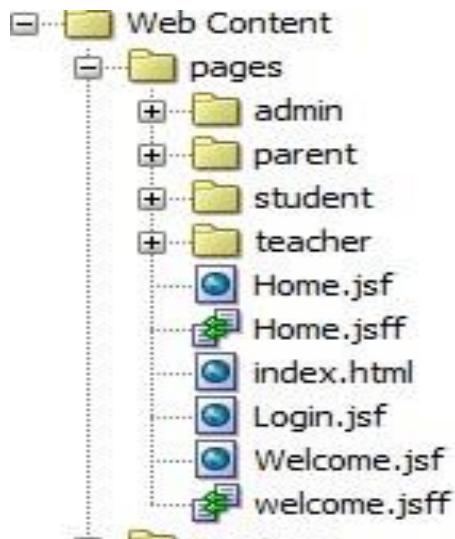


Figure 55 web pages

- Admin Pages:

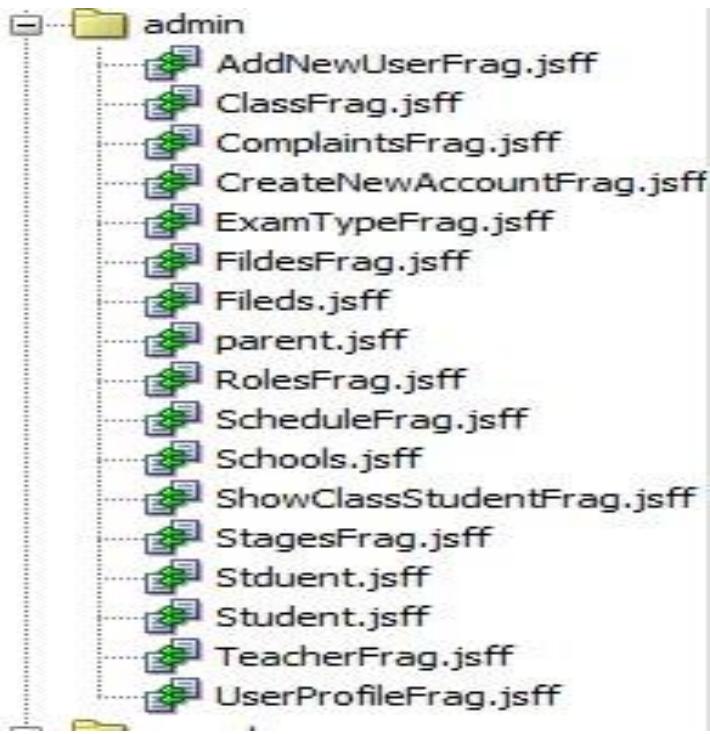


Figure 56 Admin Pages

- Student, Parent and Teacher pages

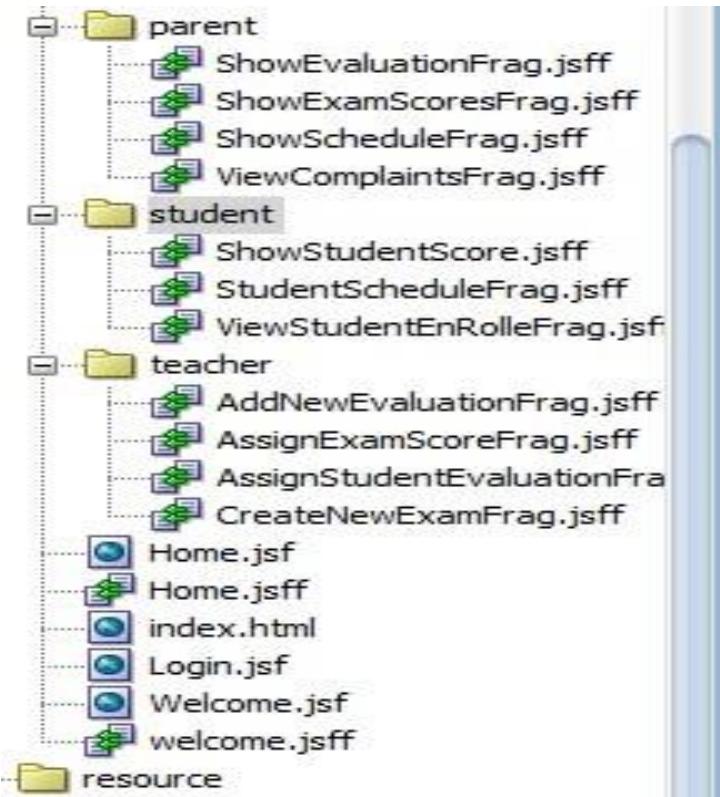


Figure 57 Student, parent, teacher pages

- Task Flow: adfc-config

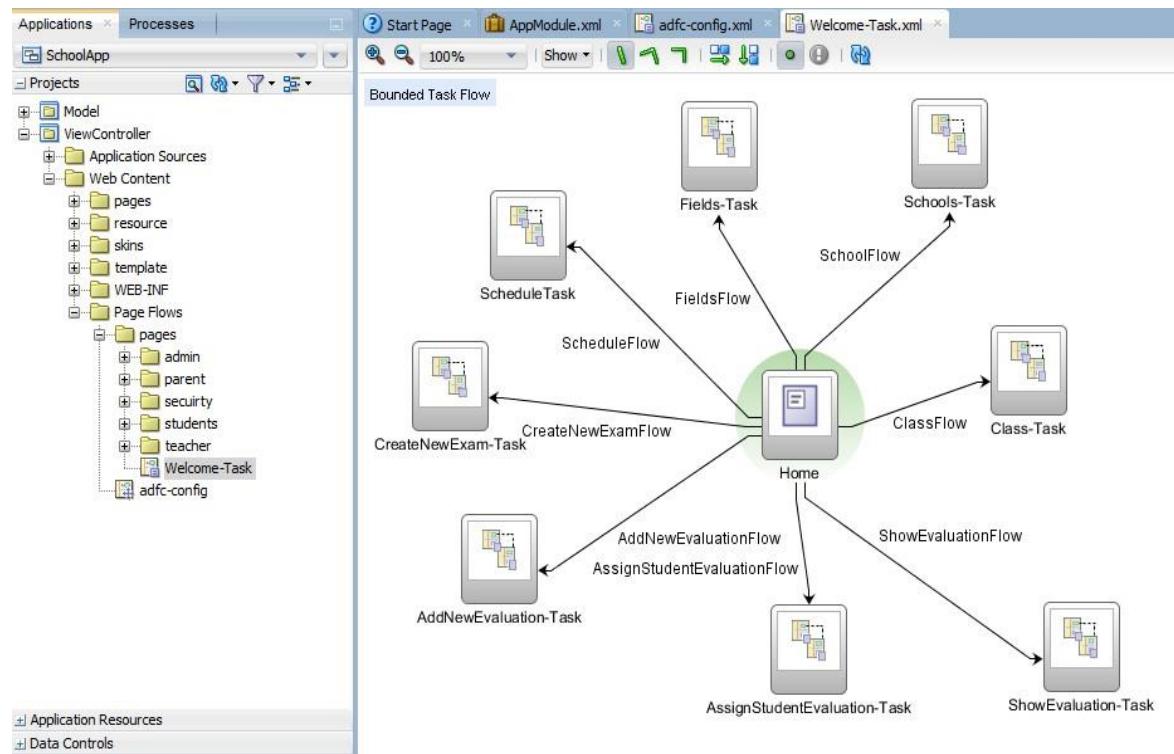


Figure 58 Task Flow

- DB Connection establish

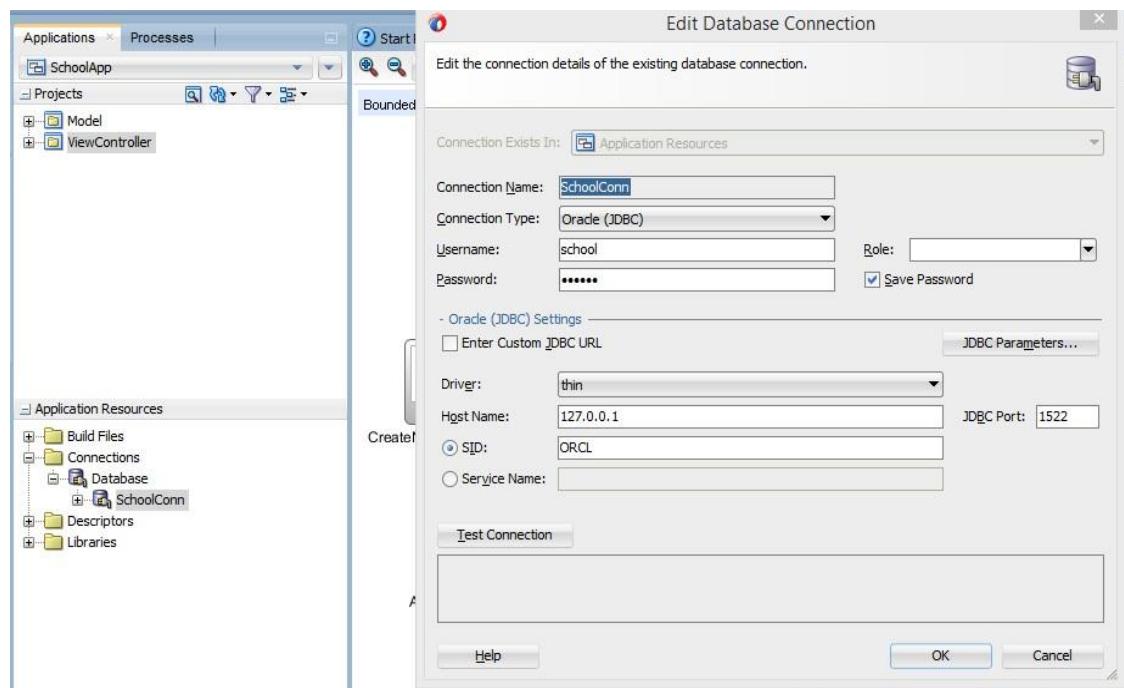


Figure 59 DB Connection

- Data Control

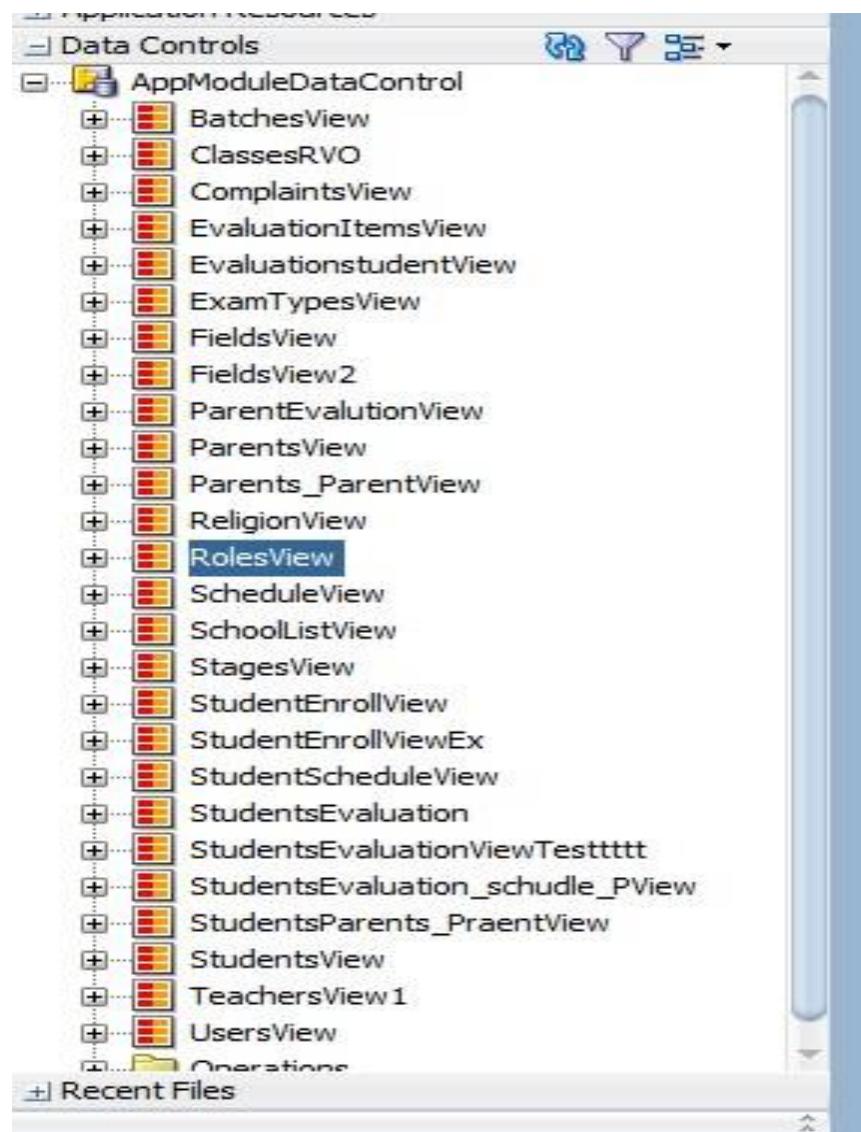


Figure 60 Data Control

6. Conclusions

- There has become a system that can work in any school putting Link of server in any browser and log in through writing your (User Name & Password).
- There will be a use of time and not wasted.
- The lack of manual mistakes.
- Ease of communication between members of the system.
- Lack of fatigue.
- Low cost.

7.Future Work

Make online exam more effective, efficient and more dynamic so that it helps to get a good support from the student.

It has been marking while this application opens in other browser due to designing support. It's more support to Mozilla as compare to other browser.

Online result printing and more user interaction functionality and features need to be developed.

Needs to improve more security for the purpose of the safety.

Convert this web app to mobile app

References

Referenced Sites:

<http://books.google.com>

<http://sourceforge.net>

<https://creately.com>

<https://en.wikipedia.org>

<https://www.oracle.com>

<http://www.oracle.com/technetwork/developer-tools/adf/overview/adfessentials-1719844.html>

<http://www.oracle.com/technetwork/developer-tools/adf/overview/index.html;jsessionid=6R39V8WhqTQ7HMb2vTQTkzbP5XRFgs4RQzyxQ7fqxH9y6p6vKXk4!-460884186>

<http://software.dell.com/products/toad-for-oracle/>

Referenced books:

---1st Rank free project download site

-----ERP documentation by **Gandhi Amit Kumar K**

The End
