
Software Design Description

Version 1.0

May 10, 2013

mini

Schedule Auto-Generator system

Fady Aziz

Nesma Talaat Abbas

Kiro Magdy

Maram El-Sayed

Yaser Abdelaziz

Submitted in partial fulfillment Of the requirements of CS
251 Software Engineering -1

Contents

1	Introduction	4
1.1	Purpose	4
1.2	Scope of Project	4
1.3	Glossary	4
1.4	References	5
1.5	Overview of Document	5
2	Test scenarios and Test Cases	7
3	Deployment Design	10
4	Architectural Design	11
4.1	find_by_id	11
4.2	open_connection	12
4.3	validate	13
4.4	viewtable	13
4.5	newsfeed	14
5	Data Structure Design	15
5.1	TimeTable Database	15
6	Real-Time Design	18
7	User Design	19
7.1	Student User Interface	19
7.2	Student User Interface	19
8	Help Design	20

List of Figures

3.1	Deployment of System Environment.	10
4.1	Architectural of System Environment.	11
8.1	Class Diagram of System Environment.	20
8.2	ERD Diagram of System Environment.	21
8.3	Activity Diagram of System Environment.	22
8.4	State Diagram of System Environment.	23
8.5	Access and post information Diagram.	23
8.6	Online voting Diagram.	24
8.7	View schedule Diagram.	24
8.8	State Diagram of System Environment.	24
8.9	State Diagram of System Environment.	25

1 Introduction

1.1 Purpose

This document includes the complete description of the generic timetable; this includes the architectural features of the system down through details of what operations each code module will perform and the database layout. It also shows how the use cases detailed in the SRS will be implemented in the system using this design.

The primary audiences of this document are the software developers.

1.2 Scope of Project

This software system should generate all the optimal solutions for a college shedule depending On its resources, the system designed to minimize conflicts between rooms, professor time and student group place, also it makes sure that number of seats at the room sufficient for number of students at each group. The system provides a control panel for the Users with different tools that meet the user's needs while remaining easy to understan and use.

The software system is designed of three modules the first is the registration where each student will register in the website his cources and personal information also the profesor will register and add his free time and there data will be saved in the database and login page will be for each of them to ensure their personality includes the password and username the secound module is the admin where he is allowed to generate the schedule or cofigure it manually he can also add professors,Courses, add rooms, add students and delete them all.

Other users can View schedule and interact with it simply.the admin also is the only one that can access the database the third module is the schedule auto/manual generation where the admin will generate the scheduale automatically according to the doctors free time , free rooms if conflicts happeaned he will configure it manually.

1.3 Glossary

These are definitions of infamous terms.

Term	Definition
SAG	Schedule Auto-Generator.
CMS	Content management system.
Database	Collection of all the information monitored by this system.
Genetic Algorithm	Check Appendix B.
Stakeholder	Any person with an interest in the project who is not a developer.
User	Student or professor.

1.4 References

- Teamleader, Joan, Paul Adams, Bobbie Baker, Charlie Charles. Web Publishing System, Software Requirements Specification. Jacksonville State University, 2004.
- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- Webopedia: Database. Retrieved 2 April, 2013, Available from the community site, <http://www.webopedia.com/TERM/D/database.html>
- Wikipedia: Entity relationship model, Retrieved 2 April, 2013, Available from the community site, http://en.wikipedia.org/wiki/Entity-relationship_model
- Wikipedia: Database, Retrieved 2 April, 2013, Available from the community site, <http://en.wikipedia.org/wiki/Database>

1.5 Overview of Document

- Chapter 2 is a Deployment Diagram that shows the physical nodes on which the system resides. This allows a clear explanation of where each design entity will reside. No design unit may straddle two nodes but must have components on each, which collaborate to accomplish the service.
- Chapter 3 is the Architectural Design. This is the heart of the document. It specifies the design entities that collaborate to perform the functionality of the system. Each of these entities has an Abstract Specification and an Interface that expresses the services that it provides to the rest of the system. In turn each design entity is expanded into a set of lower-level design units that collaborate to perform its services.
- Chapter 4 is the basic Data Structure Design, which for this project is a relational database. While it is separated out here for emphasis, it is really the lowest level of the Architectural Design.
- Chapter 5 is on User Interface Design and discusses the methodology chosen, why it was chosen and why it is expected to be effective.

- Chapter 6 describes the structure of the Help System.
- Chapter 7 exhibits the Use Case Realizations. The implementation of each use case identified in the SRS is shown using the services provided by the design objects.

Since there are no real time considerations, that chapter has been removed from this document.

2 Test scenarios and Test Cases

- **Test scenario :** Checking the functionality of (Login) button.
Test case :
 1. Click the button without entering user name and password.
 2. Click the button only entering User name.
 3. Click the button while entering wrong user name and wrong password.
- **Test scenario :** Checking the functionality of (edit profile) button.
Test case :
 1. Click the button edit and make change to any thing and save.
 2. Click the button edit and do not make any change to any thing and save.
- **Test scenario :** Checking the functionality of (configure of courses) button.
Test case :
 1. Click the button configure and do not choose any course.
 2. Click the button configure and do choose any course and click (add course).
- **Test scenario :** Checking the functionality of (request removal from the system) button.
Test case :
 1. Click the button remove when there is no thing.
 2. Click the button remove when there is any thing.
- **Test scenario :** Checking the functionality of (the schedule) button.
Test case :
 1. Press the button to go to the schedule when there is no schedule.
 2. Press the button to go to the schedule when there is more than schedule .
- **Test scenario :** Checking the functionality of (configure profile whether information or pictures) button.
Test case :
 1. Click the button edit profile pic.
 2. Click the button edit profile information.
- **Test scenario :** Checking the functionality of (configure courses) button for professor.
Test case :

1. Click the button change course.
 2. Click the button add the description of the syllabus.
 3. Click the button delete course when there are no courses.
 4. Click the button delete course when there are courses.
- **Test scenario :** Checking the functionality of (the schedule) for student button.
Test case :
 1. Press the button to go to the schedule when there is no schedule.
 2. Press the button to go to the schedule when there is schedule.
 - **Test scenario :** Checking the functionality of (what schedule level) for student button.
Test case :
 1. Press the button choose level and enter 0.
 2. Press the button choose level and enter 1.
 3. Press the button choose level and enter 3 when student at level 1.
 - **Test scenario :** Checking the functionality of (check information about university) for student button.
Test case :
 1. Click the button university information .
 - **Test scenario :** Checking the functionality of (Access and post in forums) for student button.
Test case :
 1. Press the add post and write no thing and press post .
 2. press the add post and write thing and press post.
 - **Test scenario :** Checking the functionality of (online voting) button.
Test case :
 1. Vote in more than choice.
 2. Vote in all choice.
 3. Vote in one choice.
 - **Test scenario :** Checking the functionality of (generate the schedule automatic) for student button.
Test case :
 1. Press generate more than once.
 - **Test scenario :** Checking the functionality of (manual generation) button.
Test case :

1. Press button manual generation and edit in the schedule.
- **Test scenario :** Checking the functionality of (send mails to students including schedule) button.
- Test case :**
1. Press send mails.
 2. press send mails and press include schedule.

3 Deployment Design

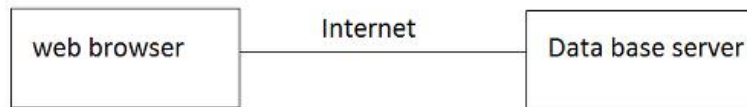


Figure 3.1: Deployment of System Environment.

A user accesses the Timetable website through the Internet using a Web browser (not part of this system although Web pages will run on it). The Timetable website resides on a dedicated Server with a permanent Web connection. The Admin manages all of the Timetable preparation work on his/her personal computer communicating with the existing Historical Society Database on the Database Server when needed, and uploading completed timetables to the timetable website Server when they are ready. The Admin Processor contains a local file system.

4 Architectural Design

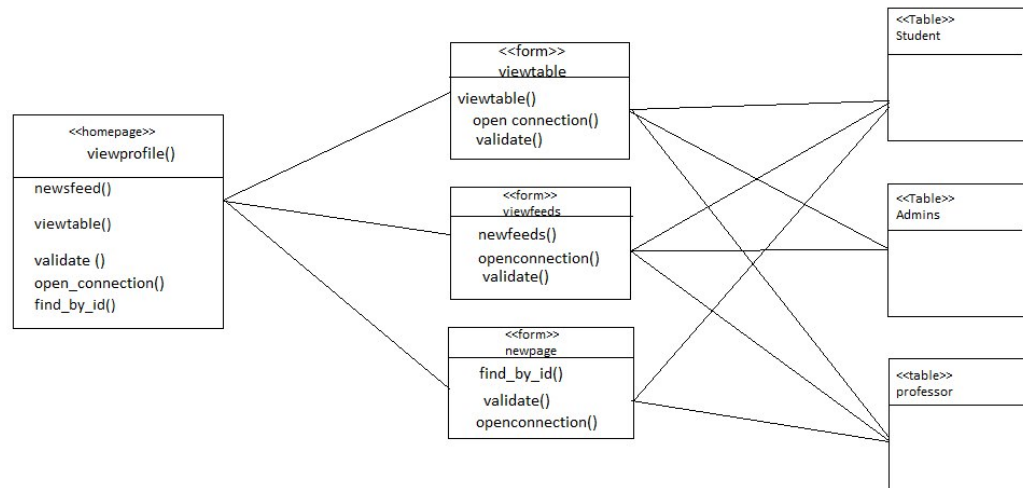


Figure 4.1: Architectural of System Environment.

The Schedule Auto-Generator system has four active actors and one cooperating system. The Admin, Professor, Student or Guest user access mini through the Internet. Admin, Professor and Student accesses the entire system directly after passing the login phase successfully.

4.1 find_by_id

Name: `find_by_id(id)`

Arguments: `id`

Returns: the object related to this `id`

Description: this function is used to find by `id` which searches the database to find this user and get him.

Pre-condition: The user is already exists in the database

Post-condition: the object is returned

Exceptions: the id doesnt exist in the database

Flow of Events:

1-The user login into the website

2-The database searches for this ID to validate his entrance

3-The system let the user to enter the website

4.2 open_connection

Name:open_connection()

Arguments: None

Returns: true or false

Description: This function checks if the connection to the database is working or nt

Pre-condition:The server is turned on

Post-condition: Returns true as the database is connected

Exceptions:the connection may be nt working

Flow of Events:

1-when the user enters the website we ensure that the website is connected to the database

2-if this function is true so the website will work

3-if this function returns false the website will die

4.3 validate

Name: validate ()

Arguments: None

Returns: true or false

Description: This function checks if the user is logged on or not in this session

Precondition: the user is signed in

Post-condition: the user will be able to see the page he is in it

Exceptions: the user hasnt logged in

Flow of Events:

1-the user enters in the site

2-when he moved in next session the website will ensure that he is logged in

3-he will be able to stay in the website

4.4 viewtable

Name: viewtable()

Arguments: table

Returns: the view of the table

Description: this function view the table for the users

Precondition:the table is already exists

Post-condition:it view the table

Exceptions: the table is not exist

Flow of Events:

- 1-when the user press button view table he will see the table
- 2-the system will know the type of the user
- 3-if the user is student the system will show him his schedule

4.5 newsfeed

Name:newsfeed()

Arguments: none

Returns: all the news

Description: this function view all the news posted in the website

Precondition: new exists

Post condition: it view all the news

Exceptions: None

Flow of Events:

- 1-the user press the button new feeds
- 2-the system will show him all the news that is posted

5 Data Structure Design

There is one database embedded in this product. It's TimeTable and resides on the College Server.

5.1 TimeTable Database

This database contains thirteen tables. The Query feature of the database management system is utilized extensively by the program.

Users Entity

Data Item	Type	Description	Comment
ID	Int	Primary key	
User_name	Text	Name of user	
Hashed_password	Int	Password of the user	

Students Entity

Data Item	Type	Description	Comment
ID	Int	ID given by the college(Primary key)	
First_name	Text	First name of the student	
Secound_name	Text	Second name of the student	
Username	Text	The username of the student	
Hashed_password	Int	The password of the student	
Age	Int	Age of the student	
Grades	Int	The grade of the student	
Desc	Text	Description of student	
Student_Group_ID	Int	Group number	

Student Group Entity

Data Item	Type	Description	Comment
ID	Int	Primary Key	
Name	Text	Name of student group	
Size	Int	Number of students	
Level	Int	The level of the students	

Room Entity

Data Item	Type	Description	Comment
ID	Int	The ID of the room	
Name	Text	The name of the room	
Is_lab	Int	It checks if the lab is lab or room	
No_seats	Int	The number of seats In the room	

Student Groups Entity

Data Item	Type	Description	Comment
ID	Int	The Student Groups ID	
Class_ID	Int	The class ID	
Student_Group_ID	Int	Student Group ID	

Courses Entity

Data Item	Type	Description	Comment
ID	Int	Primary Key	
Name	Text	Course names	
Description	Text	Details about courses	

Professor Entity

Data Item	Type	Description	Comment
ID	Int	Primary Key	
First_name	Text	The name of professor	
Second_name	Text	The second name of professor	
Username	Text	The name will use to log in	
Hashed_password	Text	Encrypted password	
Description	Text	The details	
Age	Text	The age of professor	
Phone number	Text	The number of professor	
Address	Text	The address of the professor	

Events Entity

Data Item	Type	Description	Comment
ID	Int	Primary Key	
Name	Text	Name of event	
From	Int	The start	
Duration	Int	Long of the Event	
Day	Text	The day of the event	
Content	Text	The details of the event	
desc	Text	The details of description	

Admins Entity

Data Item	Type	Description	Comment
ID	Int	Primary Key	
First_name	Text	The name of admin	
Second_name	Text	The second name of admin	
Username	Text	The name admin will use to log in	
Hased_password	Int	Pass used to log in	
Description	Text	Details about himself	
Age	Int	Age of the admin	

News Entity

Data Item	Type	Description	Comment
ID	Int	Primary Key	
Author	Text	The person who publish the new	
Title	Text	The title of the new post	
Content	Text	The details of news	

Courses Entity

Data Item	Type	Description	Comment
ID	Primary key	Primary key	
Name	Text	Name of course	
Description	Text	Details about courses	

Course Class Entity

Data Item	Type	Description	Comment
ID	Primary key	Primary key	
Professor_ID	Pointer	Pointer to professor entities	
Course_ID	Pointer	Pointer to course entities	
Room_ID	Pointer	Pointer to room entities	
Student_group_ID	Pointer	Pointer to student group entities	
Duration	Int	Duration of a course	
Lab	Int	The course needs lab or not	

Equipments Entity

Data Item	Type	Description	Comment
ID	Primary key	Primary key	
Type	Text	The type of the equipment	
Student_ID	Int	The Id of the student who reserved the equipment	
Reserved_from	Text	The starting date of the reservation	
Duration	Int	The duration of the equipment with the student	
Name	Text	The name of the equipment	

6 Real-Time Design

Real-time considerations are minimal in this project.

The On-Line Timetable website is designed for multiple users but the concurrent usage is handled client-side. That is, the Home Page will execute on the client (users) computer and will make requests of the timetable server. These requests are handled sequentially by the server with no transient data storage.

7 User Design

7.1 Student User Interface

The On-Line User interface will feature the logo of the website.

He has many options; first he can edit his profile so he can change his name or any information about himself, second he can click the button of view schedule so he will be able to see his schedule and the time slots, third he can see all the newsfeeds that are posted on the forum .Finally he can vote on the current schedule.

All the data of the student is saved in the database, so his new changes will be updated on the database, the vote of the student will be sent to the admin to check the validation of the table and change in it if it need.

7.2 Student User Interface

This will be a standard Access form. It will contain a list of all services grouped by type.

The professor has some options; first he can edit his profile, he also can edit the description of his courses, second he can view his schedule by clicking the button view table, third he can post in the forum and see the old posts, fourth he can update his free time as the timetable generation depends on this time to generate the table.

All the professor data is saved on the database so his new changes will be updated on the database.

8 Help Design

There is only a minimal Help System for this project.

The admin will have the feedback of the users on the schedule through the voting system.

The website will have a button Contact us for a user to report problems or send suggestions for the admin.

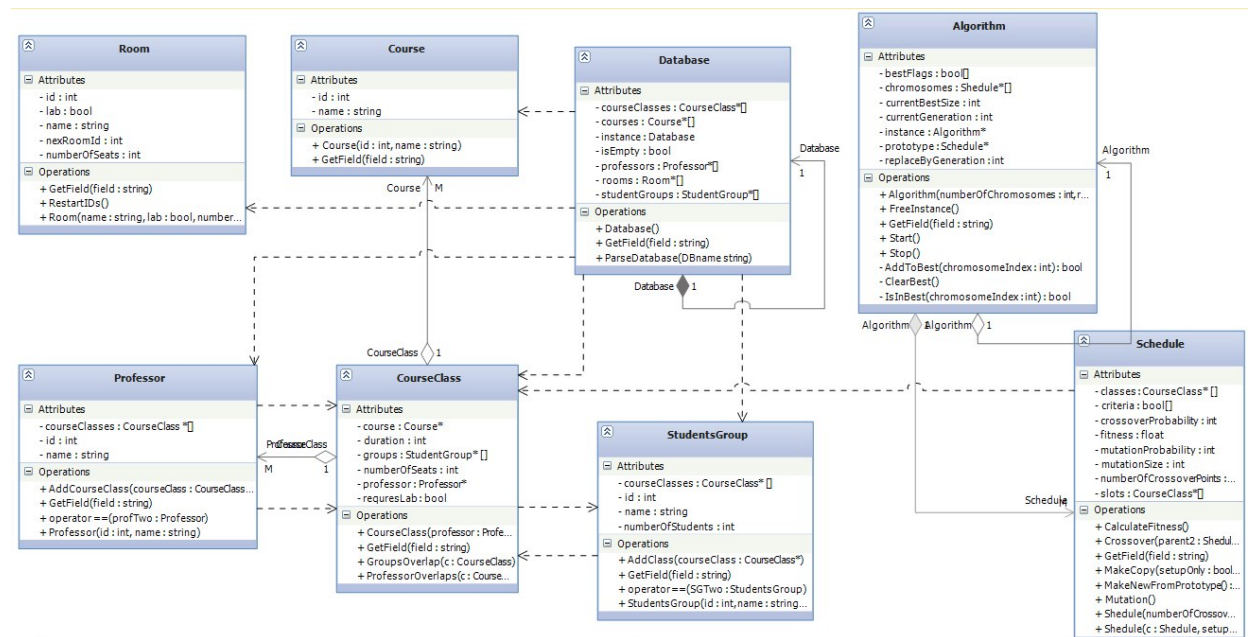


Figure 8.1: Class Diagram of System Environment.

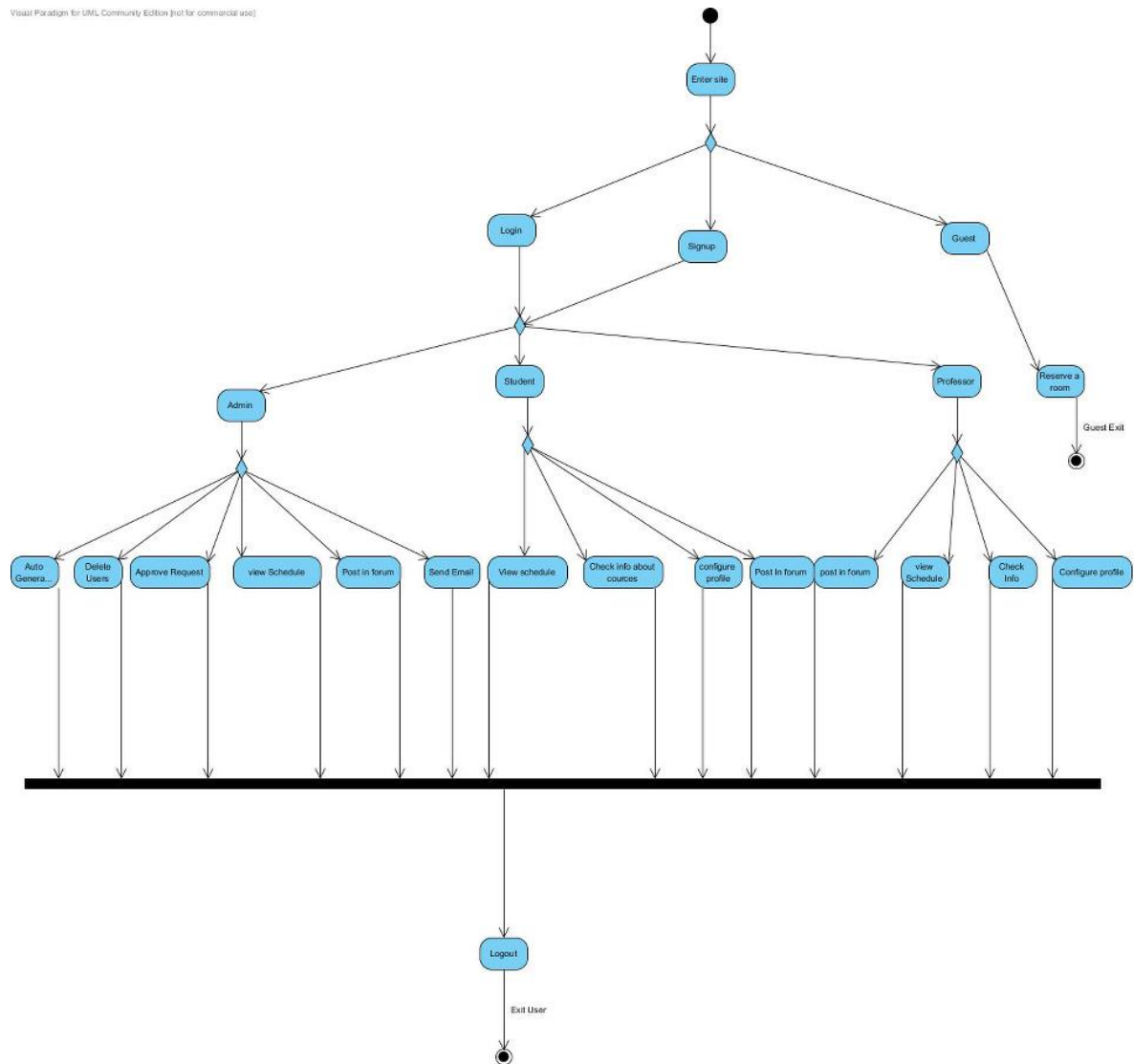


Figure 8.3: Activity Diagram of System Environment.

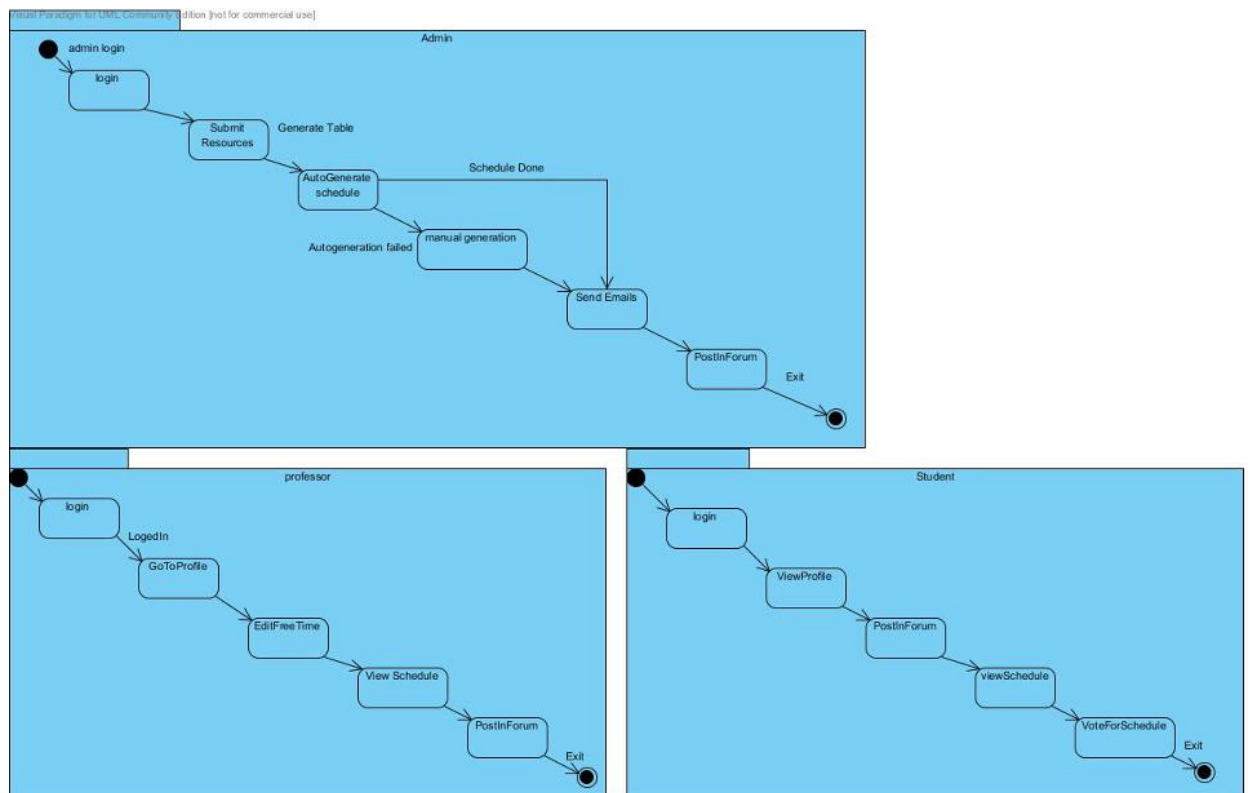


Figure 8.4: State Diagram of System Environment.

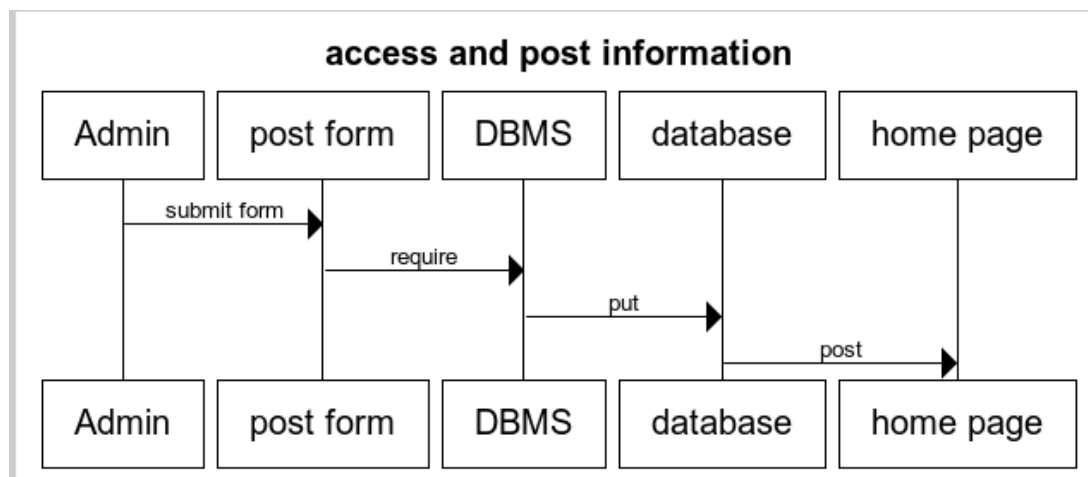


Figure 8.5: Access and post information Diagram.

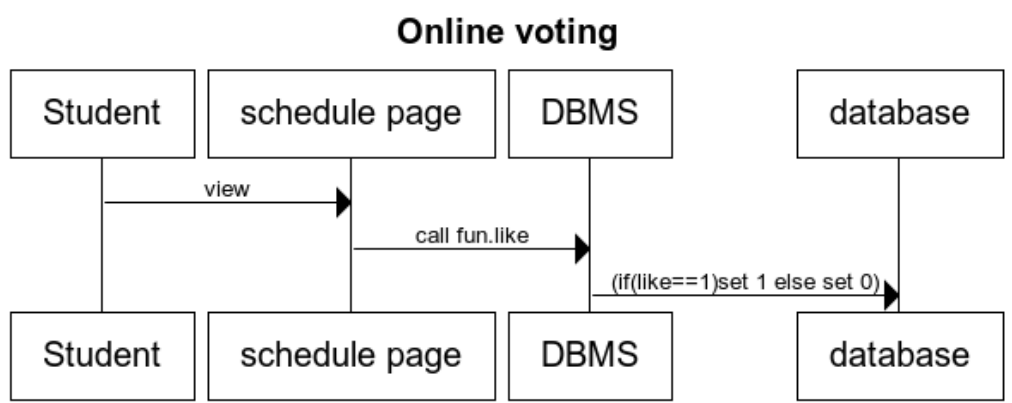


Figure 8.6: Online voting Diagram.

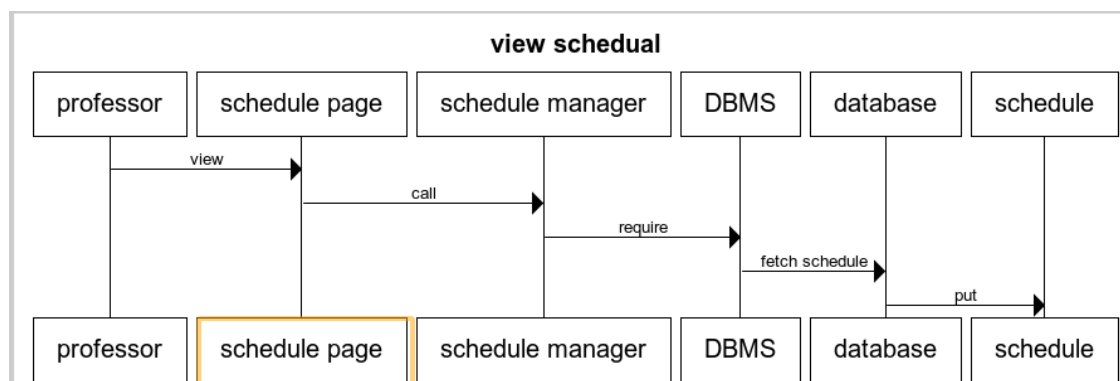


Figure 8.7: View scheduale Diagram.

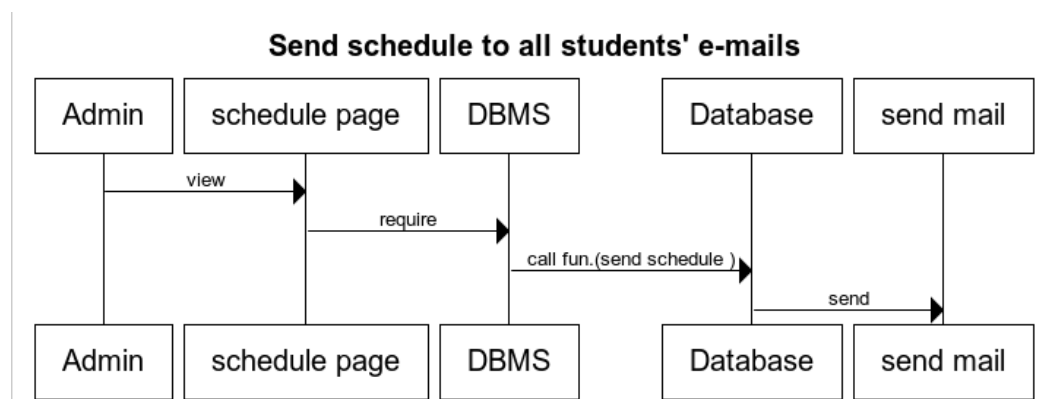


Figure 8.8: State Diagram of System Environment.

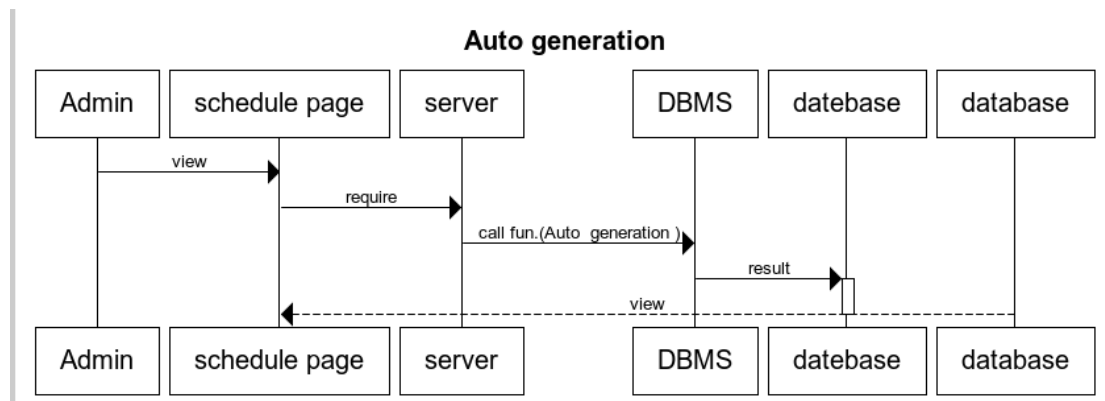


Figure 8.9: State Diagram of System Environment.