VITAL EDUCATION SYSTEM (VITAL-EDU01)

DEVELOPED BY - DIGITIZER

TECHNICAL DOCUMENT

VOLUME 1.0

Contents

[PREFACE 3](#_Toc432022904)

[USE OF THIS DOCUMENT 4](#_Toc432022905)

[INTRODUCTION 5](#_Toc432022906)

[SYSTEM OVERVIEW 6](#_Toc432022907)

[SYSTEM CHARACTERISTICS 7](#_Toc432022908)

[USER CHARACTERISTICS 8](#_Toc432022909)

[SYSTEM CONTEXT 10](#_Toc432022910)

[SYSTEM DESIGN DIAGRAMS 13](#_Toc432022911)

[DATA FLOW DIAGRAM LEVEL 0 13](#_Toc432022912)

[SOFTWARE DEVELOPMENT TOOLS 14](#_Toc432022913)

[INTERFACES 15](#_Toc432022914)

[SOFTWARE REQUIREMENT TRACEABILITY MATRIX 16](#_Toc432022915)

[FORWARD AND BACKWARD TRACTABILITY MATRIX 17](#_Toc432022916)

[USE CASE SCENARIOS 18](#_Toc432022917)

[REGISTER TO THE SYSTEM 18](#_Toc432022918)

[CODE SPECIFIC ADMIN PANEL 19](#_Toc432022919)

[DOCUMENT CONTROL 21](#_Toc432022920)

[DOCUMENT SIGNOFF 21](#_Toc432022921)

[DOCUMENT CHANGE RECORD 21](#_Toc432022922)

# PREFACE

This document is a generic Technical Document for use by Vital Education Pyt Ltd to extend their system in future. It provides technical guidance and technical overview of the system. This document serves as a visual map for technical personals to understand the system and enable them to edit the system for further improvement of the system.

# USE OF THIS DOCUMENT

This document can be used for following purposes:

#1 To understand the full functionality of Vital Education Online System and grasp technical overview as a whole.

#2 To study detail structure of Vital Education Online System providing readers vast understanding of the system architecture for further enhancement of the system.

#3 To understand programmatic techniques used to develop this system, to study the technical connectivity with backend and frontend.

#4 To understand the middle tier of the system which is the functional unit of overall system.

#5 Making concise decision to elaborate the system is a crucial point for developer, this document can be of great help for this purpose.

#6 Admin and maintenance staff of vital education can also use this document for simple exploration of the system and get overall understanding.

#7 To get idea about software requirement and hardware requirement to operate the online system.

# INTRODUCTION

This document is a technical manuscript which provides technical, structural and architectural understanding of the system. It is believed to be a technical guide for further extension of the system. It covers all the cornerstones of the system with great details. It also serves as user manual in some cases like understanding the interfaces of the system to the outer world.

This document is divided into three sections, one serve as basic overview of the system, other present detail functions implementation and the last part provides extensive structure and further enhancement parts of the system. Each section contains technical jargons which are further elaborated in simple language for clear understanding of the system. Some parts contain technical diagrams like context diagram which is the heart of the system. It is further elaborated in small parts with clarified diagrams. In the middle part, it contains system requirements for operating this system. This lists all the requirements to make this system work in World Wide Web (WWW).

This document will clearly describe all the details from database to front end with step by step manner. Technical personal will surely extract in-depth knowledge about the system going through this document. It will be a pocket guide for future expansion of the system. The diagrams mentioned in the document are all accurate and easy to understand to a technical personal. It is mandatory to check through this document once to understand the system fully before going through development phase. This document will also give a quick view of the overall system in the middle part of the section. It will be a quick guidance for short term maintenance.

It is all about further development of the system. Someone who is willing to extend the system in future will be having different option to follow. This document will serve as a first sight document to those who want to make the changes. It also focuses in the weak and strength of the system which will give hints to the developers for what part might need further development in future purposes. It will give some basic building concepts to the developers without further delay in confusions.

Digitizer team fully believes that this document will be of great help for developers who want to extend this system in future endeavor. Thank you.

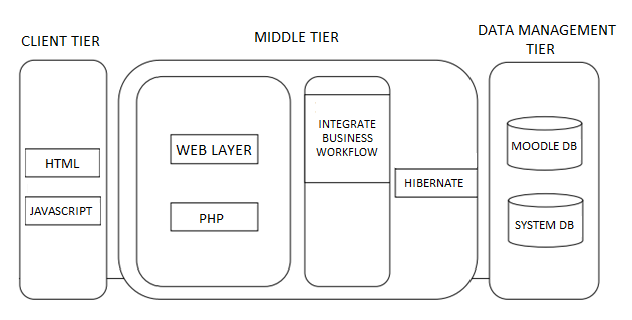
# SYSTEM OVERVIEW

Vital education online system has two gateways; one for admin and another for normal users like students. The system gives highest privilege to admin for all kind of operations. He/she has full control over the system. For storing student details and course details, there is database system which will handle this problem. MySQL query used to communicate with the database. The system is developed in such a way that admin can make changes in database simply via admin panel by clicking and typing. This will make the system easy to operate for non-technical people.

Every members are required to log in to the system by passing user Id and password. It needs to be valid or else user cannot log in to the system; for validating user Id and password, first system will checks the data fields are filled or not. If it is filled, the system takes the entered data to database comparator which will compare the given data with stored user Id and password. It will check every details, like case sensitivity. And once the comparison is passed, the system opens door for that particular user. All the user should provide their passwords to the system before entering. To avoid unauthorized access, there is used prepared statements which will avoid injection commonly used by hackers to hack the system.

## SYSTEM CHARACTERISTICS

The system has different components. They interact each other to do specific task as user want. It is shown in the following diagram the top view of the system. The following diagram will only show main parts of the system and their connectivity.

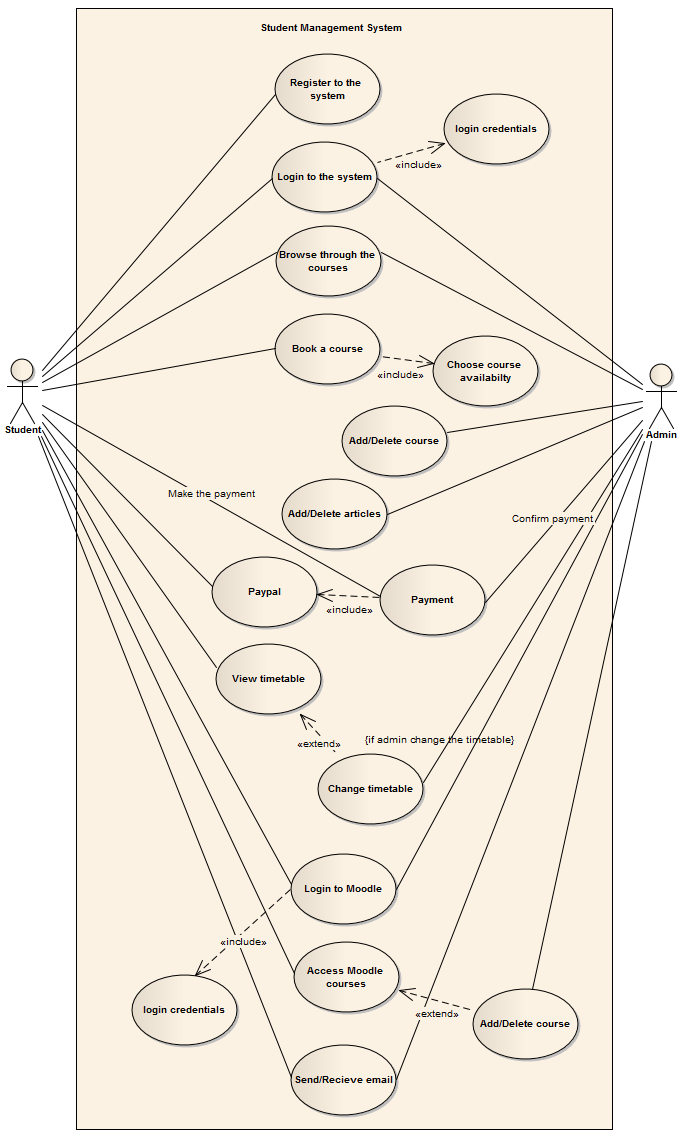


## USER CHARACTERISTICS

Mainly there are three type of users:

* Admin
* Registered users
* Guest users.

There is different user interaction for each user. Every individual has specific functionality. For example admin might have different functionality than students. Like admin can assign tutor for students but students can’t do it. So to understand all these functionalities, following use case diagram will help to make mind map to the system and its user’s interactivities.



# 

Admin

System database

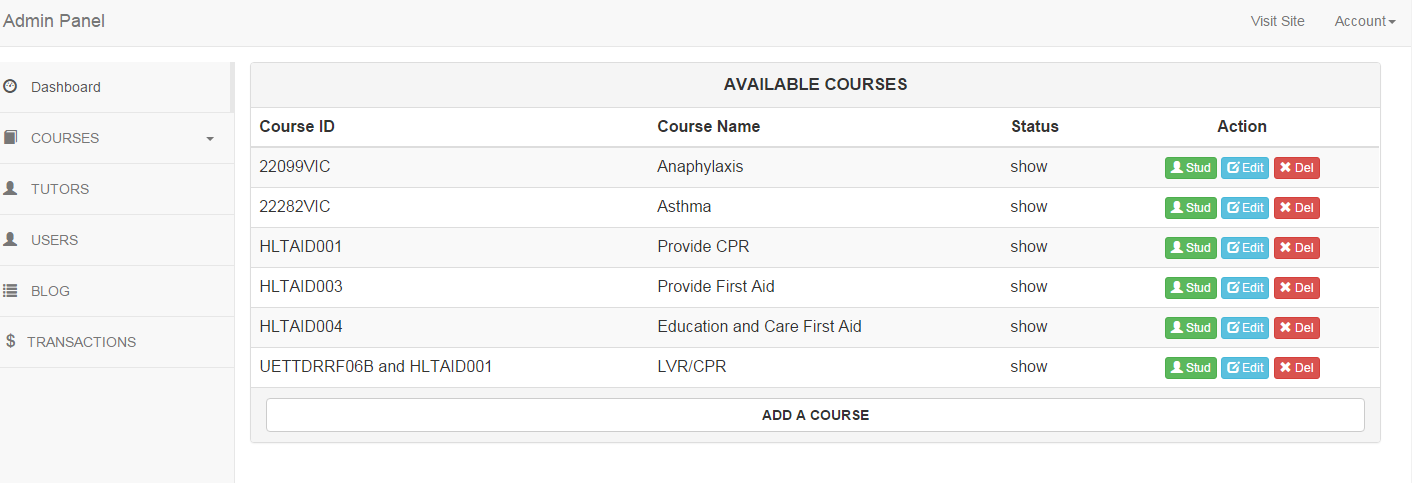
Add/delete courses. How?

Vital Edu Online

System

Where this changes resides then?

What happens to the updates?



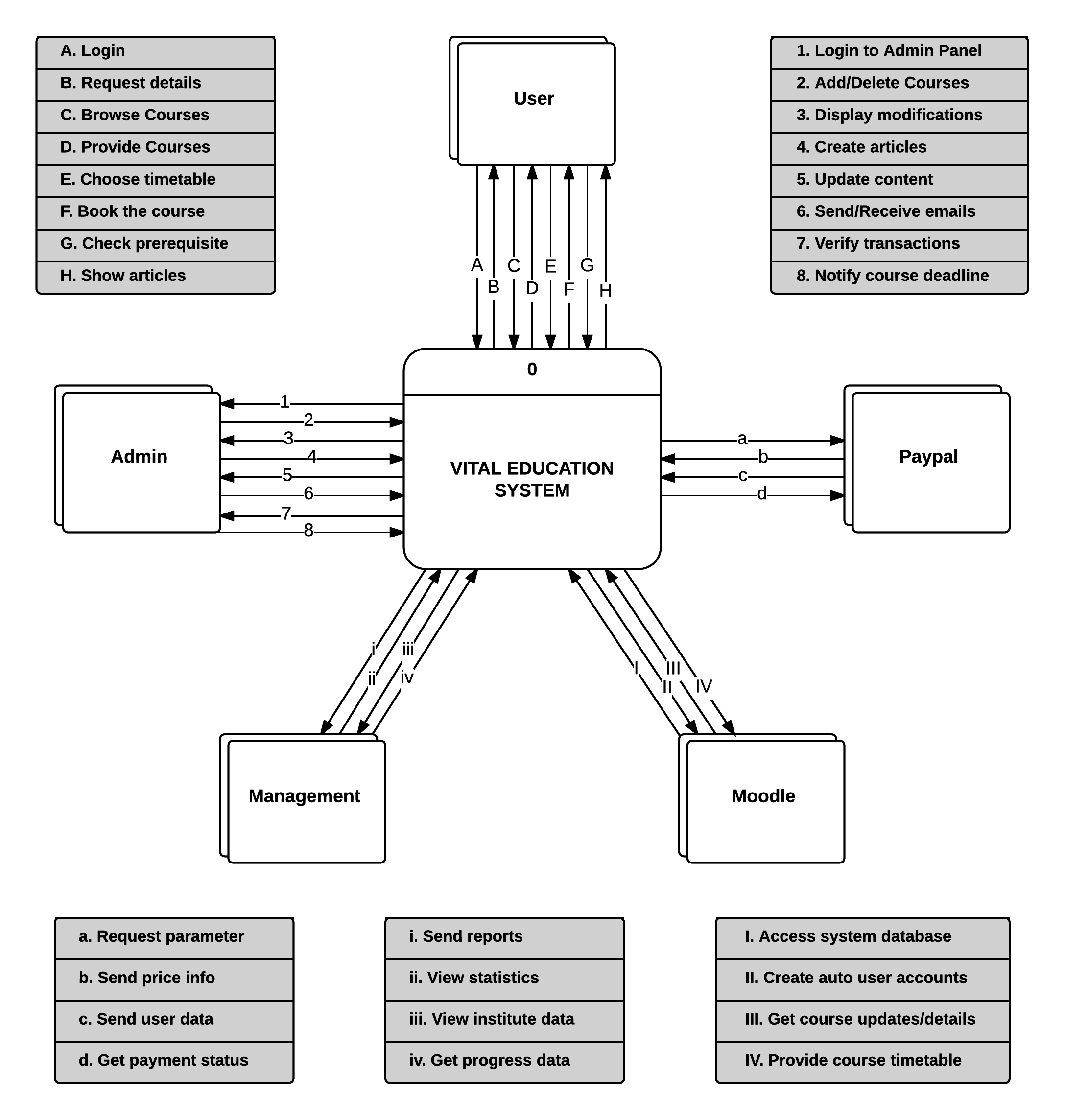
# SYSTEM CONTEXT

|  |  |
| --- | --- |
| *Class Diagram* | *Describes the structure of a system* |
| *Object Diagram* | *Expresses possible object combinations of a specific Class Diagram* |
| *Statechart Diagram* | *Expresses possible states of a class (or a system)* |
| *Activity Diagram* | *Describes activities and actions taking place in a system* |
| *Sequence Diagram* | *Shows one or several sequences of messages sent among a set of objects* |
| *Collaboration Diagram* | *Describes a complete collaboration among a set of objects* |
| *Use-case Diagrams* | *Illustrates the relationships between use cases* |
| *Component Diagram* | *A special case of a Class Diagram used to describe components within a software system* |
| *Deployment Diagram* | *A special case of a Class Diagram used to describe hardware within the overall system architecture* |
| *System Block diagram* | *A diagram showing the major components of the system with its interconnections and external interfaces* |

We have five external agents where the main system interacts. The five agents it Interacts are User, Admin, Moodle,Paypal,Management.

**Admin** has the authority of controlling whole system such as adding, deleting or updating the courses, posting articles in the blog, managing the UI, changing the webpage content etc. Admin can also provide the deadlines for the course selections. Admin in the article section can add or delete or even modifying the blog at any instant of time. Admin can send the e-mails to the users regarding any information and receive the mails from their users regarding any queries.

**Management** recieves the updates weekly on the whole system and goes through the transactions between the user and the admin on payment,etc.Management manages the reports of the whole system and also have an overview of the statistics day by day. Management views the institute data of the number of students that are enrolled, payment data. It gets the progress data of the institution.

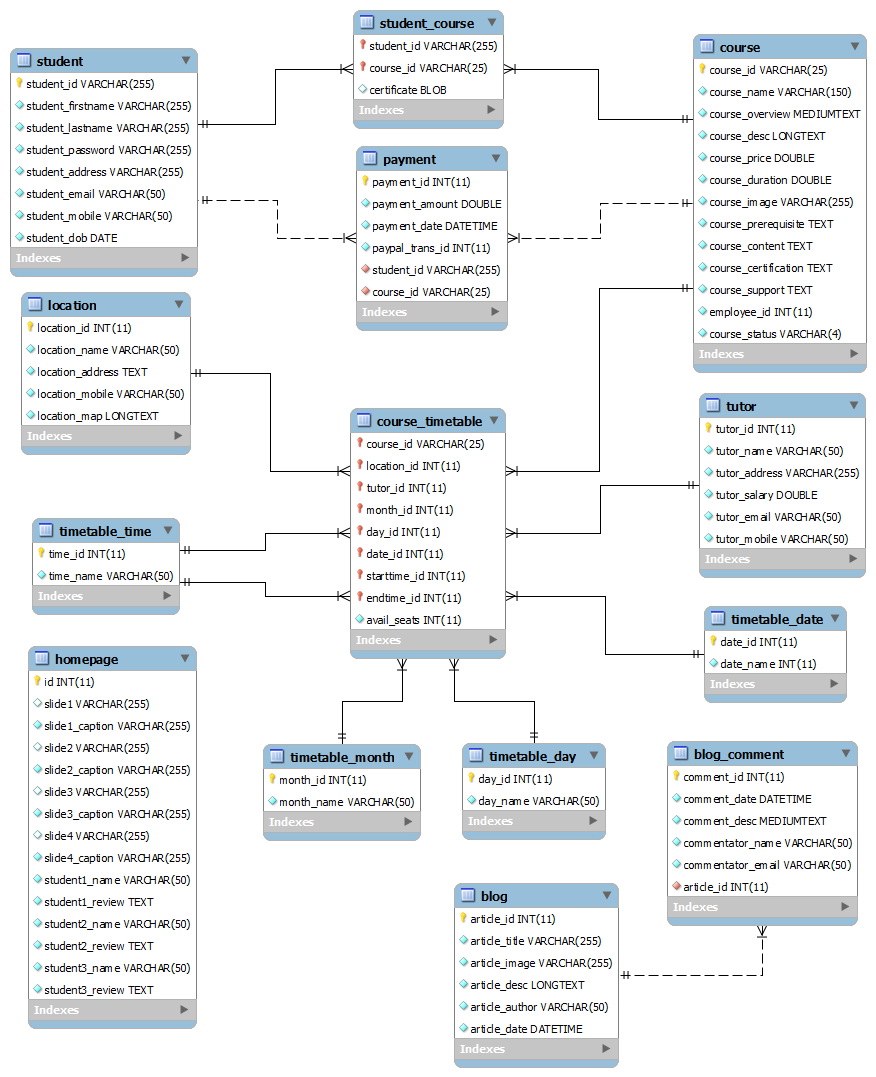


**Users** play the most important role as the external agents of the system. They have access of booking the courses, accessing the moodle, choosing the timetable, navigate through the webpages, etc. Users first needs to get registered for the site and get and id with password and using that can access the login page. Unless he gets registered he cannot buy a course. Selecting the course after logged in leads to the paypal page. Users also can view the articles in the site and also can add the comments to the articles. Users in the case of booking the courses they even can check the pre-requisit required for the course.

**Paypal** helps the payment process, once the course gets selected and is to be booked it leads to the paypal page. Next to the selection of the course the user will be directed to the paypal page to continue with the transaction providing his card details or login with the paypal account, in this stage they also can create a new paypal account. On the successful payment, user gets the confirmation mail and gets a moodle account created for the new user. Through this moodle account user can access the institution moodle.

**Moodle** system databse is there connected to the vital education database through which the medium users will access the courses that are present in the Moodle. Users in the moodle can there view the courses which they have been enrolled and the content of the course. In this moodle they also can view the data or any material related to that course. In this moodle admin can make the changes with the courses such activation or deactivation, adding or deleting the courses. Admin can the schedules for the courses in this moodle.

# SYSTEM DESIGN DIAGRAMS



## DATA FLOW DIAGRAM LEVEL 0

# SOFTWARE DEVELOPMENT TOOLS

Those software tools and technology we used for doing this assignment are as follows.

1. The list include:

an application development tool ;

server side language: PHP;

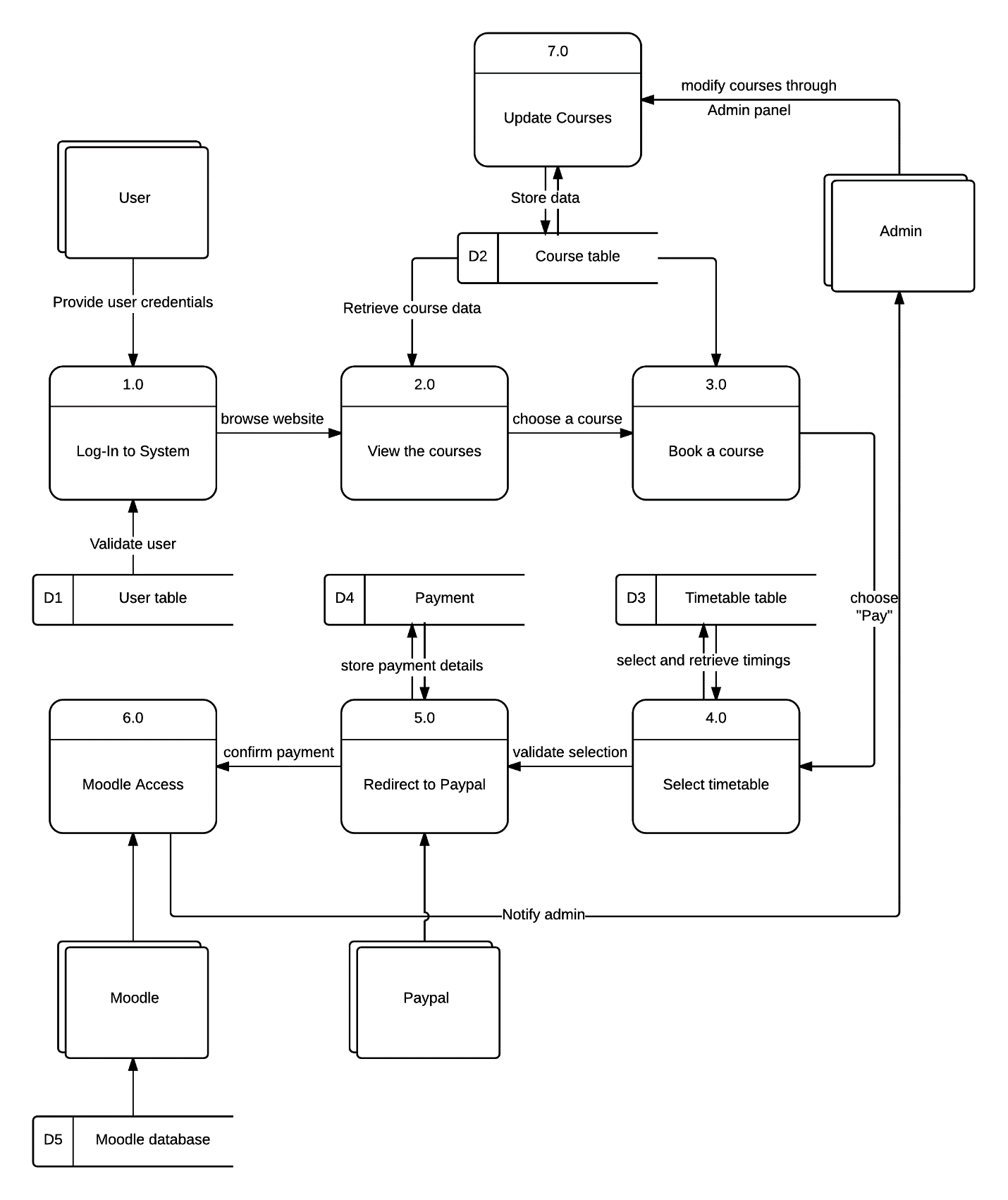
HTML ;

A word processor for documentation: Microsoft word.

a tool for drawing diagrams: Enterprise architect

Automated testing tools: nUnit testing.

INTERFACES



# SOFTWARE REQUIREMENT TRACEABILITY MATRIX

As per the client requirements, an admin panel is to be designed through which the admin can add, remove, change or update almost the whole GUI of the system as well as the system database which will reflect the main system with which the users interact.

**Login to Admin Panel:** Admin can login to the admin panel using the admin credentials. As per the client specifications, only one admin account will be created and anyone in the management panel will use the same credentials to login to the system.

**Add/Delete courses:** Admin can add and delete the courses into the database through admin panel. Adding a course involves providing all the details of the course in the given fields.

**Update courses:** Admin can modify the available courses according to the management requirements. A frequent updating of time table is required since the availability of the trainer changes.

**Activate/Deactivate courses:** Courses can be either activated or deactivated by changing the status of the course which will reflect the changes in the live system. A deactivated course will not be visible in the live system.

**Reset Password:** If a user cannot reset the password for any reason, the admin should be able to access the user table and reset the password for the user.

**GUI modification:** Admin will be provided with a GUI that is similar to the live system where he/she can change the web elements like course description, pictures etc.

**Delete/Block users:** Admin should be able to block or delete a user from the database.

**Send and receive emails:** Admin can send and receive emails through the admin panel.

**Assign tutor:** Assigning tutors to the course can be done in the course updating page where the admin can get the list of tutors from which he/she can choose one.

**Change user access permission:** User access in the Moodle depends upon the level of permissions they have. Admin can change the access level of the users in the Moodle admin panel.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Project Name | | Vital Education | | | | |
| Project Manager | | Raja Sekhar Masina | | | | |
| Project Description | | Developing a Web system for Vital Education Pty Ltd for conducting short term courses which involves an admin panel and Moodle. | | | | |
| ID | **Action** | **Functional requirements** | **Status** | **Test case #** | **Priority** | **Comments** |
| ADMINISTRATOR | | | | | | |
| FWD01 | Login to Admin Panel | RQ01 | Completed | TCF01 | High | Implemented |
| FWD02 | GUI modification | RQ02 | Partially Implemented | TCF02 | Med | Implemented |
| FWD03 | Assign tutor | RQ03 | Incomplete | TCF03 | High | Implemented |
| BCK01 | Activate/Deactivate courses | RQ04 | Completed | TCB01 | Med | Implemented |
| BCK02 | Send and receive emails | RQ05 | Incomplete | TCB02 | Med | Implemented |
| BCK03 | Delete/Block users | RQ06 | Completed | TCB03 | Med | Implemented |
| BCK04 | Update courses | RQ07 | Completed | TCB04 | High | Implemented |
| BCK05 | Add/Delete courses | RQ08 | Completed | TCB05 | High | Implemented |
| BCK06 | Reset Password | RQ09 | Incomplete | TCB06 | Med | Implemented |
| BCK07 | Change user access permission | RQ10 | Incomplete | TCB07 | Med | Implemented |
| ENROLLED USER | | | | | | |
| FWD04 | Login to the web system | RQ11 | Completed | TCF04 | High | Implemented |
| FWD05 | Upload assignments | RQ12 | Incomplete | TCF05 | Med | Implemented |
| FWD06 | Browse the webpage | RQ13 | Completed | TCF06 | High | Implemented |
| FWD07 | Book a course | RQ14 | Partially Implemented | TCF07 | High | Implemented |
| FWD08 | Choose Availability | RQ15 | Partially Implemented | TCF08 | High | Implemented |
| FWD09 | Make PayPal Transaction | RQ16 | Incomplete | TCF09 | High | Implemented |
| FWD10 | Access Moodle courses | RQ17 | Completed | TCF10 | Med | Implemented |
| BCK08 | Auto generate Moodle account | RQ18 | Incomplete | TCB08 | Med | Implemented |
| BCK09 | Send account verification mail | RQ19 | Incomplete | TCB09 | Med | Implemented |
| GUEST USER | | | | | | |
| FWD11 | Register to the web system | RQ20 | Completed | TCF11 | High | Implemented |
| FWD12 | Access Blog articles | RQ21 | Completed | TCF12 | Med | Implemented |
| FWD13 | Post Comment in Blog article | RQ22 | Completed | TCF13 | Med | Implemented |
| FWD14 | Make an enquiry | RQ23 | Completed | TCF14 | Med | Implemented |
| FWD15 | Get location map | RQ24 | Completed | TCF15 | Med | Implemented |

### FORWARD AND BACKWARD TRACTABILITY MATRIX

## USE CASE SCENARIOS

### REGISTER TO THE SYSTEM

The below Use Case Scenario describes about how a new user registers into the Vital Education Web System.

|  |  |
| --- | --- |
| Use case name | Register to the system |
| Primary actor | User |
| Supporting actor(s) | Vital Education System |
| Summary | Any guest user can register to the system by selecting the registration option available in the webpage |
| Pre-Conditions | Navigate to the registration page  Provide necessary details |
| Flow of events | Select the register now button  Fill all the necessary fields  Confirm password  Answer the human identification question  Hit the register button  Open email and confirm the account by clicking the link in the received mail |
| Exceptions | User email id already given  Username already taken  Server issues |
| Post-Conditions | Create a new user in the database  Send an automated email |

# CODE SPECIFIC ADMIN PANEL

The following pseudo code explains how the system is being implemented and all the major SQL queries constructed to retrieve or store the data in the database.

To start with the database implementation, we created a course\_timetable table which was connected to most of the tables to get the timetable. This table includes course\_id, tutor\_id, location\_id, date, start\_time, end\_time, and available\_seats. To retrieve the data from the database, the query that was constructed is

SELECT course\_name, location\_name,tutor\_name,month\_name,day\_name,date\_name,st.time\_name AS 'starttime',et.time\_name AS 'endtime',avail\_seats FROM course\_timetable

JOIN course ON course.course\_id = course\_timetable.course\_id AND course.course\_id = '$id'

JOIN location ON location.location\_id = course\_timetable.location\_id

JOIN tutor ON tutor.tutor\_id = course\_timetable.tutor\_id

JOIN timetable\_month ON timetable\_month.month\_id = course\_timetable.month\_id AND timetable\_month.month\_id >= MONTH(NOW())

JOIN timetable\_day ON timetable\_day.day\_id = course\_timetable.day\_id

JOIN timetable\_date ON timetable\_date.date\_id = course\_timetable.date\_id

JOIN timetable\_time as st ON st.time\_id = course\_timetable.starttime\_id

JOIN timetable\_time as et ON et.time\_id = course\_timetable.endtime\_id

ORDER BY month\_name DESC, date\_name ASC";

The server-side language used in the project is Php. To access any information from the database, Php statements had been implemented inside the html pages where ever necessary. To login into the system, the Php code implemented is

<?php

require ('dbconn.php');

if(!isset($\_POST['submit'])) {

// Some HTML code to display the login page

} else {

require ('dbconn.php');

$userid = mysqli\_real\_escape\_string($conn, $\_POST['userid']);

$password = mysqli\_real\_escape\_string($conn, md5($\_POST['password']));

$query = "SELECT student\_id,student\_firstname,student\_password FROM student WHERE student\_id='$userid' AND student\_password='$password'";

$run\_query = mysqli\_query($conn, $query);

$row = mysqli\_num\_rows($run\_query);

if($row == 1){

$\_SESSION['userSession']=$userid;

echo "<script>alert('Log In successful')

window.location.href='index.php'

</script>";

} else {

echo "<script>alert('Invalid username or password. Try again!')

window.location.href='login.php'

</script>";

}

}

?>

To keep the whole website more consistent, we used bootstrap framework which comes with some basic predefined styling. The design style used in the website is material design where the web components come up as cards and with some border and shadows. The implemented pseudo code is

<div class = “panel panel-default”>

<div class = “panel-heading”>

//Some code that goes in heading

</div>

<div class = “panel-body”>

//Some code that goes to body

</div>

<div class = “panel-footer”>

//Some code that goes to footer

</div>

</div>

# DOCUMENT CONTROL

|  |  |
| --- | --- |
| **Title:** | Technical Document |
| **Issue:** | Issue 1 |
| **Date:** | 6 October 2015 |
| **Author:** | Digitizer team |
| **Distribution:** | XP Consultant – Ravish Goyal |
| **Reference:** | IDA-MS-TD |
| **Filename:** | IDA-MS-TD-i1 |
| **Control:** | Reissue as complete document only |

# DOCUMENT SIGNOFF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nature of Signoff** | **Group** | **Signature** | **Date** | **Role** |
| DEVELOPER | Digitizer team | *Raja Sekhar* |  | Project Member |
| REVIEWERS | XP CONSULTANTS | *Ashok Kumar* |  | Consultant |

# DOCUMENT CHANGE RECORD

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Change Details** |
| 02 January 2015 | Issue 1 Draft 2 | Digitizer | First complete draft |
| 08 April 2015 | Issue 1 Draft 3 | Digitizer | Review and update |
| 04 October 2015 | Issue 1 Draft 4 | Digitizer | Updated |
| 08 October 2015 | Issue 1 | *Ashok Kumar Najani* | Apply review comment and issue |