

Online College Administration System

Vinayaka V.M.¹, Chidambara M.L.²

¹RJS Polytechnic , Koramangala, Bengaluru

²Institute of Management Studies, Davanagere

vmvinayaka@gmail.com

Abstract—This system provides facility to store and access the student details such as personal details, academic details and fee details for the college administration and also provides read access to the students as well. In this system the administrator will have to enter the details of the students in to the database. He can generate reports and he can view the data category wise also It is very difficult for the college staff to maintain the students records manually and all the details are not available at once. Also, the manual work is tedious process. Hence, we got up with an idea to computerized this system. The result is this work. We prepared a system, called Online College Administration. It allows students to find their academic, personal and fee details at their fingertips.

Keywords—Personal details, Academic details, Fee details

I. INTRODUCTION

1.1 Preamble

The college administration is made online for ease of managing things that are required by the college staff. This system provides facility to store and access the student details such as personal details, academic details and fee details for the college administration and also provides read access to the students as well. In this system the administrator will have to enter the details of the students in to the database. He can generate reports and he can view the data category wise also. The students can access the data of individual students provided they know their Student ID.

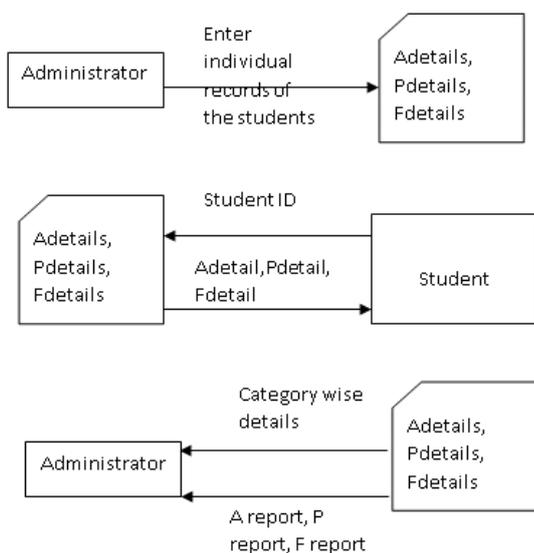


Figure 1. Online College Administration system

Objectives:

*To build Student database consisting of Personal details, Academic details, and Fee details of the students.

*To display the records of the students whose Student ID is given to the program.

*To build category wise listing of the student details.

*To generate reports of Personal, Academic and Fee details of the students.

1.2 Motivation

It is very difficult for the college staff to maintain the students records manually and all the details are not available at once. Also, the manual work is tedious process. Hence, we got up with an idea to computerized this system. The result is this work. This project removes the burden for the college staff from maintaining books and books of student data and searching manually the student details.

1.3 Taxonomy

SQL: SQL is a query language for databases, able to retrieve and manipulate data stored in MYSQL database. It is the key technology at the back end which is used worldwide.

JSP/HTML: JSP stands for java server page and html means hypertext mark-up language. These are the front end technologies used in this project for GUI design.

JAVA: JAVA is the high level machine independent language which is used to design the controller component which is the core of this project.

1.4 Existing System:

Manual record maintenance with record books. The design and implementation of a comprehensive Online College Management System and user interface is to replace the current paper records. College Staff are able to directly access all aspects of a student's academic progress through a secure, online interface embedded in the college's website.^[1]

1.5 Proposed System

Online web based technology used to computerize the administration of the college work. Every detail of the student is available online in the college LAN. The administrator of the system is able to generate reports and see category wise student details. Automatic generation of report on one click. The resources like the human effort, paper, printer ink and cost for manual changing of the notices in the notice board will be reduced using our proposed system.^[2]

II. SCOPE OF THE PROJECT

2.1. Advantages

*Simple User Interface.

*On implementing this project the organization will get error free data to analyse.

*Maintenance is much easier and accurate than the existing manual system.

*Cost effective.

*Security features are somewhat higher than that of manual approach.

2.2. Limitations

As a computer based system it is easier to fetch data from the database but hackers may steal the data from the computers.

2.3. Applications

This project is intended to make the administration of schools and colleges as paper free offices.

III. SRS

3.1. Introduction

The SRS begins with purpose of the project, scope of the project, definitions, abbreviations, references and overview of SRS, in the Introduction section and continues with project perspective, project functions, assumptions and dependencies. Then covers functional and non-functional requirements.

3.1.1 Purpose

To integrate fuzzy web data tables for getting accurate data as well as nearest answers to the query.

3.1.2 Scope

User will be able to get data from the web data tables and use them for research.

3.1.3 Overview

Rest of the SRS contains General Description, Specific requirements and Analysis Models.

(1) In general description project perspective, what functionalities our project should implement, user characteristics, general constraints, assumptions and dependencies are explained.

(2) Specific requirements, interfaces, functional and non functional requirements and data flow diagrams are explained in the rest of the part.

3.2. General Description

3.2.1 Project Perspective

This project is going to provide data tables of personal, academic and fee details of the students.

3.2.2 Project Functions

- i) Create category wise listing of details.
- ii) Retrieve individual details of the students.
- iii) Generate automatic reports.

3.2.3 User Characteristics

A registered user can input a Student ID as a query to the system.

Administrator can input the records and generate the reports.

3.2.4 General Constraints

The project works only with the authenticated credentials.

3.2.5 Assumptions and Dependencies

We assume that the user will not provide meaningless words and long sentences as input. The Student ID does not exceed 10 characters and also not less than that.

3.3. Specific Requirements

3.3.1 External Interface Requirements

3.3.1.1 User Interface

The user having credentials should be able to get the correct results at any time using a web browser.

3.3.1.2 Hardware Interface

Normal computer hardware. Naturally including keyboard, mouse and a monitor with the better known parameters like 512MB RAM, 80GB hard disk, Pentium 4 or above processor.

3.3.1.3 Software Interfaces

Eclipse, Java swings and web browser like Mozilla firefox.

3.3.2 Functional Requirements

1) Admin must be able to input the records into the database.

2) Students must be able to retrieve individual details of the student.

3) Admin must be able to generate report and view records based on category.

3.3.3 Non-Functional Requirements

1) Performance

System should be as fast as possible.

2) Availability

System should be available 24 hours X 365 days in a year.

3) Security

Proper authentication mechanism should be used.

4) Maintainability

Further addition of any features should be possible.

5) Portability

System should be accessible from any type of computer like PCs and laptops.

IV LITERATURE SURVEY

Javaserver pages (jsp) is a technology that helps software developers create dynamically generated web pages based on html, xml, or other document types. Released in 1999 by sun microsystems, jsp is similar to php, but it uses the java programming language.

To deploy and run javaserver pages, a compatible web server with a servlet container, such as apache tomcat or jetty, is required. Architecturally, jsp may be viewed as a high-level abstraction of java servlets. Jsps are translated into servlets at runtime; each jsp servlet is cached and re-used until the original jsp is modified.

Jsp allows java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent java libraries, use java byte code rather than a native software format. Like any other java program, they must be executed within a java virtual machine (jvm) that integrates with the server's host operating system to provide an abstract platform-neutral environment.

Jsp pages use several delimiters for scripting functions. The most basic is `<% ... %>`, which encloses a jsp scriptlet. A scriptlet is a fragment of java code that is run when the user requests the page. Other common delimiters include `<%= ... %>` for expressions, where the scriptlet and delimiters are replaced with the result of evaluating the expression, and directives, denoted with `<%@ ... %>`.

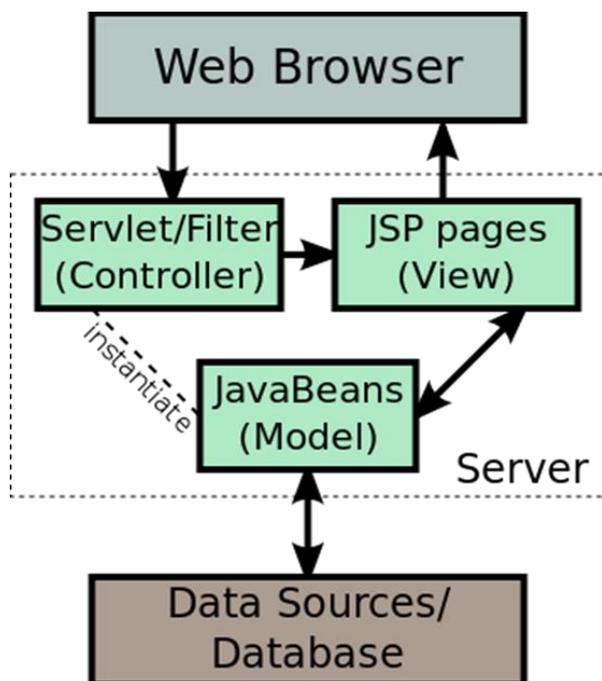


Figure : system architecture using jsp

Eclipse

In computer programming, eclipse is an integrated development environment (ide). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in java, eclipse can be used to develop applications. By means of various plug-ins, eclipse may also be used to develop applications in other programming languages: ada, abap, c, c++, cobol, fortran, haskell, javascript, lasso, lua, natural, perl, php, prolog,

python,r, ruby (including ruby on rails framework), scala, clojure, groovy, scheme, and erlang. It can also be used to develop packages for the software mathematica. Development environments include the eclipse java development tools (jdt) for java and scala, eclipse cdt for c/c++ and eclipse pdt for php, among others.

The initial codebase originated from ibm visualage. The eclipse software development kit (sdk), which includes the java development tools, is meant for java developers. Users can extend its abilities by installing plug-ins written for the eclipse platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Eclipse uses plug-ins to provide all the functionality within and on top of the runtime system. Its runtime system is based on equinox, an implementation of the osgi core framework specification

In addition to allowing the eclipse platform to be extended using other programming languages, such as c and python, the plug-in framework allows the eclipse platform to work with typesetting languages like latex and networking applications such as telnet and database management systems. The plug-in architecture supports writing any desired extension to the environment, such as for configuration management. Java and cvs support is provided in the eclipse sdk, with support for other version control systems provided by third-party plug-ins.

With the exception of a small run-time kernel, everything in eclipse is a plug-in. This means that every plug-in developed integrates with eclipse in exactly the same way as other plug-ins; in this respect, all features are "created equal.eclipse provides plug-ins for a wide variety of features, some of which are through third parties using both free and commercial models. Examples of plug-ins include for uml, for sequence and other uml diagrams, a plug-in for db explorer, and many others.

The eclipse sdk includes the eclipse java development tools (jdt), offering an ide with a built-in incremental java compiler and a full model of the java source files.

Apache tomcat is an open-source web server and servlet container developed by the apache software foundation (asf). Tomcat implements several java eespecifications including java servlet, javaserver pages (jsp), java el, and websocket, and provides a "pure java" http web server environment for java code to run in.

Mysql is a relational database management system (rdbms), and ships with no gui tools to administer mysql databases or manage data contained within the databases. Users may use the included command line tools, or use mysql "front-ends", desktop software and web applications that create and manage mysql databases, build database structures, back up data, inspect status, and work with data records. The official set of mysql front-end tools, mysql workbench is actively developed by oracle, and is freely available for use.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the

library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers.

Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Here we are using big data generated by colleges and universities to build an automated system for Institutional Administration.

Amazon Web Services (AWS) is a dynamic, growing business unit within Amazon.com. It is a Cloud platform which is alternative to Microsoft Azure.

V PROPOSED METHODOLOGY

5.1 proposed system model explanation

The administrator stores data in a data base. He/she can retrieve data any time, and generate the report. He/she can also see the results category wise. The students can register as a new user and login to the system. He/she can just see the individual details. The admin can only see the entire data base records and also category wise listing where as student has to enter the student id for getting a particular student detail.

A MIEL++ query system is used here. A MIEL++ query is an instantiation of a given view by the end user, by specifying, among the set of queryable attributes of the view, which are the selection attributes and their corresponding searched values, and which are the projection attributes. An important feature of a MIEL++ query is that searched values may be expressed as continuous or discrete fuzzy sets.^[3]

We have integrated the student tables based on fuzzy set to give the detailed report of the students academic, personal, and fee details. The reports generated by the system is in .xlsx format and it can be shared with the concerned staff.

5.2 Methodology Steps

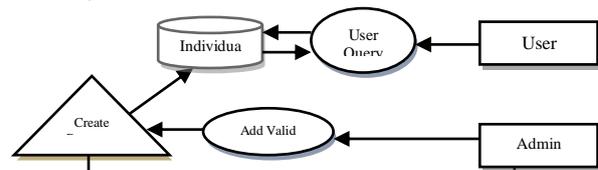
- 1) login as admin@onlineadmin.com with password admin.
- 2) click on admin dropdown and click on add student details.
- 3) add students' academic, personal and fee details.
- 4) see the details category wise.
- 5) generate report of student's academic, personal and fee details.
- 6) logout.
- 7) register new user. Select Tutor or Parent or Student and enter the ID given by the administrator.
- 8) login as normal user.
- 9) click on student dropdown and click on individual details.
- 10) enter the student id and submit. Click Generate report.
- 11) logout.

5.3 advantages in proposed system

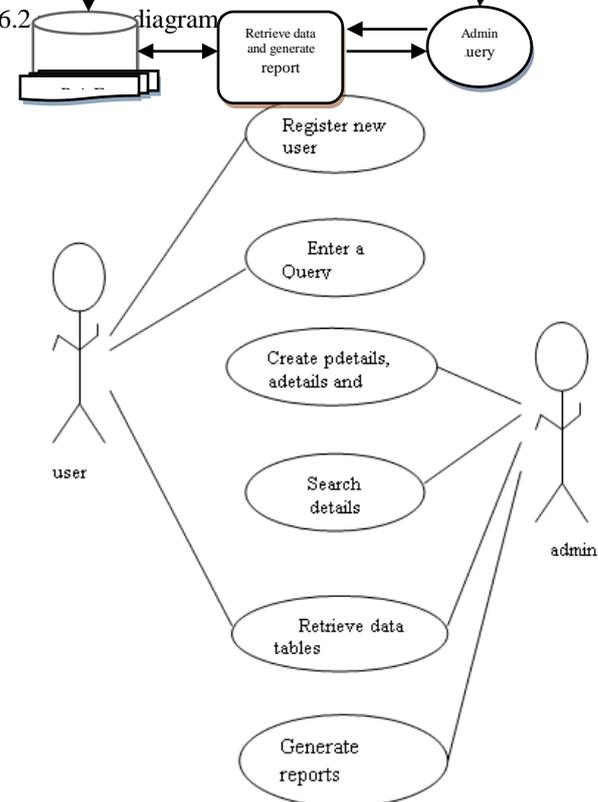
*everything is computerized. Students are given limited access compared to administrator

VI DESIGN DETAILS

6.1 system architecture



6.2 Use Case diagram



Use case diagram are usually referred to as behavior diagrams used to describe a set of actions that some system or systems (subject) should or can perform in collaboration with one or more external user of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Using Hadoop technology, we extend this to the Bigger Universities as well. Hadoop includes a fault tolerant storage system called the Hadoop Distributed File System. HDFS is able to store huge amounts of information, scale up incrementally and survive the failure of significant parts of the storage infrastructure without losing data.^[4]

VII TESTING

Software testing is a predominant verification and validation technique. Testing involves exercising the program using data like the real data processed by the program. The existence of program defects is inferred from unexpected system outputs. Testing may be carried out during the implementation phase to verify the software behaves, as intended by its designer and after the implementation is complete.

7.1 Goals of testing

- provide a mechanism to test (s)
- to be able to test single module (unit testing)
- to be able to test multiple modules (integration testing)

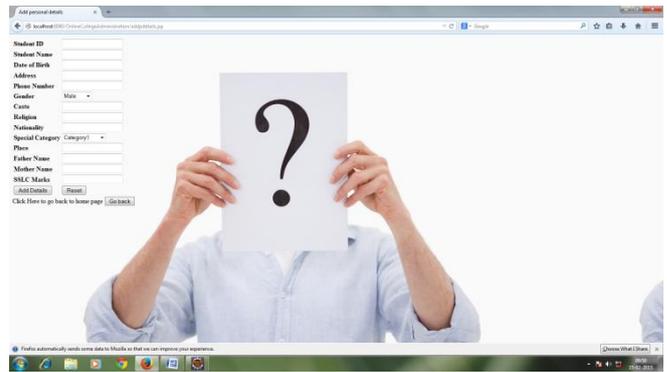
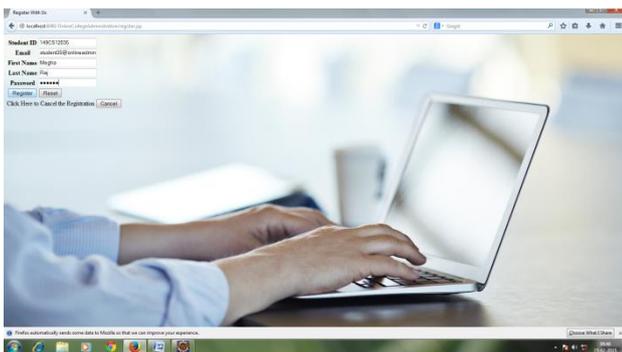
7.2 Test cases the system is tested using a number of test cases the important ones are listed below. The test description explains about what type of test is done and the expected result is noted prior to testing and observed against the test case. If the expected and observed value matches, then a decision is made that the test is pass else fail.

7.2.1 Unit testing the unit testing conducted during this phase involved testing of each module individually. We tested each functional requirement stated above in software requirement specification.

VII CONCLUSION

We prepared a system, called online college administration. OCA system allows admin to create student details, search the knowledge base and prepare reports based on data mining.

Category wise results and generate reports. It allows students to find their academic, fee and personal details related to college at their fingertip.



ACKNOWLEDGMENT

I acknowledge my Principal and Head of the Department for their sincere support.

REFERENCES

- [1] Kartiki Datarkar et al, "Online college managemnet system" International Journal of Computer Science and Mobile Computing, Vol.5 Issue.4, April- 2016, pg. 118-122
- [2] D.Sunitha et al, "Digital notice board using Smart Phones- Speech Recognition Voice command", Proceeding of 2018 IEEE International Conference on Current Trends toward Converging Technologies, Coimbatore, India
- [3] Vinayaka V.M., "Integration of Data Tables Using Ontological and Terminological Resource", Volume 8, Issue 6, ISSN 2321 3361 © 2018 IJESC
- [4] Asha T. et al, "Building Machine Learning Algorithms on Hadoop for Bigdata", International Journal of Engineering and Technology (IJET) – Volume 3 No. 2, February, 2013

