

INTRODUCTION

The system as the name clarifies is developed School. Colleges and Universities to help automate the admission procedure. The system is an internet based application which can be accessed from anytime anywhere basis. the system has three tier access models.

The increasing numbers of students seeking admission in the Academic Institutes (School, Colleges, Universities) are causing tremendous pressure on the administrative body of the institutes to manage and arrange the admission process manually. It is difficult now to conduct the process accurately and in timely manner. Hence, the need for online admission is inevitable.

In case of a manual system, it is a time taking process and involves huge manpower wherein the online admission system ensures accurate and very fast computerized information. Maintaining backup is also very easy using 'Online Admission System'. This system is also helpful for the administrator because he/she can easily bring changes to the records of the students. The mobile application would require connecting to the database on a remote server using Wi-Fi technology.

Our system primarily focuses on building an efficient and user friendly Android mobile application for a Cloud based Intra-College Communication Information System using Mobile Clients. The application will be installed on the users (student/teachers) Smartphones. Here the concept of unique ID is also included using which the each student gets one unique identification number by email. This id will help to access his info or find him from multiple students

The goal of 'Online Admission System' is to automate the Academic Institute's admission structure and its related operation and functionality. The objective of the initiative is to provide support to the administration and admission seeking candidates by providing a faster, transparent and easy way of keeping records and use them for reference and further proceedings

3. SYSTEM SPECIFICATION

2.1 HARDWARE SPECIFICATION

CPU: Pentium VI Processor

RAM: 2 GB

HDD: 40GB

Key Board, Monitor, Mouse, Printer

2.2 SOFTWARE SPECIFICATION

OPERATING SYSTEM: Microsoft Windows 10

FRONT END : [ASP.NET](#)

BACK END : MS ACCESS

4.SYSTEM SPECIFICATION

ASP.NET

- **Definition:** ASP.NET is an open-source, server-side web application framework.
- **Purpose:** Designed for web development to create dynamic web pages.
- **Developer:** Developed by Microsoft.
- **Features:** Allows programmers to build dynamic web sites, web applications, and web services.
- **Web Development Platform:** Offers a programming model, comprehensive software infrastructure, and various services to create robust web applications.
- **Compatibility:** Supports PC and mobile devices.
- **Communication:** Uses HTTP protocol and commands for browser-to-server communication.
- **Part of .NET Platform:** ASP.NET applications are compiled codes that utilize extensible and reusable components within the .NET framework.
- **Programming Languages:** ASP.NET applications can be written in:
 - C#
 - Visual Basic.Net
 - Javascript
 - J#
- **Interactive Applications:** ASP.NET produces interactive, data-driven web applications that use controls like text boxes, buttons, and labels for assembly, configuration, and code manipulation.
- **HTML Pages:** ASP.NET code can be used to create HTML pages.

ASP.NET Web Forms Model

- **Event-Driven Model:** Extends the event-driven model of interaction to web applications.
- **Web Form Submission:** The browser submits a web form to the web server.
- **Server Response:** The server returns a full markup page or HTML page in response.

ASP.NET Framework & Component Model

- **Client-Side User Activities & Server-Side Processing:**
 - Client-side user activities are forwarded to the server for stateful processing.
 - The server processes the output of client actions and triggers the reactions.
- **ASP.NET & HTTP Protocol:**
 - HTTP is a stateless protocol.
 - ASP.NET framework helps in storing application state information.
- **Application State Components:**
 - **Page State:** The state of the client, i.e., the content of various input fields in the web form.
 - **Session State:** The collective information obtained from various pages visited and worked with, i.e., the overall session state.
 - **Example:** A shopping cart where items are selected from one page and the total collected items and price are shown on a different page. ASP.NET session state and server-side infrastructure keep track of this information across pages.
- **ASP.NET Runtime & State Management:**
 - The ASP.NET runtime carries the page state to and from the server across page requests.
 - It generates ASP.NET runtime codes and incorporates the state of the server-side components in hidden fields.
 - This enables the server to be aware of the overall application state and operate in a two-tiered connected way.
- **ASP.NET Component Model:**
 - Provides various building blocks of ASP.NET pages.
 - This is an object model that describes:
 - **Server-Side Counterparts:** Server-side counterparts of almost all HTML elements or tags.
 - **Server Controls:** Server controls that help in developing complex user-interface, e.g., Calendar control or Gridview control.

- **ASP.NET & .NET Framework:**

- ASP.NET is a technology that works on the .NET framework, which contains all web-related functionalities.
- The .NET framework is made of an object-oriented hierarchy.

ASP.NET runtime transforms the .aspx page into an instance of a class, which inherits from the base class page of the .Net framework.

- Each ASP.NET page is an object.
- All its components, like server-side controls, are also objects.
- ASP.NET is a unified Web development model that includes the services necessary to build enterprise-class Web applications with a minimum of coding.
- ASP.NET is part of the .NET Framework.
- While coding ASP.NET applications, you have access to classes in the .NET Framework.
- You can code your applications in any language compatible with the common language runtime (CLR).
- The languages include Microsoft Visual Basic and C#.
- These languages enable you to develop ASP.NET applications that benefit from the common language runtime, type safety, inheritance, and so on.

Components of .Net Framework

The table describes the components of the .Net framework 3.5 and the job they perform:

(1) Common Language Runtime or CLR

It performs memory management, exception handling, debugging, security checking, thread execution, code execution, code safety, verification, and compilation.

The code that is directly managed by the CLR is called the managed code.

When the managed code is compiled, the compiler converts the source code into a CPU independent intermediate language (IL) code.

A Just In Time(JIT) compiler compiles the IL code into native code, which is CPU specific.

(2) .Net Framework Class Library

It contains a huge library of reusable types. classes, interfaces, structures, and enumerated values, which are collectively called types.

(3) Common Language Specification

It contains the specifications for the .Net supported languages and implementation of language integration.

(4) Common Type System

- It provides guidelines for declaring, using, and managing types at runtime, and cross-language communication.

(5) Metadata and Assemblies

- Metadata is the binary information describing the program, which is either stored in a portable executable file (PE) or in the memory.
- Assembly is a logical unit consisting of the assembly manifest, type metadata, IL code, and a set of resources like image files.

(6) Windows Forms

- Windows Forms contain the graphical representation of any window displayed in the application.

(7) ASP.NET and ASP.NET AJAX

- ASP.NET is the web development model and AJAX is an extension of ASP.NET for developing and implementing AJAX functionality.
- ASP.NET AJAX contains the components that allow the developer to update data on a website without a complete reload of the page.

(8) ADO.NET

- It is the technology used for working with data and databases.
- It provides access to data sources like SQL server, OLE DB, XML etc.

- The ADO.NET allows connection to data sources for retrieving, manipulating, and updating data.

(9) Windows Workflow Foundation (WF)

- It helps in building workflow-based applications in Windows.
- It contains activities, workflow runtime, workflow designer, and a rules engine.

(10) Windows Presentation Foundation

- It provides a separation between the user interface and the business logic.
- It helps in developing visually stunning interfaces using documents, media, two and three dimensional graphics, animations, and more.

(11) Windows Communication Foundation (WCF)

- It is the technology used for building and executing connected systems.

(12) Windows Card Space

- It provides safety for accessing resources and sharing personal information on the internet.

(13) LINQ

- It imparts data querying capabilities to .Net languages using a syntax which is similar to the tradition query language SQL.

User controls

- User controls are encapsulations of sections of page sections that are registered and used as controls in ASP.NET

Custom controls

- Programmers can also build custom controls for ASP.NET applications. Unlike user controls, these controls do not have an ASCX markup file, having all their code compiled into a

dynamic link library (DLL) file. Such custom controls can be used across multiple Web applications and Visual Studio 2013 projects.

Rendering technique

- .NET uses a "visited composites" rendering technique. During compilation, the template (.aspx) file is compiled into initialization code that builds a control tree (the composite) representing the original template. Literal text goes into instances of the Literal control class, and server controls are represented by instances of a specific control class. The initialization code is combined with user-written code (usually by the assembly of multiple partial classes) and results in a class specific for the page. The page doubles as the root of the control tree.
- Actual requests for the page are processed through a number of steps. First, during the initialization steps, an instance of the page class is created and the initialization code is executed. This produces the initial control tree, which is now typically manipulated by the methods of the page in the following steps. As each node in the tree is a control represented as an instance of a class, the code may change the tree structure as well as manipulate the properties/methods of the individual nodes. Finally, during the rendering step a visitor is used to visit every node in the tree, asking each node to render itself using the methods of the visitor. The resulting HTML output is sent to the client.

MS ACCESS

- Microsoft Access is a Database Management System (DBMS) from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software development tools. It is a member of the Microsoft Office suite of applications, included in the professional and higher editions.
 - Microsoft Access is just one part of Microsoft's overall data management product strategy.
 - It stores data in its own format based on the Access Jet Database Engine.
 - Like relational databases, Microsoft Access also allows you to link related information easily. For example, customer and order data. However, Access 2013 also complements other database products because it has several powerful connectivity features.
 - It can also import or link directly to data stored in other applications and databases.
 - As its name implies, Access can work directly with data from other sources, including many popular PC database programs, with many SQL (Structured Query Language) databases on the desktop, on servers, on minicomputers, or on mainframes, and with data stored on Internet or intranet web servers.
 - Access can also understand and use a wide variety of other data formats, including many other database file structures.
 - You can export data to and import data from word processing files, spreadsheets, or database files directly.
 - Access can work with most popular databases that support the Open Database Connectivity (ODBC) standard, including SQL Server, Oracle, and DB2.
 - Software developers can use Microsoft Access to develop application software.
- Microsoft Access stores information which is called a database. To use MS Access, you will need to follow these four steps

Database Creation

- Create your Microsoft Access database and specify what kind of data you will be storing.

Data Input

- After your database is created, the data of every business day can be entered into the Access database.

Query

- This is a fancy term to basically describe the process of retrieving information from the database.

Report (optional)

- Information from the database is organized in a nice presentation that can be printed in an Access Report.

Architecture

- Access calls anything that can have a name an object. Within an Access desktop database, the main objects are tables, queries, forms, reports, macros, data macros, and modules.
- If you have worked with other database systems on desktop computers, you might have seen the term database used to refer to only those files in which you store data.
- But, in Access, a desktop database (.accdb) also includes all the major objects related to the stored data, including objects you define to automate the use of your data.
- The Access Jet engine supports most database features, such as referential integrity, row-level locking and multiuser support with up to 255 concurrent user connections.
- Access is well-suited for creating small databases such as those used by single departments. It is not suitable for larger, enterprise-level databases. For these, Microsoft offers its SQL Server database engine.
- Access was first released as version 1.0 in November 1992. The software was fine while working with small databases but in larger files there was the danger of data corruption. With each release of Office, Microsoft has also included a newer version of Access, each of which includes new or updated features.

- With the release of Access 2007, the database file format changed from the previous ".mdb" to ".accdb". This new format supports more complex data types, but unfortunately is not compatible with prior versions of the Access software. This is

Components of MS Access

- There are seven major components of Microsoft Access. When we create a component in Access it is an object, and several similar objects grouped together will constitute a class. Let us take a look at the components.

Tables

- In MS Access Tables store the data or the information that you feed the program. The data table can be created with the required field names, properties etc as desired by the user. If the table is not created correctly, the database may be slow in giving results or give wrong results entirely. So it is essential the table be created properly and carefully.

Relationships

- Relationships are the connections that form between one or more table. So the relationships can be one-on-one, one-to-many and then of course many-to-many.

Queries

- Queries are when the user or the programmer requests the database for information. The objects help create a SQL compatible query, which can help store data and retrieve information from the database. Queries can also sort and filter data as per the query statement.

Forms

- This is an object class which allows the programmer/designer to create a user interface for MS Access. A form is defined by tables and queries.

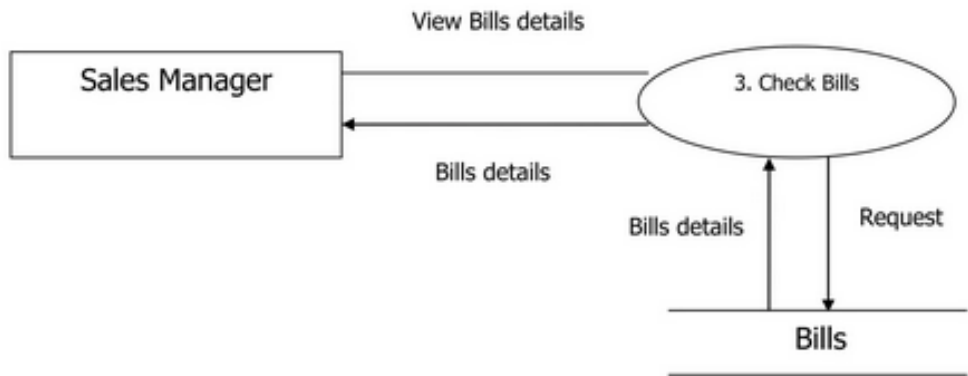
Reports

- Once all the data has been entered into the database, the user will want to view the information. For this he can run reports which will allow him to view the information, or summarize the information. The reports can be customized by the user. The source of the information in these reports are the tables and the queries.

4.PROJECT DESCRIPTION

- **Purpose:** To streamline the process of managing inventory and generating invoices for sales.
- **Key Features:**
 - **Inventory Management:**
 - Track stock levels, reorder points, and inventory history.
 - Manage multiple warehouses or locations.
 - Generate purchase orders and track supplier information.
 - **Sales and Billing:**
 - Create sales orders and invoices.
 - Manage customer accounts and track payment history.
 - Calculate discounts, taxes, and shipping costs.
 - **Reporting and Analytics:**
 - Generate reports on sales, inventory, and customer activity.
 - Analyze trends and identify areas for improvement.
- **Benefits:**
 - **Improved Efficiency:** Automates tasks like order processing and invoicing.
 - **Accurate Inventory Tracking:** Prevents stockouts and overstocking.
 - **Enhanced Customer Service:** Provides faster order fulfillment and accurate billing.
 - **Financial Visibility:** Tracks sales, revenue, and profitability.
 - **Reduced Errors:** Minimizes manual data entry and potential for mistakes.
- **Typical Users:**
 - **Retailers:** Businesses that sell products directly to consumers.
 - **Wholesalers:** Businesses that sell products to other businesses.
 - **Manufacturers:** Businesses that produce goods for sale.
 - **Service Providers:** Businesses that offer services and need to track inventory of materials or supplies.
- **Integration:**
 - **E-commerce Platforms:** Connect with online stores for automated order fulfillment.
 - **Point of Sale (POS) Systems:** Integrate with POS systems for seamless inventory management.

- **Accounting Software:** Sync with accounting software for financial reporting.
- **Types of Systems:**
 - **On-premise:** Software installed and maintained on a company's own servers.
 - **Cloud-based:** Software accessed through the internet.
- **Selection Criteria:**
 - **Scalability:** Ability to handle future growth in inventory and sales.
 - **Features:** The features needed to meet specific business requirements.
 - **Cost:** The cost of the software and any ongoing maintenance or support fees.
 - **Ease of Use:** User-friendliness and ease of training for staff.
 - **Integration Capabilities:** Ability to integrate with other systems.
- **Implementation:**



DFD Level 1.3 – Function Process : Check Bill

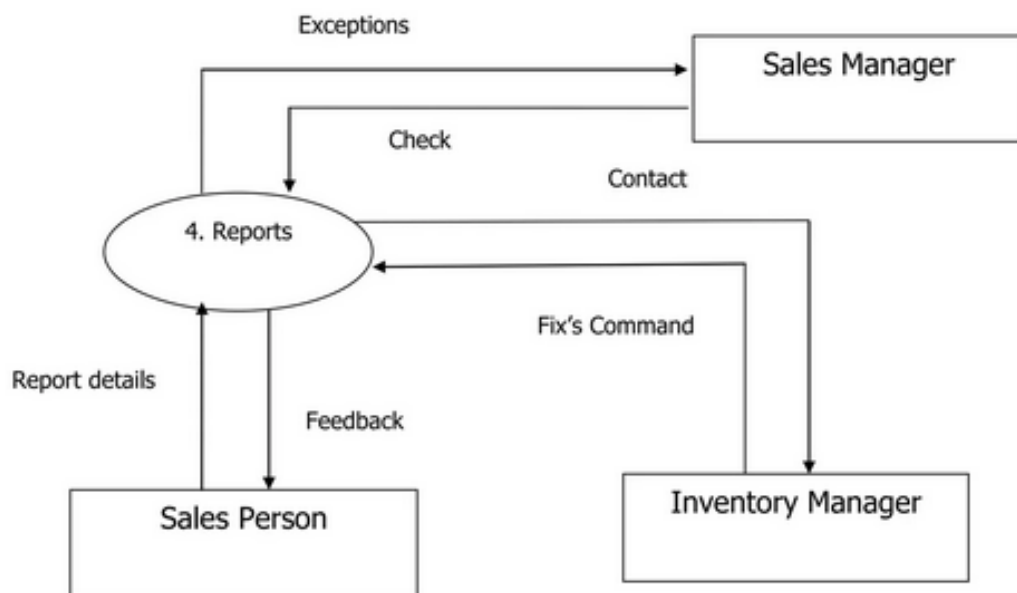


Table Design: Customer Table

S.NO	FIELD NAME	DATA TYPE
1	CustomerID	INT
2	CustomerName	VARCHAR
3	ContactNumber	VARCHAR
4	Email	VARCHAR
5	Address	Text

Table Design: product

S.NO	FIELD NAME	DATA TYPE
1	ProductID	INT
2	ProductName	VARCHAR
3	Category	VARCHAR
4	Price	DECIMAL (10,2)
5	QuantityInStock	INT

Table Design: Users Table

S.NO	FIELD NAME	DATA TYPE
1	Name	Text
2	Email	Text
3	Password	Text

CONCLUSION

The project sought to develop an online fees payment system that addresses the challenges associated with traditional modes of paying fees in the university. Issues faced by students and their sponsors, such as long queues, manual processing delays, and limited accessibility, were thoroughly identified and analyzed. In response, a web-based solution was designed and implemented, enabling seamless fee payments using credit and debit cards. This system not only simplifies the payment process but also allows users to make payments conveniently from any location, reducing the burden of in-person transactions.

Through testing and user feedback, the system proved to be efficient, reliable, and user-friendly. Both students and their sponsors expressed satisfaction with its functionality, emphasizing its potential to eliminate most of the recurring challenges associated with manual fee payments. Furthermore, the system's scalability and integration capabilities ensure that it can accommodate future enhancements, such as payment tracking, reminders, and integration with other university management systems.

BIBLIOGRAPHY

- [1] Borah M.D., Application of knowledge based decision technique to predict student enrollment decision, Recent Trends in Information Systems (ReTIS), 21-23 Dec. 2011,180-184.
- [2] Ragab, A.H.M., Hybrid recommender system for predicting college admission, Intelligent Systems Design and Applications (ISDA), 29 Nov. 2012, 107-113.
- [3] Rensong Dong, The module of prediction of College Entrance Examination aspiration, Fuzzy Systems and Knowledge Discovery (FSKD), 31 May 2012,1559-1562.