

Contents

1	INTRODUCTION	1
1.1	Project Profile	1
2	ABOUT THE DEVELOPING TOOLS	2
2.1	Introduction to ASP.NET	2
2.1.1	C# and Features	5
2.2	SQL Server 2008	7
2.2.1	Features of Microsoft SQL Server 2008	8
2.3	GitHub	9
3	SYSTEM ANALYSIS	10
3.1	Introduction	10
3.2	Existing System	11
3.3	Limitations Of Existing System	11
4	Feasibility Study	12
4.1	Economic Feasibility	12
4.2	Technical Feasibility	13
4.3	Operational Feasibility	13
4.4	Behavioural Feasibility	13
5	Proposed System	14
5.1	Advantages of Proposed System	14
6	FACT FINDING TECHNIQUES	15
6.1	Interview:	15
6.2	Record View:	15
6.3	Onsite observation:	15
7	SYSTEM SPECIFICATION	16
7.1	Hardware Specification:-	16
7.2	Software Specification:-	17
8	SYSTEM DESIGN	18
8.1	Introduction of System Design	18
8.2	Module Design	18
8.3	Input Design	20
8.4	Output Design	21
8.5	Database Design	22
8.6	Architectural Design	22

8.7	Form Design	23
8.8	Table Structure	24
8.9	Data Flow Diagram	28
9	SYSTEM TESTING	34
9.1	Introduction to System Testing	34
9.2	Unit Testing	35
9.3	Integration Testing	35
9.4	Validation Testing	36
9.5	User Acceptance Testing	36
9.6	Alpha Testing	36
9.7	Beta Testing	37
9.8	Test Cases	37
9.9	Bugzilla	38
10	SYSTEM IMPLEMENTATION	39
10.1	Introduction to System Implementation	39
10.2	Training	40
10.3	Conversion	41
10.4	Post Implementation Review	41
10.5	System Maintenance	42
11	SYSTEM EVALUATION	43
12	CONCLUSION	44
13	SCOPE FOR FUTURE ENHANCEMENTS	45
14	APPENDIX	46
14.1	APPENDIX A	46
14.1.1	Sample Source Code / Pseudo Code	46
14.2	APPENDIX B	61
14.2.1	Screen Shots	61

1 INTRODUCTION

1.1 Project Profile

The project named as Automated ER Diagram Maker deals with the designing of automated ER diagrams. This software will overcome all the disadvantages of the existing system. This project deals with four processes named Registration, Capture , Analysis and Representations. This system may have mainly four processes. They are registration ,capture, analysis and representation. The registration process is used to register the users. The capture includes collection of details; analysis includes the analysis of provided data. Finally as respond to all other process the desired output will be represented by the last process named representation.

2 ABOUT THE DEVELOPING TOOLS

2.1 Introduction to ASP.NET

ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites. It allows you to use a full featured programming language such as C# or VB.NET to build web applications easily. ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices. ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation. ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework. The ASP.NET application codes can be written in any of the following languages:

- C#
- Visual Basic.Net
- Jscript
- J#

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

- **Enhanced Performance:** ASP.NET is compiled common language runtime code running on the server. Unlike interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
- **World-class Tool Support:** A rich toolbox and designer in the visual studio integrated development environment complement of the ASP.NET framework. WYSIWYG editing, drag-and-drop server controls and automatic deployment are just a few of the features this powerful tool provides.
- **Power and Flexibility:** Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers. The .NET framework class library, Messaging and Data Access solutions are all seamlessly accessible from the Web.
- **Simplicity:** ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic-like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

- **Manageability:** ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied
- **Without the aid of local administration tools.** The Zero local administration philosophy extends to deploying ASP.NET framework applications as well. An ASP.NET Framework application is deployed to server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.
- **Scalability and Availability:** ASP.NET has been designed with scalability in mind, with features specifically tailored to improved performance in clustered and multiprocessor environments. Further, processes are closely monitored and managed by the ASP.NET Runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps your application constantly available to handle requests.
- **Customizability and Extensibility:** ASP.NET delivers a well-factored architecture that allows developers to plug-in their code at appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET with your own custom-written component. Implementing custom authentication or state services has never been easier.
- **Security:** With built in Windows authentication and per-application configuration, assured that applications are secure.

2.1.1 C# and Features

Microsoft C# is a new programming language designed for building a wide range of enterprise applications that run on the .NET Framework. C# code is compiled as managed code, which means it benefits from the services of the common language runtime. These services include language interoperability, garbage collection, enhanced security and improved versioning support. The C# language is an evolution of C and C++. It uses many C++ features in the areas of statements, expressions and operators. C# provides access to the common API styles: .NET Framework, COM, Automation and C-style APIs. It also supports unsafe mode, where you can use pointers to manipulate memory that is not under the control of the garbage collector.

C# is introduced as Visual C# in the Visual Studio, Net suit. Support for Visual C# includes project templates, designers, property pages, code wizards, an object model and other features of the development environment. The library for Visual C# programming is the .NET Framework. C# is an elegant, simple, type-safe, object-oriented language that allows enterprise programmers to build a breadth of applications. C# also gives you the capability to build durable system-level components by virtue of the following features:

- Full COM/Platform support for existing code integration.
- Robustness through garbage collection and type safety.
- Security provided through intrinsic code trust mechanisms.
- Full support of extensible metadata concepts.

You can also interoperate with other languages, across platforms, with legacy data, by virtue of the following features:

- Full interoperability support through COM+ 1.0 and .NET Framework services with tight library-based access.
- XML support for Web-based component interaction.
- Versioning to provide ease of administration and deployment.

Visual C#.NET currently part of the Visual Studio .NET .Microsoft Visual C#.Net is the tool set for creating XML Web services and Microsoft .NET connected applications for Microsoft Windows and the Web. This robust development package which uses the component oriented C# development language, offers a language and environment for creating next generation software. Visual C#.NET delivers superior functionality for streamlining business processes, including Rapid design, development and deployment support for creating and consuming XML Web services.

- Form designers and visual control for creating rich Windows-based applications.
- Authorizing tools and services for building powerful Microsoft .NET sever-based solutions.
- Migration tools for converting Java-based projects to the Microsoft .NET development environment.

With Visual C# .NET, the developers can build solutions for the broadest range of clients, including Windows, the Web, and mobile or embedded devices. Using this elegant programming language and tool, developers can leverage their existing C++ and Java skills and knowledge to be successful in the .NET environment.

2.2 SQL Server 2008

MS SQL Server is database management software, which is rich and powerful application. It is a language that enables us to create and operate on relational databases, which are sets of related information stored in tables. Because of its elegance and independence from machine specifics, as well as its support by the industry leaders in relational database technology, SQL become the standard language and will remain as it for the foreseeable future. The SQL standard is recognized by ANSI and is currently accepted by ISO. Even though most commercial database programs extend SQL beyond the ANSI definition, we will follow the ANSI standard with an eye towards the most common variations. SQL also makes it easy for the user's even beginners to work with the database. We can create tables, edit data and use queries to find the data we want with very little effort and time.

2.2.1 Features of Microsoft SQL Server 2008

- **Internet Integration.** The SQL Server 2008 database engine includes integrated XML support. It also has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites.
- **Scalability and Availability.** The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows 98 through large, multiprocessor servers running Microsoft Windows 2008 Data Center Edition. SQL Server 2008 Enterprise Edition supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Web sites.
- **Enterprise-Level Database Features** The SQL Server 2008 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database.
- **Ease of installation, deployment, and use** SQL Server 2008 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing, and using SQL Server across several sites. SQL Server 2008 also supports a standards-based programming model integrated with the Windows DNA, making the use of SQL Server databases and data warehouses a seamless part of building powerful and scalable systems.
- **Data warehousing** SQL Server 2008 includes tools for extracting and analyzing summary data for online analytical processing. SQL Server also includes tools for visually designing databases and analyzing data using English-based questions.

2.3 GitHub

Version control is a system that manages changes to a file or files. These changes are kept as logs in a history, with detailed information on what file(s) was changed, what was changed within the file, who changed it, and a message on why the change was made. This is extremely useful, especially when working in teams. To understand how incredibly powerful version control is: How many files of different versions of a manuscript or thesis do you have laying around after getting feedback from your supervisor or co-authors? Have you ever wanted to experiment with your code or your manuscript and need to make a new file so that the original is not touched? Have you ever deleted something and wish you hadn't? Have you ever forgotten what you were doing on a project? All these problems are fixed by using version control (git)!

3 SYSTEM ANALYSIS

3.1 Introduction

System Analysis works with users to identify goals and build system to achieve them. System Analysis is an important phase of any system development process. System analysis is a step-by-step process used to identify and develop or acquire the software need to control the processing of specific application. System analysis is a continuing activity the stages of the systems development. The system is studied to the minutes details and analyzed. In analysis, a detailed study of these operation performed by a system and their relationships within and outside of the system is done.

The aim of the proposed system is to develop a system with improved facilities. The proposed system can overcome all the limitation of the existing system, such as information is maintained in the database, it gives more security to data, ensures data accuracy, reduces paper work and save time, only eligible students get chance, it makes information flow efficient and paves way for easy report generation, reduce the space. proposed system is cost effective.

3.2 Existing System

The manual system is backed by several numbers of disadvantages. Number of error and frauds will be more. There will be more wastage of energy and time. By making this into computerized we can maintain high accuracy and reliability. To overcome these problems we are implementing a computerized system with combination of Visual Basic and Microsoft SQL Server 2005 .

3.3 Limitations Of Existing System

The major limitations of existing system are

1. Limitations in diagram generation.
2. Excess manpower is wasted.
3. More time needed.
4. Chance of errors.
5. Difficulty in correcting errors.

4 Feasibility Study

After the problem is clearly understood and solutions are proposed, the next step is to conduct the feasibility study, which is the part of system analysis as well as system design process.

The main objective of this study is to determine whether the proposed system is feasible or not .Mainly there are four types of feasibility study.

1. Economic feasibility
2. Technical feasibility
3. Behavioural Feasibility
4. Operational feasibility

4.1 Economic Feasibility

Justification of any salary is that it will reduce loss of pay, improve the quality of generating reports, which in turn may be expected to provide increase user friendly. The technique of cost benefit analysis is often used as a basis for accessing economic feasibility.

4.2 Technical Feasibility

The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures. This can be qualified in terms of volume of data, trend, frequency of updating, cycle of activity etc., in order to give introduction of technical system.

4.3 Operational Feasibility

Proposed project are beneficial only if they can be turned into information systems. That will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. The purpose of the operational feasibility study is to determine whether the new system will be used if it is developed and implemented. And whether there will be resistance from users that will undermine the possible application benefits.

4.4 Behavioural Feasibility

People are inherently resisted to change the computers have been known to facilitate change. Since a new proposed system has nothing to do with the ordinary staff, the worker resistance to the system is very much less. The users might be scared of job losses and other problems when the new system is introduced.

5 Proposed System

This system studies the problems existing in the manual functioning and tries to resolve them by the computerization of the apartment management. Proposed system is an interactive system, highly user-friendly, and designed exclusively for apartment management.

5.1 Advantages of Proposed System

The proposed system simplifies the task of storage of data and diagram generation. This system is fully computerized thus making it user friendlier and is more efficient, reliable, fast and accurate.

The advantages of the proposed system are

1. Use of new technology.
2. Search facility is more advance.
3. Speed in processing
4. Reduce manpower
5. High security is provided in the software. So that the data remains confidential and tampers proof.
6. Validity of input database is automatically checked and errors will be immediately signalled.

6 FACT FINDING TECHNIQUES

The success of any project depends upon the accuracy of available data. Accurate information can be collected with the help of certain methods / techniques. These specific methods for finding information of the system are termed as fact finding techniques. Interview, Questionnaire, Record View and Observations are the different fact finding techniques used in this project.

6.1 Interview:

This method is used to collect the information from groups or individuals. We select the people who are related with the system for the interview. In this method, we sit face to face with people and record their responses.

6.2 Record View:

The information related to the system is available in the source like company's documents, websites and other records. This record review helped me to get valuable information about the system.

6.3 Onsite observation:

Unlike the other fact finding techniques, in this method we visit the organization and observe and understand the working of the existing system, flow of the system, the users of the system etc.

7 SYSTEM SPECIFICATION

7.1 Hardware Specification:-

- Processor Intel core i3-2370M
- System Bus 32 BIT
- RAM 4.00GB
- Hard disk 298.09GB
- Monitor HP

7.2 Software Specification:-

- Operating System Windows XP Professional or Higher
- IDE Microsoft Visual Studio 2008
- Front-end asp.net
- Scripting Language HTML
- Back-end MYSQL SERVER
- Web Server IIS 8.0
- Browser Internet Explorer, Mozilla Firefox, Google Chrome

8 SYSTEM DESIGN

8.1 Introduction of System Design

System designs main aim is to identify the modules that should be in the system, and the specifications of these modules and how they interact with each other to produce the desired results. At the end of the system design all the major data structures, file formats and the major modules in the system and their specification are decided.

8.2 Module Design

1. Registration
 - User Registration
2. Capture
 - Entities
 - Relationship
 - Attributes
3. Analysis
 - Entered data
4. Representation
 - ER Diagram

1. Registration

The admin have most privilege in the registration process. Only the admin can control the registration of users. The admin have the supreme power all over the process taken place in the automated ER Diagram Maker.

2. Capture

In the capture process all the information that has to be represented in the diagram is capturing. For this purpose there are several forms are designed. The user can enter data that has to be displayed in the diagram.

3. Analysis

The analysis is the process in which the accuracy is tested and appropriate design process will be in progress in this stage. The final diagram will be produced only if the entered data is accurate. If the entered data is accurate then the system will move on to the next process named representation.

4. Representation

In the final process named representation the diagram is represented on the screen which can be used for the further development of the project in development. So the desired final output will be represented in the screen by the end of the process named representation. Automated ER Diagram Maker is software developed using Asp.net language as front end and SQL Server as back end. Our project is the complete solution for managing all activities for the development of an ER Diagram maker.

8.3 Input Design

Inaccurate input data is the most cause of errors in data processed. Errors by the data entry operators can be controlled by input validation. Input design is the process of converting user-oriented inputs to a computer based format. Input design is a part of an overall system design, which makes the system user friendly. Input design is the link between their formation system and users. It comprises developing specification and those steps that are necessary to put transactions data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by instructing the computer to read from a written or printed document or it can occur by having people key data directly in to the system. The design of input focuses on controlling the amount of inputs required, controlling errors, avoiding delay , avoiding extra steps and keeping the process simple. The system analysis decides the following input design details:

- What data to input.
- What medium to use.
- How the data should be arranged or coded
- The dialogue to guide the users in providing input.

Methods for performing input validations and steps to follow when error occurs. The input screens are displayed in a way that the user can understand entries very easily. Proper validation and controlling of input data will reduce the chance of errors. All major operations are done only after checking the results of confirmation message. This makes the system more reliable and user friendly.

8.4 Output Design

Efficient and transport output design improves the system relationship with the user and help him to understand faster. A complicated output design contributes nothing to the system but leave the user confused. For a successful system it is a must that it maintains a clear output design. The users of the output its purpose and sequence of details to be printed are all considered. The output form is the justification for its existence. If the outputs are inadequate in any way the system itself is inadequate. Hence it is necessary to design output so that the objectives of the system are met in the best possible manner. The outputs are in the form of reports. Output design is a process that in valves designing necessary output in the form of reports that should be given to the users according to the requirements. While designing the output, the following steps are considered.

- Determine what information to be presented to the user.
- Arrange the presentation of information in an acceptable format.
- Decide how to distribute the output to the intended recipients.

This project has a wide range of flexible diagrams that are accurate with less ambiguity. This includes

8.5 Database Design

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage. The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

1. Determine the data to be stored in the database.
2. Determine the relationships between the different data elements.
3. Superimpose a logical structure upon the data on the basis of these relationships.

In this project database design generally the data is to be stored in the database whether it can more relation for each modules. And it provides the logical relation between them.

8.6 Architectural Design

Architectural design is of crucial importance in software engineering during which the essential requirements like reliability, cost, and performance are dealt with. Architectural design is the responsibility of developers, some other people like user representatives, systems engineers, hardware engineers, and operations personnel are also involved. All these stakeholders must also be consulted while reviewing the architectural design in order to minimize the risks and errors.

8.7 Form Design

A form designing means deciding the contents and layout of forms for the purpose of collecting and processing the required information economically and efficiently. The importance of forms designing can be understood because of the following points:

1. Forms are used to collect record and communicate the required information according to the expectations of the needy persons. Therefore, forms are treated as tools of office work. If the forms are badly designed, it reduces the speed of operation of office work.
2. The forms create psychological impact on the people who use it. The people may be frustrated and get tired if the forms are not designed properly.
3. The badly designed forms results in more number of mistakes in clerical work. Hence, there is a need of well-designed forms to avoid mistakes in clerical work.
4. Sometimes, the designed form may project a poor image in the minds of the customers. This may adversely affect the good will of the company.
5. System is the basis for form design. Hence, forms are designed according to the needs of the system. If forms are badly designed, they can ruin a whole system.
6. The well-designed forms contribute much to the efficiency of employees of an organization and efficiency of the system.
7. The cost of forms is less than the cost of completing office forms, transporting and filling of office forms. The ratio will be greater if the forms are badly designed.

8.8 Table Structure

Table Number : 1

Table Name : Table Attribute

Primary Key : Attribute_id

Field Name	Data type	Description	Constraints
Attribute_id	Int	Attribute_id	Primary key
AttributeName	varchar(50)	AttributeName	Not null
Entity_id	Int	Entity_id	Foreign key

Table Number : 2

Table Name : Table Derived Attribute

Primary Key : Attribute_id

Field Name	Data type	Description	Constraints
DerivedAttribute_id	Int	DerivedAttribute_id	Primary key
DerivedAttributeName	varchar(50)	DerivedAttributeName	Not null
Entity_id	Int	Entity_id	Foreign key

Table Number : 3

Table Name : Table Entity

Primary Key : Entity_id

Field Name	Data type	Description	Constraints
Entity_id	Int	Entity_id	Primary key
EntityName	varchar(50)	EntityName	Not null
User_id	Int	User_id	Foreign key

Table Number : 4

Table Name : Table Login

Primary Key : User_id

Field Name	Data type	Description	Constraints
User_id	Int	User_id	Primary key
UserName	varchar(50)	UserName	Not null
Password	Int	Password	Not null

Table Number : 5

Table Name : Table MultyValuedAttribute

Primary Key : MultyValuedAttribute_id

Field Name	Data type	Description	Constraints
MultyValuedAttribute_id	Int	MultyValuedAttribute_id	Primary key
MultyValuedAttributeName	varchar(50)	MultyValuedAttributeName	Not null
Entity_id	Int	Entity_id	Foreign key

Table Number : 6

Table Name : Table Relation

Primary Key : Relation_id

Field Name	Data type	Description	Constraints
Relation_id	Int	Relation_id	Primary key
RelationName	varchar(50)	RelationName	Not null
Entity_id	Int	Entity_id	Foreign key

Table Number : 7

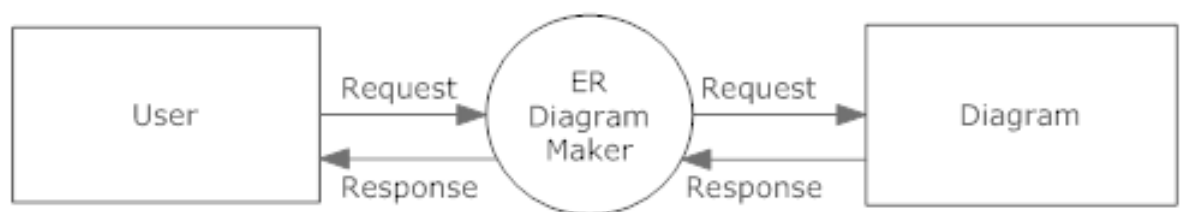
Table Name : Table Registration

Primary Key : User_id

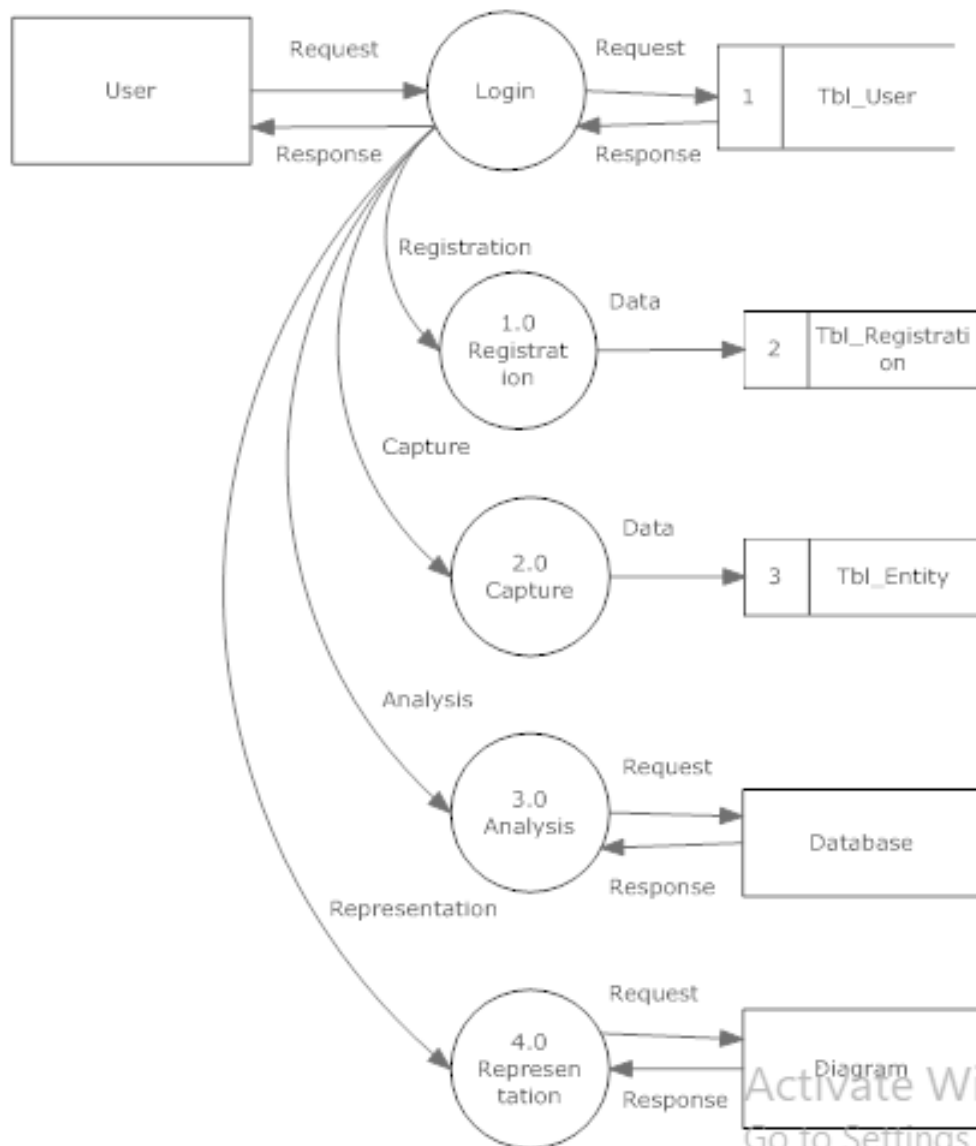
Field Name	Data type	Description	Constraints
User_id	Int	User_id	Primary key
Name	varchar(50)	Name	Not null
Address	Varchar(250)	Address	Not null
Email	Varchar(250)	Email	Not null
ContactNo	Varchar(15)	Contact Number	Not null
BankName	Varchar(250)	Bank Name	Not null
AccountNumber	Varchar(15)	Accout Number	Not null
AdarNumber	Varchar(15)	Adar Number	Not null
User Name	Varchar(15)	User Name	Not null
Password	Varchar(15)	Password	Not null

8.9 Data Flow Diagram

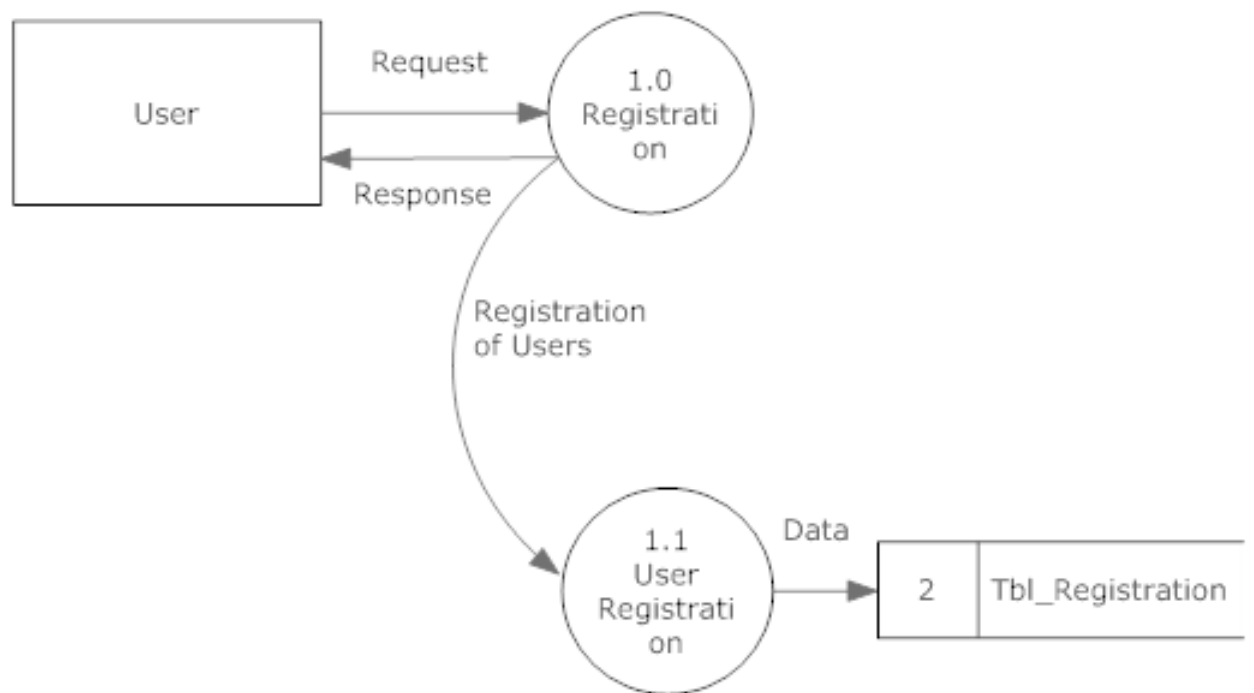
CONTEXT LEVEL (Level-0)



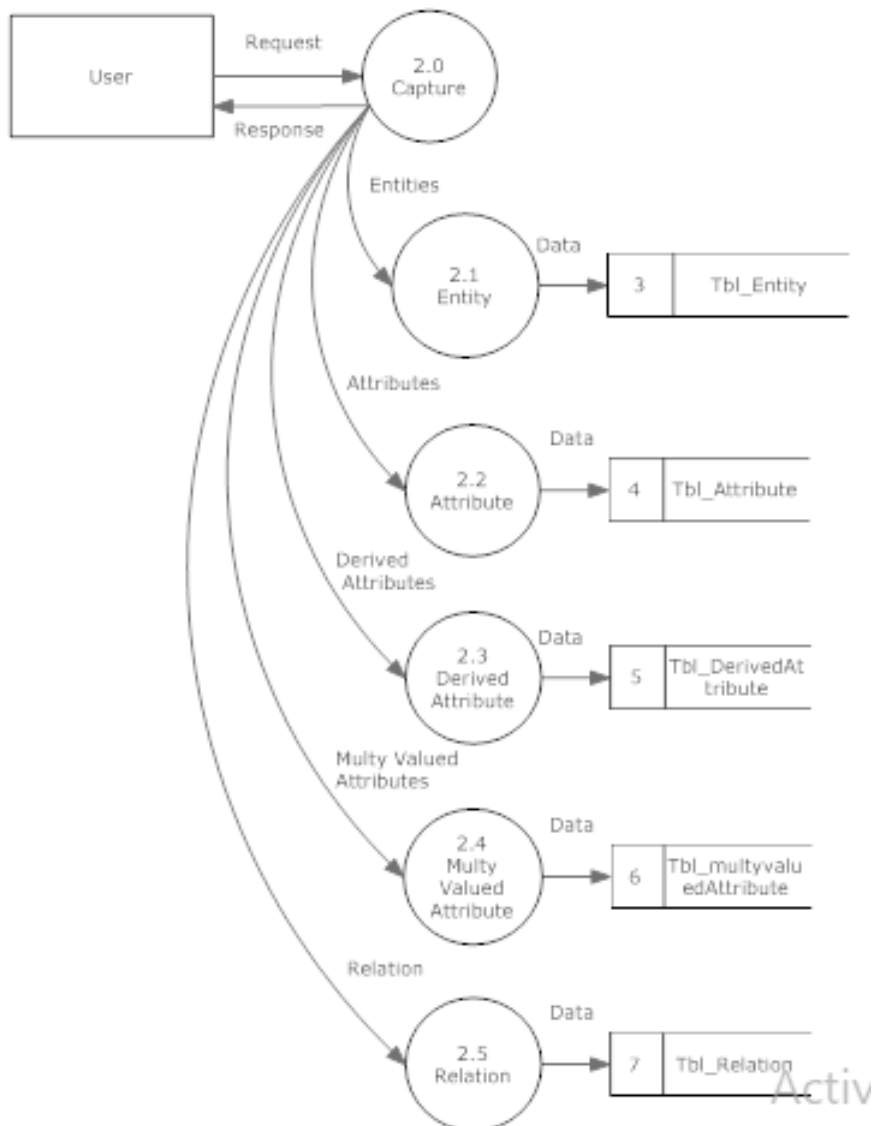
LEVEL-1



LEVEL-2 (REGISTRATION)

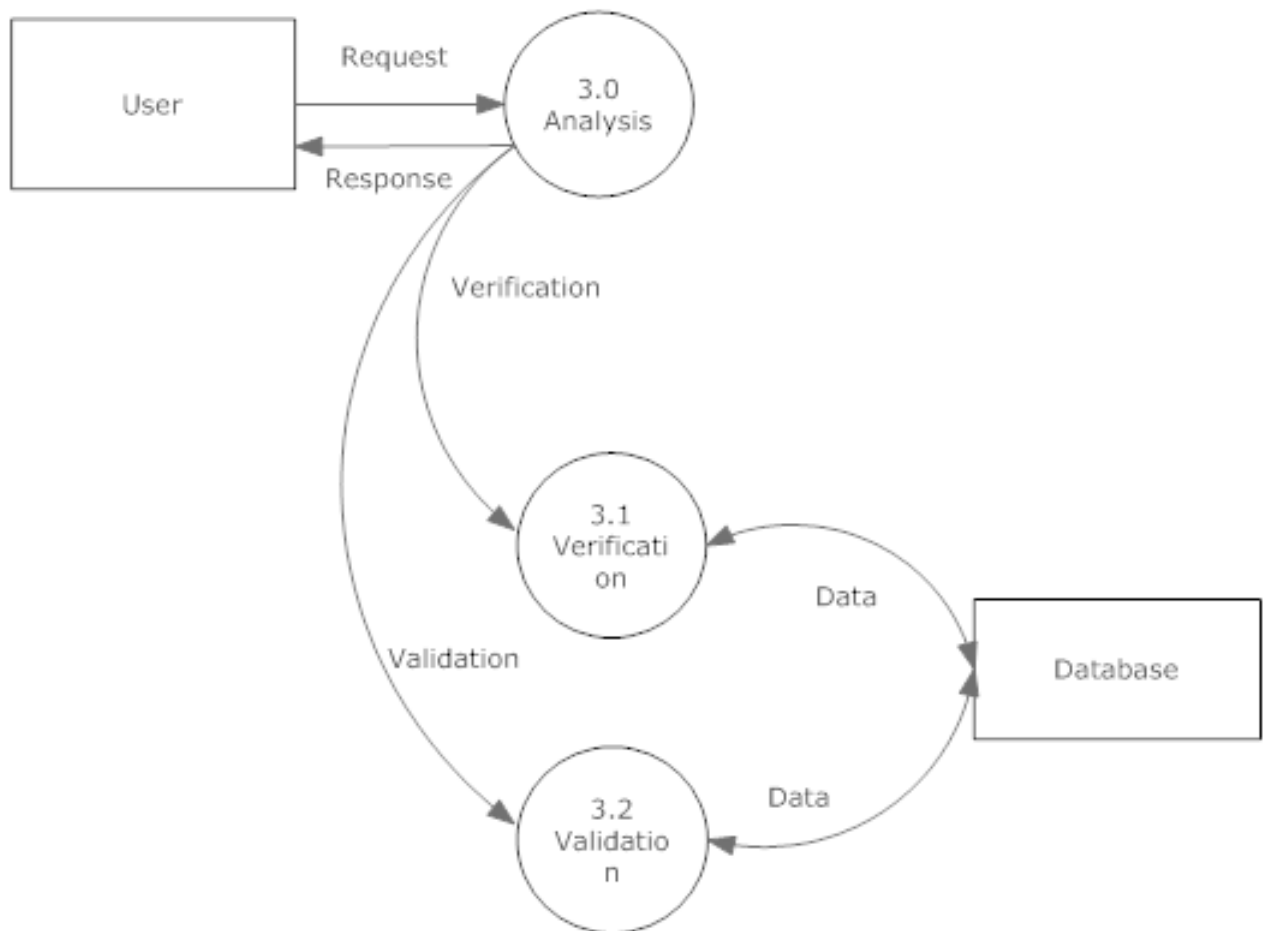


LEVEL-2 (CAPTURE)

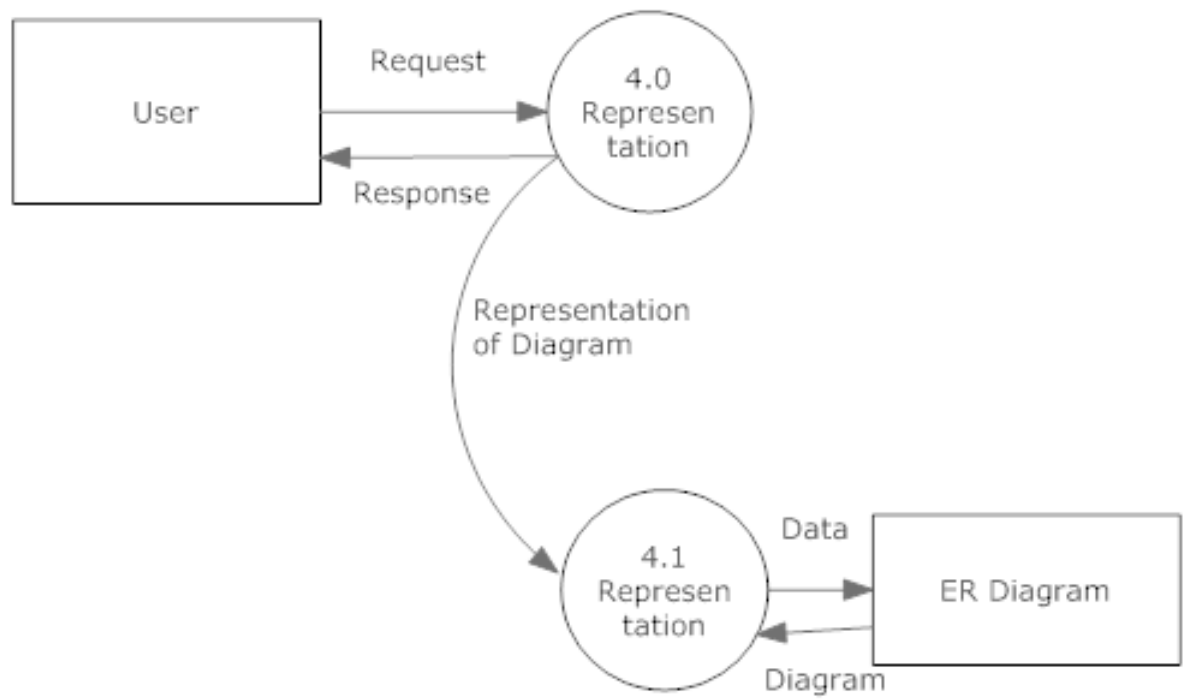


Activ
Go to :

Level-2 (ANALYSIS)



LEVEL-2(REPRESENTATION)



9 SYSTEM TESTING

9.1 Introduction to System Testing

This section discussed about the business, technical or resources related constraint that may keep us from performing all tests necessary. Time schedule is major constraint when talk about testing at the site. Testing of software is to ensure the security. Software testing is the process used to measure the quality of developed computer software. Usually, quality is constrained to certain topics such as correctness, completeness, security, but can also include more technical requirements as described under the ISO standard, such as capability, reliability, efficiency, portability, maintainability, compatibility and usability.

System Testing is a critical aspect of Software Quality Assurance and represents the ultimate review specification, design and coding. Testing is a process of executing a program with the intent of finding an error. A good test is one that has a probability of finding a yet undiscovered error. The purpose of testing is to identify and correct bugs in the developed system. Nothing is complete without testing. Testing is vital to the success of the system .

9.2 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module, this is also known as module testing. These modules are tested separately. These testing are carried out during programming stage itself. The programmer itself did the testing. We had test unit of the software separately and found that every unit was working perfectly. Several validations were done for making the system from accepting the wrong data.

9.3 Integration Testing

Integration testing is the systematic technique for constructing tests to uncover errors within the interface. In this testing all the modules are combined and the entire program is tested as a whole. The programmer and the quality assurance group of the company did this testing together. The volume testing and stress testing were conducted and system passed and testing successfully. The integrated software is completed testing to see that every unit in the software work together without any problem.

9.4 Validation Testing

Validation testing is where requirements established as a part of software requirements analysis is validated against the software that has been constructed. This test provides the final assurance that the software meets all functional, behavioural and performance requirements.

9.5 User Acceptance Testing

User acceptance of the system is the key factor for the success of any system. The system under considerations is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making the hardware management system software.

9.6 Alpha Testing

Alpha testing is one of the most common software testing strategies used in software development. It is specially used by product development organizations. 1. This test takes place at the developers site. Developers observe the users and note problems. 2. Alpha testing is testing of an application when development is about to complete. Minor design changes can still be made as a result of alpha testing. 3. Alpha testing is typically performed by a group that is independent of the design team, but still within the company, e.g. in-house software test engineers, or software QA engineers. 4. Alpha testing is final testing before the software is released to the general public. It has two phases: 5. In the first phase of alpha testing, the software is tested by in-house developers. They use either debugger software, or hardware-assisted debuggers. The goal is to catch bugs quickly. 6. In the second phase of alpha testing, the software is handed over to the software QA staff, for additional testing in an environment that is similar to the intended use. 7. Alpha testing is simulated or actual operational testing by potential users/customers or an independent test team at the developers site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing.

9.7 Beta Testing

Beta testing comes after alpha testing and to be considered a form of external user of acceptance testing versions of the software known as beta version are released to a limited audience outside the programming team. The software released to groups of people so that further testing can ensure the products has few faults or bugs. Sometimes, beta versions are made available to open public to increase the feedback fields to a maximal number of future users.

9.8 Test Cases

A Test Case is a script, program, or other mechanism that exercises a software component to ascertain that a specific correctness assertion is true. In general, it creates a specified initial state, invokes the tested component in a specified way, observes its behavior, and checks to ensure that the behavior was correct.

9.9 Bugzilla

Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database. Bugzilla is a Defect tracking tool, however it can be used as a test management tool as such it can be easily linked with other Test Case management tools like Quality Center, Testlink etc. This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the data-management chain. Key features of Bugzilla includes

1. Advanced search capabilities
2. E-mail Notifications
3. Modify/file Bugs by e-mail
4. Time tracking
5. Strong security
6. Customization
7. Localization

10 SYSTEM IMPLEMENTATION

10.1 Introduction to System Implementation

Implementation is a stage in the project where the theoretical design is turned into working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation, of the current system and its constraints on implementation, design of methods to achieve change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more involved will be the system analysis and design effort required just for implementation.

Implementation is the final and important phase. The system can be implemented only after through testing is done and it is found to working according to the specification. This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

10.2 Training

An analysis of user training focuses on two factors:

1. User capabilities
2. Nature of the system to be installed.

Users range from the native to the highly sophisticated. They approach it as concrete learners, learning how to use the system without trying to understand which abstract principles determine which function. The distinction between concrete and formal (student type) learning says about what one can expect from trainees in general. These project also sophisticated the user capabilities and the corresponding nature of the system to be installed.

10.3 Conversion

Conversion refers to changing from one design to another system. The main objective of conversion is to put tested system into operation while holding costs, risks, and personal irritation to minimum. The various tasks involved in conversion are:

1. Creating computer compatible files.
2. Training the operating staffs.
3. Installing terminals and hardware.

10.4 Post Implementation Review

Every system requires periodic evaluation after implementation. A post implementation review measures the systems performance against predefined requirements. Unlike system testing, which determines where the system fails so that the necessary adjustments can be made, a post-implementation review determines how well the system continues to meet performances specifications. It is done after design and conversion are complete. It also provides information to determine whether major redesign is necessary.

10.5 System Maintenance

Software maintenance is the modification of a software product after delivery to correct faults, to improve performance or other attributes. This section describes the six software maintenance processes as:

1. The implementation processes contains software preparation and transition activities, such as the conception and creation of the maintenance plan, the preparation for handling problems identified during development, and the follow-up on product configuration management.
2. The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance group. The maintenance programmer must analyze each request, confirm it (by reproducing the situation) and check its validity, investigate it and propose a solution, document the request and the solution proposal, and, finally, obtain all the required authorizations to apply the modifications.
3. The process considering the implementation of the modification itself.
4. The process acceptance of the modification, by confirming the modified work with the individual who submitted the request in order to make sure the modification provided a solution.
5. The migration process is exceptional, and is not part of daily maintenance tasks. If the software must be ported to another platform without any change in functionality, this process will be used and a maintenance project team is likely to be assigned to this task.
6. Finally, the last maintenance process, also an event which does not occur on a daily basis, is the retirement of a piece of software.

11 SYSTEM EVALUATION

Although system evaluation is an ongoing process throughout the performance testing effort, it offers greater value when conducted early in the test project. The intent of system evaluation is to collect information about the project as a whole, the functions of the system, the expected user activities, the system architecture, and any other details that are helpful in guiding performance testing to achieve the specific needs of the project.

1. Your need to evaluate and select software that meets your business requirements.
2. Your need to evaluate and select a partner that is capable of delivering the most benefit to your business from your software investment, as well as managing the risks inherent in system implementation projects.
3. Your time and ours is valuable; at each step along the way we will each decide whether or not it is beneficial to proceed.

To help you with your selection, this evaluation process is designed to give us both a clear understanding of the systems to be implemented and the corresponding benefits of the partnership. This information provides a foundation for collecting the performance goals and requirements, characterizing the workload, creating performance-testing strategies and plans, and assessing project and system risks. A thorough understanding of the system under test is critical to a successful performance-testing effort. The measurements gathered during later stages are only as accurate as the models that are developed and validated in this stage. The evaluation provides a foundation for determining acceptable performance; specifying performance requirements of the software, system, or component(s); and identifying any risks to the effort before testing even begins. System evaluation providing in these project is needed to evaluate and select the requirements and managing the risk in system implementation on project. Also it is valuable in time so that way it is beneficial in each step.

12 CONCLUSION

The project entitled Automated ER Diagram Maker was tested with proper date. The system is more helpful and has advantages over the existing system. The entire system is menu assisted and highly interactive. In this system, neat formatted reports can be printed within a short period of time. The system is very user friendly and reports are screen oriented. Accurate updating, data validation and integrity are observed in the system. The system was developed to overcome the difficulties encountered in presently used system. The development of this project underwent the various states of project developments like System analysis, System design, System testing and System implementation. After considering the various feasible solutions, the most feasible one was selected for designing taking into consideration the time and efficiency constraints.

All the effects have put to make sure that the system can manage the details efficiently. The results are obtained in a timely and constrained manner since each process is implemented using single module. The system is reliable to further modification. It also provides easy maintenance adequate security has also provided to ensure that only authorized persons may use this system.

13 SCOPE FOR FUTURE ENHANCEMENTS

The project entitled Automated ER Diagram Maker was successfully designed and developed and tested. The system is developed and designed such a way that further expansion and modification can be made to permit the evaluation. The focus of the system is to inherit the requirements and update the system as per the needs. This project is so perfectly designed that it satisfies the requirements. Nothing can be ended in a single step. It is the fact that nothing is permanent in this world. So this project also has some future enhancements in the evergreen and booming IT industry. Change is inevitable. The system and the architecture is a compatible one, so addition of new module can be done without much difficulty. Since this module has its unique properties it can extend further to make this system a complete one.

14 APPENDIX

14.1 APPENDIX A

14.1.1 Sample Source Code / Pseudo Code

- LoginPage

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;
using System.Configuration;

public partial class Login_new : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Session["E1"] = "";
            Session["M1"] = "";
            Session["R1"] = "";
            Session["A1"] = "";
            //Session["A2"] = "";
            //Session["B2"] = "";
            Session["D1"] = "";
        }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        SqlConnection con = new SqlConnection(ConfigurationManager.
```



```

ConnectionStrings["ConnectionString"].ConnectionString );
con.Open();

SqlCommandcmd = newSqlCommand("select * from
Tbl_Login where UserName='" + TextBox1.Text +"' and
Password='" + TextBox2.Text +"'",con);
SqlDataAdaptersda = newSqlDataAdapter(cmd);
DataTabledt = newDataTable();
sda.Fill(dt);
con.Close();

if (dt.Rows.Count == 0)
{
Response.Write("<script>alert('Invalid UserName or Password');
</script>");
}
else

{
Session["uid"] = dt.Rows[0][0].ToString();
Response.Redirect("Entity_new.aspx");
}
}
}

```

- Entity Page

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.Data;

public partial class Entity_new : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {

        int id;
        id = Convert.ToInt16(Session["uid"].ToString());
        SqlConnection con = new SqlConnection(ConfigurationManager.
            ConnectionStrings["ConnectionString"].ConnectionString);
        con.Open();
        SqlCommand cmd = new SqlCommand("insert into Tbl_Entity
            values('" + TextBox1.Text + "', " + id + ")", con);
        cmd.ExecuteNonQuery();
        SqlCommand cmd1 = new SqlCommand("select max(Entity_id)
            from Tbl_Entity", con);
        SqlDataAdapter sda = new SqlDataAdapter(cmd1);

        DataTable dt = new DataTable();
        sda.Fill(dt);
        Session["Entity_id"] = dt.Rows[0][0].ToString();
        Session["E1"] = Session["E1"].ToString() + "," + TextBox1.Text;

        con.Close();

        TextBox1.Text = "";
    }
}

```

```

}
protectedvoid Button3_Click(object sender, EventArgs e)
{
Response.Redirect("Attribute_new.aspx");

}
protectedvoid Button2_Click(object sender, EventArgs e)
{
Response.Redirect("FinalER.aspx");
}
}

```

- Relationship Page

```

using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Web;
usingSystem.Web.UI;
usingSystem.Web.UI.WebControls;
usingSystem.Data.SqlClient;
usingSystem.Configuration;
usingSystem.Data;

publicpartialclassRelationship_new : System.Web.UI.Page
{
protectedvoidPage_Load(object sender, EventArgs e)
{
if (!IsPostBack)

filldrop();
fillDrops();
}
publicvoidfilldrop()
{
SqlConnection con = newSqlConnection(ConfigurationManager.
ConnectionStrings["ConnectionString"].ConnectionString);
con.Open();
SqlCommandcmd = newSqlCommand("select *from Tbl_Entity", con);

```

```

SqlDataAdaptersda = newSqlDataAdapter(cmd);
DataTabledt = newDataTable();
sda.Fill(dt);
DropDownList1.DataSource = dt;
DropDownList1.DataTextField = "EntityName";
DropDownList1.DataValueField = "Entity_id";
DropDownList1.DataBind();

DropDownList1.Items.Insert(0, "Select Entity1");

}
publicvoidfillDrops()
{
SqlConnection con = newSqlConnection(ConfigurationManager.
ConnectionStrings["ConnectionString"].ConnectionString);
con.Open();
SqlCommandcmd = newSqlCommand("select *from Tbl_Entity", con);
SqlDataAdaptersda = newSqlDataAdapter(cmd);
DataTabledt = newDataTable();
sda.Fill(dt);
DropDownList2.DataSource = dt;
DropDownList2.DataTextField = "EntityName";
DropDownList2.DataValueField = "Entity_id";
DropDownList2.DataBind();

DropDownList2.Items.Insert(0, "Select Entity2");

}
protectedvoid Button3_Click(object sender, EventArgs e)
{
Response.Redirect("Entity_new.aspx");
}

protectedvoid Button1_Click(object sender, EventArgs e)
{
intEid;
Eid = Convert.ToInt16(Session["Entity_id"].ToString());
// id = Convert.ToInt16(Session["Entity_id"].ToString());

```

```

SqlConnection con = new SqlConnection(ConfigurationManager.
ConnectionString["ConnectionString"].ConnectionString);
con.Open();
SqlCommandcmd = new SqlCommand("insert into Tbl_Relation
values('" + TextBox1.Text + "'," + Eid + ")", con);
cmd.ExecuteNonQuery();
con.Close();
TextBox1.Text = "";
}
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{

}
protected void DropDownList2_SelectedIndexChanged(object sender, EventArgs e)
{

}

protected void Button2_Click(object sender, EventArgs e)
{
Response.Redirect("FinalER.aspx");
}
}

```

- Design of diagram

```
<%@PageLanguage="C#"AutoEventWireup="true"CodeFile=
"FinalER4.aspx.cs"Inherits="FinalER4"%>

<!DOCTYPEhtmlPUBLIC"-//W3C//DTD XHTML 1.0 Transitional//
EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<htmlxmlns="http://www.w3.org/1999/xhtml">
<headrunat="server">
<title></title>
<styletype="text/css">

.style1
    {
width: 97px;
    }
.style3
    {
width: 62px;
    }
.style2
    {
width: 100px;
    }
.style4
    {
width: 27px;
    }
.style5
    {
width: 70px;
    }

.style6
    {
width: 63px;
    }
.style7
    {
```

```

width: 87px;
    }
</style>
<script>
functionfunPrint()
    {
window.print();
    }

</script>
</head>
<body>
<formid="form2"runat="server">
<div>

<tablestyle="width:100%;"width="100">
<tr>
<td>
<div>

<tablestyle="width: 100%;">
<tr>
<tdclass="style1">
&nbsp;</td>
<tdclass="style3">
&nbsp;</td>
<tdclass="style2">
&nbsp;</td>
<tdclass="style4">
&nbsp;</td>
<tdclass="style5">
<asp:LabelID="Label16"runat="server"Text="Label"Width="90px">
</asp:Label>
</td>
<tdclass="style6">
&nbsp;</td>
<td>
&nbsp;</td>
<tdclass="style7">
&nbsp;</td>
<td>

```

```

&nbsp;</td>
</tr>
<tr>
<tdclass="style1">
&nbsp;</td>
<tdclass="style3">
&nbsp;</td>
<tdclass="style2">
&nbsp;</td>
<tdclass="style4">
&nbsp;</td>

<tdclass="style5">
<imgalt=""height="50"src="imgs/multivalue%20attribute.png"width="90"/>
</td>

<tdclass="style6">
&nbsp;</td>
<td>
&nbsp;</td>
<tdclass="style7">
&nbsp;</td>
<td>
&nbsp;</td>
</tr>
<tr>
<tdclass="style1">
&nbsp;</td>
<tdclass="style3">
&nbsp;</td>
<tdclass="style2">
<asp:LabelID="Label14"runat="server"Text="Label"Width="90px">
</asp:Label>
</td>
<tdclass="style4">
&nbsp;</td>
<tdclass="style5">
<imgalt=""height="40"src="imgs/vertical.jpg"width="90"/></td>
<tdclass="style6">
&nbsp;</td>
<tdwidth="90">

```



```

<asp:LabelID="Label15"runat="server"Text="Label"Width="90px">
</asp:Label>
</td>
<tdclass="style7">
&nbsp;</td>
<td>
&nbsp;</td>
</tr>
<tr>
<tdclass="style1">
&nbsp;</td>
<tdclass="style3">
&nbsp;</td>
<tdclass="style2">
<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
<tdclass="style4">
<imgalt=""height="50"
src="imgs/horizontal%20line.png"style="margin-top: 0px"width="60"/>
</td>
<tdclass="style5">
<asp:TextBoxID="TextBox3"runat="server"BorderStyle="Solid"Height="50px"
Width="90px"ontextchanged="TextBox1_TextChanged"></asp:TextBox>
</td>
<tdclass="style6">
<imgalt=""height="40"
src="imgs/horizontal%20line.png"style="margin-top: 0px"width="60"/>
</td>
<td>
<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
<tdclass="style7">
&nbsp;</td>
<td>
&nbsp;</td>
</tr>
<tr>
<tdclass="style1">
<asp:LabelID="Label8"runat="server"Text="Label"Width="90px">
</asp:Label>
</td>

```

```

<tdclass="style3">
&nbsp;</td>
<tdclass="style2">
<asp:LabelID="Label1"runat="server"Text="Label"Width="90px">
</asp:Label>
</td>
<tdclass="style4">
&nbsp;</td>
<tdclass="style5">
<imgalt=""height="40"src="imgs/vertical.jpg"width="90"/></td>
<tdclass="style6">
&nbsp;</td>
<td>
<asp:LabelID="Label4"runat="server"Text="Label"Width="90px">
</asp:Label>
</td>
<tdclass="style7">
&nbsp;</td>
<td>
<asp:LabelID="Label13"runat="server"Text="Label"Width="90px"Height="40px">
</asp:Label>
</td>
</tr>
<tr>
<tdclass="style1">
<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
<tdclass="style3">
&nbsp;</td>
<tdclass="style2">
<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
<tdclass="style4">

&nbsp;</td>
<tdclass="style5">
<imgalt=""src="imgs/diagonal.png"
style="width: 91px; height: 50px"/></td>
<tdclass="style6">
&nbsp;</td>
<td>
<imgalt=""height="50"src="imgs/multivalue%20attribute.png"width="90"/>
</td>

```

```

<tdclass="style7">
&nbsp;</td>
<td>
<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
</tr>
<tr>
<tdclass="style1">
&nbsp;</td>
<tdclass="style3">
<imgalt=""height="40"
src="imgs/2.png"style="margin-top: 0px"width="60"/></td>
<tdclass="style2">
<imgalt=""height="40"src="imgs/vertical.jpg"width="90"/></td>
<tdclass="style4">
<imgalt=""height="50"src="imgs/1.png"width="60"/></td>
<tdclass="style5">

<asp:LabelID="Label17"runat="server"Text="Label"Width="90px"Height="40px">
</asp:Label>
</td>
<tdclass="style6">
&nbsp;</td>
<td>
<imgalt=""height="50"src="imgs/vertical.jpg"width="90"/></td>
<tdclass="style7">
<imgalt=""height="50"src="imgs/1.png"width="60"/></td>
<td>
&nbsp;</td>
</tr>
<tr>
<tdclass="style1">
<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
<tdclass="style3">
<imgalt=""height="50"
src="imgs/horizontal%20line.png"style="margin-top: 0px"width="60"/>
</td>
<tdclass="style2">
<asp:TextBoxID="TextBox1"runat="server"BorderStyle="Solid"Height="50px"
Width="90px"ontextchanged="TextBox1_TextChanged">
</asp:TextBox>

```

```

</td>

<tdclass="style4">
<imgalt=""height="50"
src="imgs/horizontal%20line.png"style="margin-top: 0px"width="60"/>
</td>
<tdclass="style5">
<imgalt=""src="imgs/diagonal.png"
style="width: 94px; height: 45px"height="50"width="90"/></td>
<tdclass="style6">
<imgalt=""height="40"
src="imgs/horizontal%20line.png"style="margin-top: 0px"width="60"/></td>
<td>
<asp:TextBoxID="TextBox2"runat="server"BorderStyle="Solid"Height="50px"
Width="90px"ontextchanged="TextBox1_TextChanged"></asp:TextBox>
</td>
<tdclass="style7">
<imgalt=""height="70"
src="imgs/horizontal%20line.png"style="margin-top: 0px"width="60"/></td>
<td>
<imgalt=""height="60"src="imgs/ellipse.jpg"width="90"/></td>
</tr>
<tr>
<tdclass="style1">
<asp:LabelID="Label2"runat="server"Text="Label"Width="90px"Height="50px">
</asp:Label>
</td>
<tdclass="style3">
<imgalt=""height="50"src="imgs/1.png"width="60"/></td>
<tdclass="style2">
<imgalt=""height="40"src="imgs/vertical.jpg"width="90"/></td>
<tdclass="style4">
&nbsp;</td>
<tdclass="style5">
<asp:LabelID="Label6"runat="server"Text="Label"Width="90px"Height="50px">
</asp:Label>

</td>
<tdclass="style6">
<imgalt=""height="40"src="imgs/1.png"width="60"/></td>
<td>

```

```

<imgalt=""height="40"src="imgs/vertical.jpg"width="90"/></td>
<tdclass="style7">
<imgalt=""height="50"
src="imgs/2.png"style="margin-top: 0px"width="60"/></td>
<td>
<asp:LabelID="Label7"runat="server"Text="Label"Width="90px"Height="50px">
</asp:Label>
</td>
</tr>
<tr>
<tdclass="style1">

<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
<tdclass="style3">
&nbsp;</td>
<tdclass="style2">
<imgalt=""height="50"
src="imgs/derived%20attribute.png"width="90"/></td>
<tdclass="style4">
&nbsp;</td>
<tdclass="style5">
<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
<tdclass="style6">
&nbsp;</td>
<td>
<imgalt=""height="50"src="imgs/ellipse.jpg"width="90"/></td>
<tdclass="style7">
&nbsp;</td>
<td>
<imgalt=""height="50"
src="imgs/derived%20attribute.png"width="90"/></td>
</tr>
<tr>
<tdclass="style1">
<asp:LabelID="Label9"runat="server"Text="Label"Width="90px"></asp:Label>
</td>
<tdclass="style3">
&nbsp;</td>
<tdclass="style2">
<asp:LabelID="Label3"runat="server"Text="Label"Width="90px"></asp:Label>
</td>

```

```

<tdclass="style4">
&nbsp;</td>
<tdclass="style5">
<asp:LabelID="Label10"runat="server"Text="Label"Width="90px"></asp:Label>
</td>
<tdclass="style6">
&nbsp;</td>
<td>
<asp:LabelID="Label5"runat="server"Text="Label"Width="90px"></asp:Label>
</td>
<tdclass="style7">
&nbsp;</td>
<td>

<asp:LabelID="Label12"runat="server"Text="Label"Width="90px"></asp:Label>

</td>
</tr>
</table>

</div>
</td>
</tr>
<tr>
<td>
<inputid="Button1"type="button"value="Print"onclick="funprint()"/></td>
</tr>
</table>

</div>
</form>
<%--<form id="form1" runat="server">
<div>

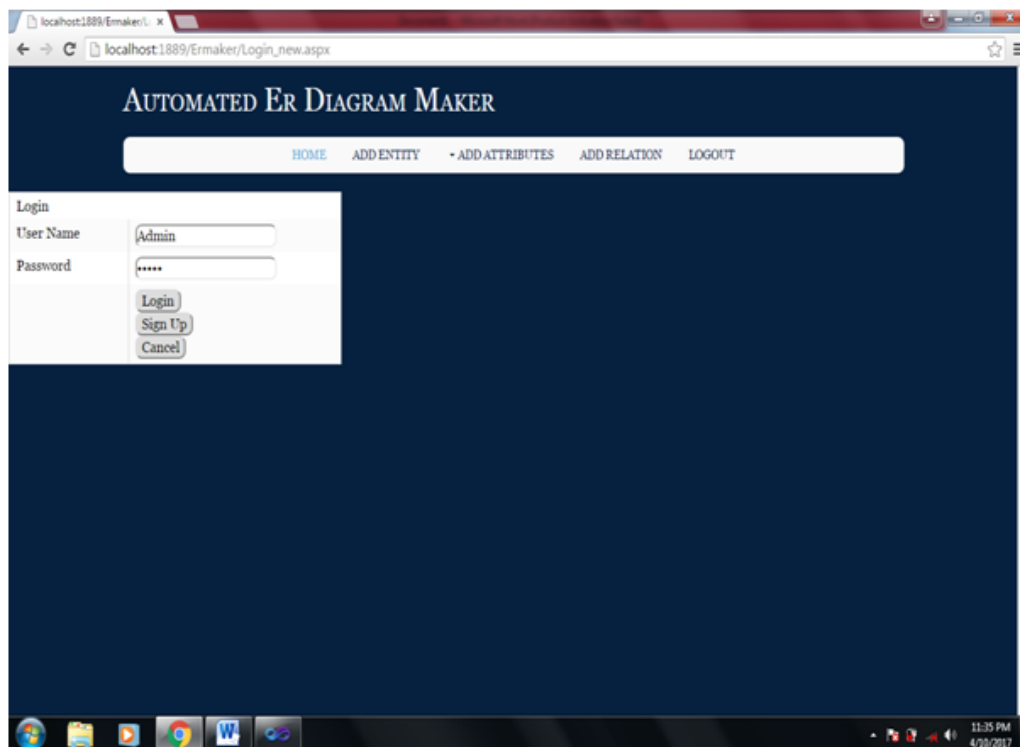
</div>
</form>--%>
</body>
</html>

```

14.2 APPENDIX B

14.2.1 Screen Shots

1. Home Page



2. Registration Page

localhost:1889/Ermaker/ X
localhost:1889/Ermaker/Registration_new.aspx

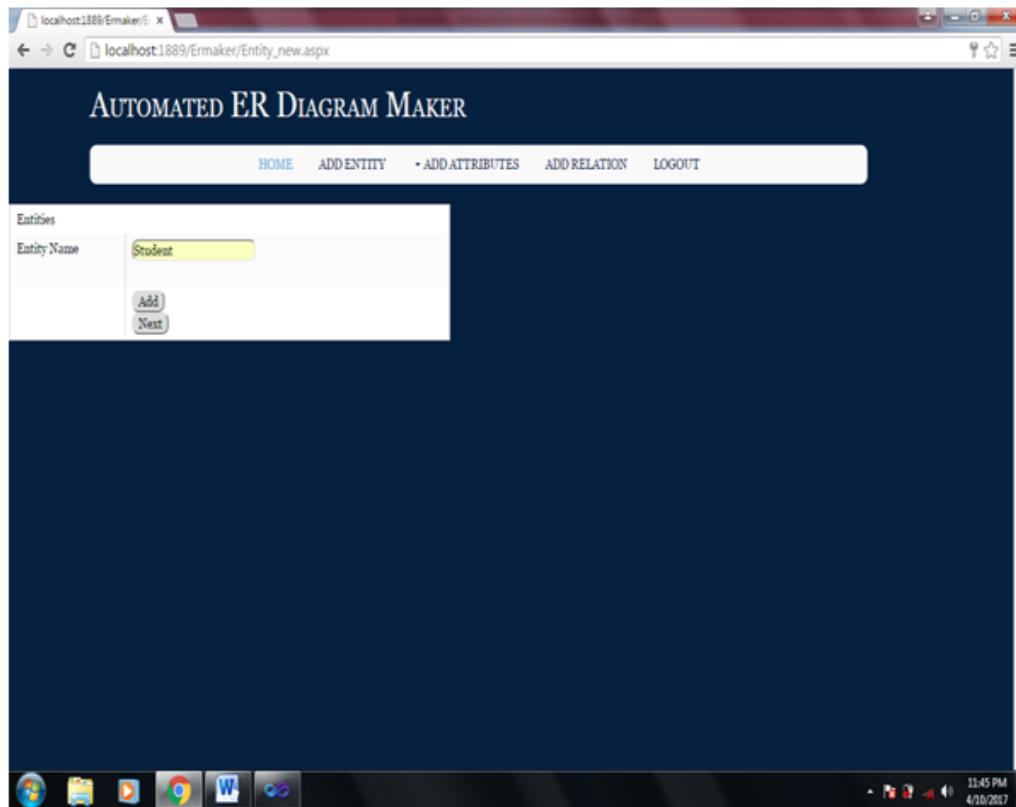
AUTOMATED ER DIAGRAM MAKER

Registration Details

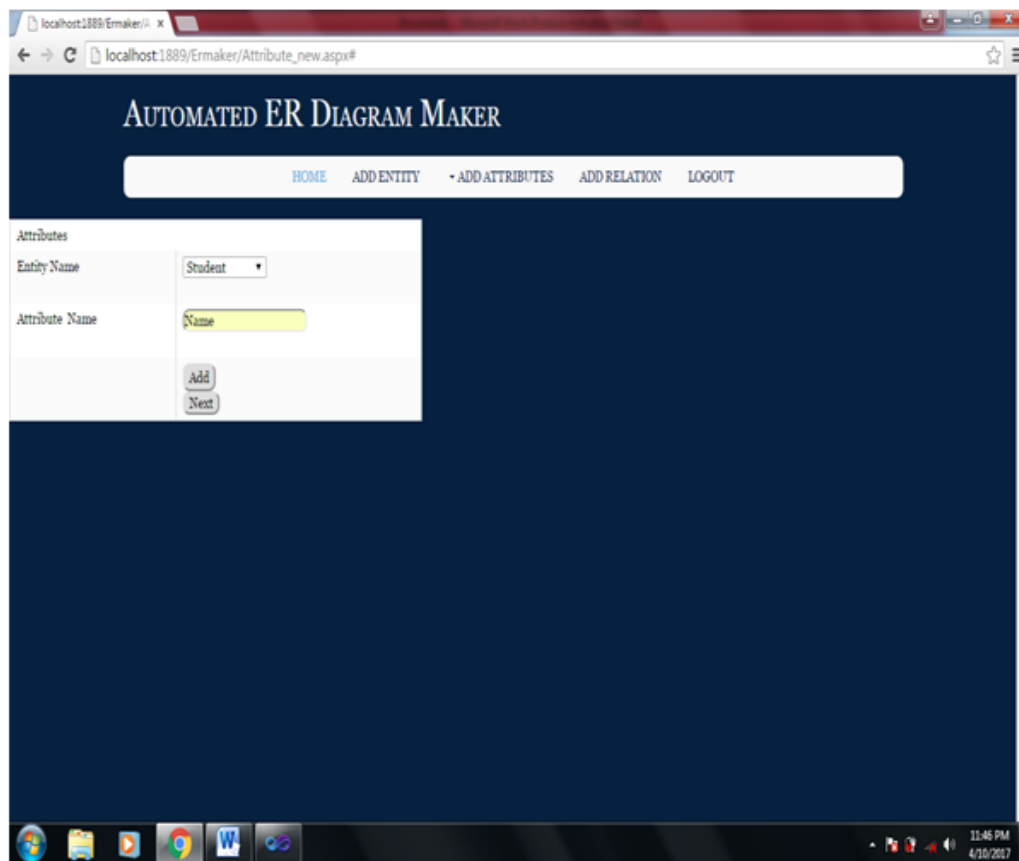
Name	John Sterrior
Address	Palakkuzha(P.O) Palakkuzha
Contact Number	9025331543
Bank Name	SBI
Account Number	645630988
Date of Birth	31-12-1991
Date of Registration	12-12-2017
Adhar Number	345678
Email	johnsterior@gmail.com
User Name	john
Password	john123

11:44 PM
4/10/2017

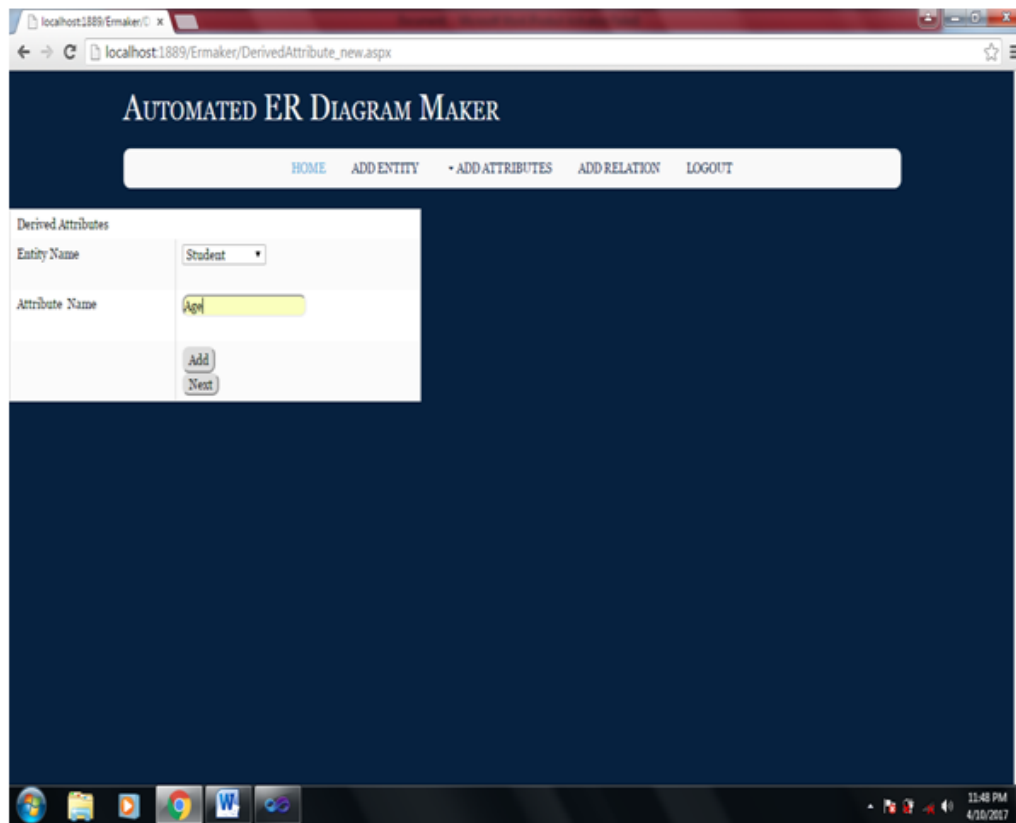
3. Entity Page



4. Attribute Page



5. Derived Attribute Page



6. Multy-Valued Attribute Page

The screenshot shows a web browser window with the address bar displaying 'localhost:1889/Ermaker/MultyvaluedAttribute_new.aspx'. The page title is 'AUTOMATED ER DIAGRAM MAKER'. A navigation bar contains links: HOME, ADD ENTITY, ADD ATTRIBUTES (active), ADD RELATION, and LOGOUT. The main content area is titled 'Multy Valued Attributes' and contains a form with the following fields and controls:

Multy Valued Attributes	
Entity Name	Teacher
Attribute Name	Knowledge
	<input type="button" value="Add"/>
	<input type="button" value="Next"/>

The Windows taskbar at the bottom shows the time as 11:49 PM on 4/10/2017.

7. Relationship Page

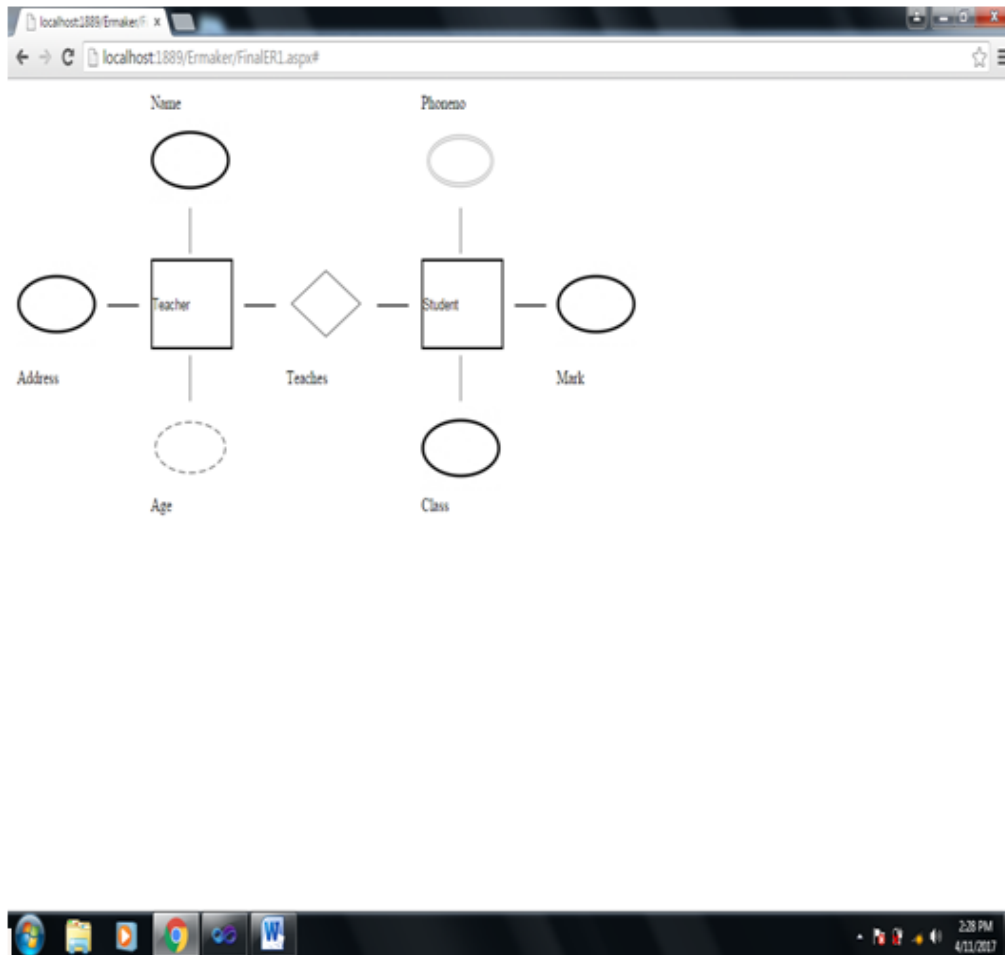
The screenshot displays a web browser window with the URL `localhost:1889/Ermaker/Relationship_new.aspx#`. The application title is "AUTOMATED ER DIAGRAM MAKER". The navigation bar includes links for [HOME](#), [ADD ENTITY](#), [ADD ATTRIBUTES](#), [ADD RELATION](#), and [LOGOUT](#).

The "Relationships" section contains the following form fields:

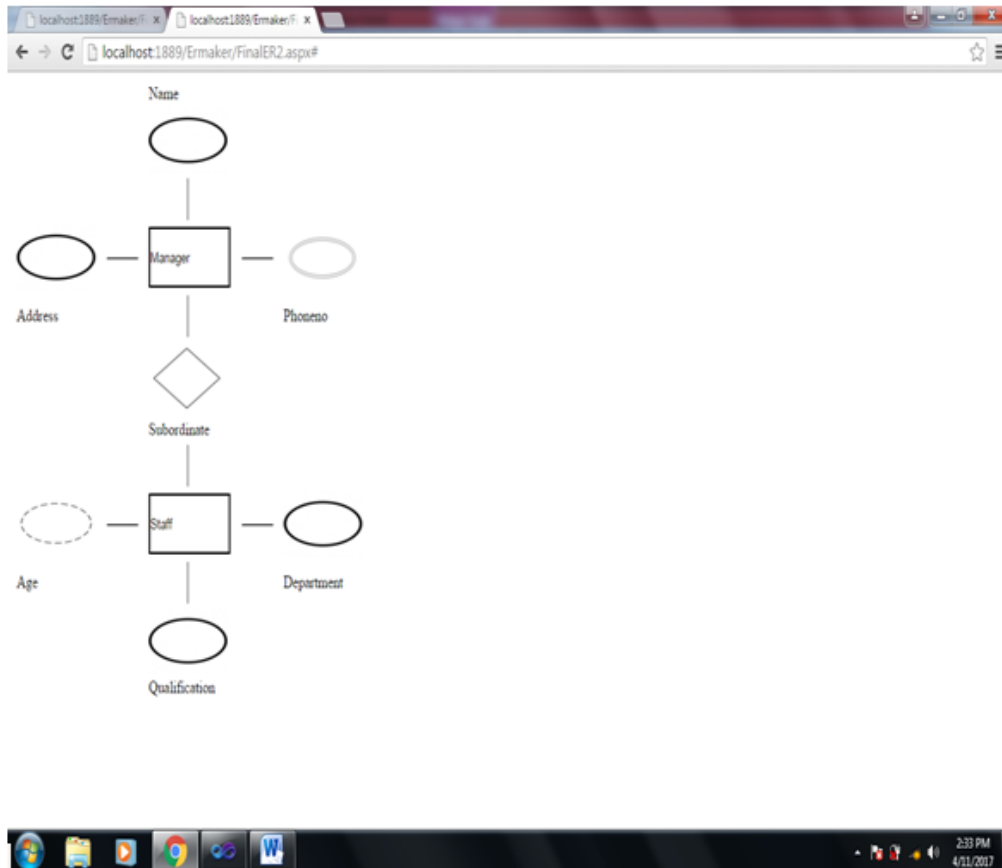
- Entity Name 1:
- Entity Name 2:
- Relationship:

At the bottom of the form, there are two buttons: [Add](#) and [Finish](#).

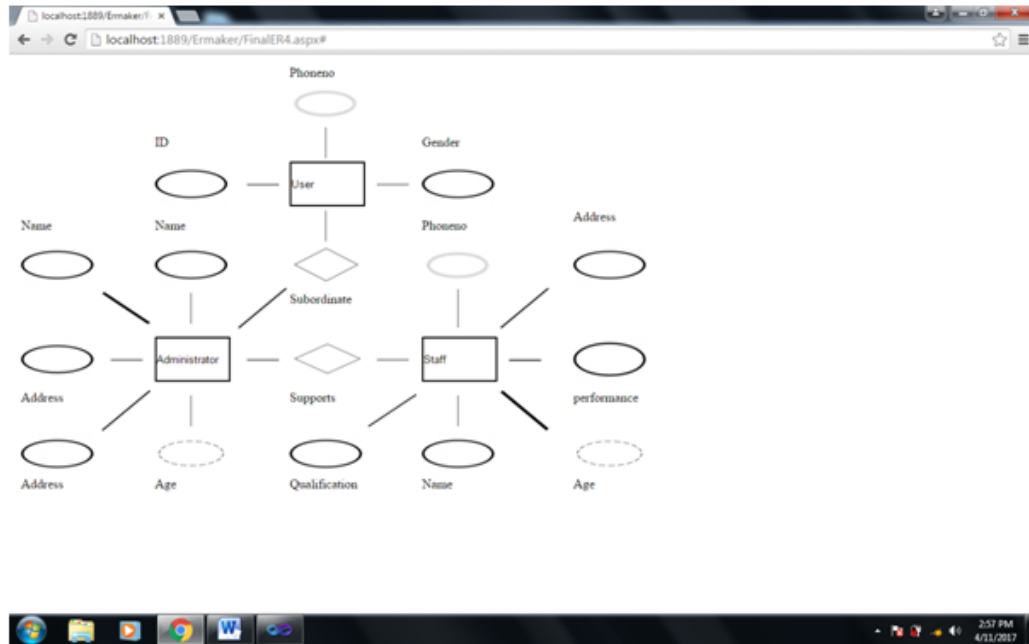
8. An ER Diagram of two entities named Teacher and Student



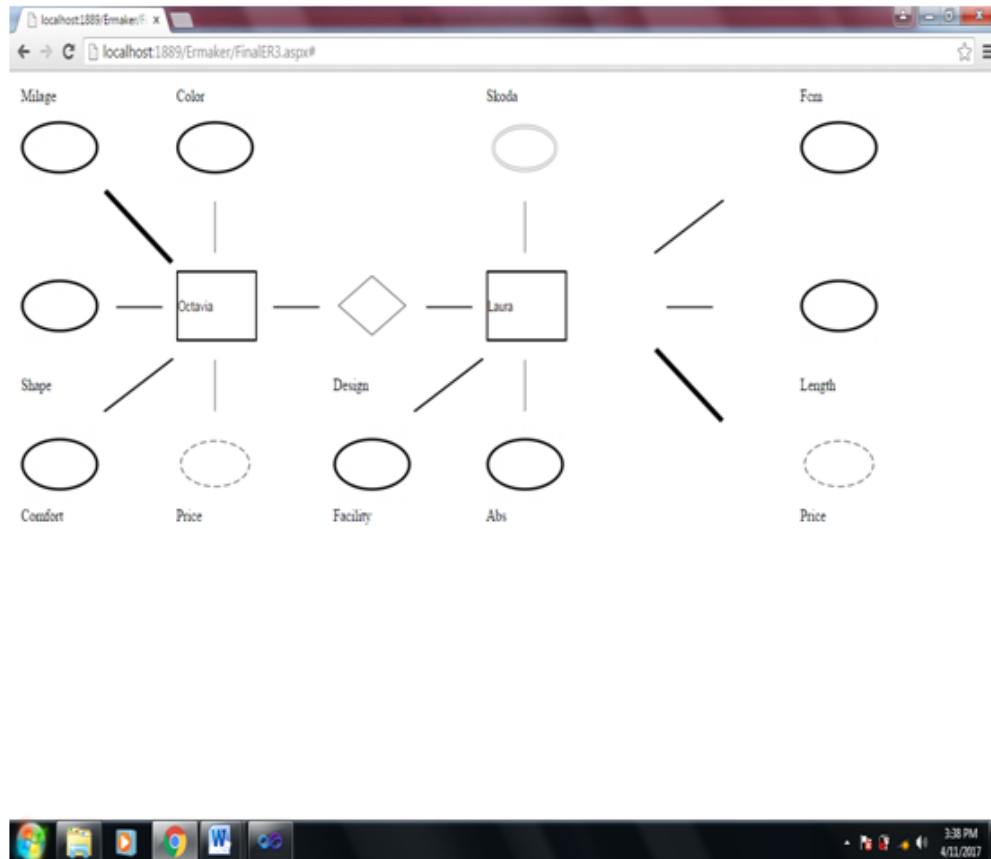
9. An ER Diagram of two entities named Manager and Staff



10. An ER Diagram of two entities named Administrator, User and Staff



11. An ER Diagram of two entities named Octavia and Laura



References

- [1] BUILD YOUR OWN ASP.NET 2.0 WEBSITE USING C# AND VB -
by CristianDarie and Zak Ruvalcaba
- [2] PRO ASP.NET 2.0 WEBSITE PROGRAMMING-by Damon Arm-
strong
- [3] SYSTEM ANALYSIS AND DESIGN- by Elias M Award,Second Edi-
tion, 1999, Galgotia Publications, Delhi
- [4] DATABASE SYSTEM CONCEPTS- by Silberschatz, Korth,Sudarshan,
Fourth Edition
- [5] <http://www.wikipedia.com>
- [6] <http://www.w3schools.com>
- [7] <http://www.asp.net>