

# **ACKNOWLEDGEMENT**

## **ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to the **I.T. Department of DGET'S College** of Arts, Commerce and Science. Many individuals have contributed to the preparation of this project.

I would like to thanks the most valuable personalities for me the project guide and professor in charge **Mrs. Sarmila Rajesh** for project selection and permitting me on the project work and also for the valuable comments and suggestions given by her, without their timely guidance and great help this project could not have been completed.

Hearty thanks to respect**Mrs. Leena Ajish** (Principal of our College) for her kind co-operation in the completion of our project.

I also thank to all who directly and indirectly involve in this project.

Finally, I thank to all my friends and well-wishers for their inspiration and help given to me.

**Sheetal Singh &**

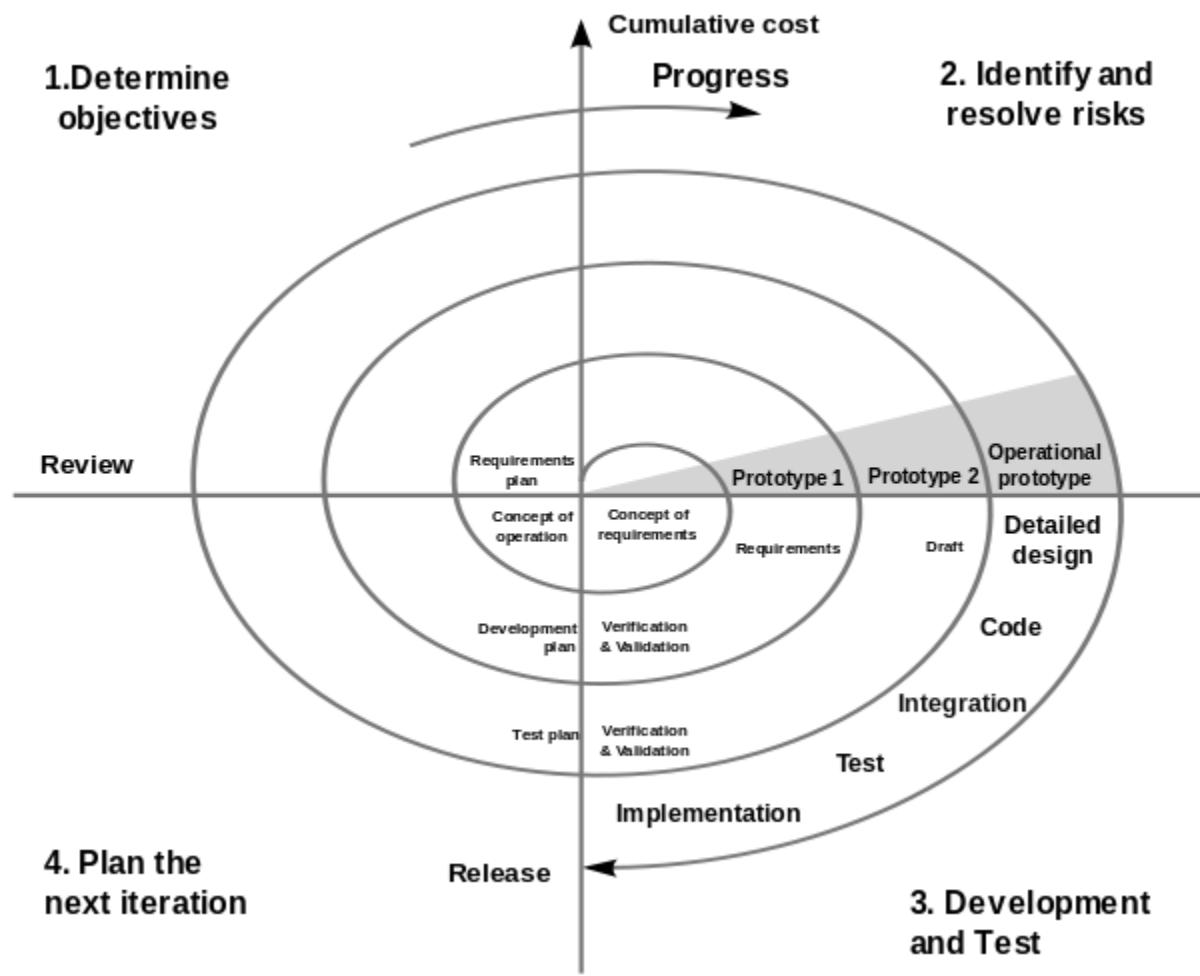
**Vijayashri Gujaran**

**T.Y.B.Sc. IT**

# **PROJECT LIFECYCLE MODEL**

## **SPIRAL MODEL:**

The spiral model is a software development process combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts. Also known as the spiral lifecycle model (or spiral development), it is a systems development method (SDM) used in information technology (IT). This model of development combines the features of the prototyping and the waterfall model. The spiral model is intended for large, expensive and complicated projects.



### **The Model:**

The spiral model combines the idea of iterative development (prototyping) with the systematic, controlled aspects of the waterfall model. It allows for incremental releases of the product, or incremental refinement through each time around the spiral. The spiral model also explicitly

includes risk management within software development. Identifying major risks, both technical and managerial, and determining how to lessen the risk helps keep the software development process under control.

The spiral model is based on continuous refinement of key products for requirements definition and analysis, system and software design, and implementation (the code). At each iteration around the cycle, the products are extensions of an earlier product. This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk assessment, and the building of prototypes and simulations.

The spiral lifecycle model allows for elements of the product to be added in when they become available or known. This assures that there is no conflict with previous requirements and design. This method is consistent with approaches that have multiple software builds and releases and allows for making an orderly transition to a maintenance activity. Another positive aspect is that the spiral model forces early user involvement in the system development effort. For projects with heavy user interfacing, such as user application programs or instrument interface applications, such involvement is helpful.

Starting at the center, each turn around the spiral goes through several task regions:

- Determine the objectives, alternatives, and constraints on the new iteration.
- Evaluate alternatives and identify and resolve risk issues.
- Develop and verify the product for this iteration.
- Plan the next iteration.

### Why we use spiral model?

- It is simple and easy to use.
- Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller project where requirements are very well understood.

# **ORGANIZATION PROFILE**

## **ORGANIZATION PROFILE**

### **What is Matrimonial Web Application?**

The main objective of Matrimonial Web Application is to provide Grooms and Brides with excellent matchmaking experience by exploring the opportunities and resources to meet true potential partner. Keeping our objective in mind, we have created a world renowned online matchmaking services that will touch the souls of millions of people all over the globe.

### **The purposes of the Matrimonial Web Application are:**

- The main purpose of this application is to facilitate matchmaking business by applying the information in the field.
- It helps the user by providing profiles of perspective “Bride” or “Groom” and other information regarding them online.
- User can get information regarding their dream life partner at his/her home at his/her convenience.
- This application also provides a search utility which helps those users who have a certain criteria of qualities in mind to make online matrimonial easier.
- Since internet is a pivot for modern business, our project which is based on internet paves a path for modernization in trade.

Matrimonial Web Application will allow a new user to register and after successfully registration user can get email confirmation, after completing registration users profile will be visible to other users.

Matrimonial website whichis provide platform to a lot of Bride/Groom for finding their perfect match. There are different sectors like Registration, Partner, Search, etc. So the Bride/Groom can get their interest for find their partner. Bride/Groom can directly search

## **Matrimonial Web Application**

Partner according to their required criteria. The Bride/Groom can use match By Email functionality so he/she can get directly E-mail alert for the match which fulfill their required criteria.

### **For This Application, we will provide following capabilities:**

- (1) Admin Module.
- (2) User Registration Module
- (3) All match module
- (4) User Login module
- (5) Chat box module
- (6) Contact us module
- (7) Success Story module
- (8) Feedback module
- (9) User profile module
- (10) Image Uploading module
- (11) Creating album
- (12) Sending Express Interest
- (13) Sending Personal messages
- (14) User Update Module
- (15) Paid Membership
- (16) Search Module.
- (17) Non Paid Member details module

The application will have to be completed using Visual Studio 2008 and SQL Server 2008R2 with help of other office productivity tools.

The application testing criteria and installation requirements will be part of the detailed application architecture document.

## **ABOUT PROJECT**

### **Project profile**

<b>Company Name</b>	:	MASS BUSINESS SOLUTION PVT. LTD.
<b>Website</b>	:	www.massbusinesssolution.com
<b>Project Title</b>	:	Matrimonial Web Application.
<b>Objective Of System</b>	:	Matrimonial Web Application will allow a new user to register and after successfully registration user can get email confirmation, after completing registration users profile will be visible to other users.
<b>Operating Systems</b>	:	Microsoft Windows 7 Home Premium
<b>Hardware Requirement</b>	:	Pentium 90 MHZ or Faster and 96 MB Ram (Client) Pentium 133 MHZ or Faster and 128 Ram (Server)
<b>Software Requirement</b>	:	Microsoft Visual Studio Asp .Net 2008
<b>Front End</b>	:	Microsoft Visual Studio 2008
<b>Back End</b>	:	Microsoft SQL Server 2008 R2
<b>Others</b>	:	<ul style="list-style-type: none"><li>• Microsoft Visio</li><li>• .Net Framework</li></ul>
<b>Guided By</b>	:	Mrs. Sharmila Rajesh

**PURPOSE:**

Matrimonial website which is provide platform to a lot of Bride/Groom for finding perfect match. There are different sectors like Registration, Login, Search, etc. So the Bride/Groom can get their interest for find their partner. Bride/Groom can directly search Partner according to their required criteria. The Bride/Groom can use match By Email functionality so he/she can get directly E-mail alert for the match which fulfill their required criteria.

**The purposes of the Matrimonial Web Application are:**

- The main purpose of this application is to facilitate matchmaking business by applying the information in the field.
- It helps the user by providing profiles of perspective “Bride” and “Groom” and other information regarding them online.
- User can get information regarding their dream life partner at his/her home at his/her convenience.
- This application also provides a search utility which helps those users who have a certain criteria of qualities in mind to make online matrimonial easier.
- Since internet is a pivot for modern business, our project which is based on internet paves a path for modernization in trade.

**TECHNOLOGY AND LITERATURE REVIEW:**

<b>Operating System</b>	<b>Window 7 Home Premium</b>
<b>Technology</b>	<b>Microsoft Visual Studio 2008 with 3.56 architecture</b>
<b>Language</b>	<b>Asp.net</b>
<b>Database</b>	<b>MS-SQL server 2008 R2</b>

# **STAKEHOLDERS**

## **STAKEHOLDERS**

Stakeholders are those persons who are successfully involved in completion of the project at the specified time period. Different stakeholders included in the completion of the project

### **The Client or Administrator:-**

He is the one for whom the project is being made. He is successfully engaged for the completion of the software as soon as possible.

### **The Technical Staff:-**

They are the people who provide guidance in area of programming language to the programmer. The guidance is given to the student or the programmer so that he can complete the given project properly.

### **The Creator or Programmer:-**

The student itself is the programmer who wants the project to be completed in the specified time. He is the main stakeholder for the project.

# **TECHNOLOGY AND**

# **HARDWARE USED**

## **The .net framework:**

A frame work is commonly thought of as a set of class libraries that aid in the development of applications. The .net framework is more than just a set of classes. The .net framework is targeted by compliers using a wide variety of applications. Including everything from small components that run on handheld devices to large Microsoft ASP.ET application that span web farms, where multiple web serves act together to improve the performance fault tolerance of a web site. The .NET framework is responsible for providing a basic platform that these applications can share. This basic platform includes a runtimes set of services that oversee the execution of applications. A key responsibility of the runtime is to manage execution so that software written by different programming languages uses classes and other types safely.

## **Microsoft .net framework architecture:**

Microsoft's .NET Framework is comprised of two main components - the Common Language Runtime (CLR) and the .NET Framework class libraries. The CLR is the real foundation of the .NET Framework. It is the execution engine for all .NET applications. Every target computer requires the CLR to successfully run a .NET application that uses the .NET Framework. The main features of CLR include:

- Automatic Memory Management
- Thread Management
- Code Compilation & Execution
- Code Verification
- High level of security
- Remoting
- Structured Exception Handling
- Interoperability between Managed and Unmanaged code.
- Integration with Microsoft Office System

All .NET applications are compiled into Intermediate Language code (MSIL). When executed on the CLR, MSIL is converted into native machine code specific to the operating platform. This process is done by a Just in Time (JIT) compiler. The code executed by the CLR is called as Managed Code. This code is type safe and thoroughly checked by the CLR before being deployed. The .NET runtime also provides a facility to incorporate existing COM components and DLL's into a .NET application. Code that is not controlled by the CLR is called Unmanaged Code.

The .NET Framework is further comprised of Common Type System (CTS) and Common Language Specification (CLS). The CTS defines the common data types used by .NET programming languages. The CTS tells you how to represent characters and numbers in a program. The CLS represents the guidelines defined by for the .NET Framework. These specifications are normally used by the compiler developers and are available for all languages, which target the .NET Framework.

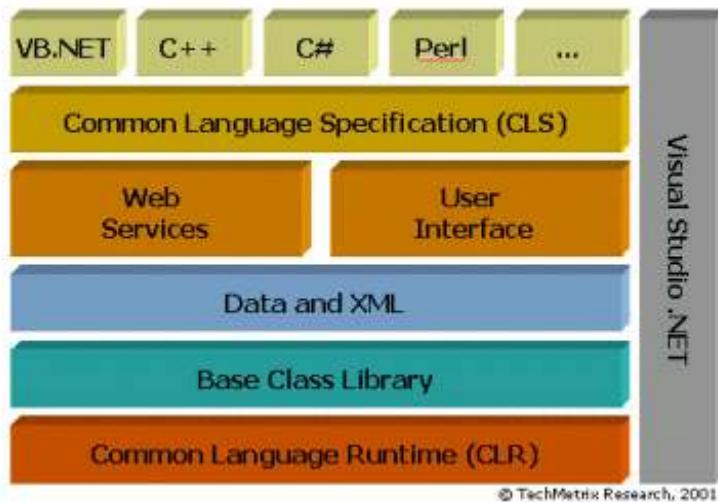


Fig.Net architecture

## **Common Language Runtime:**

The Common Language Runtime (CLR) is the virtual machine component of Microsoft's .NET initiative. It is Microsoft's implementation of the Common Language Infrastructure (CLI) standard, which defines an execution environment for program code. The CLR runs a form of byte code called the Microsoft Intermediate Language (MSIL), Microsoft's implementation of the Common Intermediate Language.

Developers using the CLR write code in a high level language such as C# or VB.Net. At compile-time, a .NET compiler converts such code into MSIL (Microsoft Intermediate Language) code. At runtime, the CLR's just-in-time compiler (JIT compiler) converts the MSIL code into code native to the operating system. Alternatively, the MSIL code can be compiled to native code in a separate step prior to runtime. This speeds up all later runs of the software as the MSIL-to-native compilation is no longer necessary.

Although some other implementations of the Common Language Infrastructure run on non-Windows operating systems, the CLR runs on Microsoft Windows operating systems.

The virtual machine aspect of the CLR allows programmers to ignore many details of the specific CPU that will execute the program. The CLR also provides other important services, including the following:

- Memory management
- Thread management
- Exception handling
- Garbage collection
- Security

## **Introduction to ASP.NET:**

Although so Microsoft Visual Basic.NET is a powerful but simple language aimed primarily at developers creating web applications for the Microsoft .NET platform. It inherits many of the best features of C++ and Microsoft Visual Basic, but with some of the inconsistencies and anachronisms removed, resulting in cleaner and logical language. VB also contains a variety of useful new innovations that accelerate application development, especially when used in conjunction with Microsoft Visual Studio .NET.

The Common Language Runtime provides the services that are needed for executing any application that's developed with one of the .NET languages. This is possible because all of the .NET languages compile to a common Intermediate Language. The CLR also provides the common type system that defines those data types that are used by all the .Net languages. That way, you can use same data types regardless of what.NET language you're using to develop your application implementations.

## **Features of ASP.NET:**

- (1) Component Infrastructure.
- (2) Language Integration.
- (3) Internet Interoperation.
- (4) Simple Development.
- (5) Simple Deployment.
- (6) Reliability.
- (7) Security

## **Introduction to Microsoft SQL Server:**

**Microsoft SQL Server** is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time). Its primary query languages are T-SQL and ANSI SQL.

## **SQL Server 2008 R2:**

**SQL Server 2008 R2** (10.50.1600.1, formerly codenamed "Kilimanjaro") was announced at TechEd 2009, and was released to manufacturing on April 21, 2010. SQL Server 2008 R2 adds certain features to SQL Server 2008 including a master data management system branded as Master Data Services, a central management of master data entities and hierarchies. Also Multi Server Management, a centralized console to manage multiple SQL Server 2008 instances and services including relational databases, Reporting Services, Analysis Services & Integration Services.

**SQL Server 2008 R2** includes a number of new services, including PowerPivot for Excel and SharePoint, Master Data Services, Stream Insight, Report Builder 3.0, Reporting Services Add-in for SharePoint, a Data-tier function in Visual Studio that enables packaging of tiered databases as part of an application, and a SQL Server Utility named **UC** (Utility Control Point), part of **AMSM** (Application and Multi-Server Management) that is used to manage multiple SQL Servers.

The first SQL Server 2008 R2 service pack (10.50.2500, Service Pack 1) was released on July 11, 2011.

The second SQL Server 2008 R2 service pack (10.50.4000, Service Pack 2) was released on July 26, 2012.

# **FEASIBILITY STUDY**

## **FEASIBILITY STUDY:**

Feasibility study is a process to check possibilities of system development. It is a method to check various different requirements and availability of financial & technical resources.

Before starting the process various parameters must be checked like:

- Estimated finance is there or not?
- The man power to operate the system is there or not?
- The man power is trained or not?

All the above conditions must be satisfied to start the project. This is why in depth analysis of feasibility is carried out.

There are three different ways feasibility can be tested:

- 1) Economical Feasibility
- 2) Technical Feasibility
- 3) Operational Feasibility.

### **Economical Feasibility:**

In economic feasibility, analysis of the cost of the system is carried out. The system should be only developed if it is going to give returned the current manual system user can get the price only by purchasing the newspapers. In addition if he/she wants to see archives of particular equity then he has to refer to all the old newspapers. For research reports he has to buy another magazine. So Instead of buying no of magazines user has to just go online and with a single click he can get whatever information he wants. So our project of online share news passes the test of economic feasibility.

### **Technical Feasibility:**

It is basically used to see existing computer, hardware and software etc., weather it is sufficient or additional equipment's are required? Minimum System Requirement is such that it can be affordable by of the user who is having computer. All the user requires is compatible browser and .net framework installed so our system is fully technical feasible.

**Operational Feasibility:**

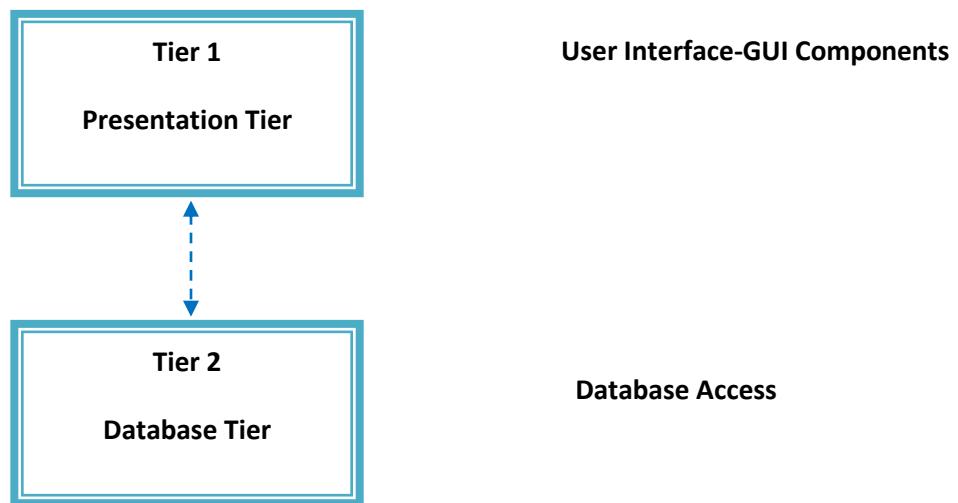
Once the system is designed there must be trained and expert operator. If there are not trained they should give training according to the needs of the system.

From the user's perspective our system fully operational feasible as it just requires some knowledge of computer. Operators only need add daily prices of various equities and there are enough validations available so operator does not require any special technical knowledge. So our system also passes the test of operational feasibility.

# **ARCHITECTURE AND DESIGN OF THE PROJECT**

## **ARCHITECTURE AND DESIGN OF THE PROJECT**

Since the application will have client server architecture, it will have two-tier architecture.



# TABLE DESIGN

**TABLEDESIGN**

## Matrimonial Web Application

Table:admin_login			
Field Name	Data type	Null	Description
Lid	int	No	Stores the admin id
User_nm	varchar	No	Stores the admin name
Password	varchar	No	Stores the admin password

Table:Bank_Details_cc			
Field Name	Data type	Null	Description
Cc_id	int	No	Stores the id
Card_no	varchar	No	Stores the card number
Card_type	varchar	No	Stores the type of card
Card_exp_month	varchar	No	Stores card's expiry month
Card_exp_year	varchar	No	Stores card's expiry year
Pin_code	varchar	No	Stores card pin code
Name_on_card	varchar	No	Stores user name on the card
Credit_limit	numeric	No	Stores limited amount of card
Avail_bal	int	No	Stores available balance amt in card

Table:Chat_Box			
Field Name	Data type	Null	Description

## Matrimonial Web Application

To_uid	varchar	No	Stores the receiver's email ids
From_uid	varchar	No	Stores the sender's email ids
Messages	varchar	Yes	Stores the text messages

**Table:Contact\_us**

Field Name	Data type	Null	Description
Ct_id	int	No	Stores the contact id
Name	varchar	No	Stores the user name
Mobile	varchar	No	Stores the contact number
Email id	varchar	No	Stores the user's e-mail id
Details	varchar	Yes	Stores the details
Ct_date	DateTime	No	Stores the contact date and time

**Table:Express\_interest**

Field Name	Data type	Null	Description
To_uid	varchar	No	Stores the receiver's email ids
From_uid	varchar	No	Stores the sender's email ids
Messages	varchar	Yes	Stores the text messages

**Table: feedback**

## Matrimonial Web Application

Field Name	Data type	Null	Description
fd_id	varchar	No	Stores the feedback id
Name	varchar	No	Stores the name of user
Comment	varchar	Yes	Stores the comments
Feed_date	datetime	No	Stores the date of feedback

**Table:image\_gallery**

Field Name	Data type	Null	Description
Img_id	int	No	Stores the id of image
Images	varchar	No	Stores the image
Pro_id	int	No	Stores the user id

**Table: Profile**

Field Name	Data type	Null	Description
pro_id	int	No	Stores the user profile id
Profile_creator	varchar	No	Stores the data of profile creator
Names	varchar	No	Stores the name of user
Marital_status	varchar	No	Stores the data of marital status
Body_type	varchar	No	Stores the data of body type
Age	varchar	No	Stores the age of user
Physical_status	varchar	No	Stores the physical status of user

## Matrimonial Web Application

Height	varchar	No	Stores the height of user
Weight	varchar	No	Stores the weight of user
Mother_tongue	Varchar	No	Stores the mother tongue of user
Religion	Varchar	No	Stores the religion of user
Caste	Varchar	No	Stores the caste of user
Gothram	Varchar	No	Stores the gothram of user
Zodiac	Varchar	No	Stores the zodiac of user
Star	Varchar	No	Stores the star of user
Eating_habit	Varchar	No	Stores the habit of user
Drinking_habit	Varchar	No	Stores the habit of user
Smoking_habit	Varchar	No	Stores the habit of user
Country	Varchar	No	Stores the country of user
City	Varchar	No	Stores the city of user
State	Varchar	No	Stores the state of user
Education	Varchar	No	Stores the qualification of user
Occupation	Varchar	No	Stores the occupation of user
Employeed_in	Varchar	Yes	Stores the employee status of user
Salary	Varchar	No	Stores the salary data of user
Mobile_no	Varchar	No	Stores the contact no of user
Images	Varchar	No	Stores the profile image of user
About_me	Varchar	Yes	Stores the data about user
Require_details	Varchar	Yes	Stores the detail of user
Membership	Varchar	No	Stores the info about membership

## Matrimonial Web Application

Paid_status	Varchar	No	Stores the paid status of user
Start_date	Varchar	No	Stores the start date of account
End_date	Datetime	No	Stores the end date of account
Email_id	varchar	No	Stores the email id of user
Password	Varchar	No	Stores the password of user a/c
Gender	Varchar	No	Stores the gender of user
Profession	Varchar	No	Stores the profession of user
Address	Varchar	No	Stores the address of user
Pincode	Varchar	No	Stores the address pin code
Phoneno	Varchar	No	Stores the phone no of user
Paid_date	Datetime	No	Stores the date of payment
Login_status	int	No	Stores the login status of user

**Table:Search\_setting**

Field Name	Data type	Null	Description
Pro_id	Int	No	Stores the user's profile id
Caste	varchar	No	Stores the caste of user
salary	varchar	No	Stores the salary of user
Education	varchar	No	Stores the education of user
Occupation	Varchar	No	Stores the occupation of user
maritalstatus	Varchar	No	Stores the marital status of user

## Matrimonial Web Application

**Table:send\_msg**

Field Name	Data type	Null	Description
To_uid	varchar	No	Stores the receiver's email ids
From_uid	varchar	No	Stores the sender's email ids
Messages	varchar	Yes	Stores the text messages

**Table:success\_story**

Field Name	Data type	Null	Description
Usr_id	varchar	No	Stores the user id
Story	varchar	No	Stores the success story
Ad_date	Datetime	no	Stores the upload date of story

**Table: zone-master**

Field Name	Data type	Null	Description
Country	varchar	No	Stores the name of country
State	varchar	No	Stores the name of state
city	varchar	No	Stores the name of city

# **CODING STANDARDS**

## **CODING STANDARD**

<b>Checks for Normal Working</b>		<b>To Be Checked?</b>
Does your screen save data in the database?		Yes
Does your screen save correct data in the database?		Yes
In update does your screen load correct data?		Yes
<b>Fields are showing the data in correct format? [USE REGULAR EXPRESSIONS DEFINED AS PER TESTING STANDARDS]</b>		
DateShould always be selectable in addition to manual entry		Yes
Integer		Yes
Float, Double		Yes
Percentage		Yes
For web site entry field, populate by default with value		Yes
All email entry fields should be multiple email addresses enabled		Yes
Will your screen crash if wrong data type is entered?		Yes
Does your screen sequence is working?		Yes
If the logic updated / approved?		Yes

<b>Basic Validations</b>		
Maximum allowed length property is set?		Yes
Validation for Required field is done?		Yes
Validation for Integer, Float, Double, Date, Time is done?		Yes
Spell check		Yes

## Matrimonial Web Application

Type Check / Type Safety	Yes
Boundary Value Analysis (for highest order value and lowest order value)	Yes
Date Format (dates must be converted from SQL only) NEVER DISPLAY 12:00:00 AM	Yes
Is page validation enabled? [must be done for user entry fields]	Yes

# ENTITY RELATIONSHIP

## DIAGRAM

### ER-DIAGRAM

Entity Relation Diagram:

The Entity Relationship Diagram (Model) is based on perception of a real world that consists of a collection of basic objects called as Entity and relationships among these objects. Entities in database are described as set of attributes.

- A **Relationship** is an association among several **Entities**.
- The set of **Entities** of the same type are called as **Entity Set**.
- The set of Relationships of same type are called as **Relationship Set**.

A graphical model of the data needed by the system, including think about which information is stored and the relationship among them, produced in structured analysis and information engineering.

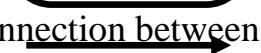
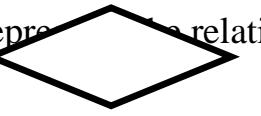
The relational approaches to system development places a great deal of emphasis on data storage requirements include the data entities, their attributes and the relationship among the data entities. The model used to define the data storage requirements is called the Entity Relationship Diagram.

On the Entity Relationship Diagram, a rectangle represents data entities, and lines connecting the rectangle show the relationship among data entities.

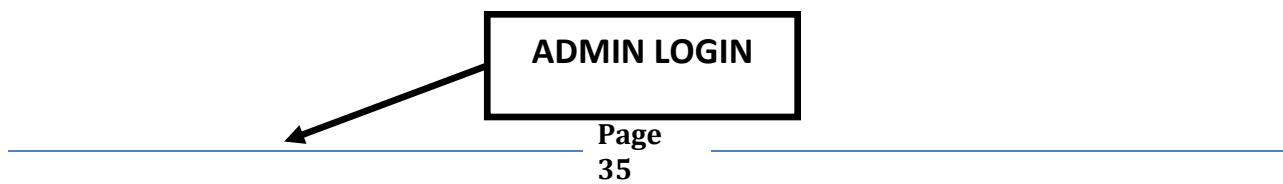
### Notations used in ER-Diagram:

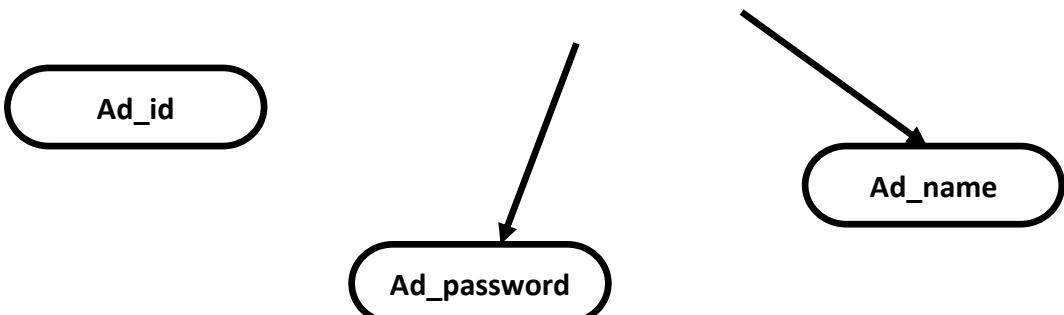
- = Represents Data Entity.  

- = Attribute  

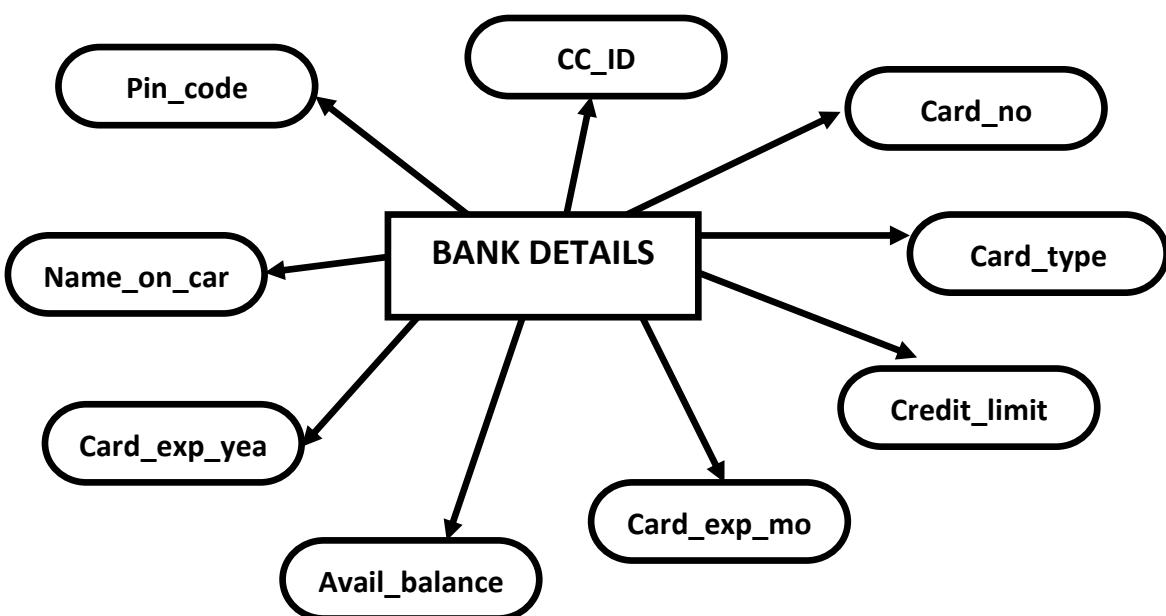
- = Connection between two or more entities.  

- = Represents a relationship.  

- = Connection between two or more entities.  


### 1. Admin Table:

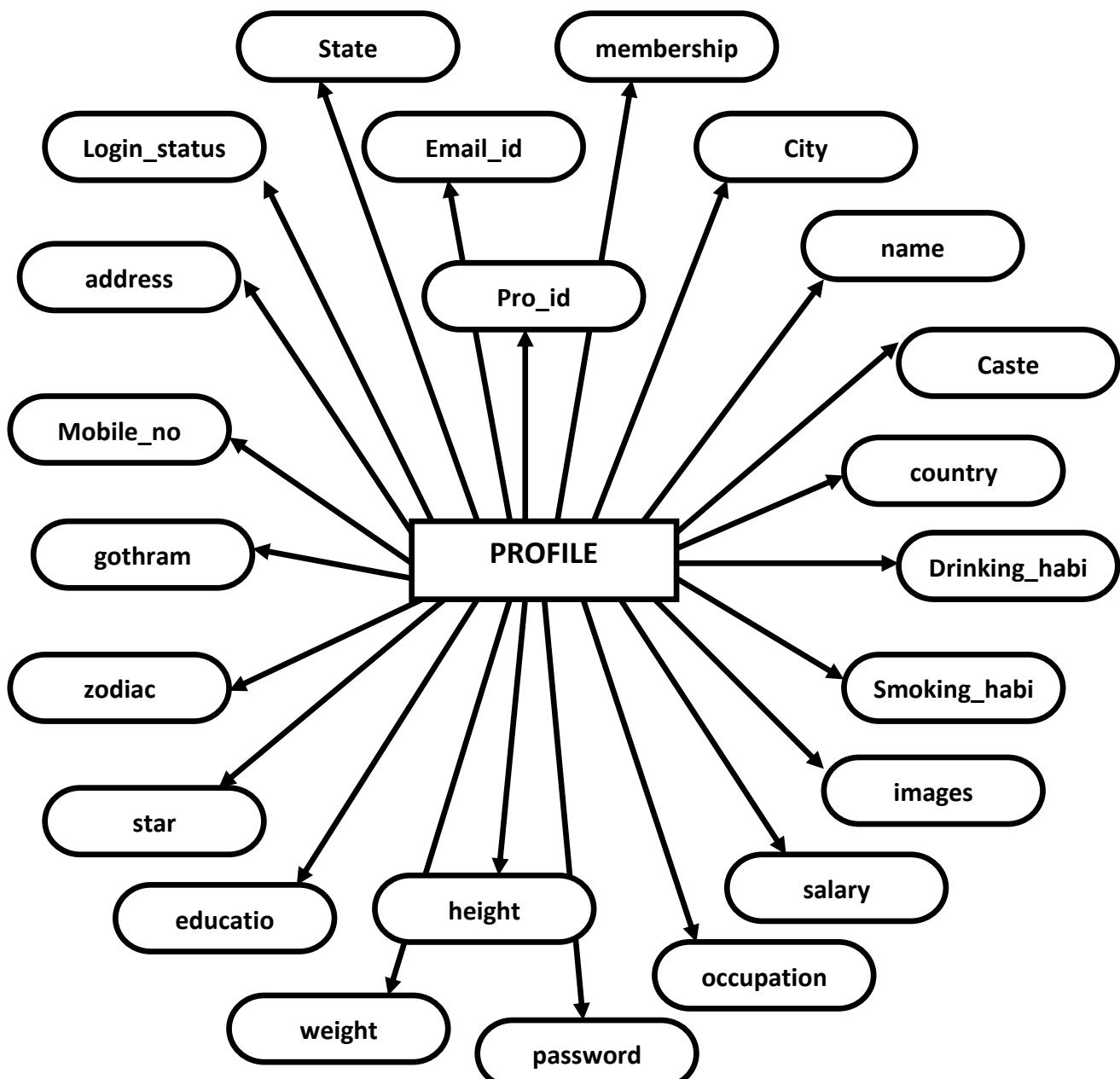




## 2. Bank Details Table:



## 3. Profile Table:

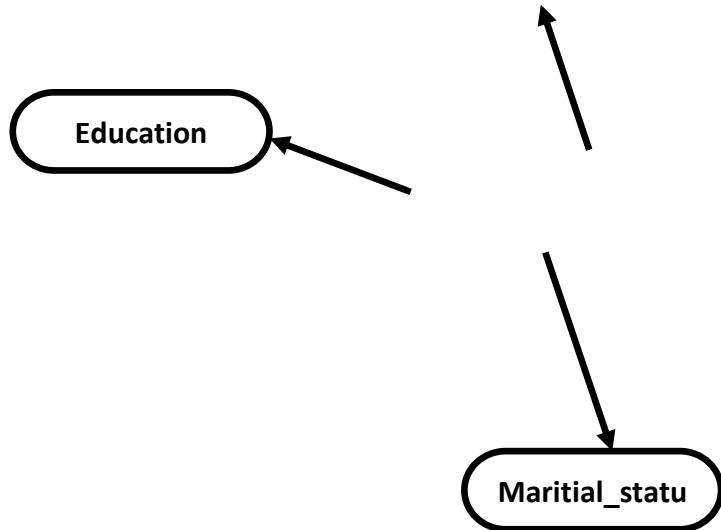


#### 4. Search Profile Table:

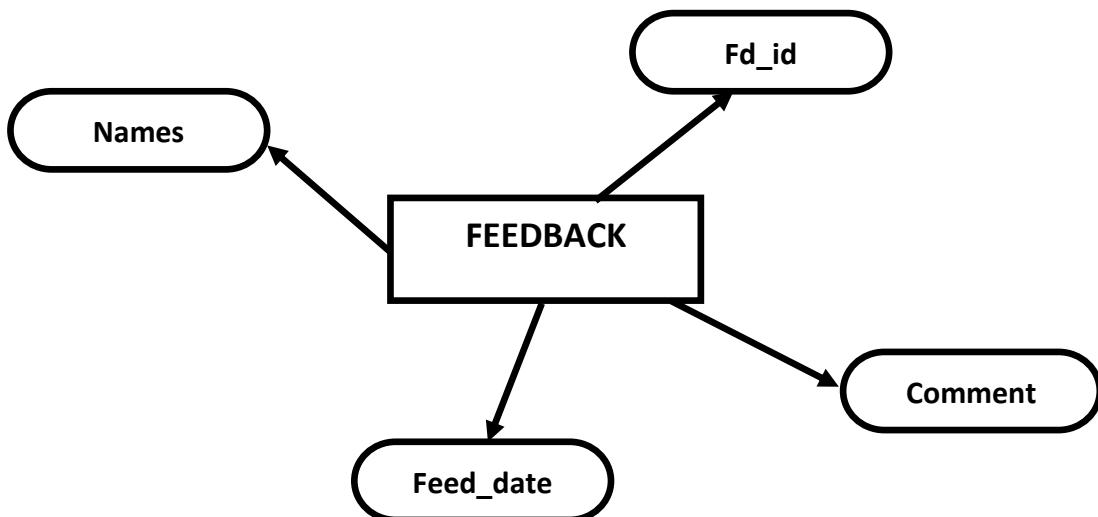
Pro\_id

Page  
37

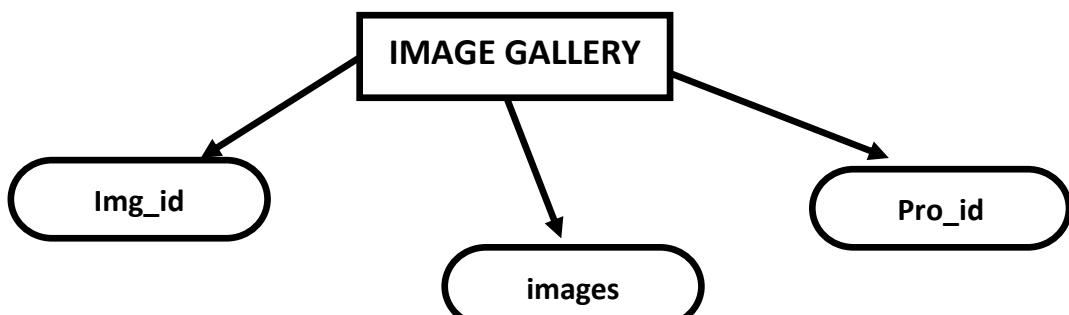
Caste



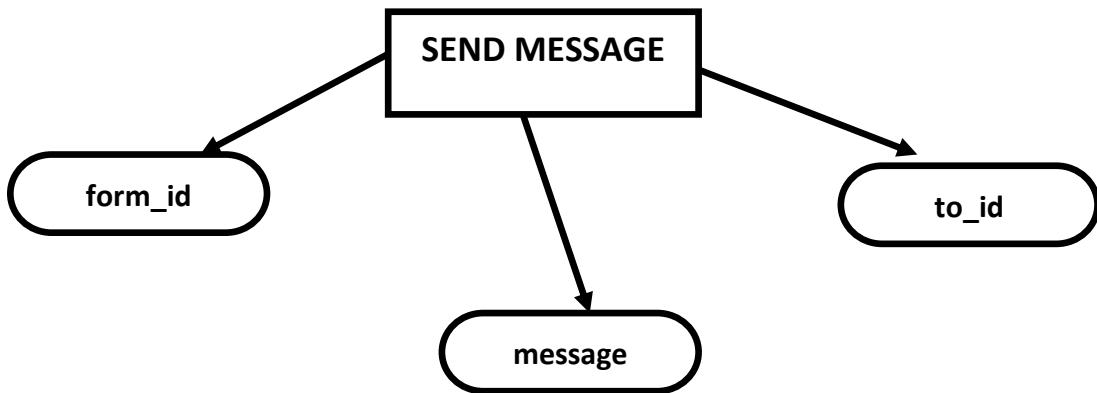
**5. Feedback Table:**



**6. Image Gallery Table:**

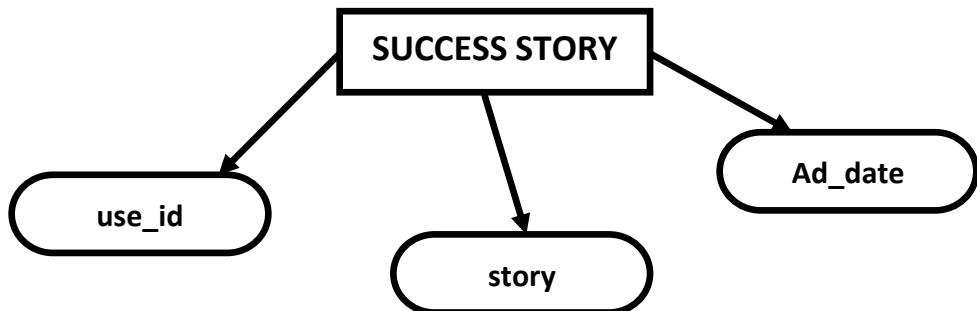


#### 7. Send Message Table:



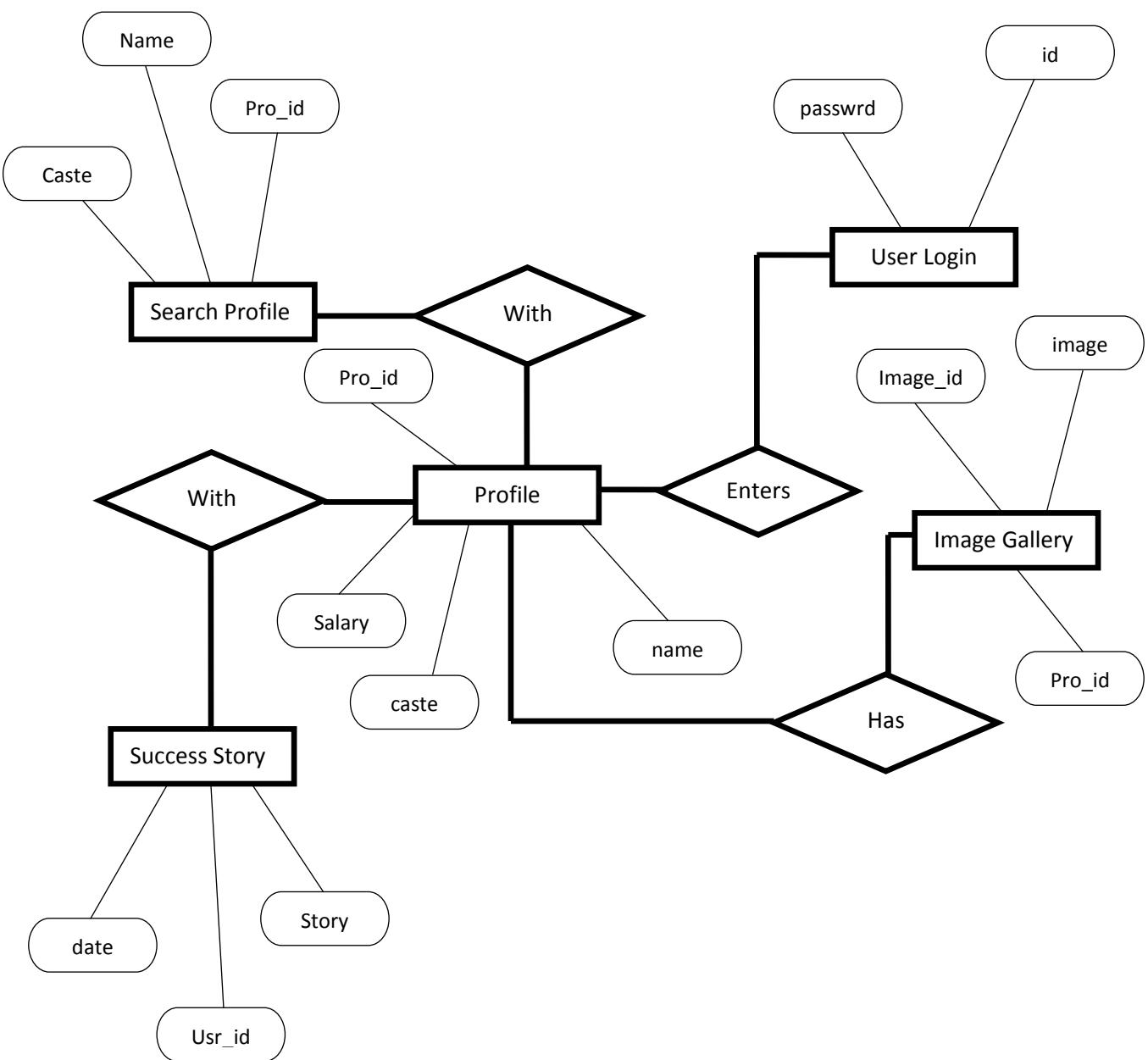
#### 8. Success Story Table:

## Matrimonial Web Application



### Relation between Tables:

## Matrimonial Web Application



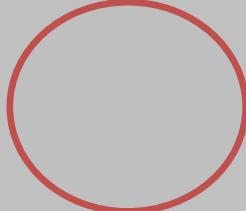
# DATA FLOW DIAGRAM

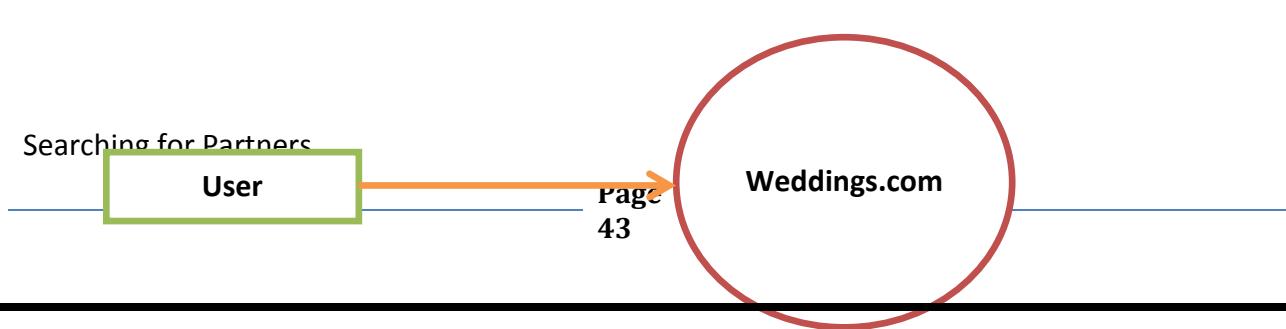
Data Flow Diagram:

A Data flow diagram (DFD) is a graphical system model that shows all of the main requirements for an information system in one diagram inputs and outputs, processes, data storage. A DFD describes what data flows rather than how it is processed. Everyone working on the development project can see all aspects of the system working together at once with DFD. That is one reason for its popularity. The DFD is also easy to read because it is graphical model.

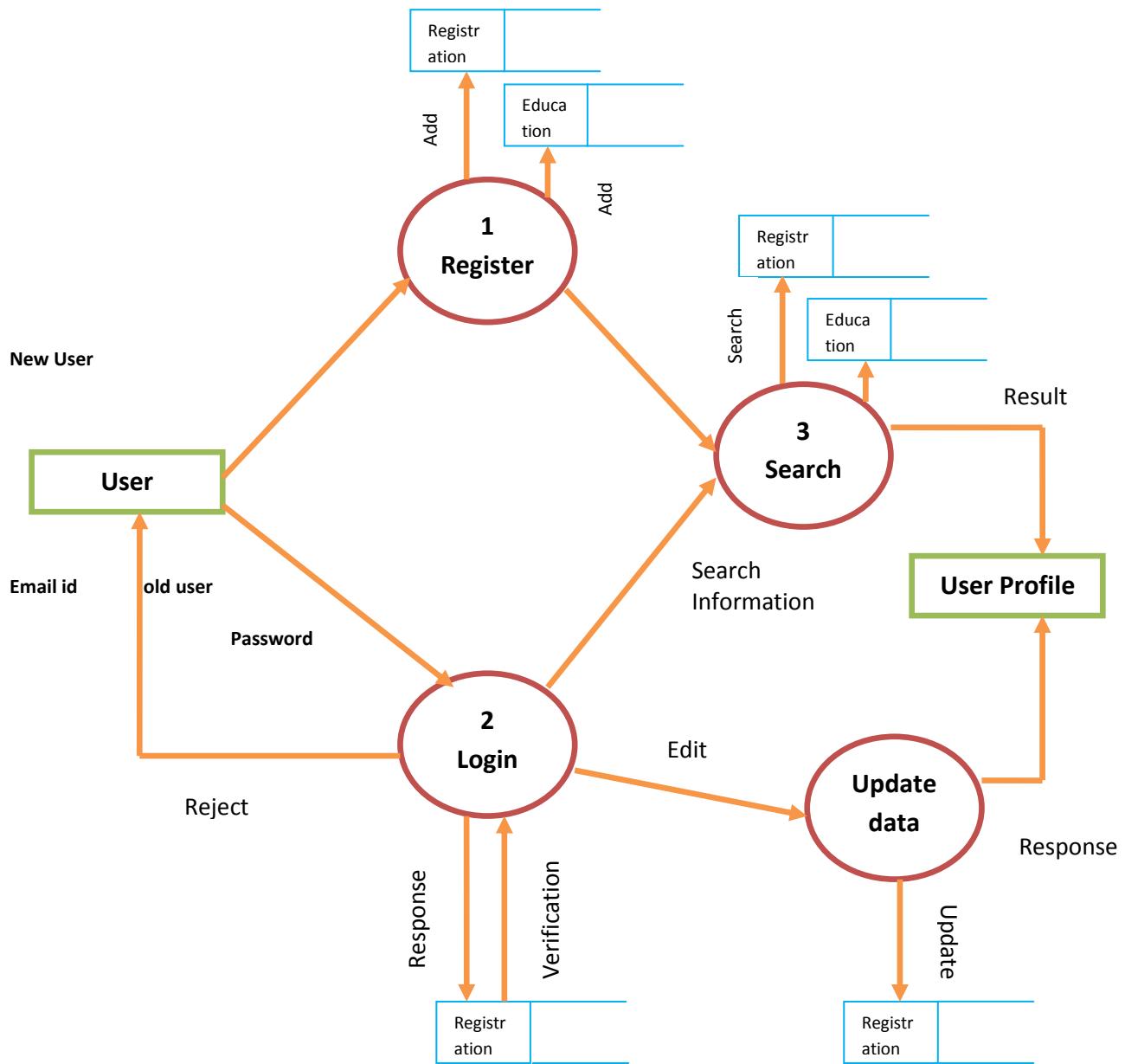
The DFD is mainly used during problem analysis. End user, management and all information system workers typically can read and interpret the DFD with minimal training.

**Symbol:**

Sr. No.	Symbol	Name	Description
1		External entity	An external entity is source or destination of data flow which is outside the area of study.
2		Process	A process shows a transformation or manipulation or multiplication of data flow within the system.
3		Dataflow	A data flow shows the flow of information from source to destination .A data flow is represented by a line with arrowheads Showing the flow.



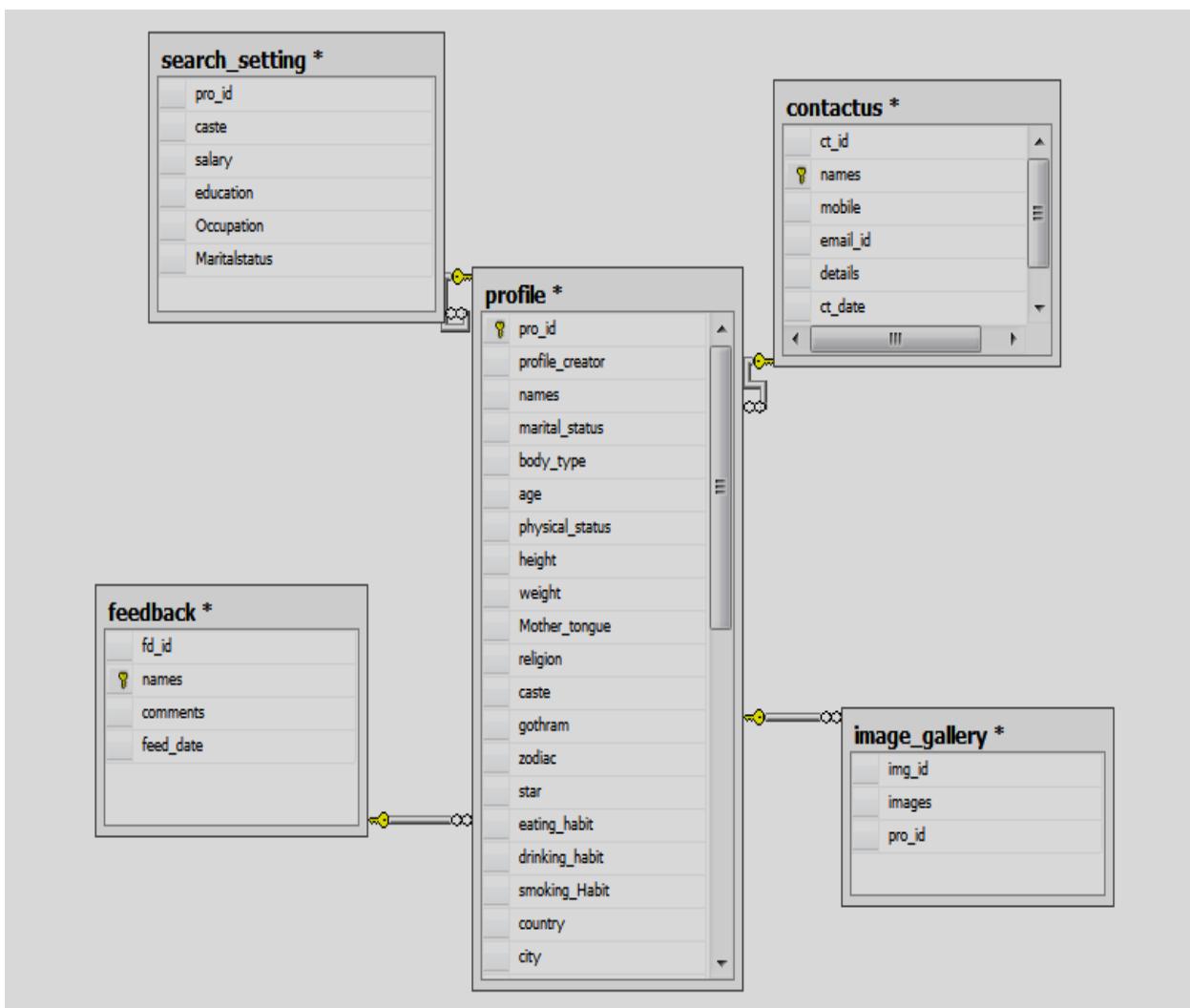
## Matrimonial Web Application



# **CLASS DIAGRAM**

**CLASS DAIGRAM:**

## Matrimonial Web Application



## Matrimonial Web Application

**send\_msg**

from\_id  
to\_id  
msg

**success\_story**

us\_id  
story  
ad\_date

**admin\_login**

id  
user\_nm  
pass

**zone\_master \***

country  
state  
city

**Bank\_Details\_cc**

cc\_id  
Card\_no  
Card\_Type  
card\_exp\_month  
card\_exp\_yr  
pin\_code  
name\_on\_card  
credit\_limit  
avail\_bal

**chat\_box**

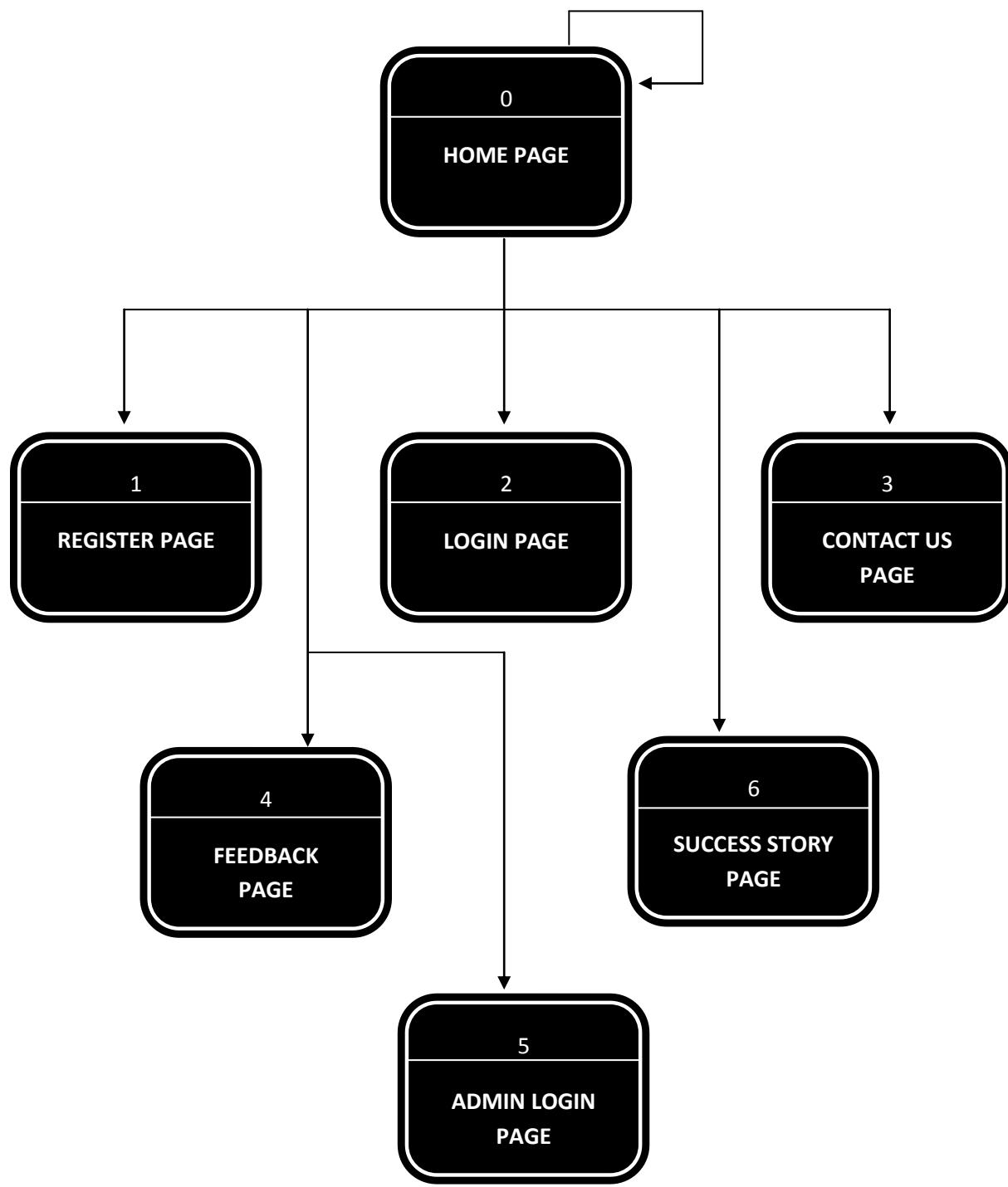
to\_uid  
from\_uid  
messages

**express\_interest**

from\_id  
to\_id  
msg

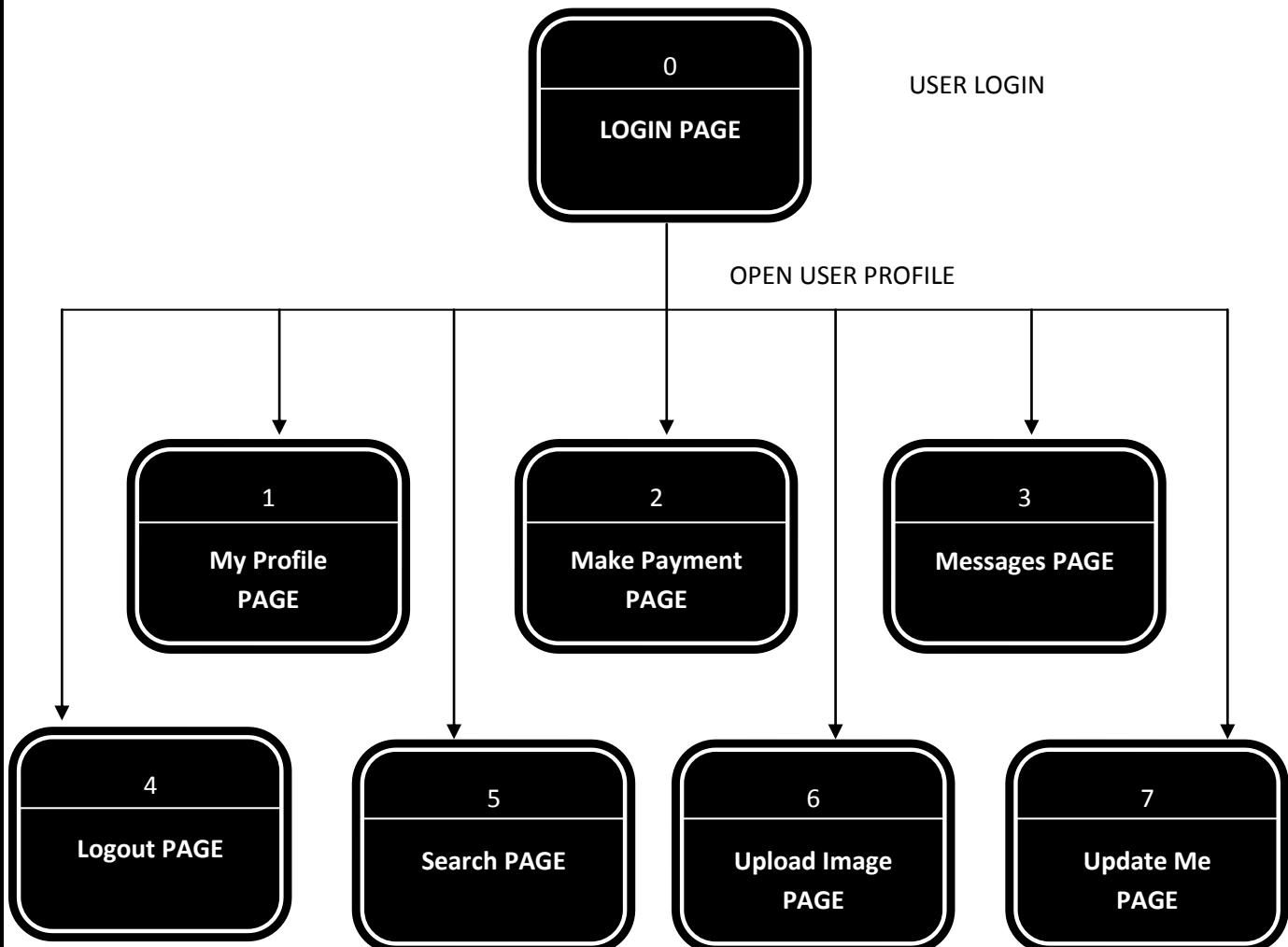
# APPLICATION NAVIGATION

## Matrimonial Web Application

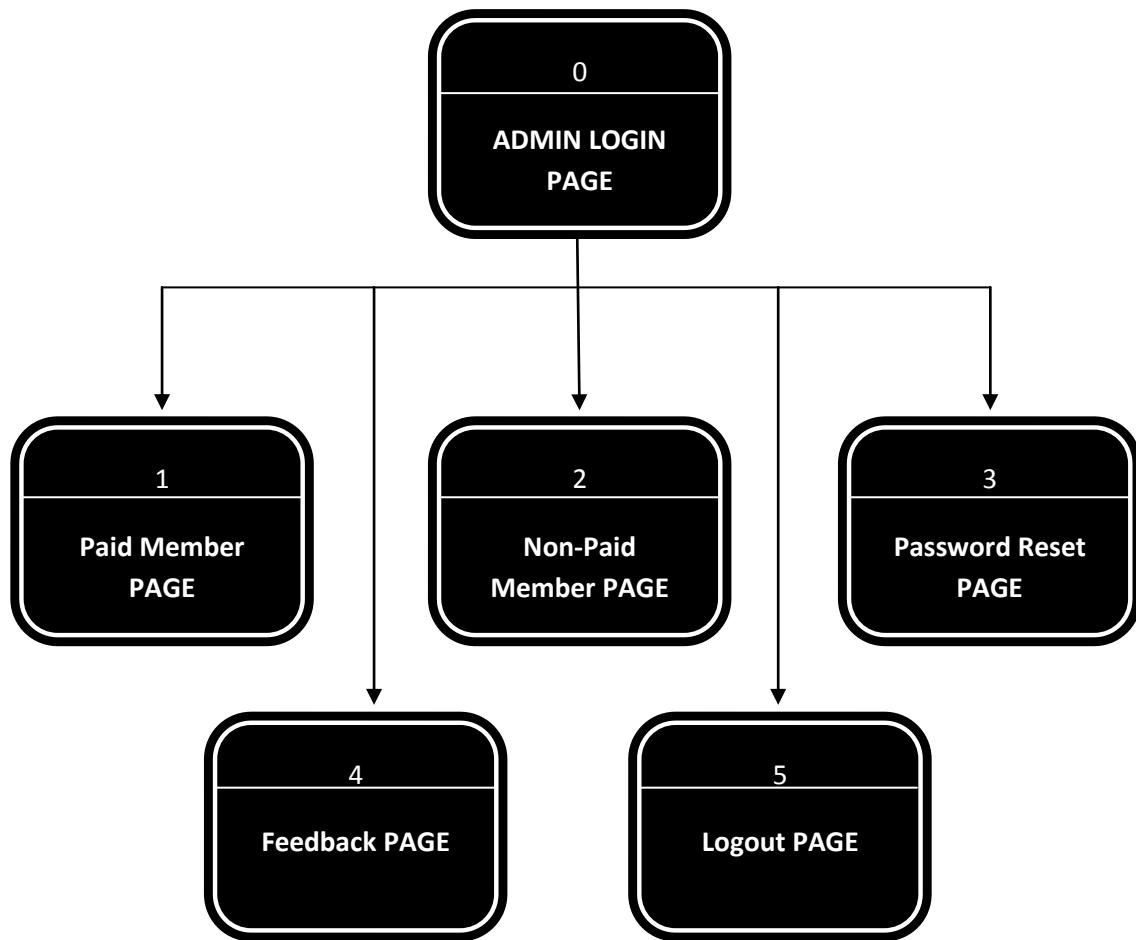


### HOME PAGE NAVIGATION

## Matrimonial Web Application



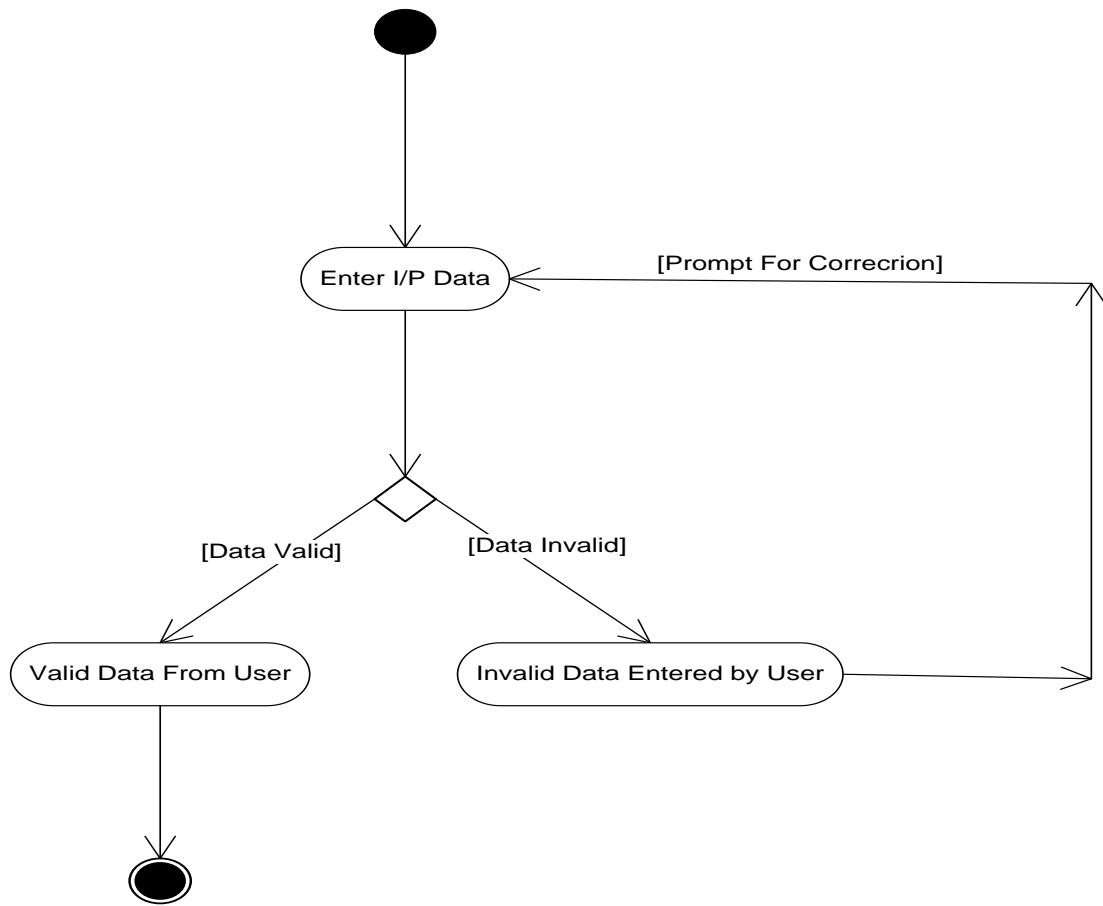
### LOGIN PAGE NAVIGATION



ADMIN LOGIN PAGE NAVIGATION

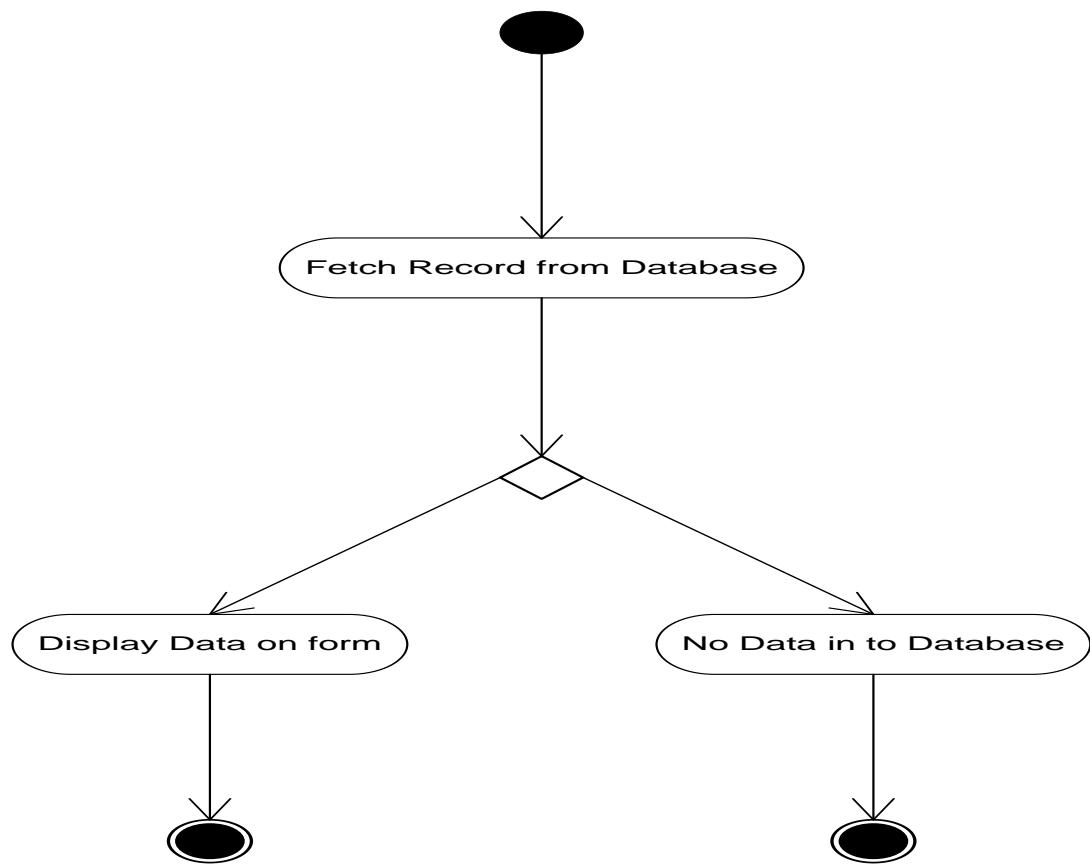
# **ACTIVITY DIAGRAM**

**ACTIVITY DIAGRAM:**

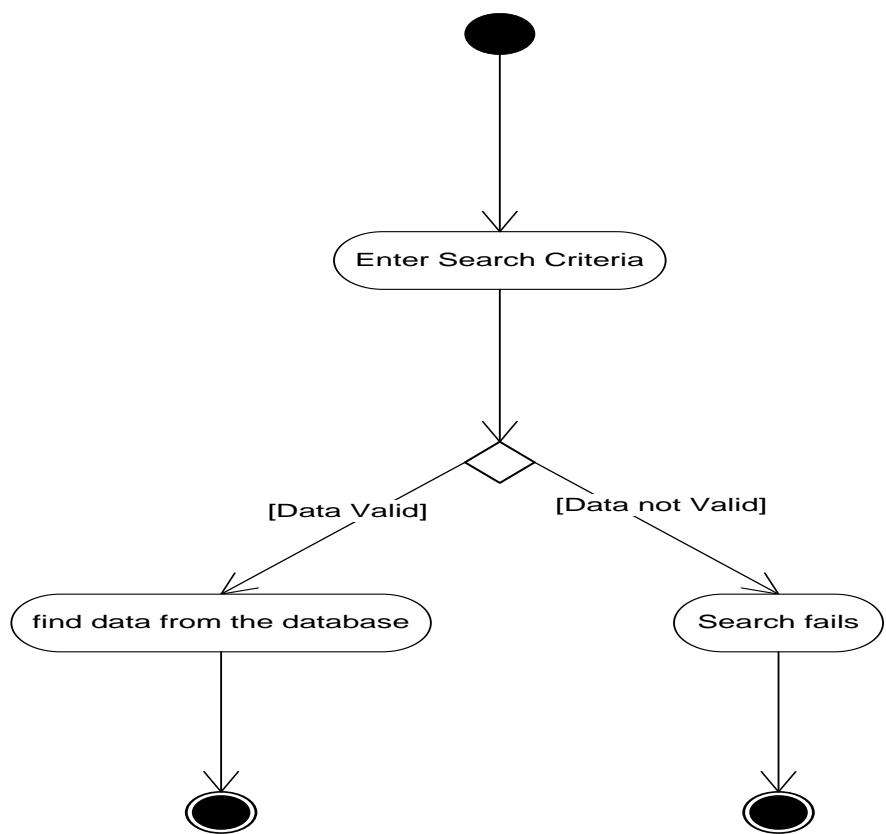


**INPUT VALIDATION**

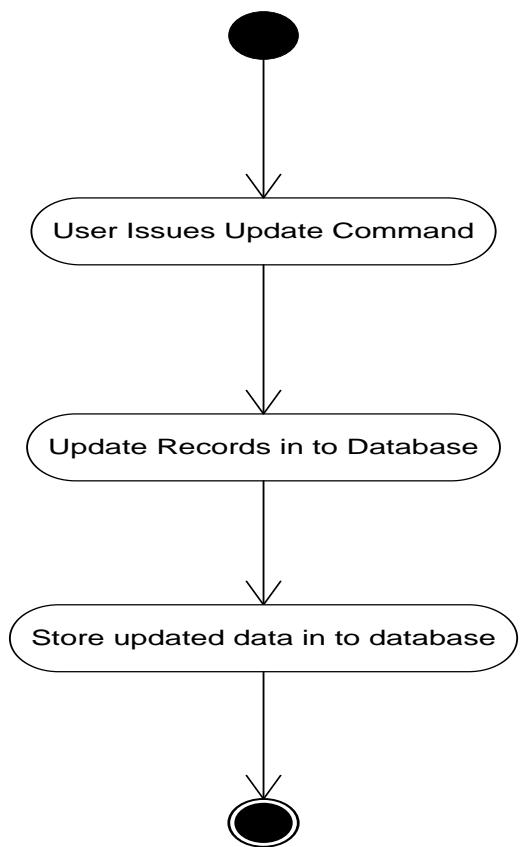
**DISPLAY RECORDS:**



## SEARCH RECORDS:



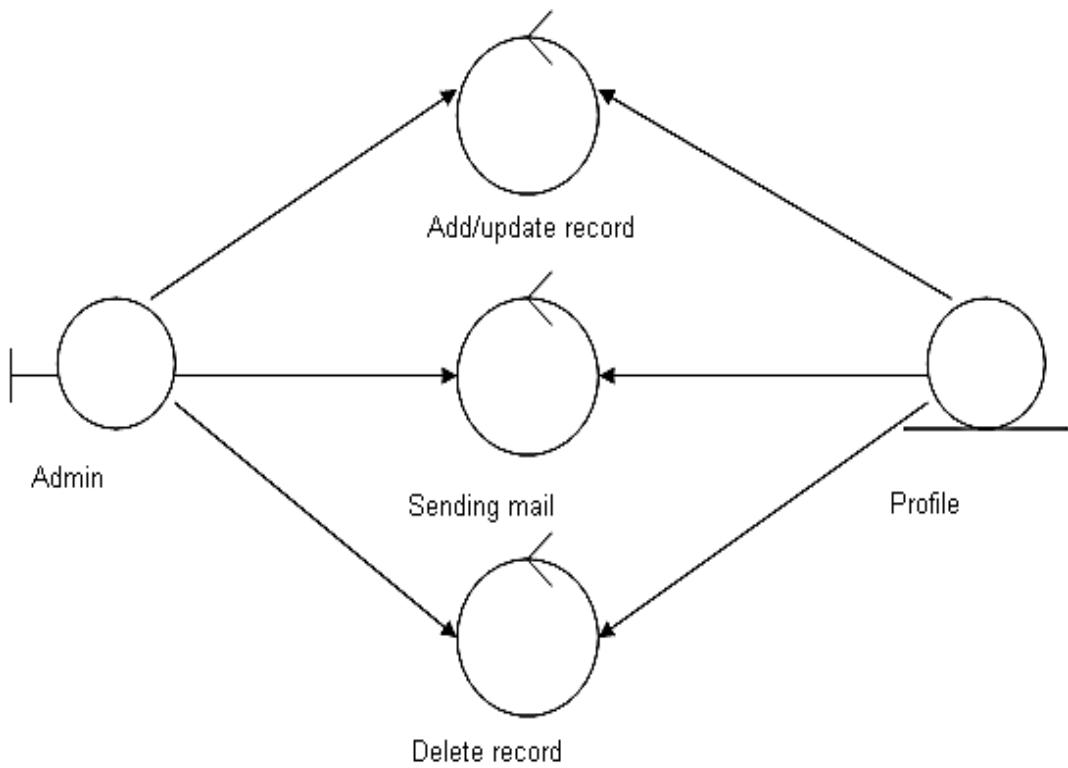
## UPDATE RECORDS:



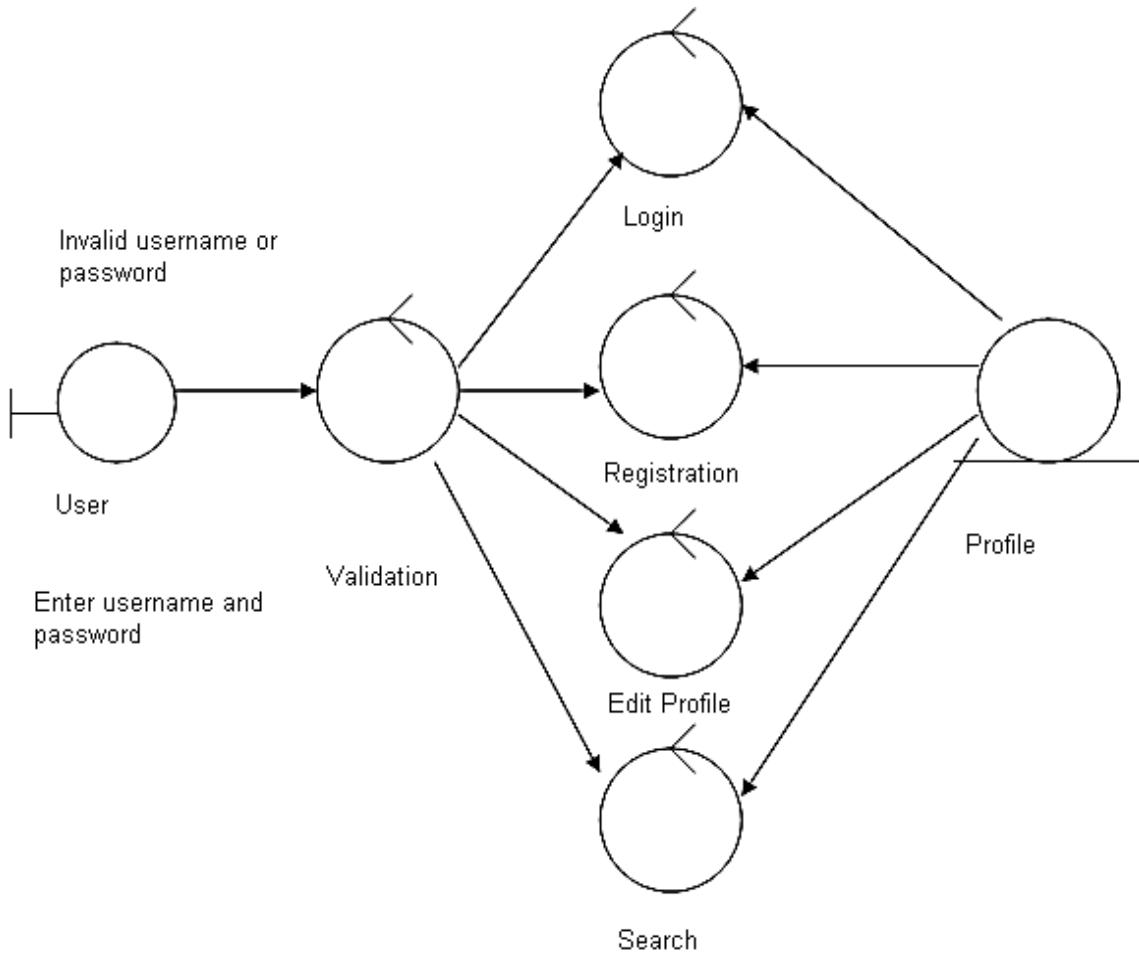
# **CASE DIAGRAM**

## **CASE DIAGRAMS**

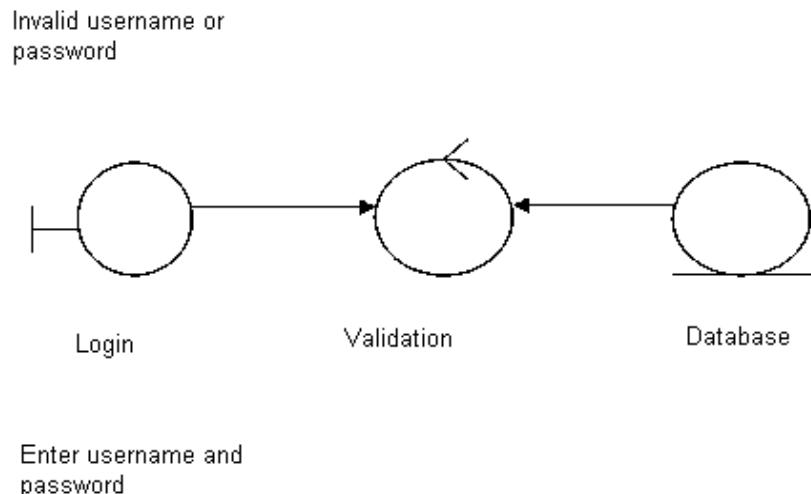
### **Admin case diagram:**



**User Case Diagram:**



**Login Case Diagram:**



# **EVENT TABLE**

## **EVENT TABLE**

A table that lists events in tabular format that is in rows and key pieces of information about each event in columns is known as event table.

While developing the list of events, the analyst should note additional information about each event for later use.

This information is entered in an event table.

An event table comprises of rows and columns.

Each row in an event table records information about one event.

And each column is about its key piece of information about that event.

### **Attributes of Event Table:**

#### **Trigger:**

An occurrence that tells the system that has occurred, either the arrival of data needing or at a point in time.

#### **Source:**

An external agent or actor that supplies data to the system.

#### **Activity:**

Behavior of that the system performs when an event occurs.

#### **Response:**

An output produced by the system, that goes to a destination.

#### **Destination:**

An external agent or an actor that receives data from the system.

## Matrimonial Web Application

Sr. No.	Event	Trigger	Source	Activity	Response	Designation
1	Click on Admin	To open admin page	Admin	Display Admin page	Form open	Admin
2	Click on login	To open profile of user	User	Display User profile	Form open	User
3	Click on Feedback	To send feedback	User	Display Feedback form	Form open	User
4	Click on Contact us	To contact	User	Display contact us form	Form open	User
5	Click on Register	To create new user profile	User	Display Registration form	Form open	User
6	Click on Success Story	To see the story	User	Display Form of success story	Form open	User
7	Click Update me	To edit user profile	User	Display update form	Form open	User
8	Click on generate	To generate paid member list	Admin	Retrieve data	Display the list in table	Admin
9	Click on Upload photo	To upload pictures	User	Display Picture upload	Form open	User
10	Click on find password	To send user back	Admin	Retrieve data	Open dialog box	Admin
11	Click on logout	To exit to the profile	Admin and user	Display the home page	Form open	Admin and user
12	Click on message	To express interest	User	Display the send message form	Form open	User

## Matrimonial Web Application

13	Click on Search	To search the matches	User	Display the search form	Form open	User
14	Click on the profile image	To seethe details	User	Display Details of user	Form open	User
15	Click on Success Story	To write the story	User	Display success story form	Form open	User
16	Click on home	To get back to the home	User	Display home page	Form open	User
17	Click on make payment	To become a paid member	User	Display select type of membership	Form open	User
18	Click on submit	To open the make payment page	User	Display payment form	Form open	User
15	Click on chat	To chat with member	User	Display members online	Form open	User

# **TESTING**

## **TESTING PLAN:**

Similar to the project plan, due to confidentiality issues, we cannot provide details test plan to the development team. We will still add the core components that make up our test plan.

- Test plan identifier
- References
- Introduction
- Test items (functions)
- Application risk issues.
- Features to be tested
- Features not to be tested
- Approach (strategy)
- Item pass/fail criteria
- Entry & exit criteria
- Suspension criteria & resumption requirements
- Test deliverables
- Remaining test tasks
- Environmental needs
- Staffing and training needs
- Responsibilities
- Planning risks and contingencies
- Approvals
- Glossary

## **TESTING STRATEGY:**

Test More and Test Frequent is organization's tagline for testing. A typical screen in asp.net is tested at four levels before it goes for production.

Level 1 is generally the work to be tested by other developers or other interns (this is typical first level of testing where focus is not on requirement but end user testing) Ratio: 0% end user: 100% Technical

Level 2 is level where a senior programmer comes into the testing cycle of the screen that was unit tested by the developer in this phase the onus is to test software for technical requirements specified.

Ratio: 80% Technical: 20% end user

Level 3 is where a tester will come into picture. The tester will test the software for both end user as well as technical point of view.

The ratio here is: 50% Technical: 50% end user

Level 4 is where we make the code at Release-Ready. Here screen is tested to the core and each and every standard must be followed and verified.

Ratio here is: 80% User Testing – 20% Technical

This allows us to test a screen at four levels and at the end of four weeks when the screen goes to production, it is generally bug free because more people have looked at this screen from different viewpoints.

References available while testing:

- Project Plan.
- System Requirements specifications.
- High Level design document.

- Detail design document.
- Development and Test process standards.
- Methodology.
- Low level design.

Also organization has Separate module to store all bugs

So each screen is released for testing as a build and all information for that screen (till release) is maintained using this particular build.

### **CONTENT TESTING:**

Errors in Web Application content can be as trivial as minor typographical error as incorrect information, improper organization or validation of intellectual property laws. Content Testing attempt to uncover this and many other problems before the user encounter them.

#### **Content Testing Objectives**

There are three types of objectives.

- To uncover syntactic errors in text-based documents, graphical representation and other media.
- To uncover semantic errors in any content object represented as navigation occurs, and
- To find errors in organization or structure of content that is presented to the end-user

### **DATABASE TESTING:**

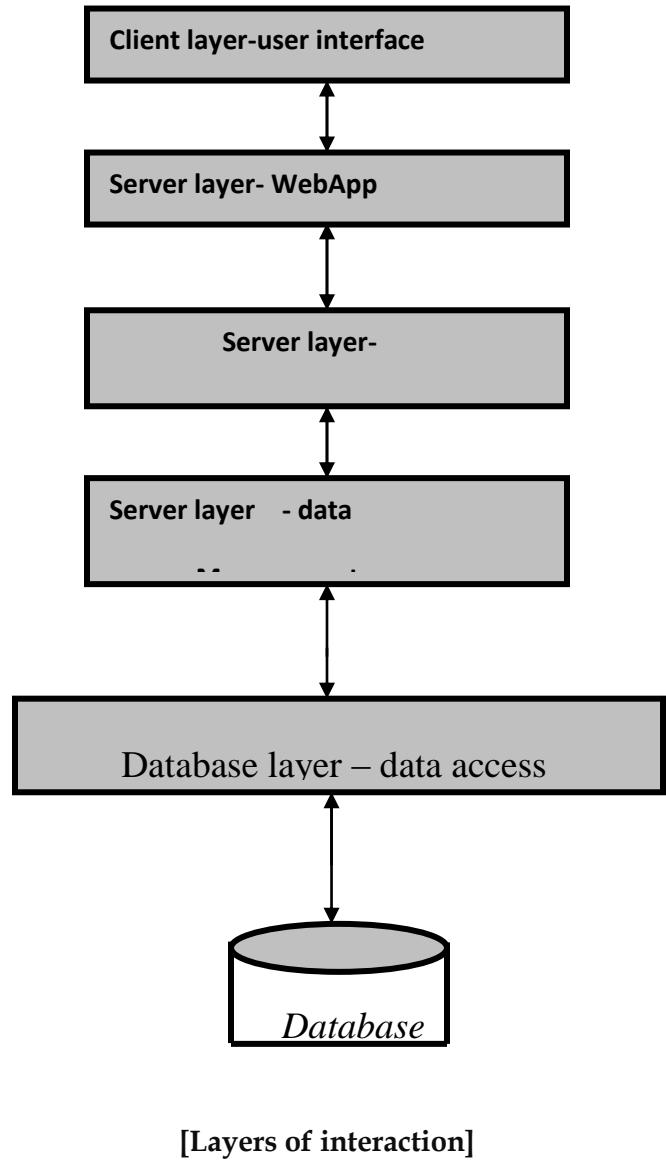
## **Matrimonial Web Application**

Modern Web Application does much more than present static content objects. In many application domains, Web Application interface with sophisticated database management system and build dynamic content object that are created in real time using the data acquired from a database.

Database Testing for Web Application is complicated by a variety of factor.

- 1) The original client side request for information is rarely presented in the form that can be input to a database management system.
- 2) The database may be remote to the server that houses the Web application.
- 3) RAW data acquired from the database must be transmitted to the Web application Server and properly formatted for subsequent transmittal to the client.
- 4) The dynamic content objects must be transmitted to the client in a form that can be displayed to the end user.

## Matrimonial Web Application



In figure testing should be ensure that,

1. Valid information is passed between the client and server from the interface layer
2. The Web application process script correctly and properly extracts or formats user data.
3. Queries are passed to a data management layer that communicates with database access routines.

4. User data are passed correctly to a server side data transformation function that format appropriate queries.

## **INTERFACE TESTING:**

Interface design model is reviewed to ensure that generic quality criteria established for all user interfaces have been achieved and that application specific interface design issue has been properly addressed.

### **Interface testing strategy:**

The overall strategy for interface testing is to (1) Uncover error related to specific Interface mechanisms (2) uncover errors in the way the interface implements the semantics of navigation, Web Application functionality, or content display. To accomplish this strategy, a number of objectives must be achieved.

Interface futures are tested to ensure that design rules, aesthetics, and related visual content are available for the user without error. Individual interface mechanisms are tested in a manner that is a logout to unit testing for examples; tests are designed to exercise all forms, client-side scripting, dynamic HTML. Each interface mechanism is tested within the context of a use-case or NSU for a specific user category. The interface is tested within a variety of environments to ensure that it will be compatible.

### **Testing Interface Mechanisms:**

When a user interacts with a Web Application, the interaction occurs through one or more interface mechanisms.

### Links: -

Each link is tested to ensure that the proper content object or function is reached. The Web engineer builds a list of all links associated with interface layout. And then executes each individually.

### Forms: -

At a microscopic level, tests are performed to ensure that labels correctly identified fields within the form and that mandatory fields are identified visually for the user. The server receives all information content within the form and their no data are lost in the transmission between client and server. Appropriate defaults are used when the user does not select from a pull down menu or set of buttons. Browser function don't corrupt data enter in a form and Scripts that perform error checking on data entered work Properly and provide meaningful error message.

### Client side scripting:-

Black box tests are conducted to uncover any error in processing As the script is executed . These tests are coupled with forms testing because script input is often derived from data provided as part of forms processing

### Dynamic HTML:-

Each Web page that contains dynamic HTML is executed to ensure that the dynamic display is correct. In addition a compatibility test should be conducted to ensure that the dynamic HTML is work properly in the environmental configuration that support the Web application.

### **Application specific interface mechanisms:-**

Test conforms to a checklist of functionality and features that are defined by the interface mechanism. Boundary test is minimum and maximum number of items that can be placed in to shopping chart. Test to determine persistence of shopping chart contents. Test to determine whether the Web Application can be record shopping chart content at some future date.

### **USABILITY TESTING:**

Usability test may be designed by Web engineering team. Define a set of usability testing categories and identify goal for each. Design test that will enable each goal to be evaluated. Select participants who will conduct test. Instrument participant's interaction with Web Application while testing is conducted. Develop a mechanism for assessing the usability of the Web Application.

**The following test categories and objective illustrate establish testing:**

#### **Interactivity –**

Are interaction mechanisms easy to understand and use?

#### **Layout-**

Are navigation mechanisms, content and function place in a manner that allows the user to find them quickly?

#### **Readability-**

Is text well written and understandable?

#### **Aesthetics-**

Do layout color, typeface, and related characteristics lead to ease of use ?

**Display Characteristics-**

Does the Web Application make optimal use of screen size and resolution?

**Time Sensitivity-**

Can important features, functions and content be used in a timely manner?

**Accessibility-**

Is the Web application accessible to people who have Disabilities?

**COMPATIBILITY TESTING:**

Web application must operate within environment that differs from one another. Different computer, display device, OS, browser and network connection speed can have significant on Web application operation. Different browser some time produced slightly different results, regardless of the degree of HTML standardization within the Web application.

The Web Engineering team derives a series of compatibility, validation tests, derived from existing interface tests, navigation tests, performance tests and security tests.

**TESTING METHOD:**

Testing is presents an interesting anomaly for the software engineering activities, the engineer attempts to build software from an abstract concept to a tangible product. Now comes testing. The engineer creates a series of test case that are initiated to "demolish" the software that has been build. Infect, testing is the one step in the software process that could be viewed (psychologically, at least) as destructive rather than constructive.

## **Models of Testing:-**

There are different Models of testing. On the basis of testing methods there are two types of testing:

**1. Black-box testing.**

**2. White-box testing**

Black-box tests are used to demonstrate that software functions are operational, that input is properly accepted and output is correctly produced, and that integrity of external information is maintained.

White-box tests are used to examine the procedural details. It checks the logical paths by test case. It can also check the conditions, loops used in the software coding. It checks that loops are working correctly on defined boundary value.

## **WHITE-BOX TESTING:**

White-box testing sometimes called glass-box testing, is a test case design method that uses the control structure of the procedural design to drive the test case. Always we are thinking that there is no necessary to execute or check the loops and conditions. And large number of errors is uncovered. With using white-box testing methods, we have checked that; all independent paths within a function have been executed at least once.

All logical decisions are their true and false side. All loops working correctly at their boundary values and within their specified conditions.

In our coding we test that all the loops work truly in each module. The one technique of white-box testing is basis path testing. It contains two parts, one is flow graph notation and the second is cyclometer complexity. In flow graph notation we are checking

logical control of flow. By using cyclometer complexity we find complexity of our project structure.

### **BLACK-BOX TESTING:**

Black-box testing focuses on the functional requirements of the software. That is black-box testing enables the software engineer to drive sets of input conditions that will fully exercise all functional Requirements for the program. Black-box testing is not an alternative to white-box testing techniques. Rather, it is a complementary approach that is likely to uncover a different class of errors than white-box methods.

We use in our coding to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in database
- Performance errors
- Initialization and termination errors.

Unlike white-box testing, which is performed earlier in the testing process, black-box testing tends to be applied during later stages of testing. Because black-box testing purposely disregards control structure, attention is focused on the information domain.

By applying black-box techniques, we derive a set of test cases that satisfy following criteria .

Test cases that reduce, by a count that is greater then one, the number of additional test cases must be designed to achieve reasonable testing.

#### **Level 1 - Build Acceptance Tests**

Other related test cases ensure that adopters received the proper Development Release Document plus other build related information (drop point, etc.). The objective is to determine

if further testing is possible. If any Level 1 test case fails, the build is returned to developers un-tested.

### **Level 2 - Smoke Tests**

The objective is to determine if further testing is possible. These test cases should emphasize breadth more than depth. All components should be touched, and every major feature should be tested briefly by the Smoke Test. If any Level 2 test case fails, the build is returned to developers un-tested.

### **Level 2a - Bug Regression Testing**

Every bug that is “Open” during the previous build, but they marked as “Fixed, Needs Re-Testing” for the current build under test, is need to be regressed, or re-tested. Once the smoke test is completed, all resolved bugs need to be regressed. It should take between 5 minutes to 1 hour to regress most bugs.

### **Level 3 - Critical Path Tests**

Critical Path test cases must pass by the end of every 2-3 Build Test Cycles. They do not need to be tested every drop, but must be tested at least once per milestone. Thus, the Critical Path test cases must all be executed at least once during the Iteration cycle, and once during the Final Release cycle.

### **Level 4 - Standard Tests**

Test Cases that need to be run at least once during the entire test cycle for this release. These cases are run once, not repeated as are the test cases in previous levels. Functional testing and detailed Design Testing (Functional and Design Test Cases, respectively).These can be tested multiple times for each Milestone Test Cycle (Iteration, Final Release, etc.).

Standard test cases usually include Installation, Data, GUI, and other test areas.

### **Level 5 - Suggested Test**

These are Test Cases that would be nice to execute, but may be omitted due to time constraints.

**Bug Regression:**

Bug Regression will be a central tenant throughout all testing phases. When a Severity 1 bug fails regression, adopters testing team should also put out an immediate email to development. The Test Lead will be responsible for tracking and reporting to development and product management the status of regression testing.

# **TEST CASES**

**Test cases:**

Test Case No.	1
Test Case Action	Checks system behavior when credentials provided by user are correct.
Input	Click on login button by user.
Expected output	Result page which contain information about the user.
Pass/Fail	Pass

Test Case No.	2
Test Case Action	Checks system behavior when credentials provided by admin are correct.
Input	Click on login button by admin.
Expected output	Result page which contain information about only all member.
Pass/Fail	Pass

## Matrimonial Web Application

Test Case No.	3
Test Case Action	Checks system behavior when credentials provided by admin are correct.
Input	Click on generate button by admin.
Expected output	Result page which contain information about only paid member.
Pass/Fail	Pass

Test Case No.	4
Test Case Action	Checks system behavior when credentials provided by admin are correct.
Input	Click on generate button by admin.
Expected output	Result page which contain information about only non-paid member.
Pass/Fail	Pass

## Matrimonial Web Application

Test Case No.	5
Test Case Action	Checks system behavior when credentials provided by admin are correct.
Input	Click on generate button by admin.
Expected output	Result page which contain information about only non-paid member.
Pass/Fail	Pass

Test Case No.	6
Test Case Action	Checks system behavior when credentials provided by the user are not correct.
Input	In Login page user enters incorrect credentials in respected text fields.
Expected output	Login page with message saying that credentials are incorrect.
Pass/Fail	Pass

## Matrimonial Web Application

Test Case No.	7
Test Case Action	Checks system behavior when credentials provided by user are not correct.
Input	If user enters in-correct credentials in respected text fields of register pages.
Expected output	Same register page with alert message
Pass/Fail	Pass

Test Case No.	8
Test Case Action	Checks system behavior when credentials provided by user are incorrect.
Input	User enters incorrect credentials in respected text fields of EmailId.
Expected output	Register page with message saying that credentials are incorrect.
Pass/Fail	Pass

## Matrimonial Web Application

Test Case No.	9
Test Case Action	Checks system behavior when credentials provided by the user are correct.
Input	In quick search page user enters required information for quick search.
Expected output	Display result according to match with information given by user.
Pass/Fail	Pass

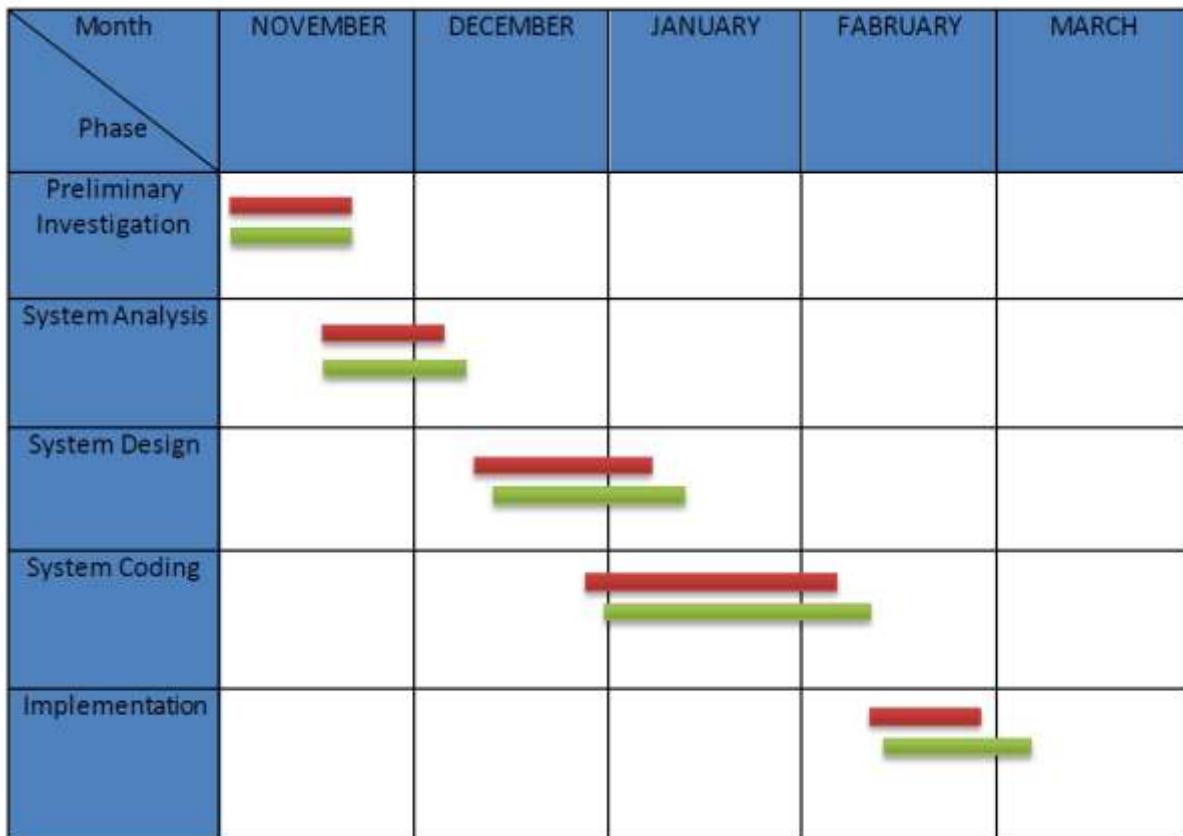
Test Case No.	10
Test Case Action	Checks system behavior when credentials provided by the user are correct.
Input	In make payment page user enters required information for make a payment.
Expected output	Display result according to match with information given by user.
Pass/Fail	Pass

# GANNT CHART

## The Gantt chart

- A Gantt chart is a horizontal bar chart developed as a production control tool in 1917 by Henry L. Gantt, an American engineer and social scientist
- Gantt chart is the Graphical Representation about the progress of the project. Gantt chart shows the details about the how much time taken for any task.
- Gantt chart is a type of bar chart that illustrated the project schedule. It shows the start & finish of the project.
- Gantt chart shows the time taken by the task for each phase. Gantt chart decides the standard of the system. It is a useful tool for planning & scheduling projects.
- Gantt chart contains two columns. First is task and other is month (timing). In the task columns there are five options or phase find in the month's columns shows the progress of the project.

**Testing Phase:**



Planning      Execution

# PERT CHART

## **The Pert Chart**

### **What it is?**

A PERT (Project Evaluation & Review Techniques) chart is a graphic representation of a project's schedule, showing the sequence of tasks, which tasks can be performed simultaneously. The critical path of task that is must be completed on time in order of their project to meet its completion deadline.

The chart can be constructed with a variety of attributes, such as earliest and latest start dates for each task, earliest and latest finish dates for each task, and slack time between tasks.

A PERT chart can document an entire project or key phase of a project.

The chart allows a team to avoid unrealistic timetables and schedule expectations, to help identify and shorten tasks that are bottlenecks, and to focus attention on most critical tasks.

### **When to use it?**

Because it is primarily a project-management tools, a PERT chart is most useful for planning and tracking entire projects or for scheduling and tracking the implementation phase of a planning or improvement effort.

### **How to use it?**

#### **Identify all tasks or project components.**

Make sure the team includes people with firsthand knowledge of the project so that during the brainstorming session all component tasks needed to complete the project are captured. Document the tasks on small note cards.

#### **Identify the first task that must be completed.**

Place the appropriate card at the extreme left of the working surface.

#### **Identify any other tasks that can be started simultaneously with task #1.**

Align these tasks either above or below task #1 on the working surface.

#### **Identify the next task that must be completed.**

Select a task that must wait to begin until task #1(or a task that starts simultaneously with task #1) is completed. Place the appropriate card to the card showing the preceding task.

**Identify any other tasks that can be started simultaneously with task #2.**

Align these tasks either above or below task #2 on the working surface.

Continue this process until all component tasks are sequenced.

### **Identify task durations:**

Using the knowledge of team member, reach a consensus on the most likely amount of time each task will require for completion.

Duration time is usually considered to be elapsed time for the task, rather than actual number of hours/days spent doing the work. Document is duration time on the appropriate task cards.

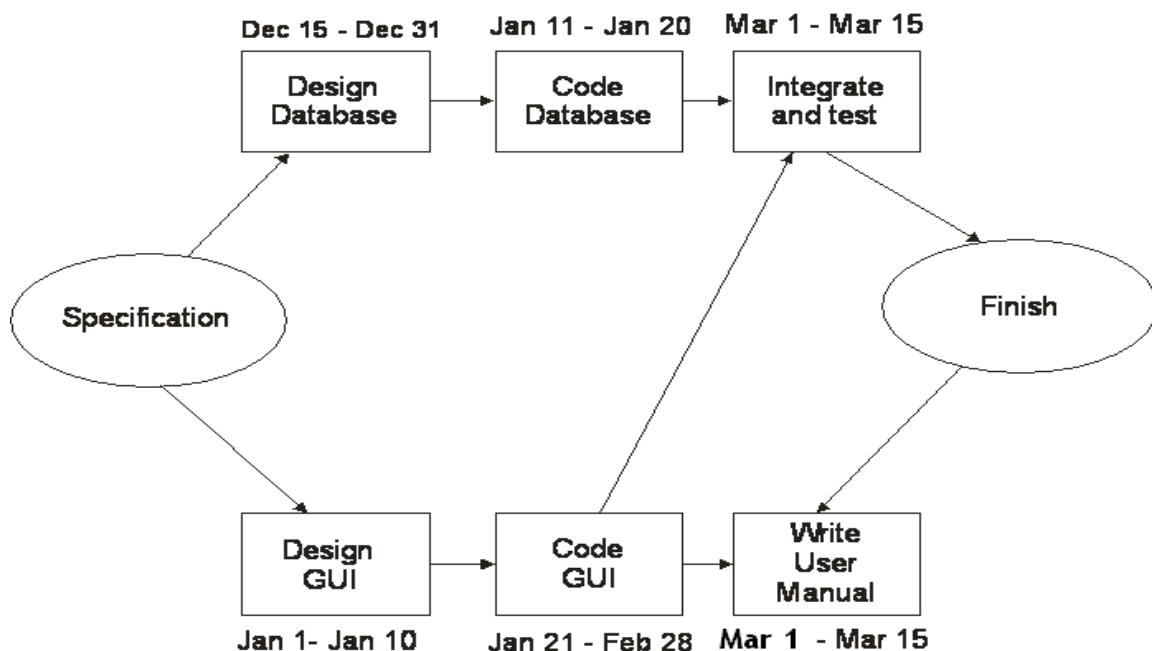
### **Construct the PERT chart:**

Number each task, draw connecting arrows, and task characteristics such as duration, anticipate start date, and anticipated end date.

### **Determine the critical path:**

The project's critical path includes those tasks that must be started or completed on time to avoid delays to the total project. Critical paths are typically displayed in red.

**Note:** most commercially available project management software will routinely generate a PERT chart.



# **REQUIREMENT**

# **SPECIFICATION**

## Requirement Specification

### **Software Specification:-**

<b>Operating System</b>	<b>Windows XP, Windows 7 and 8.</b>
<b>Front End</b>	MS Visual Studio 2008
<b>Back End</b>	MS SQL Server 2008 R2
<b>Web Browser</b>	Google Chrome
<b>Web Server</b>	INTERNET INFORMATION SERVISE (IIS)

### **Hardware Specification:-**

#### **1. Minimum Requirement:-**

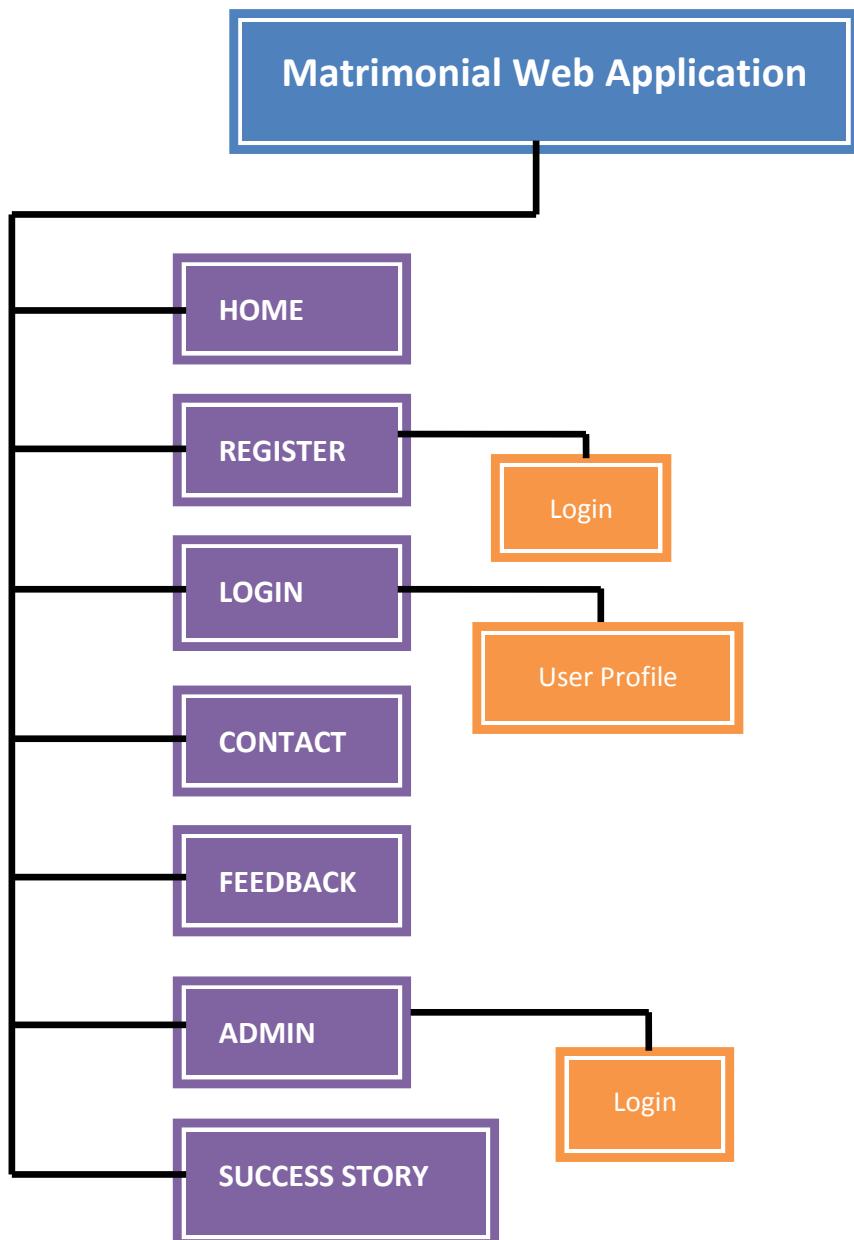
<b>Processor</b>	<b>1.36 GHz</b>
<b>RAM</b>	512 MB
<b>Screen revolution</b>	1024X768 display 5400 RPM Hard disk
<b>Supported Architecture</b>	X86 and x64

#### **2. Maximum Recommended:-**

<b>Processor</b>	<b>2.2 GHz</b>
<b>RAM</b>	1024 MB or more
<b>Screen revolution</b>	1280X1024 display 5400 RPM Hard disk
<b>Supported Architecture</b>	X86 and x64

# **STRUCTURED TREE DIAGRAM**

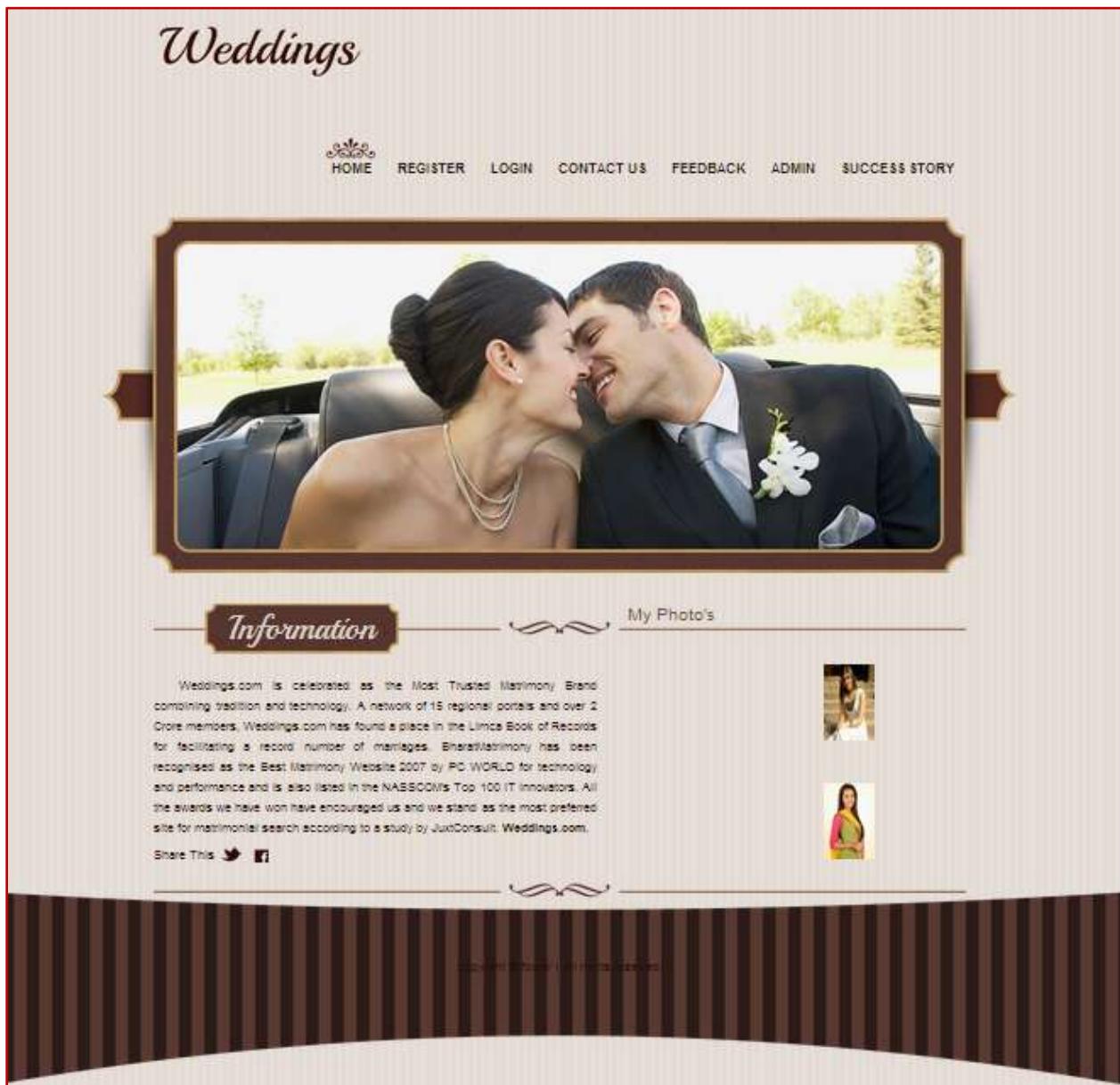
## Structured Tree Diagram:



# **LAYOUT AND CODING**

## 1. Home Page

This is the home page of Weddings.com.



## **Coding:**

```
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass MasterPage
Inherits System.Web.UI.MasterPage
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf
    str1 = "update profile set login_status=0 where pro_id=''"&
Session("#UID#") &"""
    CMD = New SqlCommand(str1, CONN)
    i = CMD.ExecuteNonQuery()

EndSub
EndClass
```

2. Register Page

Here user can register their profile into the weddings.com site.

The screenshot shows the 'Weddings' website's registration form. The page has a light beige background with a subtle striped pattern. At the top center is the word 'Weddings' in a large, elegant, cursive font. To the right of the title is a small decorative logo of a crown or floral emblem. Below the title, there are several navigation links: HOME, LOGIN, CONTACT, and FEEDBACK. On the left side of the main content area, there is a dark grey sidebar containing a 'Register Your Profile' button. The main form consists of various input fields and dropdown menus:

- Name: [Text input field]
- Gender: Male [Dropdown menu]
- Email: [Text input field]
- Profession: [Text input field]
- Mobile No: [Text input field]
- Phone No: [Text input field]
- Address: [Text input field]
- Pin Code: [Text input field]
- Country: AB-Selected-AB [Dropdown menu]
- State: -Selected- [Dropdown menu]
- City: -Selected- [Dropdown menu]
- Profile: Created by Self [Dropdown menu]
- Creator: [Text input field]
- Marital Status: Unmarried [Dropdown menu]
- Status: [Text input field]
- Body Type: Slim [Dropdown menu]
- Physical Status: Normal [Dropdown menu]

## Matrimonial Web Application

Coding:

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass UserControl_Register
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf

    IfNot IsPostBack Then
        Call load_Country()
    EndIf

EndSub

ProtectedSub btnSubmit_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles btnSubmit.Click
Try
    If chkTerm.Checked Then

        Dim strReturn AsString = ""

            str1 = "select count(*) as users from profile where
email_id=' " + Trim(txtEmail.Text).ToString + "'"

            CMD = New SqlCommand(str1, CONN)
            da = New SqlDataAdapter(CMD)
```

## Matrimonial Web Application

```
ds = New DataSet()
da.Fill(ds)

If ds.Tables(0).Rows(0)(0).ToString() <>"0"Then
    ScriptManager.RegisterStartupScript(Me.Page,
Me.[GetType](), "Alert", "alert('This Email ID Already Registered');", True)

    txtEmail.Text = ""
    txtEmail.Focus()

ExitSub
EndIf

str1 = "insert into
profile(profile_creator, names, marital_status, body_type, age, physical_status, height, weight, Mother_tongue, religion, caste, gothram, zodiac, star, eating_habit, drinking_habit, smoking_Habit, country, city, state, Education, occupation, Employeee_id, salary, mobile_no, about_me, require_details, membership, paid_status, start_date, end_date, email_id, pass, images, gender, Profession, address, pincode, Phoneno)
values ('"& cmbprofilecreator.Text &"', '& txtName.Text &"', '&
maritalstatus.Text &"', '& bodutype.Text &"', '& age.Text &"', '&
physicalstatus.Text &"', '& height.Text &"', '& weight.Text &"', '&
mothertongue.Text &"', '& Religion.Text &"', '& caste.Text &"', '&
Gothram.Text &"', '& Zodiac.Text &"', '& Star.Text &"', '& food.Text &"', '&
drinks.Text &"', '& smok.Text &"', '& drpCountry.Text &"', '& cmdcity.Text
&"', '& cmstate.Text &"', '& education.Text &"', '& employeein.Text &"', '&
employeein.Text &"', '& salary.Text &"', '& txtMobile.Text &"', '& abtme.Text
&"', '& expt.Text &"', 'Free', 'Free', Getdate(), Getdate(), "& txtEmail.Text
&"', '& txtPassword.Text &"', '& lblBigImage.Text &"', '& cmbgender.Text
&"', '& txtProfession.Text &"', '& txtAddress.Text &"', '& txtPinCode.Text
&"', '& txtPhone.Text &"')"

CMD = New SqlCommand(str1, CONN)
i = CMD.ExecuteNonQuery()

If i >= 0 Then
    ScriptManager.RegisterStartupScript(Me.Page,
Me.[GetType](), "Alert", "alert('Registered
successfully.');window.location.href='Login.aspx'", True)
Else
    ScriptManager.RegisterStartupScript(Me.Page,
Me.[GetType](), "Alert", "alert('Error while
Registering');window.location.href='Register.aspx'", True)

EndIf

If strReturn = "Thank You For Submitting Your Details with us."Then
    ScriptManager.RegisterStartupScript(Me.Page,
Me.[GetType](), "Alert", "alert('"& strReturn
&"');window.location.href='Register.aspx'", True)
Else
    ScriptManager.RegisterStartupScript(Me.Page,
Me.[GetType](), "Alert", "alert('"& strReturn &"')", True)
EndIf
Else
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('You must agree with terms and conditions.'; ", True)
```

## Matrimonial Web Application

```
EndIf

Catch ex As Exception
EndTry
EndSub

ProtectedSub bigImageUpload_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles bigImageUpload.Click

If fuBig.HasFile Then

Dim strImageNameAsString = Path.GetFileName(fuBig.FileName)
If fuBig.PostedFile.ContentLength < 5024000 Then
Dim strExtensionAsString =
strImageName.Substring(strImageName.LastIndexOf("."c))
Dim strNewImageNameAsString = Guid.NewGuid().ToString() & strExtension
fuBig.SaveAs(Server.MapPath("~/Images/") & strNewImageName)
lblBigImage.Text = strNewImageName
fuBig.Visible = False
bigImageUpload.Visible = False

Else
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Image Size should be less than 5 MB.');", True)
EndIf

Else
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Select a Image.');" , True)
EndIf

EndSub

Sub load_Country()
    str1 = "select Distinct country from zone_master"

Dim cmd AsNew SqlCommand
cmd.Connection = CONN
cmd.CommandType = CommandType.Text
cmd.CommandText = str1
Dim da AsNew SqlDataAdapter(cmd)
Dim ds AsNew DataSet
da.Fill(ds)

drpCountry.DataSource = ds
drpCountry.DataTextField = "country"
drpCountry.DataBind()
EndSub

Sub load_State()
    str1 = "select Distinct state from zone_master where country=''" &
drpCountry.Text & "'"
Dim cmd AsNew SqlCommand
cmd.Connection = CONN
```

## Matrimonial Web Application

```
        cmd.CommandType = CommandType.Text
        cmd.CommandText = str1
Dim da As New SqlDataAdapter(cmd)
Dim ds As New DataSet
da.Fill(ds)

        cmstate.DataSource = ds
        cmstate.DataTextField = "state"
        cmstate.DataBind()
EndSub

Sub load_City()
    str1 = "select Distinct city from zone_master where country=''"&
drpCountry.Text &"' and state=''"& cmstate.Text &"""

Dim cmd As New SqlCommand
    cmd.Connection = CONN
    cmd.CommandType = CommandType.Text
    cmd.CommandText = str1
Dim da As New SqlDataAdapter(cmd)
Dim ds As New DataSet
da.Fill(ds)

        cmdcity.DataSource = ds
        cmdcity.DataTextField = "city"
        cmdcity.DataBind()
EndSub

ProtectedSub drpCountry_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles drpCountry.SelectedIndexChanged
Call load_State()
EndSub

ProtectedSub cmstate_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles cmstate.SelectedIndexChanged
Call load_City()
EndSub
EndClass
```

### 3. Login Page

Through this page user get login to his/her profile page.



Coding:

## Matrimonial Web Application

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass UserControls_Login
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf

    IfNot IsPostBack Then

        If Request.Cookies("#UserEmail#") IsNotNothingAndAlso
        Request.Cookies("#Password#") IsNotNothingThen
            txtEmail.Text =
        Request.Cookies("#UserEmail#").Value.ToString()
            txtpassword.Text =
        Request.Cookies("#Password#").Value.ToString()
        EndIf
        If Request.QueryString("ref") = "bd"Then
            ViewState("PreUrl") = Request.UrlReferrer.ToString()
        EndIf
    EndIf

EndSub

ProtectedSub btnLogin_Click(ByVal sender AsObject, ByVal e As EventArgs)
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
```

## Matrimonial Web Application

```
EndIf

Try
Dim dsLogin AsNew DataSet()

    str1 = "select * from profile where email_id=''" +
Trim(txtEmail.Text) + "' and pass=''" + Trim(txtpassword.Text) + "'"

    CMD = New SqlCommand(str1, CONN)
    da = New SqlDataAdapter(CMD)
    dsLogin = New DataSet()
    da.Fill(dsLogin)
If dsLogin.Tables.Count > 0 Then
If dsLogin.Tables(0).Rows.Count > 0 Then
    Session("#UID#") =
dsLogin.Tables(0).Rows(0)("pro_id").ToString()
    Session("#UNAME#") =
dsLogin.Tables(0).Rows(0)("names").ToString()
    Session("#UEMAIL#") =
dsLogin.Tables(0).Rows(0)("email_id").ToString()
    Session("#GENDER#") =
dsLogin.Tables(0).Rows(0)("gender").ToString()
    Session("#PROID#") =
dsLogin.Tables(0).Rows(0)("pro_id").ToString()
    Session("#paystatus#") =
dsLogin.Tables(0).Rows(0)("paid_status").ToString()

If chkRemember.Checked Then
Dim Usercookie AsNew HttpCookie("#UserEmail#")
    Usercookie.Value = txtEmail.Text.ToString().Trim()
    HttpContext.Current.Response.AppendCookie(Usercookie)

Dim Pwdcookie AsNew HttpCookie("#Password#")
    Pwdcookie.Value = txtpassword.Text.ToString().Trim()
    HttpContext.Current.Response.AppendCookie(Pwdcookie)
EndIf
If ViewState("PreUrl") IsNotNothingThen
    Response.Redirect(ViewState("PreUrl").ToString(),
False)
Else
    str1 = "update profile set login_status=1 where
pro_id=''" & Session("#UID#") & "'"
    CMD = New SqlCommand(str1, CONN)
    i = CMD.ExecuteNonQuery()

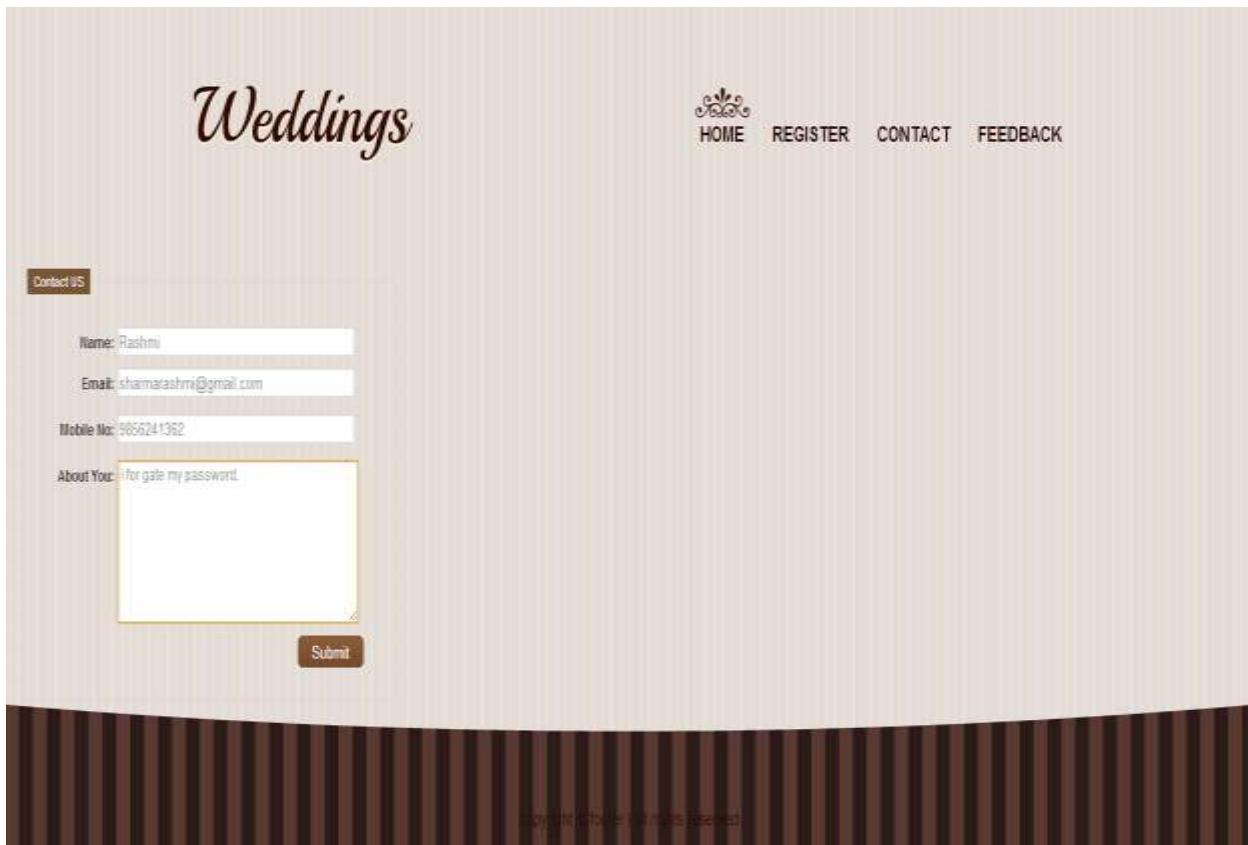
    Response.Redirect("~/Automatchs.aspx", False)
EndIf
Else
    ScriptManager.RegisterStartupScript(Me.Page,
Me.[GetType](), "Alert", "alert('Invalid Email or Password!');", True)
EndIf
Else
```

## Matrimonial Web Application

```
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),  
    "Alert", "alert('Login not Successfull!');", True)  
EndIf  
  
Catch ex As Exception  
EndTry  
EndSub  
  
EndClass
```

## 4. Contact Us

User sends their request to the Administrator.



Coding:

## Matrimonial Web Application

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
Imports System.Text.RegularExpressions
PartialClass UserControls_Contactus
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf
EndSub

ProtectedSub btnSubmit_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles btnSubmit.Click

If txtName.Text = ""Then
ExitSub
EndIf
    str1 = "insert into contactus(names,mobile,email_id,details)
values('"& txtName.Text & "','"& txtMobile.Text & "','"& txtEmail.Text & "','"&
txtAddress.Text & ')"
    CMD = New SqlCommand(str1, CONN)
    i = CMD.ExecuteNonQuery()
    txtName.Text = ""
    If i >= 0 Then
        ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Thank For Submitting your Details We Will Contact to
Soon');window.location.href='Contactus.aspx'", True)
    Else
        ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Error while Saving your
informations');window.location.href='Contactus.aspx'", True)

```

EndIf  
EndSub  
EndClass

## **5 . Feedback**

User sends the feedback according to their experience for this web site



Coding:

## Matrimonial Web Application

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
Imports System.Text.RegularExpressions
PartialClass UserControls_Feeback
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub btnLogin_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles btnLogin.Click

If TextBox1.Text = ""Then
ExitSub
EndIf
str1 = "insert into feedback(names,comments) values ('"& TextBox1.Text
&"', '"& TextBox2.Text &"')"
CMD = New SqlCommand(str1, CONN)
i = CMD.ExecuteNonQuery()
TextBox1.Text = ""
If i >= 0 Then
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Thank you for your
feedback');window.location.href='Feedback.aspx'", True)
Else
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Error while Saving your
informations');window.location.href='Feedback.aspx'", True)
EndIf
EndSub

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
CONN = New
SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())

If CONN.State = Data.ConnectionState.Open Then
    CONN.Close()
Else
```

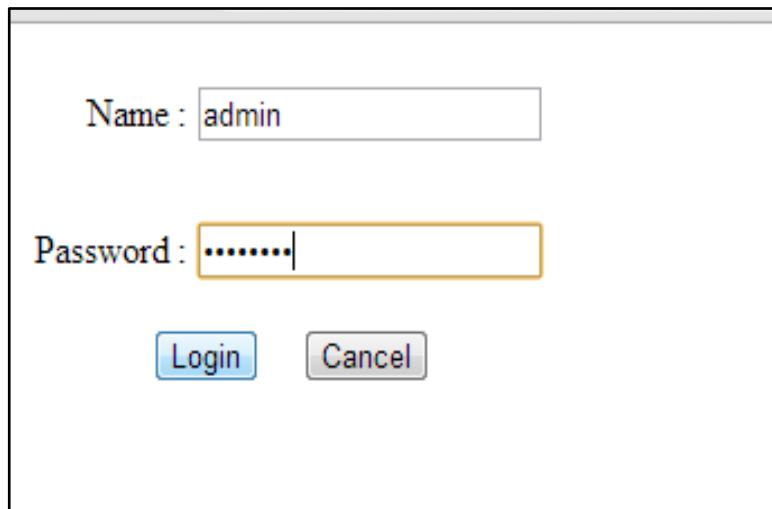
## Matrimonial Web Application

```
    CONN.Open()  
EndIf
```

```
EndSub  
EndClass
```

## 6. Admin Login

Admin get login to enter their account.



Coding:

```
Imports System.Data
```

## Matrimonial Web Application

```
Imports System.Configuration
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
Imports System.Data.SqlClient
PartialClass UserControls_aLogin
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim I As Integer
Dim str1 As String
Dim da As SqlDataAdapter
Dim ds As DataSet

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())

    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf
EndSub

ProtectedSub Button1_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles Button1.Click
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf

    str1 = "select user_nm from admin_login where user_nm='"
    Trim(TextBox1.Text) + "' and pass='"
    Trim(TextBox2.Text) + "'"

    CMD = New SqlCommand(str1, CONN)
    da = New SqlDataAdapter(CMD)
    ds = New DataSet()
    da.Fill(ds)
    If ds.Tables(0).Rows.Count = 0 Then
        ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('User Name Not Exist or incorrect
password');window.location.href='newuser.aspx'", True)

        TextBox1.Text = ""
        TextBox2.Text = ""
    EndIf
EndSub
```

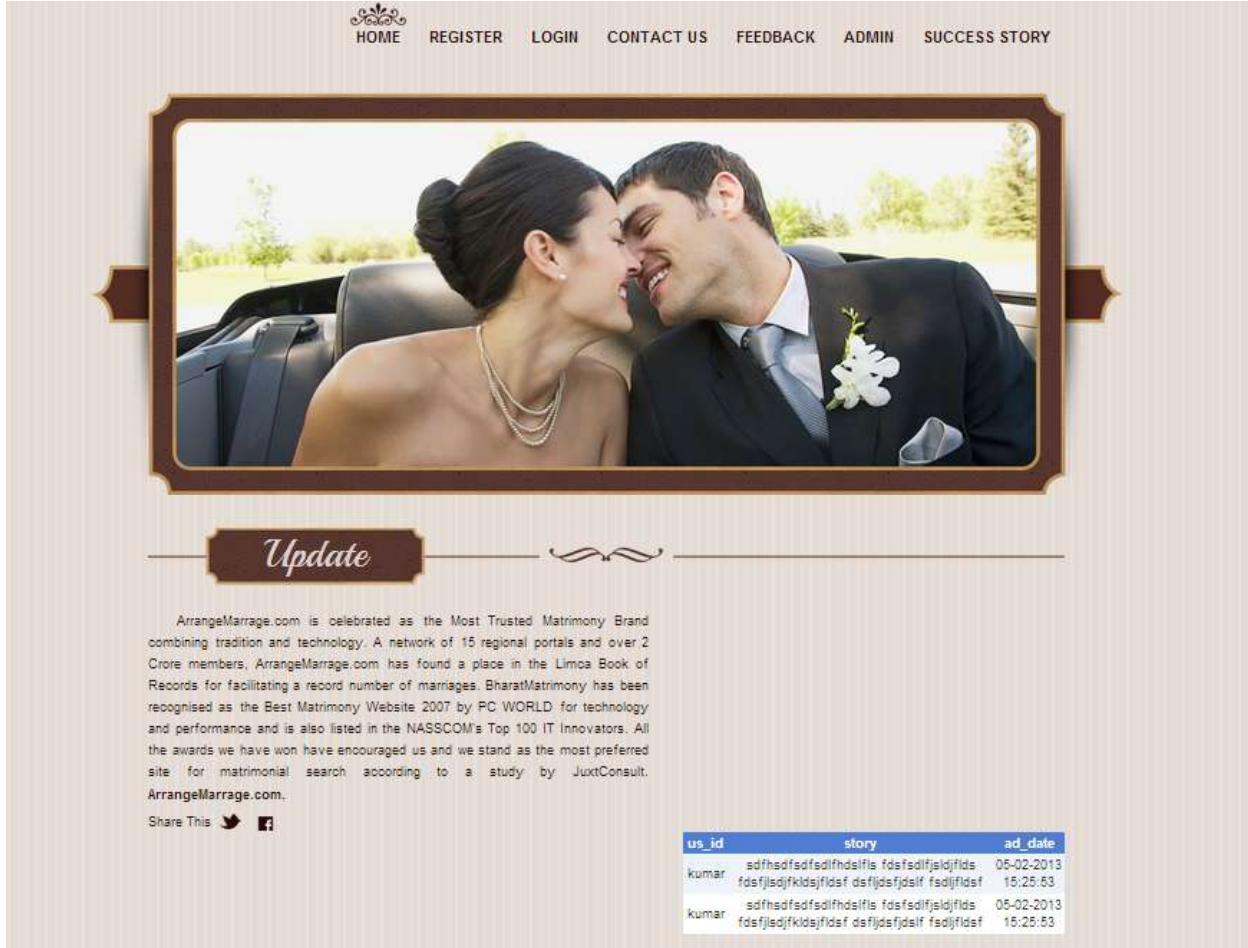
```
    TextBox1.Focus()
ExitSub

EndIf
    Response.Redirect("AdminHome.aspx")
EndSub
EndClass
```

## 7. Success Story

User see their or others success story through this page.

## Matrimonial Web Application



## Coding :

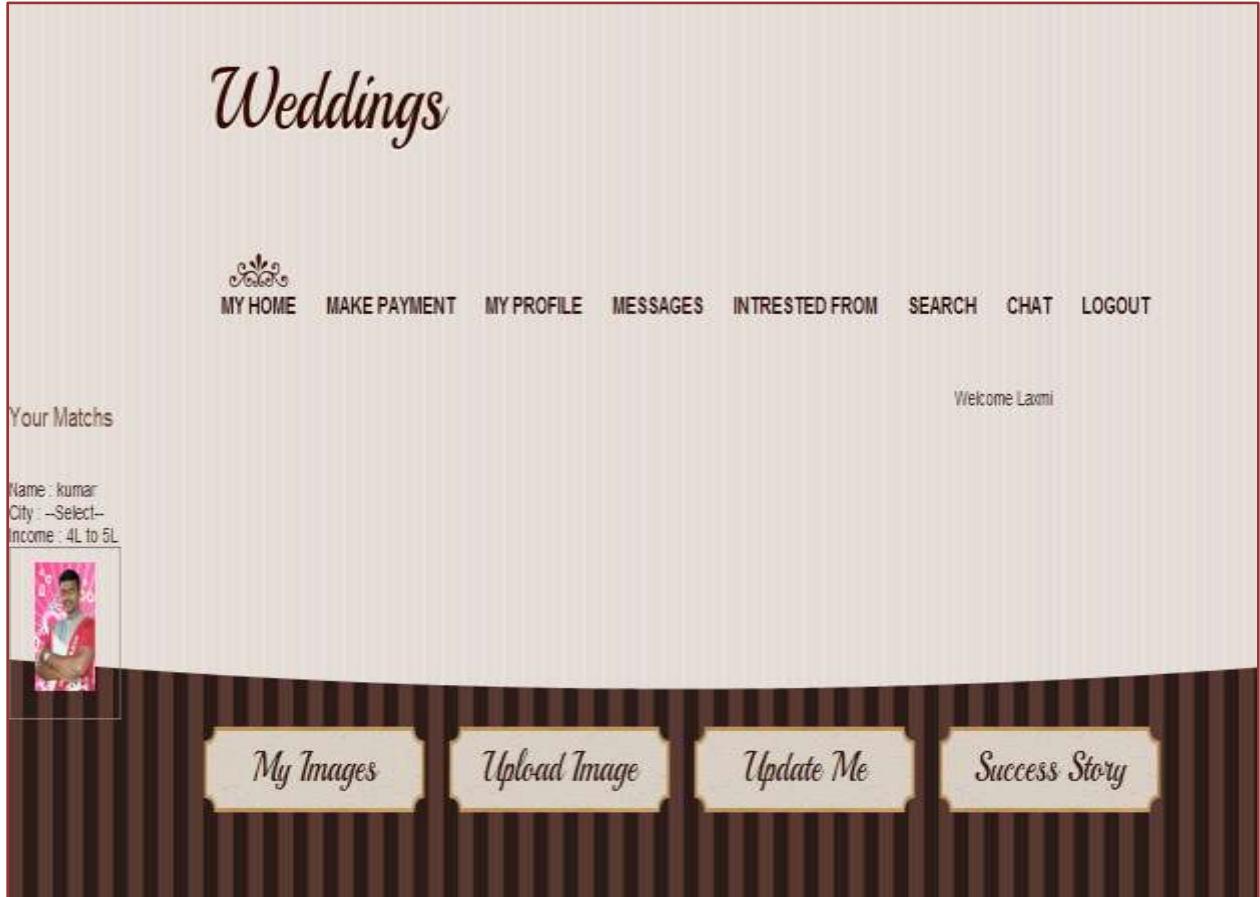
```
Imports System.Data.SqlClient
Imports System
```

## Matrimonial Web Application

```
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass User_Controls_viewstory
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger
ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf
    Call FiilGrid()
EndSub
PublicSub FiilGrid()
Try
    Dim ds1 AsNew DataSet()
        str1 = "select * from success_story"
        CMD = New SqlCommand(str1, CONN)
        da = New SqlDataAdapter(CMD)
        ds1 = New DataSet()
        da.Fill(ds1)
        If ds1.Tables.Count > 0 Then
            GridView1.DataSource = ds1
            GridView1.DataBind()
        EndIf
    Catch ex As Exception
    EndTry
EndSub
EndClass
```

## 8. User's Home Page

This is the user's profile page.



## Coding:

```
Imports System.Data.SqlClient  
Imports System  
Imports System.Data
```

## Matrimonial Web Application

```
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass User_Controls_DashBoard
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim ds1 As DataSet
Dim i AsInteger
Dim caste, occupation, education, salary, matrial, proid, gender AsString

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load

    proid = Session("#UID#").ToString
    gender = Session("#GENDER#")

    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()

    IfNot IsPostBack Then
    IfNot IsPostBack Then
        Call load_details()

    EndIf
    EndIf

    EndIf

EndSub

Sub load_details()
Try

    str1 = "select * from search_setting where pro_id=''"& proid &"'"

    CMD = New SqlCommand(str1, CONN)
    da = New SqlDataAdapter(CMD)
```

## Matrimonial Web Application

```
ds1 = New DataSet()
da.Fill(ds1)
If ds1.Tables.Count > 0 Then
If ds1.Tables(0).Rows.Count > 0 Then
    caste = ds1.Tables(0).Rows(0)("caste").ToString()
    salary = ds1.Tables(0).Rows(0)("salary").ToString()
    education = ds1.Tables(0).Rows(0)("education").ToString()
    occupation =
ds1.Tables(0).Rows(0)("occupation").ToString()
    matrial =
ds1.Tables(0).Rows(0)("Maritalstatus").ToString()
EndIf
EndIf

Dim ds AsNew DataSet()

str1 = "select * from profile where (Education=''" & education & "'"
or salary=''" & salary & "' or caste=''" & caste & "' or occupation=''" & occupation
& "' or marital_status=''" & matrial & "') and gender<>''" & gender & "'"

CMD = New SqlCommand(str1, CONN)
da = New SqlDataAdapter(CMD)
ds = New DataSet()
da.Fill(ds)

If ds.Tables.Count > 0 Then
    rptPromotional.DataSource = ds
    rptPromotional.DataBind()
EndIf

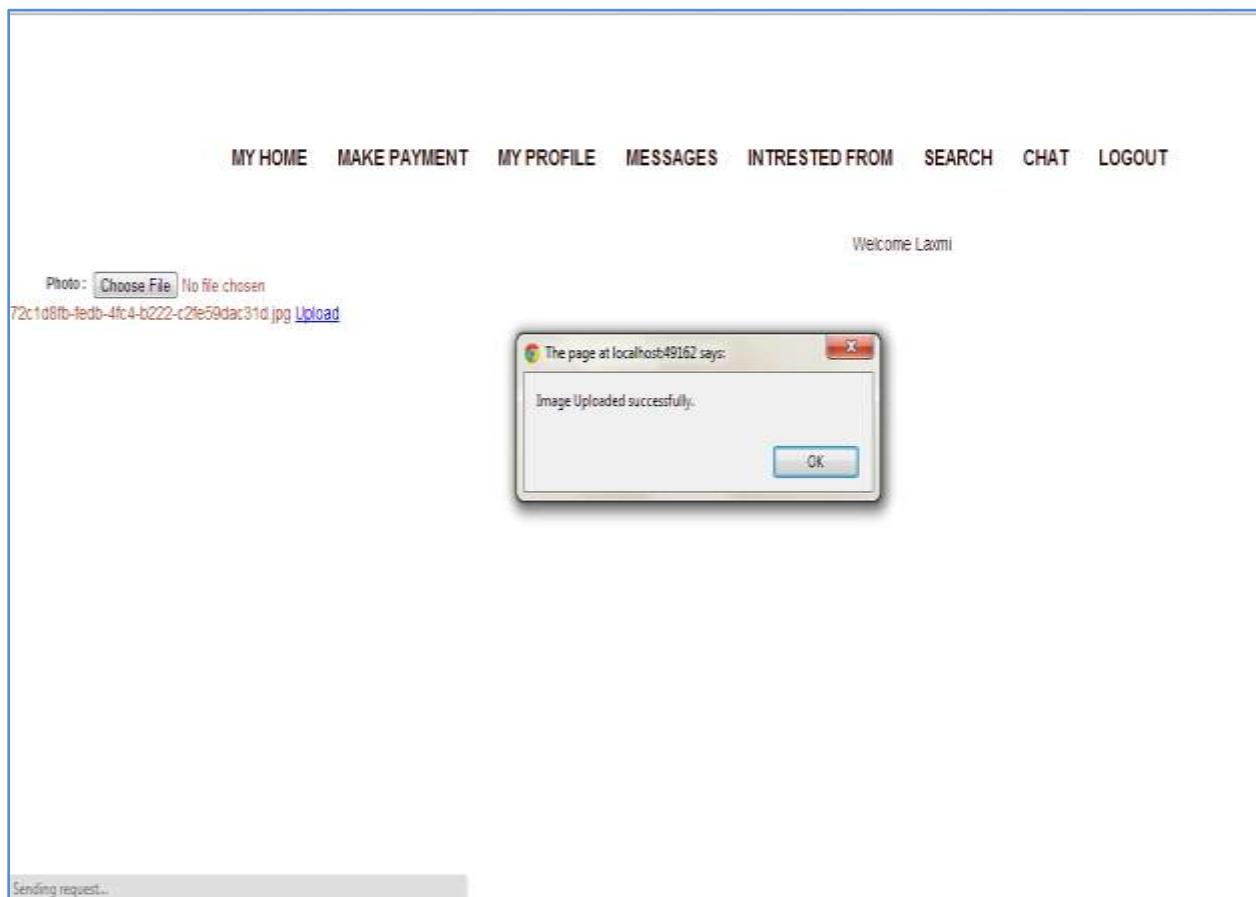
Catch ex As Exception
EndTry

EndSub
EndClass
```

## 9. Upload Images

**Users can upload their images into their profile.**

## Matrimonial Web Application



### Coding:

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
```

## Matrimonial Web Application

```
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass User_Controls_image_upload
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf
EndSub

ProtectedSub bigImageUpload_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles bigImageUpload.Click
If fuBig.HasFile Then

    Dim strImageName AsString = Path.GetFileName(fuBig.FileName)
    If fuBig.PostedFile.ContentLength < 5024000 Then
        Dim strExtension AsString =
        strImageName.Substring(strImageName.LastIndexOf("."c))
        Dim strNewImageName AsString = Guid.NewGuid().ToString() & strExtension
        fuBig.SaveAs(Server.MapPath("~/Images/") & strNewImageName)
        lblBigImage.Text = strNewImageName
    ' fuBig.Visible = False
    ' bigImageUpload.Visible = False

        str1 = "insert into image_gallery(images,pro_id) values ('"&
        lblBigImage.Text &"','" & Session("#UID#") &"')"
        CMD = New SqlCommand(str1, CONN)
        i = CMD.ExecuteNonQuery()

    If i >= 0 Then
        ScriptManager.RegisterStartupScript(Me.Page,
        Me.[GetType](), "Alert", "alert('Image Uploaded
        successfully.');" & window.location.href='Imgeuploader.aspx'', True)
    Else
        ScriptManager.RegisterStartupScript(Me.Page,
        Me.[GetType](), "Alert", "alert('Error while Uploading
        Image');" & window.location.href='Imgeuploader.aspx'', True)
    EndIf
EndSub
```

```
EndIf

Else
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Image Size should be less than 5 MB.');", True)
EndIf

Else
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Select a Image.');" , True)
EndIf

EndSub
EndClass
```

## 10. Search Matches

User can search their match through this search page.



### Coding:

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
```

## Matrimonial Web Application

```
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass User_Controls_Search_Setting
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim prod_id AsInteger
Dim i AsInteger
ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf
    prod_id = Session("#UID#").ToString
EndSub

ProtectedSub btnSubmit_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles btnSubmit.Click
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf

    Try
        Dim dsLogin AsNew DataSet()

        str1 = "select * from search_setting where pro_id=''"& prod_id
        &"'"

        CMD = New SqlCommand(str1, CONN)
        da = New SqlDataAdapter(CMD)
        dsLogin = New DataSet()
        da.Fill(dsLogin)
        If dsLogin.Tables.Count > 0 Then
            If dsLogin.Tables(0).Rows.Count <= 0 Then
                Call SaveSetting()
            Else
                Call UpdateSetting()
            EndIf
        EndIf
    EndSub
```

## Matrimonial Web Application

```
Catch ex As Exception
EndTry

    Session("#Education#") = cmeducation.Text
    Session("#Salary#") = salary.Text
    Session("#Caste#") = caste.Text
    Session("#Occupation#") = cmboccupation.Text
    Session("#marital#") = cmbmarital.Text

EndSub

Sub SaveSetting()
    str1 = "insert into search_setting values ('" & prod_id & "','" &
caste.Text & "','" & salary.Text & "','" & cmeducation.Text & "','" &
cmboccupation.Text & "','" & cmbmarital.Text & "')"
    CMD = New SqlCommand(str1, CONN)
    i = CMD.ExecuteNonQuery()

If i >= 0 Then
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Searching Started');window.location.href='Allmatch.aspx'", True)
Else
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Error while
Searching');window.location.href='Search_Setting.aspx'", True)

EndIf
EndSub

Sub UdateSetting()
    str1 = "update search_setting set caste='" & caste.Text & "',salary=''" &
salary.Text & "',education=''" & cmeducation.Text & "',Occupation=''" &
cmboccupation.Text & "',Maritalstatus=''" & cmbmarital.Text & "' where pro_id=''" &
prod_id & "'"
    CMD = New SqlCommand(str1, CONN)
    i = CMD.ExecuteNonQuery()

If i >= 0 Then
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Searching Started');window.location.href='Allmatch.aspx'", True)
Else
    ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),
"Alert", "alert('Error while
Searching');window.location.href='Search_Setting.aspx'", True)

EndIf
```

EndSub

EndClass

## **11. Chat**

### **Coding:**

```
Imports System.Data.SqlClient  
Imports System
```

## Matrimonial Web Application

```
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
Imports System.Text.RegularExpressions
PartialClass User_Controls_chat_box
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub btnLogin_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles btnLogin.Click

    str1 = "select messages from chat_box where to_uid=''"& TextBox1.Text
    &"'"'

    CMD = New SqlCommand(str1, CONN)
    da = New SqlDataAdapter(CMD)
    ds = New DataSet()
    da.Fill(ds)

    If ds.Tables(0).Rows.Count > 0 Then
        str1 = "update chat_box set messages=messages +'"& TextBox2.Text
        &"' where to_uid=''"& TextBox1.Text &"'"
        CMD = New SqlCommand(str1, CONN)
        i = CMD.ExecuteNonQuery()

    Else
        str1 = "insert into chat_box(to_uid,from_uid,messages) values(''"&
        TextBox1.Text &"','"& Session("#UID#") &"','"& TextBox2.Text &"')"
        CMD = New SqlCommand(str1, CONN)
        i = CMD.ExecuteNonQuery()
    EndIf
    Call load_chat()

EndSub

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
```

```
EndIf
    TextBox1.Text = ""
    TextBox1.Text = Request.QueryString.[Get]("ID")

EndSub

Sub load_chat()
    str1 = "select messages from chat_box where to_uid='"
    str1 = str1 & TextBox1.Text
    str1 = str1 & "' and from_uid='"
    str1 = str1 & Session("#UID#")
    str1 = str1 & "'"
Dim cmd AsNew SqlCommand
    cmd.Connection = CONN
    cmd.CommandType = CommandType.Text
    cmd.CommandText = str1
Dim da AsNew SqlDataAdapter(cmd)
Dim ds AsNew DataSet
    da.Fill(ds)
    TextBox2.Text = ds.Tables(0).Rows(0)("messages").ToString()
EndSub
EndClass
```

## 12. User's Success Story

User can write their success story to share with others.



## Coding:

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
Imports System.IO
Imports System.Web
```

## Matrimonial Web Application

```
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
Imports System.Text.RegularExpressions
PartialClass User_Controls_chat_box

Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub btnLogin_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles btnLogin.Click

If TextBox1.Text = ""Then
ExitSub
EndIf

str1 = "insert into success_story(us_id,story) values('"&
TextBox1.Text & "','"& TextBox2.Text & "')"
CMD = New SqlCommand(str1, CONN)
i = CMD.ExecuteNonQuery()
TextBox1.Text = ""
ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](), "Alert",
"alert('Your Story Addedd
successfully.');" + window.location.href='successstory.aspx'", True)

EndSub

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())

If CONN.State = Data.ConnectionState.Open Then
    CONN.Close()
Else
    CONN.Open()
EndIf
    TextBox1.Text = Session("#UNAME#")

EndSub
```

EndClass

### **13 . Update User**

**Users can update their profile also.**

## Matrimonial Web Application

The screenshot shows a web application interface for registering a profile. At the top, there is a navigation bar with links: MY HOME, MAKE PAYMENT, MY PROFILE, MESSAGES, INTERESTED FROM, SEARCH, CHAT, and LOGOUT. Below the navigation bar, it says "Welcome Laxmi". On the left, there is a sidebar with a "Register Your Profile" button. The main form contains the following fields:

- Name: Laxmi
- Email: lax@gmail.com
- Profession: Software
- Mobile No.: 1111111111
- Phone No.: 256783000
- Address: There
- Pin Code: 400080
- Country: INDIA
- State: -Select-
- City: -Select-
- Profile: Created by Parents
- Creator:
- Body Type: Slim
- Physical Status: Normal
- Religion: Hindu
- Mother: Hindu
- Tongue:
- Caste: Patil
- Gotra: Shiva
- Zodiac: Aries
- Star: Aswini (Asathy)

## Coding:

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
```

## Matrimonial Web Application

```
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass User_Controls_User_Update
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim uname AsString
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf
    uname = Session("#UEMAIL#")
    Call LoadDetails(uname)
    Call load_Country()
EndSub

ProtectedSub btnSubmit_Click(ByVal sender AsObject, ByVal e As System.EventArgs) Handles btnSubmit.Click
Try
    If chkTerm.Checked Then

        Dim strReturn AsString = ""

                str1 = "update profile set email_id='& txtEmail.Text
&', Profession='& txtProfession.Text &', address='& txtAddress.Text
&', pincode='& txtPinCode.Text &', Phoneno='& txtPhone.Text &',
profile_creator='& cmbprofilecreator.Text &', body_type='& bodutype.Text
&', age='& age.Text &', physical_status='& physicalstatus.Text
&', height='& height.Text &', weight='& weight.Text &', Mother_tongue='&
mothertongue.Text &', religion='& Religion.Text &', caste='& caste.Text
&', gothram='& Gothram.Text &', zodiac='& Zodiac.Text &', star='&
Star.Text &', eating_habit='& food.Text &', drinking_habit='& drinks.Text
&', smoking_Habit='& smok.Text &', country='& drpcCountry.Text &', city='&
cmdcity.Text &', state='& cmstate.Text &', Education='& education.Text
&', occupation='& employeein.Text &', Employeed_in='& employeein.Text
&', salary='& salary.Text &', mobile_no='& txtMobile.Text &', about_me='&
```

## Matrimonial Web Application

```
abtme.Text & "", require_details='& expt.Text & ', pass='& txtPassword.Text  
&'' where email_id='& uname & ''  
        CMD = New SqlCommand(str1, CONN)  
        i = CMD.ExecuteNonQuery()  
  
If i >= 0 Then  
        ScriptManager.RegisterStartupScript(Me.Page,  
Me.[GetType](), "Alert", "alert('Details Updated  
successfully.');window.location.href='Myprofile.aspx'", True)  
Else  
        ScriptManager.RegisterStartupScript(Me.Page,  
Me.[GetType](), "Alert", "alert('Error while Updating  
Details');window.location.href='Update_profile.aspx'", True)  
EndIf  
  
If strReturn = "Thank You For Submitting Your Updated Details with us."Then  
        ScriptManager.RegisterStartupScript(Me.Page,  
Me.[GetType](), "Alert", "alert('& strReturn  
&');window.location.href='Myprofile.aspx'", True)  
Else  
        ScriptManager.RegisterStartupScript(Me.Page,  
Me.[GetType](), "Alert", "alert('& strReturn &')", True)  
EndIf  
Else  
        ScriptManager.RegisterStartupScript(Me.Page, Me.[GetType](),  
"Alert", "alert('You must agree with terms and conditions.')", True)  
EndIf  
  
Catch ex As Exception  
EndTry  
EndSub  
  
PublicSub LoadDetails(ByVal strUserEmail AsString)  
    CONN = New  
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").ConnectionString.ToString())  
    If CONN.State = Data.ConnectionState.Open Then  
        CONN.Close()  
    Else  
        CONN.Open()  
    EndIf  
  
    Try  
        str1 = "select * from Profile where email_id='& strUserEmail  
&'"  
        CMD = New SqlCommand(str1, CONN)  
        da = New SqlDataAdapter(CMD)  
        ds = New DataSet()  
        da.Fill(ds)  
        If ds.Tables.Count > 0 Then  
            If ds.Tables(0).Rows.Count > 0 Then  
                txtName.Text = ds.Tables(0).Rows(0).("names").ToString()  
    EndIf
```

## Matrimonial Web Application

```
        cmbprofilecreator.Text =
ds.Tables(0).Rows(0) ("profile_creator").ToString()
            txtMobile.[ReadOnly] = True
            txtEmail.[ReadOnly] = True
            bodutype.Text =
ds.Tables(0).Rows(0) ("body_type").ToString()
            age.Text = ds.Tables(0).Rows(0) ("age").ToString()
            physicalstatus.Text =
ds.Tables(0).Rows(0) ("physical_status").ToString()
            height.Text = ds.Tables(0).Rows(0) ("height").ToString()
            weight.Text = ds.Tables(0).Rows(0) ("weight").ToString()
            mothertongue.Text =
ds.Tables(0).Rows(0) ("Mother_tongue").ToString()
            Religion.Text =
ds.Tables(0).Rows(0) ("religion").ToString()
            caste.Text = ds.Tables(0).Rows(0) ("caste").ToString()
            Gothram.Text = ds.Tables(0).Rows(0) ("gothram").ToString()
            Zodiac.Text = ds.Tables(0).Rows(0) ("zodiac").ToString()
            Star.Text = ds.Tables(0).Rows(0) ("star").ToString()
            food.Text =
ds.Tables(0).Rows(0) ("eating_habit").ToString()
            drinks.Text =
ds.Tables(0).Rows(0) ("drinking_habit").ToString()
            smok.Text =
ds.Tables(0).Rows(0) ("smoking_Habit").ToString()
            drpcCountry.Text =
ds.Tables(0).Rows(0) ("country").ToString()
            cmddcity.Text = ds.Tables(0).Rows(0) ("city").ToString()
            cmstate.Text = ds.Tables(0).Rows(0) ("state").ToString()
            education.Text =
ds.Tables(0).Rows(0) ("Education").ToString()
            employeein.Text =
ds.Tables(0).Rows(0) ("occupation").ToString()
            employeein.Text =
ds.Tables(0).Rows(0) ("Employeed_in").ToString()
            salary.Text = ds.Tables(0).Rows(0) ("salary").ToString()
            txtMobile.Text =
ds.Tables(0).Rows(0) ("mobile_no").ToString()
            abtme.Text = ds.Tables(0).Rows(0) ("about_me").ToString()
            expt.Text =
ds.Tables(0).Rows(0) ("require_details").ToString()
            txtEmail.Text =
ds.Tables(0).Rows(0) ("email_id").ToString()
            txtProfession.Text =
ds.Tables(0).Rows(0) ("Profession").ToString()
            txtAddress.Text =
ds.Tables(0).Rows(0) ("address").ToString()
            txtPinCode.Text =
ds.Tables(0).Rows(0) ("pincode").ToString()
            txtPhone.Text =
ds.Tables(0).Rows(0) ("Phoneno").ToString()
EndIf
EndIf

Catch ex As Exception
```

## Matrimonial Web Application

```
EndTry
EndSub

Sub load_Country()
    str1 = "select Distinct country from zone_master"

    Dim cmd AsNew SqlCommand
        cmd.Connection = CONN
        cmd.CommandType = CommandType.Text
        cmd.CommandText = str1
    Dim da AsNew SqlDataAdapter(cmd)
    Dim ds AsNew DataSet
        da.Fill(ds)

        drpCountry.DataSource = ds
        drpCountry.DataTextField = "country"
        drpCountry.DataBind()
EndSub

Sub load_State()
    str1 = "select Distinct state from zone_master where country=''"&
drpCountry.Text & "'"

    Dim cmd AsNew SqlCommand
        cmd.Connection = CONN
        cmd.CommandType = CommandType.Text
        cmd.CommandText = str1
    Dim da AsNew SqlDataAdapter(cmd)
    Dim ds AsNew DataSet
        da.Fill(ds)

        cmstate.DataSource = ds
        cmstate.DataTextField = "state"
        cmstate.DataBind()
EndSub

Sub load_City()
    str1 = "select Distinct city from zone_master where country=''"&
drpCountry.Text & "' and state=''"& cmstate.Text & "'"

    Dim cmd AsNew SqlCommand
        cmd.Connection = CONN
        cmd.CommandType = CommandType.Text
        cmd.CommandText = str1
    Dim da AsNew SqlDataAdapter(cmd)
    Dim ds AsNew DataSet
        da.Fill(ds)

        cmdcity.DataSource = ds
        cmdcity.DataTextField = "city"
        cmdcity.DataBind()
EndSub

ProtectedSub drpCountry_SelectedIndexChanged(ByVal sender AsObject, ByVal e
As System.EventArgs) Handles drpCountry.SelectedIndexChanged
```

```
Call load_State()
EndSub

ProtectedSub cmstate_SelectedIndexChanged(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles cmstate.SelectedIndexChanged
Call load_City()
EndSub

EndClass
```

## 14. Message History

**Users see the messages which are sending by others for them.**



## Coding:

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
Imports System.IO
```

## Matrimonial Web Application

```
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass User_Controls_message_reci
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load

    CONN = New
SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())
If CONN.State = Data.ConnectionState.Open Then
    CONN.Close()
Else
    CONN.Open()

IfNot IsPostBack Then
IfNot IsPostBack Then
Call load_details()

EndIf
EndIf

EndIf

EndSub

Sub load_details()
Try
Dim ds AsNew DataSet()

        str1 = "select p.pro_Id,Images,names,City,Salary,msg,from_id from
profile p,send_msg where to_id='& Session("#UEMAIL#") &"' and p.pro_id
in(select pro_Id from profile where email_id in(select from_id from send_msg
where from_id<>'& Session("#UEMAIL#") &"'))"

        CMD = New SqlCommand(str1, CONN)
        da = New SqlDataAdapter(CMD)
        ds = New DataSet()
        da.Fill(ds)

If ds.Tables.Count > 0 Then
    rptPromotional.DataSource = ds
    rptPromotional.DataBind()
EndIf
```

```
Catch ex As Exception  
EndTry  
EndSub  
EndClass
```

## **15. Interested From**

**User can see and also send messages when he/she have interested with others profile.**



### Coding:

```
Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration
```

## Matrimonial Web Application

```
Imports System.IO
Imports System.Web
Imports System.Web.Security
Imports System.Web.UI
Imports System.Web.UI.WebControls
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI.HtmlControls
PartialClass User_Controls_express
Inherits System.Web.UI.UserControl
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load

    CONN = New
SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())
If CONN.State = Data.ConnectionState.Open Then
    CONN.Close()
Else
    CONN.Open()

IfNot IsPostBack Then
IfNot IsPostBack Then
Call load_details()

EndIf
EndIf

EndIf

EndSub

Sub load_details()
Try
Dim ds AsNew DataSet()

    str1 = "select p.pro_Id,Images,names,City,Salary,msg,from_id from
profile p,express_interest where to_id='& Session("#UEMAIL#") &' and
p.pro_id in(select pro_Id from profile where email_id in(select from_id from
express_interest where from_id<>'& Session("#UEMAIL#") &'))"

CMD = New SqlCommand(str1, CONN)
da = New SqlDataAdapter(CMD)
ds = New DataSet()
da.Fill(ds)
```

```
If ds.Tables.Count > 0 Then
    rptPromotional.DataSource = ds
    rptPromotional.DataBind()
EndIf

Catch ex As Exception
EndTry
EndSub

ProtectedSub rptPromotional_ItemCommand(ByVal source AsObject, ByVal e As
System.Web.UI.WebControls.RepeaterCommandEventEventArgs) Handles
rptPromotional.ItemCommand

EndSub
EndClass
```

## 16. Paid Member report

Admin generate the list of paid members and non-paid member list also.

## Matrimonial Web Application

The screenshot shows a web-based matrimonial application interface. At the top, there is a navigation bar with links: Logout, Paid Member List, Non-Paid Member List, Password Reset, New User, Feedback, and Contact Us. Below the navigation bar is a date range selector. On the left, it says "From Date" and shows a calendar for January 2013 with dates from 31 to 10 highlighted. In the center, it says "To Date" and shows a calendar for February 2013 with dates from 28 to 10 highlighted. Below the date range selector is a toolbar with icons for back, forward, search, and other functions, followed by the text "Main Report". To the right of the toolbar are zoom controls (100%, 125%, 150%) and the text "Business Objects".

04-03-											
pro	names	religion	Mother to	caste	zodiac	2013 city	state	Education	salary	mobile no	email id
1	kumar	Hindu	Tamil	Patel	Leo	-Select-	-Select-	Graduate	4L to 5L	996956256	shiva@gma

### Coding:

```

Imports System.Data.SqlClient
Imports System
Imports System.Data
Imports System.Configuration

```

## Matrimonial Web Application

```
Imports System.Diagnostics
Imports CrystalDecisions.CrystalReports.Engine
Imports CrystalDecisions.Shared
Imports CrystalDecisions.Web
Imports CrystalDecisions.ReportSource
PartialClass paid_rpt
Inherits System.Web.UI.Page
Dim CONN As SqlConnection
Dim CMD As SqlCommand
Dim str1 AsString
Dim da As SqlDataAdapter
Dim ds As DataSet
Dim i AsInteger

ProtectedSub Page_Load(ByVal sender AsObject, ByVal e As System.EventArgs)
HandlesMe.Load
    CONN = New
    SqlConnection(ConfigurationManager.ConnectionStrings("MATRIMONIAL").Connectio
nString.ToString())
    If CONN.State = Data.ConnectionState.Open Then
        CONN.Close()
    Else
        CONN.Open()
    EndIf
EndSub

ProtectedSub Button1_Click(ByVal sender AsObject, ByVal e As
System.EventArgs) Handles Button1.Click
Dim str1 AsString

    str1 = "select * from Profile where paid_date between '"&
Calendar1.SelectedDate &"' and '"& Calendar2.SelectedDate &"' and
paid_status<>'Free' order by pro_id"

    CMD = New SqlCommand(str1, CONN)
    da = New SqlDataAdapter(CMD)
    ds = New DataSet()
    da.Fill(ds)

    Session("dtable") = ds.Tables(0)

    Dim dt AsNew DataTable
        dt.Clear()
        dt = CType(Session("dtable"), DataTable)
    If (dt.Rows.Count > 0) Then

        Dim RptDoc AsNew ReportDocument()
            RptDoc.Load(Server.MapPath("~/paid_rpt.rpt"))
            RptDoc.SetDataSource(dt)
            CrystalReportViewer1.ReportSource = RptDoc
            CrystalReportViewer1.DataBind()

    EndIf
EndSub
EndClass
```



# **MAINTENANCE**

## **Maintenance**

**System Maintenance:-**

The last part of system development life cycle is system maintenance, which is actually the implementation of the post implementation review plan. The maintenance can be classified as corrective, adaptive, perfective and prevention.

**Corrective Maintenance:-**

Corrective maintenance changes the software to correct defects. Means repairing, processing or performance failures or making alteration because of previously defined problem.

**Adaptive Maintenance:-**

Adaptive maintenance results in modification to the software to accommodate changes to its external environment.

**Perfective or Enhancement Maintenance:-**

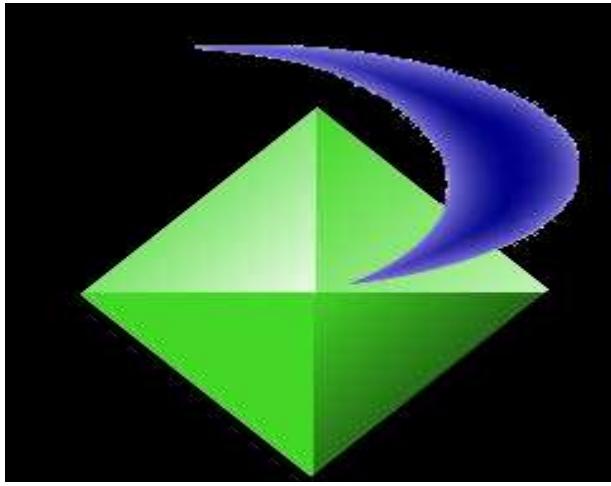
Enhancing the performance or modifying the program according to the user's additional or changing needs. Perfective maintenance extends the software beyond its original functional requirement.

**Presentation Maintenance:-**

Computer software deteriorates due to change, and because of this, preventive maintenance must be conducted to enable the software to serve the needs of its users.

# **REPORTS**

**CRYSTAL REPORTS:**



Crystal reports offers a new, comprehensive web reporting solution to access, analyze and report on data. We can view, create or distribute reports or integrate them into applications. Crystal Report is a powerful tool designed to help everyone analyze and interpret information that is important to us.

- Data source supported are MS Access, SQL server, Oracle etc.
- Reports can be distributed in variety of formats including the web.
- It offers complete control over report formatting.
- It is powerful analytical tool to make a sense of data.

Generating MIS reports was a basic user requirement. Thus we decided to use crystal reports in our project so as to view data in desired formats. We have developed a crystal report for generating the MIS reports that will be used by the Middle Management.

**Purpose of Report:**

Reports are management tools. Their purpose is to help you quickly grasp the essential elements and relationships found in raw data, to help you make effective decisions. For a report to be effective, it has to present the correct data in a logical way. If it presents the

wrong data or if it presents a right data in haphazard manner the report may slow the decision making process or May even encourage incorrect decisions. So we have used Crystal Reports to support the decision making system relating to the store inventory. Since each user of the report has different interests, it is important to plan the report so that it includes the information each user is looking for.

# **CONCLUSION**

## **Conclusion:**

Matrimonial Web Application is to provide Grooms and Brides with excellent matchmaking experience by exploring the opportunities and resources to meet true potential partner.

Matrimonial website which is provide platform to a lot of Bride/Groom for finding perfect match. There are different sectors like

## **Matrimonial Web Application**

Registration, Partner, Search, etc. So the Bride/Groom can get their interest for find their partner. Bride/Groom can directly search Partner according to their required criteria. The Bride/Groom can use match By Email functionality so he/she can get directly E-mail alert for the match which fulfill their required criteria. It helps the user by providing profiles of perspective "Bride" or "Groom" and other information regarding them online.

Matrimonial web application provides facility like quick tour. This is a module that contains the flow of the website .Here user can have a idea how he can commit himself in the website.

Matrimonial web application provides facility to change preference about partner.

This application provide facility like edit profile, update photo and delete photo, hide profile, create album, send express interest, send personal message, apply for loan to the user.

# **FUTURE EXTENSION**

## **Future Extensions:**

- It is possible to provide the web space to the users for creating his portal.
- It is possible to create our own mail server.

- It is possible to create chat server so that user can communicate with each other.
- It is possible to provide facility like create video album.

# BIBLIOGRAPHY

## **Bibliography :**

### **Books**

- ASP.NET (Black Book).
- Professional ASP.NET (Wrox Publication).

## **Matrimonial Web Application**

- C# Vijaymukhi.
- ASP.NET Complete Reference.
- Software engineering Concepts By Roger S.Presman
- UML IN A NUTSHELL By Alhir
- Fundamentals of Software Engineering By Rajib Mall
- SQL Server 2005 (Wrox Publication) .

## **Web Sites**

- [www.bharatmatrimonial.com](http://www.bharatmatrimonial.com)
- [www.shadi.com](http://www.shadi.com)
- [www.jeevansathi.com](http://www.jeevansathi.com)
- [www.patelvivah.com](http://www.patelvivah.com)
- [www.simplymary.com](http://www.simplymary.com)