Happily Married

A
Project Report
Submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Engineering
In
Computer Science & Engineering

Submitted to
RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA,
BHOPAL (M.P.)

Guided by
Prof.Kalash Chandra Bandhu

Submitted By
Rajat Gupta
0875CS131113

Head

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ACROPOLIS TECHNICAL CAMPUS,
INDORE (M.P.) 452001
2015-2016

Declaration

I hereby declared that the work, which is being presented in the project entitled "**Title of the project**" partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering in Computer Science & Engineering**, submitted in the department of Computer Science & Engineering at **Acropolis Technical Campus, Indore** is an authentic record of my own work carried under the supervision of "**Prof./Mr./Ms./Dr. Project Guide**". I have not submitted the matter embodied in this report for award of any other degree.

<Student Name>

<Roll No>

Prof./Mr./Ms./Dr. Project Guide

Supervisor

Acropolis Technical Campus

Deparment of Computer Science & Engineering

Certificate

The project work entitled “Title of The Project” submitted by Name of the student (Enrollment No) is approved as partial fulfillment for the award of the degree of Bachelor of Engineering in Computer Science & Engineering by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.).

Internal Examiner	**External Examiner**
Name:	Name:
Date:	Date:

Acknowledgement

Student Name

Enrollment No

Abstract

Table of Content

Chapter 1

Introduction

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.

“Happily Married” is a desktop based matrimonial application designed to work on windows platform. As the name suggests, it is a social media network through which people looking for a partner to get married can find their perfect life partner. ‘Happily Married’ is a web application designed to provide a platform to such love seekers. A person can make his/her own profile and update it with all the useful information that others might be looking for in their future Mr. or Ms. Right.

1.1 Rationale

We chose this project because a wedding is the most important step in anyone’s life and there should not be any mistakes in selecting a life partner that not only understands you but also trusts you. And we wanted that people who are looking for someone special can find him/her through our web application, knotting two people in a relation is as beautiful as it gets. And since our project is about someone’s life decision we wanted to make sure that there comes no error or fakeness in between so we have put up some verification steps so that fake people cannot join our network and damage/hurt others.

1.2 Goal

The goal of the project is to put in place new solutions for marriages, and to change the face of relations. The interactive and easy to use interface of our web application stands on all the end user expectations. The purposes of the Matrimonial Web Appilcation are:

a) The main purpose of this application is to facilitate matchmaking business by applying the information in the field.

b) It helps the user by providing profiles of perspective “Bride” or “Groom” and other information regarding them online.
c) User can get information regarding their dream life partner at his/her home at his/her convenience.
d) 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.
e) Since internet is a pivot for modern business, our project which is based on internet paves a path for modernization in trade.

1.3 Objective

The objective of the matrimonial website is to develop a web portal and propose options through which users who are looking for partners for marriage can submit their profile and find perfect match for marriage.

In present trend most of the work is done through online like booking, buying goods, money transactions..etc. Implementing online website for finding match is very helpful . In present situation there are many websites like bharath matrimony, shaadi..etc which are helping users to find matches. To do this it requires to:

(a) Review and study different working methods of systems
(b) Find out difference between different types of approaches that can be taken.
(c) Propose a faster solution for database access and updation.

1.4 Methodology

Rational unified process. How it is been taken to achieve your goals and objectives. Activities, milestones, checkpoints etc.

1.5 Role

Rajat Gupta (Front End Designer)

The designing of the ‘Happily Married’ web application is done by Rajat Gupta. He has handled the complete designing of the website with help of other team mate and their suggestions. The complete designing is done on MS Visual Studio 2015 with ASP.NET

as the programming language on the aspx page (Active Server Page Extension), CSS and HTML.

Pranshur Dubey (Database Administrator)

The database is completey handled by Pranshur Dubey alone under the guidance of Prof. Kailash Chand Bandhu. The database is handled on SQL Server 2008 R2 with the backend coding on .aspxcs (active server page extension c#) with C# as the programming language.

Nayan Dubey (Content Editor)

He took up the responsibility of collecting all the information and come up with new ideas. Photos, data, information, searching about project related data, he is the person to call.

Mohini Patel

She took up the responsibility of information gathering of all enteries in the database. . Search all about the database and help pranshur dubey in his work.

Contribution of Project

1.5.1 Market Potential

This application is what a community, society, religion needs right now to understand the importance of marriage and its changing way.

1.5.2 Innovativeness

The work of the application is to generate an attractive matrimonial profile.

1.5.3 Usefulness

The software is useful in reducing the stress on parents and the person himself/herself by reducing the need for going here and there for meeting different people.

1.6 Report Organization

The remaining section of the report is structured as follows. Chapter 2 provides detailed business and technical requirements. Different steps taken for information collection are also discussed in this chapter. Chapter 3 provides details analysis…….. and so on.

Chapter 2

Requirement Engineering

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to mold user expectations to fit the requirements.

Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

2.1 Requirement Collection

2.1.1 <Collection Type 1>

2.1.2 <Collection Type N>

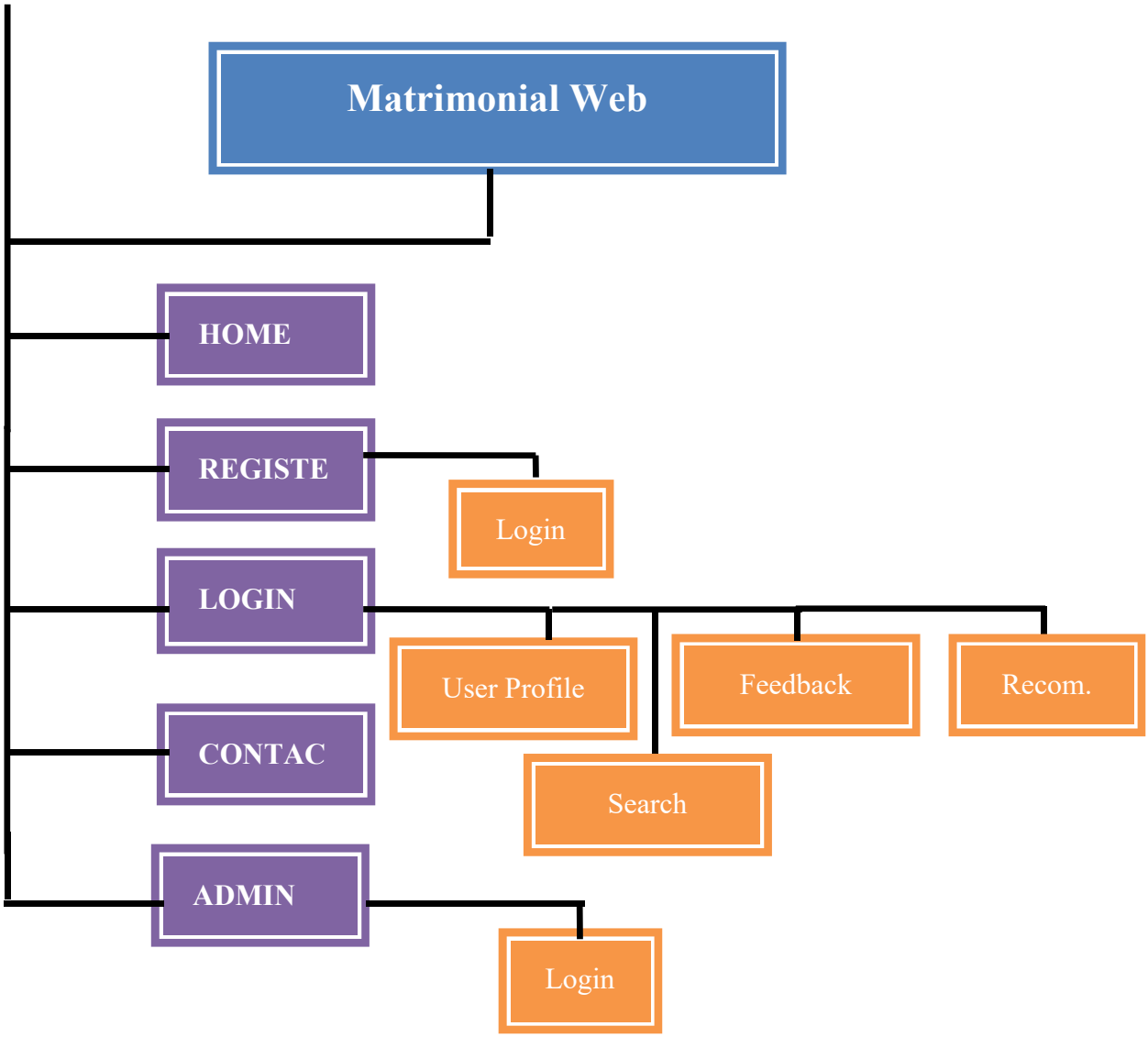
2.2 Requirements

BR 1:

BR N:

TR 1:

TR N:


Matrimonial Web
HOME
REGISTE
Login
LOGIN
User Profile
Feedback
Recom.
CONTAC
Search
ADMIN
Login

Chapter 3

Analysis & Design

3.1 Use-case Diagrams

<some description about functionalities of your system> <set outline of this image canvas to No outline from format Tab>

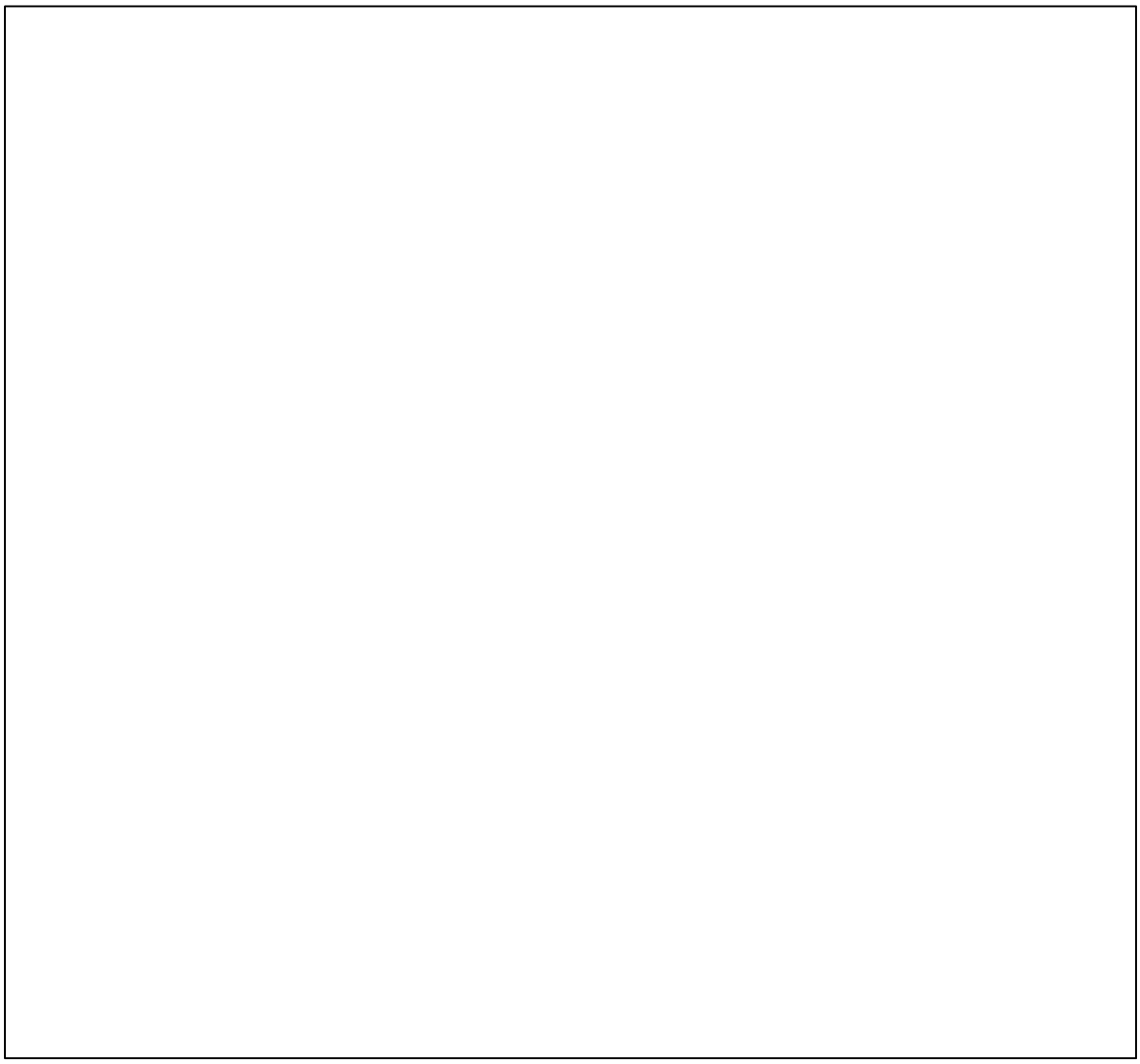


Figure 3.1: Use-case Diagram of <<>>

3.2 Activity Diagrams

Similar case with all diagrams as explained for 3.1

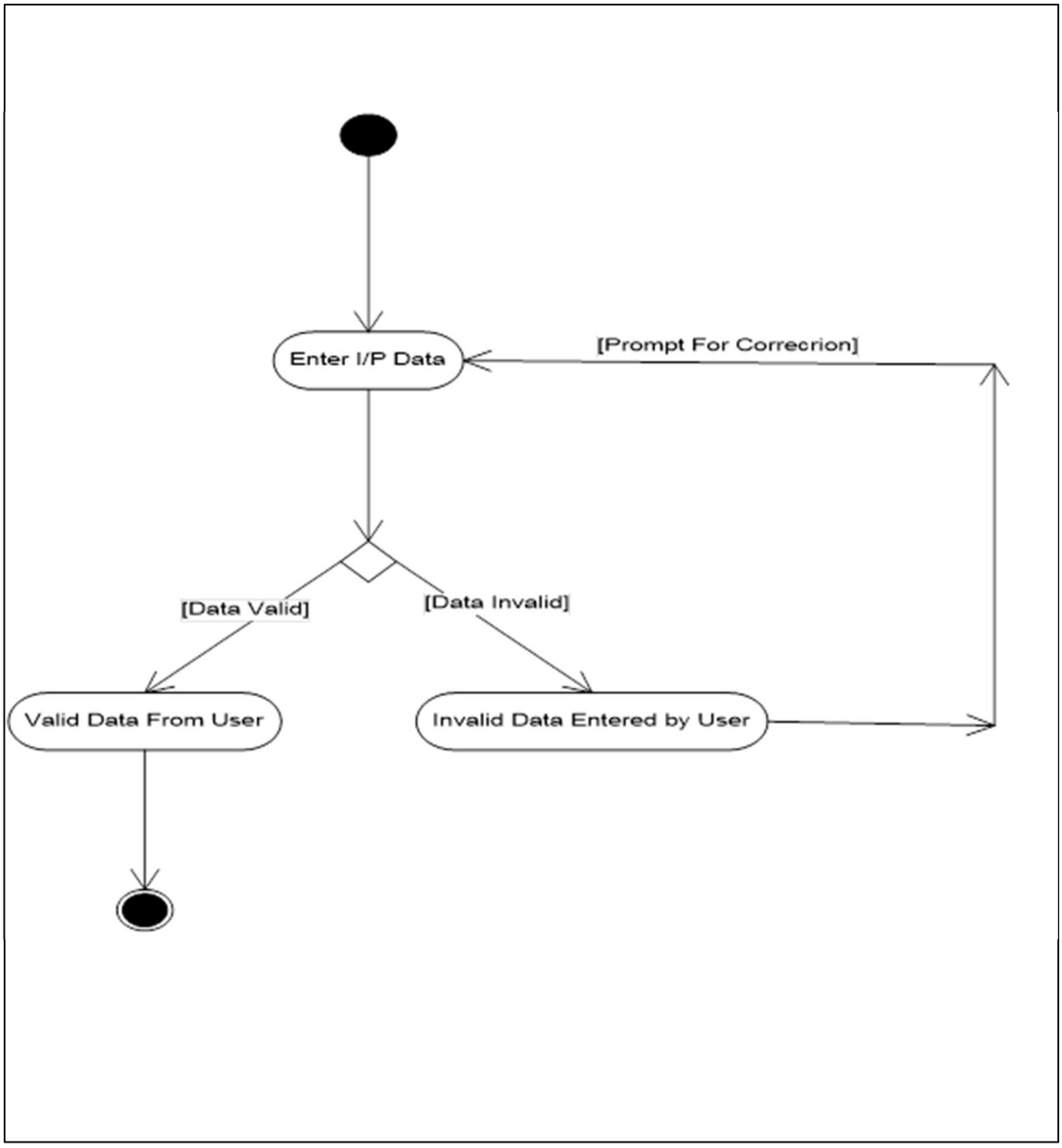


Figure 3.2: Activity Diagram of the input validation

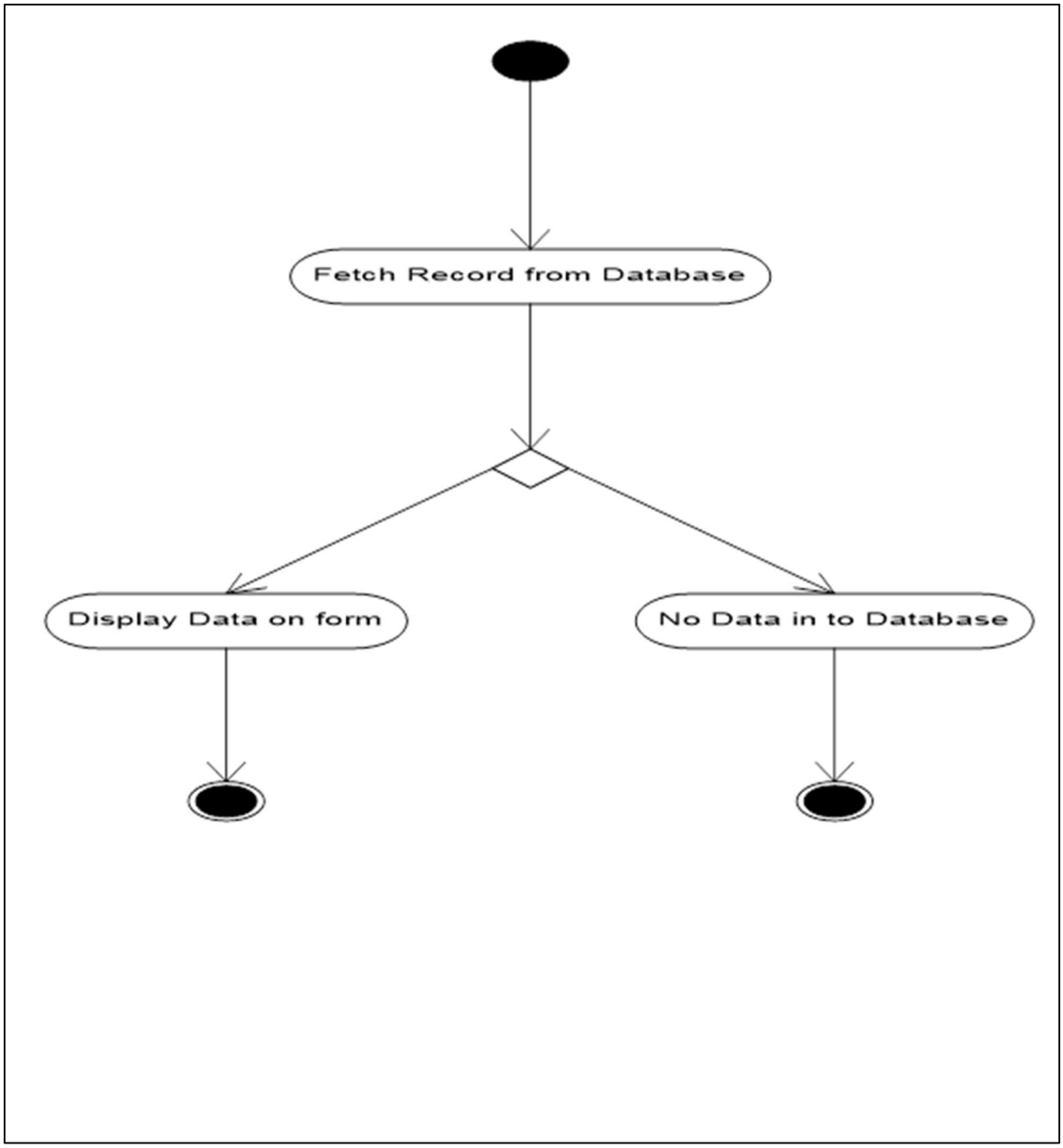


Figure 3.2.1: Activity Diagram of the display Records

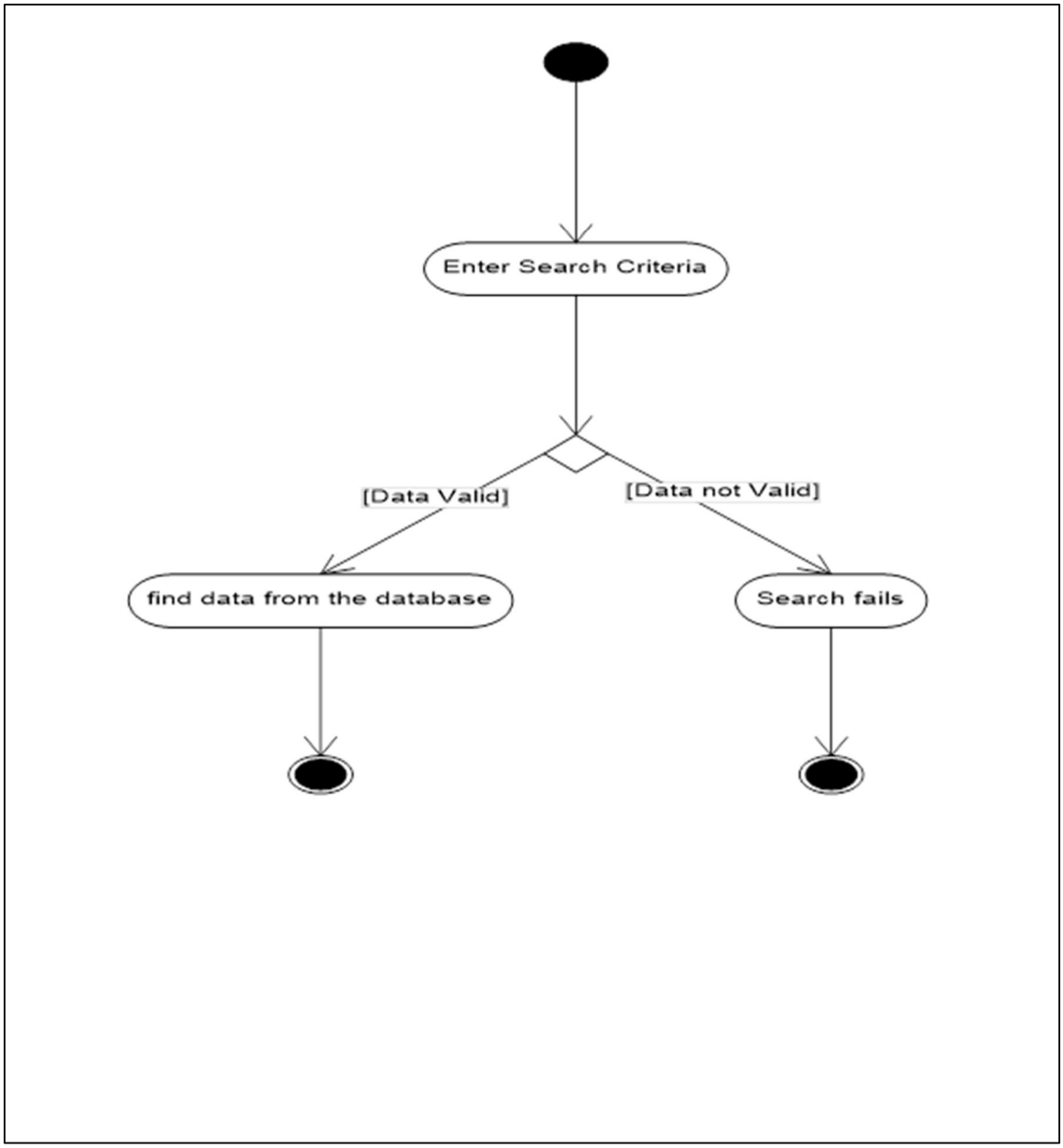


Figure 3.2.1: Activity Diagram of the Search Records

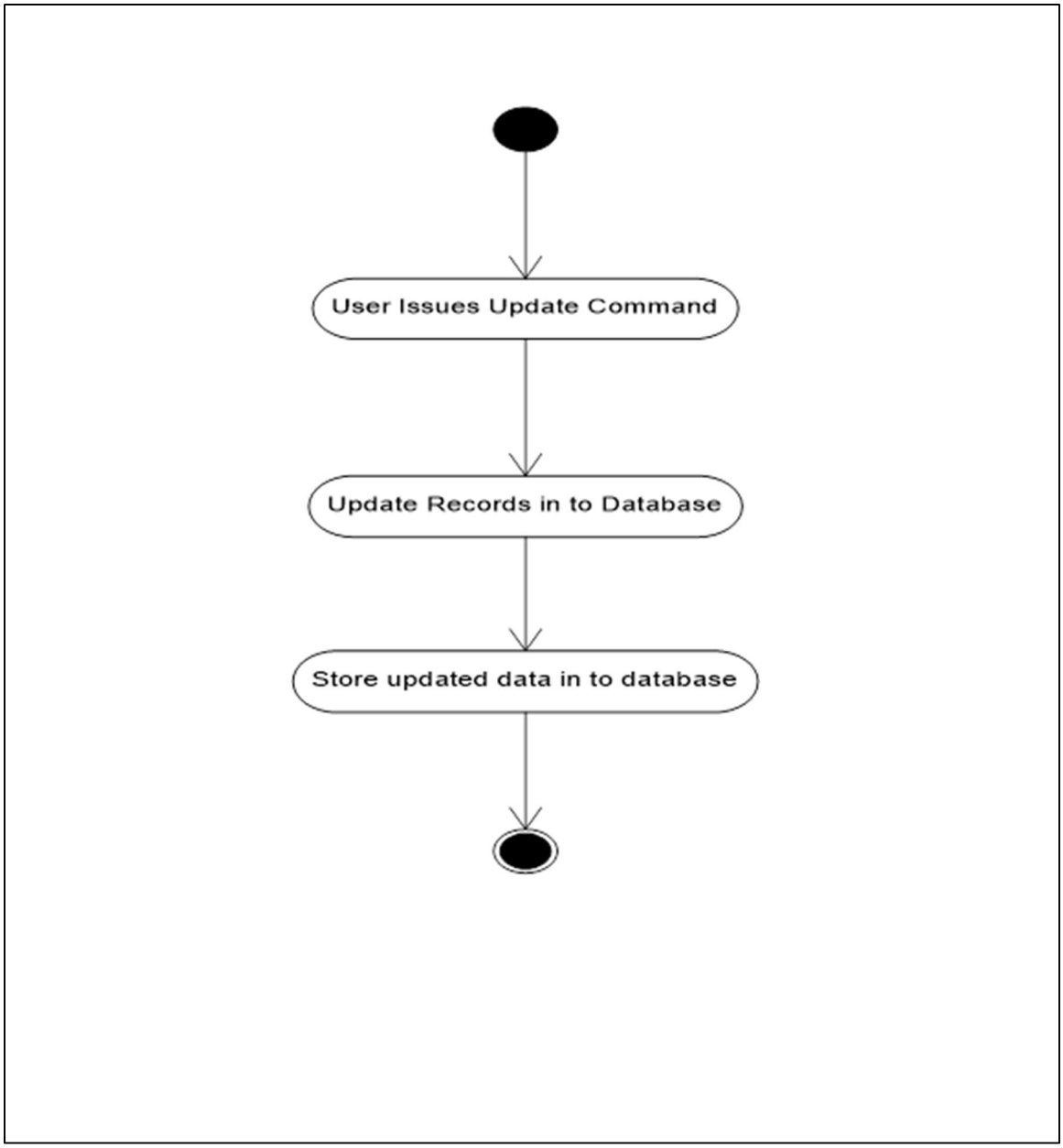


Figure 3.2.1: Activity Diagram of the Update Records

3.3 Sequence Diagrams

3.4 Class Diagrams

3.5 Data Design

3.5.1 Schema Definitions

Table 3.1: Schema for <<>>

3.5.2 Integrity Constraints

3.6 System Architecture

Chapter 4

Construction

4.1 Implementation

4.1.1 Implementation Details

4.1.1.1 Software Details

Operating System	**Windows XP, Windows 7, Windows 8.1 and 10.**
Front End	MS Visual Studio 2015
Back End	MS SQL Server 2008 R2
Web Browser	Google Chrome
Web Server	INTERNET INFORMATION SERVISE (IIS)

4.1.1.2 Hardware Details

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

4.2 Testing

4.2.1 White Box Testing

White-box testing sometimes called glass-box testing, is a test case design method that users the control structure of the procedural design to drive the test case. Always we are thinking that there is no necessary to execute or checks 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 works 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.

Test Case: 1

Table 4.1: Test Case for <<>>

Test Case: N

4.2.2 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 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

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

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

Test Case No.	7
Test Case Action	Checks system behavior when credentials provided by userthe 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

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

Conclusion & Future Works

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 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 Works:

a. It is possible to provide the web space to the users for creating his portal.
b. It is possible to provide chatting feature.
c. It is possible to create our own mail server.
d. It is possible to create chat server so that user can communicate with each other.
e. It is possible to provide facility like create video and photo album.

Appendix A

<<Snapshots of the project>>