ARGS

Automatic Report Generating System

By

NC Muhammad Ali Azhar

NC Syed Danish Azeem

Maj. Shehryar Niazi

Submitted to the Faculty of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology, Islamabad in partial fulfillment forthe requirements of a B.E Degree in Computer Software Engineering

June 2013

CERTIFICATE

Certified that the contents and form of project report entitled **“Automatic Report Generating System”** submitted by 1) Muhammad Ali Azhar, 2) Danish Azeem, and 3) Shehryar Niazi have been found satisfactory for the requirement of the degree.

Supervisor: _____

Col Naveed Sarfaraz Khattak

ABSTRACT

The information which is most critical to a business success lies in the reports. These reports, numbering from dozens to thousands, are typically generated in a manual fashion by a group of analyst and administrators.These administrative personals are highly educated and trained professionals at high posts who have quite an experience in their field. This is a very time consuming and tedious task for professionals who can spend their useful time in their fields of interest and gain some productive results. Having said that, how tedious and unproductive this task might be, it can't be given to any person other than the administrative heads', as these reports are very critical to a business success and only administrative heads can judge their subordinates and write pen picture of them.

"Automatic Report Generating System" is the solution to the above problem which keeps the task of report writing at the hands of administrative heads and saves their precious time by a providing an application which evaluates a person on the basis of feedback form and generate a report in natural language (English). The input Automatic Report Generating System requires is a feedback of a person and user can select the qualities on which he wants to generate a pen picture of that person and he gets a complete well-formed syntactically correct pen picture of that person. First it analyzes the feedback form. User selected qualities will be analyzed and will be graded by the analyzer as excellent, above average, high average, average and poor. Analyzer sends its results to language generator which will generate a report. Language database, consisting of hundreds of definitions for different quality traits, is used by the report generator to join these definitions and present them in a valid sentence structure. These sentences shall collectively form a report and can be imported into a Microsoft Word Document.

This application proved to be very time saving and make the task of writing reports easier for administrators and free them for more productive tasks.Various Testing and evaluation results conducted on the product are extremely promising.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our parents, without whose unflinching support and unstinting cooperation,

a work of this magnitude would not have been possible

ACKNOWLEDEMENTS

There is no success without the will of ALLAH. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality.

We are also grateful to our parents for their unwavering faith in us, their continuous support and love without which we would not have been able to succeed.

We would like to thank our supervisor Col Naveed Sarfaraz Khattak from MCS who in addition to providing valuable technical help and guidance also provided us moral support and encouraged us throughout the development of the project. We are highly thankful to all of our teachers and staff of MCS who supported and guided us throughout our course and research work. Their knowledge, guidance and training enabled us to carry out this work.

In the end we would like to acknowledge the support provided by all our friends, colleagues and a long list of well-wishers whose prayers and faith in us propelled us towards our goal.

Table of Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Introduction

The main objective of the Automatic Report Generating system is to generate automatic report of an entity on the basis of the feedback form. This system is intelligent and evaluates any entity on the basis of feedback form and generates a report in natural language(English). It is developed specific to Military College of Signals but it will evolve into a generic and dynamic system which any organizations may amend and use according to their own requirements.

1.2 Background

At present organization's administration has to observe and write reports of their subordinates in a manual fashion. These administrative personals are highly educated and trained professionals at high posts who have quite an experience in their field. In Military College of Signals, Head of Department (HoD) is assigned to perform this task. But it does not stop at HoD level, it makes it way all way to the top going through Chief Instructor, Dean, Commandant and Rector NUST. All of them have to write remarks. This is a very time consuming and uncreative task for the Doctors and Professors at already mentioned posts.

This issue is not limited to Military College of Signals. Almost all the organizations around the world, somehow or another, is facing problems like this. Many organizations are moving towards automation and developing applications according to their needs but none of them have specifically worked on the report generation on the basis of feedback form.

1.3 Problem Statement

The information which is most critical to a business success lies in the reports. These reports, numbering from dozens to thousands, are typically generated in a manual fashion by a group of analyst and administrators. It is possible to teach a computer to do it, thus freeing engineers and managers for more creative tasks.

1.4 Objectives

The objective is the development of an application which will allow user to give input feedback and generate a report in natural language on the basis of feedback of that person. It willalso allow user to amend the feedback form according to his needs and generate reports.

1.5 Deliverables

i. First Progress Report: including SRS Document
ii. Second Progress Report: including System Architecture and Design
iii. Third Progress Report: including Software Code, Integrated System and its Demo
iv. Final Report: including complete documentation of the system and user manual.

1.6 Technological Requirements

i. Microsoft Visual Studio 2012
ii. MySQL
iii. Windows 7 or 8

Chapter 2

LITERATURE REVIEW

2.1 Previous Work

Natural Language generation systems have been generating jokes very effectively, but from a commercial perspective, the NGL applications which have seen the most success are data to text systems which generate textual summaries of databases. In particular, several systems have been developed over the years to generate textual weather forecast from weather data. The earliest such system to be deployed was FoG [1], which was used by Environment Canada to generate weather forecast in French and English. Recent research showed that users occasionally preferred computer generated weather forecast to human written ones, because computer forecast used more consistent terminology [2], and a demonstration that statistical techniques could be used to generate high quality weather forecasts [3]. Recent applications include the ARNS system used to summarize conditions in US ports.

NGL has also been used to summarize financial and business data. For example the SPOTLIGHT system developed at A.C. Nielsen automatically generated readable English text based on the analysis of large amounts of retail sales data [4]. The effectiveness of these systems created a great opportunity which resulted into an appearance of commercial applications [5]. These system are also been very effectively used for decision support aids for medical professionals [6].

University of Aberdeen has worked on some projects in the NGL field. They have developed BabyTalk which presents patient information to medical professionals and family members [7]. They have developed STOP which generates tailored letters to help people stop smoking [8]. NGL has also been used for safety purposes of scuba divers. A project called SUBATEXT has been developed to produce textual (English) reports of dive data recorded by dive computers. The computer generated report contains for issues across multiple dive profiles such as necessary and unnecessary stops. And also produce reports for unsafe dive profiles with special patterns such as square and reverse and saw-tooth profiles [9]. RoadSafe is a collaborative project between the Computing Science Department at Aberdeen University and Aerospace

& Marine International. The RoadSafe project aims to build upon the expertise Aerospace & Marine International has in weather forecasting and the expertise the Computing Science Department at Aberdeen University has in building real world Natural Language Generation Systems. The main objective of the project is to use the advisory texts produced by RoadSafe as a guide to local councils for grit and salting applications during the winter[10].NEONAT is the decision support in the neonatal intensive care unitimplement and evaluate computerized aids designed to support clinical decision making[11].

2.2 Shortcomings

The systems mentioned above are effective in their respective domains but there is very little work on the systems generate pen picture of a person on the basis of a feedback form. And more over there is no system which give its users the freedom to add to the language or definitions database they already have.

2.3 Issues Solved by ARGS

Developing Automatic Report Generating System will resolve the above mentioned issues. This system will be intelligent and evaluate a person on the basis of a feedback form. It will also provide its users to add to the language database. Which will result in a huge database of language and bring diversity to this application

ARGS has the ability to evolve by time. It provides a dynamic nature which will not only allow users to add to language database but also create their own feedback forms and completely new set of definitions in the language database.

Chapter 3

SYSTEM REQUIREMENTS

3.1 Introduction

System requirements chapter describes the functional and nonfunctional requirements of the Automatic Report Generating System. Objectives are briefly summarized followed by detailed description of the system's scope, vision, use case, features and other related requirement issues.

3.2 Product Scope

Automatic report generating system is designed to generate automatic report of an entity on the basis of the feedback form. This system will be intelligent and evaluate any entity and generate a report in natural language (English) provided the feedback form data.

More specifically, automatic report generating system eases the work of the administrative heads', to whom the tedious task of writing the reports is being assigned. It is designed to maximize the administrative productivity by providing tools to assist in automating the analysis and generation of reports, which would otherwise have to be performed manually. It maximizes the administrative heads' work efficiency and production. And it will meet the administrative heads' needs while remaining easy to understand and use.

3.3 Product Perspective

At present organization's administration has to observe and write reports of their subordinates in a manual fashion. These administrative personals are highly educated and trained professionals at high posts who have quite an experience in their field. In Military College of Signals, Heads of Department (HoDs) are assigned to perform this task. But it does not stop at HoD level, it makes it way all way to the top going through Chief Instructor, Dean, Commandant and Rector NUST. All of them have to write remarks.

How tedious and unproductive this task might be, this can't be given to any employee other than the administrative heads', as these reports are very critical to a business success. These reports, numbering from dozens to thousands, are typically generated in a manual fashion by a specific group of analyst and administrators. It is possible to teach a computer to do it, thus freeing engineers and managers for more creative tasks.

3.4 Product Functions

Major functions of the automatic report generating system are form builder, feedback form analyzer and report generator. Form builder provides the user with necessary tools to build a new form. Feedback analyzer analyzes the feedback of person given by the user.Report generator generates the pen picture of the person on the basis of feedback.

3.5 User Classes and Characteristics

There are three user classes of automatic report generating system which are Data Entry Operator, User and Administrator. Data entry operator will have the least privileges to this system. He will only be able to feed the data of feedback form into the system for an entity. He may enter data for multiple entities and save it. This will help the Report Analyzer, who actually needs to generate reports, by saving his time. User will perform all the tasks to generate the report from the data entered above. Administrator will have the most rights to this system. He can perform all tasks of the system anytime and will have access to the database of the language generator. He can add, remove and modify definitions of quality traits in the database, which can affect the generated sentences by the system.

3.6 Operating Environment

Automatic report generating system will operate on Windows based operating system which includes Windows 7 or Windows 8 only.

Hardware requirements for the system are the minimum operating system requirements suggested by Microsoft for its corresponding Windows version.

3.7 Design and Implementation Constraints

Automatic report generating system works on Windows based system with the minimum operating system requirements provided by Microsoft. It does not require any specialized hardware like GPUs to run it.

Automatic report generating system has MySQL databases, which needs a MySQL server installed on the machine with its connector to the application otherwise application will not be able to access the database. MySQL version should be compatible with available version of windows.

3.8 User Documentation

A user manual is provided at the end of this document.

3.9 External Interface Requirements

External interface requirements specify hardware and software elements with which a system or component must interface

3.9.1 User Interfaces

Automatic report generating system will provide its users with an interface in which users will be navigated from 3 to 6 screens, maximum, to perform his desired operation. This system is made to automate the manual driven tasks, however these tasks are quite diverse and may differ in every user perspective hence a freedom of taking over the system and manually changing the data at certain instance will be available. The anticipated user interfaces are login screen, feedback form selection/opening, form building, data Entry, report generation, report finalization.

3.9.2 Hardware Interfaces

Automatic report generating system will not use any specialized hardware. All it needs is hardware capable of running the Windows 7 or Windows 8 operating system.

3.9.3 Software Interfaces

Automatic report generating system is a Windows based application. It will have a huge database for its language generation which will be on MySQL. In order to connect a Windows based application with a MySQL database user will need to install a MySQL connector. Application will use complex algorithms to analyze the feedback form and will request the database for the sentences in order to generate the reports. User will also be able to manipulate the data through application. Application will create an instance of database and present it in the application domain. User will be able to see the database and make changes according to his needs, which will send the database query to MySQL database and change the actual database.

3.10 System Features

System features of automatic report generating system consist of the main functionality provided by the system. These features collectively make up the system. System features of automatic report generating system are form builder, report analyzer, report generator.

3.10.1 Form Builder

3.10.1.1 *Description and Priority*

Form builder is one of the main features of this system which will allow this application to be dynamic and generally available to any organization and amendable for their use. This is of very high priority as different organizations have different ways to analyze the entities under their charge. This makes it practically impossible to make single general feedback form which anyone can use. Hence this feature will provide its users with necessaryfunctionality to build or amend the application, suiting their needs.

3.10.1.2 *Stimulus/Response Sequences*

User choses the no. of levels which one trait will have.From builder will respond by making all feedback traits, when commanded, with the no. of levels selected.User presses the “ADD” button in order to add another feedback quality. Application respond by adding another quality for the feedback form with the no. of levels selected in the 1st stimulus. Application will also generate a serial no. with every quality which will be an ID for that specific quality and will be used in connecting it with the database.User manipulates the database corresponding to the dynamic feedback form. Application will

save the user given values in the actual database. This will result in connection of feedback Qualities with its user defined definitions.

3.10.1.3 *Functional Requirements*

Functional requirements of form builder are given in the table 3.1.

Table 3.1: Functional Requirements for Form Builder

Req #	Description	Priority
1.	The system shall allow the user to choose the no. of Quality shades for a specific Quality	[Priority = High]
2.	The system shall allow the user to add more Qualities.	[Priority = High]
3.	The system shall allow the user to access language generation database.	[Priority = High]
4.	The system shall allow the user to manipulate the database.	[Priority = High]
5.	The system shall allow its user to save the dynamic form created.	[Priority = High]
6.	The system shall allow its user to open a saved dynamic form.	[Priority = High]

3.10.2 Feedback Analyzer

3.10.2.1 *Description and Priority*

Feedback analyzer will check the format of the feedback form. Once the format is identified, it will allow user to select which qualities to analyze. Once selected, system will analyze those qualities and send results to sentence generation module.

3.10.2.2 *Stimulus/Response Sequences*

User selects the qualities and presses the button for analyzing of feedback report. System will analyze selected qualities and send results to sentence generation module. Then user unselects and selects others qualities and presses the button for analysis. System will respond by only analyzing those selected qualities and send results to sentence generation module.

3.10.2.3 *Functional Requirements*

Functional requirements of feedback analyzer are given in the table 3.2.

Table 3.2: Functional Requirements for Feedback Analyzer

Req #	Description	Priority
1.	The system shall allow the user to select qualities.	[Priority = High]
2.	The system shall allow the user to unselect qualities.	[Priority = High]
3.	The system shall analyze selected qualities.	[Priority = High]

3.10.3 Report Generator

3.10.3.1 *Description and Priority*

This is the top priority feature of the system. This function will generate the report of the given entity, after receiving results from the feedback analyzer, in natural language. It will generate sentences and allow the user to select from the sentences for a particular report or if the user wants it may select sentences randomly to form a report by itself.

3.10.3.2 *Stimulus/Response Sequences*

The user asks the system to start the automated report generation. System will run the analyzed data through its algorithms and will present the user with possible sentences for the report.

User chooses the desired sentences for the report. System will combine those sentences into a well formed report. User asks the system to choose the sentences randomly. System will combine random sentences into a well formed report.

3.10.3.3 *Functional Requirements*

Functional requirements of report generator are given in the table 3.3.

Table 3.3: Functional Requirements for Report Generator

Req #	Description	Priority
1.	The system shall allow its user to generate the report.	[Priority = High]
2.	The system shall show its user the generated sentences.	[Priority = High]
3.	The system shall allow its user to edit sentences.	[Priority = High]
4.	The system shall allow user to save all data to database.	[Priority = High]

3.11 Other Nonfunctional Requirements

Nonfunctional requirements of the automatic report generating system consist of performance, security requirements.

3.11.1 Performance Requirements

Production of simple report shall take less than 30 seconds for 90% of the cases. In standard workload, the CPU usage shall be less than 50%, leaving 50% for background jobs.

3.11.2 Security Requirements

Automatic report generating system has 3 user classes. These are Data Entry Person, User and Administrator. Data Entry Person shall be able to open an available form and enter data into it. Data Entry Person shall be able to save the entered data. User shall enter his/her credentials in order to sign in to system and access the limited features. User shall have rights to build form and generate reports. Administrator shall enter his/her credentials in order to sign in to system to access all the features. Administrator shall have access to language generator database. Administrator shall be able to manipulate the database.

3.12 Software Quality Attributes

Software quality attributes of automatic report generating system are robustness, usability measures and maintainability.

3.12.1 Robustness:

Less than 30 seconds shall be needed to restart the system after a failure, 95% of the time.No more than 1 per 50 report generated shall result in a failure.

3.12.2 Usability Measures:

Novice users shall be able to enter data and generate report in 15 minutes.Experienced users shall perform the above tasks in 5 minutes.

3.12.3 Maintainability

Installation of a new version shall leave all database contents unchanged.

3.13 Other Requirements

All requirements are already mentioned above.

Chapter 4

System Design

4.1 Introduction

System Design chapter provides a comprehensive architectural overview of the Automatic Report Generating System. It presents a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

In order to depict the software as accurately as possible, this chapter will show the logical, implementation and dynamic view with use cases of the system.

4.2 Scope

The scope of system designchapter is to depict the architecture of the Automatic Report Generating System. This will allow various stakeholders to find what they need in the software architecture.

4.3 Architecture Design

4.3.1 Architecture Pattern:

In automatic report generating system the Model-View-Controller architecture pattern is used. This system has to generate the report in natural language on the basis of feedback form. Considering the functionality itself to be very complex, this makes the user interface equally complex and difficult to make. It is very important that user should be provided with an appropriate user interface in order to successfully give the input to the system. As user interfaces are especially prone to change requests. This model makes it easier to response to those requests effectively. As the MVC architectural pattern divides an interactive application into three components as shown in the figure 4.1.

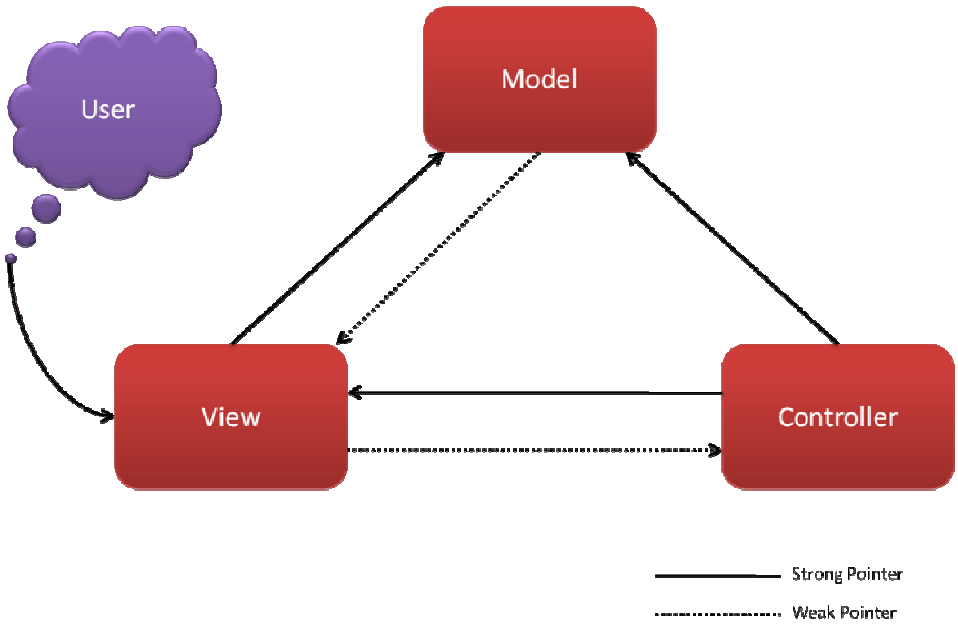


Figure 4.1:MVC Pattern

View contains all the user interfaces of system. It displays information to the user. A view communicates with the controllers to process the input and obtains the data from the model. It will allow user to enter the feedback values and select operations.

Controller handles the input from the user, usually as events. Events are translated to service requests for the model or the view. Controller will be responsible for delivery of user request for particular operations like data analysis, report generation and report finalization.

Model contains the functionality of data analysis, report generation and report finalization. It will also contain all the databases for the generation of sentences.

4.4 Detailed Design

Detail design of automatic report generating system consists of uses cases, class diagram, sequence diagram, activity diagram and package diagram.

4.4.1 Use Cases:

Use cases show the interaction between the actors and system to achieve a goal. Use cases of automatic report generating system are shown in figure 4.2 and figure 4.3.

Actors of automatic report generating system are Data Entry Operator, User and Admin.Use cases of automatic report generating system are Data Entry, Form Building, Feedback Analysis, Report Generation, and Data Manipulation.

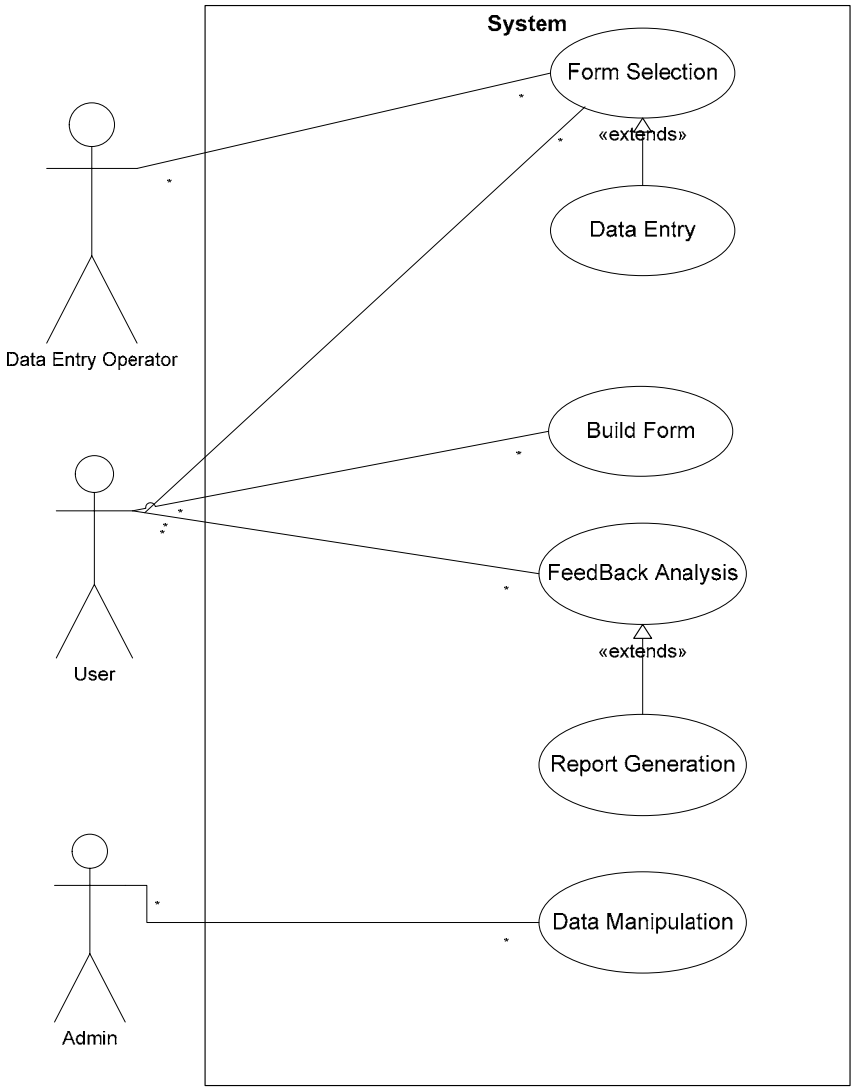


Figure 4.2: Complete System Use Case

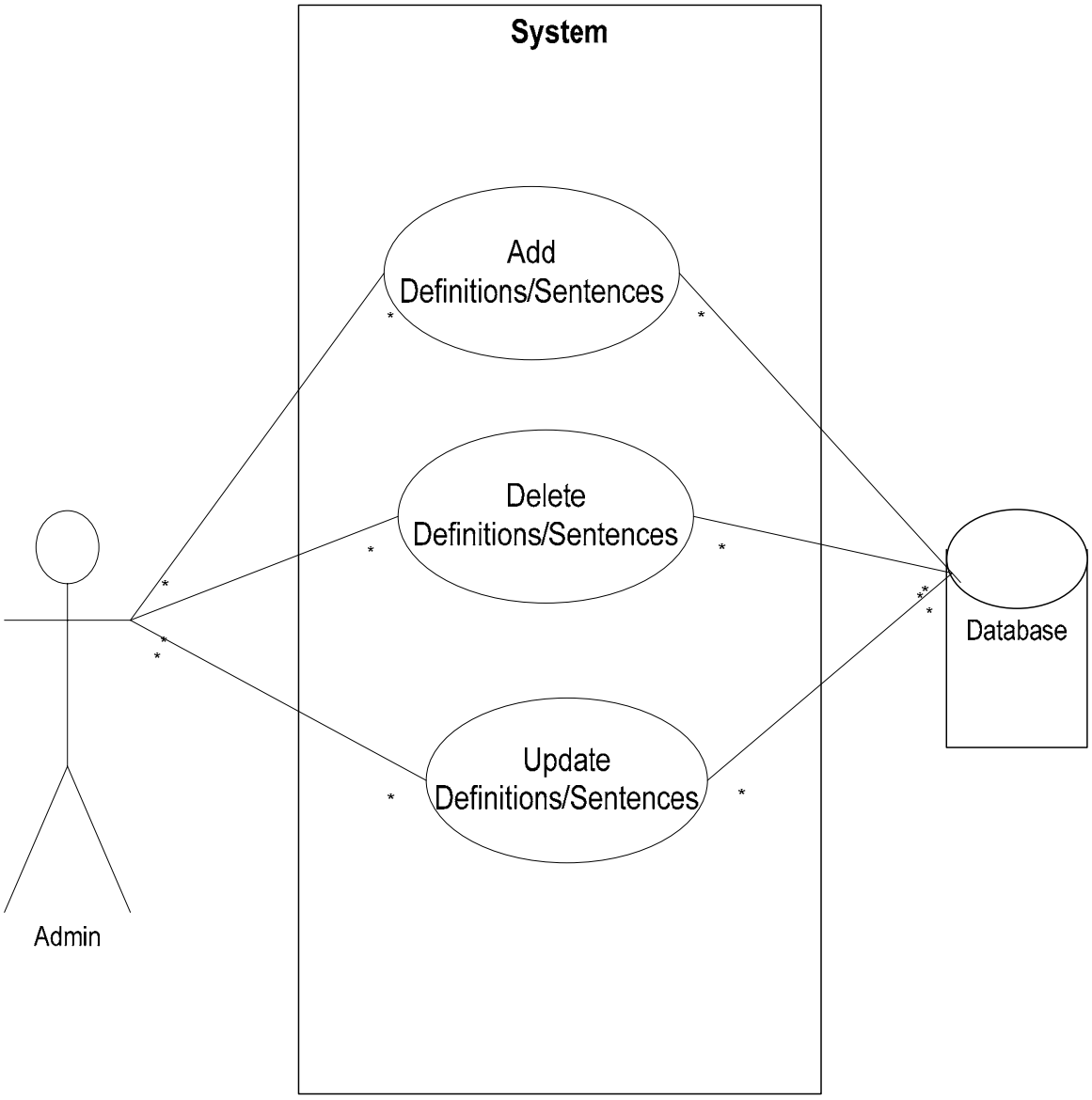


Figure 4.3: Administrator Use Case

4.4.1.1 *Use Case Descriptions:*

Use case descriptions provide the details of each use case.

i. **Form Selection**:

Actor (Initiator): Data Entry Operator.

Purpose: To select a form for data entry.

Overview: System will open selected form for data entry

Preconditions: System must be running.

Flow of Events:

Actor Action: Press button of required form.

System Response: Opens the select form.

Post Conditions: System is running.

ii. **Data Entry**:

Actor (Initiator): Data Entry Operator.

Purpose: To enter data into selected form.

Overview: System will provide with an interface to enter data.

Preconditions: Selected form must be open.

Flow of Events:

Actor Action: Enters data.

System Response: Displays data.

Post Conditions: System is running.

iii. **Build Form**:

Actor (Initiator): User.

Purpose: To build a new form.

Overview: System will build a new form according to his needs.

Preconditions: System must be running.

Flow of Events:

Actor Action: PressAdd/Delete buttons.

System Response: Add/Delete qualities.

Post Conditions: System is running.

iv. **Feedback Analyzer**:

Actor (Initiator): User.

Purpose: To analyze feedback.

Overview: System will perform calculations on feedback values.

Preconditions: All feedback values must be entered.

Flow of Events:

Actor Action: Presscalculates buttons.

System Response: Perform calculations and show them to user.

Post Conditions: System is running.

v. **Report Generation**:

Actor (Initiator): User.

Purpose: To generate report.

Overview: System will generate report on the basis of feedback values.

Preconditions: All feedback values must be entered.

Flow of Events:

Actor Action: Pressgenerate button.

System Response: Generate pen picture and show it to user.

Post Conditions: System is running.

vi. **Data manipulation**:

Actor (Initiator): Administrator.

Purpose: To manipulate definitions database.

Overview: Buttons are provided to build new form.

Preconditions: System must be running and connected to database..

Flow of Events:

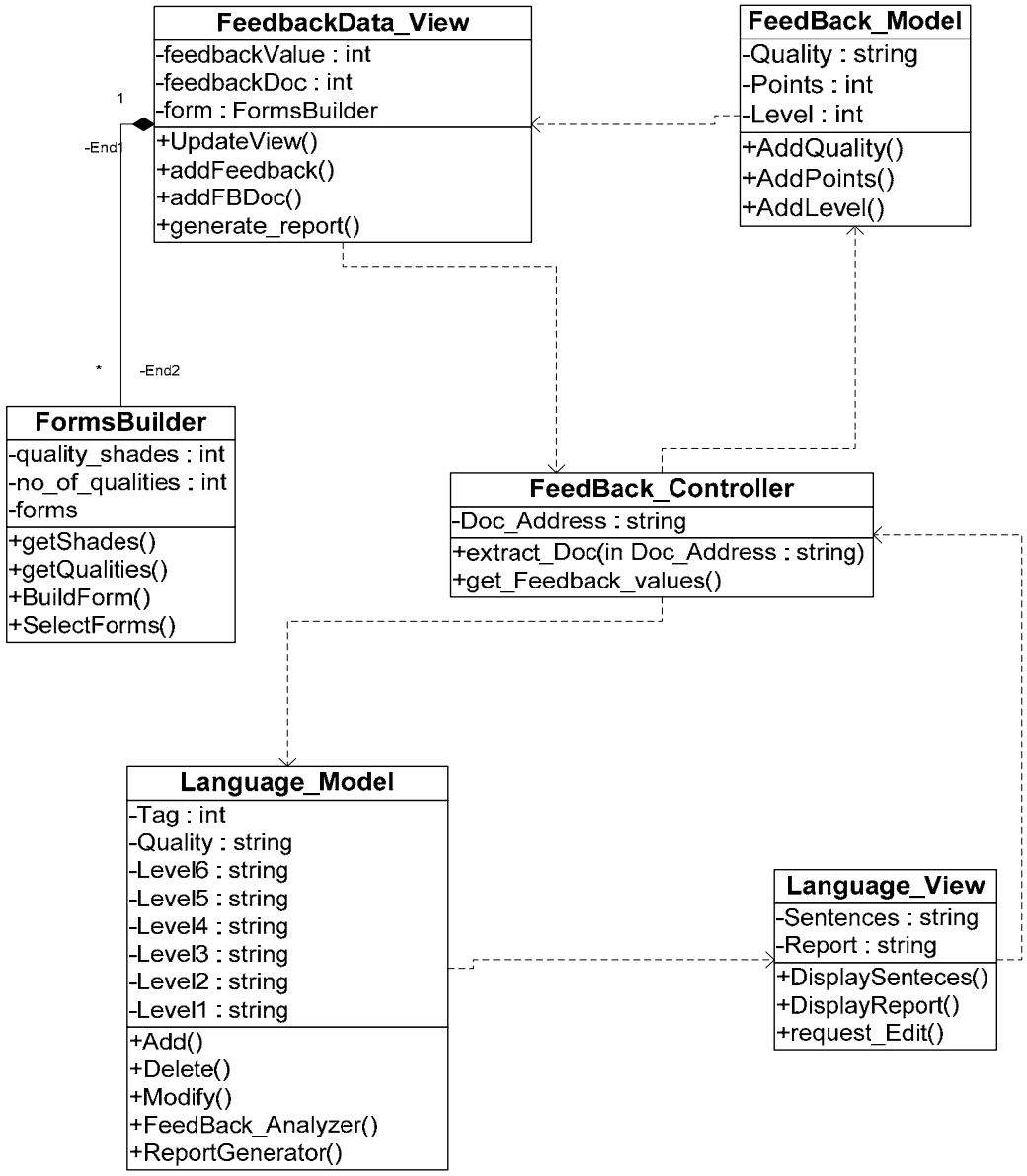
Actor Action: Manipulate data values.

System Response: Update database.

Post Conditions: System is running.

4.4.2 Class Diagram:

Automatic report generating system comprises of feebackdata_view class, formbuilder class, feedback_model class, feedback_controller class and language_model class and language_view class. Class diagram of automatic report generating system is shown in figure 4.4.



4.4.3 Sequence Diagram

Sequence diagrams shows how process operates with each other and operates in which order. Sequence diagram of automatic report generating system is shown infigure4.5. First user opens a forms builder. Form builder asks for number of quality shades.User selects the number of quality shades.Form builder asks for number of qualities. User enters the number of qualities. User enters data and invokes the system to generate

report. Request is handled by form analyzer which analyzes the feedback values and sends results to report generator. Report generator gets the sentences from database and applies algorithms to form a report. After forming a complete pen picture, report generator displays it to user.

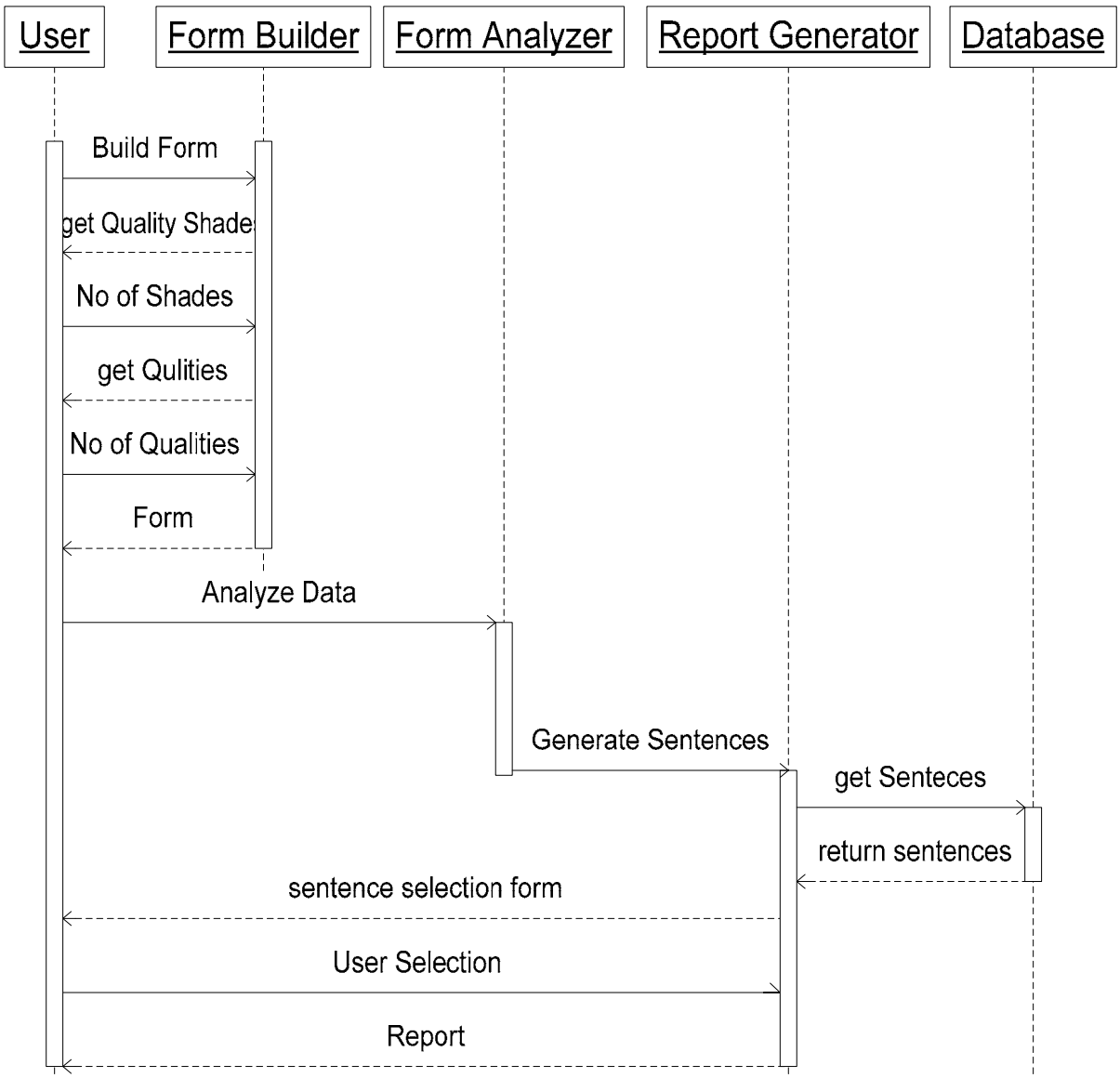


Figure 4.5: Sequence Diagram

4.4.4 Activity Diagram:

Activity diagram shows the graphically workflows of stepwise activities with options. Activity diagram of automatic report generating system is shown at figure 4.6.System gives option to user to make a new form or select from existing forms.If user selects an existing form, system opens the selected form then user enters the feedback data and system analyzes the data and generates the report. But if user selects to build a form system opens a form builder. User enters number of quality shades and number of qualities.System builds a form and displays it to user. User enters the feedback data which system analyzes and generates a report of it.

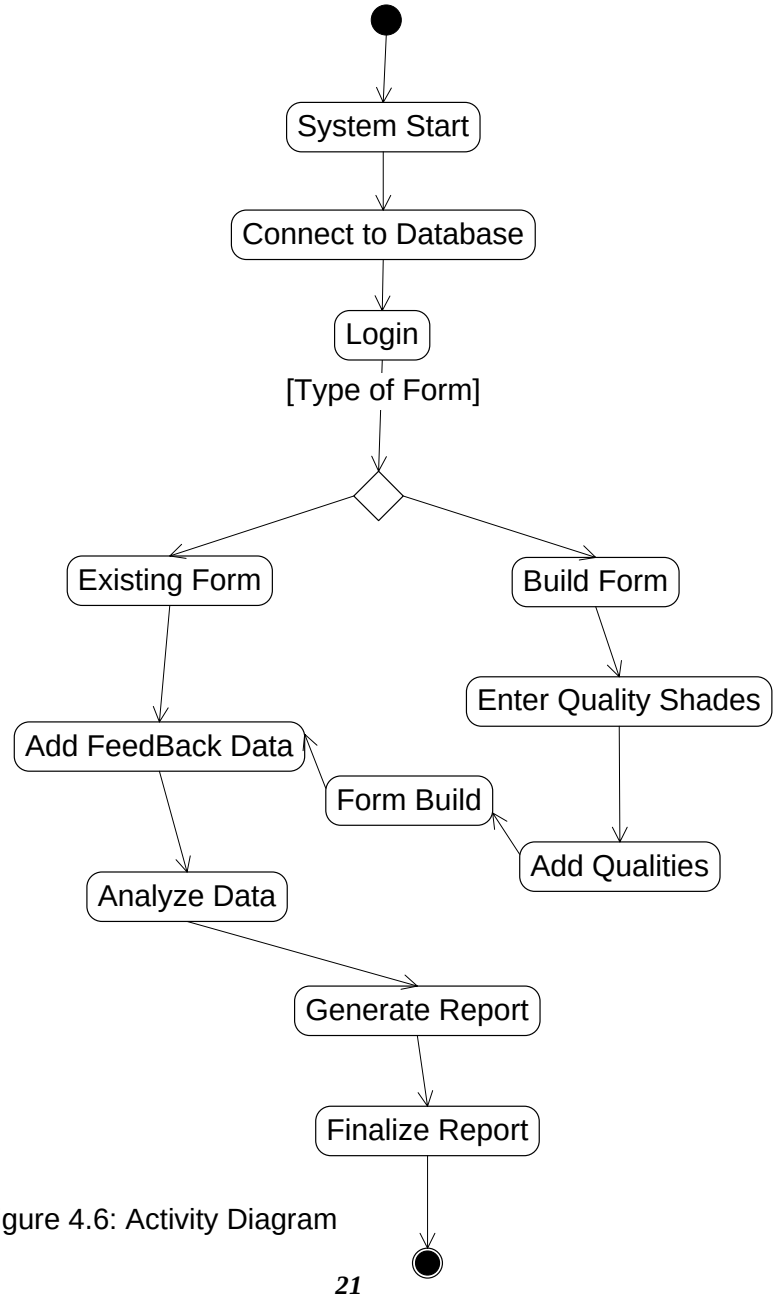


Figure 4.6: Activity Diagram

Chapter 5

SYSTEM IMPLEMENTATION

5.1 Tools and Technologies

5.1.1 Microsoft Visual Studio 2012

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interfaceapplications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for allplatforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F#.Automatic report generating system has been implemented in C#.

5.1.2 MySQL

MySQL is one the world's most popular open source database software. With its top speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

5.2 Software Implementation

Automatic report generating system has been developed in Visual Studio 2012 using C# .Net Framework 4.5. It is following a Model View Controller pattern. Following are the implantation details specific to application functions.

5.2.1 Server Connection

Automatic report generating system needs to connect with a server which has its database. It can connect to a local server on the same machine as client or it can have a remote server on some other machine. It is connected through an IP address. This IP address is inserted into a connection string at runtime and it tries to connect the server at given IP address. Application needs to remain connected to same server at all times in order to work. It has been done by using class which can serve as global variable for all window forms.

```
class Global
  {
    public static string cstring = "";



    public static string ip
    {
      get { return cstring; }
      set { cstring = value; }
    }
  }
```

Through this class IP address on all forms remains constant, using it as following

```
static string connStr = "server="+Global.ip+";user=root;database=fyp;port=3306;
```

5.2.2 Dynamic Form

In dynamic form construction all controls on the forms are created dynamically which include Text Boxes, Labels and Check Boxes.

Text Boxes:

```
textbox[textbox_no] = new TextBox();
            textbox[textbox_no].Location = new Point(rx + j, ry);
            textbox[textbox_no].Width = 80;
            textbox[textbox_no].Text = "";
```

Controls.Add(textbox[textbox_no]);

Labels:

label[q] = new Label();

label[label_no].Location = new Point(x, y);

label[label_no].Width = 80;

label[label_no].Text = (number).ToString();

Controls.Add(label[label_no]);

Check Boxes:

checkbox[label_no] = new CheckBox();

checkbox[checkbox_no].Location = new Point(x,y);

checkbox[checkbox_no].Width = 15;

Controls.Add(checkbox[checkbox_no]);

All of these controls locations are been set by application to keep it well structured.

5.2.3 Form Analyzer

Form analyzer uses averaging algorithm. First it identifies how many quality levels are there in the form, numbering from 4 to 6. Using that information it applies its modified algorithm for the corresponding quality levels. It can be given as for six quality levels as

*Rating={(6*6th level value)+(5*5th level value)+(4* 4th level value)+ (3*3rd level value)+(2*2nd level value)+(1*1st level value)}/Total Values*

```
if (Rating >= 48)
    level = 6;
else if (Rating < 48 && Rating >= 42)
    level = 5;
else if (Rating < 42 && Rating >= 36)
    level = 4;
else if (Rating < 36 && Rating >= 33)
    level = 3;
else if (Rating < 33 && Rating > 30)
    level = 2;
else if (Rating <= 30)
    level = 1;
```

while MAX Ranking=60 and MIN Ranking=1

5.2.4 Language Generation

Language generation module takes the results of form analyzer and follows the language generation algorithm. First language generation modulewill check whether the user want a negative pen picture of the person or a positive pen picture. It is done by checking how many selected qualities for report generation have rank above and below average. If above average is greater then it's a positive report and if below average is greater than it's a negative report.It is done

```
for (int i = 1; i <= total_checks; i++)
        {
            if (level[i] > above_average)
    positive++;
            if (level[i] < below_average)
negetive++;
        }
```

If positive result of the first step, application will collect all the above average selected qualities and put them first in the array of sentence generation followed by all the below average selected qualities. And in case of negative result of the first step application will collect all the below average selected qualities and put them first in the array of sentence generation followed by all the above average selected qualities.

```
for (int i = 1; i <= total_checks; i++)
    {
        if (positive >= negetive)
        {
            if (level[i] > 2)
            { level_check[i] = true; }
            else
            { level_check[i] = false; }
        }

        else
        {
```

```
            if (level[i] < 3)
            { level_check[i] = true; }
            else
            { level_check[i] = false; }
        }
    }
```

Check if the gender of the person who's pen picture has to be generated. Import sentences form the definitions database corresponding to the above steps results. Check if it's the first sentence of the report. If yes, start sentence with the name of that person.If no, start sentence with personal pronouns i.e. he\she.Check if the sentence is followed by more than one sentence of that part of the report. If yes, add a full stop. If no, add an "and". Check if last sentence ended with a full stop, comma or "and". If full stop, start next sentence with a capital letter. If "and" or comma, start sentence with small letter. For the second part of the report, start sentence with a connecter which will keep the report in well-formed structure and make sense to human beings. These connecters include "However", "That being said", "On the contrary", "On the other hand", "Having said that". End the connecter with a comma. If the sentence starts with a connecter then personal pronoun must start with a small letter.Check if the sentence includes him, his, her, herself, himself.If yes, make its gender corresponding to the gender of the person whose report is being generated.

5.2.5 Report Exporter

Report exporter connects with the Microsoft word file and report values are inserted into the document. It is implemented by using "Microsoft word interop word" library which is used to connect with the already existing word file. Tags were inserted into an already formatted word document and they were replaced by their corresponding values in the database to generate a complete report in word document. Sample code of how word file is being connect to application is

```
        //Open the word document
    aDoc = wordApp.Documents.Open(ref fileName, ref missing,
        ref readOnly, ref missing, ref missing, ref missing,
        ref missing, ref missing, ref missing, ref missing,
        ref missing, ref isVisible, ref missing, ref missing,
```

```
    ref missing, ref missing);
// Activate the document
aDoc.Activate();
```

5.3 Software Issues

5.3.1 OS issues

Automatic report generating system is developed in .Net Framework 4.5, hence the only compatible versions of windows in which it can run is Windows 7 or higher.

5.3.2 MySQL issues

Automatic report generating system is developed in Visual studio 2012, which is only compatible with the MySQL .NET connecter 6.6.5+.

Chapter 6

TESTING AND RESULT ANALYSIS

This chapters shows the various test cases applied to automatic report generating system and their results.

6.1 Testing

Testing not only maintains the software and system quality but also improves over all usability and stability of the project. At different stages of development suitable testing techniques were used to ensure product worked accurately and efficiently. Almost all the errors detected during testing were removed.

The overall approach of this test plan is Gray Box testing i.e. hybrid of Black Box and White Box testing. Testing approach will be same for all features of the system.

Test case 1

Test case 1 was conducted to see whether the system is retrieving quality traits values from the access file or not. The values are determined on the basis of the feedback provided by the students. These values then further help in generating the report.

Table 6.1: Test Case 1

Test Case ID	01
Test Case name	Retrieving Quality Traits Values
Input(s)	Faculty Member Name
Output	Retrieves correct values from access file
Sequence of Action(s)	Select faculty member name->Click on IMPORT button
Result	Success

Test Case 2

Test case 2 was conducted to see whether system is generating correct report or not. Report is generated on the basis of the selected quality traits.

Table 6.2: Test Case 2

Test Case ID	02
Test Case name	Generating Report on the basis of selected traits
Input(s)	Tick the traits on the basis of which you want to generate the report
Output	Semantically correct report is generated
Sequence of Action(s)	Select the traits->Click on ANALYZE button
Result	Success

Test case 3

Test case3 was conducted to see whether the system is calculating correct scores or not. Scores are calculated for different sections. These scores are used to calculate the overall standing of the faculty member.

Table 6.3: Test Case 3

Test Case ID	03
Test Case name	Score Calculation
Input(s)	Fill in all the fields of the form
Output	Correct score is calculated
Sequence of Action(s)	Fill in all the fields of the form->press calculate for calculating scores
Result	Success

Test Case 4

Test case 4 was conducted to whether the system is calculating correct overall standing or not. The overall standing is calculated on the basis of the scores calculated in different sections of the form.

Table 6.4: Test Case 4

Test Case ID	04
Test Case name	Overall Standing
Input(s)	Scores calculated above
Output	Correct Overall Standing
Sequence of Action(s)	Fill in all the fields->click calculate for calculating scores for different sections ->click on CALCULATE button to calculate the overall standing
Result	Success

Test Case 5

Test case 5 was conducted to see whether the system is generating semantically correct report or not. Report is calculated on the basis of the selected quality traits. Report must not contradict with the overall standing and scores given to the faculty member.

Table 6.5: Test Case 5

Test Case ID	05
Test Case name	HoD, CI and DEAN Report
Input(s)	Select traits on the basis of which you want to generate the report
Output	Semantically correct report is generated
Sequence of Action(s)	Fill in all the fields->calculate score for 3 different sections->calculate overall standing->select quality traits on the basis of which you want to generate the report ->click on GENERATE button
Result	Success

Test Case 6

Test case 6 was conducted to see whether the report generated on the basis of the extra quality traits is augmented with the previous report or not and to check whether it is semantically correct or not.

Table 6.6: Test Case 6

Test Case ID	06
Test Case name	Extra Traits For Report Generation
Input(s)	Press EXTRA button
Output	Report is updated after with new sentences based on additional traits selected
Sequence of Action(s)	Fill in all the fields->calculate score for 3 different sections->calculate overall standing->select quality traits on the basis of which you want to generate the report ->click on EXTRA button->select additional traits->click on GENERATE button->click on SAVE button
Result	Success

Test Case 7

Test case 7 was conducted to see whether all information is correctly exported to the word document or not.

Table 6.7: Test Case 7

Test Case ID	07
Test Case name	Exporting to WORD Document
Input(s)	Fully Compete Form
Output	All the values are exported to MS word
Sequence of Action(s)	Fill in all the fields->calculate score for 3 different sections->calculate overall standing->select quality traits on the basis of which you want to generate the report ->click on EXTRA button->select additional traits->click on GENERATE button->click on SAVE button->save the form by pressing the SAVE button->click on EXPORT button->click on CREATE DOCUMENT button
Result	Success

Test Case 8

Test case 8 was conducted to see whether a particular record is correctly imported from the database or not. To test this functionality it is necessary to have at least one record in the database which can be used for importing purpose.

Table 6.8: Test Case 8

Test Case ID	08
Test Case name	Importing a Record from Database
Input(s)	Select a name from a list of faculty members present in the database
Output	All the records should be correctly imported
Sequence of Action(s)	Click on OPEN button->Select a name from the list->click on OPEN button
Result	Success

Test Case 9

Test case 9 was conducted to see whether user can update the database at runtime or not. For this test you have to enter the data in all columns of the data grid view and then you will have to run MYSQL to check whether the database is updated or not.

Table 6.9: Test Case 9

Test Case ID	09
Test Case name	Updating Database At Run time
Input(s)	Fill in all columns of the data grid view
Output	Database should be Updated
Sequence of Action(s)	Enter values in every column of data grid view->press update->open SQL Workbench->check whether data which you enter at run time exist or not
Result	Success

Test Case 10

Test case 10 was conducted to see whether the dynamic form builder is working correctly or not. Dynamic form contains buttons which respond to user request and change display accordingly. Even if one buttons fails complete dynamic form will be useless. All buttons of the dynamic form were tested in every order to make sure it was working successfully.

Table 6.10: Test Case 10

Test Case ID	010
Test Case name	Dynamic Form
Input(s)	Enter number of traits and no of shades or press open to load already created forms
Output	User should be able to create a dynamic form which meets the requirements of his organization and then using that form should be able to generate a semantically correct report
Sequence of Action(s)	Select no of shades->Select no of traits->Press NEXT->Enter qualities and shades ->Press add quality button to add new quality-> Press add shade button to add new shades-> Press del quality button to delete a quality-> Press del shade button to delete a shade->Press NEXT->Press ANALYZE button to generate a pen picture
Result	Success

Test case 11

Test case 11 was conducted to see whether the menu strip is working correctly or not. All the items of the menu strip including the sub items were tested.

Table 6.11: Test Case 11

Test Case ID	011
Test Case name	Menu Strip
Input(s)	Select any item from the menu strip
Output	All items of the menu strip should be working correctly
Sequence of Action(s)	Select each item one by one and check its sub items
Result	Success

Test Case 12

Test case 12 was conducted to see whether the system is generating semantically correct report or not. Report is generated on the basis of the selected quality traits.

Table 6.12: Test Case 12

Test Case ID	012
Test Case name	Sentence Generation
Input(s)	Select quality shades on the basis of which you want to generate a report
Output	Semantically correct report should be generated
Sequence of Action(s)	Select quality traits on the basis of which you want to generate the report->press GENERATE button->Manually read the report to find any semantic errors
Result	Success

Test Case 13

Test case 13 was conducted to see whether the login function is working correctly or not. Only legitimate users should be able to login to the system.

Table 6.13: Test Case 13

Test Case ID	013
Test Case name	Login Function
Input(s)	Enter user name and password
Output	Only people with correct user name and password should be able to login to the system.
Sequence of Action(s)	Enter user name->Enter password->Press Next
Result	Success

Test Case 14

Test case 14 was conducted to see whether different clients from their own machines can access the database which is present on the server. For this test connection between client and server was tested.

Table 6.14: Test Case 14

Test Case ID	014
Test Case name	Centralized Database
Input(s)	Enter IP address of the server
Output	Client should be able to proceed to the next page i.e. login page this is only possible if the connection between client and server is established. The client should be able to access the database
Sequence of Action(s)	Enter IP address of server->Press Next
Result	Success

6.1.1 System Testing

System testing was performed at the end of developing all the forms and algorithms for sentence generation, form analyzer and dynamic form builder. Complete system was tested starting from LOGIN from a client machine, entering data into all the forms and all the way to generating a semantically correct report. The system testing enabled us to detect bugs and defects in our code as well as assembly of the system and improve the system by removing a maximum number of them.

6.2 Results and Analysis

The automatic report generating system has been developed to work most efficiently in the Military College of Signals. It is supposed to save critical time of the all the administrative heads of the college and help them write reports of the officers. It has to provide its users with the freedom to add definitions of their own choice, which restricted us from developing more strong and complex algorithm which might not have been able to give access to its users to change or add more definitions of their own choice. Regardless of that this system was able to deliver impressive results and very well formed reports.

6.2.1 Results

Results of automatic report generating system show its efficiency and precision. No more than 1 in 50 reports results in failure and most of it was because of the simple mistake of entering a wrong definition in the database and wrong user entries.

After analyzing 20 users in carrying out necessary actions to generate a report, results showed that expert users are able to generate report in less in 5 minutes and novice user are talking about 10 minutes.

Also a comparison of how much time it takes to write a report manually and by generating through automatic report generating system showed that automatic report generating system saves more than 50% of time in 90% of the cases.

The overall results of Automatic Report Generating system are satisfactory enough to make it to the system of HoD EE Department of Military College of Signals.

6.2.2 Analysis

Since the system is a natural language generation performance, reliability and usability are important features. Reliability is an important aspect of this system thus it has been given special attention. The system is kept simple and its use is very easy.The system consists of separate interacting modules which provide a great opportunity for future development.

Chapter 7

CONCLUSION AND FUTURE WORK

The goal of the project “Automatic report generating system” was to learn about and develop a natural language generation system and dynamic nature of windows application. The project was chosen from a list of proposed projects. Pen Picture generation on the basis of feedback form was the original problem statement. We envisioned developing a system which is generic and dynamic which users can amend according to their own needs. And also provide Military College of Signals which with pen picture generation functionality for their own ACRs. The goal was achieved at the end but as with every other project team we faced a critical issue of limited resources which includes time and lack of help available specific to its domain. The limitations forced us to compromise somewhat on the scope of the application. But nevertheless the final product is quite impressive and good inspiration for future development and research. The design of the system is kept very simple and code is also very well structured with adequate amount of comments in order to make it easier for others to understand. Software engineers can easily understand the system and code underlying it with help of this document.

Improvement in the area of dynamic form will be very appreciating. Dynamic nature suggests endless possibilities. We worked only on the dynamic feedback form and we suggest any future teams working on same project should expand the dynamic nature of feedback form by allowing more controls to be dynamically added and connecting them with the database at runtime.

There is also room for improvement in the integration of this application with Microsoft Word. We successfully connected these two statically and find some great results and very beneficial for organizations. Making the integration more dynamic will be a great leap forward.

The team intends to give full support to this project and any future team working on this project as the real aim was to help Military College of Signals and all the other organizations in Pakistan. We hope that one day our nation will be among the fore-

runners in the field of science, research and information technology and Pakistan will earn its rightful place among the international community. Aamin!

Bibliography

[1] Goldberg E, Driedger N, Kittredge R (1994). "Using Natural-Language Processing to Produce Weather Forecasts".

[2] Reiter E, Sripada S, Hunter J, Yu J, Davy I (2005). "Choosing Words in Computer-Generated Weather Forecasts".

[3] Belz A (2008). "Automatic Generation of Weather Forecast Texts Using Comprehensive Probabilistic Generation-Space Models"

[4] Anand, Tej; Kahn, Gary (1992). "Making Sense of Gigabytes: A System for Knowledge-Based Market Analysis". In Klahr, Philip; Scott, A. F. *Innovative applications of artificial intelligence 4: proceedings of the IAAI-92 Conference*. Menlo Park, Calif: AAAI Press

[5] Harris MD (2008). "Building a Large-Scale Commercial NLG System for an EMR". *Proceedings of the Fifth International Natural Language Generation Conferenc*

[6] Portet F, Reiter E, Gatt A, Hunter J, Sripada S, Freer Y, Sykes C (2009). "Automatic Generation of Textual Summaries from Neonatal Intensive Care Data". *Artificial Intelligence* 173 (7–8): 789–816

[7] http://www.abdn.ac.uk/ncs/computing/research/nlg/projects/previous/babytalk/

[8] http://www.abdn.ac.uk/ncs/computing/research/nlg/projects/previous/stop/

[9] http://www.abdn.ac.uk/ncs/computing/research/nlg/projects/previous/scubatext/

[10] http://www.abdn.ac.uk/ncs/computing/research/nlg/projects/previous/roadsafe/

[11] http://www.abdn.ac.uk/ncs/computing/research/nlg/projects/previous/neonate/

[12] Turner R., Sripada S., Reiter E. and Davy I. (2007). Selecting the Content of Textual Descriptions of Geographically Located Events in Spatio-Temporal Weather Data. In Applications and Innovations in Intelligent Systems XV,pages 75-88

[13] Gatt, A. and van Deemter, K. (2007). Incremental generation of plural descriptions: Similarity and partitioning. Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP-07

[14] van Deemter, K. (2006). Generating Referring Expressions that involve gradable properties. Computational Linguistics, 32(2).

Appendix A: Glossary

Dynamic Forms: These are the forms created at run time either by user or the system. In former case user is provided with necessary tools to build the form.

UML: Unified Modeling Language

MVC: Model-View-Controller

NGL: Natural Language Generator