

Resume Bots

A MINI PROJECT REPORT

Submitted by

MANIARASAN.S (11MSE1083)

PRABAKARAN.A (11MSE1108)

in partial fulfillment for the award of degree of

M.S Software Engineering



School of Computing Science and Engineering
VIT Chennai, Chennai-600127, Tamil Nadu, India

November, 2015



School of Computing Science and Engineering

DECLARATION

I/We hereby declare that the project entitled “**Resume Bots**” submitted by me/us to the School of Computing Science and Engineering, VIT University, Chennai Campus, Chennai – 127 in partial fulfillment of the requirements for the award of the degree of **M.S Software Engineering** is a record of bonafide work carried out by me/us under the supervision of **Prof.Hema N, Assistant Professor (SR)**. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or university.



School of Computing Science and Engineering

CERTIFICATE

The project report entitled “**Resume Bots**” is prepared and submitted by **Maniarasan.S (11mse1083)** and **Prabakaran.A (11mse1108)**. It has been found satisfactory in terms of scope, quality and presentation as partial fulfillment of the requirements for the award of the degree of **M.S Software Engineering** in VIT University, Chennai Campus, Chennai, India.

Signature
(Name of the Supervisor/Guide)

Examined by:

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

I obliged to give our appreciation to a number of people without whom I could not have completed this thesis successfully.

I would like to place on record our deep sense of gratitude and thanks to my internal guide **Prof. Hema N, Assistant Professor(SR), School of Computing Science and Engineering (SCSE), VIT University** whose esteemed support and immense guidance encouraged us to complete the project successfully.

I would like to thank our Program manager, **Dr. N.Maheshwari, School of Computing Science and Engineering (SCSE), VIT University**, for her valuable support and encouragement to take up and complete this thesis. I also like to extend our deepest gratitude to our panel members and other Professors of our Department for their advices and comments in each review, which motivated us to explore more and impart accuracy in the thesis.

Special mention to our dean **to Dr. L. Jagannathan, School of Computing Science and Engineering (SCSE), VIT University**, for spending his valuable time and efforts in sharing his knowledge and for helping us in every minute aspect of software engineering.

I thank our management of VIT University, Chennai campus for permitting us to use the library resources. I also thank all the faculty members for giving us the courage and the strength that we needed to complete our goal. This acknowledgement would be incomplete without expressing the whole hearted thanks to our family and friends who motivated us during the course of our work.

I thank the Almighty, whose showers of grace were felt during the entire course of our project.

CONTENTS

CHAPTER 1	10
INTRODUCTION	10
1.1 General	10
1.2 Motivation	11
1.3 Problem Description	11
1.4 Related Work	11
Literature Study	11
1.4.1 System Requirements	13
CHAPTER 2	14
OVERVIEW OF THE PROPOSED WORK	14
2.1 Introduction of problem and its related concepts	14
2.2 Overview of the proposed system	15
CHAPTER 3	16
ANALYSIS AND DESIGN	16
3.1 Brief Introduction	16
3.2 Requirement Analysis	17
3.2.1 Functional Requirements	17
3.2.2 Non Functional Requirements	18
3.3 Detailed design	20
3.4 Module Description	23
CHAPTER 4	24
IMPLEMENTATION	24
4.1 Tools Used	24
4.2 Methodology	27
Phases	28
Usage	28
Implementation guidelines	29
4.3 Implementation screen shots	30
4.4 Unit Test Cases	35
CHAPTER 5	37
RESULTS AND DISCUSSION	37
5.1 Results	37
5.2 Performance Analysis	37
CHAPTER 6	39
CONCLUSIONS AND FUTURE ENHANCEMENTS	39
REFERENCES	40
APPENDIX -I	40

LIST OF ABBREVIATIONS

Abbreviation	Expansion
WWW	World Wide Web
PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
CSS	Cascading Style Sheet
JSP	Java Script
WAMPP	Windows,Apache,MySQL,PHP/Perl/Python
GB	Giga Bytes
RAM	Random Access Memory
GUI	Graphical User Interface
UR	User Requirement
DR	Domain Requirement
NFR	Non-Functional Requirement
DB	Database
TC	Testcase

LIST OF TABLES

Title	Page
Table 1.1 : Functional requirements	17
Table 1.2 : Non Functional requirements	19
Table 2.1 : Unit test case	35

LIST OF FIGURES

Title	Page
Fig 1: Architecture Diagram	16
Fig 2: Data Flow Diagram	20
Fig 3: Collaboration	20
Fig 4: Use case	20
Fig 5: Class	21
Fig 6: Sequence	22

ABSTRACT

Recruitment of employees has always been a sophisticated process in scrutinizing and selecting the best individuals for the organization. From a pile of lakhs of resumes it is not pragmatic for the HR to glance and evaluate all the resumes. So Resume Bots needs to be engaged.

Yet another problem that arises with bots is HR cannot program bot's evaluation criteria every time. So a Rule Engine can be developed that makes the work very much easier for HR. Rule Engine should be provided with a Dash Board .Our project aims at developing one such Rule Engine with a Dash-Board so as to enable non-programmers to create/alter/delete the Business Logic of the application.

CHAPTER 1

INTRODUCTION

1.1General

In computing, a web application or web app is a client-server software application in which the client (or user interface) runs in a web browser. Web Application gives convenience of using a web browser as a client to update and maintain web applications without distributing and installing software on potentially thousands of client computers. Web applications use web documents written in a standard format such as HTML5, CSS, PHP, JQuery, Ajax, Bootstrap framework and JavaScript, which are supported by a variety of web browsers. Web applications can be considered as a specific variant of client-server software where the client software is downloaded to the client machine when visiting the relevant web page, using standard procedures such as HTTP. Client web software updates may happen each time the web page is visited. During the session, the web browser interprets and displays the pages, and acts as the universal client for any web application.

Web developers often use client-side scripting to add functionality, especially to create an interactive experience that does not require page reloading. Recently, technologies have been developed to coordinate client-side scripting with server-side technologies such as PHP. Ajax, a web development technique using a combination of various technologies, is an example of technology which creates a more interactive experience. An emerging strategy for application software companies is to provide web access to software previously distributed as local applications. Depending on the type of application, it may require the development of an entirely different browser-based interface, or merely adapting an existing application to use different presentation technology.

Your data is in your domain objects, the logic is in the rules. This is fundamentally breaking the OO coupling of data and logic, which can be an advantage or a disadvantage depending on your point of view. The upshot is that the logic can be much easier to maintain as there are changes in the future, as the logic is all laid out in rules. This can be especially true if the logic is cross-domain or multi-domain logic. Instead of the logic being spread across many domain objects or controllers, it can all be organized in one or more very distinct rules files.

1.2 Motivation

Our data is in your domain objects, the logic is in the rules. This is a fundamental break from the object-orientated coupling of data and logic, which can be an advantage or a disadvantage depending on your point of view. The advantage is that the logic can be much easier to maintain when there are changes in the future, because it is all laid out in rules. This can be especially true if the logic is cross-domain or multi-domain logic. Instead of the logic being spread across many domain objects or controllers, it can all be organized in one or more very distinct rules files.

1.3 Problem description

Now a days, identifying the best candidate for the company has become a crucial task and has become a tedious job. Moreover finding a candidate for a particular job from the list of lakhs of resumes is a very challenging task. Resume Bots will make this process simple by eradicating such a tiresome task which consumes money, effort and time.

1.4 Related Work

Literature Study

The use of web application frameworks can often reduce the number of errors in a program, both by making the code simpler, and by allowing one team to concentrate on the framework while another focuses on a specified use case. In applications which are exposed to constant hacking attempts on the Internet, security-related problems can be caused by errors in the program. Frameworks can also promote the use of best practices such as GET after POST. In addition, there is potential for the development of applications on Internet operating systems, although currently there are not many viable platforms that fit this model.

HTML: Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. [1] Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with

cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold", leaving pure semantic HTML markup that

asserts "this text is a level 1 heading" without formatting code such as a <bold> tag indicating how such text should be displayed.

This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille, tactile devices. It can also be used to display the web page differently depending on the screen size or device on which it is being viewed. Although the author of a web page typically links to a CSS file within the markup file, readers can specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author has specified. If the author or the reader did not link the document to a style sheet, the default style of the browser will be applied. Another advantage of CSS is that aesthetic changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in one file, rather than by a laborious (and thus expensive) process of crawling over every document line by line, changing markup.

Related work in Resume Bots

A business rules engine is a software component that allows non-programmers to add or change business logic in a business process management system. A business rule is a statement that describes a business policy or procedure. Business logic describes the sequence of operations that is associated with data in a database to carry out the rule.

A business rules engine works by separating execution code for business rules from the rest of the business process management system. This allows the end user to change business rules without having to ask a programmer for help. When a change is made, the engine will evaluate the change's effect on other rules in the system and flag the user if there is a conflict.

1.4.1 System Requirements

Hardware required:

- RAM :2 GB and above
- Processor : dual core and above
- Hard Disk : 80 GB and above

Software required:

- Operating System : Windows XP / 7/8 ,Linux
- Browser :Chrome and firefox
- Front end Language : HTML5,CSS,JS,Ajax,
- Framework : Bootstrap
- Scripting Language :PHP
- Database :MySQL
- Software Tool : Wamp server 2.2

CHAPTER 2**OVERVIEW OF THE PROPOSED WORK****2.1 Introduction of problem and its related concepts (Existing system)****Problem:**

One of the prominent problem that persists to exist is recruitment of employees in an organization. It has never been a cakewalk for H.R managers of the organization. This process involves inviting applications, scrutinizing, filtering and selecting the resumes. The process of filtering involves finding the best candidate for a job description who has the required skill set. This process of filtering is not as easy as it seems to be. H.R's have to skim through lakhs and lakhs of application in order to find their employee. Our system caters the needs of avoiding this problem and reducing its complexity manifold.

Existing system

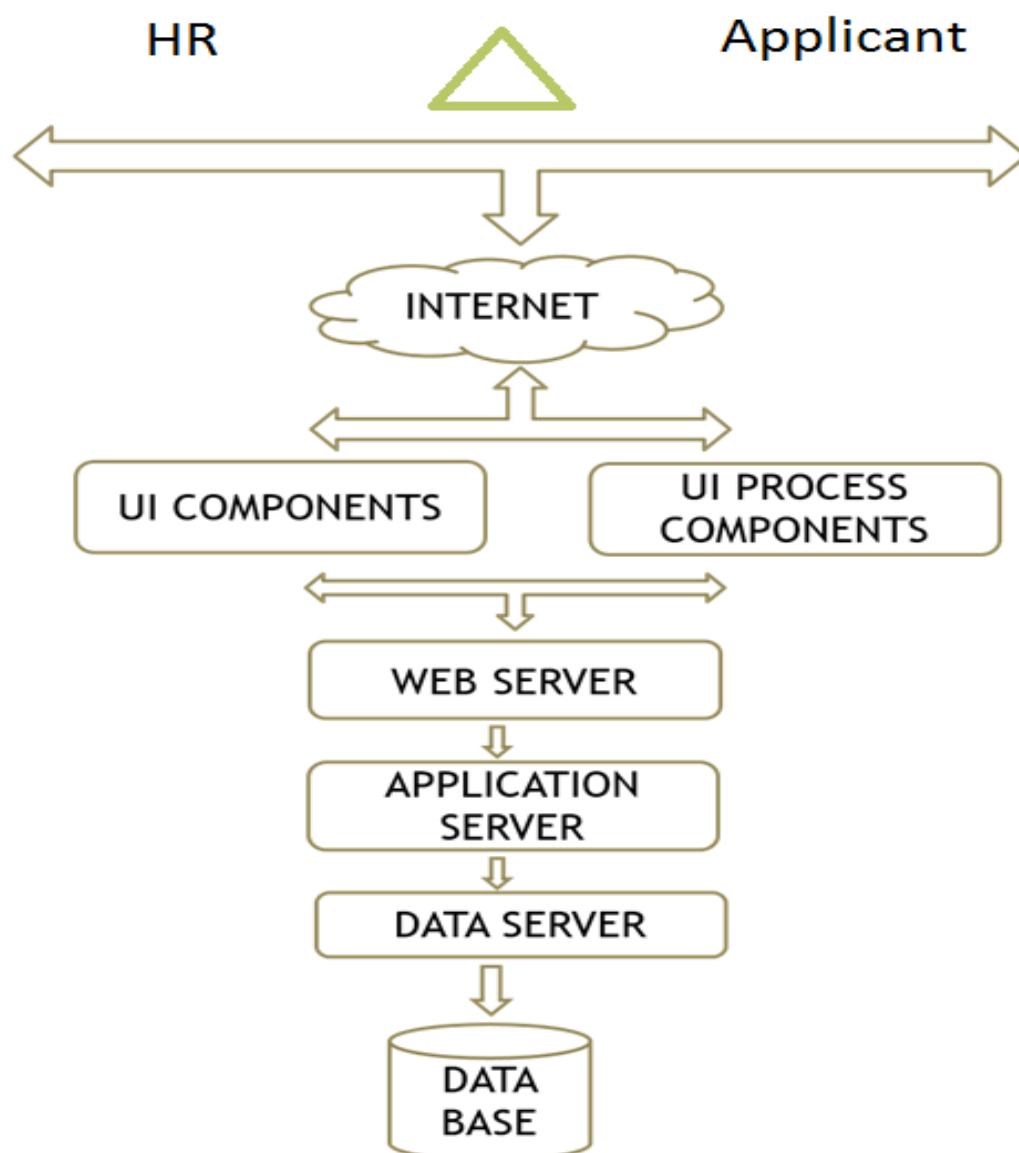
First, the software removes all formatting from the resume and scans for specific recognized keywords and key phrases. Next, it sorts the content of your resume into individual categories: Education, Contact Information, Skills, Work Experience. Then, the employer's list of desired skills and keywords are matched against the results of the resume to determine your potential value to the organization. Resumes with the highest scores relevant to the employer's specified keywords and phrases combined with your years of experience will be moved up for further review. In the end, the software simply scores the resume in order to determine which candidates are most qualified to move up the ladder for an actual human within the organization to review.

Reference

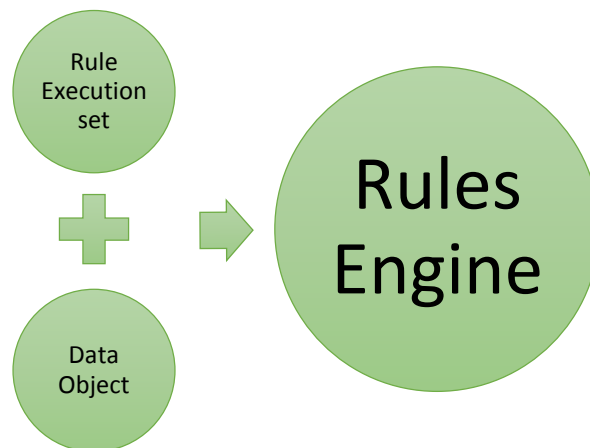
www.applicanttrackingsystem.co

2.2 Overview of the proposed system

Our System is aimed at developing a drag and drop interface for creating and defining rules over our preferred domain based in the Job Description. This drastically changes the way the recruiters use the resume filters. This system is aimed at making even non-programmers create their rules and filter resumes. Resume Bot will filter the resume accordingly to their expertise and Educational qualification etc. Resume bot will parse the resume, and identifies certain keywords then rank the resume according keywords. It won't support certain condition which is greedily needed by the recruiters. So, we proposing a rule engine to overcome this issue.



ARCHITECTURE DIAGRAM



Explanation of System architectural design:

An architecture description is the formal description and representation of the system, organized in a way that supports the reasoning about the structures and behaviors of the system. A System architecture can compromise system components, the externally visible properties of those components and the relationship between them. It can provide a plan from which the products can be procured, and systems developed that will work together to implement the overall system. The System architectural design above describes about the entire working of the web application. First the H.R and applicants will be provided with separate login credentials to access the system. Through the appropriate web browser they can view the web application. The User Interface components are loaded and the Graphical user interface will be visible to the people through the web server. Then the application components are loaded and the process of searching and booking the doctor based on their name, location, type, specialization by the user took place in the application server. Later the required data's are loaded in the data server and the in formations are retrieved from the appropriate database and provided to the user.

CHAPTER 3

ANALYSIS AND DESIGN

3.1 Brief Introduction

Systems development can generally be thought of as having two major components: Systems analysis and Systems design. Systems analysis is a process of collecting factual data, understand the processes involved, identifying problems and recommending feasible

suggestions for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals. System Analysis also includes subdividing of complex process involving the entire system, identification of data store and manual processes. Based on the user requirements and the detailed analysis of the existing system, the new system must be designed. This is the phase of system designing. It is the most crucial phase in the developments of a system. The logical system design arrived at as a result of systems analysis is converted into physical system design.

3.2. Requirement Analysis

3.2.1. Functional Requirements

User Requirement

The User Requirements Specification describes the business needs for what users require from the system. User Requirements Specifications are written early in the validation process, typically before the system is created. They are written by the system owner and end-users, with input from Quality Assurance.

Table 1: User Requirements for Resume Bots

REQ ID	REQUIREMENT DESCRIPTION	PRIORITY LEVEL
UR_01	A web-based interface would be more usable and requires less expertise than the command line interface.	1
UR_02	Always H.R and applicant should be provided with a valid login credentials such as username and password for accessing the system in order to prevent the legal attacks.	1
UR_03	The H.R should be able to create rules and save rules	2

UR_04	The H.R must be able to execute rules from the set of rules created	1
UR_05	The valid format of the resumes are docs, docx.	2
UR_06	H.R must be able add, delete and view list of users.	2
UR_07	The applicant must be able upload his qualification details.	1
UR_08	If the request is accepted successfully, the status have to be updated as success	1
UR_09	The H.R can view the rules created	1
UR_10	Searching is based on the rule group selected	3
UR_11	The Applicant can create # tag.	2
UR_12	The H.R can respond to the created # tags.	2
UR_13	The H.R can add or delete # tags.	3
UR_14	The user can logout once the purpose is over.	3

3.2.2. Non-Functional Requirements

Non-functional requirements define the overall qualities or attributes of the resulting system and it place restrictions on the product being developed, the development process, and specify external constraints that the product must meet.

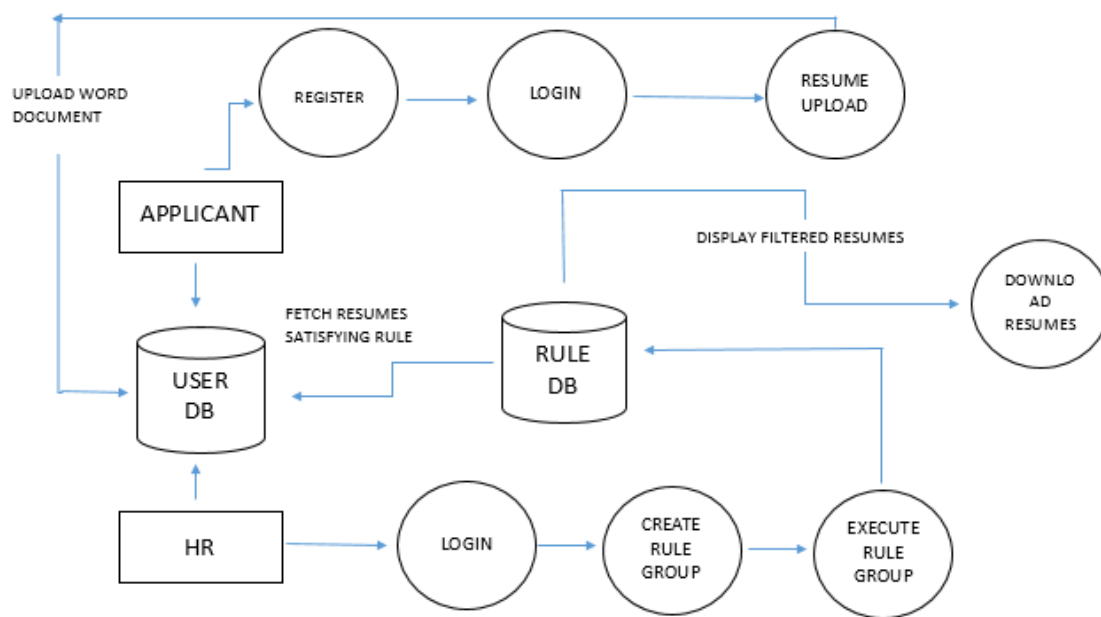
Table 2: Non-Functional Requirement for Resume Bots

REQ ID	REQUIREMENT NAME	REQUIREMENT DESCRIPTION
NFR_01	Reliability	The system must be able to perform its required functions under stated conditions for a specific period of time.
NFR_02	Performance	Performance requirements concern the speed of operation of a system.
NFR_03	Security	Unauthorized access to the system and its data is not allowed.
NFR_04	Safety	Ensure the integrity of the system from accidental or malicious damage
NFR_05	Usability	The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of system
NFR_06	Availability	The system should be available for service 24/7 when requested by end-users.
NFR_07	Adaptability	The ability to change the system to deal with additional application domain concepts.

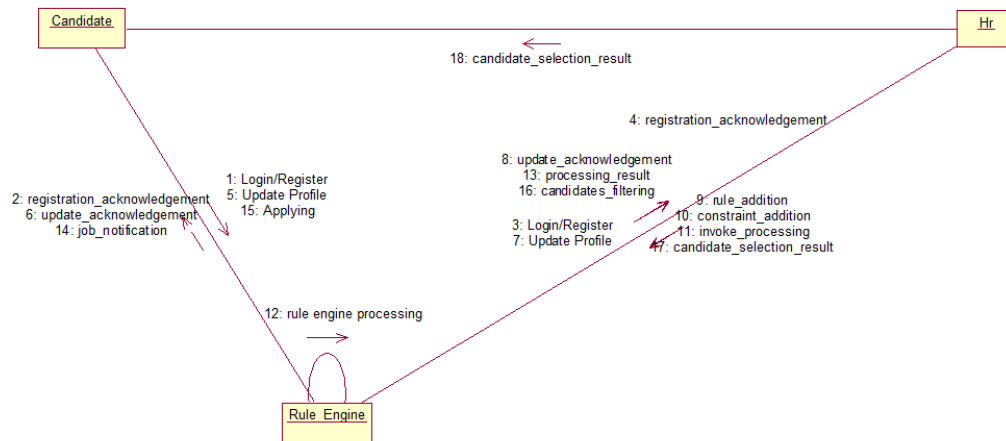
NFR_08	Maintainability	The ability to change the system to deal with new technology or to fix defects.
NFR_09	Portability	The ease with which a system or component can be transferred from one environment to another.

3.3. DETAILED DESIGN

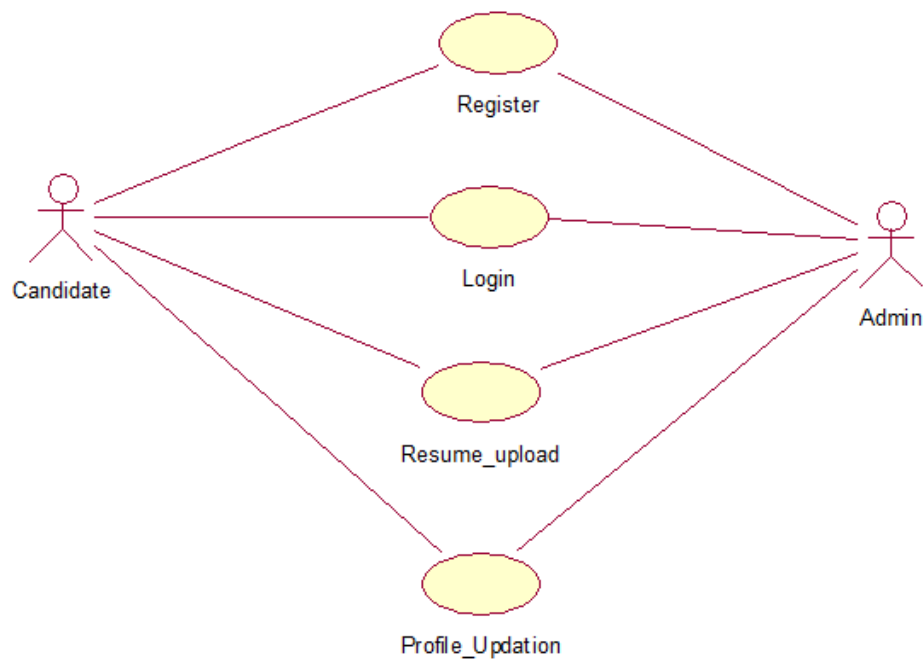
This context diagram gives the total description of the project “Resume Bots”. The context diagram is the most basic dataflow diagram that shows the interaction between the system and the user.



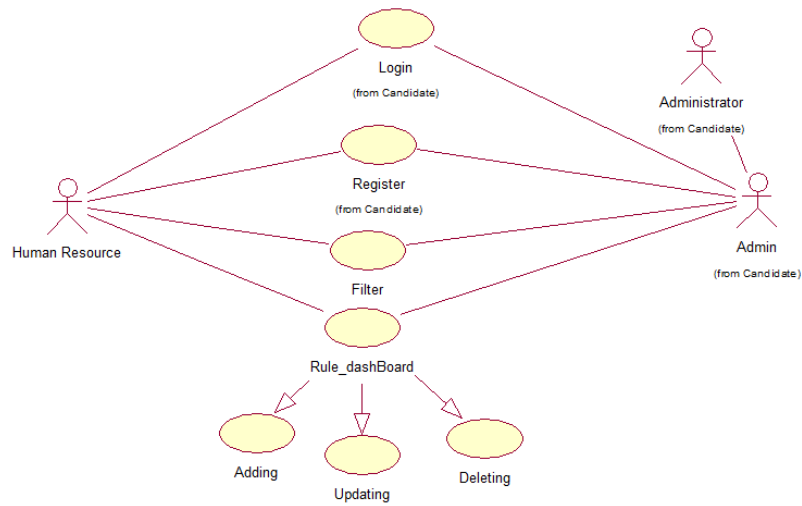
BLOCK DIAGRAM



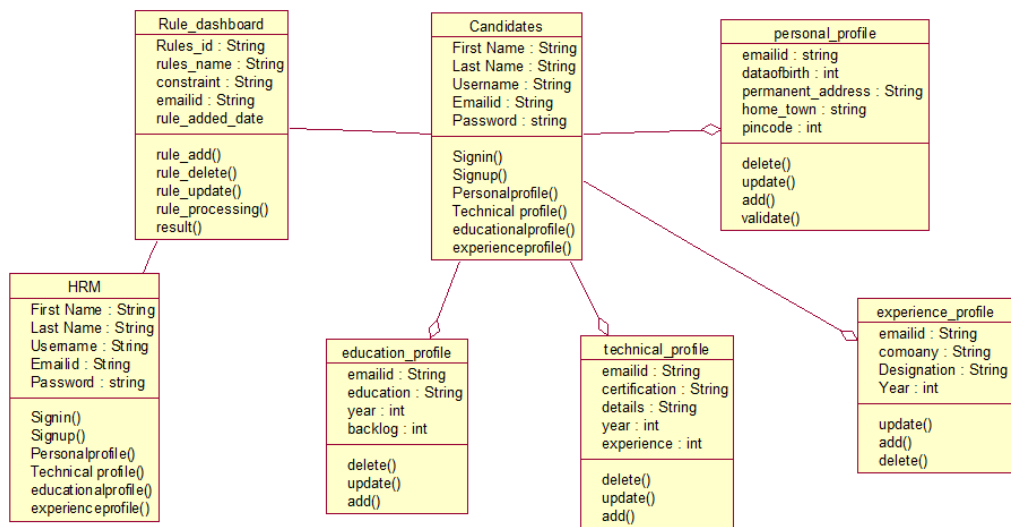
COLLABORATION DIAGRAM



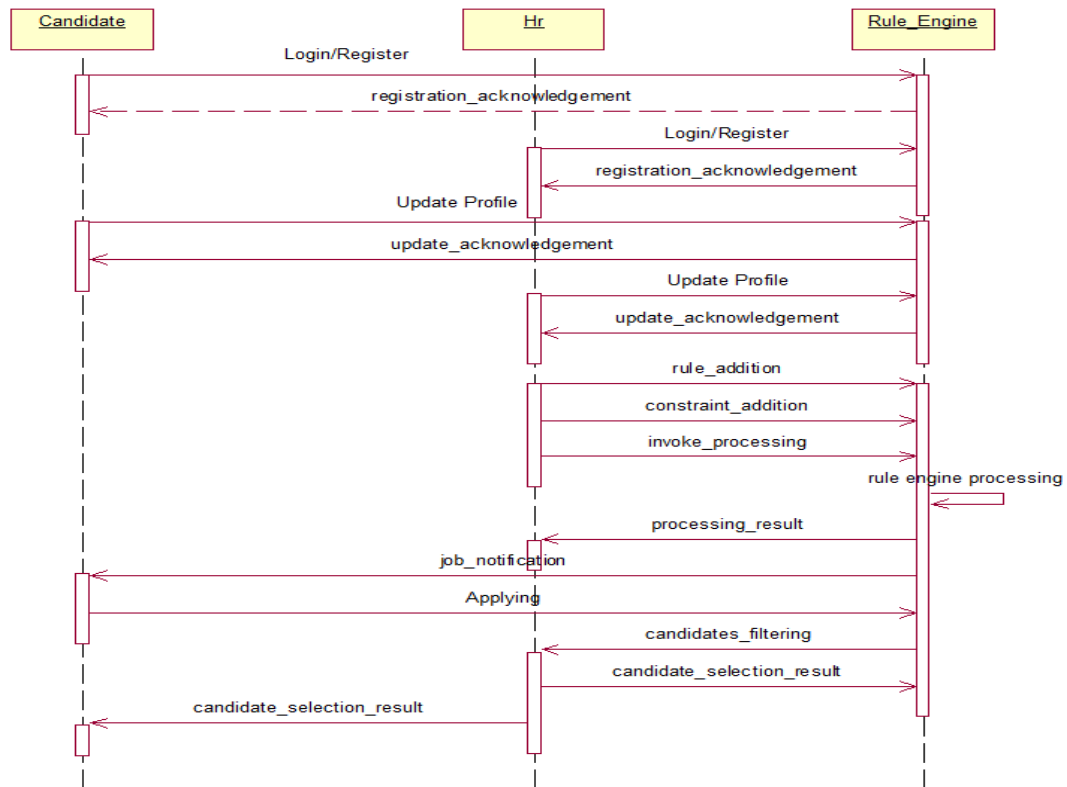
APPLICANT USECASE



HR USECASE



CLASS DIAGRAM



SEQUENCE DIAGRAM

3.4. Module Description

3.4.1. Login

Login module comprises of two segment, one designated for the applicant and other for H.R to Login.

3.4.2. Add Rule

This module is designated for the H.R to create a rule. Each rule has to be created with list of keywords that we add to database. Once the rule is defined it is named with a suitable name according to the user's wish.

3.4.3. Dashboard

The Dashboard is where H.R gets access to all the rules created. The H.R can execute a rule group from dash board. H.R can either execute or delete the set of rules from the dashboard

3.4.4. Hashtags

The H.R and applicant can create hashtags to add their query over some domain. Hashtags are

3.4.5 Add Users

This module allows H.R to add users to the system.

3.4.6. Delete User

This module allows H.R to delete existing user.

3.4.7. Application

This module allows applicant to update his qualification details and resume.

CHAPTER-4 IMPLEMENTATION

4.1. Tools Used

Software Testing Tool: Selenium Ide

Selenium IDE is an integrated development environment for Selenium scripts. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests. Selenium IDE includes the entire Selenium Core, allowing you too easily and quickly record and play back tests in the actual environment that they will run in.

Reason for selecting selenium ide:

- Easy record and playback.

- Intelligent field selection will use IDs, names, or XPath as needed.
- Auto complete for all common Selenium commands.
- Walk through tests.
- Debug and set breakpoints.
- Save tests as HTML, Ruby scripts, or any other format.
- Support for Selenium user-extensions.js file.
- Option to automatically assert the title of every page.

PHP scripting language:

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. While PHP originally stood for Personal Home Page it now stands for Hypertext Preprocessor. PHP code can be simply mixed with HTML code, or it can be used in combination with various template engines and web frameworks. PHP code is usually processed by a PHP interpreter, which is usually implemented as a web server's native module or a Common Gateway Interface (CGI) executable. After the PHP code is interpreted and executed, the web server sends the resulting output to its client, usually in the form of a part of the generated web page; for example, PHP code can generate a web page's HTML code, an image, or some other data. PHP has also evolved to include a command-line interface (CLI) capability and can be used in standalone graphical applications. The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge. PHP 5 included new features such as improved support for object-oriented programming, the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements. PHP 5 became the only stable version under development. Late static binding had been missing from PHP and was added in version 5.3. Over time, PHP interpreters became available on most existing 32-bit and 64-bit operating systems, either by building them from the PHP source code, or by using pre-built binaries. For the PHP versions 5.3 and 5.4, the only available Microsoft Windows binary distributions were 32-bit x86 builds, requiring Windows 32-bit compatibility mode while using Internet Information Services (IIS) on a 64-bit Windows platform. PHP version 5.5 made the 64-bit x86-64 builds available for Microsoft Windows.

Reason for choosing php:

PHP is available at free of cost under PHP General Public License and most of its associative required software's like MySQL, Text Editors and Apache Server are also freely available, so it proves very cost effective for the developers. PHP Platform independent can run on Windows Linux or Mac servers. It provides high compatibility with leading operating systems and web servers such as thereby enabling it to be easily deployed across several different platforms.

Cascading style sheets:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpage, user interfaces for web applications, and user interfaces for many mobile applications. CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, and enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate. css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file.

Reason for choosing css:

CSS helps to make the changes of the layout and design of the site very easily. The file size of the CSS is very small hence your website takes minimal loading time. The external CSS has made easy for the visitors who want to see only the content from the website. By creating the CSS, you can make the web design flexible. Layout and position of navigation can be completely consistent across a site.

WAMPP server:

WampServer refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, OpenSSL for SSL support, MySQL database and PHP programming language. The acronym WAMP refers to a set of free open source applications, combined with Microsoft Windows, which are commonly used in Web server environments. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. Any computer can be turned into a Web server by installing server software like WAMP, XAMP etc. WAMP stands on Windows, Apache HTTP Server, MySQL and PHP. in XAMPP, X meaning cross-platform it runs any of four different operating system such as Microsoft Windows, Linux, Solaris, and Mac OS X, Apache HTTP Server, MySQL, PHP and Perl . There are many Web server software applications, including public domain software from NCSA and Apache, and commercial packages from Microsoft, Netscape and others. WampServer is a collection of programs you can use to turn your regular desktop PC to a fully compatible web server with HTTP, PHP, MySQL, PHPMyAdmin, SQLBuddy applications.

Reason for choosing wamp server:

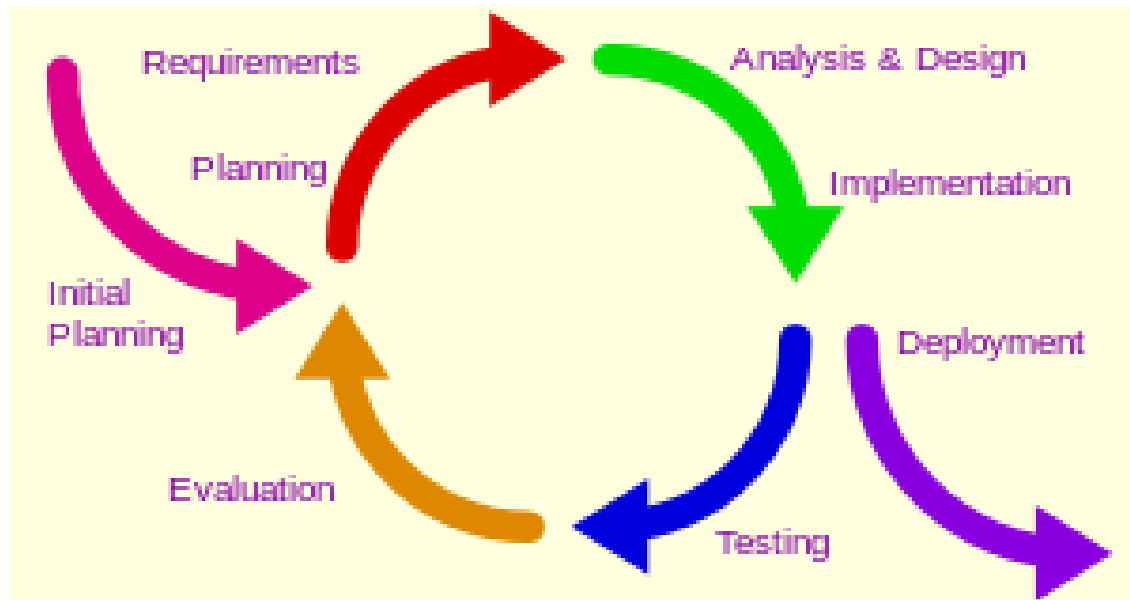
It is easy configurable with the built-in tools. Also it is structured in the way that you have everything you need at a click distance. The configuration screens provide extra information how settings should look like. The another thing would be that the packed applications are configured to be compatible between them, all being automatic configured, you just unpack your script, database, application and run the installer .

4.2 Methodology

Resume Bots uses Iterative development model. **Iterative and Incremental development** is any combination of both iterative design or iterative method and incremental build model for software development. The combination is of long standing and has been widely suggested for large development efforts. For example, the 1985 DOD-STD-2167 mentions (in section 4.1.2): "During software development, more than one iteration of the software development cycle may be in progress at the same time." and "This process may be described as an 'evolutionary acquisition' or 'incremental build' approach." The relationship between iterations and increments is determined by the overall software development methodology and software development process. The exact number and nature of the particular

incremental builds and what is iterated will be specific to each individual development effort. Iterative and incremental development are essential parts of the Modified waterfall models, Rational Unified Process, Extreme Programming and generally the various agile software development frameworks.

It follows a similar process to the plan-do-check-act cycle of business process improvement.



The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental), allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system. Learning comes from both the development and use of the system, where possible key steps in the process start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration process, a project control list is created that contains a record of all tasks that need to be performed. It includes such items as new features to be

implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The iteration involves the redesign and implementation of iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project control list. The level of design detail is not dictated by the iterative approach. In a light-weight iterative project the code may represent the major source of documentation of the system; however, in a critical iterative project a formal Software Design Document may be used. The analysis of an iteration is based upon user feedback, and the program analysis facilities available. It involves analysis of the structure, modularity, usability, reliability, efficiency, & achievement of goals. The project control list is modified in light of the analysis results.

Phases

Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross-discipline work, from the requirements to the deployment. The Unified Process groups increments/iterations into phases: inception, elaboration, construction, and transition.

- Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.
- Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
- Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
- Transition delivers the system into the production operating environment.

Each of the phases may be divided into 1 or more iterations, which are usually time-boxed rather than feature-boxed. Architects and analysts work one iteration ahead of developers and testers to keep their work-product backlog full.

Usage

Many examples of early usage are provided in Craig Larman and Victor Basili's article "Iterative and Incremental Development: A Brief History",^[3] with one of the earliest being NASA's 1960s Project Mercury.

Another is an "early and striking example of a major IID success is the very heart of NASA's space shuttle software—the primary avionics software system, which FSD built from 1977 to 1980. The team applied IID in a series of 17 iterations over 31 months, averaging

around eight weeks per iteration. Their motivation for avoiding the waterfall life cycle was that the shuttle program's requirements changed during the software development process".

Some organizations, such as the US Department of Defense, have a preference for iterative methodologies, starting with MIL-STD-498 "clearly encouraging evolutionary acquisition and IID".

The current DoD Instruction 5000.2, released in 2000, states a clear preference for IID: "There are two approaches, evolutionary and single step [waterfall], to full capability. An evolutionary approach is preferred. ... [In this] approach, the ultimate capability delivered to the user is divided into two or more blocks, with increasing increments of capability...software development shall follow an iterative spiral development process in which continually expanding software versions are based on learning from earlier development. It can also be done in phases.

Implementation guidelines

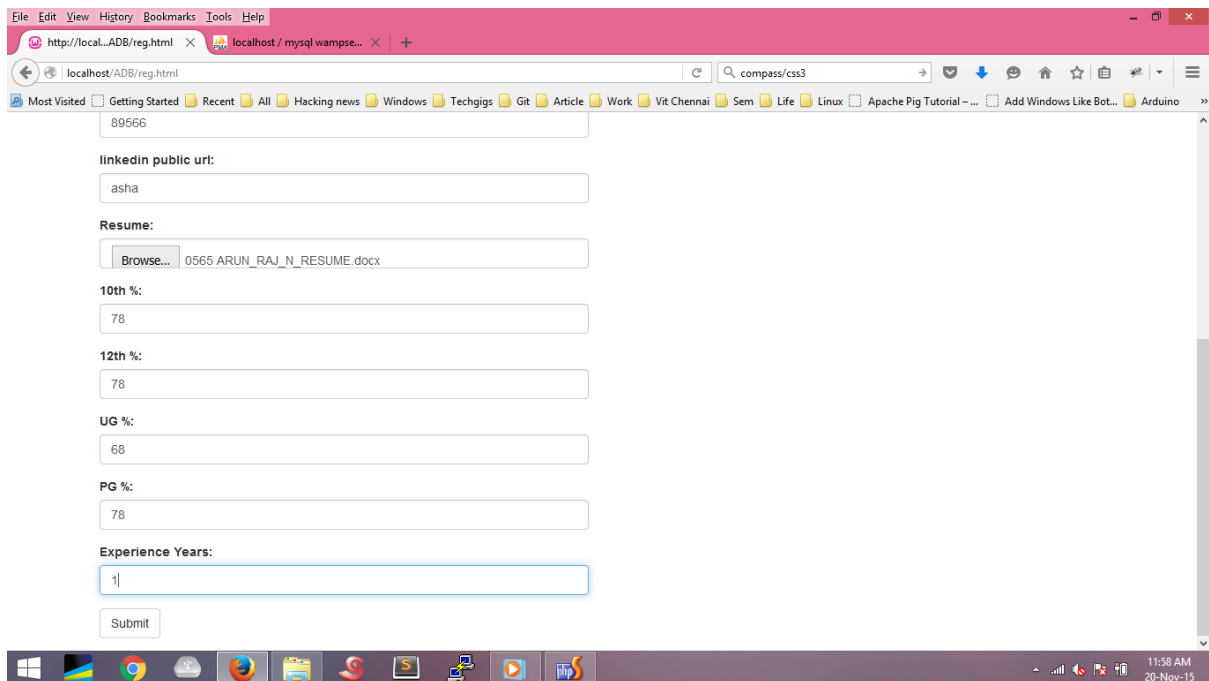
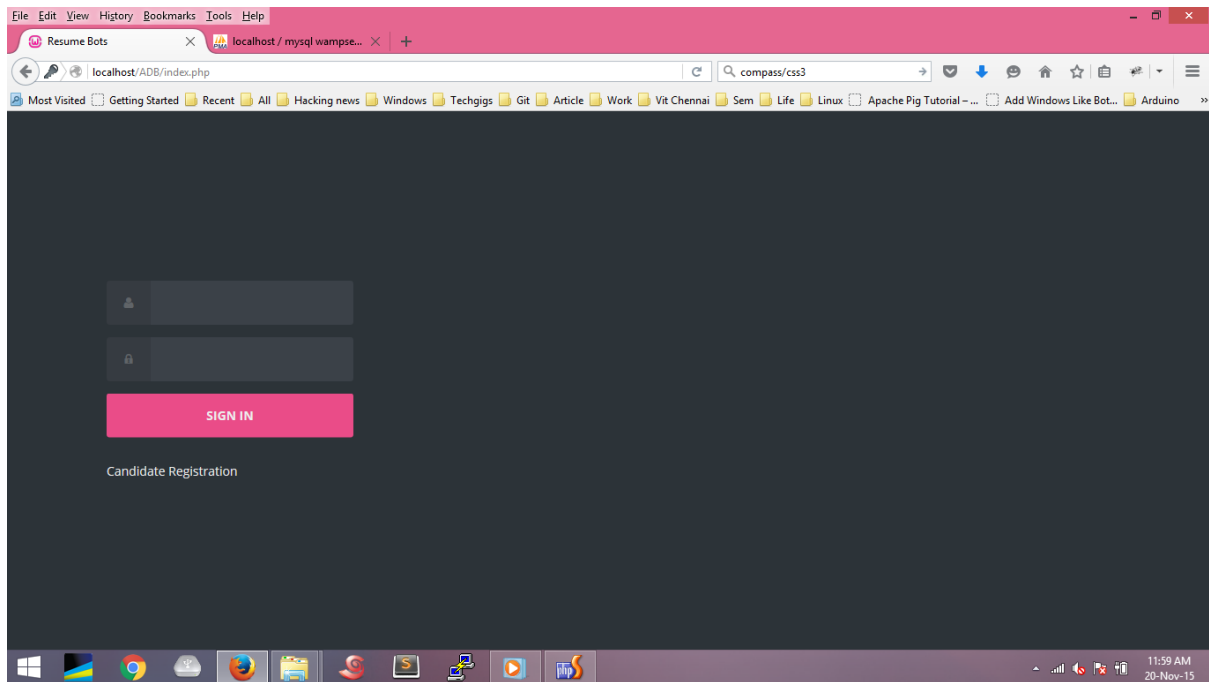
Guidelines that drive the implementation and analysis include:

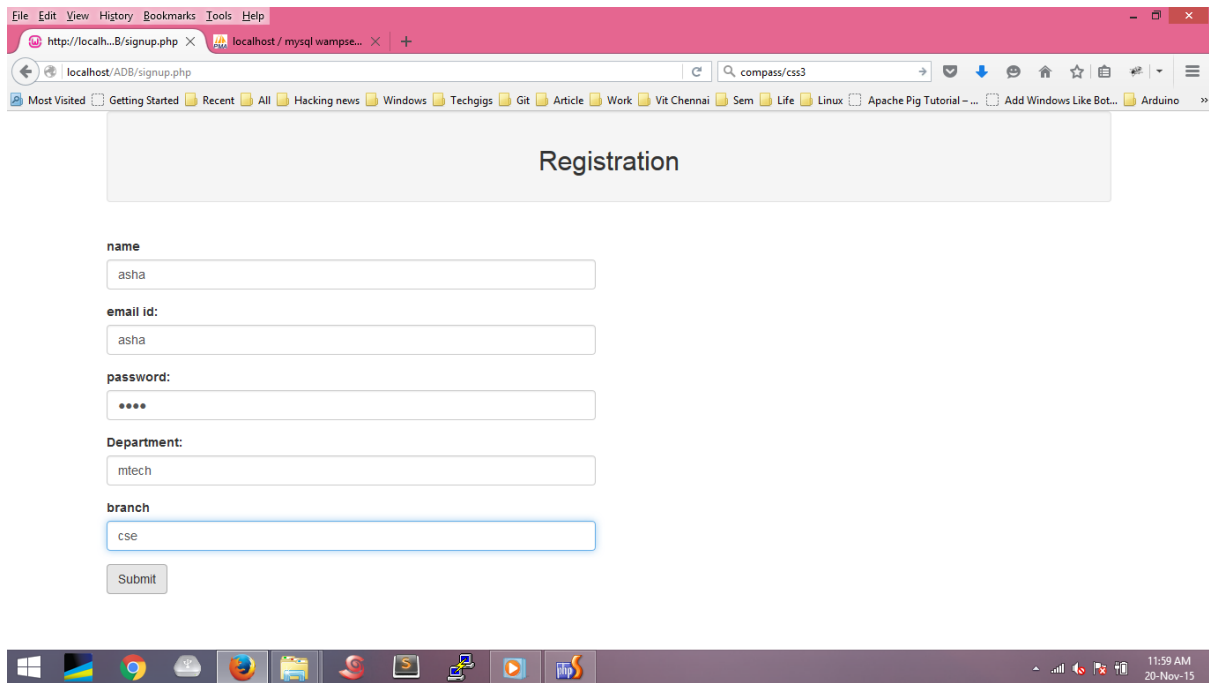
- Any difficulty in design, coding and testing a modification should signal the need for redesign or re-coding.
- Modifications should fit easily into isolated and easy-to-find modules. If they do not, some redesign is possibly needed.
- Modifications to tables should be especially easy to make. If any table modification is not quickly and easily done, redesign is indicated.
- Modifications should become easier to make as the iterations progress. If they are not, there is a basic problem such as a design flaw or a proliferation of patches.
- Patches should normally be allowed to exist for only one or two iterations. Patches may be necessary to avoid redesigning during an implementation phase.
- The existing implementation should be analyzed frequently to determine how well it measures up to project goals.
- Program analysis facilities should be used whenever available to aid in the analysis of partial implementations.
- User reaction should be solicited and analyzed for indications of deficiencies in the current implementation.

4.3 Implementation

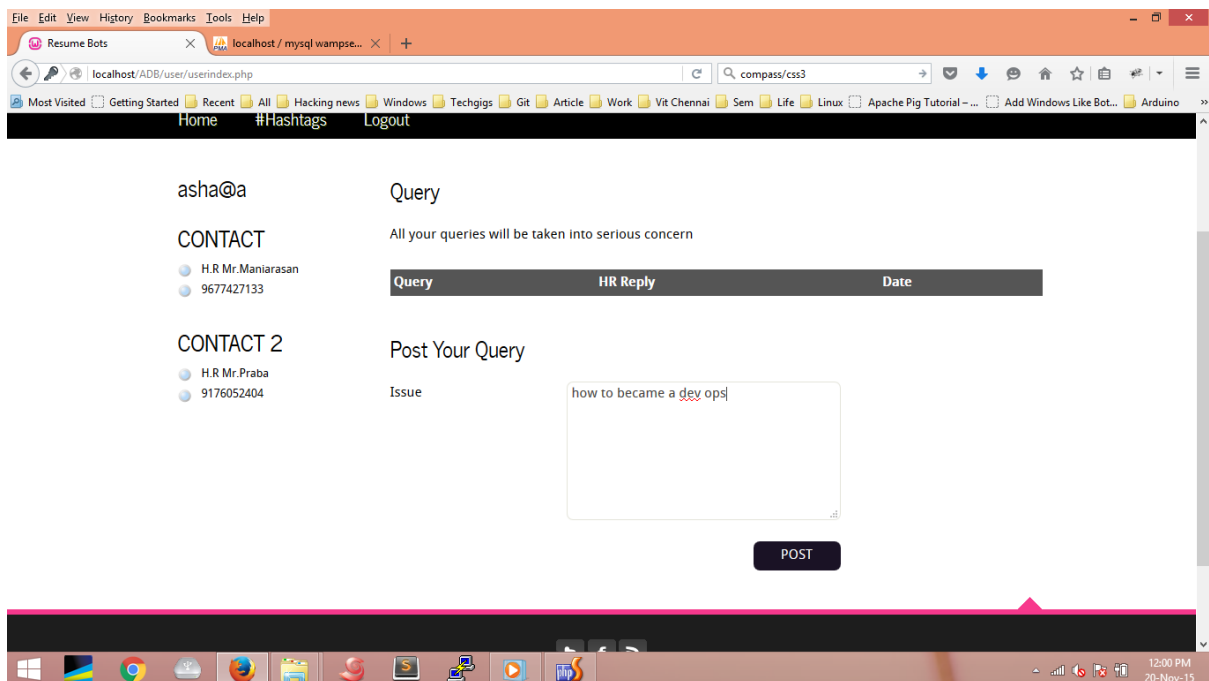
Screen shots:

Register:





User Login :



User Query Posting:

The screenshot shows a web browser window with the URL `localhost/ADB/user/index.php`. The page has a dark header with the ResumeBots logo and navigation links: Home, #Hashtags, and Logout. The main content area is divided into two columns. The left column contains the user's profile information for 'asha@a', including contact details and two contact entries. The right column is titled 'Query' and contains a message: 'All your queries will be taken into serious concern'. Below this is a table with the following data:

Query	HR Reply	Date
how to become a dev ops	Awaiting Admin Response	2015-11-20

Below the table is a section titled 'Post Your Query' with a label 'Issue' and an empty text input field.

The screenshot shows the same web browser window, but the URL is `localhost/ADB/user/another_page.php`. The page has the same header. The main content area is divided into two columns. The left column contains the user's profile information for 'asha@a', including contact details and two contact entries. The right column is titled 'HASH TAGS' and contains a form with the following fields:

HASH:

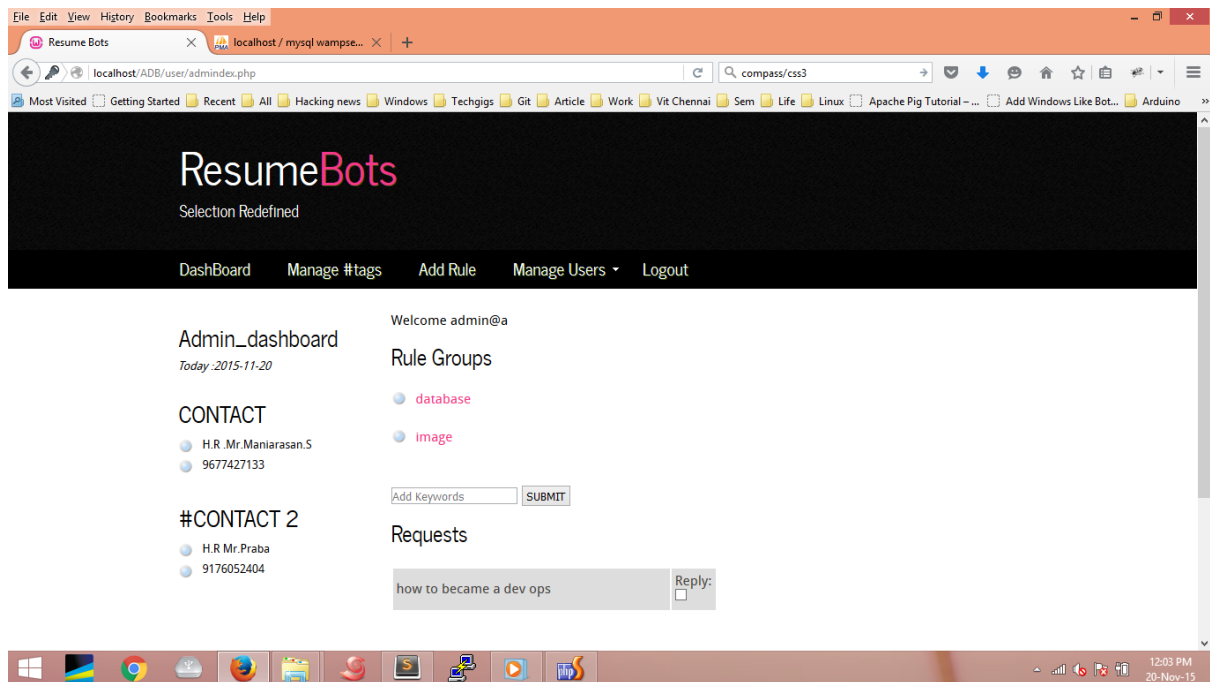
COMMENT:

ADD HASH

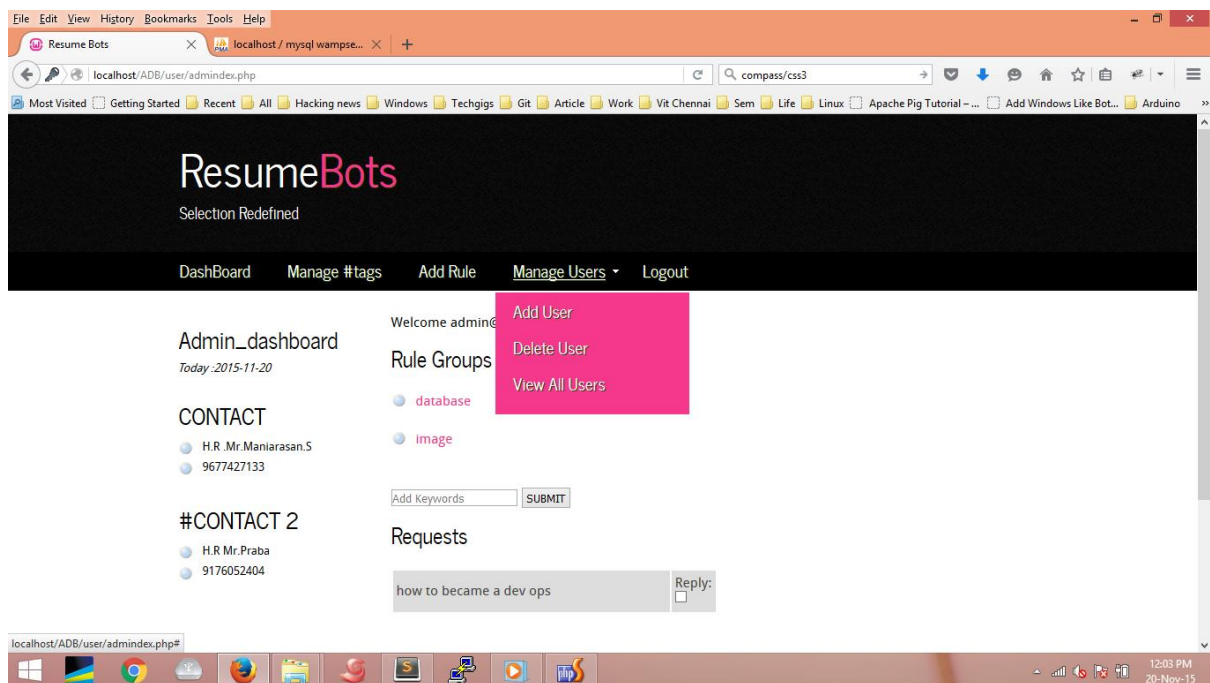
Below the form is a table with the following data:

HASHtag	Creator	Comment	Date
Sr.SW Dev	mani@vit.edu	what experience is required for becoming a senior	2015-11-19
analyst	asha@a	can analyst switch over to development	2015-11-20

Admin Dashboard:

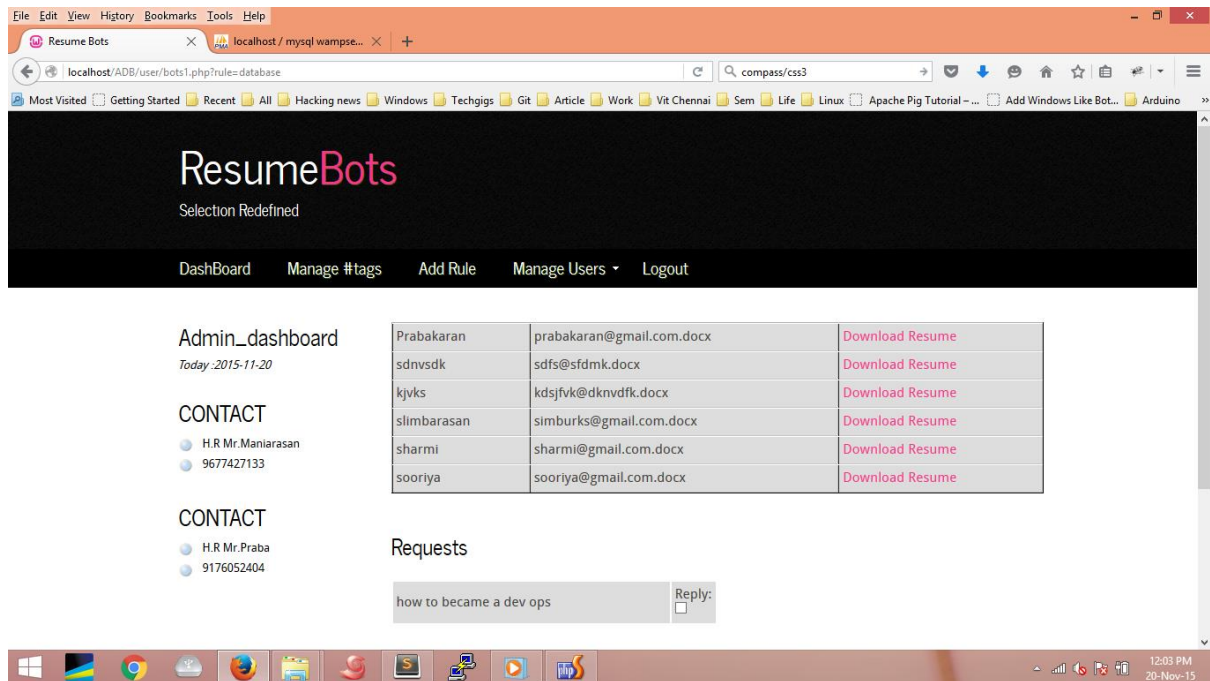


The screenshot shows the ResumeBots Admin Dashboard in a web browser. The browser's address bar displays 'localhost/ADB/user/adminindex.php'. The dashboard has a dark header with the 'ResumeBots' logo and the tagline 'Selection Redefined'. Below the header is a navigation bar with links: 'DashBoard', 'Manage #tags', 'Add Rule', 'Manage Users', and 'Logout'. The main content area is divided into two columns. The left column contains a 'Welcome admin@a' message, 'Rule Groups' (listing 'database' and 'image'), an 'Add Keywords' input field with a 'SUBMIT' button, and a 'Requests' section with a text input 'how to became a dev ops' and a 'Reply:' button. The right column contains a 'Admin_dashboard' section with a timestamp 'Today :2015-11-20', a 'CONTACT' section with two entries (H.R. Mr. Maniarasan.S and 9677427133), and a '#CONTACT 2' section with two entries (H.R Mr. Praba and 9176052404). The Windows taskbar at the bottom shows various application icons and the system clock indicating 12:03 PM on 20-Nov-15.



This screenshot is similar to the one above, but with the 'Manage Users' dropdown menu open. The menu is pink and contains three options: 'Add User', 'Delete User', and 'View All Users'. The rest of the dashboard layout, including the header, navigation bar, and main content sections, remains the same. The browser's address bar still shows 'localhost/ADB/user/adminindex.php'.

Rule Output:



4.4 Unit Test Cases

Table 4: Test cases for Resume Bots.

Test Case Id	Check Field	Objective	Expected Output	Expected Output	Result
TC-01	Website	Open the website	The home page should have to be displayed.	The home page is displayed	Pass
TC-02	New user registration	Fill all the fields and press register	The values should store in database and	Successfully registered	pass

			show message of successfully registered		
TC-03	Check the H.R login	enter the correct username and password	Should direct to the home page of the website.	Successfully Logged in	Pass
TC-04	Add rules	Drag and drop the keywords to to filter.	The rule should be added.	Successfully Updated	pass
TC-05	Dashboard	Click on Dashboard	Should successfully display all the rules added	Successfully Displayed	Pass
TC-06	Upload Resume	Click the filepicker's "Browse" button.	File picker should choose path and upload the resume.	Successfully Uploaded	pass
TC-07	Add user	Fill the details of user and click add user button.	Users should be updated in database	Successfully Updated	Pass

TC-08	Create Hash	Type the name of the hash and click create hash	The status has to be updated to database	Status displayed successfully.	Pass
TC-09	Delete hash	Type the name of the hash and click delete	The hash should be deleted in database.	Successfully Deleted.	pass
TC-10	Logout	Click on the Logout button.	The user should be logged out and redirected to index page.	Successfully logged out.	Pass

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Results

The system had produced the expected results with well-designed user interface and a very simple execution of functionalities. The system's layout, buttons, its outlook are specially designed and it gives a very appealing look. All the functionalities are performing well like expected and none of the functionalities go disabled anywhere. The data produced are more accurate and the

system gives precise outputs. Thus the system is used to book the doctors through online, the users can store their records and also enables user the view the most frequently visited doctors list.

5.2 Performance Analysis

The simplest applications have the web and app tiers combined while more complex ones may have multiple application tiers (called “middleware”) as well as multiple datastores. The Front end refers to the web tier that generates the html response for the browser. The Back end refers to the server components that are responsible for the business logic. Note that in architectures where a single web/app server tier is responsible for both the front and back ends, it is still useful to think of them as logically separate for the purposes of performance analysis.

Front End Performance:

When measuring front end performance, we are primarily concerned with understanding the response time that the user (sitting in front of a browser) experiences. This is typically measured as the time taken to load a web page. Performance of the front end depends on the following:

- Time taken to generate the base page
- Browser parse time
- Time to download all of the components on the page (css,js,images,etc.)
- Browser render time of the page

For most applications, the response time is dominated by the 3rd bullet above i.e. time spent by the browser in retrieving all of the components on a page. As pages have become increasingly complex, their sizes have mushroomed as well – it is not uncommon to see pages of 0.5 MB or more. Depending on where the user is located, it can take a significant amount of time for the browser to fetch components across the internet.

Front end Performance Tools:

Front-end performance is typically viewed as waterfall charts produced by tools such as the Firebug Net Panel. During development, firebug is an invaluable tool to understand and fix client-side issues. However, to get a true measure of end user experience on production systems, performance needs to be measured from points on the internet where your customers typically are. Many tools are available to do this and they vary in price and functionality. Do your research to find a tool that fits your needs.

Back End Performance

The primary goal of measuring back end performance is to understand the maximum throughput that it can sustain. Traditionally, enterprises perform “load testing” of their applications to ensure they can scale. I prefer to call this “scalability testing”. Test clients drive load via bare-bones HTTP clients and measure the throughput of the application i.e. the number of requests per second they can handle. To increase the throughput, the number of client drivers need to be increased until the point where throughput stops to increase or worse stops to drop-off.

For complex multi-tier architectures, it is beneficial to break-up the back end analysis by testing the scalability of individual tiers. For example, database scalability can be measured by running a workload just on the database. This can greatly help identify problems and also provides developers and QA engineers with tests they can repeat during subsequent product releases.

Many applications are thrown into production before any scalability testing is done. Things may seem fine until the day the application gets hit with increased traffic (good for business!). If the application crashes and burns because it cannot handle the load, you may not get a second chance.

Back End Performance Tools

Numerous load testing tools exist with varying functionality and price. There are also a number of open source tools available. Depending on resources you have and your budget, you can also outsource your entire scalability testing.

Web Performance Analysis Receive a thorough analysis of your web applications. We’ll diagnose, analyze and provide custom recommendations or provide a detailed comparison of your website’s performance versus your competitors’ sites. As your analysis and assessment progress, your lead engineer will run a series of tests against your web applications, analyze performance metrics and generate a report. When the analysis is completed, he/she will prepare a summary document and review it with you carefully.

Typical Metrics Used Performance by Browser:

- Internet Explorer
- Chrome
- Firefox

Application Source Code Analysis:

- Full-page breakdown analysis
- JavaScript optimization

- Cascading style sheet (CSS) optimization
- HTML /image optimization
- Third-party content and advertisements

Geographical Analysis:

- Performance analysis by region (city, state, country and continent) with the Neustar global agent network in more than 100 major cities worldwide
- Latency analysis

Weekly Trending Analysis:

- Weekly snapshots searching for bottlenecks or performance degradation trends
- Statistical variation
- Worst hour/worst day analysis

CHAPTER 6

CONCLUSIONS AND FUTURE ENHANCEMENTS

We created a system named “Resume Bots” that includes almost every possible functionality in order to make the system hassle free, also efficient and effective. One of the most useful functions in this system is grouping rules and evaluating the resume. But still there are few more functionalities like passing parameters to rules, providing web services to the functionality. In future this project can be extended easily to include more options thereby making this system best to accomplish their goal.

REFERENCES

- [1] Elastic search, Apache Open Source Search Engine
- [2] Elastic Search PHP API, www.github.com
- [3] JQuery UI, www.jqueryui.com
- [4] ZOHO Recruit, www.zoho.com

[5] Widely Used Application Tracking System, www.capterra.com/applicant-tracking-software

APPENDIX –I

Source Code

Adminindex.php

```
<!DOCTYPE HTML>
<html>

<head>
  <title>Resume Bots</title>
  <meta name="description" content="website description"/>
  <meta name="keywords" content="website keywords, website keywords"/>
  <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
  <link rel="stylesheet" type="text/css" href="css/style.css"/>
  <link rel="stylesheet" type="text/css" href="css/style.css"/>
  <!-- modernizr enables HTML5 elements and feature detects -->
  <script type="text/javascript" src="js/modernizr-1.5.min.js"></script>
  <script>
    function toggle(id) {
      if (document.getElementById(id).style.display == 'none') {
        document.getElementById(id).style.display = 'block';
      } else {
        document.getElementById(id).style.display = 'none';
      }
    }
  </script>
</head>

<body>
<div id="main">
  <header>
    <div id="logo">
```

```

<div id="logo_text">
    <!-- class="logo_colour", allows you to change the colour of the text -->
    <h1><a href="adminindex.php">Resume<span
class="logo_colour">Bots</span></a></h1>

    <h2>Selection Redefined</h2>

</div>
</div>
<nav>
<div id="menu_container">
    <ul class="sf-menu" id="nav">
        <li><a href="adminindex.php">DashBoard</a></li>
        <li><a href="managehash.php">Manage #tags</a></li>

        <li><a href="addrule.php">Add Rule</a></li>

        <li><a href="#">Manage Users</a>
            <ul>
                <li><a href="adduser.html">Add User</a></li>
                <li><a href="delete.html">Delete User</a></li>
                <li><a href="view.php">View All Users</a></li>

            </li>
        </ul>
        </li>
        <li float="right"><a href="logout.php">Logout</a></li>
    </ul>
</div>
</nav>
</header>
<div id="site_content">

```

```

<div id="sidebar_container">
  <div class="sidebar">
    <h3>Admin_dashboard</h3>
    <h5>Today :<?php echo date("Y-m-d"); ?></h5>

  </div>
  <div class="sidebar">
    <h3>CONTACT</h3>
    <ul>
      <li>H.R .Mr.Maniarasan.S</li>
      <li>9677427133</li>

    </ul>
  </div>
  <div class="sidebar">
    <h3>#CONTACT 2</h3>
    <ul>
      <li>H.R Mr.Praba</li>
      <li>9176052404</li>

    </ul>
  </div>
</div>
<div class="content">

  <?php
  include "mysqlidb.php";
  session_start();
  $email = $_SESSION['name'];

  //$_SESSION['email'];

  echo "Welcome " . $email;

  $query = "select * from rule where email='$email'";
  $r = mysqli_query($con, $query);

```

?>

```
<ul>
    <h2>Rule Groups</h2>
    <?php while ($row = $r->fetch_assoc()) {
        echo "<li><a href=bots1.php?rule=" . $row['name'] . ">" . $row['name'] .
"</a>&nbsp;&nbsp;<a href=delete.php?rule=" . $row['name'] . ">DELETE</a></li></br>";
    } ?>
</ul>
<div>
    <div>
        <input placeholder="Add Keywords" type="text" id="addkey"/>
        <input onclick="sendarr()" value="SUBMIT" type="submit" >
    </div>
</div>
<script type="application/javascript">
    function sendarr()
    {
        var keys=document.getElementById("addkey").value;
        if(keys!="") {
            var xmlhttp = new XMLHttpRequest();
            xmlhttp.onreadystatechange = function () {
                if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
                    if (xmlhttp.responseText == 1) {
                        document.getElementById("addkey").value="";
                        alert("keyword added");
                    }
                    else
                    {
                        //alert(xmlhttp.responseText);
                    }
                }
            };
        }
    }
}
```

```

        xmlhttp.open("GET", "addkeywords.php?keys="+keys, true);
        xmlhttp.send();
    }
</script>

<h2>Requests</h2>

<?php

include 'connection.php';

$result = mysql_query("SELECT * FROM grievance where status=0");

while ($row = mysql_fetch_array($result)) {
    echo "<table width='50%' >";
    echo "<tr>";
    echo "<form action='reply.php' method='POST'><td>" . $row['grv'] . "<td
width='1'> <input type='hidden' value='" . $row['sl'] . "' name='id'>";
    ?>

    Reply:<input type="checkbox" onclick="toggle('Comments')"> <br/>
    <div id="Comments" style="display:none;">
        <textarea name="rep" cols="200" rows="3">Your Query will addressed at the
earliest possible duration</textarea>
        <input type="submit" value="Reply">
    </div>

    <?php
    echo "</tr>";
    echo "</table>";
    echo "</form><br>";

}

```

```

        mysql_close($con);
    ?>

</div>
</div>
<div id="scroll">
    <a title="Scroll to the top" class="top" href="#"></a>
</div>
<footer>
    <p>&nbsp;&nbsp;</p>

    <p><a href="index.html">Home</a> | <a href="examples.html">Examples</a> | <a
href="page.html">A Page</a> | <a
href="another_page.html">Another Page</a> | <a href="contact.php">Contact
Us</a></p>

    <p>Copyright &copy; Resume Bots | <a
href="http://www.css3templates.co.uk">VIT</a></p>
</footer>
</div>
<!-- javascript at the bottom for fast page loading -->
<script type="text/javascript" src="js/jquery.js"></script>
<script type="text/javascript" src="js/jquery.easing-sooper.js"></script>
<script type="text/javascript" src="js/jquery.sooperfish.js"></script>
<script type="text/javascript">
    $(document).ready(function () {
        $('ul.sf-menu').sooperfish();
        $('top').click(function () {
            $('html, body').animate({scrollTop: 0}, 'fast');
            return false;
        });
    });

```

</script>

</body>

</html>