SOFTWARE REQUIREMENT SPECIFICATION FOR CAMPUS RECRUITMENT AND CAREER GUIDANCE SYSTEM

Prepared by,

Akash O K
MCA A
Roll No : 5

1.INTRODUCTION

1.1 ABSTRACT

A college campus recruitment & career guidance system that consists of a student login, company login and an admin login. The project is beneficial for final year college students, various companies visiting the campus for recruitment and even the college placement officer. The software system allows the students to register their profiles and upload all their details including their marks onto the system. After student registration he/she can able to choose two options.

1. Campus recruitment :- The software system allows students to view a list of companies who have posted for vacancy

2. Career guidance: - The software system allows students to know about their eligible pg courses, Other Courses and the list of colleges and university who offer that courses and also able to know about the entrance exams.

3. The admin can check each student details and can remove faulty accounts. The system also consists of a company login where various companies visiting the college can view a list of students in that college and also their respective resumes. The admin has overall rights over the system and can moderate and delete any details not pertaining to college placement rules. The

system handles student as well as company data and efficiently displays all this data to respective sides

1.2MODULE DESCRIPTION

The modules in this system are:

1. Admin module

Under this module could manage all the functions of campus recruitment and career guidance. Admin can handle the student, company, career guidance, notification, reports

2. Student module

Under this module could register students, view interviews, apply for a post, etc. on recruitment and search universities, select courses, view scope of the course etc. on career guidance and to send feedbacks to the admin.

3. Company module

Under this module company could add or remove vacancies, select students based on their applications

2.SYSTEM ANALYSIS

2.1 SYSTEM REQUIREMENT SPECIFICATION

The following requirements are only the minimal requirements to run this utility more successfully and efficiently, there should be sufficient memory and software tools for efficient processing.

2.1.1 HARDWARE REQUIREMENTS:

Processor : Pentium IV or Above.
Main Memory : 2 GB RAM.
Cache Memory : 11,011,968 Bytes
Hard Disk Capacity : 20 GB or Above.
Virtual Memory : 32 MB

2.1.2 SOFTWARE REQUIREMENTS:

Front End : HTML, Js ,CSS
Server software : Python 3.6.7
Frame work : Django
Back End : MySQL

Operating System : Windows 7 or Above
Browser : Mozilla Firefox or Google Chrome

2.2 OVERVIEW OF PYTHON

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is a popular programming language. It was created in 1991 by Guido van Rossum. It is used for: web development (server-side), software development, mathematics, System scripting.
Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.
Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of application
Python's features include −
Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
Easy-to-read − Python code is more clearly defined and visible to the eyes.
Easy-to-maintain − Python's source code is fairly easy-to-maintain.
A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
Databases − Python provides interfaces to all major commercial databases.
GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

It supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

It supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Features of Python Programming

A simple language which is easier to learn: Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

Free and open-source: You can freely use and distribute Python, even for commercial use. Not only can you use and distribute softwares written in it,

you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

Portability: You can move Python programs from one platform to another, and run it without any changes. It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

Extensible and Embeddable: Suppose an application requires high performance. You can easily combine pieces of C/C++ or other language with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

A high-level, interpreted language: Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on. Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations

Object-oriented: Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively. With OOP, you are able to divide these complex problems into smaller sets by creating objects.

Large standard libraries to solve common tasks: Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQL db . Standard libraries in Python are well tested and used by hundreds of people

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension .py. Type the following source code in a test.py file −

Python Identifiers

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).
Python does not allow punctuation characters such as @, $, and % within identifiers. Python is a case sensitive programming language. Thus, Manpower and manpower are two different identifiers in Python.
Here are naming conventions for Python identifiers −
Class names start with an uppercase letter. All other identifiers start with a lowercase letter.
Starting an identifier with a single leading underscore indicates that the identifier is private.
Starting an identifier with two leading underscores indicates a strongly private identifier.
If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.
Applications of Python
Web Applications
You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS.
Sites like Mozilla, Reddit, Instagram and PBS are written in Python.
Scientific and Numeric Computing
There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on.
Also, the language is heavily used in machine learning, data mining and deep learning.
Creating software Prototypes
Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must.
However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.
Good Language to Teach Programming
Python is used by many companies to teach programming to kids and newbies.
It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use syntax.
Django Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source. Django is a high-level Python Web framework that encourages rapid development and clean pragmatic design. A Web framework is a set of components that provide a standard way to develop websites fast and easily. Django’s primary goal is to ease the creation of complex database-driven websites. Some well known sites that use Django include PBS, Instagram, Disqus, Washington Times, Bitbucket and Mozilla Django (/ˈdʒæŋgoʊ/ jang-goh) is a free and open source web application framework, written in Python. A web framework is a set of components that helps you to develop websites faster and easier. When you're building a website, you always need a similar set of components: a way to handle user

authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Luckily for you, other people long ago noticed that web developers face similar problems when building a new site, so they teamed up and created frameworks (Django being one of them) that give you ready-made components to use.
Frameworks exist to save you from having to reinvent the wheel and to help alleviate some of the overhead when you're building a new site. Why do you need a framework? To understand what Django is actually for, we need to take a closer look at the servers. The first thing is that the server needs to know that you want it to serve you a web page. Imagine a mailbox (port) which is monitored for incoming letters (requests). This is done by a web server. The web server reads the letter and then sends a response with a webpage. But when you want to send something, you need to have some content. And Django is something that helps you create the content. When a request comes to a web server, it's passed to Django which tries to figure out what is actually requested. It takes a web page address first and tries to figure out what to do. This part is done by Django's urlresolver (note that a website address is called a URL – Uniform Resource Locator – so the name urlresolver makes sense). It is not very smart – it takes a list of patterns and tries to match the URL. Django checks patterns from top to bottom and if something is matched, then Django passes the request to the associated function (which is called view). Imagine a mail carrier with a letter. She is walking down the street and checks each house number against the one on the letter. If it matches, she puts the letter there. This is how the urlresolver works! In the view function, all the interesting things are done: we can look at a database to look for some information. Maybe the user asked to change something in the data? Like a letter saying, "Please change the description of my job." The view can check if you are allowed to do that, then update the job description for you and send back a message: "Done!" Then the view generates a response and Django can send it to the user's web browser. The description above is a little bit simplified, but you don't need to know all the technical things yet. Having a general idea is enough. So instead of diving too much into details, we will start creating something with Django and we will learn all the important parts along the way! Django is a free and open source web application framework written in Python. A framework is nothing more than a collection of modules that make development easier. They are grouped together, and allow you to create applications or websites from an existing source, instead of from scratch. This is how websites - even simple ones designed by a single person - can still include advanced functionality like authentication support, management and admin panels, contact forms, comment boxes, Built by experienced developers, file upload support, and more. In other words, if you were creating a website from scratch you would need to develop these components yourself. By using a framework instead, these components are already built, you just need to configure them properly to match your site. Django as "a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source."
Django offers a big collection of modules which you can use in your own projects. Primarily, allowing developers to augment the template frameworks exist to save developers a lot of wasted time and headaches and Django is no different. You might also be interested in learning that Django was created with front-end developers in mind. "Django's template language is

designed to feel comfortable and easy-to-learn to those used to working with HTML, like designers and front-end developers. But it is also flexible and highly extensible, allowing developers to augment the template language as needed."

2.3 OVERVIEW OF MYSQL

Advantages of MYSQL MYSQL is one of the top databases available in the market. MYSQL is a relational database with many advanced features and options. Over time, MYSQL has proved itself to be a fast, reliable and cost effective competitor to the other more expensive databases like MS SQL Server and Oracle. Here are a few of the advantages of using MYSQL in database development.
Open Source MYSQL is an open source database system which means that anyone can use it for free. Developers can amend its code to suit their requirements which means that MYSQL is highly customizable as well. Another edge of using MYSQL over other database systems is that; it is available widely in the market with no ownership cost.
Fast Development
A lot of people around the globe are continuously developing new modules for integration with MYSQL. This means that it has a wider and faster
development circle. Patches, upgrades and fixes are developed fast and become available in forums, blogs and developer sites on the internet.
Better for Small Businesses This relational database system is free so it reduces the cost of overall database solution for small businesses and companies. This database is relatively easy to learn and operate, so operational cost is reduced substantially which is again an important factor in classifying MYSQL as an applicable and practical tool for small businesses.
Cross Platform Operability MYSQL is easily installable and operable on different platforms including Windows, Linux, OS2 and Solaris. Cross platform operability makes it a favourable choice for development companies. MYSQL database system also contains APIs for integration with C, C++, PHP, Java, Perl, Python, Tcl, and Ruby etc. You can connect it easily with different development platforms so you can actually integrate applications developed in different OS and development platforms. Security MYSQL as a relational database is secure as all access passwords are stored in an encrypted format restricting any unauthorised access to the system. It also encrypts the transactions so eavesdroppers and data harvest tools cannot replicate or regenerate the database transactions once they are processed. Connectivity
MYSQL clients can access this relational database through standard TCP/IP sockets, named pipes, UNIX sockets and many more. Standard ODBC 2.5 and above functions and commands

2.4 DATABASE SERVERS

A database server is used to store data in a database. Users can access the data and manipulate it. A web application can provide the user with the interface to the database. There are many types of databases. The most popular among them is the Relational Database Management System (RDBMS).
RDBMS:

RDBMS is a type of database management system that stores data in the form of related tables. Relational databases are powerful because they require few assumptions about how data is related or how it will be extracted from the database. As a result, the same database can be viewed in many different ways. An important feature of relational systems is that a single database can be spread across several tables. This differs from flat-file databases, in which each database is self-contained in a single table.

SQL:

The structured Query Language (SQL) comprises one of the fundamental building blocks of modern database architecture. SQL is an ANSI (American National Standards Institute) standards computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. SQL works with database programs like MS Access, Oracle, DB2, Informix, MS SQL Server and Sybase etc.

A database most often contains one or more tables. Each table is identified by a name (E.g. “staff” or “travellor”). A table contains record (rows) with data. With SQL we can query a database and have a result set returned. SQL is the syntax for executing queries. But the SQL language also includes the syntax to insert and delete records. These query and update commands together form the Data Manipulation Language (DML) part of SQL. The Data Definition Language (DDL) part of SQL permits database tables to be created or detected. We can also define indexes (keys), specify links between tables and imposes constraints between databases.

2.5 FEASIBILITY STUDY

The key conditions involved in the feasibility analysis are: -

Economical Feasibility.

Operational Feasibility.

Technical Feasibility.

Economical Feasibility: -

Economical feasibility analysis is the most frequently used evaluating technique for the effectiveness of the candidate system. This checks, are their sufficient benefits in creating the system to make cost acceptable. In the case of the proposed system it is very necessary to implement in such a firm and when its necessity compared its cost for implementing is very low and it is very acceptable by the users of the system. When its advantages and efficiency evaluated it is economically feasible.

Operational Feasibility: -

The proposed systems are beneficial only if they can turn into information system that will meet the organization’s operating requirements. This test of feasibility asks if the system will work when it is developed and installed. Since this system is developed in vb.net its operation is very simple, very attractive and user friendly. The software is very much available, and it is the most using package now, and since it is a Microsoft version and it provides many tools which helps to make the usage very easier. It is very much acceptable by the users of the system and it is keeping its standard

Technical Feasibility: -

This checks that, can the work for the project be done with current equipment and existing software technology and available personnel. Since there is no other technical supports are necessary for the function of the system we can conclude that it is technically feasible. When it is implemented there is only need of a computer (PC) and its other peripherals as usual so it is very satisfied in the matter of technical feasibility.

3.SYSTEM DESIGN

System design is the process of developing for a candidate system that meet the criteria established in system analysis. User requirements are translated into systems characteristics during system design. System design involves firstly the logical design and then physical construction of new system. The logical design describes the detailed specifications for the new system, the input/output, files and databases, procedures, all in a manner that must be project requirement. Physical construction, the activity following design produces software files and a working system.

3.1 INPUT DESIGN

Input design is a part of the overall system design, which requires very careful attention. Often the collection of input data is the most expensive part of the system, in terms of both the equipment used and people involved. If the data going into the system is incorrect, then the processing and output will magnify the errors. Thus the clear objectives of input design are:
To produce a cost-effective method of input.
To achieve the highest possible level of accuracy.
To ensure that the input is acceptable to and understood by the user.

3.2 OUTPUT DESIGN

The output design is done so that the result of processing could be committed to the user and to provide a hard copy of these results and evaluations for later consultations. Effective output design will improve the clarity and performance of outputs. Output design phase of the system is concerned with the convergence of information's to the end user-friendly manner. The output design should be efficient, intelligible so that system relationship with the end user is improved and thereby enhancing the process of decision making. Outputs from the computer systems are required primarily to communicate the results of the processing to the users. They are also used to provide a permanent copy of these results of processing to the users.
They are also used to provide a permanent copy of these results for late consultation. There are various types of output required by most systems, the main ones are:
External outputs, whose destination is outside the organization and which require special attention because they project the image of the organization.
Internal outputs, whose destination is within the organization and which require careful design because they are the user's main interface with the computer.
Operational outputs, whose use is purely within the computer department.

Turn around outputs, to which the data will be added before they are returned to the computer for further processing.

3.3 DATAFLOW DIAGRAM

The data flow diagram (DFD) is one of the most important tools used by system analyst. Data flow diagrams are made up of a number of symbols, which represent system components .Most data flow modelling methods use four kinds of symbols. These symbols are used to represent four kinds of system components. Processes, data source, data flows and external entities. Processes are represented by circles in DFD. Dataflow represented by a thin line in the DFD and each data store has a unique name and square or rectangle represents external entities Unlike detailed flow charts, data flow diagrams do not supply detailed descriptions of the modules but graphically describes a systems data and how the data interact with the system.

3.4 TABLE DESIGN

NORMALIZATION
Normalization is the process of decomposing a set of relations with anomalies to produce smaller and well-structured relations that contain minimum redundancy. It is a formal process of deciding which attributes should be grouped together in a relation.
First Normal Form
First Normal form (1NF) is now considered to be part of the formal definition of relational model.1NF is designed to disallow multivalve attribute, composite attributes, and their combinations. It states that the domain of an attribute must include only atomic values. A domain is atomic, if elements of the domain are considered to be indivisible units. We say that a relational schema R is in 1NF if the domain of all attributes of R is atomic.
Second Normal Form
Second Normal form (2NF) is based on the concept of functional dependency. A relation R is in 2NF if it is in 1NF and every non key attribute A of R is fully dependent on the primary key. That is, relation is said to be in 2NF if each attribute An in R meets one of the following criteria:
(a) It appears in the primary key. (b) It is fully functionally dependent on the primary key.
The tables designed in the proposed system, contain a primary key for uniquely identifying each user.
Third Normal Form
Third Normal form (3NF) is based on the concept of transitive dependency. A relation is said to be in 3NF if it is in 2NF and has no transitive dependencies. That is all the non key attribute should be functionally determined by the primary key. In the proposed system all attributes of tables are fully depends on the primary key only that is all non-key attributes are mutually independent.

4.SYSTEM TESTING

4.1 TESTING

Testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Thus the system test should be a confirmation that all is correct and an opportunity to show the users that the system works. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing is the process of executing a program with the explicit; it can only show that software defects at present.
System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.
The testing steps are:
Unit testing
Integration testing
Validation
Output testing
User acceptance testing
UNIT TESTING
Unit testing focuses verification efforts on the smallest unit of software design, the module. This is also known as “module testing”. The modules are tested separately. This testing is carried during programming stage itself. In this testing step each module is found to be working satisfactorily as regard to the expected output from the module
INTEGRATION TESTING
Integration testing focuses on the design and the construction of the software architecture. Data can be lost across an interface; one module can have adverse effect of another sub function and so on. Thus integration testing is a part that the software meets all functional, behavioral and performance requirements. The errors, which are uncovered during integration testing, are corrected during this phase.
Integration testing systematic techniques for conducting the program structure. While at the same time conducting tests to uncover errors associated with the interfacing.
VALIDATION TESTING
Errors discovered where corrected prior to completion of this project with the help of the user by negotiating to establish a method of resolving deficiencies. Thus the proposed system under validation testing and found to be working satisfactorily.
In validation testing the requirements established as a part of software requirements are validated against the software that has been constructed. Validation testing provides the final assurance that software meets all functional behavioral and performance requirements.
OUTPUT TESTING
After performing the validation testing, the next step is output testing of the proposed system since no system could be useful if it does not produce the required output in the specific format. The outputs generated or displayed by the system under consideration are tested asking the users about the format required by them. Here, the output is consideration into two ways: one is the screen and the other is printed format.

The output format on the screen is found to be correct as the format designed according to the users need. For the hard copy also, the output comes out as specified by the user. Hence output testing does not result in any correction in the system.

USER ACCEPTANCE TESTING

Acceptance testing includes final testing of the complete system to user satisfaction and supervision of the new system.

User acceptance testing is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. This is done with regard to the following points.

Input screen design

Process screen design

Selecting avi and txt file

Return to the code directory

The above testing is done, by taking various kinds of data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system by using test data errors are given uncovered and corrected by using above testing and corrections are also noted for future use.

4.2 PROCESS TESTING

The test team consists of the company staff, the user, and myself. The company staffs are very much experienced in the testing field. We test the system by running it in several ways by inputting various The main concerns in our testing are following:

Browser compatibility: This test validates consistent applications performance on variety of browser type and configurations.

Functional correctness: These tests validates that the application function correctly.

Integration: These test the integration between browsers and servers, applications and data, hardware and software.

Usability: These test the overall usability of a web page or a web application, including appearance clarity and navigation.

Security: These test the adequacy and correctness of security controls including access control and authorizations.

Performance: These test the performance of the web applications under load.

Verification of code: This validate that the code used in building the web application has been used in a correct manner.

5.SYSTEM IMPLEMENTATION

The implementation phase of a project covers the period from the acceptance of the tested design to its satisfactory operations, supported by the appropriated user and the operation manual. It is major operation across the whole organization structure and requires the great deal of planning. Planning of implementation must begin from initial conception of the project. It

require a thorough knowledge of the new system, its personal need, hardware and software requirements, file and procedure conversion activities and of the current system where interface with the new, the change to it, the job that will be superseded, etc. Only the analyst responsible for creation the new system wills possess this knowledge. The new system analyst can plan, schedule and coordinate, but has no executive power. Planning must cover the following aspects.

Organization of implementation.
Control of resources.
Motivation of the users.
Training and production manuals.
Change over.

6.CONCLUSION

This package has been developed to reduce the strain at the user level and at the administrative level. This system is designed using a generalized application and is also a highly user-friendly one.

This project is specifically designed to list out the services along with public service.

This application is completely automated, user friendly and at most accurate. This project is a sincere attempt to develop a fully automated platform to get jobs, clear their career doubts, and we hope that this project would be of great use for the Students in this college.