**Table of Contents**

**1. INTRODUCTION 2-3**

**(A)Current System**

**(B) Need of Proposed System**

**(C) Problem Definition Statement**

**2. SYSTEM DEVELOPMENT LIFE CYCLE 4**

**3. ANALYSIS 7-30**

**(A) Requirement Analysis**

**(B) Requirement Specification**

**(C) Usecase Analysis**

(I) Usecase Diagram

(II) Usecase Description

(III) Activity Diagram

**4. DESIGN 31-36**

**(A) System Flow Diagram**

**(B) Module Identified**

**(C) Data Flow Diagram**

**(D) Class Diagram**

**(E) Database Design**

E-R DIAGRAM

**5. IMPLEMENTATION 37-41**

**(A) Platform Used**

**(I) Hardware Platform**

**(II) Software Platform**

**(B) Implementation Level Details**

**(C) Testing**

Testing Technology Used

**6. CONCLUSION 42-43**

**(A) Important Features**

**(B) Limitation**

**(C) Future Work**

1. **INTRODUCTION**

This project is developed an online Job Portal for the Placement Dept. of the company. The system is an online application that can be accessed throughout the organization and outside as well with proper login provided. This system can be used as an Online Job Portal for the Placement Dept of the organization to manage the student information with regards to placement. Students logging should be able to upload their information in the form of a CV. Visitors/Company representatives logging in may also access/search any information put up by Students. The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of Mysql Server and all the user interfaces have been designed using the Java technologies.  The  database  connectivity is  planned using the “SQL Connection” methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated reports, whichare produced as per the applicable strategies and standards that are put forwarded by the administrative staff. The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The specification has been normalized up to 3NF to eliminate all the anomalies that may arise due to the database transaction that are executed by the general users and the organizational administration. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as My-SQL.The basic constructs of table spaces, clusters and indexes have been exploited to provide higher consistency and reliability for the data storage. The MySQL was a choice as it provides the constructs of high-level reliability and security. The total front end was dominated using the java technologies . At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations.

**(A)Current System-**

* + - Cannot Upload and Download the latest updates.
    - No use of Web Services and Remoting.
    - Risk of mismanagement and of data when the project is under development.
    - Less Security.
    - No proper coordination between different Applications and Users.
    - Fewer Users - Friendly.

**(B) Need of Proposed System**

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

1.   User friendliness is provided in the application with various controls.

2.   The system makes the overall project management much easier and flexible.

3.   Readily upload the latest updates, allows user to download the alerts by clicking the URL.

4.   There is no risk of data mismanagement at any level while the project development is under process.

5.   It provides high level of security with different level of authentication.

**(C) Problem Definition Statement**

There are many barriers to good communication in the company and job seeker relationship, including candidates' anxiety and fear, companies burden of work, fear of litigation, fear of extra fees or verbal abuse, and unrealistic expectations.

**2. SYSTEM DEVELOPMENT LIFE CYCLE**

The system development life cycle framework provides a sequence of activities for system designers and developers to follow. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.

The SDLC adheres to important phases that are essential for developers, such as [planning](https://en.wikipedia.org/wiki/Planning), [analysis](https://en.wikipedia.org/wiki/Analysis), [design](https://en.wikipedia.org/wiki/Design), and [implementation](https://en.wikipedia.org/wiki/Implementation), and are explained in the section below. It includes evaluation of present system, information gathering, feasibility study and request approval. A number of SDLC models have been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, synchronize and stabilize. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next. These stages can be characterized and divided up in different ways, including the following:

* **Preliminary analysis**: The objective of phase 1 is to conduct a preliminary analysis, propose alternative solutions, describe costs and benefits and submit a preliminary plan with recommendations.

Conduct the preliminary analysis: in this step, you need to find out the organization's objectives and the nature and scope of the problem under study. Even if a problem refers only to a small segment of the organization itself, you need to find out what the objectives of the organization itself are. Then you need to see how the problem being studied fits in with them.

Propose alternative solutions: In digging into the organization's objectives and specific problems, you may have already covered some solutions. Alternate proposals may come from interviewing employees, clients, suppliers, and/or consultants. You can also study what competitors are doing. With this data, you will have three choices: leave the system as is, improve it, or develop a new system.

Describe the costs and benefits.

* **Systems analysis, requirements definition**: Defines project goals into defined functions and operation of the intended application. It is the process of gathering and interpreting facts, diagnosing problems and recommending improvements to the system. Analyzes end-user information needs and also removes any inconsistencies and incompleteness in these requirements.

A series of steps followed by the developer are:

1. Collection of Facts: End user requirements are obtained through documentation, client interviews, observation and questionnaires,
2. Scrutiny of the existing system: Identify pros and cons of the current system in-place, so as to carry forward the pros and avoid the cons in the new system.
3. Analyzing the proposed system: Solutions to the shortcomings in step two are found and any specific user proposals are used to prepare the specifications.

* **Systems design**: Describes desired features and operations in detail, including screen layouts, [business rules](https://en.wikipedia.org/wiki/Business_rule), [process diagrams](https://en.wikipedia.org/wiki/Process_Diagram), [pseudocode](https://en.wikipedia.org/wiki/Pseudocode) and other documentation.
* **Development**: The real code is written here.
* **Integration and testing**: Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
* **Acceptance, installation, deployment**: The final stage of initial development, where the software is put into production and runs actual business.
* **Maintenance**: During the maintenance stage of the SDLC, the system is assessed to ensure it does not become obsolete. This is also where changes are made to initial software. It involves continuous evaluation of the system in terms of its performance.
* **Evaluation**: Some companies do not view this as an official stage of the SDLC, while others consider it to be an extension of the maintenance stage, and may be referred to in some circles as post-implementation review. This is where the system that was developed, as well as the entire process, is evaluated. Some of the questions that need to be answered include: does the newly implemented system meet the initial business requirements and objectives? Is the system reliable and fault-tolerant? Does the system function according to the approved functional requirements? In addition to evaluating the software that was released, it is important to assess the effectiveness of the development process. If there are any aspects of the entire process, or certain stages, that management is not satisfied with, this is the time to improve. Evaluation and assessment is a difficult issue. However, the company must reflect on the process and address weaknesses.
* **Disposal:** In this phase, plans are developed for discarding system information, hardware and software in making the transition to a new system. The purpose here is to properly move, archive, discard or destroy information, hardware and software that is being replaced, in a manner that prevents any possibility of unauthorized disclosure of sensitive data. The disposal activities ensure proper migration to a new system. Particular emphasis is given to proper preservation and archival of data processed by the previous system. All of this should be done in accordance with the organization's security requirements.

**3. ANALYSIS**

**(A) Requirement Analysis-**

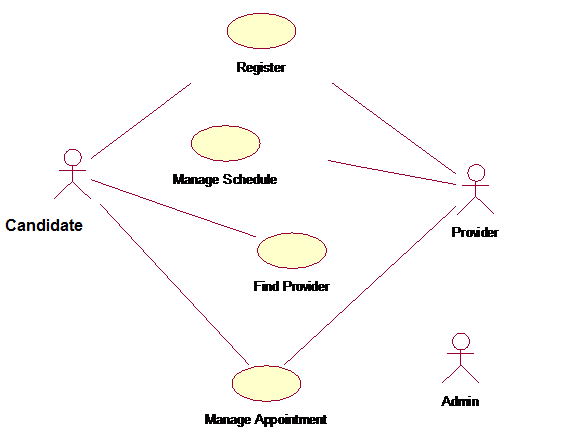
* **Developers:**in order to be sure they are developing the right project that fulfills requirements provided in this document.
* **Testers:**in order to have an exact list of the features and functions that has to respond according to requirements and provided diagrams.
* **Users:**in order to get familiar with the idea of the project and suggest other features that would make it even more functional.
* **Documentation writers:**to know what features and in what way they have to explain. What security technologies are required, how the system will response in each user’s action etc.

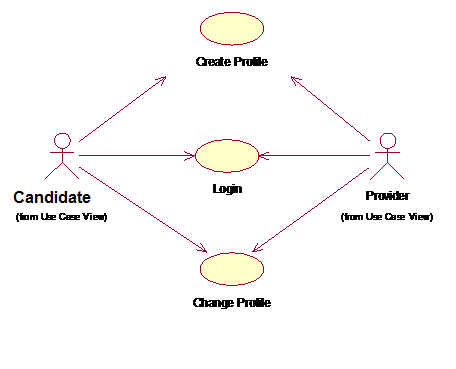
**(B) Requirement Specification-**

The functional requirements were derived from the following use case diagram. Use case diagrams represent the functional interactions of a system. The stick figures represent the actors, which are external to the system and interact with the system through interfaces. The actors of the online appointment system are the Job Seeker and the Provider. The Job Seeker uses the system to schedule appointments to obtain services from Providers. The ovals are individual use cases that represent the functions the system performs to provide the services that the actors desire. The primary use cases of the online appointment system are: Find Provider, Manage Schedule, Manage Appointment, and Register. The Job Seeker interacts with the following use cases: Register, Manage Appointment, Find Provider. The Provider interacts with the following use cases: Register, Manage Appointment, Manage Schedule.

**(C) Use case Analysis**

* 1. Use case Diagram-





(II) Usecase Description-

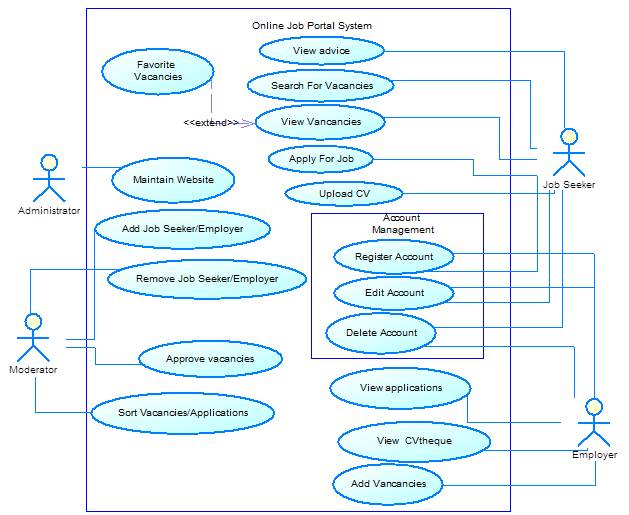
The online appointment system was also modelled as an analysis-level class diagram. Class diagrams represent the static associations between objects in the system. Analysis-level class diagrams avoid design, and only concentrate on the concepts related to the domain. Furthermore, distinguishing between classes and attributes is deferred until detailed design. Although, the leaves of the analysis-level class diagram are typically folded up into the parent object to become an attribute (e.g. Username would be an attribute of the Registered User class). Also, depending on the architecture chosen, this domain model would become the basis for designing the business layer of an enterprise architecture.

Job Seeker and Provider are external actors of the system which would also be represented as objects in the system, derived from the Registered User class. Anonymous Job Seekers that are using the system to find providers, will not be represented as objects in the system. Job Seeker, Provider, Schedule, and Appointment would be the primary objects in the system, and would be the most complex shown from the multiple relationships that they share. Another interesting item is how many attributes Appointment and Non Work Hours share. Design might modify these items such that Appointment is a specialization of Non Work Hours as to avoid code duplication during the Implementation phase. UML design artifacts, such as collaboration diagrams, would provide the exercise to flesh out the complete set of attributes and methods necessary to support the system. This analysis artifact helps the transition from analysis to design.

(III) Activity Diagram-

Finally, the dynamic aspects of the system are modelled as sequence diagrams. Each use case has its own corresponding sequence diagram. The sequence diagram shows the flow of data between the system and external actors through interfaces, as well as for the flow of data and message calls between objects internal to the system. Data cannot flow between actors and objects unless they have the appropriate association represented in the class diagram. Only some of the objects of the class diagram are required to support the function of each individual use case. The scenarios represented are the most common path through the system and do not detail alternate scenarios. The scenarios are also time dependent and happen in sequence. The transitions internal and external to scenarios might be either synchronous or asynchronous.

3.1.1 Find Provider



3.1.1.1 *Introduction*

Job seeker shall be allowed to search for providers by a variety of criteria. This criteria shall be by zip code and specialty but must be extensible so that any it is possible to sort or search on any data field.

3.1.1.2 *Inputs*

The user would input the searching criteria/values and the sorting criteria.

3.1.1.3 *Processing*

The system performs a search on the database of provider

3.1.1.4 *Outputs*

The user is presented with a list of providers that match the searching criteria sorted as defined by the sorting criteria. By clicking on any Company listed the user is taken to that Company ’s main appointment page.

3.1.2 Create Schedule



3.1.2.1 *Introduction*

Providers shall be able to define their schedule. This schedule shall determine when Job Seekers can make appointments to see that provider. The schedule shall be exception based. The schedule consists of the normal weekly schedule and then will contain exceptions to this normal schedule.

3.1.2.2 *Inputs*

The provider shall initially enter his/her normal weekly schedule. Then the provider will enter exceptions to this normal weekly schedule. These exceptions will take the form of a day/time and duration. Exceptions may reoccur. Reoccurrences may be weekly, monthly, or annually.

3.1.2.3 *Processing*

The system will create an internal representation of the specified schedule.

3.1.2.4 *Outputs*

The system will confirm the schedule addition3.1.2 View Schedule



3.1.2.1 *Introduction*

The provider shall be able to view his/her schedule. This schedule shall be in the format of a calendar. The default view will be a weekly view, and the provider shall be able to view the schedule for the current week or any week in the past/future. The provider shall have the ability to navigate weeks by using a month view on the same page.

3.1.2.2 *Inputs*

The provider will choose to view his/her schedule

3.1.2.3 *Processing*

none

3.1.2.4 *Outputs*

The schedule view as described above. 3.1.2 Change Schedule



3.1.2.1 *Introduction*

Providers shall have the ability to change their schedule much in the way they originally created it. They can either change their standard weekly schedule or add/change/delete exceptions.

3.1.2.2 *Inputs*

The provider chooses to either modify standard weekly settings or to modify exceptions. The provider will then provide the modification values.

3.1.2.3 *Processing*

none

3.1.2.4 *Outputs*

The system displays confirmation of the modification .

3.1.3 Request Appointment

3.1.3.1 *Introduction*

The Job Seeker uses a web browser to schedule an appointment pending the providers approval.

3.1.3.2 *Inputs*

The Job Seeker interacts with the providers schedule to select the appointment type and time for appointment.

3.1.3.3 *Processing*

The system recommends the 1st available appointment time to the Job Seeker, and locks the appointment time when the Job Seeker submits it.

3.1.3.4 *Outputs*

A confirmation of the appointment request is displayed to the user, and a notification of an appointment request is sent to the Provider. Another Job Seeker viewing the schedule, would now see that block of time on the schedule represented as being a pending appointment.3.1.3 Confirm Appointment

3.1.3.1 *Introduction*

The provider uses a web browser to approve or disapprove appointments that Job Seekers have requested.

3.1.3.2 *Inputs*

The provider can select to view all requested appointments in a list or upon viewing his schedule and marking the appointment(s) that he wants to approve or disapprove.

3.1.3.3 *Processing*

The system marks the appointment(s) as being approved or disapproved.

3.1.3.4 *Outputs*

 The provider receives a confirmation that the appointments were approved, and the Job Seeker receives an email notification that the appointment was approved. Another Job Seeker viewing the schedule would now see this block of time represented as an approved appointment.3.1.3 Change Appointment

3.1.3.1 *Introduction*

This allows Job Seekers or providers to change an appointment. The following sequence diagram represents the provider initiating the change.

3.1.3.2 *Inputs*

The provider selects the appointment and submits the associated information to change.

3.1.3.3 *Processing*

The system modifies the appointment and changes its state.

3.1.3.4 *Outputs*

 The Job Seeker receives an email notification of the change, and the provider receives confirmation that the change was submitted.3.1.3 Cancel Appointment

3.1.3.1 *Introduction*

This allows Job Seekers or providers to cancel an appointment. The following sequence diagram represents the Job Seeker initiating to cancel.

3.1.3.2 *Inputs*

The Job Seeker selects to delete his or her appointment.

3.1.3.3 *Processing*

The system marks the appointment as cancelled.

3.1.3.4 *Outputs*

The provider is emailed a notification of the cancellation, and the Job Seeker is displayed a confirmation that it has been cancelled. The Job Seeker might ensue some fee for cancelling after a certain time period.

3.1.3 Appointment Reminder

3.1.3.1 *Introduction*

This use case is used to remind Job Seekers of upcoming appointments.

3.1.3.2 *Inputs*

This process is periodically initiated by the system based off of time during a low-use period of the system.

3.1.3.3 *Processing*

The system determines which appointments need reminders to be sent out, and if the Job Seekers have selected to have reminders sent.

3.1.3.4 *Outputs*

The Job Seeker will receive an email notification of the upcoming appointment.3.1.3 View Appointment

3.1.3.1 *Introduction*

The Job Seeker does this to view his appointment(s).

3.1.3.2 *Inputs*

A registered Job Seeker selects to view his or her appointments.

3.1.3.3 *Processing*

The system receives the Job Seeker’s appointment info.

3.1.3.4 *Outputs*

 A dynamically generated page is displayed to the Job Seeker showing his or her past and upcoming appointments. 3.1.4 Create Profile

3.1.4.1 *Introduction*

The user shall be able to create a profile in the system. The profile shall consist of the user’s name, physical address, phone number, email address, username and password.

3.1.4.2 *Inputs*

User’s name, physical address, phone number, email address, username and password.

3.1.4.3 *Processing*

The system validates the username to make sure it is unique.

3.1.4.4 *Outputs*

Confirmation of profile creation to the screen and via email

3.1.4 Login



3.1.4.1 *Introduction*

The users shall be able to login to the system. Job Seekers and Providers will log in using the same way, but will be taken to different screens once logged in.

3.1.4.2 *Inputs*

The user’s username and password

3.1.4.3 *Processing*

Check username and password against stored values.

3.1.4.4 *Outputs*

Login confirmation if username and password match. An error message if login fails.

3.1.4 Change Profile



3.1.4.1 *Introduction*

Changing the profile is very similar to creating a profile

3.1.4.2 *Inputs*

The attributes to change and the values to change them to

3.1.4.3 *Processing*

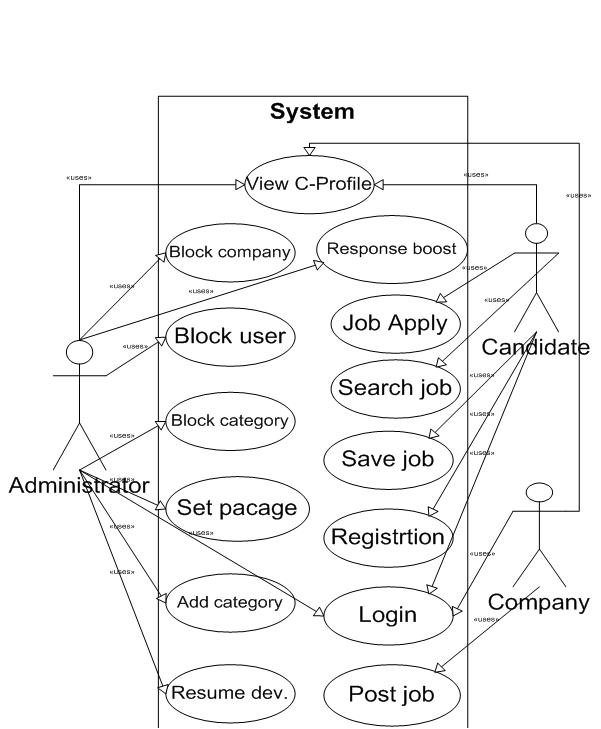
Validate username to make sure it is unique if the user changed the username

3.1.4.4 *Outputs*

Confirmation of changed profile

**4. DESIGN**

**(A) System Flow Diagram-**

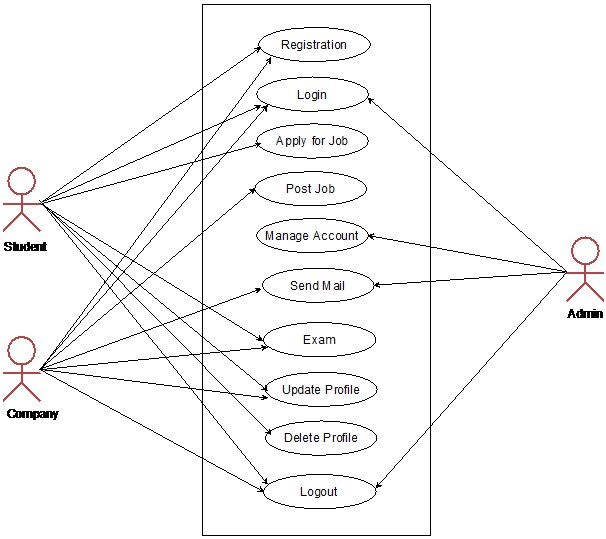
****

**(B) Module Identified-**

This adds the ability to scheduling Job Seekers for appointments then manage their flow through a clinic once they arrive for appointments. In simple terms, the module allows implementations to do the following:

* 1. Create a schedule of when providers are available to see Job Seekers
  2. Schedule appointments for Job Seekers based on provider schedule
  3. Manage the Job Seeker queue through the clinic when the Job Seekers arrive for their appointments.

**(C) Data Flow Diagram-**



**(D) Class Diagram**

The online appointment system was also modelled as an analysis-level class diagram. Class diagrams represent the static associations between objects in the system. Analysis-level class diagrams avoid design, and only concentrate on the concepts related to the domain. Furthermore, distinguishing between classes and attributes is deferred until detailed design. Although, the leaves of the analysis-level class diagram are typically folded up into the parent object to become an attribute (e.g. Username would be an attribute of the Registered User class). Also, depending on the architecture chosen, this domain model would become the basis for designing the business layer of an enterprise architecture.

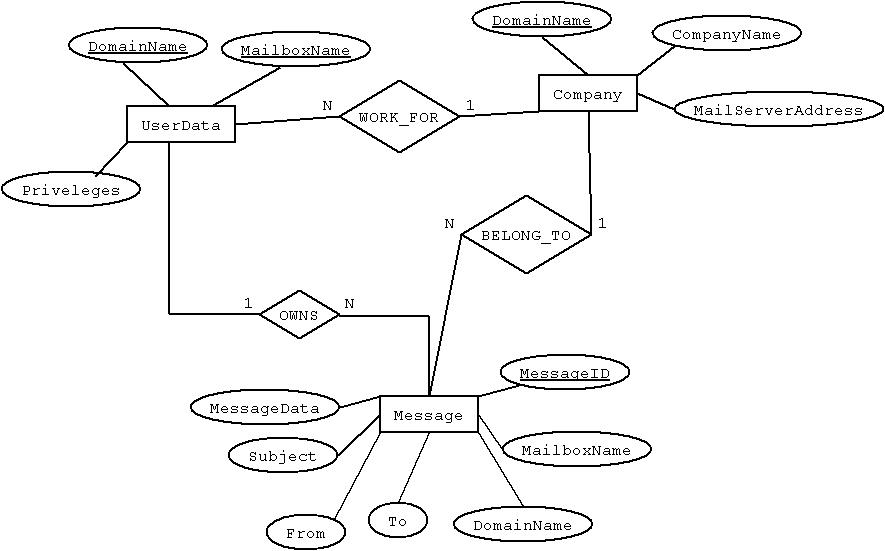
Job Seeker and Provider are external actors of the system which would also be represented as objects in the system, derived from the Registered User class. Anonymous Job Seekers that are using the system to find providers, will not be represented as objects in the system. Job Seeker, Provider, Schedule, and Appointment would be the primary objects in the system, and would be the most complex shown from the multiple relationships that they share. Another interesting item is how many attributes Appointment and Non Work Hours share. Design might modify these items such that Appointment is a specialization of Non Work Hours as to avoid code duplication during the Implementation phase. UML design artifacts, such as collaboration diagrams, would provide the exercise to flesh out the complete set of attributes and methods necessary to support the system. This analysis artifact helps the transition from analysis to design.



Seeker

**(E) Database Design**

E-R DIAGRAM



**5. IMPLEMENTATION**

**(A) Platform Used**

JAVA is a server-side, cross-platform, HTML-embedded scripting language. JAVA (recursive acronym for *JAVA:*) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

Currently there are over half a million domains running JAVA. Much of JAVA’s syntax is borrowed from C, Java and Pearl with a couple of unique JAVA-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly. JAVA eliminates the need for numerous small cgi programs by allowing you to place simple scripts directly in your HTML files. It also makes it easier to manage large web sites by placing all components of a web page in a single html file.

JAVA is mainly focused on server-side scripting, so you can do anything any other CGI program can do, such as collect form data, generate dynamic page content, or send and receive cookies.

JAVA can be [used](http://in3.php.net/manual/en/install.php) on all major operating systems, including Linux, many Unix variants (including HP-UX, Solaris and Open BSD), Microsoft Windows, Mac OS X, RISC OS, and probably others. JAVA has also support for most of the web servers today.

One of the strongest and most significant features in JAVA is its support for a [wide range of databases](http://in3.php.net/manual/en/refs.database.php). Writing a database-enabled web page is incredibly simple using one of the database specific extensions (e.g., for [mysql](http://in3.php.net/manual/en/book.mysqli.php)), or using an abstraction layer like [PDO](http://in3.php.net/manual/en/book.pdo.php), or connect to any database supporting the Open Database Connection standard via the [ODBC](http://in3.php.net/manual/en/book.uodbc.php) extension. Other databases may utilize [URL](http://in3.php.net/manual/en/book.curl.php) or [sockets](http://in3.php.net/manual/en/book.sockets.php), like Couch DB.

**Back End: MySQL**

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout it's history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

MySQL is a open source Relational Database Manage.ment System. MySQL is very fast reliable and flexible Database Management System. It provides a very high performance and it is multi-threaded and multi user Relational Database management system.

MySQL is one of the most popular relational database Management System on the web. The MySQL Database has become the world's most popular open source  Database, because it is free and available on almost all the platforms. The MySQL can run on Unix , window, and Mac OS. MySQL is used for the internet applications as it provides good speed and is very secure. MySQL was developed to manage large volumes of data at very high speed to overcome the problems of existing solutions. MySQL can be used for variety of applications but it is mostly used for the web applications on the internet.

**Application Server:  Xampp Server**

XAMPP is a [free and open source](http://software)[cross-platform](http://cross-platform)[web server](http://server)[solution stack](http://stack) package, consisting mainly of the [Apache HTTP Server](http://server), [MySQL](http://mysql)[database](http://database), and [interpreters](about:blank) for scripts written in the [JAVA](http://php) and [Perl](http://perl)[programming languages](http://language).

|  |
| --- |
|  |

XAMPP's name is an [acronym](http://initialism) for- X (to be read as "cross", meaning [cross-platform](http://cross-platform)),[Apache HTTP Server](http://server),[MySQL](http://mysql),[JAVA](http://php),[Perl](http://perl).

The program is released under the terms of the [GNU General Public License](http://license) and acts as a free [web server](http://server) capable of serving dynamic pages. XAMPP is available for [Microsoft Windows](http://windows), [Linux](http://linux), [Solaris](about:blank), and [Mac OS X](http://x), and is mainly used for web development projects. This software is useful while you are creating dynamic webpages using programming languages like JAVA, JSP, Servlets.

Requirements and features: XAMPP requires only one [zip](about:blank), [tar](about:blank), [7z](about:blank), or [exe](http://exe) file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. XAMPP is regularly updated to incorporate the latest releases of [Apache](http://server)/[MySQL](http://mysql)/[JAVA](http://php) and [Perl](http://perl). It also comes with a number of other modules including [OpenSSL](http://openssl) and [JAVAMyAdmin](http://phpmyadmin).

Installing XAMPP takes less time than installing each of its components separately. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. It is offered in both a full, standard version and a smaller version.

Use: Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. In practice, however, XAMPP is sometimes used to actually serve web pages on the [World Wide Web](http://web). A special tool is provided to password-protect the most important parts of the package.

**IDE: (Integrated Development Environment)**

An integrated development environment (IDE) (also known as integrated design environment, integrated debugging environment or interactive development environment) is a [software application](http://application) that provides comprehensive facilities to [computer programmers](http://programmer) for [software development](http://development). An IDE normally consists of:

* a [source code editor](http://editor)
* a [compiler](http://compiler) and/or an [interpreter](about:blank)
* [build automation](http://automation) tools
* a [debugger](http://debugger)

The boundary between an integrated development environment and other parts of the broader software development environment is not well-defined. Sometimes a [version control system](http://system) and various tools are integrated to simplify the construction of a [GUI](http://gui). Many modern IDEs also have a [class browser](http://browser), an [object inspector](about:blank), and a [class hierarchy](http://hierarchy)[diagram](http://diagram), for use with [object-oriented software development](http://programming).

We are using **Dreamweaver** as an IDE

**(I) Hardware Platform-**

**Processor** :  Pentium-4 or above

**Processor Speed** : 2.00 GHz CPU

**RAM**  : 512 MB or above  
**Hard Disk Utilization** : 40 GB or above

**(II) Software Platform-**

**Front End**  : JAVA

**Back End** : My SQL

**Application Server** : Xampp Server

**Operating System** : Windows, Linux.

**(B) Implementation Level Details**

* **Developers:**in order to be sure they are developing the right project that fulfills requirements provided in this document.
* **Testers:**in order to have an exact list of the features and functions that has to respond according to requirements and provided diagrams.
* **Users:**in order to get familiar with the idea of the project and suggest other features that would make it even more functional.
* **Documentation writers:**to know what features and in what way they have to explain. What security technologies are required, how the system will response in each user’s action etc.
* **Admin, Receptionist, Company s and Job Seekers:** in order to know exactly what they have to expect from the system, right inputs and outputs and response in error situations.

**(C) Testing**

Testing Technology Used-

**Methods**

Questionnaires were collected from both Job Seekers and Company s in one large tertiary public hospital in Shanghai, China.Data were analyzed to measure their satisfaction and views about the WAS.

**Results**

The 1000 out of Job Seekers randomly selected for the survey were least satisfied about the waiting time to see a Company . Even though the WAS provided a much more convenient booking method, only 17% of Job Seekers used it. Of the 197 Company s surveyed, over 90% thought it was necessary to provide alternative forms of appointment booking systems for Job Seekers. However, about 80% of those Company s who were not associated professors would like to provide an ‘on-the-spot’ appointment option, which would lead to longer waits for Job Seekers.

**6. CONCLUSION**

**(A) Important Features-**

**Login Account**

**Description:**

To open the user account the users have to enter login information.

**Stimulus/response**

User must enter valid user id and password to open user page. If it is valid then it links to user account page. If the user is new to the polyclinic he/she has to register.

**Basic data flow**

* Here first the user enters login id and password.
* After entering the login information system checks whether entered login id and password is valid or not.
* If it is valid then it is linked to the user account.
* If the user doesn’t have user account then user needs to register.

**Functional requirements**

Here administrator, receptionist, Company s and Job Seekers are using the different login pages.

**Admin**

**Description**

Admin is a super-user. He/she is able to control the whole system. Admin can add, delete, update and modify the system.

**Stimulus and response**

Admin logs into the admin account and do the relevant changes daily. Admin keeps the system up-to-date.

**Basic data flow**

* Admin logs into the system.
* Can add/delete/update/modify records.
* He/she controls the entire system.

**(B) Limitation-**

* Making appointment with Company s will be allowed only member in our system
* Company s cannot cancel their appointment

**(C) Future Work-**

In future, this project can be extended to manage clinic services and treatments of Job Seekers. Enable to extend this system for Company s to cancel their appointments or replace them with other Company s who have the same specialization.