American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science &Technology (FST)

Fall 19_20

Section: J

Group No:

JOB PORTAL

A Software Engineering project submitted
By

NAME	ID
ASHFAQ AFZAL CHOWDHURY https://github.com/aacfahim/jobportal	**17-35800-3**
MD FAHIM HOQUE https://github.com/arnob-hoque	**17-35864-3**
IFTIKHER AHMED LIUM https://github.com/Ialiam	**17-35793-3**
MD RAFIQUL ISLAM https://github.com/r0fiqulislam	**17-35850-3**
MD MINHAZUL ISLAM OMI https://github.com/MinhazOmi	**17-35811-3**

The project will be evaluated for the following Course Outcomes

CO3: Choose appropriate software engineering model in a software development environment	Total Marks
Project Background Analysis (needs, goal, benefits, etc.) [5Marks]	
Appropriate Process Model Selection [5Marks]	
Argumentation for model selection with Evidence [5Marks]	
Completeness, Spelling, Grammar and Organization of the Answer [5Marks]	
CO4: Explain the roles and their responsibilities in the software project management activities	Total Marks
Content Knowledge (e.g. System Requirements, System Design) [5Marks]	
Project Role identification [5Marks]	
Responsibility Description [5Marks]	
Completeness, Spelling, grammar and Organization of the Answer [5Marks]	

Contents

1. PROBLEM DOMAIN

1.1 Background to the Problem

Job portal is an application which connects employer and job seekers where employers are the source of the resources and the job seeker can find and apply for their targeted job. This document provides details about the entire software requirement specification for the online job portal.

There are many applications developed related to this Online Job Portal, but, however, these systems are very complex for their own purpose. Most of them do not have the centralized database and easy access to users. Unlike BD Jobs and some other current Online Job Portal system is not a fully computerized and manual system for entering job candidates and job provider data and managing it.

There is no easy access to records of people seeking work in particular.

Job seekers cannot easily navigate through the database.

1.2 Solution to the Problem

In this project, we are implementing this application through a centralized database and easy-to-use access for job seekers and job providers. The administrator is the primary actor to maintain the database, and also accepts comments from job providers and job seekers to improve and develop the application's graphical user interface. Registered job seekers can also take the company online exam. In addition, qualified applicants can be classified according to this system based on their qualifications and company requirements.

Besides since it will provide the fast operation and low-cost expense than old system so it will be beneficial for the business environment.

1.3 Existing/Related Solutions

we are focusing on some advanced job portal system like Kormo to enhance our system features and flexibility.

Basically, Kormo makes it easy for any employer to post a job in a standardised manner. Kormo then quickly connects the job post with active job seekers in the Kormo community to identify a match.

So we will be focusing on some of its feature.

2. SOFTWARE DEVELOPMENT LIFE CYCLE

2.1 Process Model

A software process model is a simplified representation of a software process. Each model represents a process from a specific perspective.

2.1.1 Agile

What is an Agile Framework?

Agile represents an overarching philosophy for software development, emphasizing the value of iterating quickly and often to satisfy customers. An agile framework can be defined as a specific software-development approach based on the agile philosophy articulated in the Agile Manifesto.

It has the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment. Agile software development is more than frameworks such as Scrum, Extreme Programming or Feature-Driven Development (FDD).Agile software development is more than practices such as pair programming, test-driven development, stand-ups, planning sessions and sprints.

2.1.2 Scrum

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value.

Scrum itself is a simple framework for effective team collaboration on complex products. Scrum co-creators Ken Schwaber and Jeff Sutherland have written The Scrum Guide to explain Scrum clearly and succinctly. This Guide contains the definition of Scrum. This definition consists of Scrum’s roles, events, artifacts, and the rules that bind them together.

Scrum is:

Lightweight

Simple to understand

Difficult to master

The Scrum Framework

Scrum is simple. It is the opposite of a big collection of interwoven mandatory components. Scrum is not a methodology. Scrum implements the scientific method of empiricism. Scrum replaces a programmed algorithmic approach with a heuristic one, with respect for people and self-organization to deal with unpredictability and solving complex problems. The below graphic represents Scrum in Action as described by Ken Schwaber and Jeff Sutherland in their book Software in 30 Days taking us from planning through software delivery.

2.1.3 FDD

Feature Driven Development (FDD) is an agile framework that, as its name suggests, organizes software development around making progress on features. Features in the FDD context, though, are not necessarily product features in the commonly understood sense. They are, rather, more akin to user stories in Scrum. In other words, “complete the login process” might be considered a feature in the Feature Driven Development (FDD) methodology.

FDD was designed to follow a five-step development process, built largely around discrete “feature” projects. That project lifecycle looks like this:

Develop an overall model

Build a features list

Plan by feature

Design by feature

Build by feature

2.2 Our Process Model

In this project we will be following the Feature-Driven Development (FDD) methodology to implement where we will Develop an Overall Model, Build a Features List, Plan by Feature, Design By Feature and finally Build By Feature

We are focusing on features on this project. Our team members will be working on it based on its features. We used the user story list to implement the entire system. In every sprint we completed the modules that indicates different features.

2.3 Time box concept in Agile/Sprint in scrum

Timeboxing is allotting a fixed, maximum unit of time for an activity. That unit of time is called a time box. The goal of timeboxing is to define and limit the amount of time dedicated to an activity.

In Scrum, timeboxing is a critical component of all five events. Some Scrum teams also use timeboxing during a Sprint to concretely define open-ended tasks. An example of an open-ended task might be conducting research that is necessary for the team to reach a decision or to estimate the size and complexity of an upcoming story.

Timeboxing is a common feature of many project management methodologies because timeboxing keeps teams focused on accomplishing the task at hand by providing a clear definition of done.

Timeboxing also encourages teams to start getting work done immediately. Temporal Motivation Theory shows that time constraints are a critical component of getting work done efficiently. In Scrum, the sooner you can inspect a deliverable, the sooner you can adapt it.

Timeboxing is a critical component of good Scrum.

All five events in Scrum are timeboxed:

Sprint: Timeboxing is used to define the length of the Sprint. The Sprint is a timebox of one month or less in which the scrum team will deliver the Sprint goals. At Scrum Inc., our Sprint timebox is one week and this is what we recommend to teams that we coach.

Sprint Planning: When a team launches, they establish the timebox for the Sprint Planning meeting. As noted in the Scrum Guide, a Sprint planning meeting should be timeboxed at 8 hours or less for a one-month Sprint. The shorter the Sprint, the shorter the timebox should be for Sprint Planning. At Scrum Inc., we recommend one-week Sprints and a two-hour timebox for Sprint Planning.

Daily Scrum: The Daily Scrum is a timebox of 15 minutes for each 24-hour period that helps the Scrum Team synchronize activities and make visible any impediments to achieving the Sprint Goal.

Sprint Review: The Sprint Review is a timebox of four hours or less for one-month Sprints. During the Sprint Review, Sprint Backlog items delivered during the sprint are demonstrated and inspected. It is also a time to adapt the backlog based on feedback.

Sprint Retrospectives: The Sprint Retrospective is a timebox of three hours or less for a one month sprint. This is an event in which the team inspects itself and identifies a process improvement that the team will implement in the following sprint.

3. PRODUCT AND PROJECT DESCRIPTION

3.1 Stakeholders

Those who are involved in the project and work on it:

- Administrator. A group of people or an individual who is maintaining the entire system by monitoring and analyzing the application along with centralized database.
- Job Seeker. Job seeking people who are searching for jobs through the Job Portal Application.
- Job Provider. Who are recruiting employees by posting job offers through the application.

Internal and external stakeholders:

- Project implementation members
- Documentation writer
- Database handler
- UML designing member

3.2 System Features

- Registration
- Login
- Authenticating
- Updating Profile
- Posting a job and removing or modifying job
- Searching for a job
- Feedback

1. Registration
Functional Requirements

1.1 The software shall allow users to register with a valid username and password

1.2 If the username and/or password has been qualified with the system policy then the system will register their account.

Priority Level: High
Precondition: user has valid user id and password

2. System Login
Functional Requirements

2.1 The software shall allow users to login with their given username and password

2.2 If the username and/or password has been inserted wrong for more than three times, the random verification code will be generated by the system to retry login.

2.3 If the number of login attempt exceed its limit (5 times), the system shall block the user account login for one hour *[optional function]*

Priority Level: High
Precondition: user has valid user id and password

3. Updating Profile
Functional Requirements

3.1 The user can update the given information anytime with their own information.

3.2 After saving the information the system will update the information given by the user (job seeker/job provider) on the server.

Priority Level: Medium
Precondition: user has to input valid types of information without leaving blank the mandatory information.

4. Posting a job and removing or modifying job

Functional Requirements

4.1 A Job Provider can easily post or modify job post with inserting valid information on the system.

4.2 A posted job can also be removed.

Priority Level: High

Precondition: user has to input valid types of information without leaving blank the mandatory information.

5. Searching for a job

Functional Requirements

5.1 A Job Seeker can search jobs through the system.

5.2 The system will provide the user advanced filtering search facility from which a job seeker can easily find their desired jobs.

Priority Level: High

Precondition: Need job providers post on the database in order to search.

6. Feedback

Functional Requirements

6.1 Both Job seeker and provider can give feedback to the admin.

6.2 Data processing

6.3 Well understanding User Interfaces

6.4 Specific functions

Priority Level: High

7. Optimization

Non-Functional Requirements

7.1 The database should be well maintained in order to work properly.

7.2 The UI should be well polished and stable for better experience for the users

7.3. Response time and Net processing time

7.4. Maintainability

7.5. Security

7.6. Service level

7.7. Performance

7.8 Usability

7.9 Efficiency

Priority Level: High

3.3 System Quality Attributes

Correctness: The correctness of a software system refers to:

Agreement of program code with specifications

Independence of the actual application of the software system.

Learnability: Learnability of a software system depends on:

The design of user interfaces

The clarity and the simplicity of the user instructions**.**

Robustness: Robustness reduces the impact of operational mistakes, erroneous input

Data, and hardware errors.

Extensibility: Extensibility allows required modifications at the appropriate locations to

be made without undesirable side effects.

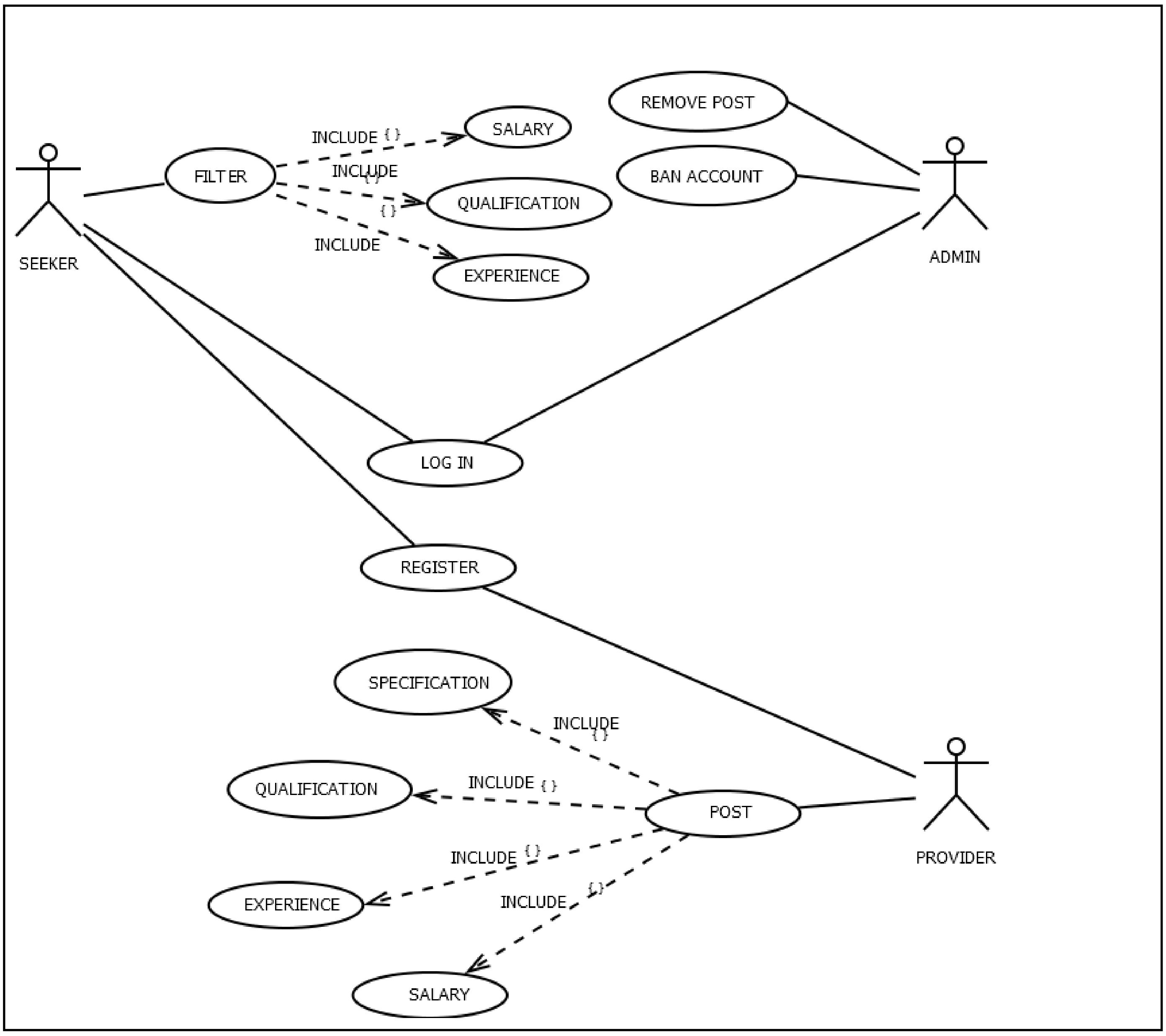
Efficiency: ability of a software system to fulfill its purpose with the best possible

utilization of all necessary resources

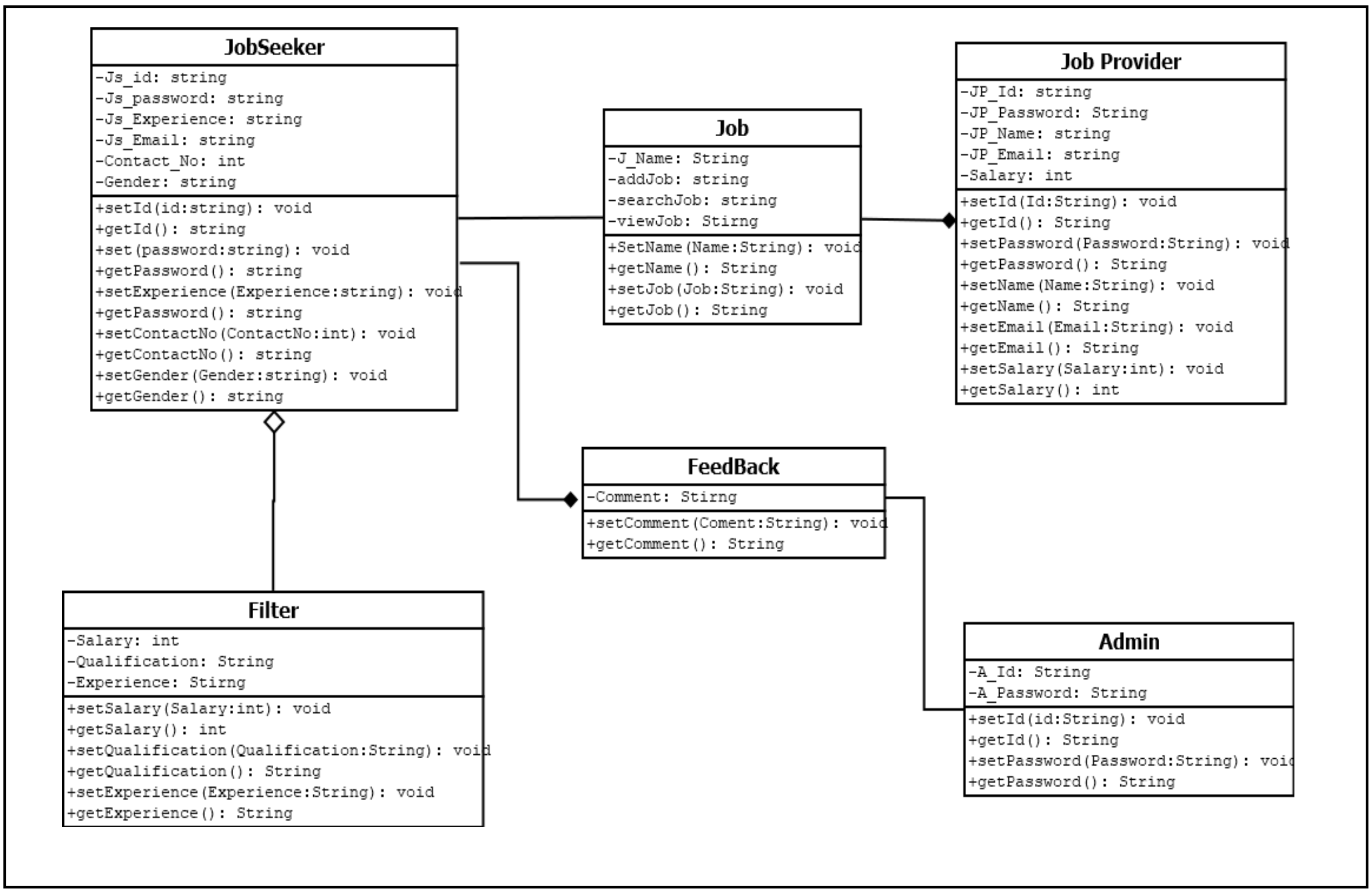
Usability: A trained user shall be able to submit a complete request for a chemical selected from a vendor catalog in an average of four and a maximum of six minutes.

3.4 System Architecture

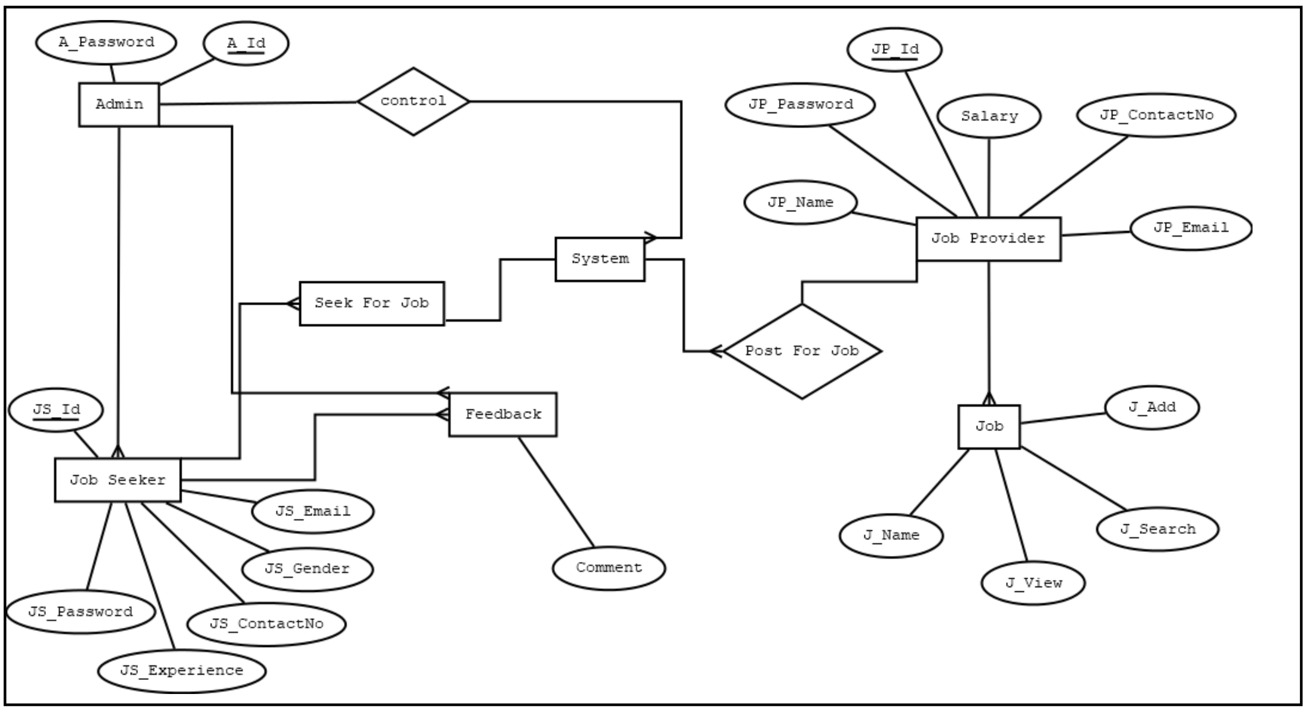
USE CASE DIAGRAM:



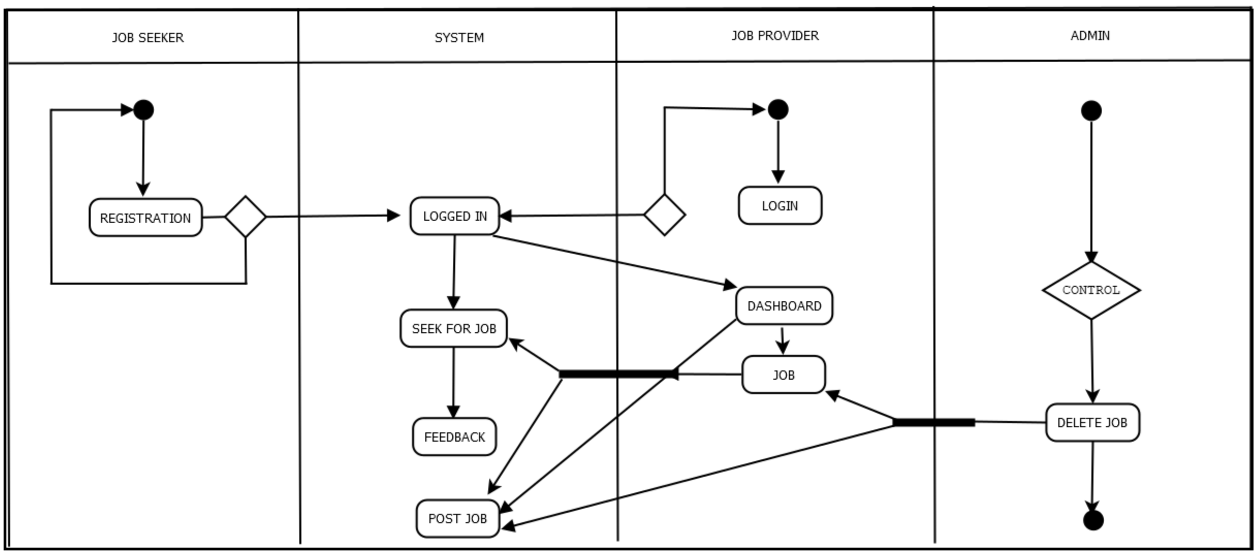
CLASS DIAGRAM



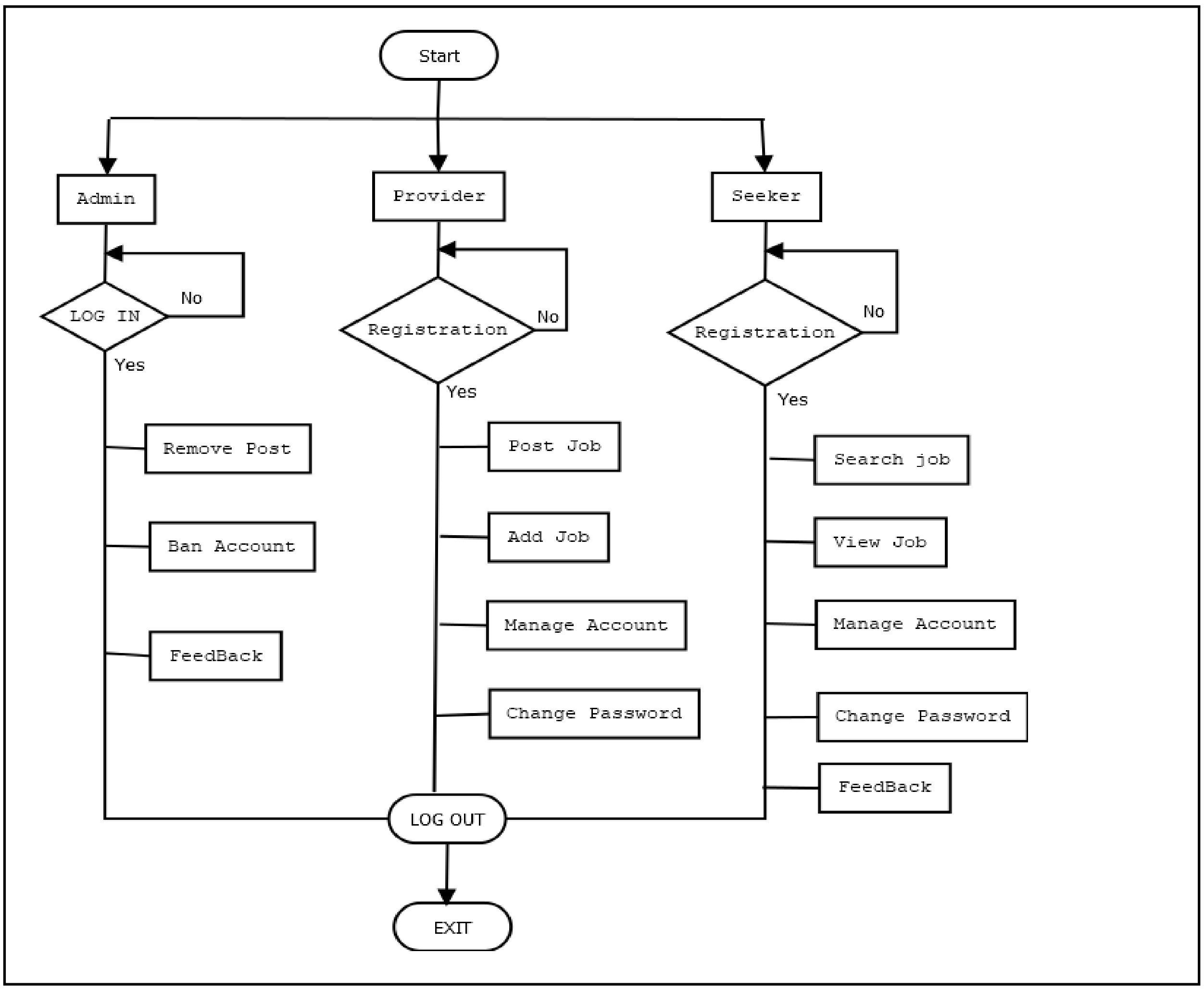
ER DIAGRAM



ACTIVITY DIAGRAM

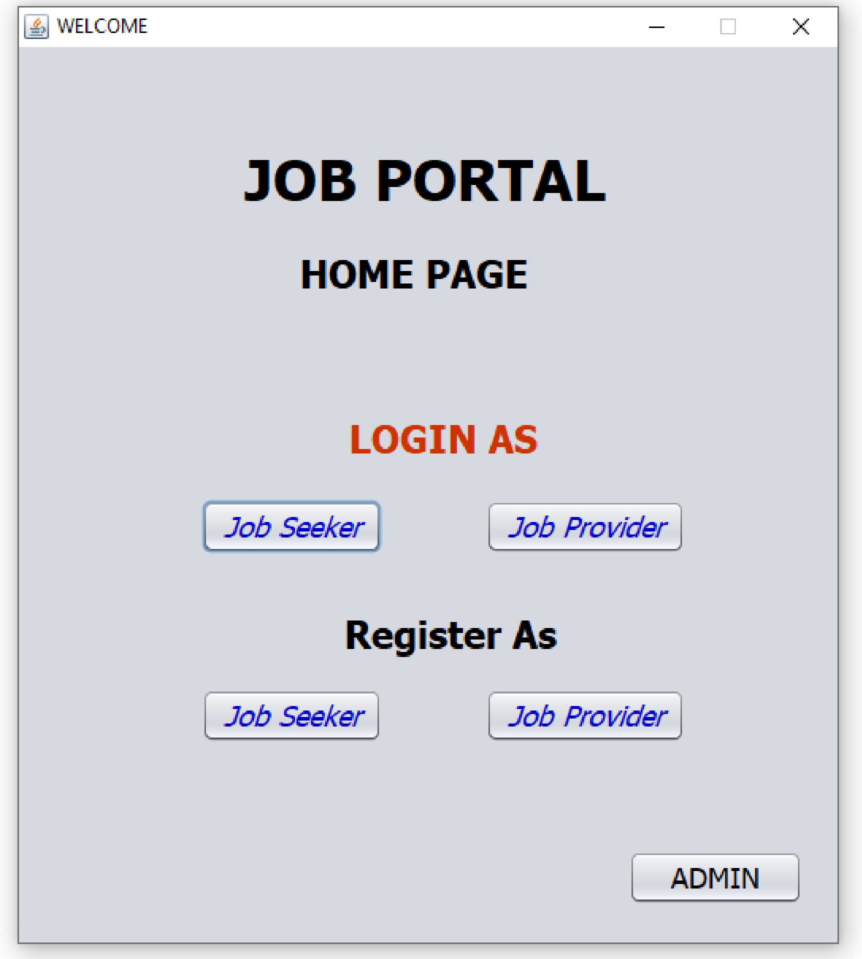


STATE CHART DIAGRAM

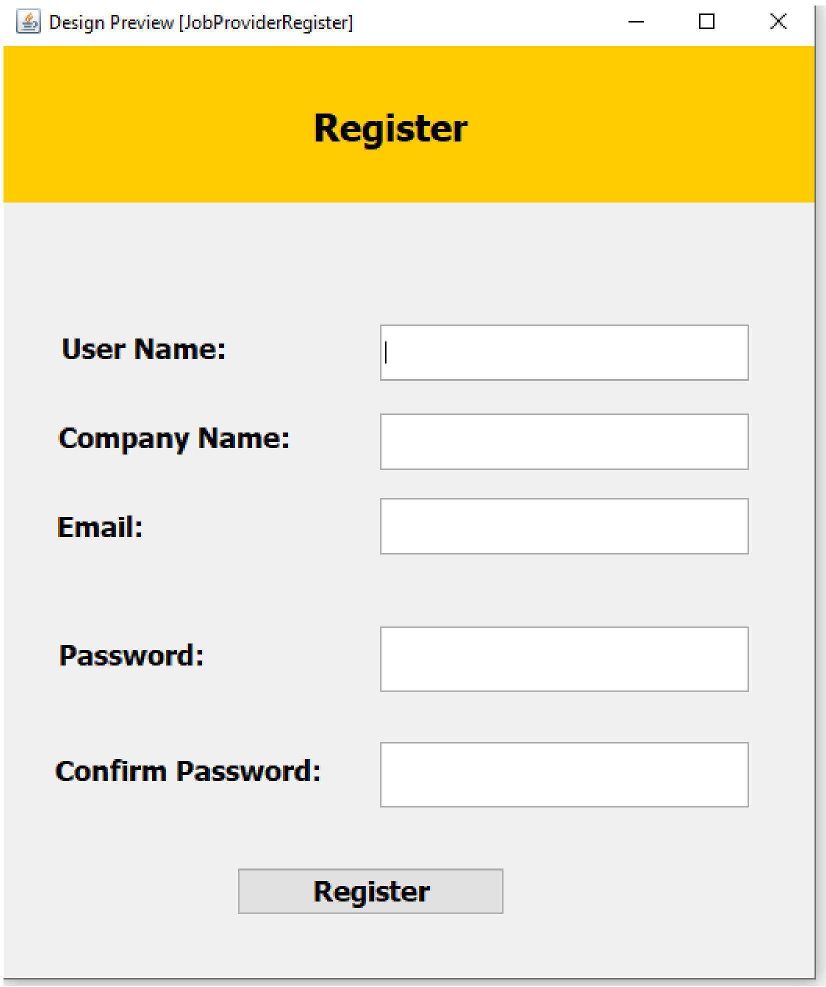


3.5 System Interface:

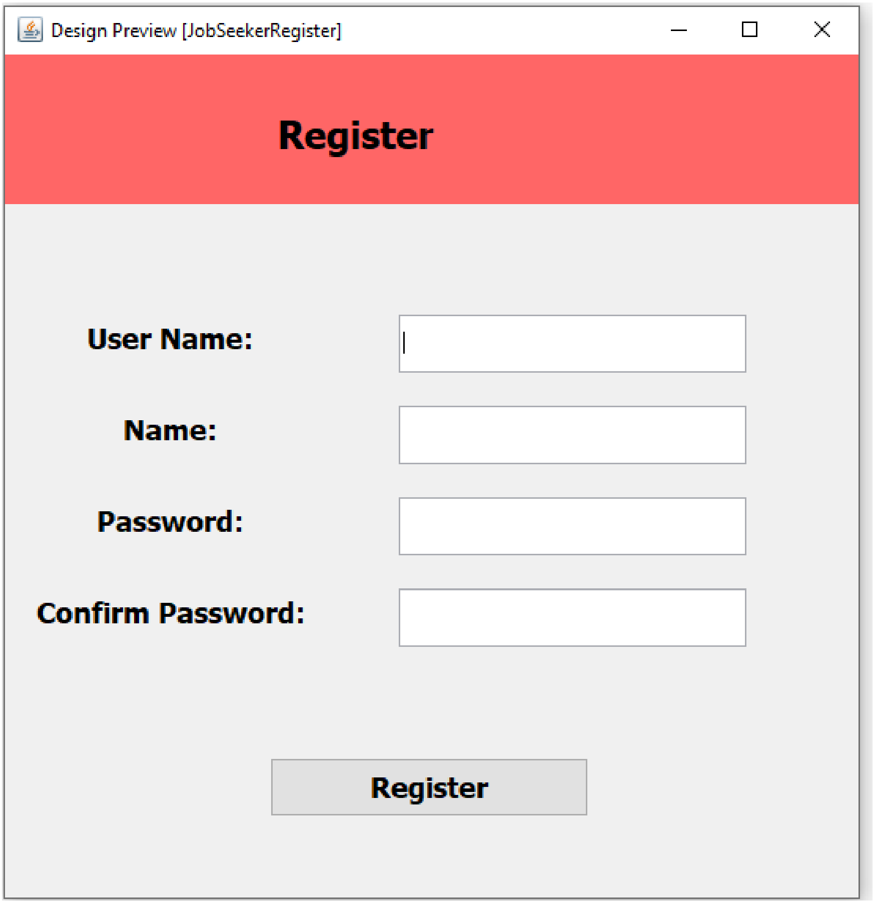
Home Page

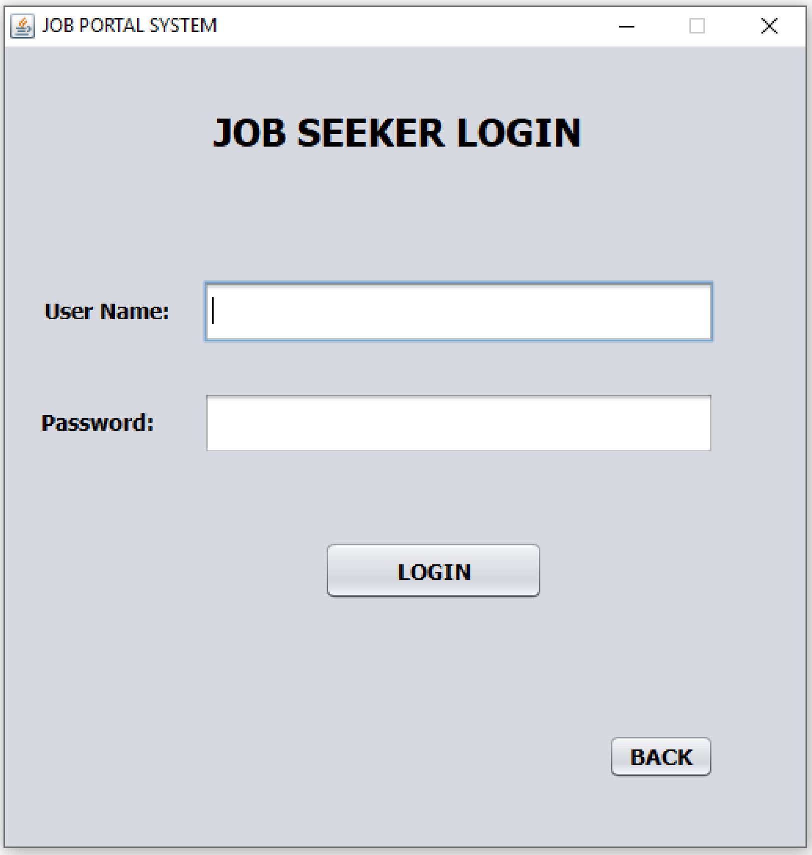


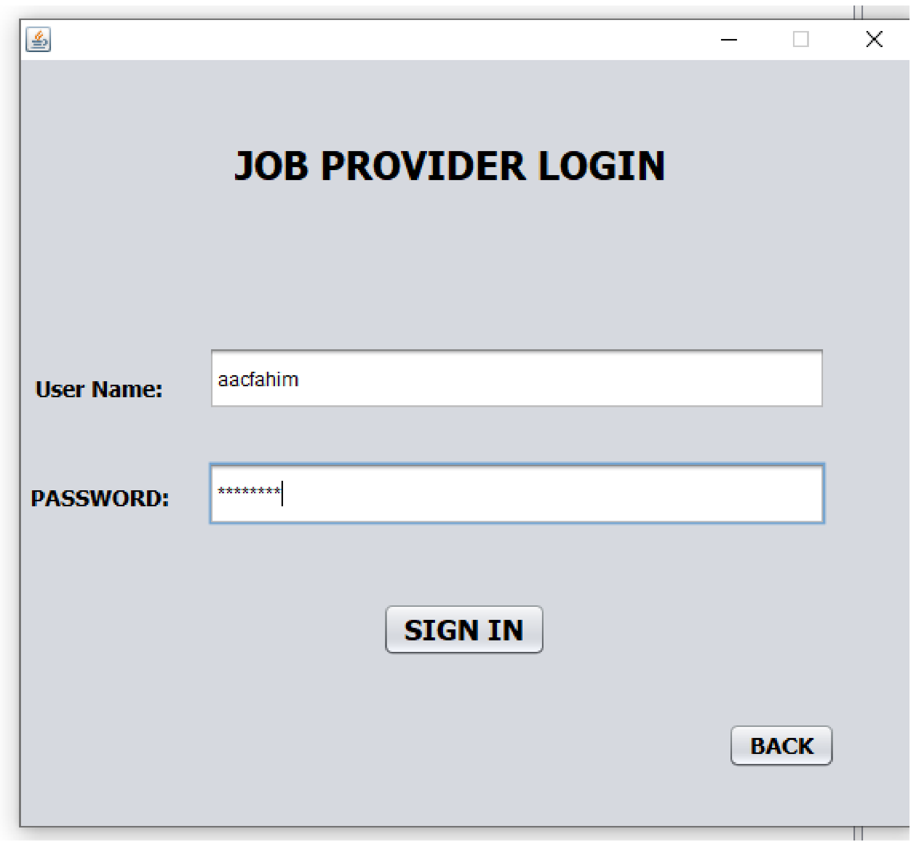
Job Provider Registration page



Job Seeker Registration page



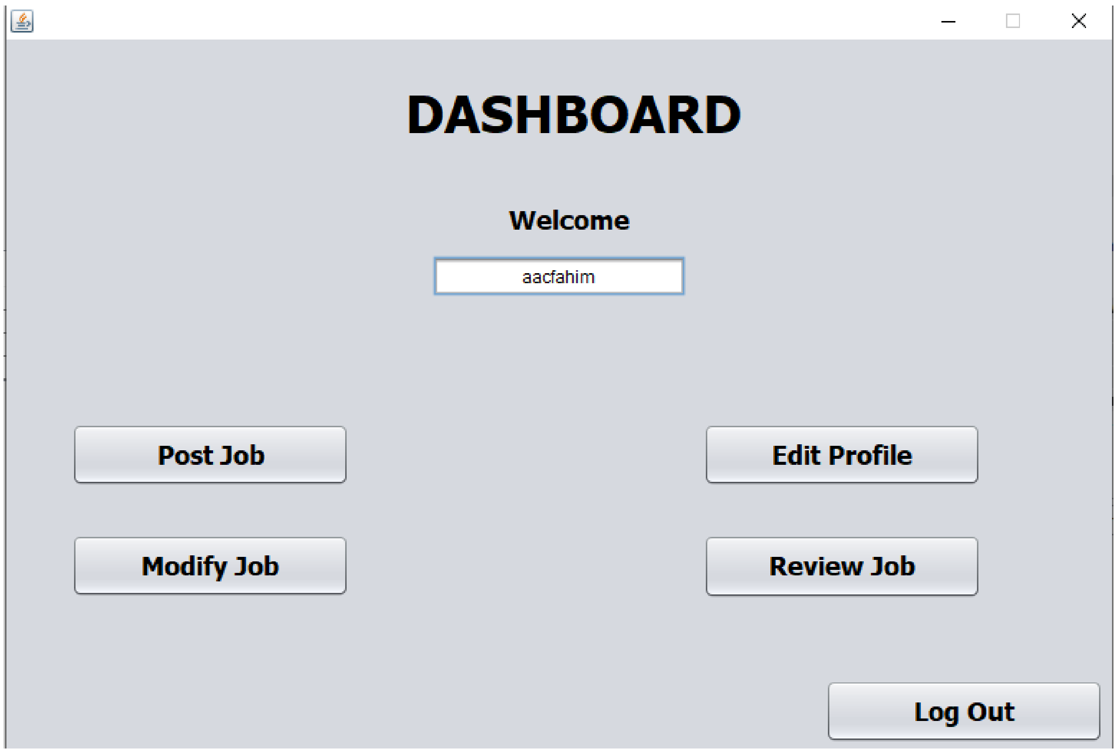




User Name: aacfahim

PASSWORD: ********

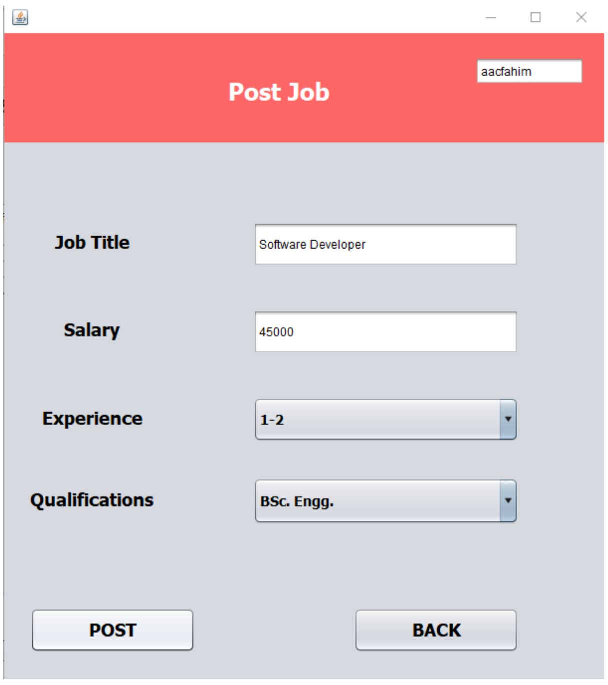
Job Provider’s Dashboard



Welcome

aacfahim

Posting jobs as job provider



aacfahim

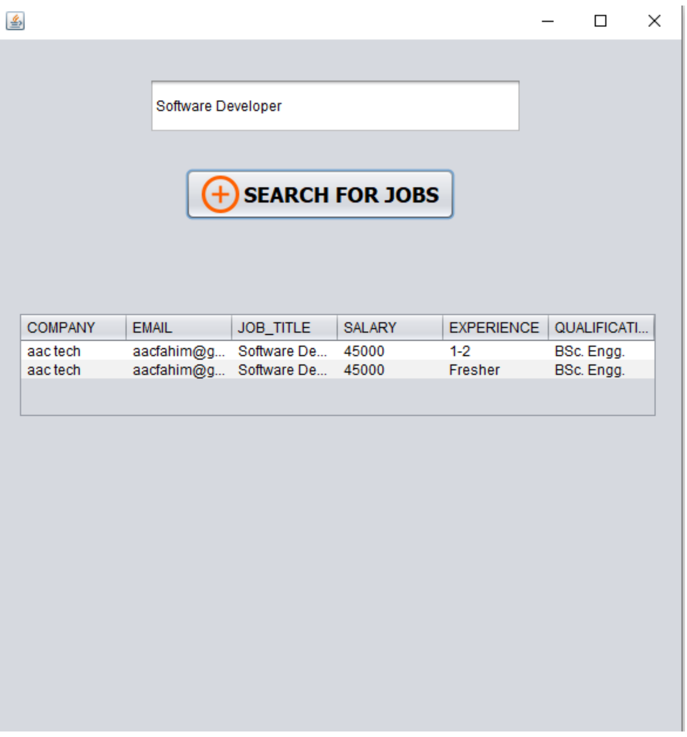
Job Title: Software Developer

Salary: 45000

Experience: 1-2

Qualifications: BSc. Engg.

Post searching as job seeker



Software Developer

COMPANY	EMAIL	JOB_TITLE	SALARY	EXPERIENCE	QUALIFICATI...
aac tech	aacfahim@g...	Software De...	45000	1-2	BSc. Engg.
aac tech	aacfahim@g...	Software De...	45000	Fresher	BSc. Engg.

4 Project Management

4.1 Project Roll Identification and Responsibilities

- Project implementation members: Ashfaq Afzal Chowdhury, Fahim Hoque, Iftikher Ahmed Lium, Rafiqul Islam,MD Minhazul Islam Omi
- Documentation writer: Iftikher Ahmed Lium, Ashfaq Afzal Chowdhury
- Database handler: Ashfaq Afzal Chowdhury, Fahim Hoque
- UML designing member: Rafiqul Islam, Iftikher Ahmed Lium
- Background studies: Minhazul Islam Omi, Ashfaq Afzal Chowdhury

4.2 COCOMO Model

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality. It was proposed by Barry Boehm in 1970 and is based on the study of 63 projects, which make it one of the best-documented models.

The key parameters which define the quality of any software products, which are also an outcome of the Cocomo are primarily Effort & Schedule:

Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.

Schedule: Simply means the amount of time required for the completion of the job, which is, of course, proportional to the effort put. It is measured in the units of time such as weeks, months.

Different models of Cocomo have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations. These characteristics pertaining to different system types are mentioned below.

Boehm’s definition of organic, semidetached, and embedded systems:

Organic – A software project is said to be an organic type if the team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.

Semi-detached – A software project is said to be a Semi-detached type if the vital characteristics such as team-size, experience, knowledge of the various programming environment lie in between that of organic and Embedded. The projects classified as Semi-Detached are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Ex: Compilers or different Embedded Systems can be considered of Semi-Detached type.

Embedded – A software project with requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

In Organic Mode

30000 LOC

Effort = 2.4 * (30 kloc) ^ 1.05

= 85.35

Delivery Time = DM = 2.50 * (85.35) ^ 0.38

= 13. 54

= 14 PER PERSON – DAY

Required Member = 85.35/13.54

= 6.30

4.3 Earned Value Analysis

Budget at completion (BAC) = 45 person-days

Planned Effort:

Budget cost of work performed (BCWP) = 102
Budget cost of work scheduled (BCWS) = 180
Actual cost of work performed (ACWP) = 103

BAC = 45
Schedule performance index (SPI) = BCWP/BCWS
= 102/130
= 0.78
Cost variance = BCWP – BCWS
= 102 – 130
= -28 person-day

CPI = BCWP/ACWP
= 102/103
= 0.99

Cost Variance = 102 – 103
= -1 person-day

5 References

1. "Kormo - Build a CV, Find jobs & Grow your career", Kormo, 2019. [Online]. Available: https://www.kormo.com/
2. "Largest Job Site in Bangladesh, Search Jobs | Bdjobs.com", Bdjobs.com, 2019. [Online]. Available: http://www.bdjobs.com/
3. "A Project Report On Online Job Portal – 1000 Projects", 1000projects.org, 2019. [Online]. Available: https://1000projects.org/project-report-online-job-portal.html.
4. "Manifesto for Agile Software Development", Agilemanifesto.org, 2019. [Online]. Available: https://agilemanifesto.org/
5. "R.S. Pressman & Associates, Inc.", Rspa.com, 2019. [Online]. Available: http://www.rspa.com/
6. Bandari, S. Sarita, S. Babu, M. Chaudhary, A. Team and G. Man, "What Are The Quality Attributes?", Softwaretestinghelp.com, 2019. [Online]. Available: https://www.softwaretestinghelp.com/what-are-the-quality-attributes/
7. "What is Feature Driven Development (FDD)? | Definition", Productplan.com, 2019. [Online]. Available: https://www.productplan.com/glossary/feature-driven-development/
8. "What is Scrum?", Scrum.org, 2019. [Online]. Available: https://www.scrum.org/resources/what-is-scrum
9. "What is Timeboxing? | How is Timeboxing used in Scrum? | Scrum Inc.", Scrum Inc, 2019. [Online]. Available: https://www.scruminc.com/what-is-timeboxing/
10. "What is an Agile Framework? | Definition and Overview", Productplan.com, 2019. [Online]. Available: https://www.productplan.com/glossary/agile-framework/
11. "Software Engineering | COCOMO Model - GeeksforGeeks", GeeksforGeeks, 2019. [Online]. Available: https://www.geeksforgeeks.org/software-engineering-cocomo-model