Project Report
On
Employment Dashboard & Services

**Submitted for the partial fulfillment of the requirement for
The award of the Degree of B. Tech**

By

Aniket kumar (2100960100012)
Harsh kr.Gupta(2100960100023)

Under the Supervision of: -
Mr. Manoj Rajora
(Faculty, CSE, Department)

Department of Computer Science and Engineering

Vishveshwarya Group of institution

Dr. A.P.J. Abdul Kalam Technical University, U.P., Lucknow

[2021-2025]

ABSTRACT

It is a simple to use and an effective web application platform that helps to provide employment & Services to countless numbers of people who are having specialization in their targeted particular field. This dashboard is exceptionally efficient, easy to use and secure.

Now a day, we should also have to implement a bridge for those who are not well educated like Carpenters, Painters, Plumbers, Electrician, Builders & etc.

This Dashboard provides a digital approach to generate the right employment opportunities, to develop connections with customers. A small alternative for both professional & unprofessional people .It gives plenty of opportunities for getting employment easily & safely.

It also allows customers to schedule events over a call. The Customer can directly make a call to the concerned person with whom they want to organize their events but the condition is, the user will have to register you to get the details of the service providers. It is free to use & any one can register you for their services & can earn money easily.

It offers personal profile hosting, helping you in stay connected to your customers and vice versa .

To catch the growth in any type of business we need a platform to provide services to the users. Therefore, this dashboard will help to every business owner to start at initial level to get employment easily.

TABLE OF CONTENT

Page No.

CHAPTER -1

INTRODUCTION

It is basically a Web Application. The main purpose of this application is to provide employment and while at the same side making easier for the customer to avail the services like Party Planners, Artist, Musician, A.C Services, Choreographers, Electricity, Photographer, Portraitist, etc. A small alternative for both professional & unprofessional people . It gives plenty of opportunities for getting employment easily & safely. The customer will get both the contact number and the email id of the Registered Service Provider from the dashboard but the condition is, the user should be already registered.

Register your name & start taking benefits and appreciation of your passion. We all know that, Appreciation leads to Future Growth in terms of Employment & many more. Now a day, we should also have to implement a bridge for those who are not well educated like Carpenters, Painters, Plumbers, Electrician, Builders & etc.

This Dashboard provides a digital approach to generate the right employment opportunities, to develop connections with customers. It also allows customers to schedule events over a call. The Customer can directly make a call to the concerned person with whom they want to organize their events but the condition is, the user will have to register yourself to get the details of the service providers. It is free to use & any one can register yourself for their services & can earn money easily.

 It offers personal profile hosting, helping you in stay connected to your customers and vice versa.

To catch the growth in any type of business we need a platform to provide services to the users. Therefore, this dashboard will help to every business owner to start at initial level to get employment easily.

Moreover, during this pandemic period, as we all know we are not allow to move outside the home due to which common man economic condition getting poor but a person having an ability to earn money can choose this application and continue their earning. Hence, this application will act as a boon for all small level businesses.

Benefits:

1) No need to go anywhere.

2) Time Saving

3) It makes financially strong & independent.

4) No need to wait for long hours to get services.

5) Ideal for last minute parties.

6) Just make a call & get multiple services.

7) Multitasking

8) User Friendly

1.1 SCOPE

The main motive of making this application is to provide employment and while at the same side making easier for the customer to avail the services like A.C Services, Laptop Repair, Party Planners, Artist, Musician, Choreographers, Electricity, Photographer, Portraitist, etc. A small alternative for both professional & unprofessional people .It gives plenty of opportunities for getting employment easily & safely.

1.2 SOFTWARE DEVELOPMENT METHODOLOGY

Agile Methodology - Agile Methodology meaning a practice that promotes continuous iteration of development and testing throughout the software development lifecycle of the project. In the Agile model in software testing, both development and testing activities are concurrent, unlike the Waterfall model.

The **Agile software development** methodology is one of the simplest and effective processes to turn a vision for a business need into software solutions. Agile is a term used to describe software development approaches that employ continual planning, learning, improvement, team collaboration, evolutionary development, and early delivery. It encourages flexible responses to change.

The agile software development emphasizes on four core values.

1. Individual and team interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

In this agile project management methodology, we have learned :

- What is Agile Methodology?
- Agile Model Vs Waterfall Model
- Scrum
- Product Backlog
- Scrum Practices
- Process flow of Scrum Methodologies:
- eXtreme Programming (XP)
- Phases of eXtreme programming:
- Crystal Methodologies
- Dynamic Software Development Method (DSDM)

Agile Model Vs Waterfall Model

Agile and Waterfall model are two different methods for software development process. Though they are different in their approach, both methods are useful at times, depending on the requirement and the type of the project.

Agile Model	Waterfall Model
• Agile methodology definition: Agile methodologies propose incremental and iterative approach to software design	• Waterfall Model: Development of the software flows sequentially from start point to end point.
• The **Agile process** in software engineering is broken into individual models that designers work on	• The design process is not broken into an individual models
• The customer has early and frequent opportunities to look at the product and make decision and changes to the project	• The customer can only see the product at the end of the project
• Agile model is considered unstructured compared to the waterfall model	• Waterfall model are more secure because they are so plan oriented
• Small projects can be implemented very quickly. For large projects, it is difficult to estimate the development time.	• All sorts of project can be estimated and completed.
• Error can be fixed in the middle of the project.	• Only at the end, the whole product is tested. If the requirement error is found or any changes have to be made, the project has to start from the beginning
• Development process is iterative, and the project is executed in short (2-4) weeks iterations. Planning is very less.	• The development process is phased, and the phase is much bigger than iteration.
• Documentation attends less priority than software development	• Documentation is a top priority and can even use for training staff and upgrade the software with another

	team
• Every iteration has its own testing phase. It allows implementing regression testing every time new functions or logic are released.	• Only after the development phase, the testing phase is executed because separate parts are not fully functional.
• In agile testing when an iteration end, shippable features of the product is delivered to the customer. New features are usable right after shipment. It is useful when you have good contact with customers.	• All features developed are delivered at once after the long implementation phase.
• Testers and developers work together	• Testers work separately from developers
• At the end of every sprint, user acceptance is performed	• User acceptance is **performed** at the end of the project.
• It requires close communication with developers and together analyze requirements and planning	• Developer does not involve in requirement and planning process. Usually, time delays between tests and coding

CHAPTER-2

EFFORT AND COST ESTIMATION

Estimation is the process of finding an estimate, or approximation, which is a value that can be used for some purpose even if input data may be incomplete, uncertain, or unstable.

Estimation determines how much money, effort, resources, and time it will take to build a specific system or product. Estimation is based on −

- Past Data/Past Experience
- Available Documents/Knowledge
- Assumptions
- Identified Risks

The four basic steps in Software Project Estimation are −

- Estimate the size of the development product.
- Estimate the effort in person-months or person-hours.
- Estimate the schedule in calendar months.
- Estimate the project cost in agreed currency.

Observations on Estimation

- Estimation need not be a one-time task in a project. It can take place during −

    o Acquiring a Project.
    o Planning the Project.
    o Execution of the Project as the need arises.

- Project scope must be understood before the estimation process begins. It will be helpful to have historical Project Data.

- Project metrics can provide a historical perspective and valuable input for generation of quantitative estimates.

- Planning requires technical managers and the software team to make an initial commitment as it leads to responsibility and accountability.

- Past experience can aid greatly.

- Use at least two estimation techniques to arrive at the estimates and reconcile the resulting values. Refer Decomposition Techniques in the next section to learn about reconciling estimates.

- Plans should be iterative and allow adjustments as time passes and more details are known.

General Project Estimation Approach

The Project Estimation Approach that is widely used is **Decomposition Technique**. Decomposition techniques take a divide and conquer approach. Size, Effort and Cost estimation are performed in a stepwise manner by breaking down a Project into major Functions or related

Software Engineering Activities.

Step 1 − Understand the scope of the software to be built.

Step 2 − Generate an estimate of the software size.

- Start with the statement of scope.

- Decompose the software into functions that can each be estimated individually.

- Calculate the size of each function.

- Derive effort and cost estimates by applying the size values to your baseline productivity metrics.

- Combine function estimates to produce an overall estimate for the entire project.

Step 3 − Generate an estimate of the effort and cost. You can arrive at the effort and cost estimates by breaking down a project into related software engineering activities.

- Identify the sequence of activities that need to be performed for the project to be completed.

- Divide activities into tasks that can be measured.

- Estimate the effort (in person hours/days) required to complete each task.

- Combine effort estimates of tasks of activity to produce an estimate for the activity.

- Obtain cost units (i.e., cost/unit effort) for each activity from the database.

- Compute the total effort and cost for each activity.

- Combine effort and cost estimates for each activity to produce an overall effort and cost estimate for the entire project.

Step 4 − Reconcile estimates: Compare the resulting values from Step 3 to those obtained from Step 2. If both sets of estimates agree, then your numbers are highly reliable. Otherwise, if widely divergent estimates occur conduct further investigation concerning whether −

- The scope of the project is not adequately understood or has been misinterpreted.

- The function and/or activity breakdown is not accurate.

- Historical data used for the estimation techniques is inappropriate for the application, or obsolete, or has been misapplied.

Step 5 − Determine the cause of divergence and then reconcile the estimates.

Estimation Accuracy

Accuracy is an indication of how close something is to reality. Whenever you generate an estimate, everyone wants to know how close the numbers are to reality. You will want every estimate to be as accurate as possible, given the data you have at the time you generate it. And of course you don't want to present an estimate in a way that inspires a false sense of confidence in the numbers.

Important factors that affect the accuracy of estimates are −

- The accuracy of the entire estimate's input data.

- The accuracy of any estimate calculation.

- The predictability of your organization's software development process.

- The stability of both the product requirements and the environment that supports the software engineering effort.

- Whether or not the actual project was carefully planned, monitored and controlled, and no major surprises occurred that caused unexpected delays.

Following are some guidelines for achieving reliable estimates −

- Base estimates on similar projects that have already been completed.
- Use relatively simple decomposition techniques to generate project cost and effort estimates.
- Use one or more empirical estimation models for software cost and effort estimation.

Refer to the section on Estimation Guidelines in this chapter.

To ensure accuracy, you are always advised to estimate using at least two techniques and compare the results.

Estimation Issues

Often, project managers resort to estimating schedules skipping to estimate size. This may be because of the timelines set by the top management or the marketing team. However, whatever the reason, if this is done, then at a later stage it would be difficult to estimate the schedules to accommodate the scope changes.

While estimating, certain assumptions may be made. It is important to note all these assumptions in the estimation sheet, as some still do not document assumptions in estimation sheets.

Even good estimates have inherent assumptions, risks, and uncertainty, and yet they are often treated as though they are accurate.

The best way of expressing estimates is as a range of possible outcomes by saying, for example, that the project will take 5 to 7 months instead of stating it will be complete on a particular date or it will be complete in a fixed no. of months. Beware of committing to a range that is too narrow as that is equivalent to committing to a definite date.

- You could also include uncertainty as an accompanying probability value. For example, there is a 90% probability that the project will complete on or before a definite date.

- Organizations do not collect accurate project data. Since the accuracy of the estimates depends on the historical data, it would be an issue.

- For any project, there is a shortest possible schedule that will allow you to include the required functionality and produce quality output. If there is a schedule constraint by management and/or client, you could negotiate on the scope and functionality to be delivered.

- Agree with the client on handling scope creeps to avoid schedule overruns.

- Failure in accommodating contingency in the final estimate causes issues. For e.g., meetings, organizational events.

- Resource utilization should be considered as less than 80%. This is because the resources would be productive only for 80% of their time. If you assign resources at more than 80% utilization, there is bound to be slippages.

Estimation Guidelines

One should keep the following guidelines in mind while estimating a project −

- During estimation, ask other people's experiences. Also, put your own experiences at task.

- Assume resources will be productive for only 80 percent of their time. Hence, during estimation take the resource utilization as less than 80%.

- Resources working on multiple projects take longer to complete tasks because of the time lost switching between them.

- Include management time in any estimate.

- Always build in contingency for problem solving, meetings and other unexpected events.

- Allow enough time to do a proper project estimate. Rushed estimates are inaccurate, high-risk estimates. For large development projects, the estimation step should really be regarded as a mini project.

- Where possible, use documented data from your organization's similar past projects. It will result in the most accurate estimate. If your organization has not kept historical data, now is a good time to start collecting it.

- Use developer-based estimates, as the estimates prepared by people other than those who will do the work will be less accurate.

- Use several different people to estimate and use several different estimation techniques.

- Reconcile the estimates. Observe the convergence or spread among the estimates. Convergence means that you have got a good estimate. Wideband-Delphi technique can be used to gather and discuss estimates using a group of people, the intention being to produce an accurate, unbiased estimate.

- Re-estimate the project several times throughout its life cycle.

CHAPTER-3

<u>SRS</u>

3.1 INTRODUCTION

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfil all stakeholders (business, users) needs. A typical SRS includes:

- A purpose
- An overall description
- Specific requirements

The best SRS documents define how the software will interact when embedded in hardware — or when connected to other software. Good SRS documents also account for real-life users.

Why Use an SRS Document?

A software requirements specification is the basis for your entire project. It lays the framework that every team involved in development will follow.

It's used to provide critical information to multiple teams — development, quality assurance, operations, and maintenance. This keeps everyone on the same page.

Using the SRS helps to ensure requirements are fulfilled. And it can also help you make decisions about your product's lifecycle — for instance, when to retire a feature.

Writing an SRS can also minimize overall development time and costs. Embedded development teams especially benefit from using an SRS.

Software **Requirements Specification vs.** *System* **Requirements Specification**

A **software requirements specification (SRS)** includes in-depth descriptions of the software that will be developed.

A **system requirements specification (SRS)** collects information on the requirements for a system.

"Software" and "system" are sometimes used interchangeably as SRS. But, a software requirement specification provides greater detail than a system requirements specification.

3.2 INTENDED AUDIENCE AND READING SUGGESTIONS

While the software requirement specification (SRS) document is written for a more general audience, this document is intended for individuals directly involved in the development of Split Pay. This includes software developers, project consultants, and team managers. This document need not be read sequentially; users are encouraged to jump to any section they find relevant. Below is a brief overview of each part of the document.

•Part 1 (Introduction) - This section offers a summary of the Split Pay project, including goals and objectives, project scope, general system details, and some major constraints associated with the intended platform.

•Part 2 (Data Design) - Readers interested in how Split Pay organizes and handles data should consult this section, which covers data structures and flow patterns utilized by the system.

 •Part 3 (Architectural and Component-Level Design) - This section describes the Split Pay system class by class, including interface details, class hierarchies, performance/design constraints, process details, and algorithmic models.

 •Part 4 (User Interface Design) - This section covers all of the details related to the structure of the graphical user interface (GUI), including some preliminary mockups of the Split Pay Android application. Readers can view this section for a tentative glimpse of what the final product will look like.

 •Part 5 (Restrictions, Limitations, and Constraints) - This section discusses the general constraints imposed upon the project

•Part 6 (Testing Issues) - Readers interested in the software testing process should consult this section, which offers a list of test cases, expected responses, and other pertinent information.

•Part 7 (Appendices) -This section includes any additional information which may be helpful to readers.

3.3 GENERAL ARCHITECTURE OF SOFTWARE

Software architecture exposes the structure of a system while hiding the implementation details. Architecture also focuses on how the elements and components within a system interact with one other. Software design delves deeper into the implementation details of the system.

3.4 REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill all stakeholders (business, users) needs.

3.4.1 FUNCTIONAL REQUIREMENTS

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the

form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

3.4.2 NON-FUNCTIONAL REQUIREMENTS

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.
They basically deal with issues like:
- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

3.5 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources.
Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.
The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Behavioral feasibilities. The following are its features:

3.5.1 OPERATIONAL FEASIBILITY

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization's needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

3.5.2 TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:
- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology. Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

3.5.3 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:
- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

3.6 SYSTEM REQUIREMENTS STUDY

System Requirement Study plays a very vital role for the user. The developer is developing the project but if the development is not done without keeping in mind for the current generation specification computers, operating system, RAM, Hard disk then at the end client will not be able to meet with their desired output. Therefore, System Requirement Study should be done before developing any project.

3.6.1 SOFTWARE REQUIREMENTS

Operating System   :   Win7 / Win8 / Win10 / MAC / LINUX / UNIX

Languages   :   Java

Front End   :   JSP, Boot Strap-4

Platform   :   JAVA (J2EE)

Backend   :   Servlet, MySQL

Supporting Software   :   Eclipse, Spring Tool Suit, SQL Query Browser

3.6.2 HARDWARE REQUIREMENTS

Processor   :   Pentium IV / AMD Processor/i3/i5/etc

RAM   :   2/4 GB

Hard disk   :   250/500 GB

Monitor   :   15 inch

Mouse   :   3 Button scroll

CD Drive   :   Not Necessary

Keyboard   :   108 keys - Normal Keyboard

3.7 USER REQUIREMENT DOCUMENT (URD)

The user requirement(s) document (URD) or user requirement(s) specification (URS) is a document usually used in software engineering that specifies what the user expects the software to be able to do.

Once the required information is completely gathered it is documented in a URD, which is meant to spell out exactly what the software must do and becomes part of the contractual agreement. A customer cannot demand features not in the URD, while the developer cannot claim the product is ready if it does not meet an item of the URD.

The URD can be used as a guide for planning cost, timetables, milestones, testing, etc. The explicit nature of the URD allows customers to show it to various stakeholders to make sure all necessary features are described.

Formulating a URD requires negotiation to determine what is technically and economically feasible. Preparing a URD is one of those skills that lie between a science and an art, requiring both software technical skills and interpersonal skills.

3.7.1   USE-CASE DIAGRAM

In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or

external systems.

- Goals that your system or application helps those entities (known as actors) achieve

- The scope of your system
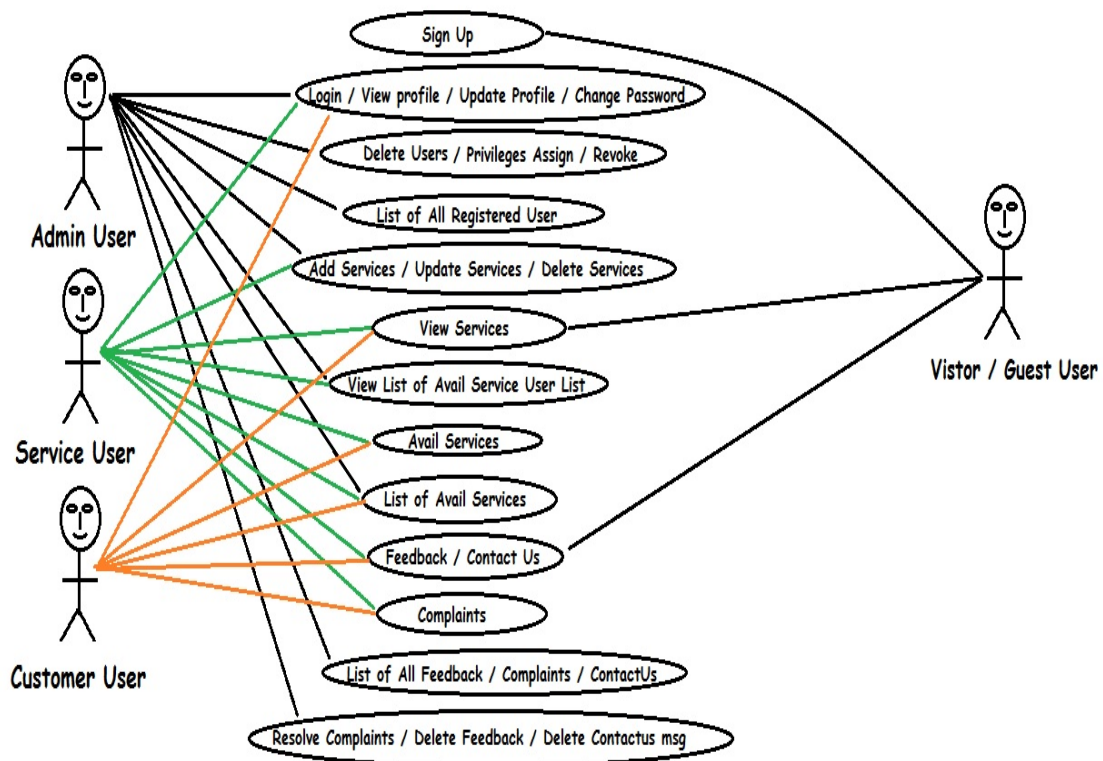
3.7.2 ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Activity Diagram for Employment Dashboard & Services
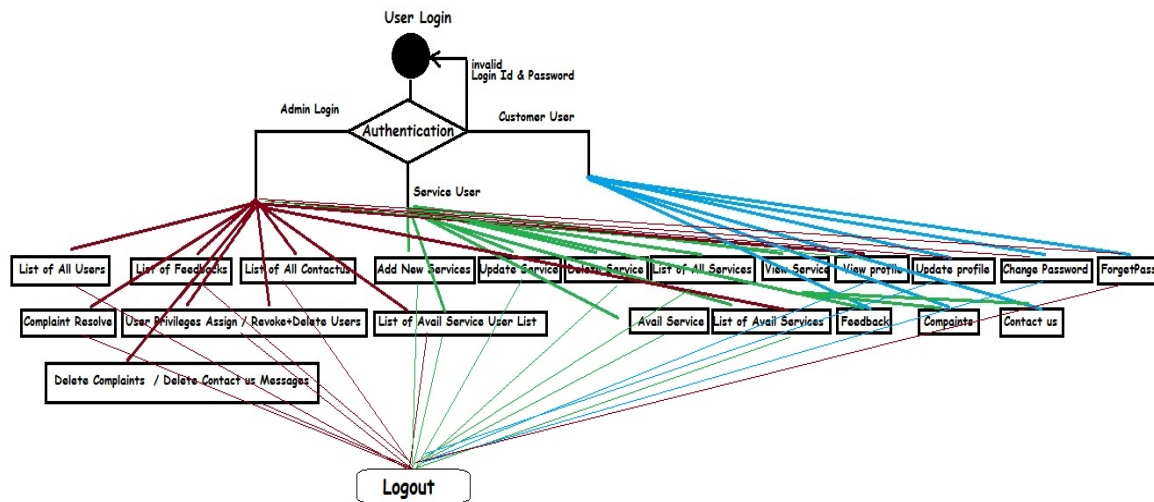
3.8 SYSTEM DESIGN

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

3.8.1 INTRODUCTION

A systemic approach is required for a coherent and well-running system. Bottom-Up or Top-Down approach is required to take into account all related variables of the system. A designer uses the modeling languages to express the information and knowledge in a structure of system that is defined by a consistent set of rules and definitions. The designs can be defined in graphical or textual modeling languages.

Some of the examples of graphical modeling languages are

a. Unified Modeling Language (UML): To describe software both structurally and behaviorally with graphical notation.

b. Flowchart: A schematic or stepwise representation of an algorithm.

c. Business Process Modeling Notation (BPMN): Used for Process Modeling language.

d. Systems Modeling Language (Sys ML): Used for systems engineering.

Design methods:

1) Architectural design: To describe the views, models, behavior, and structure of the system.

2) Logical design: To represent the data flow, inputs and outputs of the system. Example: ER Diagrams (Entity Relationship Diagrams).

3) Physical design: Defined as a) How users add information to the system and how the system represents information back to the user. b) How the data is modeled and stored within the system. c) How data moves through the system, how data is validated, secured and/or transformed as it flows through and out of the system.
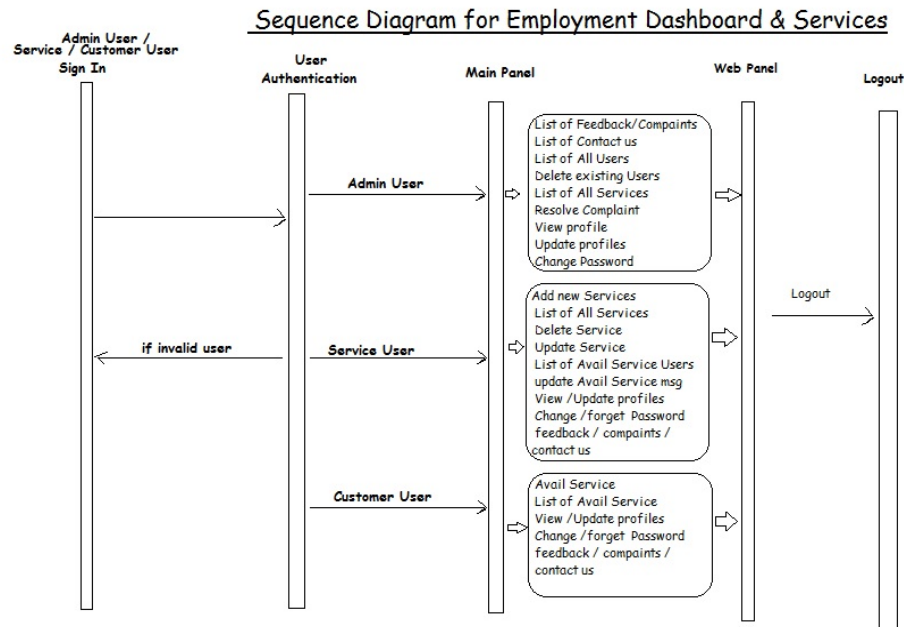
3.8.2   DATA FLOW DIAGRAM

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

<u>DATA FLOW DIAGRAM:-</u>
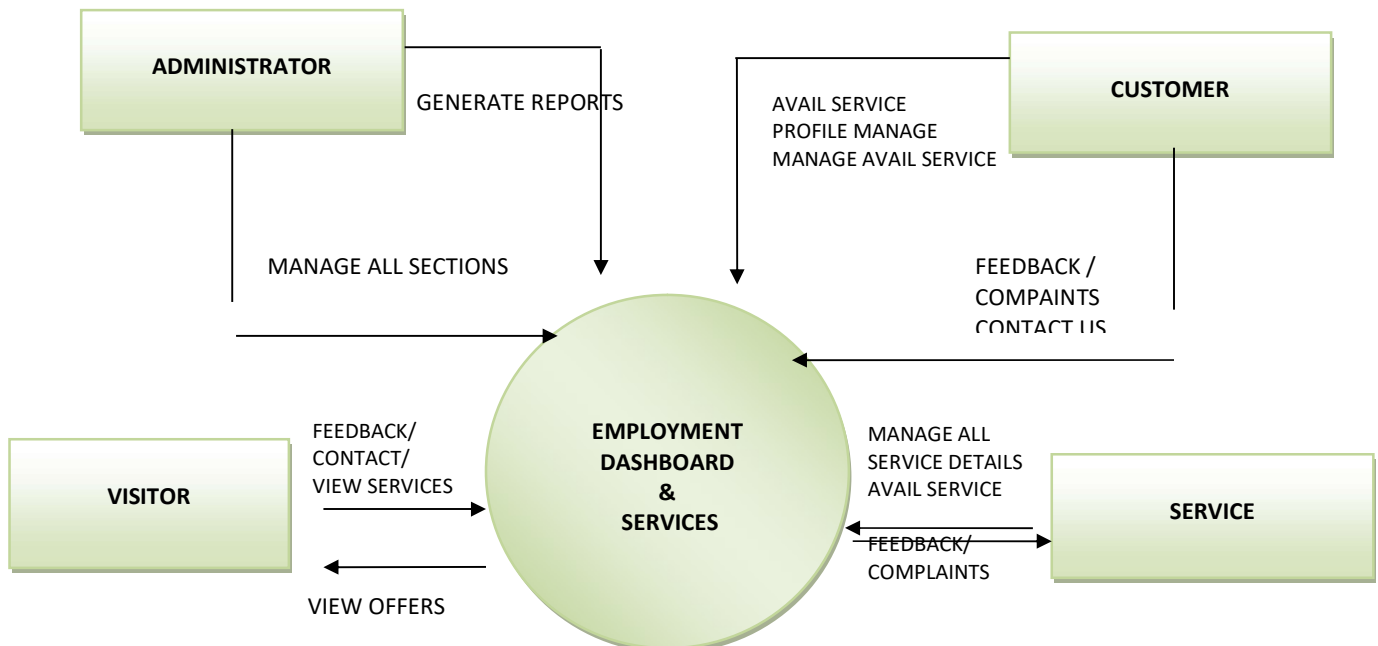
3.8.3 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios. There are two types of sequence diagrams: UML diagrams and code-based diagrams.
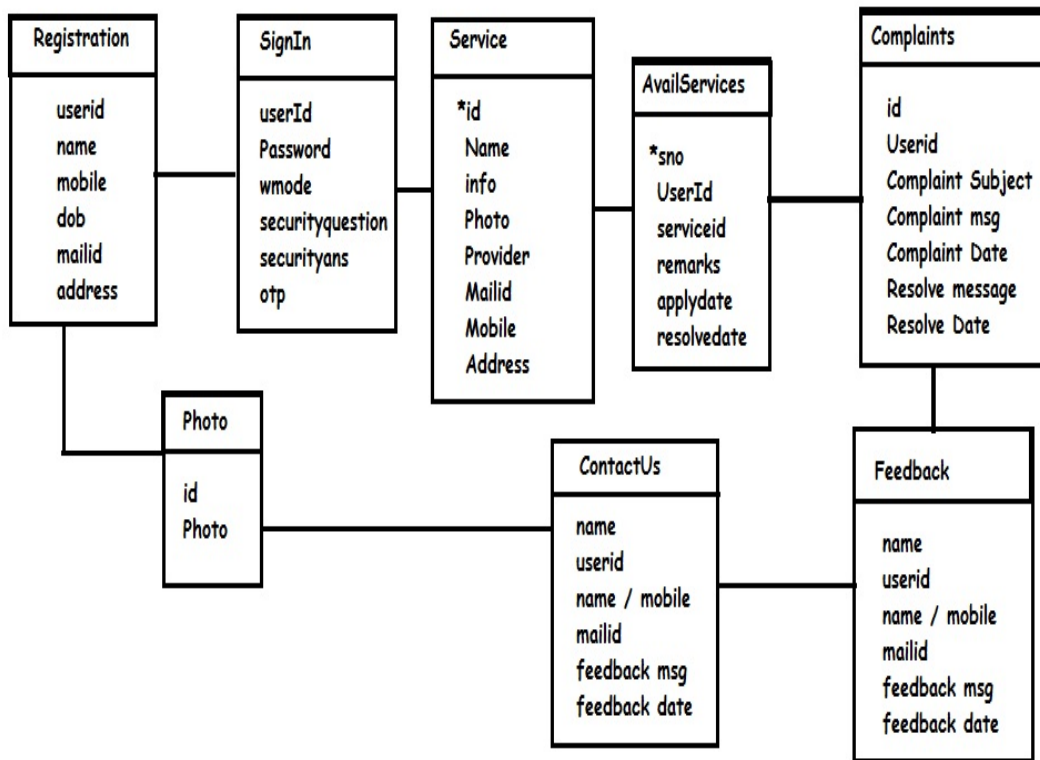
Sequence Diagram for Employment Dashboard & Services

Admin User /
Service / Customer User
Sign In

User
Authentication

Main Panel

Web Panel

Logout

Admin User

List of Feedback/Compaints
List of Contact us
List of All Users
Delete existing Users
List of All Services
Resolve Complaint
View profile
Update profiles
Change Password

if invalid user

Service User

Add new Services
List of All Services
Delete Service
Update Service
List of Avail Service Users
update Avail Service msg
View /Update profiles
Change /forget Password
feedback / compaints /
contact us

Logout

Customer User

Avail Service
List of Avail Service
View /Update profiles
Change /forget Password
feedback / compaints /
contact us

3.8.4 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

17

Class Diagram for Employment Dashboard & Services

Registration
- userid
- name
- mobile
- dob
- mailid
- address

SignIn
- userId
- Password
- wmode
- securityquestion
- securityans
- otp

Service
- *id
- Name
- info
- Photo
- Provider
- Mailid
- Mobile
- Address

AvailServices
- *sno
- UserId
- serviceid
- remarks
- applydate
- resolvedate

Complaints
- id
- Userid
- Complaint Subject
- Complaint msg
- Complaint Date
- Resolve message
- Resolve Date

Photo
- id
- Photo

ContactUs
- name
- userid
- name / mobile
- mailid
- feedback msg
- feedback date

Feedback
- name
- userid
- name / mobile
- mailid
- feedback msg
- feedback date

CHAPTER-4

<u>Snapshots</u>

List of All Complaints

List of All Feedbacks

List of All Contact Info

List of All Users

List of All Services

Search Services

SNO	USERID	SUBJECT	MESSAGE		COMPLAINT_DATE	COMPLAINT_STATUS	RESOLVEMESSAGE	RESOLVE_DATE	Action
1	manoj	regarding services	I have Register My New Service but I wan... how to remove that service.	ide me	2023-07-31 00:14:02.0	RESOLVED	We will Manage, dont worry	2023-07-31 23:31:45.0	✏ ⎮ 🗑
2	manoj	regarding UnRegister	I have Registration in ur panel, not i want procedure for that	e	2023-07-31 00:13:06.0	RESOLVED	We will Deleted, ur Details....	2023-07-31 01:06:26.0	✏ ⎮ 🗑

20

CHAPTER-5

<u>TECHNOLOGY USED</u>

5.1 JAVA

With the growing importance of web based and mobile based applications, Java today is the foundation for most networked applications and is also considered to be useful for standalone application. Java programming language was primarily developed to deal with embedded applications. Now the trend is to use Java technology in developing large web and enterprise applications. Java is more focused on web & desktop applications. Java is as a development technology for standalone desktop application development in present scenario.

• Java offers higher cross- functionality and portability as programs written in one platform can run across desktops, mobiles, embedded systems.

• Java is free, simple, object-oriented, distributed, supports multithreading and offers multimedia and network support.

• Java is a mature language, therefore more stable and predictable. The Java Class Library enables cross-platform development.

• Being highly popular at enterprise, embedded and network level, Java has a large active user community and support available.

• Unlike C and C++, Java programs are compiled independent of platform in bytecode language which allows the same program to run on any machine that has a JVM installed.

• Java has powerful development tools like Eclipse SDK and NetBeans which have debugging capability and offer integrated development environment.

• Increasing language diversity, evidenced by compatibility of Java with Scala, Groovy, JRuby, and Clojure.

• Relatively seamless forward compatibility from one version to the next. The future scope of java is good with advance technologies that are why we have decided to continue this project in java. We can easily get it to the largest scale in future by dealing with advance technologies. With the help of java, we would also able to make both android and Web application for this ideology

5.1.1 SERVLET & JSP

Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language. However, there were many disadvantages to this technology. We have discussed these disadvantages below.There are many interfaces and classes in the Servlet API such as Servlet, Generic Servlet, HTTP Servlet, Servlet Request,

Servlet Response, etc

Now Basically, JSP technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

5.2 DATABASE

It is very important to understand the database before learning MySQL. A database is an application that stores the organized collection of records. It can be accessed and manage by the user very easily. It allows us to organize data into tables, rows, columns, and indexes to find the relevant information very quickly. Each database contains distinct API for performing database operations such as creating, managing, accessing, and searching the data it stores. Today, many databases available like MySQL, Sybase, Mongo dB, SQL Server, etc. In our website, we have used MySQL Database.

5.2.1 MYSQL DATABASE

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

It is developed, marketed, and supported by **MySQL AB, a Swedish company**. The official pronunciation of MySQL is not the My Sequel; it is *My Ess Que Ell*. *However, you can pronounce it in your way*. Many small and big companies use MySQL. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java Languages.

MySQL is a Relational Database Management System (RDBMS) software that provides many things, which are as follows:

o It allows us to implement database operations on tables, rows, columns, and indexes.

o It defines the database relationship in the form of tables (collection of rows and columns), also known as relations.

o It provides the Referential Integrity between rows or columns of various tables.

o It allows us to updates the table indexes automatically.

o It uses many SQL queries and combines information from multiple tables

CHAPTER-6

<u>TESTING AND INTEGRATION</u>

6.1. TEST CASE DESCRIPTION

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values; the application has a definitive outcome and leaves the system at some end point or also known as execution post condition.

Why we write the test cases?

We will write the test for the following reasons:

To require consistency in the test case execution: we will see the test case and start testing the application.

To make sure a better test coverage: for this, we should cover all possible scenarios and document it, so that we need not remember all the scenarios again and again.

It depends on the process rather than on a person: A test engineer has tested an application during the first release, second release, and left the company at the time of third release. As the test engineer understood a module and tested the application thoroughly by deriving many values. If the person is not there for the third release, it becomes difficult for the new person. Hence all the derived values are documented so that it can be used in the future.

To avoid giving training for every new test engineer on the product: When the test engineer leaves, he/she leaves with a lot of knowledge and scenarios. Those scenarios should be documented so that the new test engineer can test with the given scenarios and also can write the new scenarios.

6.2. TYPES OF TESTING

Manual Testing (Self Testing)

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually.

Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting.

Manual testing is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software. Manual Testing requires knowledge of manual testing techniques but not of any automated testing tool.

Manual testing is essential because one of the software testing fundamentals is "100% automation is not possible."

Unit Testing (Module by Module Testing)

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.

The main aim is to isolate each unit of the system to identify, analyse and fix the defects.

Unit Testing - Advantages:

- Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.

- Reduces Cost of Testing as defects are captured in very early phase.

- Improves design and allows better refactoring of code.

- Unit Tests, when integrated with build gives the quality of the build as well.

Project Testing:

Project Testing is done after all these manual & unit testing. After this, the Project is ready to work according to the ideology.

6.3. FUTURE ENHANCEMENT

-Payment gateway is not involved Right Now.
-In Future we will also try to track the performance of our project.
-Ads mean Marketing & promotion of website.
-Add more advance features according to the feedback of the customer.
-To develop an Android Application.

<u>CONCLUSION</u>

- The project exposed us to the latest technologies in the area of java projects with Servlet (Web Projects).
- Moreover, this project ideology provides employment in easiest & simplest manner.
- We have learned a lot while developing this project like object oriented operations, database operations & many more.
- The project gives good idea on developing a full-fledged application satisfying the user requirements.
- This website has a user-friendly screen that enables the user to use without any inconvenience.
- Validation checks induced have greatly reduced errors.
- Additional modules can be easily added when necessary, according to the customer's feedback.
- The application has been tested with live data and has provided a successful result. Hence the website has proved to work efficiently.

<u>REFERENCES</u>

- Google
- Website (layoutit.com / JavaTpoint.com / w3school.com)
- Java Head First.
- Java Complete Reference.
- Orally Servlets & JDBC.

APPENDIX

Backend remains on server side and has two components i.e.

1. Server side program
2. Data Base.

Data base is the most important thing in this universe as data base gives identity to a thing without data base existence of a thing is impossible while working on a project first step is to design a database.

What is data base?

Data Base is a collection of tables and table is a collection of records in a tabular form i.e. in row and columns.

Data Base can be divided into two parts:-

1. RDBMS.
2. DBMS.

JAVA

Java is a high level, third-generation programming language, like C, FORTRAN, Perl and many others. It is a platform for distributed computing – a development and run-time environment that contains built-in support for the World Wide Web.

History of Java:

Java development began at Sun Microsystems in 1991, the same year the World Wide Web was conceived. Java's creator, James Gosling did not design java for the Internet. His Objective was to create a common development environment for consumer electronic devices which was easily portable from one device to another.

This effort evolved into a language, code named Oak and later renamed Java that retains much of the syntax and power of c ++, but is simpler and more platforms independent.

Java Features:

Some of the important features of Java are as follows:

- Simplicity
- Orientation
- Platform Independence
- Security
- High Performance
- Multi-Threading
- Garbage Collection.   27

One of the most important features of Java is Platform Independence which makes it famous and suitable language for World Wide Web.

Why java is Platform Independent?

Java is Platform Independent because of Java Virtual Machine (JVM).

Java Virtual Machine (JVM)

The client application or operating system must have a java byte-code interpreter to execute byte-code instructions. The interpreter is a part of a lager program called the JVM. The JVM interprets the byte code into native code and is available on a platform that supports java.

Connectivity using JDBC:

There are four kinds of drivers available in Jdbc:-

1. Jdbc Odbc Bridge Driver.
2. Partly Java Driver.
3. Pure Java Driver.
4. Native Driver.

Client Side Interface:

 In client side interface we are using:-

 Servlet / JSP – In Internet Based Application.

J2EE Framework and Architecture:

J2EE is one of the best solutions that we have had so far for meeting the demand of today's enterprise. J2EE specifies both the infrastructure for managing our applications, and the service APIs for building our applications.

The J2EE platform is essentially a distributed application-server environment- a java environment that provides the following: -

• A set of java extension APIs to build applications. These APIs define a programming model for J2EE applications.

• A run time infrastructure for hosting and managing applications. This is the server runtime in which our applications resides.

The applications that we could develop with the above may be programs to drive web pages, or components to implement complex database transactions, or even java applets, all distributed across the network.