Software Engineering

Practical File

Submitted By:
Deepti Sharma
D3CSE-A1
1410821
(145021)

Submitted To:

Prof. Manjot Kaur Gill
CSE Department

Department of Computer Science & Engineering
Guru Nanak Dev Engineering College
Ludhiana 141006

Practical-1

Aim: Conduct feasibility study for some given problem.

Project: Web-Octave

Octave is an open-source interactive software system for numerical computations and graphics. It is particularly designed for matrix computations: solving simultaneous equations, computing eigenvectors and eigenvalues and so on. In many real-world engineering problems the data can be expressed as matrices and vectors, and boil down to these forms of solution.

This project discusses mainly about Weboctave. I devised this tool which basically is collection of direct and iterative methods to solve problems of various equations, matrices, graph plotting and ordinary linear equations.

The increasing importance of numerical methods in applied sciences have led to enhanced demand to deal with techniques of numerical analysis. The reason for this is that numerical methods can give the solution to applied problems when ordinary analytical methods fail.

Furthermore, coming back it deals with Weboctave which is web interface to use octave i.e. anyone can use this service and need not have octave installed on their systems and can use this service remotely. Also, this project is completely open source and the entire code is available to the user as and when required. There is Complete developer's Documentation as well as User manual alongwith it that helps using it a lot easier.

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.

Objectives of feasibility study are listed below:

- To analyze whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

An Overview of the Study

Feasibility study involves:

1. **Technical Feasibility:** Technical feasibility is carried out to determine whether the project has the capability, in terms of software and hardware to handle and fulfill the user requirements. The assessment is based on an outline design of system requirements in terms of Input, Processes, Output, Cross-Platform and Procedures.
   Technological issues raised during the investigation are:

   - Does the existing technology sufficient for the suggested one?
   - Can the system expand if developed?

   Languages/Technologies used:

   - HTML & CSS
   - Octave
   - PHP
   - Apache(Web Server)

2. **Economical Feasibility:** It includes quantification and identification of all the benefits expected. This assessment typically involves a cost/ benefits analysis. It is important to identify cost and benefit factors, which can be categorized as follows:

   - *Development & Operation Cost:* It's an open source project and hence free of cost for users.

3. **Market Feasibility:** The market needs analysis to view the market demand,competitive activities etc. There is a need for this project for following things:

   - Perform most of difficult Calculation work.
   - Make it work like batch mode. So, that user can give inputs together and relax.
   - Help M.Tech and Civil Engineer to analysis structure.
   - Reduce the time for analysis.
   - Provide on-line way to analysis so that individual does not have to install anything.

4. **Legal Feasibility:** It investigates if the proposed system conflicts with legal requirements like data protection acts or social media laws.

5. **Safety Feasibility:** It refers to an analysis of whether a project is capable of being implemented and operated safely with minimum adverse effects on the environment.

6. **Behavioral Feasibility:** Behavioral feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. It is a measure of how well the solution of problems or a specific alternative solution will work in the organization.

   Functions:
   The user has been provided some test functions which he can use to test various.

   Plots:
   Octave provides fltk as the default toolkit. But we can use gnuplot for more accurate plotting by setting them as default toolkit.

   Input:
   Input values are taken from user or default values defined in the file are used.

   Output:
   The iterations are performed and it returns the output with the expected precision.

7. **Operational Feasibility:** Operational feasibility is a measure of how well a project solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements analysis phase of system development.All the operations performed in the software are very quick and satisfy all the requirements.

   Hardware Requirements:

   - Operating System: Linux/Windows
   - Processor Speed: 512KHz or more
   - RAM: Minimum 256MB

   Software Requirements:

   - Software: octave gui
   - Programming Language: octave

Practical-2

Aim: Preparation of Software Requirement Specification Document.

An SRS is basically an organization's understanding (in writing) of a customer or potential client's sy stem requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

The SRS document itself states in precise and explicit language those functions and capabilities a software sy stem (i.e., a software application, an ecommerce Web site, and so on) must provide, as well as states any required constraints by which the sy stem must abide.

The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it.

It's important to note that an SRS contains functional and nonfunctional requirements only ; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be. A well-designed, well-written SRS accomplishes four major goals:

- It provides feedback to the customer. The SRS should be written in natural language (versus a formal language, explained later in this article), in an unambiguous manner that may also include charts, tables, data flow diagrams, decision tables, and so on.
- It decomposes the problem into component parts. The simple act of writing down software requirements in a well-designed format organizes information, placesborders around the problem, solidifies ideas, and helps break down the problem into its component parts in an orderly fashion.
- The SRS serves as the parent document to subsequent documents, such as the software design specification and statement of work. Therefore, the SRS must contain sufficient detail in the functional system requirements so that a design solution can be devised.
- It serves as a product validation check. The SRS also serves as the parent document for testing and validation strategies that will be applied to the requirements for verification.

What Kind of Information Should an SRS Include?

You probably will be a member of the SRS team (if not, ask to be), which means SRS development will be a collaborative effort for a particular project. In these cases, y our company will have developed SRSs before, so y ou should have examples (and, likely , the company 's SRS template) to use. But, let's assume y ou'll be starting from scratch. Several standards organizations (including the IEEE) have identified nine topics that must be addressed when designing and writing an SRS:

1. Interfaces
2. Functional Capabilities
3. Performance Levels
4. Data Structures /Elements
5. Safety
6. Reliability
7. Security /Privacy
8. Quality
9. Constraints and Limitations

Begin with an SRS Template

The first and biggest step to writing an SRS is to select an existing template that y ou can fine tune for y our organizational needs (if y ou don't have one already).

There's not a "standard specification template" for all projects in all industries because the individual requirements that populate an SRS are unique not only from company to company , but also from project to project within any one company.

The key is to select an existing template or specification to begin with, and then adapt it to meet y our needs. In recommending using existing templates, I'm not advocating simply copy ing a template from available resources and using them as y our own; instead, I'm suggesting that y ou use available templates as guides for developing y our own.

It would be almost impossible to find a specification or specification template that meets y our particular project requirements exactly . But using other templates as guides is how it's recommended in the literature on specification development. Look at what someone else has done, and modify it to fit your project requirements. (See the sidebar called "Resources for Model Templates" at the end of this article for resources that provide sample templates and related information.)

A sample of a more detailed SRS outline

1. Scope
   (a) A Webinterface to Octave, MySQL, make, Doxygen, HTML, CSS, PHP, Shell script, PNG images.
   (b) i. Giving input to the interface in form of octave commands.

      ii. storing the information in form of png images and commands in the database.

      iii. Processing the database based on the user queries and can send the output.
   (c) Document overview. A civil engineer can use this project to infer some very useful information in matter of seconds from the already existent octave functions. At the same time the information is being stored in a database that can be made available to a community of engineers for research and reference. This document comprises six sections:
      - Scope
      - Referenced documents
      - Requirements
      - Testing and Coding
2. Referenced Documents
   (a) Web-Octave technical manual.
   (b) Material database.
3. Requirements
   (a) An interface for easy operation.
   (b) A web interface for opearting commands and excessing the processed data from any sy stem over the web.
   (c) Flexibility to incorporate any changes in the standard sy ntax of the Web-Octave scripting methods.

Practical-3

Aim: Study and usage of OpenProj.

OpenProj : OpenProj is an open source, free project management software application or solution. OpenProj operates on multiple platforms including Windows, Mac, Linux and Unix. OpenProj is ideal for desktop project management and supports opening Microsoft or Primavera files. OpenProj 1.4 is an open source desktop project management application. This application is an alternative to Microsoft Project.

OpenProj provides control, tracking and management of projects. To create a new project the only field required is the Project Name on the Create New Project window and the start date of the project can be change it if you don't want to start the day that you're creating your project. You can add tasks in the Gantt diagram providing the start and end dates of each task; also you can add information to each task such the predecessors and successors tasks, resources, notes, etc. One great feature is that provides the Work Breakdown Structure (WBS) to order and control the tasks of the project for people who manages projects this is a very important tool. Another great feature is the Resource

Breakdown Structure (RBS) to define the structure of the resources, teams, providers, etc. Task Usage and Resource Usage are features to control your project and provide a good track on it. The Report tool provides information about the current status of your project.

The features of OpenProj are as follows:-

Gantt chart:- A Gantt chart is a type of bar chart, developed by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e. precedence network) relationships between activities.

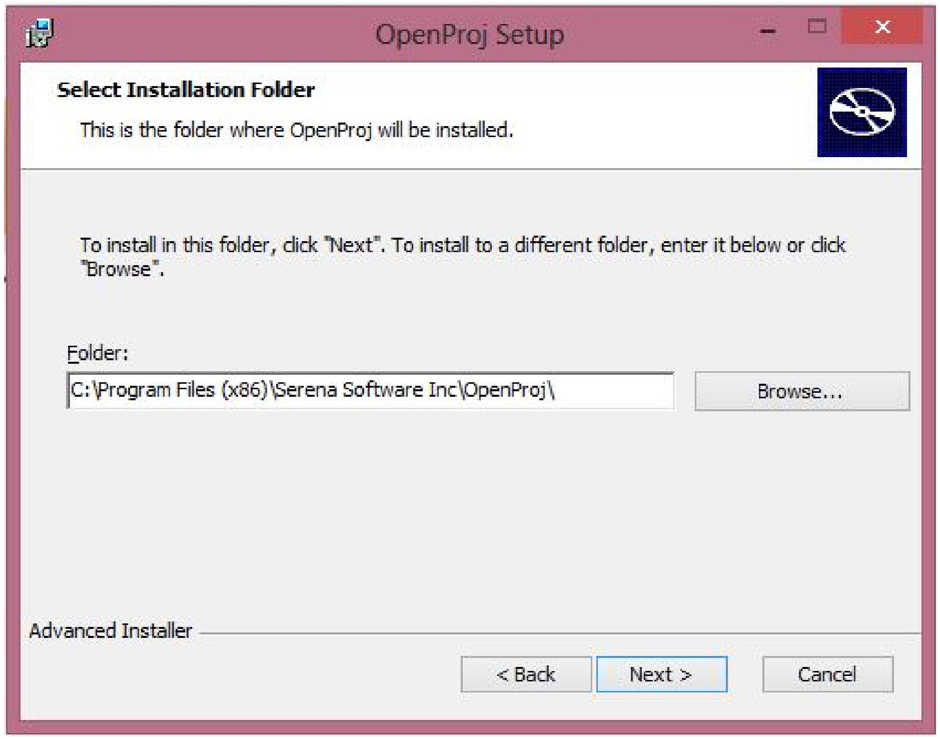
PERT graph:- The Program (or Project) Evaluation and Review Technique, commonly abbreviated PERT, is a statistical tool, used in project management, that is designed to analyze and represent the tasks involved in completing a given project. First developed by the United States Navy in the 1950s, it is commonly used in conjunction with the critical path method (CPM).

Resource Breakdown Structure (RBS) chart:- In project management, the resource breakdown structure (RBS) is a hierarchical list of resources related by function and resource type that is used to facilitate planning and controlling of project work. In some cases, a geographic division may be preferred. Each descending (lower) level represents an increasingly detailed description of the resource until small enough to be used in conjunction with the work breakdown structure (WBS) to allow the work to be planned, monitored and controlled.

Work Breakdown Structure (WBS) chart: - A work breakdown structure (WBS), in project management and systems engineering, is a deliverable oriented decomposition of a project into smaller components.A work breakdown structure element may be a product, data, service, or any combination thereof.

Steps of installation of OpenProj:

Figure 3.1: Download by clicking here



Select Installation Folder

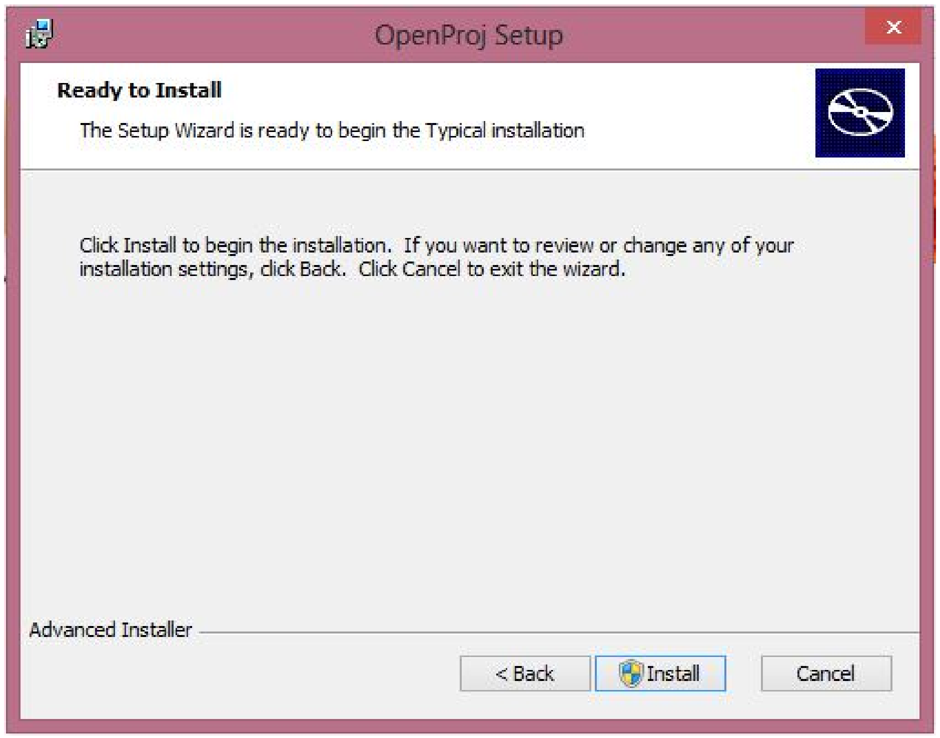
This is the folder where OpenProj will be installed.

To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".

Folder:
C:\Program Files (x86)\Serena Software Inc\OpenProj\

Advanced Installer

Figure 3.2: Click Next


OpenProj Setup

Ready to Install

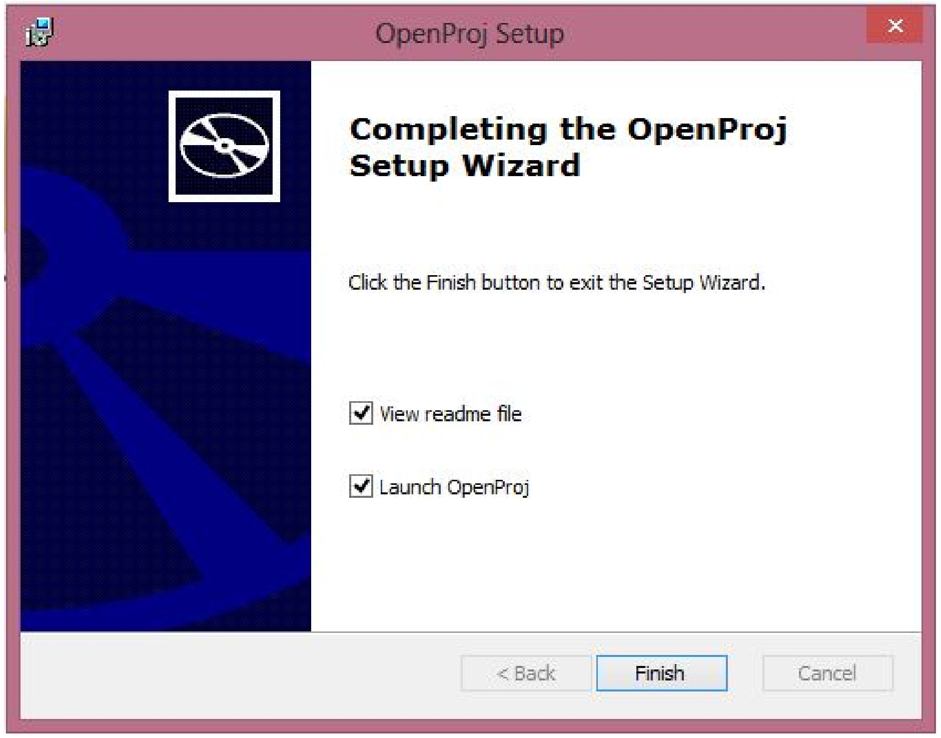
The Setup Wizard is ready to begin the Typical installation

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Advanced Installer

< Back | Install | Cancel

Figure 3.3: Click Install


OpenProj Setup

Completing the OpenProj Setup Wizard

Click the Finish button to exit the Setup Wizard.

[✓] View readme file

[✓] Launch OpenProj

< Back | Finish | Cancel

Figure 3.4: Click Finish

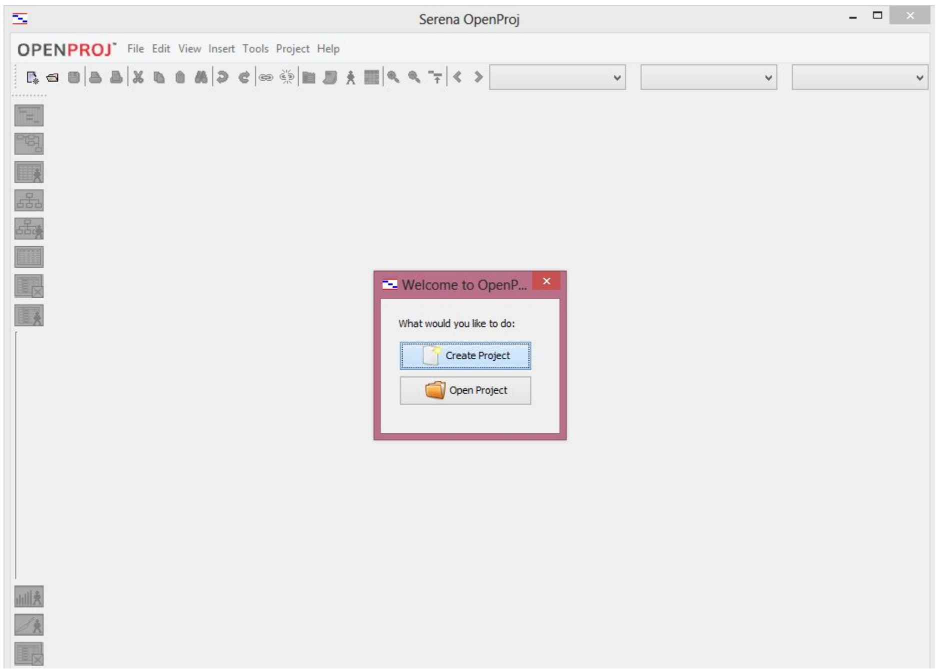
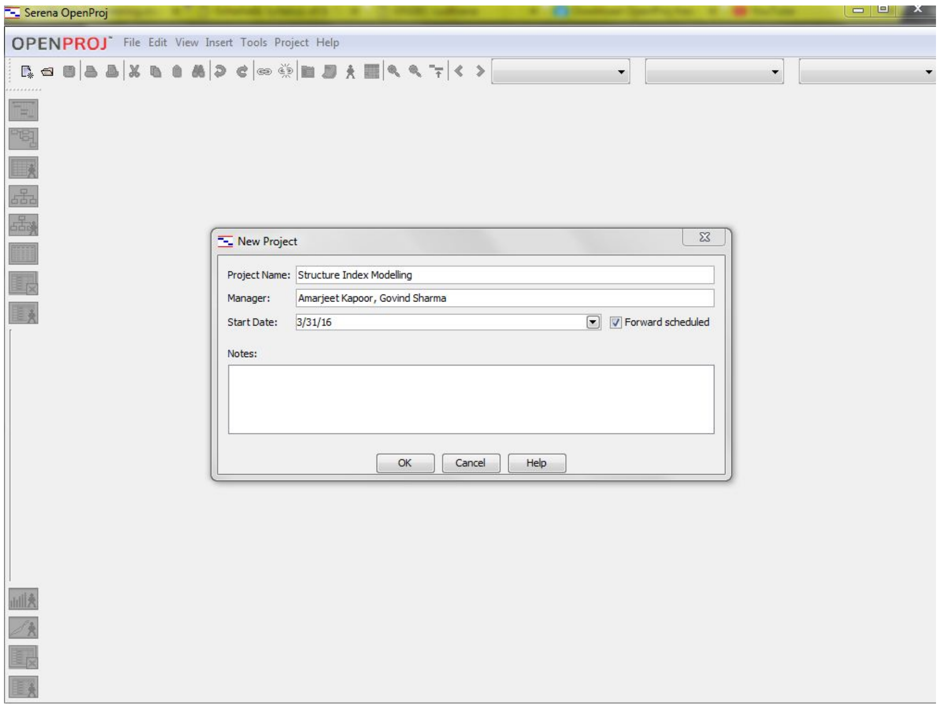


Figure 3.5: Create Project Name: Web-Octave



Project Name: Structure Index Modelling
Manager: Amarjeet Kapoor, Govind Sharma
Start Date: 3/31/16 [✓ Forward scheduled]
Notes:

Figure 3.6: Click OK

Practical-4

Aim: Usage of OpenProj or similar software to draft plan and to track the progress of a project.

The OpenProj free project management tool has now been superceded by ProjectLibre. The new program is also free for any use and is based on OpenProj. Project managers can use OpenProj, a free task tracking application, when creating effective plans
The best way to understand how a project plan may be created using OpenProj is to study a realistic example such as the one below.

From the graphical presentation of the critical path, to resource balancing and task elaboration, OpenProj gives the project manager a set of functions that help to monitor project performance.

To create projects in OpenProj Software:

1. Having downloaded and installed the OpenProj software; start OpenProj from the start menu of Windows or via the icon on your desktop. After the license agreement, OpenProj starts with a choice of creating a new project or opening an existing one. Choose Create Project.

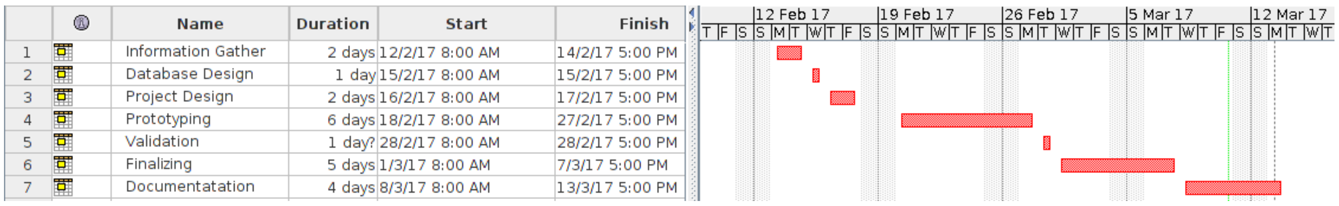
2. Enter a name for the project, the project manager's name and a start date (which can also be modified later if required). Provide a description in the Notes section as you see fit.

3. The application window will be displayed consisting of following main components:-

   - Indicator Column:- In this column, symbols are displayed to show the status of the corresponding tasks (see right).
   - Task Number Column:- Task numbers are assigned by OpenProj automatically when data is entered. The user does not enter information into this box.
   - Time Scale:- In many views, OpenProj shows a timeline whose scale can be changed via the + and - magnifying glass icon on the menu above.
   - Information View Type:- The area along the left side of the window lists a number of buttons representing the available views. Gantt, Network, Resources, WBS (work breakdown structure), RBS (resource breakdown structure), Reports, Task Usage and Resource Usage.

The lower view buttons open a split-screen where resource mapping tables and resource utilization as a graphic can be displayed. You can toggle views on and off by clicking the same button repeatedly.

- **Gantt Chart** A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity. This allows you to see at a glance: What the various activities are? When each activity begins and ends?

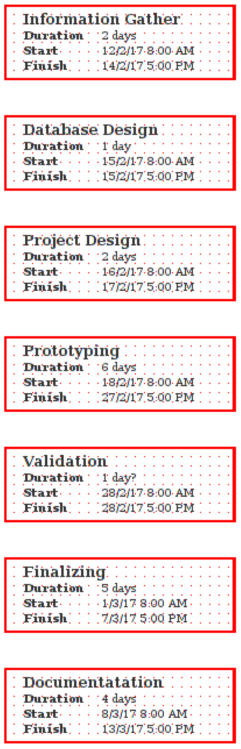


	Name	Duration	Start	Finish
1	Information Gather	2 days	12/2/17 8:00 AM	14/2/17 5:00 PM
2	Database Design	1 day	15/2/17 8:00 AM	15/2/17 5:00 PM
3	Project Design	2 days	16/2/17 8:00 AM	17/2/17 5:00 PM
4	Prototyping	6 days	18/2/17 8:00 AM	27/2/17 5:00 PM
5	Validation	1 day?	28/2/17 8:00 AM	28/2/17 5:00 PM
6	Finalizing	5 days	1/3/17 8:00 AM	7/3/17 5:00 PM
7	Documentatation	4 days	8/3/17 8:00 AM	13/3/17 5:00 PM

12 Feb 17
19 Feb 17
26 Feb 17
5 Mar 17
12 Mar 17
T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T

Figure 4.1: Gantt chart

- **Network Chart** A network diagram is essentially a flow chart that includes all of the project elements and how they relate to one another. It shows parallel activities and the links between each activity. Network logic is the collection of activity dependencies that make up a network diagram for a particular project. In other words, certain tasks are dependent on one another to complete the project. This creates a logical stream of events that will lead to completion of the project.



Information Gather
Duration 2 days
Start 12/2/17 8:00 AM
Finish 14/2/17 5:00 PM

Database Design
Duration 1 day
Start 15/2/17 8:00 AM
Finish 15/2/17 5:00 PM

Project Design
Duration 2 days
Start 16/2/17 8:00 AM
Finish 17/2/17 5:00 PM

Prototyping
Duration 6 days
Start 18/2/17 8:00 AM
Finish 27/2/17 5:00 PM

Validation
Duration 1 day?
Start 28/2/17 8:00 AM
Finish 28/2/17 5:00 PM

Finalizing
Duration 5 days
Start 1/3/17 8:00 AM
Finish 7/3/17 5:00 PM

Documentatation
Duration 4 days
Start 8/3/17 8:00 AM
Finish 13/3/17 5:00 PM

Figure 4.2: Network chart

- **Work Breakdown Structure(WBS)** A work breakdown structure (WBS),in project management and system engineering, is a deliverable-oriented decomposition of a project into smaller components.A work breakdown structure is a key project deliverable that organizes the team's work into manageable sections. The Project Management Body of Knowledge defines the work breakdown structure as a "A hierarchical decomposition of the total scope of work to be carried out by the project team to accomplish the project objectives and create the required deliverables." A work breakdown structure element may be a product,data,service or any combination thereof. A WBS also provides the necessary framework for detailed cost estimating and control along with providing guidance for schedule development and control.

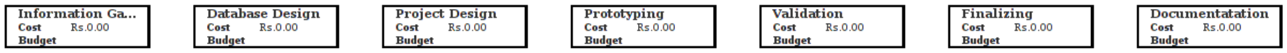
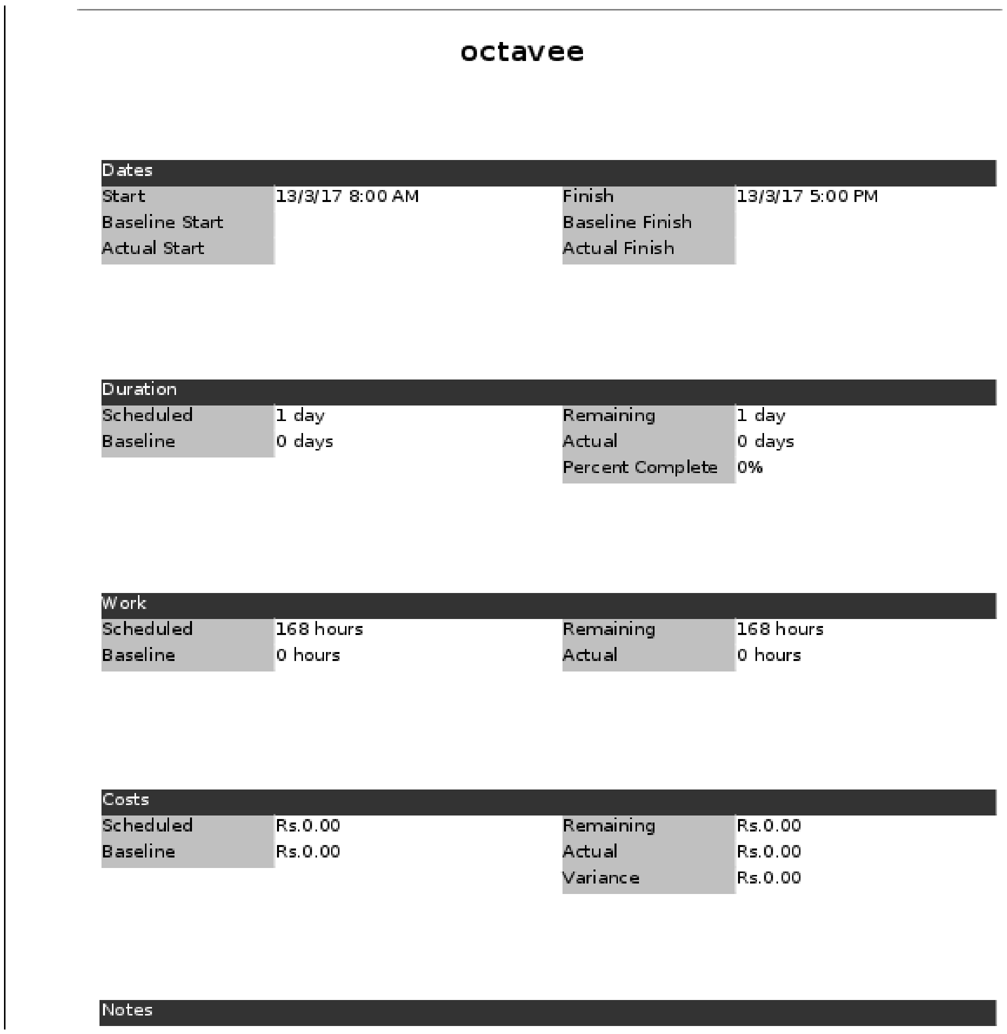


Figure 4.3: Work Breakdown Structure



octavee

Dates			
Start	13/3/17 8:00 AM	Finish	13/3/17 5:00 PM
Baseline Start		Baseline Finish	
Actual Start		Actual Finish	

Duration			
Scheduled	1 day	Remaining	1 day
Baseline	0 days	Actual	0 days
		Percent Complete	0%

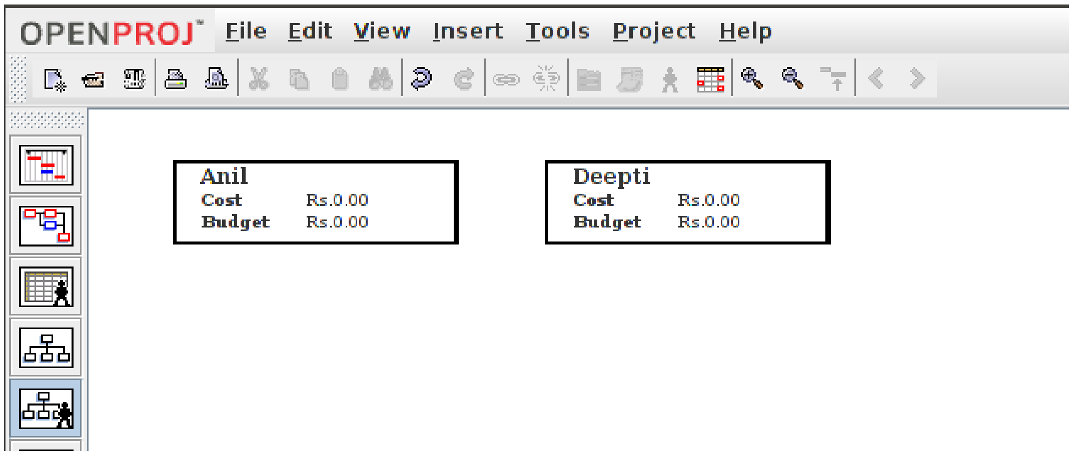
Work			
Scheduled	168 hours	Remaining	168 hours
Baseline	0 hours	Actual	0 hours

Costs			
Scheduled	Rs.0.00	Remaining	Rs.0.00
Baseline	Rs.0.00	Actual	Rs.0.00
		Variance	Rs.0.00

Notes

Figure 4.4: Report Chart

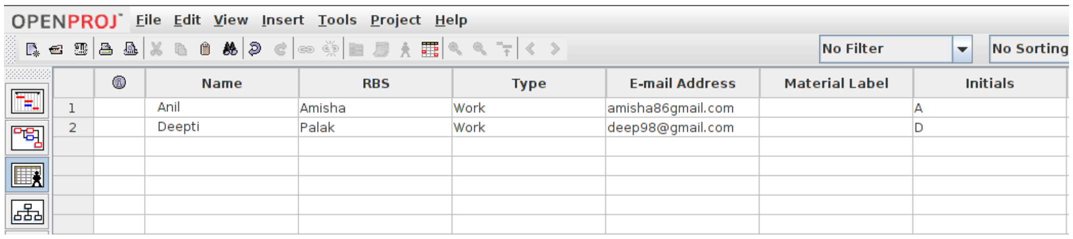
- **Resource Breakdown Structure (RBS) chart**:- In project management, the resource breakdown structure (RBS) is a hierarchical list of resources related by function and resource type that is used to facilitate planning and controlling of project work. In some cases, a geographic division may be preferred. Each descending (lower) level represents an increasingly detailed description of the resource until small enough to be used in conjunction with the work breakdown structure (WBS) to allow the work to be planned, monitored and controlled.



Anil
Cost Rs.0.00
Budget Rs.0.00

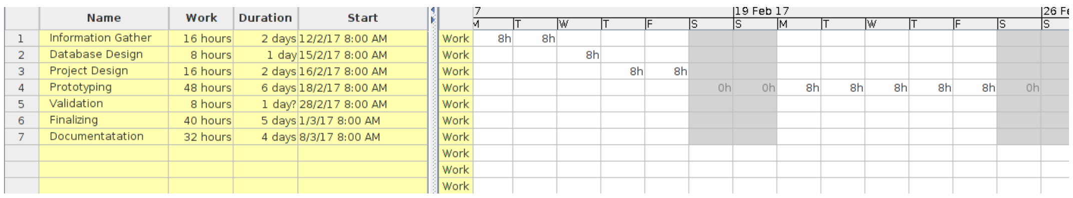
Deepti
Cost Rs.0.00
Budget Rs.0.00

Figure 4.5: RBS



		Name	RBS	Type	E-mail Address	Material Label	Initials
1		Anil	Amisha	Work	amisha86gmail.com		A
2		Deepti	Palak	Work	deep98@gmail.com		D

Figure 4.6: Resources



	Name	Work	Duration	Start		M	T	W	T	F	S	S	M	T	W	T	F	S	S
						7						19 Feb 17							26 Fe
1	Information Gather	16 hours	2 days	12/2/17 8:00 AM	Work	8h	8h												
2	Database Design	8 hours	1 day	15/2/17 8:00 AM	Work			8h											
3	Project Design	16 hours	2 days	16/2/17 8:00 AM	Work				8h	8h									
4	Prototyping	48 hours	6 days	18/2/17 8:00 AM	Work						0h	0h	8h	8h	8h	8h	8h	0h	
5	Validation	8 hours	1 day?	28/2/17 8:00 AM	Work														
6	Finalizing	40 hours	5 days	1/3/17 8:00 AM	Work														
7	Documentatation	32 hours	4 days	8/3/17 8:00 AM	Work														
					Work														
					Work														
					Work														

Figure 4.7: Task Usage

- **Histogram** A Histogram is a chart type that shows data distribution and it can help in estimating the distribution probability of a variable.

- **Resource Usage** On servers there is a limited amount of resources available for use at any one given time, for all users on that server. The main resources that we monitor for high usage would be CPU usage, and disk usage.

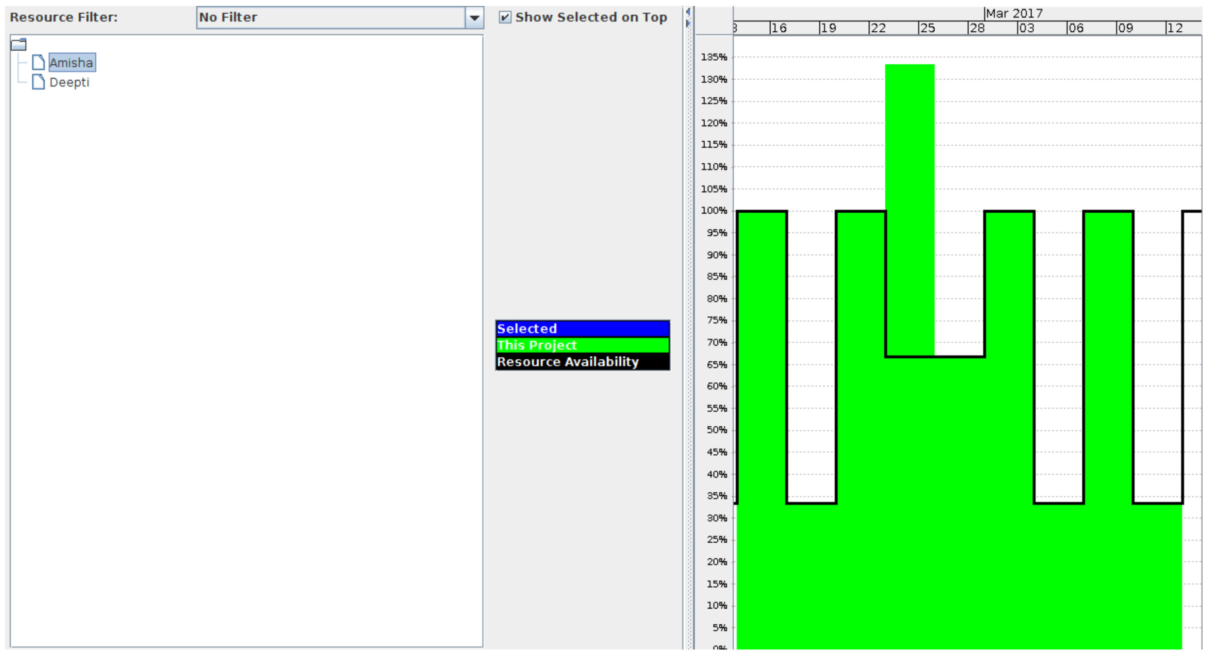
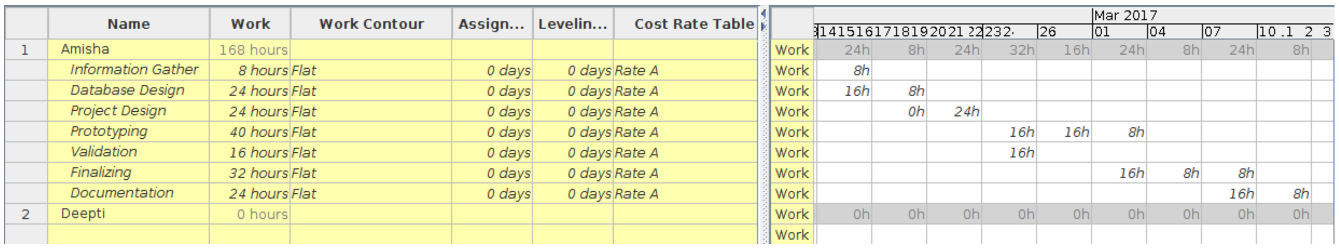


Figure 4.8: Histogram



	Name	Work	Work Contour	Assign...	Levelin...	Cost Rate Table
1	Amisha	168 hours				
	Information Gather	*8 hours*	*Flat*	*0 days*	*0 days*	*Rate A*
	Database Design	*24 hours*	*Flat*	*0 days*	*0 days*	*Rate A*
	Project Design	*24 hours*	*Flat*	*0 days*	*0 days*	*Rate A*
	Prototyping	*40 hours*	*Flat*	*0 days*	*0 days*	*Rate A*
	Validation	*16 hours*	*Flat*	*0 days*	*0 days*	*Rate A*
	Finalizing	*32 hours*	*Flat*	*0 days*	*0 days*	*Rate A*
	Documentation	*24 hours*	*Flat*	*0 days*	*0 days*	*Rate A*
2	Deepti	0 hours				

						Mar 2017			
	31415161718192021 22			232·	26	01	04	07	10 .1 2 3
Work	24h	8h	24h	32h	16h	24h	8h	24h	8h
Work	*8h*								
Work	*16h*	*8h*							
Work		*0h*	*24h*						
Work				*16h*	*16h*	*8h*			
Work				*16h*					
Work						*16h*	*8h*	*8h*	
Work								*16h*	*8h*
Work	0h	0h	0h	0h	0h	0h	0h	0h	0h
Work									

Figure 4.9: Resource Usage

Practical-5

Aim: Preparation of Software Configuration Management and Risk Management related documents

The purpose of Software Configuration Management is to establish and maintain the integrity of the products of the software project throughout the project's software life cycle. Software Configuration Management involves identifying configuration items for the software project, controlling these configuration items and changes to them, and recording and reporting status and change activity for these configuration items.

Configuration management (CM) refers to a discipline for evaluating, coordinating, approving or disapproving, and implementing changes in artefacts that are used to construct and maintain software systems. An artifact may be a piece of hardware or software or documentation. CM enables the management of artifact from the initial concept through design, implementation, testing, base lining, building, release, and maintenance.

CM is intended to eliminate the confusion and error brought about by the existence of different versions of artifacts. Artifact change is a fact of life: plan for it or plan to be overwhelmed by it. Changes are made to correct errors, provide enhancements, or simply reflect the evolutionary refinement of product definition. CM is about keeping the inevitable change under control. Without a well-enforced CM process, different team members (possibly at different sites) can use different versions of artifacts unintentionally. Successful CM requires a well-defined and institutionalized set of policies and standards that clearly define

- The set of artifacts (configuration items) under the jurisdiction of CM (Models that will be used)
- How artifacts are named (division of models into admin, student and company module)
- How artifacts enter and leave the controlled set (various logins and logouts)
- How an artifact under CM is allowed to change (admin has the sole authority)
- How different versions of an artifact under CM are made available and under what conditions each one can be used (whether to register , participate or work as admin)
- How CM tools are used to enable and enforce CM
- In single-system CM, each version of the system has a configuration associated with it that defines the versions of the configuration items that went into that system's production. In product line CM, a configuration must be maintained for each version of each product.

- In single-system CM, each product with all of its versions maybe managed separately. In product line CM, such management is untenable, because the core assets are used across all products. Hence, the entire product line is usually managed with a single, unified CM process.

- Product line CM must control the configuration of the core asset base and its use by all product developers. It must account for the fact that core assets are usually produced by one team and used in parallel by several others. Single-system CM has no such burden: the component developers and product developers are the same people.

- Only the most capable CM tools can be used in a product line effort. Many tools that are adequate for single-system CM are simply not sufficiently robust to handle the demands of product line CM. (See the "Tool Support" practice area for a more complete discussion of tools.)

The mission of product line CM is allowing the rapid reconstruction of any product version that may have been built using various versions of the core assets and development/operating environment plus various versions of product-specific artifacts. One product line manager summed up the problem.

Risk Management Plan:

Purpose of the Risk management plan

A risk is an event or condition that, if it occurs, could have a positive or negative effect on a projects objectives. Risk Management is the process of identifying, assessing, responding to, monitoring, and reporting risks. This Risk Management Plan defines how risks associated with the <Project Name> project will be identified, analyzed, and managed. It outlines how risk management activities will be performed, recorded, and monitored throughout thelifecycle of the project and provides templates and practices for recording and prioritizing risks.

The Risk Management Plan is created by the project manager in the Planning Phase of the CDC Unified Process and is monitored and updated throughout the project. The intended audience of this document is the project team, project sponsor and management.

Risk management Procedure:

The project manager working with the project team and project sponsors will ensure that risks are actively identified, analyzed, and managed throughout the life of the project. Risks will be identified as early as possible in the project so as to minimize their impact. The steps foraccomplishing this are outlined in the following sections. The <project manager or other designee> will serve as the Risk Manager for this project.

Risk Identification

Risk identification will involve the project team, appropriate stakeholders, and will include an evaluation of environmental factors, organizational culture and the project management plan including the project scope. Careful attention will be given to the project deliverables, assumptions, constraints, WBS, cost/effort estimates, resource plan, and other key project documents.

A Risk Management Log will be generated and updated as needed and will be stored electronically in the project library located at <file location>. In our project Function Organiser also we looked at various aspects of risks that could occur. We looked at the environmental factors and our organizational culture pertained to the student body of the college Project scope included serving and notifications of various events for which the students registered, so that it reduces the manual work and the system of organizing events could come easy. Assumptions of student body and companies that would participate and cost of project was also designed.

Risk Analysis

All risks identified will be assessed to identify the range of possible project outcomes. Qualification will be used to determine which risks are the top risks to pursue and respond to and which risks can be ignored Qualitative Risk Analysis. The probability and impact of occurrence for each identified risk will be assessed by the project manager, with input from the project team using the following approach:

Probability

- **High** Greater than 70 probability of occurrence
- **Medium** Between 30 and 70 probability of occurrence
- **Low** Below 30 probability of occurrence

Impact

- **High** Risk that has the potential to greatly impact project cost, project schedule or performance (Security for Functions defination)
- **Medium** Risk that has the potential to slightly impact project cost, project schedule or performance (Students not aware of the functionality of the project Function Organiser.)
- **Low** Risk that has relatively little impact on cost, schedule or Performance (connection or connectivity issues as in terms of the project)

Risks that fall within the RED and YELLOW zones will have risk response planning which may include both a risk mitigation and a risk contingency plan.

Quantitative Risk Analysis

Analysis of risk events that have been prioritized using the qualitative risk analysis process and their effect on project activities will be estimated, a numerical rating applied to each risk based on this analysis, and then documented in this section of the risk management plan.

Practical-6

Aim: Getting familiar with function oriented design tools for making dataflow diagrams.

DFD

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).
A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top-down approach to Systems Design. This context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.

Data flow diagrams are one of the three essential perspectives of the structured-systems analysis and design method SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a system's evolution. With a data flow diagram, users are able to visualize how the system will operate, what the system will accomplish, and how the system will be implemented. The old system's dataflow diagrams can be drawn up and compared with the new system's data flow diagrams to draw comparisons to implement a more efficient sy stem.

Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram model.
In the course of developing a set of levelled data flow diagrams the analyst/designer is forced to address how the system may be decomposed into component sub-systems, and to identify the transaction data in the data model.

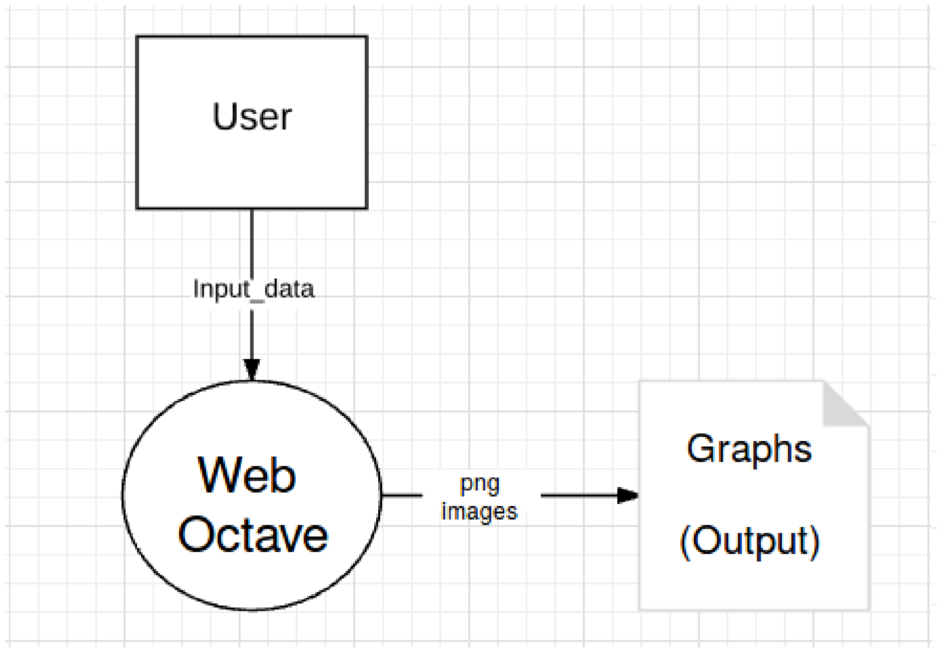


Figure 6.1: Level 0

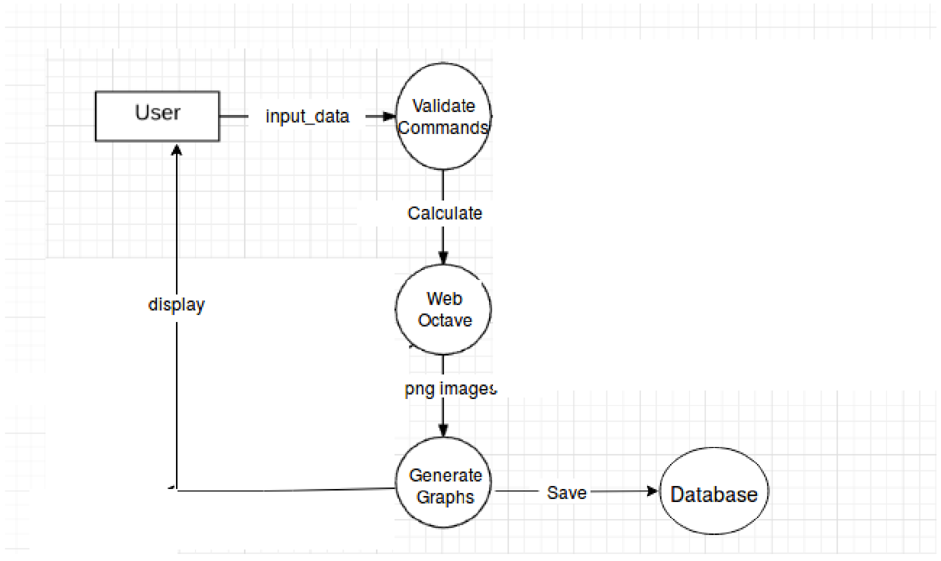


Figure 6.2: Level 1

Data Dictionary

input_data: input, calculate, input_data
Calculate: Octave Commands
image: upload image, crop image
output: png Graphs
png: Portable Network Graphic format
Result : Saved as Graphs for user and in Database

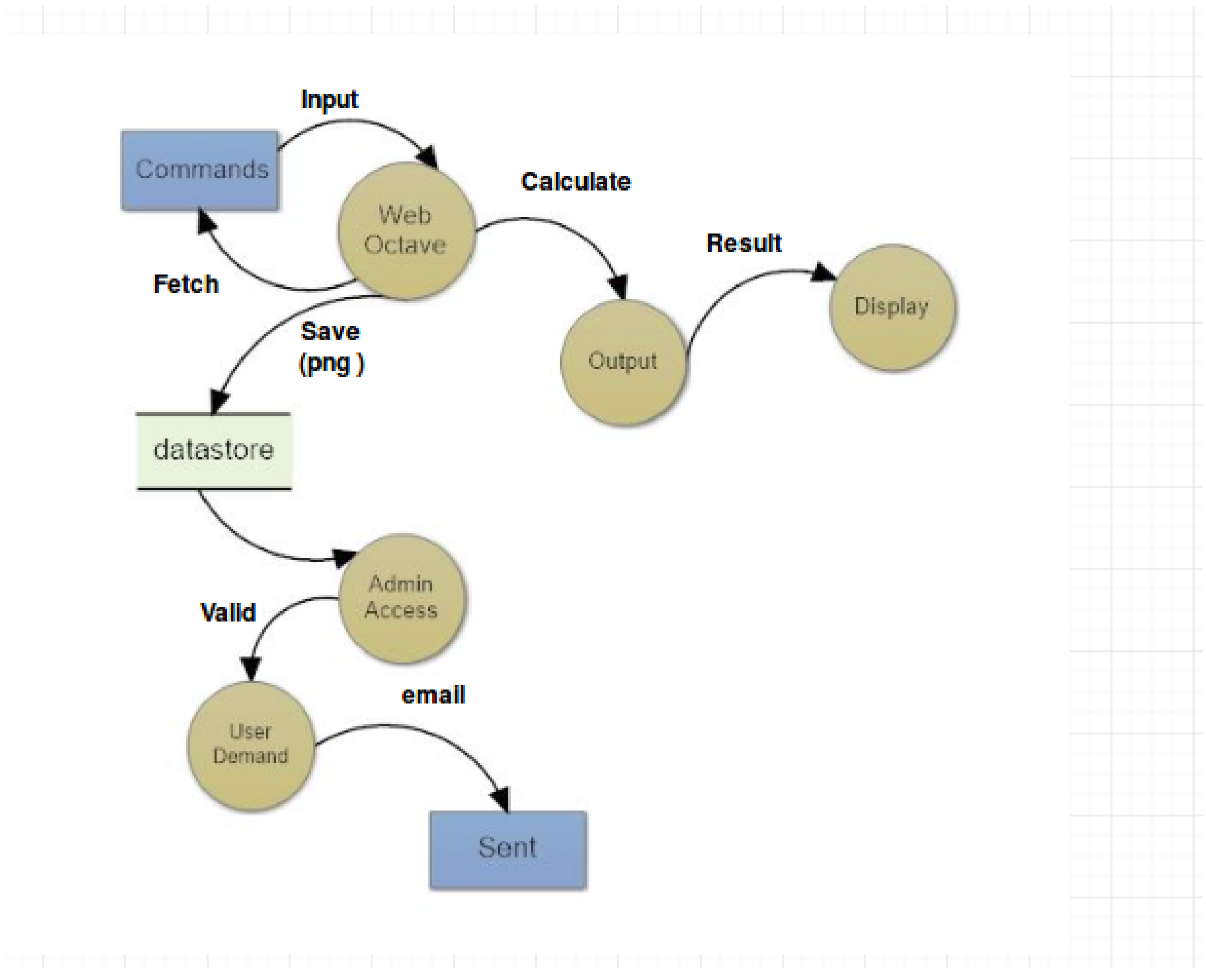


Figure 6.3: Level 2

Practical-7

Aim: Generate the test cases using boundary value analysis for some problems.

Analysis

In computer programming, testing is a software method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. Ideally, each test case is independent from the others. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

Testing

Testing a program consists of providing the program with a set of test inputs (or test cases) and observing if the program behaves as expected. If the program fails to behave as expected, then the conditions under which failure occurs are noted for later debugging and correction.
This software had been taken through rigorous test to fully found potential causes of error and system failure and full focus have been given to cover all possible exceptions that can occur and cause failure of the software.
As this software is based on intensive background process it have been taken care that if correct input and email address are given then processing of user job can even continue or a least automatically restart even after server shuts down or even crash.

Overview of Octave versions	
Date	Publication Title
September 1999	Octave framework 1.0
September 2001	Octave framework 2.0
December 2001	Octave criteria 2.0
September 2003	Octave-S v0.9
March 2005	Octave-S v1.0
June 2007	Octave 3.x
March 2016	Octave 4.0.1
July 2016	Octave 4.0.3

Table 7.1: Octave Release

Test cases for main page(index.html)			
Input	Desired Output	Actual Output	Status
Inputs range exceeds	Alert user,Don't proceed	Alert user,Don't proceed.	Pass
Incomplete Command	Alert user about range. Don't proceed	Alert user about range. Don't proceed	Pass
PNG selected: No	Error	Don't proceed	Pass
Session: Yes	Show email field after Submit	Show email field after Submit	Pass
Help pressed	Show Detailed user help	Show Detailed user help	Pass

Table 7.2: Computational Analysis

Test cases for possible source of problems			
Input	Desired Output	Actual Output	Status
URL refreshed	Send to homepage	Send to homepage	Pass
server stops or rebooted	Start processing interrupted requests	Start processing interrupted requests	Pass

Table 7.3: Test case (general)

Practical-8

Aim: Getting familiar and hands on practice with testing tools like win runner, selenium etc.

Software Testing Tools in the market. Selection of tools is totally based on the project requirements & commercial (Proprietary/Commercial tools) or free tools (Open Source Tools) you are interested. Off Course, free Testing Tools may have some limitation in the features list of the product, so itâĂŹs totally based on what are you looking for & is that your requirement fulfill in free version or go for paid Software Testing Tools. The following tools can be used for automation testing:

1. **HP Quick Test Professional:** HP QuickTest Professional was renamed to HPE Unified Functional Testing. HPE UFT offers testing automation for functional and regression testing for the software applications.Visual Basic Scripting Edition scripting language is used by this tool to register the test processes and operates the various objects and controls in testing the applications. QTP offers various features like:
   - Integration with Mercury Business Process Testing and Mercury Quality Center
   - Unique Smart Object Recognition
   - Error handling mechanism
   - Creation of parameters for objects, checkpoints, and data-driven tables
   - Automated documentation
2. **Selenium:** Selenium is a testing framework to perform web application testing across various browsers and platforms like Windows, Mac, and Linux. Selenium helps the testers to write tests in various programming languages like Java, PHP, C#, Python, Groovy, Ruby, and Perl. It offers record and playback features to write tests without learning Selenium IDE. Selenium proudly supports some of the largest, yet well-known browser vendors who make sure they have Selenium as a native part of their browser. Selenium is undoubtedly the base for most of the other software testing tools in general.
3. **TestComplete:** TestComplete is a functional testing platform that offers various solutions to automate testing for desktop, web, and mobile applications by SmartBear Software. TestComplete offers the following features:
   - GUI testing
   - Scripting Language Support: JavaScript, Python, VBScript, JScript, DelphiScript, C++Script & C#Script

4. **WinRunner:** WinRunner is a program that is responsible for the automated testing of software. It is designed to function with Windows applications, and it was designed by Mercury Interactive Corporation, an organization that is based in California. WinRunner will allow you to make comparisons between various outcomes. A series of wizards will be provided to the user, and these wizards can create tests in an automated manner. Another impressive aspect of Winrunner is the ability to record various interactions, and transform them into scripts. WinRunner is designed for testing GUIs, or Graphic User Interfaces. When the user make an interaction with the GUI, this interaction can be recorded. Recording the interactions will allow you to determine various bugs that need to be fixed. The goal of WinRunner is to make sure business processes are properly carried out. WinRunner uses TSL, or Test Script Language. The input of the user can be customized at various levels. WinRunner will test the computer program in a way that is very similar to normal user interactions. This is important, because it ensures a high level of accuracy and realism. Even if an engineer is not physically present, the Recover manager will troubleshoot any problems that may occur, and this will allow the tests to be completed without errors

5. **WATIR:** Watir is an open source testing tool made up of Ruby libraries to automate web application testing. It is pronounced as "water." Watir offers following features:

   - Tests any language-based web application
   - Cross-browser testing
   - Compatible with business-driven development tools like RSpec, Cucumber, and Test/Unit
   - Tests web page's buttons, forms, links, and their responses

   Tools Implementation - process

   (a) Analyse the problem carefully to identify strengths, weaknesses and opportunities
   (b) The Constraints such as budgets, time and other requirements are noted.
   (c) Evaluating the options and Shortlisting the ones that are meets the requirement
   (d) Developing the Proof of Concept which captures the pros and cons
   (e) Create a Pilot Project using the selected tool within a specified team
   (f) Rolling out the tool phase wise across the organization

Practical-9

Aim: Perform debugging using Big Tracking Tools like BUGZILLA or BUG BIT

SOFTWARE BUG
A software bug is an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways. Most bugs arise from mistakes and errors made in either a program's source code or its design, or in components and operating systems used by such programs. A few are caused by compilers producing incorrect code. A program that contains a large number of bugs, and/or bugs that seriously interfere with its functionality, is said to be buggy (defective). Bugs trigger errors that may have ripple effects. Bugs may have subtle effects or cause the program to crash or freeze the computer. Others qualify as security bugs and might, for example, enable a malicious user to bypass access controls in order to obtain unauthorized privileges.

PREVENTION OF SOFTWARE BUG
The software industry has put much effort into reducing bug counts. These include:

- **Typographical errors**
  Bugs usually appear when the programmer makes a logic error. Various innovations in programming style and defensive programming are designed to make these bugs less likely, or easier to spot. Some typos, especially of symbols or logical/mathematical operators, allow the program to operate incorrectly, while others such as a missing symbol or misspelled name may prevent the program from operating. Compiled languages can reveal some typos when the source code is compiled.

- **Development methodologies**
  Several schemes assist managing programmer activity so that fewer bugs are produced. Software engineering (which addresses software design issues as well) applies many techniques to prevent defects. For example, formal program specifications state the exact behaviour of programs so that design bugs may be eliminated. Unfortunately, formal specifications are impractical for anything but the shortest programs, because of problems of combinatorial explosion and indeterminacy. Unit testing involves writing a test for every function (unit) that a program is to perform. In test-driven development unit tests are written before the code and the code is not considered complete until all tests complete successfully. Agile software development involves frequent software releases with relatively small changes. Defects are revealed by user feedback. Open source development allows

anyone to examine source code. A school of thought popularized by Eric S. Raymond as Linus's Law says that popular open-source software has more chance of having few or no bugs than other software, because "given enough eyeballs, all bugs are shallow". This assertion has been disputed, however: computer security specialist Elias Levy wrote that "it is easy to hide vulnerabilities in complex, little understood and undocumented source code," because, "even if people are reviewing the code, that doesn't mean they're qualified to do so." An example of this actually happening, accidentally, was the 2008 OpenSSL vulnerability in Debian.

- **Programming language support**
  Programming languages include features to help prevent bugs, such as static type systems, restricted namespaces and modular programming. For example, when a programmer writes (pseudocode) LET REAL_VALUE PI = "THREE AND A BIT", although this may be syntactically correct, the code fails a type check. Compiled languages catch this without having to run the program. Interpreted languages catch such errors at runtime. Some languages deliberate exclude features that easily lead to bugs, at the expense of slower performance: the general principle being that, it is almost always better to write simpler, slower code than inscrutable code that runs slightly faster, especially considering that maintenance cost is substantial. For example, the Java programming language does not support pointer arithmetic; implementations of some languages such as Pascal and scripting languages often have runtime bounds checking of arrays, at least in a debugging build.

- **Code analysis**
  Tools for code analysis help developers by inspecting the program text beyond the compiler's capabilities to spot potential problems. Although in general the problem of finding all programming errors given a specification is not solvable (see halting problem), these tools exploit the fact that human programmers tend to make certain kinds of simple mistakes often when writing software.

- **Instrumentation**
  Tools to monitor the performance of the software as it is running, either specifically to find problems such as bottlenecks or to give assurance as to correct working, may be embedded in the code explicitly (perhaps as simple as a statement saying PRINT "I AM HERE"), or provided as tools. It is often a surprise to find where most of the time is taken by a piece of code, and this removal of assumptions might cause the code to be rewritten.

DEBUGGING

Finding and fixing bugs, or debugging, is a major part of computer programming. the most difficult part of debugging is finding the bug. Once it is found, correcting it is usually relatively easy. Programs known as debuggers help programmers locate bugs by executing code line by line, watching variable values, and other features to observe program behaviour. Without a debugger, code may be added so that messages or values may be written to a console or to a window or log file to trace program execution or show values. However, even with the aid of a debugger, locating bugs is something of an art. It is not uncommon for a bug in one section of a program to cause failures in a completely different section, [citation needed] thus making it especially difficult to track (for example, an error in a graphics rendering routine causing a file I/O routine to fail), in an apparently unrelated part of the system. Sometimes, a bug is not an isolated flaw, but represents an error of thinking or planning on the part of the programmer. Such logic errors require a section of the program to be overhauled or rewritten. As a part of code review, stepping through the code and imagining or transcribing the execution process may often find errors without ever reproducing the bug as such. More typically, the first step in locating a bug is to reproduce it reliably. Once the bug is reproducible, the programmer may use a debugger or other tool while reproducing the error to find the point at which the program went astray. Some bugs are revealed by inputs that may be difficult for the programmer to re-create. One cause of the Therac-25 radiation machine deaths was a bug (specifically, a race condition) that occurred only when the machine operator very rapidly entered a treatment plan; it took days of practice to become able to do this, so the bug did not manifest in testing or when the manufacturer attempted to duplicate it. Other bugs may disappear when the program is run with a debugger; these heisenbugs (humorously named after the Heisenberg uncertainty principle).

BUGZILLA

Bugzilla is a web-based general-purpose bugtracker and testing tool originally developed and used by the Mozilla project, and licensed under the Mozilla Public License. Released as open-source software by Netscape Communications in 1998, it has been adopted by a variety of organizations for use as a bug tracking system for both free and open-source software and proprietary projects and products. Bugzilla is used, among others, by the Mozilla Foundation, WebKit, Linux kernel, FreeBSD, GNOME, KDE, Apache, Red Hat, Eclipse and LibreOffice. It is also self-hosting.

REQUIREMENTS OF BUGZILLA:

Bugzilla's system requirements include:

- A compatible database management system
- A suitable release of Perl 5
- An assortment of Perl modules
- A compatible web server
- A suitable mail transfer agent, or any SMTP server

Currently supported database systems are MySQL, PostgreSQL, Oracle, and SQLite. Bugzilla is usually installed on Linux using the Apache HTTP Server, but any web server that supports CGI such as Lighttpd, Hiawatha, Cherokee can be used. Bugzilla's installation process is command line driven and runs through a series of stages where system requirements and software capabilities are checked.