

**ORDNANCE CLOTHING FACTORY**

**AVADI ,CHENNAI**

**INTERNSHIP TRAINING**

**(11-06-2018 to 15-06-2018)**

**By**

**Kailasa eswaran .I**

**Meenakshi .S**

**Parthiban. S**

**Saranya.P**

**Srinidhi .V**

**Swetha .P**

**OUR VIEW ON ORDNANCE CLOTHING**

**FACTORY:**

We the students of rajalakshmi institute of technology attend the internship training conducted by ordnance clothing factory avadi from 11 to 15 june,2018. The ordnance clothing factory which comes under the ministry of defence manufacturing cloths under camouflage pattern for Indian army/navy. The followed strict rules and regulations and with well secured environment. They maintain clean and green environment inside the ordnance clothing factory campus .There were totally 1643 employees from various department and 200 apprentices working under the feature of ordnance clothing factory. We got quality training under the department of information technology in which comes under the human resource department .The staffs and employees trained us well and thought us new and advanced software and hardware implementation for creation of our project. Thus we got enough knowledge to develop our own project.

**CONCLUSION:**

Thus these five days went on to become useful for career,we gained enough knowledge and wisdom under our own sector. We there by thanking

SHRI . K . SATHYANARANAYA , GENERAL MANAGER

SHRI . V . VENUGOPAL , JUNIOR WORKS MANAGER , HUMAN RESOURCE DEVELOPMENT

SHRI . JAIMOHAN , OFFICE SUPERINTENDENT , HUMAN RESOURCE DEVELOPMENT

SHRI . B. PRABHU , JUNIOR WORKS MANAGER , INFORMATION TECHNOLOGY SECTOR

SHRI . RAVI , JUNIOR WORKS MANAGER , INFORMATION TECHNOLOGY SECTOR

SHRI . RAM BABU , CHARGE MAN , INFORMATION TECHNOLOGY SECTOR

SHRI . PANKAJ , CHARGE MAN , INFORMATION TECHNOLOGY SECTOR

SHRI . C . PALANI , JUNIOR WORKS MANAGER , TRADE SECTOR .

**Software requirement specification**

Vision:15th june 2018

Prepared by:

Kailasa Eswaran. I

Meenakshi .S

Parthiban. S

Saranya .P

Srinidhi .V

Swetha. P

**CONTENT :**

**1.INTRODUCTION**

**2.AIM**

**3.OBJECTIVE**

**4.SCOPE**

**5.PROECT ANALYSIS**

**i.Existing system**

**ii.Proposed System**

**6.PROTECT DESCRIPTION**

**7. GENERAL REQUIREMENTS**

**8. FUNCTIONS**

**9.INTERFACE REQUIREMENTS**

**10.PERFORMANCE REQUIREMENTS**

**11.DESIGN CONTRAINS**

**12.NON FUNCTIONAL ATTRIBUTES**

**13.OPERATIONAL SENARIO**

**14.PRELIMINARY ATTRIBUTES**

**15.BUGET**

**16.CHARACTERISTICS OF SOFTWARE REQUIREMENTS SPECIFICATIONS**

**17.KEY NOTE**

**18.DEFINITION**

**19.ACRONYMS**

**20.ABBREVIATON**

**21.REFERENCE**

**22.TECHNOLOGY TO BE USED**

**23.TOOLS TO BE USED**

**24.SOFTWARE SPECIFICATION**

**25.PROECT DESIGN**

**26. PROCESS**

**27.PROJECT**

**28.PROJECT DIAGRAM**

**# use case diagram**

**# activity diagram**

**# sequence diagram**

**# class diagram**

**29.CONCLUSION**

**Software requirements specification:**

The software requirements provide a basic for creating the software requirements specifications. Which is called the srs the software requirements specification is useful in estimation the cost , plannings , team work , team activites , performance , team performance , performing task , and tracking the team progress through the developed activites.

Typically the maintenance of the software is design ed by the requirements and the demand of the software which is needed and the specification of the project which is needed by the facilities , not only the works and the maintenance of the software requirements specification the main useful thing is that the estimation and the progession of the software.

The estimating cost , planning team activites , performing the tasks and the tracking of all the softwares needed to be displayed these all are called the developed activites for the software requirements specification.

The software requirements document is the specification of the system. It should be included both the definition and a specification of the requirements , it is not a design document , as far as possible it should be set of what the system should do for the basic need of the software requirements specification process to be held .

The system should be rather the than how should it do the software requirements specification , typically software designers use IEEE STD 830 1998 as the basic software need for the system to process and for the entire software specifications. The standard templates for waiting SRS is as given below in detailed .



**The characteristic of software requirements specification:**

The software requirements specification should address the following basic issues that shall process the project:

Functionality

External interfaces

Performance

Attributes

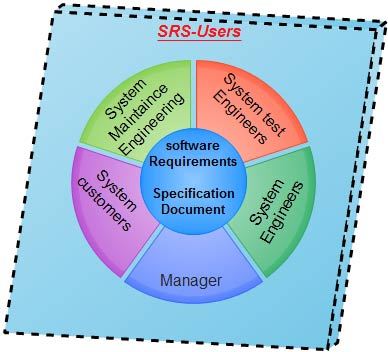
Design constraints imposed on an implementations

Concepts

Understanding level of the process

Including the internal concepts of software

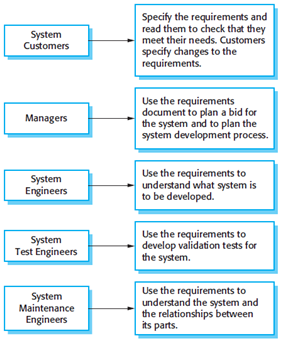
Product validation



**The good software requirements specification contains of the following :**

An software requirements specification should be:

1. Correct
2. Unambiguous
3. Complete
4. Consistent
5. Ranked for importance
6. Stability
7. Verifiable
8. Modifiable
9. traceable



Now let us see a sample of a basic srs outline:

1. Introduction:

purpose

document conventions

intended audience

additional informations

contact information

srs team members

reference

2.Overall description:

Product perspective

Product function

User class

Operating environment

User environment

Design

Assumptions and dependencies

3.External interface requirements

User interface

Hardware interface

Software interface

Communicational protocols and interface

4.System features:

Description

Priority

Action

Result

Functional requirements

5.Other non functional requirements :

Performance requirements

Software quality attributes

Safety requirements

Security requirements

Software requirements

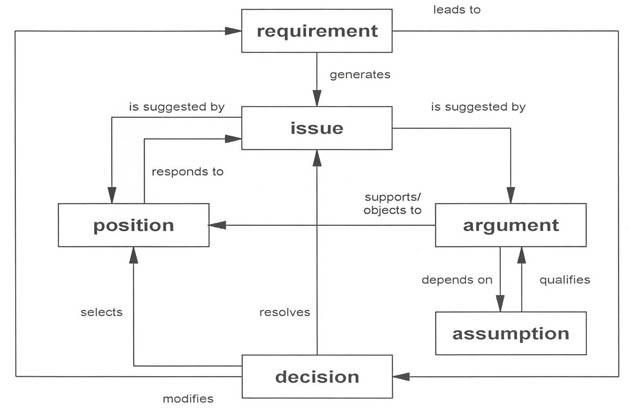
Project documentation

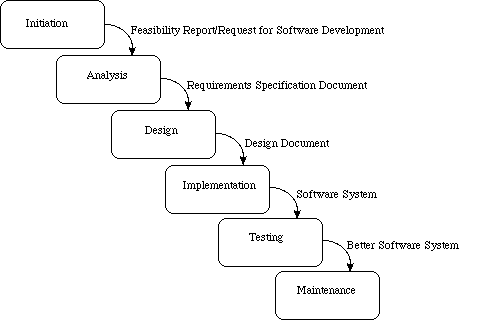
User documentation

There are four software development phases they are requirements analysis planning next comes the design phase and next is the coding and the last is the testing and manintance.

The types of requirements are

1. Customer
2. Functional
3. Non functional
4. Performance
5. Design
6. Derived
7. allocated





**INTRODUCTION:**

The main objective of this software requirement specification is to develop a feedback form for employee undergoing training in the company.we have developed the feedback form for the employees to rate the grading their facilities and this document gives us the details of the function and non function requirements for the maintaince of the feedback form.the purpose of the document is that the requirements mentioned in it should be utilized by the software developers to implement the system.

**AIM:**

The software requirement specification defines external interfacing,performance and the software system attributes requirement of the feedback form.this document is idented for the group of people,staff,trainer,hr.

**OBJECTIVE:**

The main objective of this project is about the development of employee feedback form, to give feedback from different department for different concepts. The employee must login into feedback form by giving his username and password as input. Then the employee must answer the given questions and it will be stores in a database. Another specified user can be able to view all the employees feedback which they have mentioned in the form.

**SCOPE:**

This document applies the feedback form for judging the staff experience and provides the complete information and the details about the feedback system and this allows us to know the availability of the management system and process the application manually .

This also describes the scope of the requirements definition efforts and this section also details any contrains that were placed upon the requirements elicitation process such as achedules and the cost.

**KEYNOTE :**

The php programming language is a server side html embedded scripting language.the php language runs on the server side and it means that the execution of the scripts are done on the server where the website is hosted. system prototyping is a rapid software development for validating the requirements.

**General description:**

the general description is about the general functions and the functionalities of the products such as similar system. And the information , users characteristics , users objects , general constraints placed on design team.

Now lets us discuss the most used and the important elements that are processed in the general purpose of the project

Software document

Software requirements specifications

User interface prototype

Functional requirements

Critically

Technical issues

Cost and schedule

Risk

Dependencies

Interface requirements

GUI

CLI

API

Hardware interface

Communication interface

Software interface

Performance requirements

Design constraints

Other non functional attributes

1. security
2. binary compatibility
3. reliability
4. maintainability
5. portability
6. extensibility
7. reuseability
8. application compatibility
9. resourse
10. utilization
11. serviceability
12. and other as appropriate

**Process:**

The process is the way in which we produce the software and it provides the frame work from which a comprehensive plan for the software development can be established id the process is a weak process the end product will undoubteldly suffer . there are many life cycles models and the process improvements models depending on the type of project it may be suitable or not but the grading of the software is very important for the work to process on continuesly .

A perfect model is to select the perfect software to process the process and the project that are to be selected now a days CMM capability maturity model has been almost a standard for the process framework the process priority is after people and the product and however it plays the main role in the process of the creation in the software process.

A small number of framework activities are applicable to all the software projects regarding the size and the complexity a number of diirent task is performed bt the software which are the tasks , milestones , works , work products , and the last is the quantity and the quality measure . There are some of the framework activities which are to be adopted bty the software and the process to process the valid form to provide the proper characteristics of the projects and the requirements of the project team.

**Project:**

A proper planning is required to monitor the status of the developed and the development and to control the complexity most of the are coming late with cost overruns of more than 100% in order to manage the successful project , we must understand what can go wrong and how to do it right and at the same time we should define concrete requirements although very difficult and freeze the size these requirements . change should not be incorporated to avoid software surprises , software surprises are always risky and we should minimize them.

We should have a planning mechanism to give before the occurrence of amy surprises and all the four factors which are :

1.people

2.product

3.process

4. project

Are very important for the success of the project and mainly these relative important helps us to organize and develop the process using the software and development activities in more scientific and professional way.

**PROJECT DIAGRAM:**

The project diagram consist of the intoducyion to ths UML according to the OMG specification the UML is defined as the unified modeling language is the graphical lamguage for visualizing , specifications , constructions , and the documentating the artifacts of the software intensive system .

The UML is purely process independent that is it is not tied up with the SDLC which is software development life cycle process the basic purpose behind the UML is the modelling the visualization and constructing the specification and documenting the system. It should be noted that the UML is a standard and it is not a methodology process or a procedure .

Rather we use the UML as a standard that uses the predefined standard notation with the review to the modelling and the defining the software system to the document it and the defined artifacts involved .

The UML is restricted to modelling software and it is also used for the business process modelling system engineering modelling and the representation organization and there structure .

**The UML diagrams :**

UML diagrams are the ultimate output of the entire discussiom and all the elements relationships are used to make a complete UML diagram and the diagram represent a system .

The visual effects of the UML diagram is the most important part of the entire process and the other elements are used to make it a complete one.

UML includes the following diagrams :

Class diagram

Object diagram

Use case diagram

Sequence diagram

Collaboration diagram

Activity diagram

State chat diagram

Deployment diagram

Component diagram

The UML plays an important role in defining different perspectives of a system these perspective are the very important to distinguish between the UML model. Different diagrams are used for different types of UML modeling there are four different of methods important tyes of UML modelings .

1. design
2. implementation
3. process
4. deployment
5. modeling

Usecase diagram:

Usecase diagram consists of usecase and actors nad shows the intraction between cases and actors. A use case diagram at its simplest is a representation of a of a users interaction with the system and depicting the specification of the use case . A use case diagram can portray the different types of users of the system and the various ways that they interact with the system this type of diafram is typically used in the conjunction with the textual use case and the will often be accompanied by others types of diagram as well.

The following are the elements are available in the usecase diagram:

Actor

Usecase

Association

Direct association

Indirect association

Generalization

Dependency

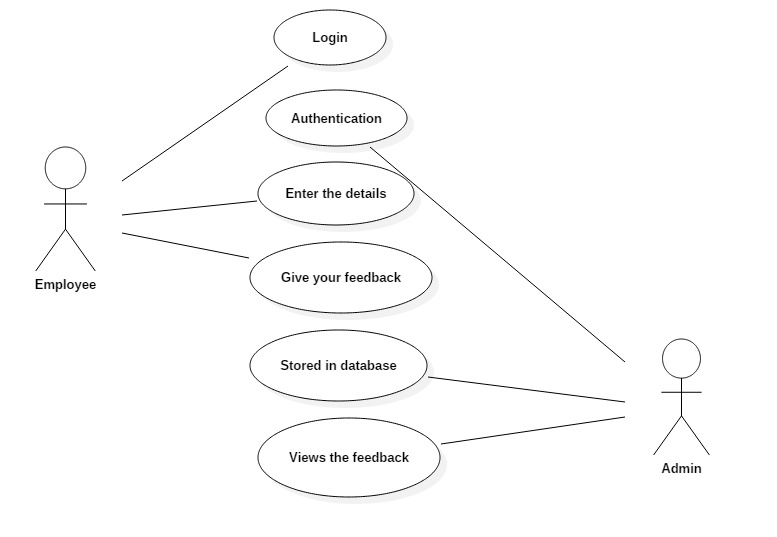
Include

Extend

System boundary

Package

**DOCUMENTATION OF USECASE DIAGRAM**



Activity diagram:

Activity diagram is another important diagram in uml to describe dynamic aspects of the system . it is basically a flow chart to represent the flow for of one activity to another activity. This diagram also describes the flow of the control of the system and in the system . so it consists of activities and links the flow can be sequential and concurrent or branched also. Activities are nothing but the functiona of a system. Numbers of activity diagrams are prepared to captured the entire flow in the system. Activity diagrams are used to visualize the flow of controls in the system this is prepared to have an idea of how the system will work when executed.

The following elements are available in the activity diagram:

Action state

Subactivity state

Initial state

Final state

Synchronization

Decision

Flow final

Object flow

Signal accept state

Signal send state

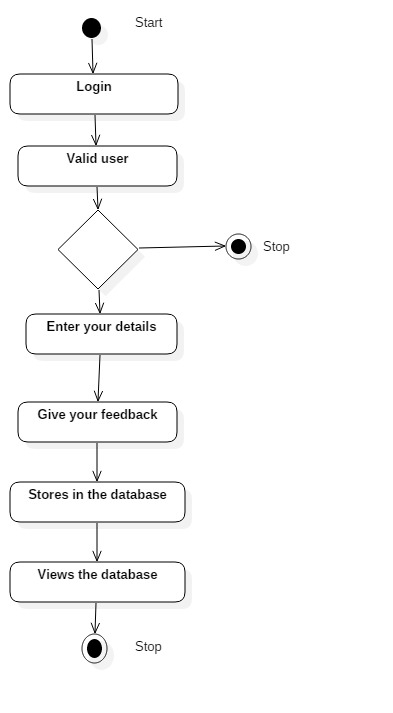
Transition

Self transition

Swimlane

**DOCUMENTATION OF ACTIVITY DIAGRAM**

­



**Sequence diagram:**

A sequence diagram is a set of interaction diagram from the name it is clear that the diagram deals with the some of the sequence which are the sequence of message flowing from one object to the another objects. Interactions among the components of the system is very important from the implementation and the execution perspective.

The following elements are used in the sequence diagram:

Object

Stimulus

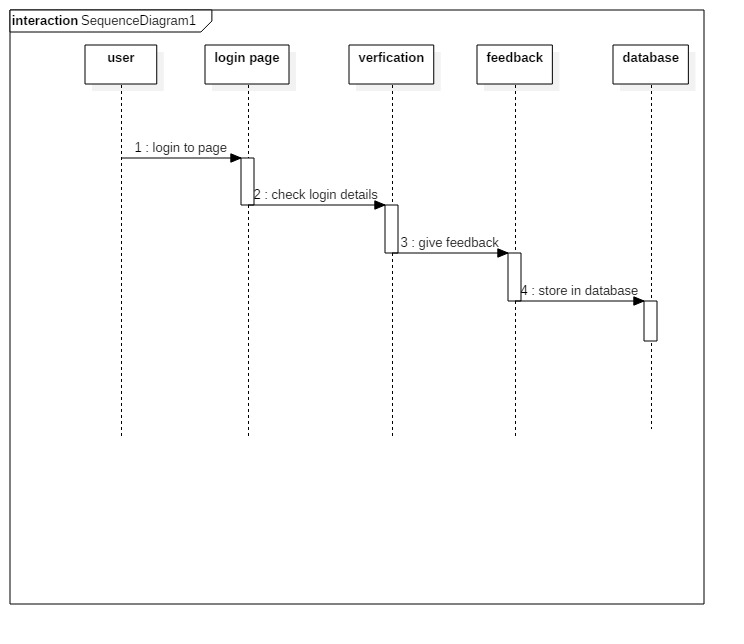
Self stimulus

Combined fragments

Interaction operand

Frame subsystem

**DOCUMENTATION OF SEQUENCE DIAGRAM:**



**Class diagram:**

Class diagrams are the most common diagrams used in the UML a class diagram consists of the class , interfaces , associations , collaborations. The class diagram basically represents the object oriented viewof the system which is static in nature .

The following are the elements used in the class diagram:

Link

Object

Subsystem

Package

Class

Interface

Enumeration

Signal

Exception

Port

Part

Associations

Direct associations

Indirect associations

Aggregation

Composition

Generalization

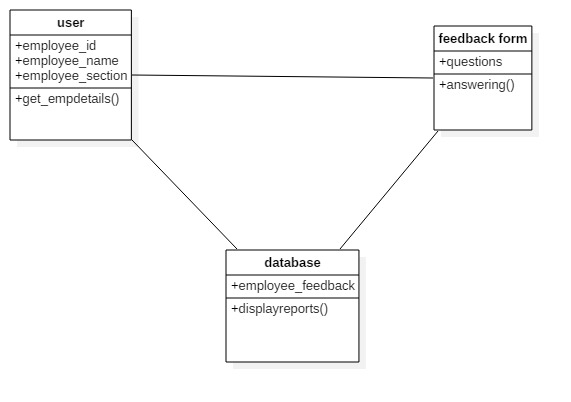
Dependency

Realization

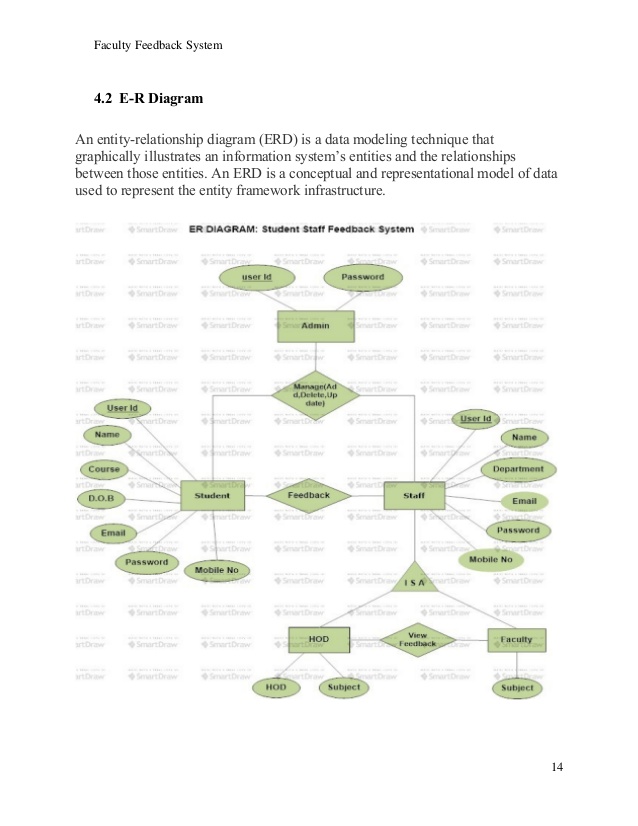
Association class

Connector

**DOCUMENTATION OF CLASS DIAGRAM:**



**ER DIAGRAM:**



**DEFINITION,ACRONYMS,ABBREVIATION**:

Microsoft office,star uml ,html-hypertext markup language ,php.

**SOFTWARE USED:**

Html

Bluevoda

Css

Php

Wamp server

**CONCLUSION:**

Thus we used many software to implement our project and the results are successfully verified in software requirement specification. We came across the basic scopes and purpose of many software and hardware modules for this project.

Software engineering has become very important discipline of the study , practice and research .