SOFTWARE ENGINEERING LAB

Dr:- shimaa saad Assistant lecture,

Department of computer science

Email : shimaa_saad@yahoo.com

Telephone : 01222956341

Requirement of the lab

- **Hardware Requirements**: Pentium 4 processor (2.4 GHz), 128 Mb RAM, Standard keyboard n mouse, colored monitor.
- **Software Requirements:** Visio or Rational Rose, Windows XP/2000,ms-office.

- This lab deals with the analysis and design of a software problem.
- The tool used in a lab is Visio or rational rose .
- This tool is used for a object oriented design of a problem .
- We draw a UML diagram in a rational rose or visio which deals with the objects and classes in a system .

The Unified Modeling Language (UML)

- It is a mostly graphical modelling language that is used to express designs.
- It is a standard language for specifying, visualizing, constructing, and documenting the artifacts and components of software systems, as well as for business modeling and other non-software systems.
- It is important to understand that the UML describes a notation and not a process. It does not put forth a single method or process of design, but rather is a standardized tool that can be used in a design process.

The Unified Modeling Language (UML)

- هي لغه نمذجة رسومية تقدم صيغة لوصف العناصر الرئيسية للنظم البرمجية(artifacts)
- تستخدم لعمل رسوم تخطيطية لوصف برامج الكمبيوتر من حيث العناصر المكونة لها و خط سير العمليات به.
- تمكن من انشاء نماذج و تصميم متكامل لمشروعك البرمجى.

The Unified Modeling Language (UML)

- The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.
- The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.
- Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

EXCERCISE NO. 1

- AIM: To prepare PROBLEM STATEMENT for any project.

REQUIREMENTS:

- **Hardware Interfaces**

▪ Pentium(R) 4 CPU 2.26 GHz, 128 MB RAM

▪ Screen resolution of at least 800 x 600 required for proper and complete viewing of screens. Higher resolution would not be a problem.

▪ CD ROM Driver

- **Software Interfaces**

▪ Any window-based operating system (Windows 95/98/2000/XP/NT)

▪ WordPad or Microsoft Word

- **THEORY:**
- The problem statement is the initial starting point for a project.
- It is basically a one to three page statement that everyone on the project agrees with that describes what will be done at a high level. The problem statement is intended for a broad audience and should be written in nontechnical terms.
- It helps the non-technical and technical personnel communicate by providing a description of a problem. It doesn't describe the solution to the problem.
- The input to requirement engineering is the problem statement prepared by customer. It may give an overview of the existing system along with broad expectations from the new system. The first phase of requirements engineering begins with requirements elicitation i.e. gathering of information about requirements.
- Here, requirements are identified with the help of customer and existing system processes. So from here begins the preparation of problem statement. So, basically a problem statement describes what needs to be done without describing how.

EXCERCISE NO. 2

- Aim: Understanding an SRS.
- Software requirements specification (SRS)

SRS

- مجموعه منظمة من المعلومات التي تجسد متطلبات النظام و تصف كل التفاعلات بين المستخدم و البرمجية.
- عادة ما يكون مستنداً يتضمن غرض و وصف شامل للمشروع و متطلبات محددة و فيه يكون وصف إمكانات النظام المحتملة مع مراعاة كيفية عمل النظام المستقبلي و تحقيق اهداف المستخدمين.

Requirements:

- **Hardware Requirements:**

• PC with 300 megahertz or higher processor clock speed recommended; 233 MHz minimum required.

• 128 megabytes (MB) of RAM or higher recommended (64 MB minimum supported)

• 1.5 gigabytes (GB) of available hard disk space

• CD ROM or DVD Drive

• Keyboard and Mouse(compatible pointing device).

- **Software Requirements:** Rational Rose, Visio, Windows XP,

---

- **Theory:**

An SRS is basically an organization's understanding (in writing) of a customer or potential client's system 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.

Theory:

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

Theory:

- 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. An SRS is the customer's assurance that the development organization understands the issues or problems to be solved and the software behavior necessary to address those problems. Therefore, 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, places borders around the problem, solidifies ideas, and helps break down the problem into its component parts in an orderly fashion.
- • It serves as an input to the design specification. As mentioned previously, 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.

SRS should address the following

- a) **Functionality.** What is the software supposed to do?
- b) **External interfaces**. How does the software interact with people, the system‘s hardware, other hardware, and other software?
- c) **Performance.** What is the speed, availability, response time, recovery time of various software functions, etc.?
- d) **Attributes.** What are the portability, correctness, maintainability, security, etc. considerations?
- e) **Design constraints imposed on an implementation**. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s)

Characteristics of a good SRS

An SRS should be

- a) Correct
- b) Unambiguous
- c) Complete
- d) Consistent
- e) Ranked for importance and/or stability
- f) Verifiable
- g) Modifiable
- h) Traceable

SAMPLE SRS (9)

SOFTWARE REQUIREMENTS SPECIFICATION

ATM Version 1.0 September 8, 2006

AN AUTOMATED TELLER MACHINE

ATM

- **1. Introduction**
- The software ATMExcl 3.0TM version1.0 is to be developed for Automated Teller Machines (ATM). An automated teller machine (ATM) is computerized telecommunications device that provides a financial institution's customers a secure method of performing financial transactions, in a public space without the need for a human bank teller.
- Through ATMExcl 3.0TM ,customers interact with a user-friendly interface that enables them to access their bank accounts and perform various transactions.

ATM

- **1.1 Purpose**
- This SRS defines External Interface, Performance and Software System Attributes requirements of ATMExcl 3.0TM.
- This document is intended for the following group of people:-
- ✓ Developers for the purpose of maintenance and new releases of the software.
- ✓ Management of the bank.
- ✓ Documentation writers.
- ✓ Testers.

ATM

- **Table of Contents p. 13,14,15,16**

ATM

- 2. The Overall Description
- 2.1 Product Perspective
- 2.2 Product Functions **p 17,18**
- 2.3 User Characteristics
- 2.4 Constraints **p 18,19**
- 2.5 Assumptions and Dependencies

ATM

- 3. External Interface Requirements **p.20,21**
- 3.1.1 User Interface Requirements
- 3.1.2 Hardware Interface Requirements **p.21,22**
- 3.1.3 Software Interface Requirements
- 3.1.4 Communication Interface Requirements

ATM

- 4. System Features **p.22,23,24**
- 1. Remote Banking and Account Management
- Description
- Validity Checks
- Sequencing Information
- Error Handling/ Response to Abnormal Situations
- 2. Receipt Generation

ATM

- 5. Other Nonfunctional Requirements
- 5.1 Performance Requirements **p.25**
- 5.2 Software System Attributes **p.26**
- 5.3 Business Rules **p.27,28**
- Appendix A: Glossary