

The IEEE standard for requirements documents

The most widely known requirements document standard is [IEEE/ANSI 830-1998](#) (IEEE, 1998). This IEEE standard suggests the following structure for requirements documents:

1. Introduction

- 1.1 Purpose of the requirements document
- 1.2 Scope of the product
- 1.3 Definitions, acronyms and abbreviations
- 1.4 References
- 1.5 Overview of the remainder of the document

2. General description

- 2.1 Product perspective
- 2.2 Product functions
- 2.3 User characteristics
- 2.4 General constraints
- 2.5 Assumptions and dependencies

3. Specific requirements, covering functional, non-functional and interface requirements. This is obviously the most substantial part of the document but because of the wide variability in organisational practice, it is not appropriate to define a standard structure for this section. The requirements may document external interfaces, describe system functionality and performance, and specify logical database requirements, design constraints, emergent system properties and quality characteristics.

4. Appendices

5. Index

Although the IEEE standard is not ideal, it contains a great deal of good advice on how to write requirements and how to avoid problems. It is too general to be an organisational standard in its own right. It is a general framework that can be tailored and adapted to define a standard geared to the needs of a particular organisation.

(c) Ian Sommerville 2008

Activate Windows
Go to Settings to activate Windows.

1. INTRODUCTION

1.1 PURPOSE

The reason for this system, To make the process of management of clients registration simple and easy instead harder one traditional way such (Excel, Notebook).

1.2 PROJECT SCOPE:

It is a Software Desktop Application .

We develop and implement this application for all participants in the club, with the objective of preventing not participants to enter the club. Further, Administration the club thru the system.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE:

It is a Software desktop application based system implementing client-server model.

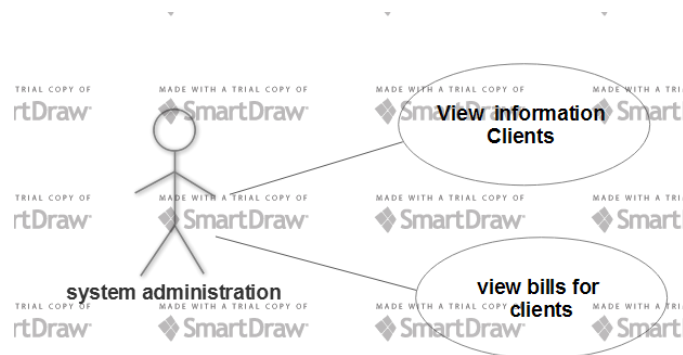
The following are the main features that are included:

- Make process for registration easy.

- Easy to CRUD (Create,Read,Update,Delete).
- Easy to know expired date.
- Easy to know detail of payment such (Received,Residual)
- A simple way to inform a client about The rest of subscription -

2.2 Use Cases and CHARACTERISTICS:

The system will support admin user , admin has know View Registration for clients & bills.View estimate data,valid thru ,period , know information for given client (such phone,address,email... so on) , and View associated bill for given client.



2.3 OPERATING ENVIRONMENT:

Operating environment for the power gym club system is as listed below:

- Operating system: Windows.
- database: mysql+ database
- platform: /Java/PHPmyadmin local host

2.4 Design and Implementation Constraints:

- SQL commands for above queries/applications
- The global schema, fragmentation schema, and allocation schema.

Using Mysql database (phpmydamin) for implementation database

FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS:

CLIENT/SERVER SYSTEM &Database

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

All the data resides at the server sites.

Records database is store in server side

Can CRUD Database ,

USER INTERFACES:

Front-end software: Java Gui

Back-end software: SQL

HARDWARE INTERFACES:

Windows.

A browser which supports database (mysql phpMyAdmin)

This project supports all types of web browsers.

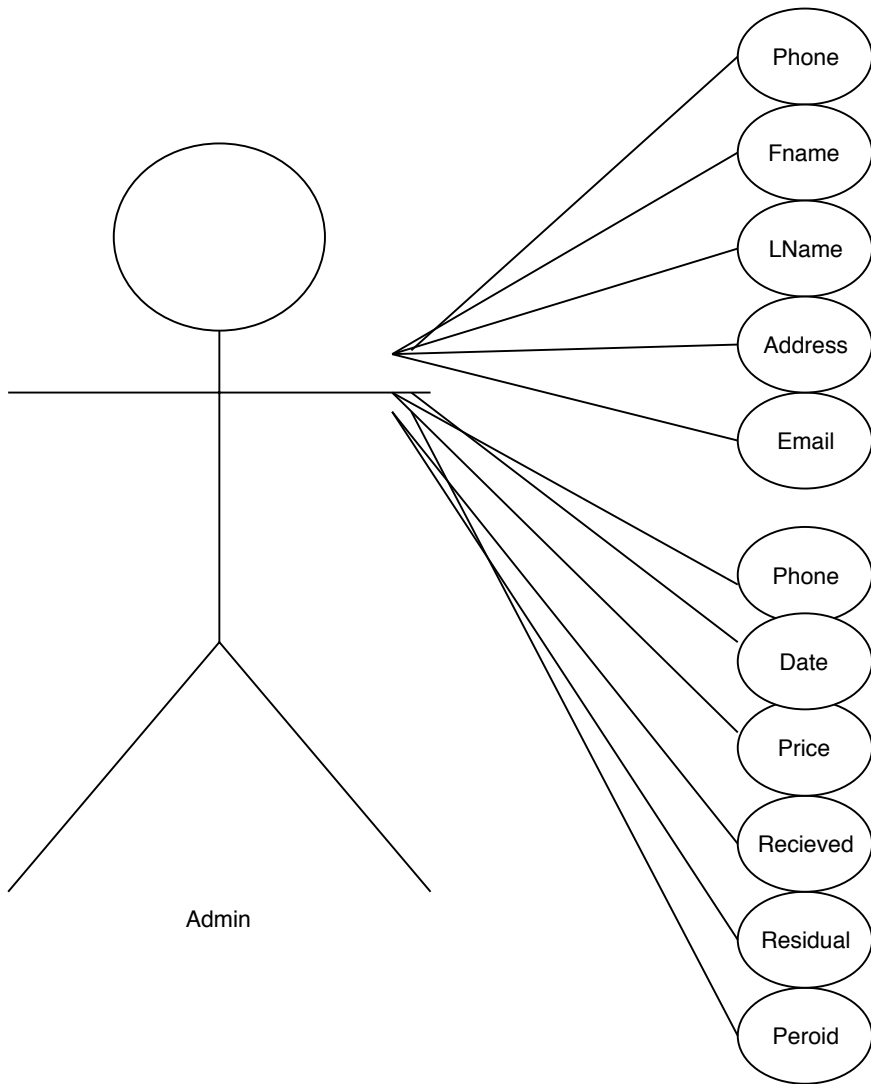
SOFTWARE INTERFACES:

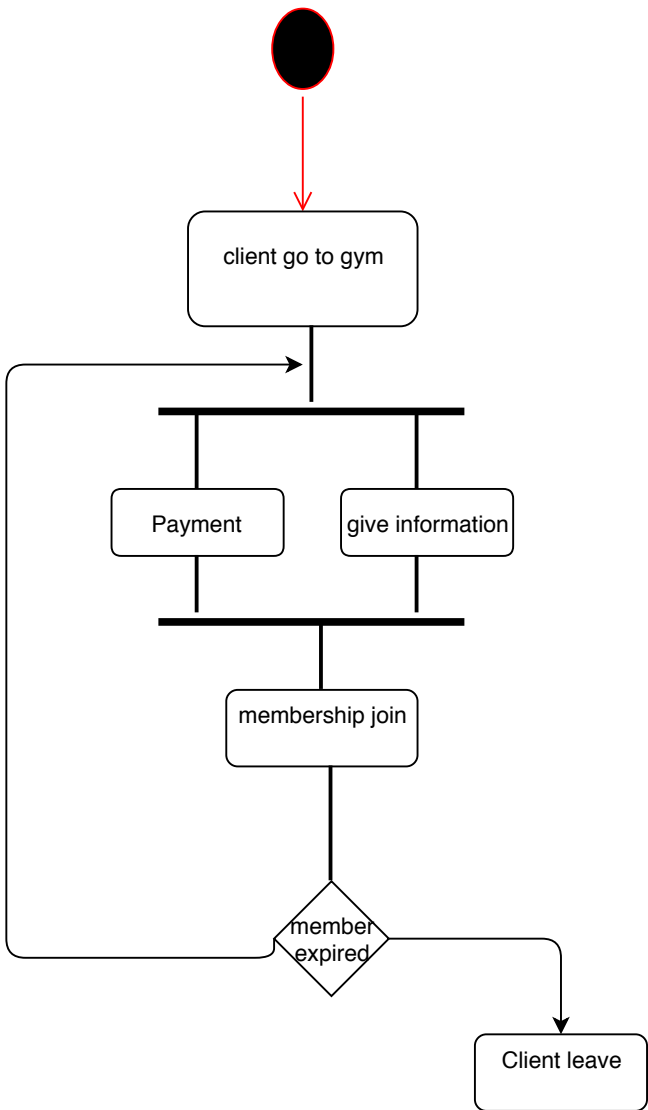
Following are the software used for the power gym club system :

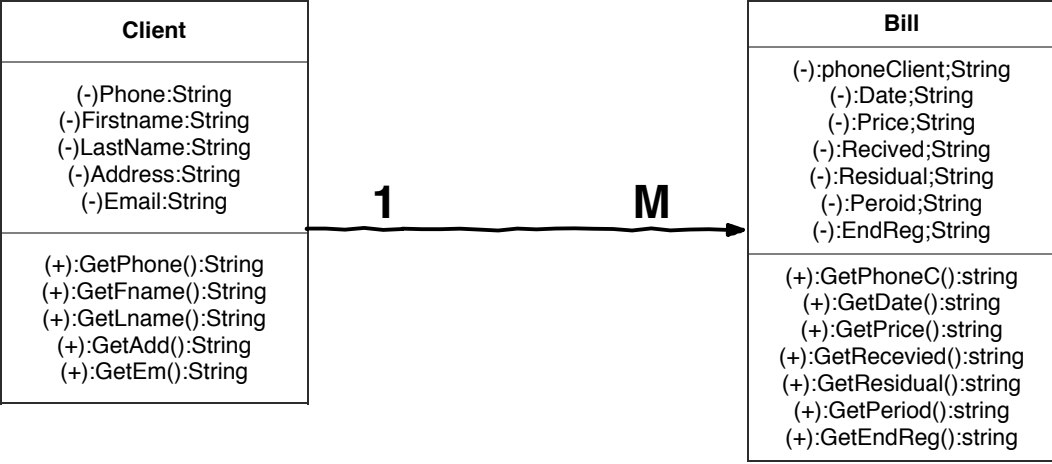
Software used	Description
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the client records, bills records we have chosen SQL+ database.

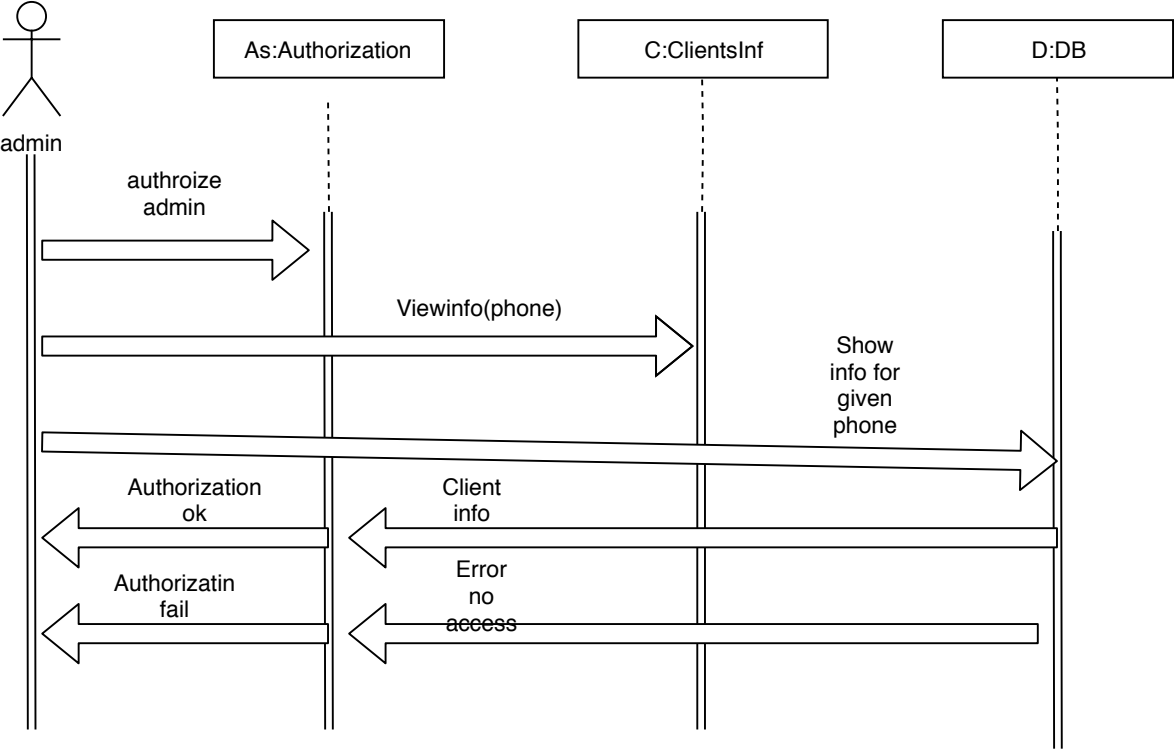
Detailed Non-Functional Requirements/other non-functional requirements/Software attributes:

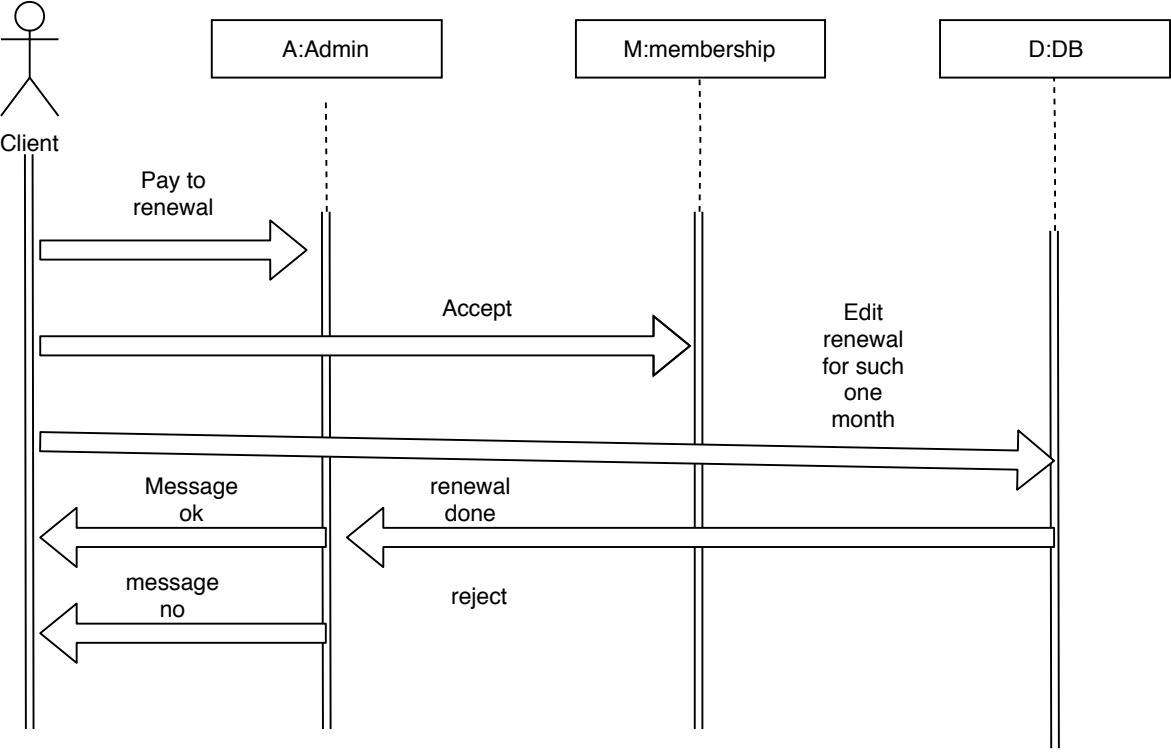
- Scalability: developers can expand and add new features without affecting the other functionalities within the system.
- Availability: the system developed and tested, in order to minimize it's single point of failure opportunities,
- Security: developers of the system were very careful in validating all user input.
- Recoverability: we care about our client and made sure to keep their personal information and data safe as much as possible, we developed a risk management plan and also the system load all data to the backup in real time.
- Reliability: all function of the system are complete.

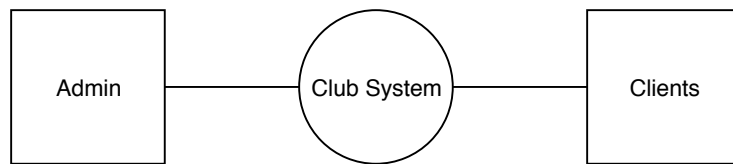






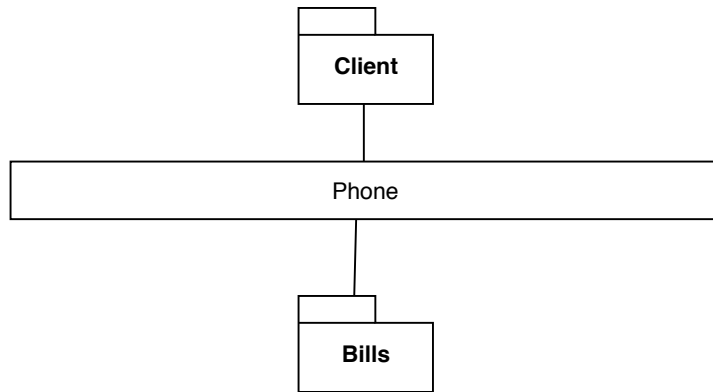






System Context Diagrams ... represent all external entities that may interact with a system ... Such a diagram pictures the system at the center, with no details of its interior structure, surrounded by all its interacting systems, environments and activities. The objective of the system context diagram is to focus attention on external factors and events that should be considered in developing a complete set of systems requirements and constraints.

System context diagrams are used early in a project to get agreement on the scope under investigation.^[4] Context diagrams are typically included in a requirements document. These diagrams must be read by all project stakeholders and thus should be written in plain language, so the stakeholders can understand items within the document.



Client:information for client

Bills :details for given client

Relation between them is phone to relaize bills for appropriate person