INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

SOFTWARE Engineering session

2015-16 ASSIGNMENT 4

Swimming Pool Management Software Structural Analysis and Structured Design

Document By:

MADHU KUMAR DADI: 14CS10012

KONIDALA MUNI LOHITH KRISHNA MOHAN: 14CS10026

Contents

INTRODUCTION

1.1 Purpose

This Structured Analysis (SA) document provides a complete analysis of the internal functioning of the software via Sequence, Collaboration, Activity and State chart Diagrams. The Structured Design (SD) document provides all the design specifications in terms of the operating systems, programming languages, databases etc.

1.2 Scope

This project aims to computerize the administration of a swimming pool. There are several aspects to this, including granting membership, scheduling pool activities, booking pool slots, arranging swimming courses and swimming competition, pool fee payment for members etc. The users of the system are the non-members, members of the pool, swimming course coordinators, pool committee members including event manager and the manager with different levels of access.

1.3 Glossary

Manager: A person who is head of the administration .He has all the powers regarding the swimming pool management.

Non Member: Anyone who is not a member of the Swimming Pool Club or is not in its administration board .A non-member can apply for courses and participate in the events conducted by the pool management by using the pool's portal.

Member: Anyone who is a registered member of the Swimming Pool Club. He / She should apply for the membership by submitting required specifications to the manager via Swimming pool's portal. If manager accepts he/ she becomes the member of the pool.

Committee member: Anyone who is in its administration board of Swimming Pool Club. Committee members look after the pool slots and they reserve special slots for children and women.

Coarse coordinator: Anyone who is in-charge of a certain course in Swimming. He announces his course and its slots such that interested members of the pool or non-members can apply for the course. Course coordinators would be one of the committee members.

Event Manager: The Person who manages the particular event. Event manager will be one of the committee members or the manager itself. He looks after the event and its schedule.

Calendarium: A way to make changes in the system database.

SP database: Refers to the system database.

1.4 References

[1] IEEE Software Engineering Standards Committee, IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications, October 20, 1998.

1.5 Overview of the document

The remaining chapters and their contents are listed below. Section 2 is Feasibility study which helps us understand the problem in depth. In this section, we analyse the various stakeholders and discuss the alternatives. Section 3 is the Requirements Analysis in which we first review the feasibility study and look into it for deeper information. Also in this section we gather functional and non-functional requirements. Section 4 is the final section in which we de ne the global system architecture, select platform comprising hardware, software and networking. Also we propose software architecture for the new system and do database design.

2 Feasibility Design

2.1 Understanding the problem

In this section, we try to understand the purpose of the software. The SPMS (Swimming Pool Management Software) is intended to administer a Swimming Pool Club and its activities. The SPMS will help people become a member of the club, take part in swimming competitions and en roll for swimming courses. It will help pool club members to book pool slots, book pool, en roll for courses/competitions and pay fees. The Manager/Committee member/Course-coordinators can take care of administrative duties via this software. There is a forum for social networking among pool members. More specifically, the goal of this software is to allow easy management of the Swimming Pool Club.

2.2 Scoping the problem

The following actions are to be performed for proper functioning of software.

- Accessing the system portal
- Application for membership
- Booking Pool slots
- Booking Pool for activity
- En-roll in course
- En-roll for competition
- Modify system database Accessing forum

2.3 Analysing stakeholders

The various stakeholders are:

- Non-member
- Member
- Committee Member
- Event Manager
- Coarse coordinator
- Manager

2.4 Defining alternatives

2.4.1 Connection between member and software

After opening the portal, there will be an option to apply for membership. Anyone can apply for membership. He / she should fill all the details in the membership form generated. He should be able to upload his birth certificate and medical certificate and any other specific requirements asked by the manager. All his information will be passed to the manager. After the verifying the manager may or may not accept his /her membership request. If manager accepts the request, then a user Id and a password will be mailed to the applied candidate. He can further use his account for several purposes of the pool.

2.4.2 Connection between non-member and software

After opening the portal, everyone can see the courses and events that are going to be held by the swimming pool management, but only members can login to their page. Non-members will have an 'APPLY' option to apply for the courses and events. If they apply for the course, they get a login Id and password to their mail. If they apply for an event they get a mail regarding the event, which serves as the permission to participate in the event.

2.4.3 Connection between various administrators and the software

- Every administrator have their own account, including the manager.
- Manager has the supreme power. He can make any changes in the pool's administration.

- Manager itself adds all the committee members and give them an user Id and password
- Committee Members take decisions regarding the slots and special slots for women and children.
- All the decisions will be notified to the manager and the decisions will be implemented only after the acceptance of Manager.
- Committee members announce their courses and slots .List of all courses will be displayed in the home page of SPMS .That particular committee member becomes the course coordinator of his course. Any non-member or member can apply to that particular course.
- Committee members or the manager itself can conduct events .Then they become the Event Manager and look after the event and its timing slot.

2.4.4 Hardware infrastructure

Although the software is designed to run on Linux and Windows Operating systems, it can be alternatively designed to run compatibly with MAC OS X systems. It can be designed in such a way that instead of requiring a RAM of 512 MB, it requires only 256 MB of RAM. Instead of using the internal memory of the system on which SPMS is being run, we can use external hard disk of any size given that it satisfies the minimum hardware requirements as mentioned in SRS. SPMS can be alternatively designed to run on a 32-bit machine instead of a 64-bit machine.

2.4.5 Software infrastructure

The database system used for SPMS can use SQL server, Sybase, Access, MySQL or Oracle for managing the database. Depending on our requirements, cross platform compatibility, customized controls, ease of accessibility and speed, we can use other programming languages which offer a Graphical User Interface (GUI) such as PHP.

2.4.6 Technology Used

We can use a 3-tier architecture which comprises of the client, the server and the database instead of using a 2- tier architecture which comprises only of the client and the database without any interference of the server. In case, we can compromise on the graphical aspect of the SPMS, we can use programming languages such as C, C++ instead of Java.

2.4.7 Security

The SPMS can be designed for various levels of security for different supervisors. Instead of using password protection for login into the system, we can also incorporate Face recognition, a Bar Code scanning system, or a thumb impression recognition system for login of the important stakeholders.

2.5 Defining criteria to evaluation

The primary criteria which are to be kept in mind while evaluating the alternatives are: Cost of technology Cost of infrastructure Lifetime of technology Stability of technology

2.6 Assessment of unusual circumstances

In case the SPMS uses a 3-tier architecture, it need to be hack proof by designing it using the concepts of special algorithms, data encryption and server-based cryptography. The design should take care of the fact that the data is not lost in any case, be it software or a hardware failure, system going down or any unusual circumstances that might intervene in between the smooth functioning of the SPMS. For this, we can design a MASTER system which stores the backup of all the data which is fed into the SPMS time to time. This master system would allow us to retrieve data at any point of time and restore the database to its original state.

2.7 Evaluation of alternatives

2.7.1 Connection between members and software

If the resident is given the option to register from his work place or home, the server has to adopt a 3-tier architecture which increases the security concerns. In our problem, we have not been instructed to give the customer this option. Hence, this alternative has lesser priority.

2.7.2 Connection between various administrators and the software

If we use multiple machines, we have to use server and as mentioned above we will have security concerns which in turn will increase the cost of technology. Also, multiple machines will directly increase the cost of infrastructure.

2.7.3 Hardware infrastructure

Using an external hard disk to save the database will indirectly imply a backup of the database along with the master system which can be retrieve the data at any point of time desired. Thus, this is better than using the system hard disk to store the data. This ensures that the software does not any space on the system hard disk. The SPMS can be designed for a 32-bit system as well as a 64-bit system. Designing the SPMS for a 64-bit system should be preferred for the graphical interface concerns.

2.7.4 Software infrastructure

Any database system can be used for the managing the database of SPMS and all have equal priority.

2.7.5 Technology used

If we use the 3-tier system, we will have security concerns and hence will affect the cost of technology. Thus, this alternative should have less priority and 2-tier system has high priority. Since we do not deal with algorithm intensive techniques in SPMS, we need not go for programming languages like C++ and Java would suffice.

2.7.6 Security

If the SPMS uses security systems like thumb impression recognition, the cost of technology increases but the security increases. As the cost of production is a major concern for the SPMS design, the security alternatives should be given the least priority. Also, the SPMS is less prone to hacking and security threats.

3 Detailed Design

3.1 Global system architecture

The overall system architecture is a 2-tier architecture which includes client at one end and the database at the other. There is no server based middle tier in the software being designed.

3.2 Platform

The software is developed on Java Platform, and requires Java Development Kit to be installed on the
Computer for accessing the functions of the software.

3.3 Software architecture

Object-oriented architecture forms the basis of the SPMS. In this style data representations and their
Associated prim-active operations are encapsulated in an abstract data type or object. The components of this style are the objects or instances of the abstract data types. Objects interact through function and procedure invocations. Two important aspects of this style are (a) that an object is responsible for preserving the integrity of its representation (usually by maintaining some invariant over it), and (b) that the representation is hidden from other objects. Thus the aspects of OOA mentioned justify our choice.

3.4 Report

In the feasibility study, we went through the complete details of the problem. The objectives of the SPMS have been laid out and the various scopes have been discussed in detail. Firstly we understood the complete problem and found the various functions that the software performs. Then, various cases were analysed with the help of use case diagrams along with the brief description followed by a step-by- step description of each use case. Each of the use cases were supported by the use case diagrams for ease of understanding. Then, the various alternatives were developed keeping in mind the cost and the lifetime

of the components the alternative brings with it and hence the advantages and disadvantages were highlighted. These alternatives included the hardware, software, technology, security and many other aspects which form an integral part of the software and which could be incorporated in the SPMS, if desired. The primary criteria for evaluation were expected lifetime, cost, stability, and instability of the technology. The unusual circumstances like loss of data due to hardware or software failure or hacking were taken care of by certain concepts of data backup, cryptography etc. At last, all the alternatives proposed earlier were analysed in depth and their advantages and boon to the SPMS were clearly mentioned. A very vivid comparison was made between the SPMS development without the alternatives and as it would function with the alternatives if incorporated in the software.

Under the detailed design section of the software design, the global system architecture was discussed. The
SPMS has a 2-tier architecture comprising of the client and the database with no server. Then the platform requirements for the SPMS was discussed in terms of the operating system, the processor required, the minimum and recommended hard disk space and RAM requirements, etc. The software architecture of the SPMS was later stated to be of the object-oriented type using JAVA as the core technology. The important aspects of OOD used for the SPMS are data abstraction and the preservation of integrity of the software.