

ConMan - The Contest Manager

Preliminary Report

Dilip Antony Joseph (CS00294)

D V S Ravikant (CS00158)

N R Prasad (CS00169)

January 15, 2003

1 Introduction

Online competitions have become very popular in today's world. In these contests, the participants, who may be located in different parts of the world, compete with one another using computer networks as the medium of communication. This calls for appropriate software to manage the various aspects of the competition. Tools exist for managing some specific contests, for example a programming contest. Developing a contest management tool for a specific contest from scratch is a formidable exercise. A generic framework, handling the common features of different contests, will be very useful in developing software for specific competitions.

As part of the Software Engineering Course, we aim at developing a generic contest management framework along with a few specific contest managers built on top of this framework. This document is a preliminary report on the project.

2 Project Definition

The aim of this project is to develop a generic framework for online contest management. This framework can provide a head start in developing software for specific competitions. Contest Managers for the following competitions will be developed using this framework.

1. Programming Contest
2. Housie or Thumbola
3. Quiz Contest

3 Motivation

PC2, developed at the University of California, was used in the Shaastra Programming Contest 2002. This software did not possess a few desired features and it did not provide facilities for adding new features. A more flexible contest manager can be of use in future Shaastra Programming Contests and may also be tailor-made to suit various other competitions in different environments.

4 Requirements

The requirements for the various parts of the project - Generic Framework, Programming Contest, Housie(Thumbola) and Quiz are listed separately in this section. The detailed specifications can be found in the Software Requirements Specifications (SRS) document.

4.1 Generic Framework

1. It must provide a very well-defined and documented interface, which can be used to develop contest managers for specific competitions by just adding the contest specific modules.
2. There should be a provision for registration of the participants either through a web page or in person.
3. The framework must provide administrative support for the contest - for example, mechanisms to start and stop contests, sending announcements, disqualifying users etc.
4. The required network communication facilities (for example, sending and receiving files, messages etc) must be provided by the framework.

5. Provisions for maintaining and displaying the status and scores of the contest must be available.
6. The scoring system provided by the framework must be flexible so that the organizers can use any appropriate scoring methodology.
7. Secure authentication of the participants, judges and administrators must be done.
8. The framework should support mechanisms to provide publicity for possible sponsors of the competition.
9. A comprehensive contest log must be maintained by the framework. Automatic reports can be generated on the basis of these logs.
10. The framework must be able to gracefully recover from unforeseen system crashes.
11. Since the software is going to be used in online contest, where the contestants may be working on different platforms, it is obligatory for the software to be platform independent.

Apart from the requirements for the generic framework mentioned above, the additional requirements of the various specific contest managers are described in the following sections.

4.2 Programming Contest

1. A contestant must be able to send code for validation. He can request clarifications and test cases to the judge. The replies received by the contestant are to be appropriately displayed.
2. The scores and the current position can be viewed by a contestant.
3. A complete log of the contestant's transactions with the administrator and judge will be available to him.
4. A contestant must be able to change his personal settings - for example the contestant password.

5. Judging the programs submitted by the contestants can be manual or automatic. In the manual mode, a human judge must be able to view and evaluate program submissions and also respond to clarifications from the contestants. Automatic enables scripts to be setup which will automatically evaluate the programs submitted.
6. The administrator of the contest can add and remove problems and test cases. The set of allowed programming languages can also be manipulated by the administrator.
7. Methods to analyse the log of the contest should be available to the administrator.
8. The administrator can set the validation mode (manual/automatic).

4.3 Housie(Thumbola)

1. The contestants should receive the Thumbola cards and the numbers called out each time.
2. A contestant must be able to submit claims for prizes at the appropriate instants.
3. There should be a provision for the contestants to obtain his scores.
4. The judge of the competition must be able to check the validity of claims made for prizes on the basis of the numbers called out and the contestant's card.
5. The judge also needs to answer clarifications sought by the contestants.
6. The administrator should be able to pick distinct random numbers which are then announced to the contestants. The administrator also keeps track of the numbers drawn.
7. The issue of Thumbola cards to the various contestants is to be handled by the administrator.

4.4 Quiz

1. The contestant should receive the questions posed by the quiz master.
2. The contestant can submit answers to the questions and also seek clarifications or hints from the quiz master.
3. The answers given by a contestant for the current question must be visible to all other contestants.
4. The quiz master sends questions (text, visuals, audio etc) to all the contestants.
5. There must be a provision for the quiz master to ensure that a contestant does not exceed his time limit.
6. Provisions to answer clarifications and broadcast messages to the contestants should be made available to the quiz master.

4.5 Risks

1. The number of available work hours cannot be estimated accurately due to lack of information. Further the available work hours could be unevenly spread during the period of the project. So there is a risk of not being able to complete all the specific contest managers.
2. There is a risk of the software not providing real time response to the requests due to inherent limitations of the network infrastructure.

4.6 Conclusion

This is a large and involved project which will require the use of good software engineering practices to ensure successful completion. The next stage of the project will be to develop a project plan.