# Technical Proposal for Online Judge System

**SEN Team #3**

This proposal enlists the problems encountered with existing online judge systems, proposes some treatment and lays out the plan of how the team would approach the project.

Team Leader - Avadhesh Gadia (200901028)
Chinmany Modi (200901039)
Arunangshu Bhakta (200901026)
Pankaj Bhambhani (200901047)
Unique Jain (200901036)
Anand Mudgerikar (200901056)
Harsh Chawda (200901052)
Mukesh Makwana (200901032)
Jay Oza (200901014)

**1/15/2012**

# Table of Contents

# Problem Diagnosis

- Currently most of the systems (SPOJ - www.spoj.pl, CodeChef - www.codechef.com ) are competition oriented. Competing without prior knowledge could sometimes become hazardous. It is often extremely difficult for newbie programmers to actually learn/solve more problems, since there is no specific guiding system implemented. People are expected to access internet resources and learn on their own, which is often detrimental to learning. A well designed paradigm for programming needs to be exercised in order to adopt programming in a full-fledged manner.
- In such systems, whenever a new user enters, he/she is bombarded with an enormous set of questions with varying levels of difficulty. He does not have much flexibility to choose his path. A crude user interface doesn't help either. Apart from this, when a user submits a solution that produces a wrong answer, there is very little help that the system provides to help the user comprehend his error. As a result, the comment sections of most problems are filled with hundreds of comments pleading "My code is not working. Please tell me the error". And although most websites provide tutorials to explain the concept, they don't provide tutorial problems aimed at teaching the user. They just bamboozle him with a flurry of question.
- The reward mechanism of these sites work on basis of ease of the problem. Higher the number of users who have solved a particular problem, lower is the reward for the problem. But this mechanism has a basic flaw which discourages a beginner which gets less rewarded for solving a problem.
- Among these most of the students are learning programming language for the first time. Generally the learning curve is very steep for first time learners of these languages. Hence, a practicing cum implementing cum learning cum competition platform would be best for beginners as well as advanced coders.

# Proposed Treatment

- In order to tackle the above problems, a well-designed system that includes features like better suggestions and error statements should be provided whenever the user submits an erroneous code. In order to keep the user glued to the above, a special recommendation feature should be incorporated, which recommends the proper set of questions based on the user experience i.e. based on the quality and quantity of the problems solved by the user.
- Whenever a user uploads an erroneous code, he/she should be provided with detailed list of errors that he/she has committed. By doing so the user would get to know how to approach the problem.
- A better layout and user interface should be implemented so that the user interest increases. Ranking system for users should be based on the number of solved problems. This provides a competitive edge for normal users to perform better.
- There should be availability of contest holding features with complete control to handlers for customized settings.
- The problems should be categorized according to various classes of algorithms e.g. Dynamic Programming, Greedy Algorithms, etc. This allows the user to choose an area of his choice and work on it. It allows the users to develop his skill sets incrementally. If a user, say, wants to prepare for an exam which will contain only D.P. questions, he should not waste his time trying to figure for problems should of D.P. among a large list of problems.
- Lastly, the ranking system should be such that the user receives equal credits for solving a problem irrespective of whether he is the first or the last user to solve a problem.

Our proposed system aims to achieve the above goals/objectives.

# Plan of Work

An approximate plan for the project development is listed below:

- Feasibility Studies & Technical Proposal
- Requirements Study – produce Software Requirements Specification (SRS), draft user manuals, system test cases and test plans

  *(To be finished by last week of January)*

- Division of project Team into subgroups and work allocation
- Prepare Schematic Design based on the Software Requirements Specification (SRS) – produce design blueprint

  *(To be finished by second week of Feb)*

- Set Code Guidelines, Platform convention, coding standards for the project.
- Implementation of various individual modules – produce working modules (unit tested)

  *(To be finished by third week of March)*

- System and Integration Testing – produce test reports.

  *(To be finished by first week of April).*

- Working System to be submitted for evaluation.

The project team will be meeting on an on-going basis throughout the project to discuss various aspects of the project.

# References


- Sphere Online Judge ( http://www.spoj.pl )
- Codechef (http://www.codechef.com )
- Wikipedia (http://www.wikipedia.org )