Software Requirements Specifications

Requested by: Mr. James Matthews
Professor
Siena College
Computer Science Department

Dr. Scott Vandenberg
Head of Department
Siena College
Computer Science Department

Programming Contest Submission and Scoreboard

SEG

Prepared by:

Paul Califano, Team Leader
T.J Hyne
Adam Pasquerella
George Reese
Mark St. Hilare
Melissa Hoffmann

Presentation:
November 3, 2004
8:15 am – Roger Bacon 328

Programming Contest Submission and Scoreboard
Software Requirements Specification

Table of Contents

Section 1: Product Overview and Summary

Annually the Siena College Computer Science Department hosts a programming contest for local High Schools. During this contest High School teams are placed in different rooms and given problems to solve and then submit to a panel of judges. The judges then determine if the solution given is correct and a message is sent back to the students stating if the solution was accepted or not. This process used to be done in a web-based application that was very sloppy and very tedious to administer. Communication between teams and judges was done by email and teams could not re-submit incorrect solutions if a judge had opened up a previous submission.

The Program Contest Submission and Scoreboard will allow for smooth communication between teams and judges without dealing with email. Also teams and judges will use respective GUIs in order to go through the processes that they need to go through when submitting a solution, clarifying a question, or sending a response. They will also be an up-to-date scoreboard, which will update automatically showing rankings of the teams based on number of problems they answered correctly and the time they have spent.

Section 2: Development, Operating and Maintenance Environments

This system will be developed on the workstations n the Siena College Software Engineering Lab. The data stores will be directories on the Turing server, all web page development and maintenance will be done using DreamWeaver, and the software to produce the Graphical User Interfaces has yet to be determined (either Visual Basic or Java).

The teams and the judges will primarily use the Programming Contest Submission and Scoreboard software. In addition, outside users could view the website to view past problems, history of the contest, and the scoreboard for the current contest. The teams will use a set of GUIs to submit solutions, ask questions to judges, and view the scoreboard. The judges will also have a set of GUIs to reply to the teams for either a clarification or a response to a solution. In order to see what solutions haven't been graded, the judges will view a queue that will have a list of the files to be graded. Everyone with access to the Internet will be able to view the scoreboard of the current contest. This program will be able to do this as well as have "hooks" to allow expansion in the future.

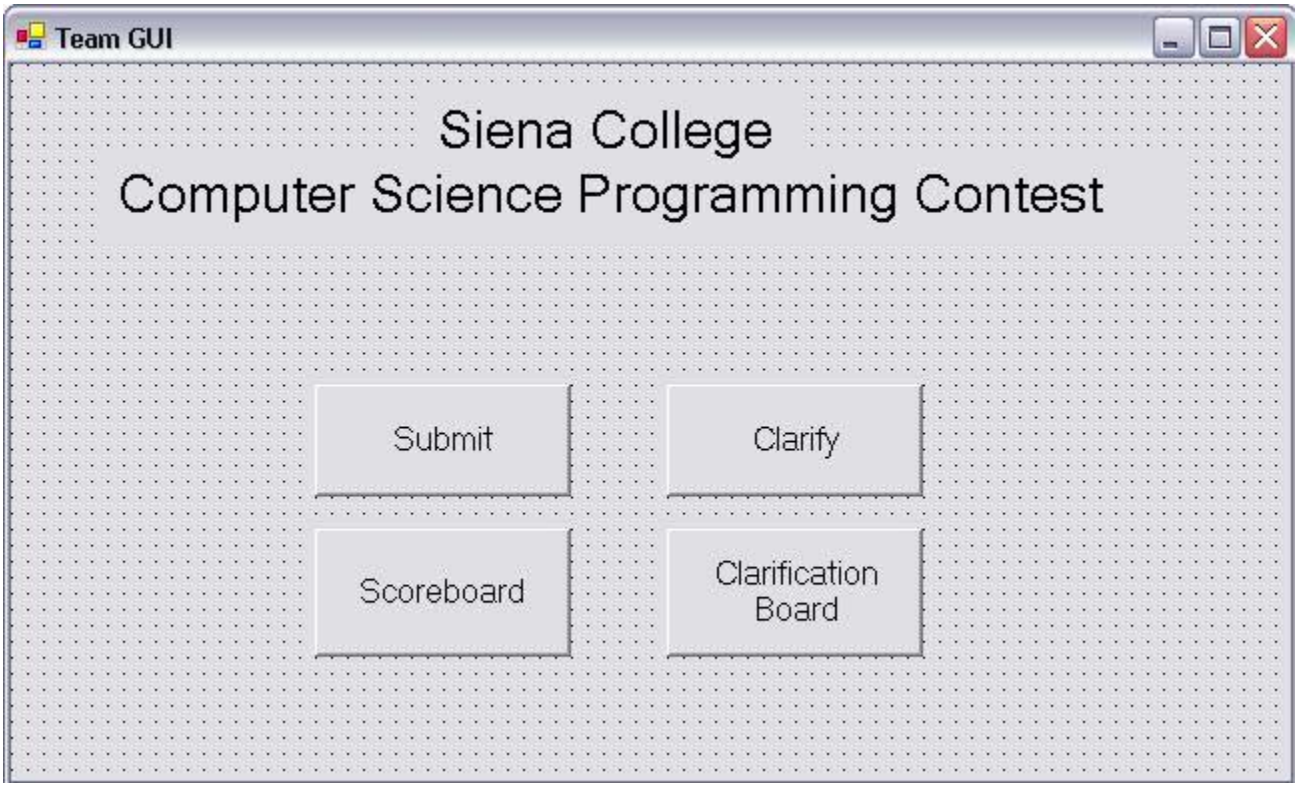
Section 3: External Interfaces and Data Flows

In this section, the prototypes for the screens, on both the team end and the judge end, will be shown and described. There will also be Data Flow Diagrams (DFD's), which will describe the flow of data as it goes through the program, from one user to another, and move from input to output.

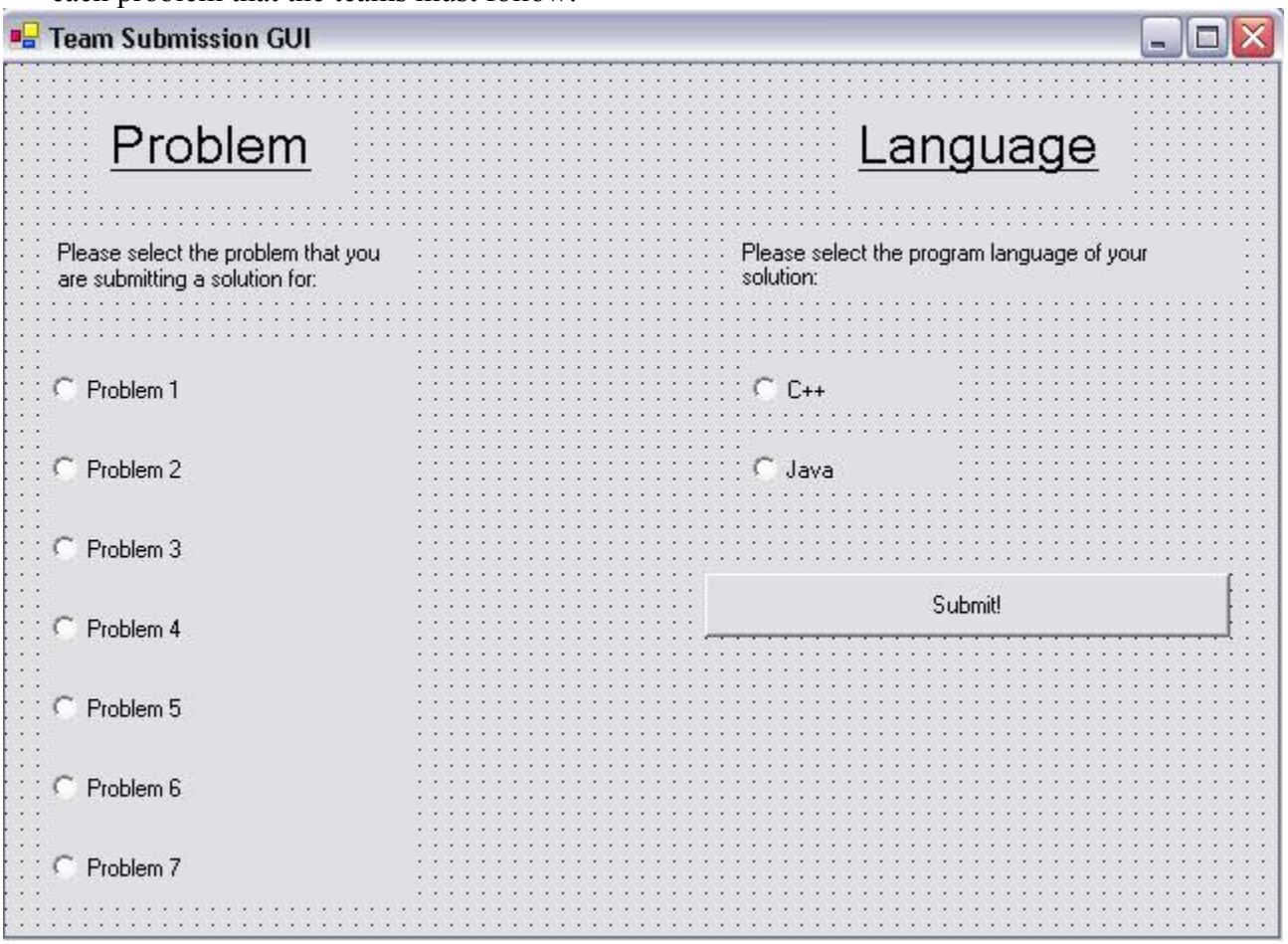
Prototypes:

Teams:

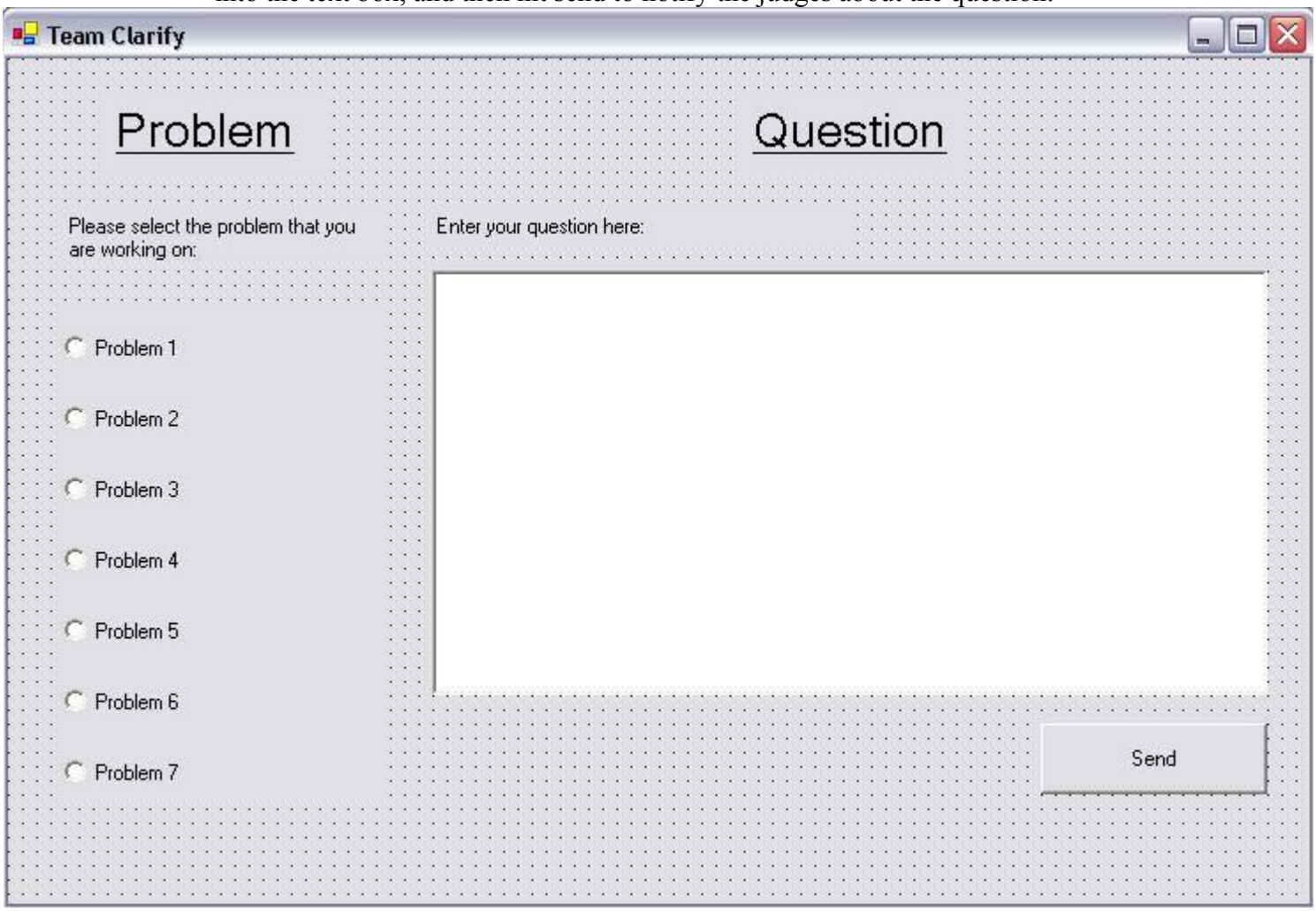
- Team Home GUI – This is the GUI that will pop-up on the teams screen when they log into the system. From this screen, teams to get to other GUI's, which allow them to submit solutions and questions to judges, as well as allowing them to get to the scoreboard and the message board of clarified questions:



- Team Submission GUI – This is the team's submission GUI that allows them to submit a solution to the judges. Teams must select the problem that they are working on, as well as selecting the language the solution was developed with. When the teams hit the submit button, the file they are submitting will automatically be copied to the judges directory, called "Submitted not Judges", on the Turing server from the teams directory, based on the naming convention for each problem that the teams must follow:

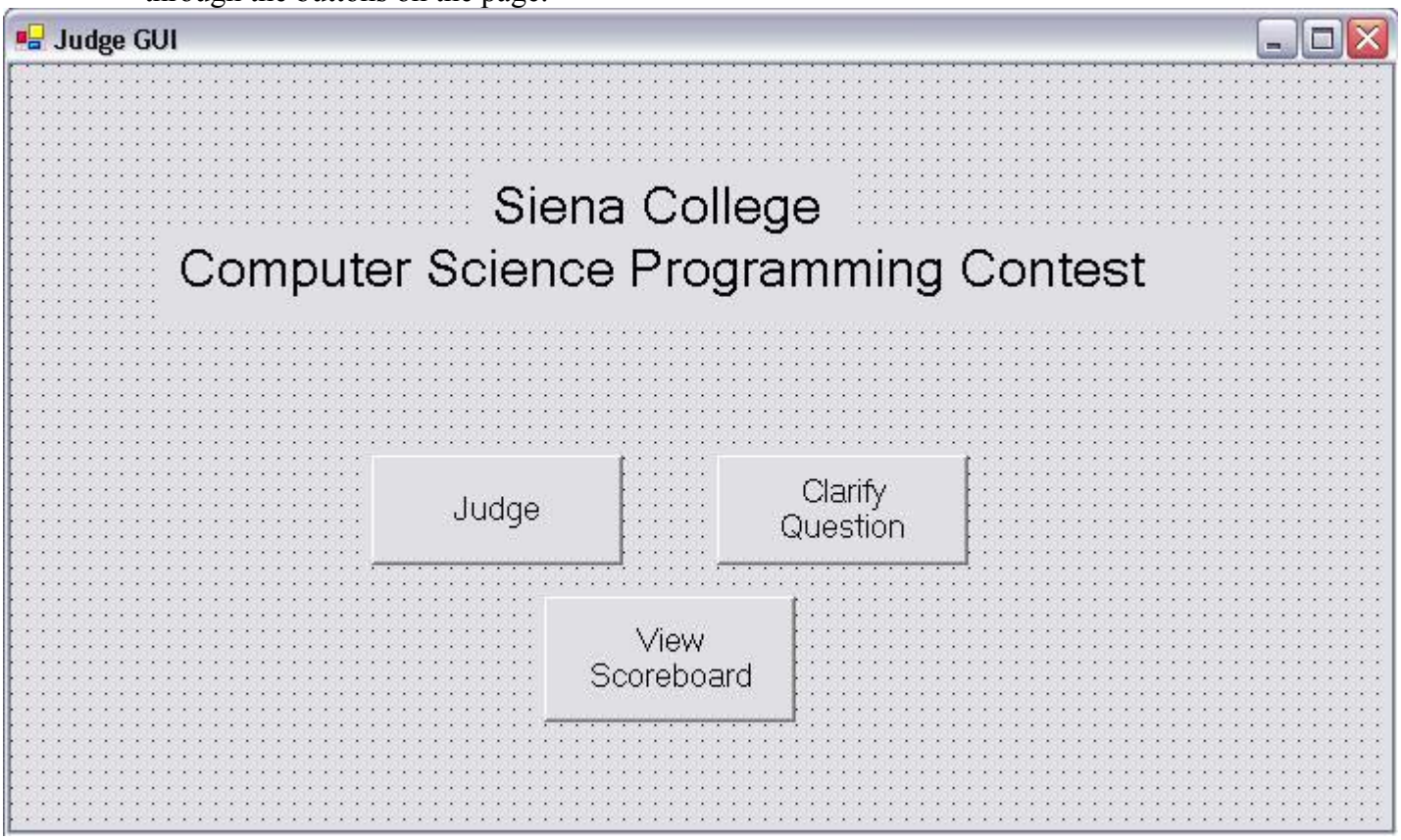


- Team Clarification GUI – This GUI will allow the teams to submit any questions to the judges, if they have for a problem they are working on. The teams must select the problem they would like to ask a question about, enter their question into the text box, and then hit send to notify the judges about the question:

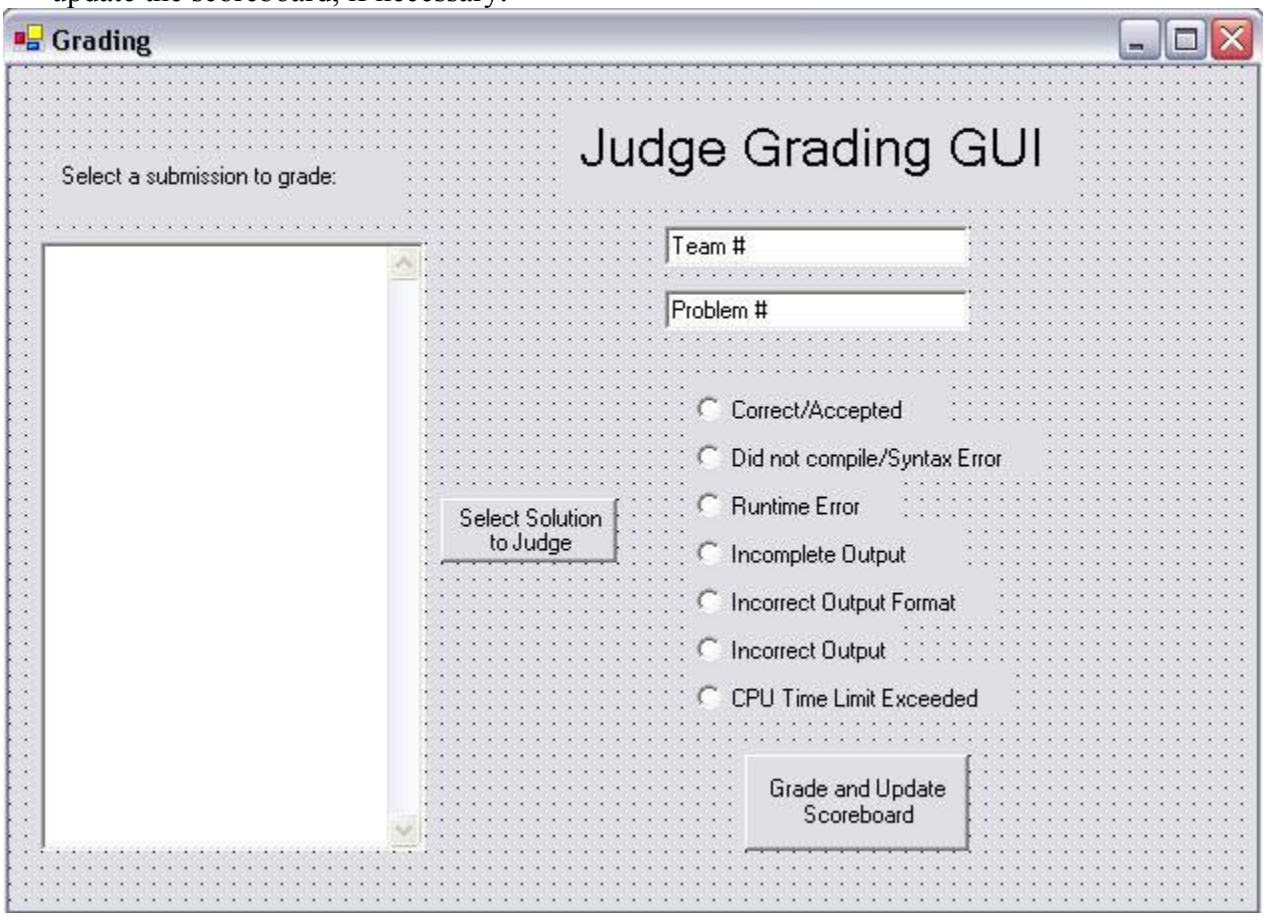


Judges:

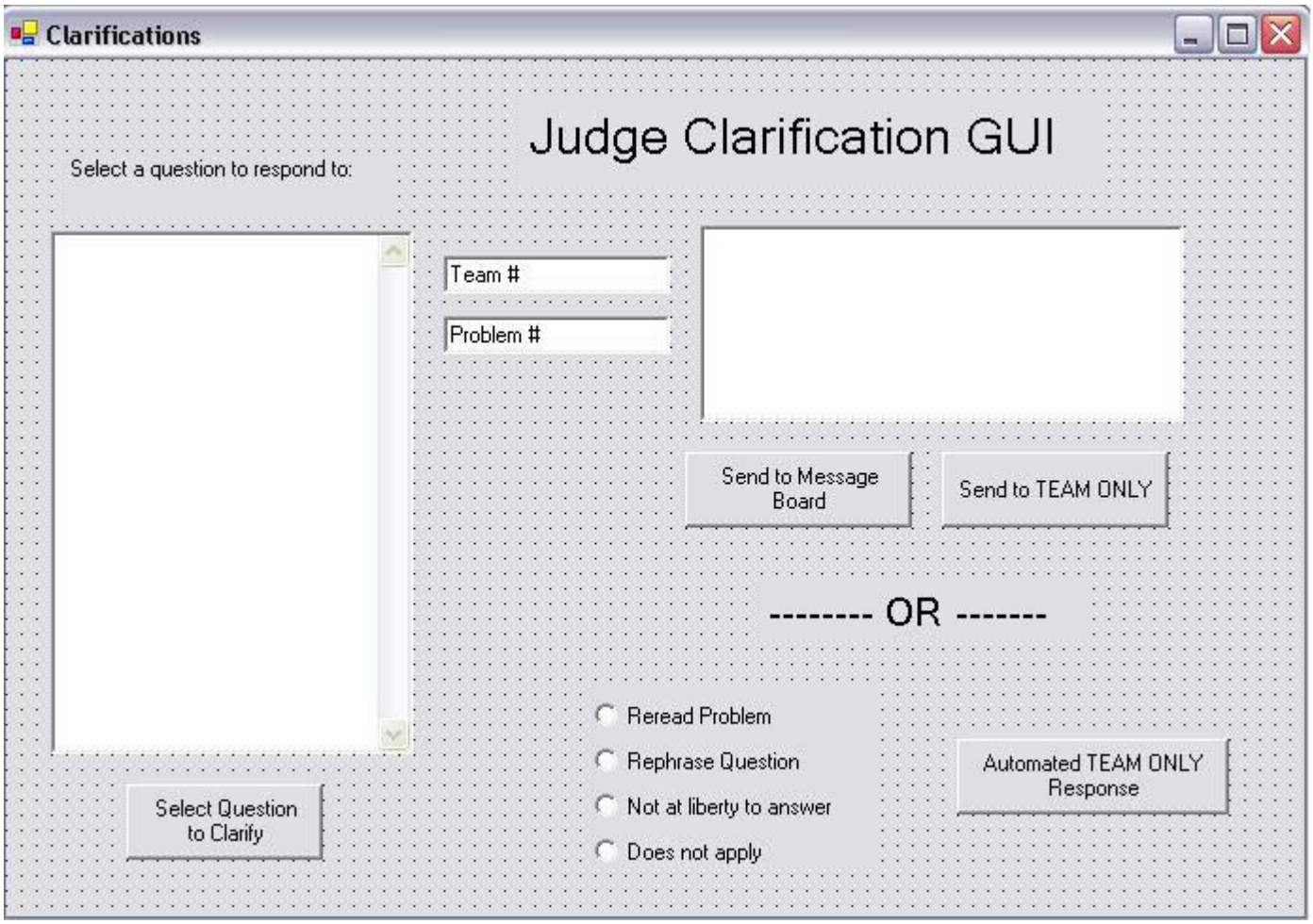
- Judge Home GUI – This GUI automatically pops-up when the judges log into the system and allows them to access three things, the Judge Grading GUI, Judge Clarification GUI, and the Scoreboard. Judges can access these other items through the buttons on the page.



- The Judge Grading GUI has a queue on the left side, which is the list of submissions that have not been graded at that time. Judges highlight one, and click the "Select Submission to Judge" button, which moves the file from the "Submitted not Judged" directory to the "Submitted Being Judged" directory, as well as putting the team number and problem number into the two text boxes automatically. Once the judge is finished grading the solution submitted by the team, they will select one of the seven options as responses to the teams about their submission, and click "Grade and Update Scoreboard" button to send a message to the teams. This will also move the file graded from the "Submitted Being Judged" directory to the "Correct" or "Incorrect" directory, on Turing, and update the scoreboard, if necessary.

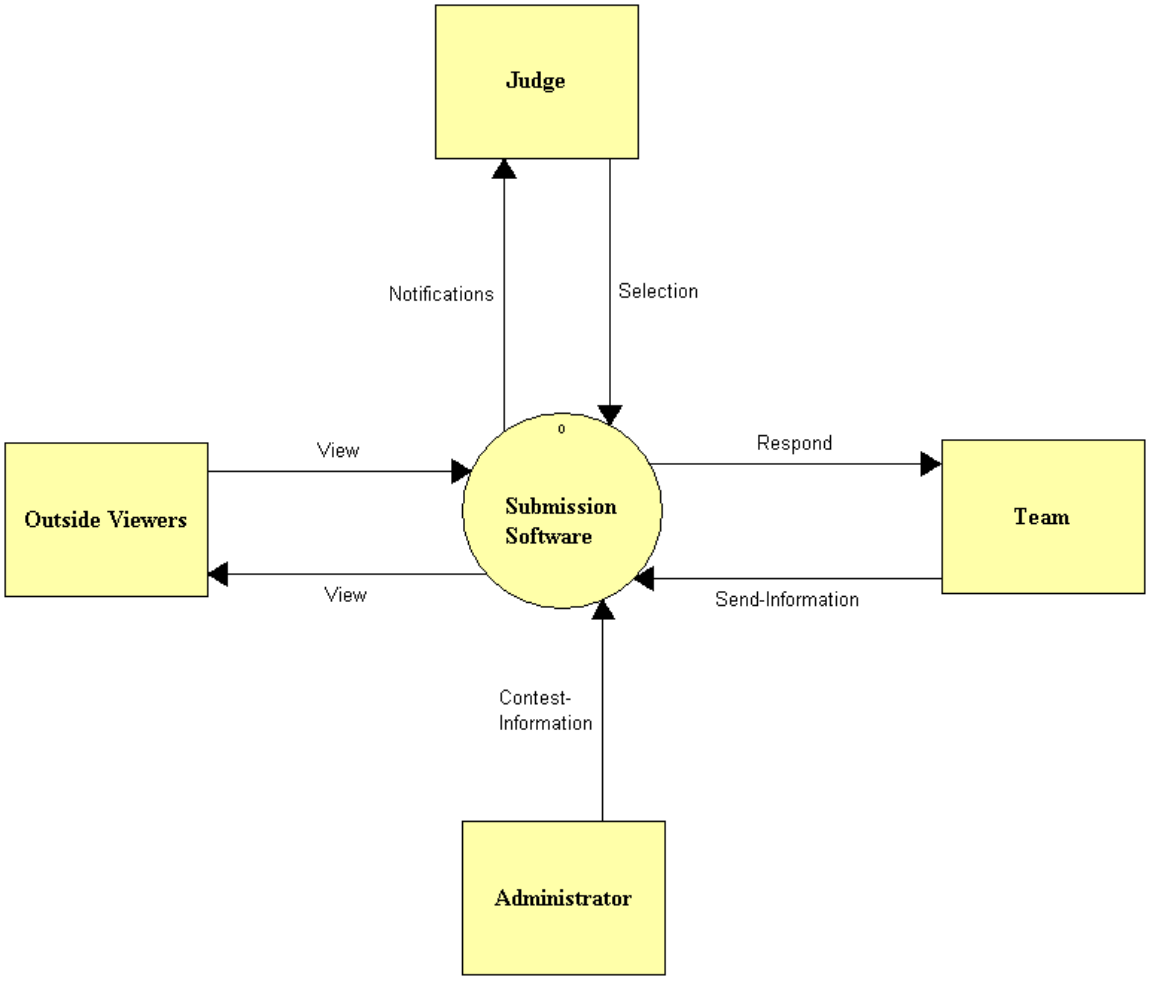


- Judge Clarification GUI – This GUI also has a queue on the left side of questions that have been submitted by the teams about specific problems. The judge selects a question to clarify by highlighting the question in the queue, and hitting the "Select Question to Clarify" button, which takes it out of all the judges' queues. Once they hit the button, the team number and problem number are placed into the text boxes to the right of the queue. When answering the teams' question, they may decide to send the response to the team only, or send it to all the teams that want to see it by sending it to a message board. If they chose to post the responses to the message board, they have to type in the response they would like to post, then click the "Sent to Message Board" button. If they decide to post just to the team that posed the question, they may enter a text response and select the "Sent to TEAM ONLY" button, or chose one of the automated responses and select the "Automated TEAM ONLY Response" button. Both of the responses that go just to the team sends a message to the team with the judge's response.

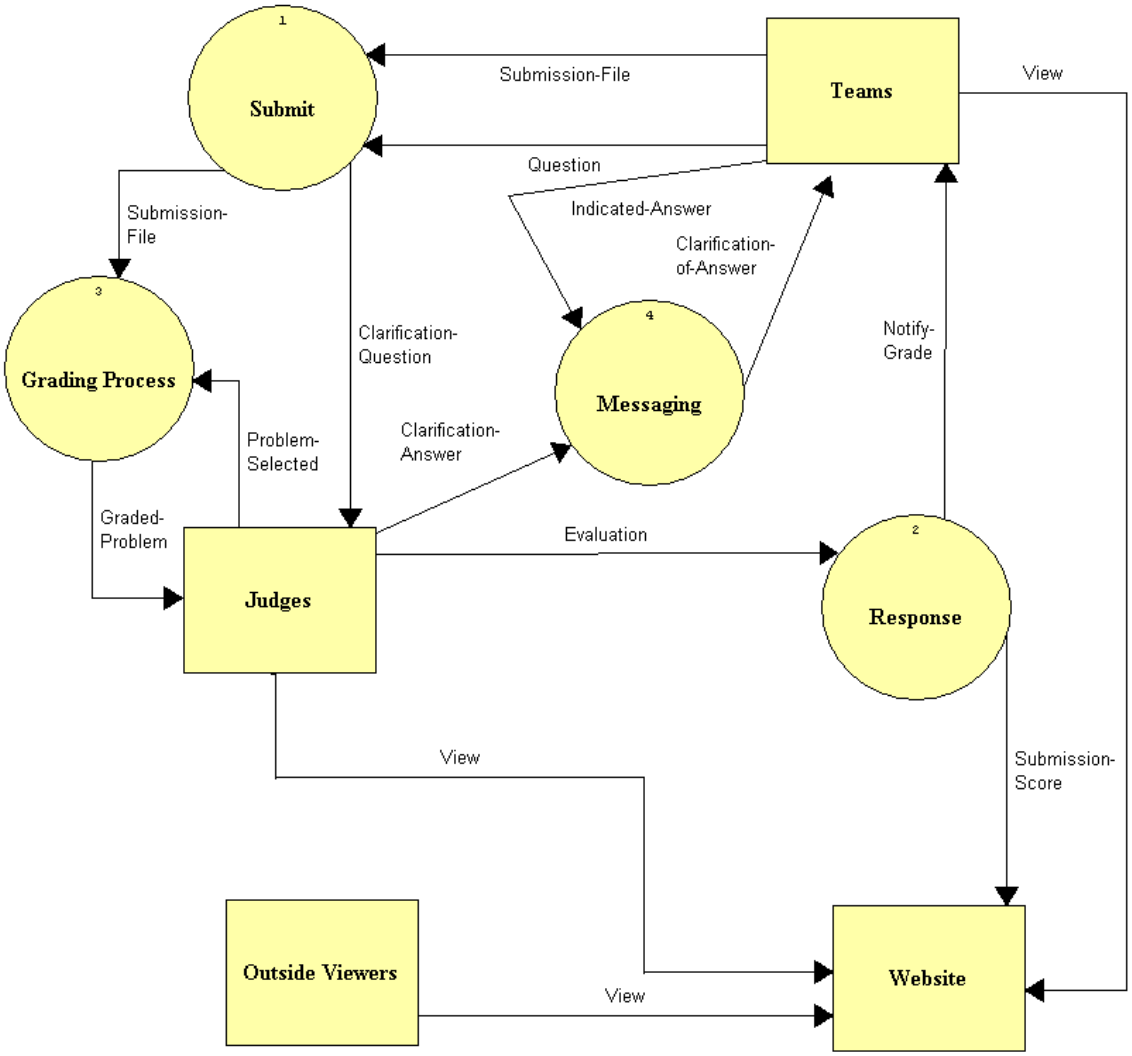


DFD's:

Context Diagram:

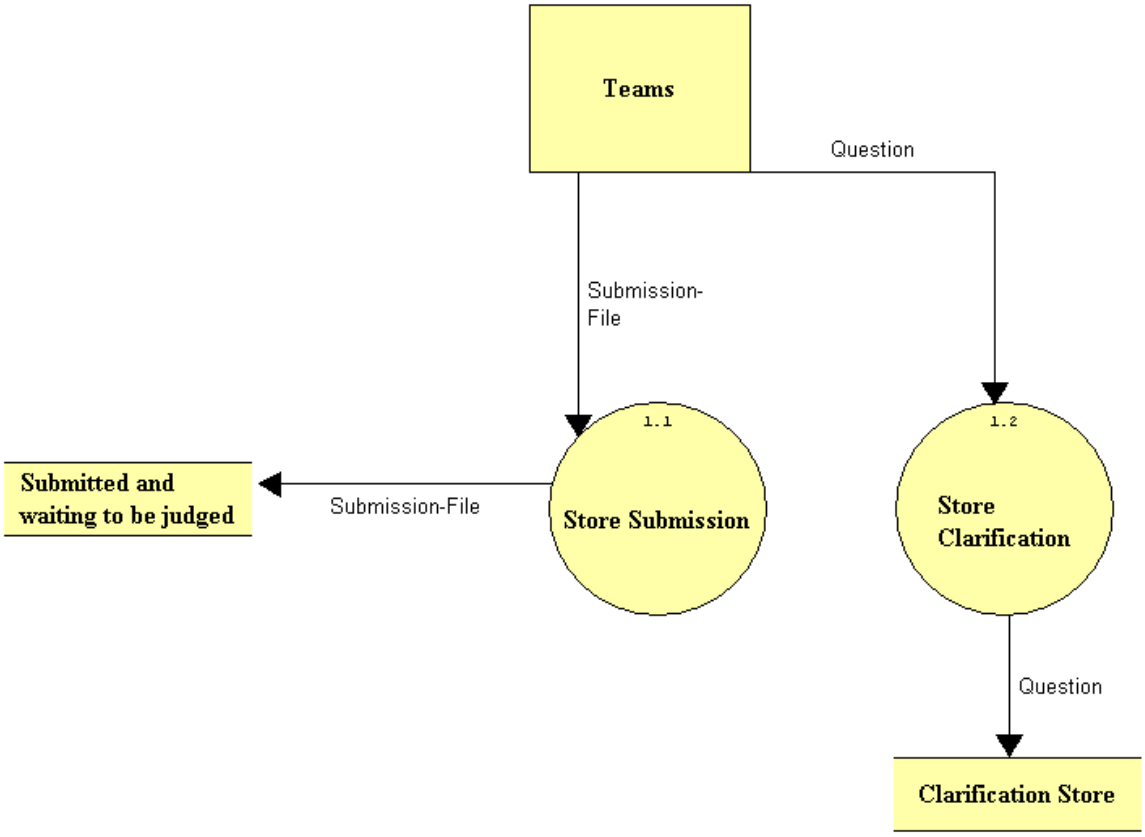


Level 0:

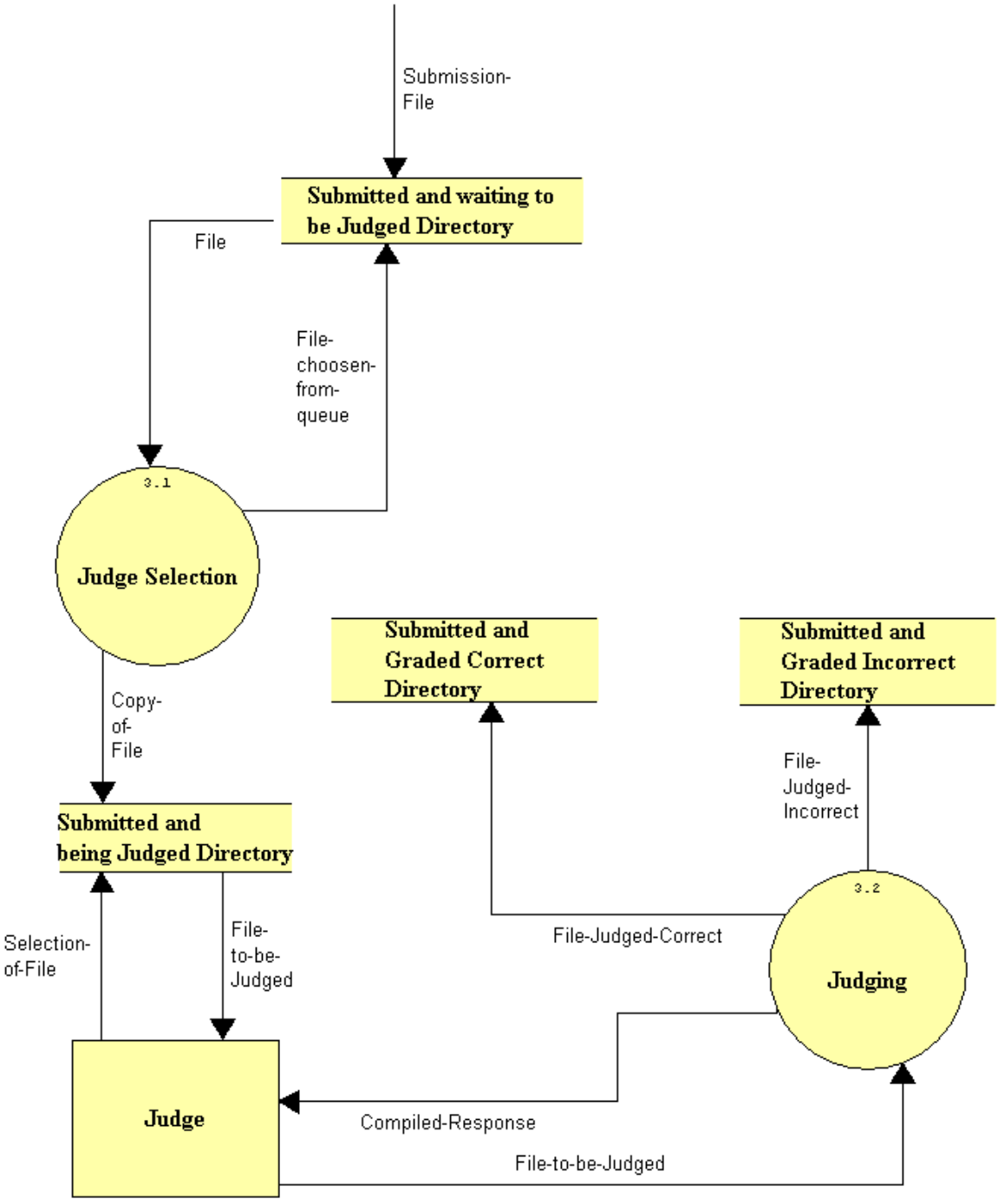

1
Submit
Submission-File
Teams
View
Question
Indicated-Answer
Clarification-of-Answer
Submission-File
3
Grading Process
Clarification-Question
4
Messaging
Notify-Grade
Problem-Selected
Clarification-Answer
Graded-Problem
Judges
Evaluation
2
Response
View
Submission-Score
Outside Viewers
View
Website

Level 1:

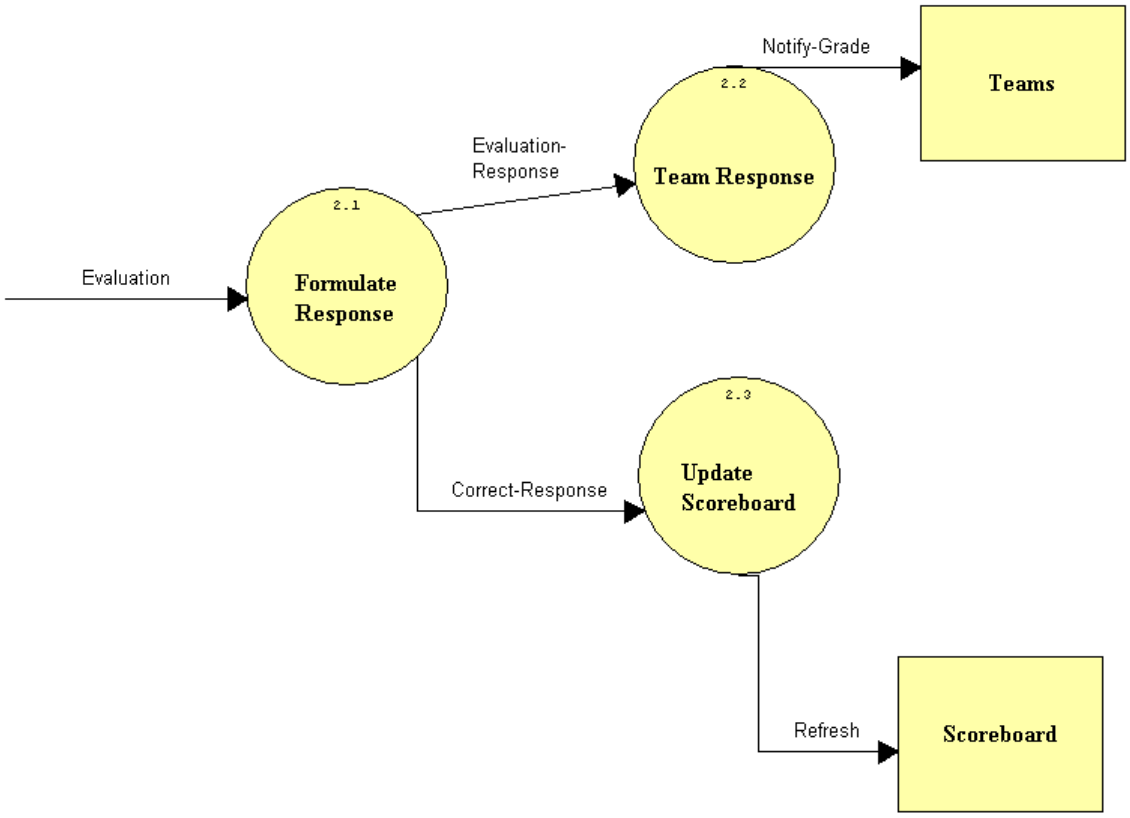
Submission



Grading:

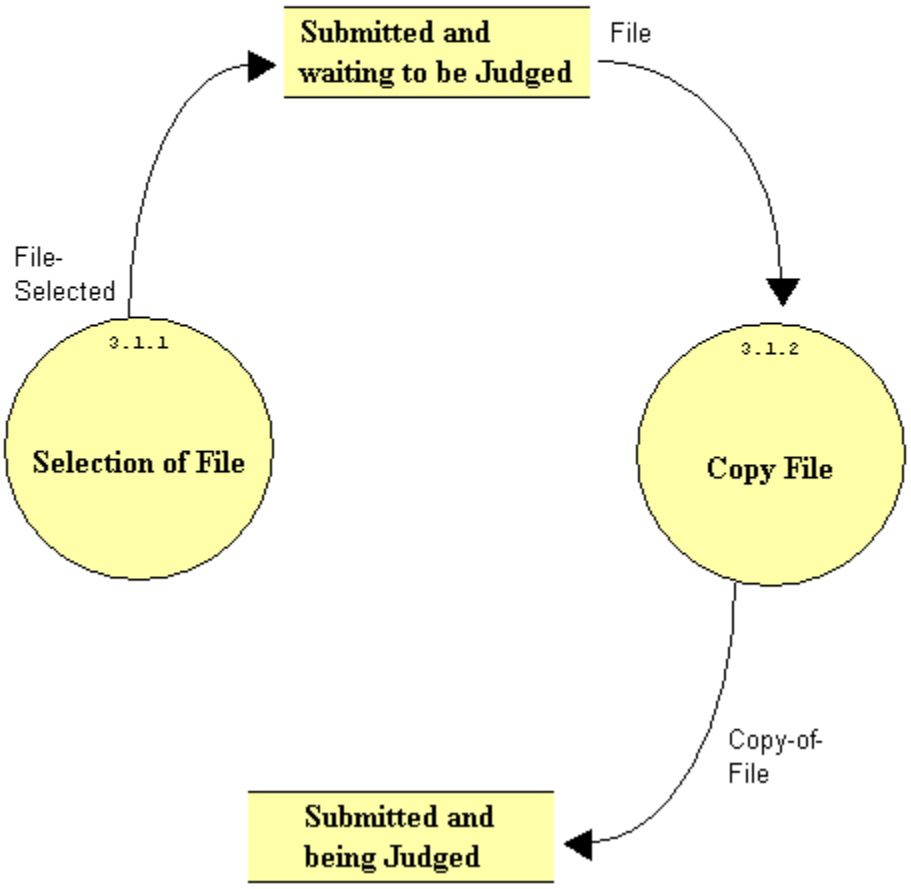

Submission-File
Submitted and waiting to be Judged Directory
File
File-choosen-from-queue
3.1
Judge Selection
Copy-of-File
Submitted and being Judged Directory
Selection-of-File
File-to-be-Judged
Judge
Submitted and Graded Correct Directory
Submitted and Graded Incorrect Directory
File-Judged-Incorrect
File-Judged-Correct
3.2
Judging
Compiled-Response
File-to-be-Judged

Response:


Notify-Grade
Teams
2.2
Team Response
Evaluation-
Response
2.1
Formulate
Response
Evaluation
2.3
Update
Scoreboard
Correct-Response
Refresh
Scoreboard

Level 2:

Judge Selection



Section 4: Functional Requirements

The Programming Contest Submission and Scoreboard will serve four types of users, teams in the contest, judges facilitating the contest, outside users, and an administrator. The following is a list of functionality requirements that the system will perform for each user:

- Teams in the Contest:
    1. A Graphical User Interface will be displayed when the teams log into the system when the contest begins that has buttons to the Submission GUI, Clarification GUI, Team Status bar, Scoreboard, and Message Board.
    2. The Submission GUI that will allow teams to select the problem they are submitting, the language in which it was produced, and the name of the file to be submitted, all of which are sent to the Judges. When the teams submit the problem, the file being submitted will be copied onto a directory called "Submitted but not Judged", on Turing.
    3. The Clarification GUI will allow teams to select a problem and enter a question into a text box to be sent to the Judges.
    4. The Message Board will be a place where Teams can see the questions asked to the Judges by other teams, with the Judges clarification.
    5. The Scoreboard allows each team to see the standings of the contest, based on the number of problems that have been answered correctly and the cumulative about of time used.

- Judges of the Contest
    1. A Graphical User Interface will be displayed when the judges log into the system. On the GUI will be buttons to the Judging GUI, Judge Clarification GUI, and the Scoreboard.
    2. Once the Judging GUI pops-up, there will be a Queue with non-graded files submitted by the teams, with buttons on the side. Once a Judge selects a file, it is moved into a second directory on Turing, "Submitted being Judges" so no other judges can try and select this file. Once a judge is ready to respond to the submission, the Judge GUI will have a section to select a certain response to the teams based on their solution. Once a judge submits his/her response, which will be sent to the team, and the file will be moved into the "Correct" directory if the solution is correct or "Incorrect" directory if not. Their response will also update the scoreboard.
    3. The Judges Clarification GUI will have a Judges Clarification Queue, which consists of all the questions that have been asked by the teams. Once the judges select the question they would like to respond to, they would have a choice to send a response to the question just to the team that asked the question, or put it onto a Message Board so that all teams may view the question and response. There will also be a text box

where the judges can type their response to be submitted, as well as automated responses that can be used when responding to just the team that asked the question.

4. Judges will be able to view a Web-based Scoreboard with up-to-date standings of the teams based on the number of questions answered correctly and the cumulative time to answer them.

- Outside Users (including teams and judges before the contest)
    1. Outside users can view a website that will contain history of the contest, old contest problems, and how you sign up to join the contest.
    2. A Web-based Scoreboard will be created to allow any individual, in the contest or not, to view the standings of a contest at it is being run. Teams will be sorted based on their ranks, which are determined by the number of questions answered correctly and the cumulative about of time to answer those questions.

- Administrator:
    1. The Administrator must update all of the GUI's to make sure all of the correct information is there, including anything new for an upcoming contest (number of problems, programming language, judges responses, etc.).
    2. Update the website with more problems from previous contest and new winners from the contests.
    3. Insert the names of the teams on the scoreboard of an upcoming contest.
    4. Make sure all of the correct information is displayed on the message board and the scoreboard.
    5. Make sure the copying and moving of the files from directory to directory is being done correctly on Turing. This should be tested before the contest begins, as well as being monitored during the contest.

There will be no databases needed for the Programming Contest Submission and Scoreboard. However, files submitted by the teams will be stored in directories on the Turing Server.

Section 5: Performance Requirements

The Programming Contest Submission Software will be able run on Windows, especially the Windows XP Operating System, since that is the operating system run in the Computer Science Department, which is where the contest is held.

The Programming Contest Website and Scoreboard will be accessible through Microsoft Internet Explorer 6.0 and Netscape Navigator 7.1.

The Programming Contest Submission Software and Scoreboard will be designed to be viewable at 1024 by 768-screen resolution. It will be viewed easily on a computer monitor, and a classroom projector.

Section 6: Exception Handling

The Programming Contest Submissions and Scoreboard will be designed through GUIs that will not have any user options to choose when viewing. However, when using the GUIs there will be different options in actions to take when using the software. If users close any of the GUIs, which are automatically popped up when they log into the system, there will be a systematic way to reproduce those windows. Users may choose Internet options when viewing the website, message board, and scoreboard. Testing will also be done to make sure files are copied and moved correctly when it is meant to happen. All of this will be verified through team testing.

Section7: Early Subsets and Implementation Priorities

Some of the essential components of the system are:

- Teams will have a set of GUIs, which will allow them to submit solutions, ask questions, and view a message board as well as the scoreboard.
- Judges will have a set of GUIs, which will allow them to view, submitted solutions and questions, respond to submitted problems and questions as well as view the scoreboard.
- A website will be set up to hold the history of the contest, background of the contest, and how teams can register for the contest. On the site will also be links to problems from previous contests as well as a link to a scoreboard of up-to-date standings.
- A web-based message board, which will allow teams to view questions asked by other teams and the responses by the judges. Only questions and responses posted on the message board will be at the discretion of the judges.
- A web-based scoreboard with automatically updates when the judges accept a teams solution. The teams will be ranked on the scoreboard based on the number of problems they have answered correctly, as well as the amount of time used to come up with solutions to these problems.

Section 8: Foreseeable Modifications and Enhancements

The Programming Contest Submission and Scoreboard will be developed in a way that will allow for future advancements. These future advancements could be, but are not limited to:

- A Team Status bar, which allows teams to see their progress on a certain problem (“Not Submitted”, “Submitted Not Judges”, “Being Judged”, “Correct”, “Incorrect”).
- To let judges run their testing script as soon as they chose a submission to judge rather than having them go into a Unix section and typing everything themselves.

Section 9: Acceptance Criteria

The Programming Contest Submission and Scoreboard will allow each user to complete the following tasks within the application:

- Teams to:
    1. Select which GUI they would like to use, or view the scoreboard, by clicking a button on the initial GUI, which automatically pops-up;
    2. Submit a solution to the judges, for one particular problem, and also indicating which programming language it was developed with. Teams will be notified to use a specific naming convention when naming their solutions, which will allow the submission application to find the file and copy it into the judge's directory;
    3. Ask a question to the judges about any problem by selecting the problem they have a question about, and entering text describing their inquiry;
    4. View a message board of all question asked by other teams, as well as the judges response, which should be ordered by the problem number;
    5. View the scoreboard of the contest they are competing in, which ranks teams based on number of problems answered correctly and the amount of time used to produce their solutions.
    6. View the website with history of the contest, previous problems, and previous winners, before the day of the contest;
    7. Open any GUI that was accidentally closed, without losing any time.

- Judges to:
    1. Select a solution to grade from a queue, check the correctness of the solution, and then send an automated response based on whether the solution was correct or not. When they send their response back to the team, the scoreboard will be automatically updated;
    2. Select an inquiry from a team to clarify from a queue and answer the question. When answering the question, the judges may send an automated response, or type in text to respond the just the team that asked the question, or post it to the message board to allow it to be viewed by every team;
    3. View the scoreboard to see the rankings of the team, as well as make sure all the information is correct;
    4. Open any GUI that was accidentally closed, without losing any time.
- Out Side Users to:
    1. View the scoreboard of the current contest;
    2. View the website with history of the contest, previous problems, and previous winners.

- Administrator to:

1. Make sure that the right information being posted to the message board and the scoreboard is accurate.
2. View files in the specific directories on the Turing machine, making sure they are being copied and moved correctly throughout the process.
3. Update the GUI's with anything new (new judge responses, number of problems in the contest, programming languages to be used, etc).
4. Update the website with previous problems from prior contests as well as update the winners on the website.
5. Insert new names of the teams on the scoreboard.

Section 10: Testing Requirements

Testing for this system will be done in the Computer Science Department. Testers will simulate creating solutions to problems and use the GUI to submit it to judges. They will also simulate teams asking questions to judges, viewing the scoreboard and message board, and test the functionality to reopen windows if need be.

Other testers will simulate the judges in sending responses to the teams based on solutions submitted and questions submitted. They will also test to make sure they can view the scoreboard and open up any closed windows if need be.

Administrative functionality and communication functionality between the teams and the judges, as well as communication from the judges to the scoreboard to make updates will be also tested.

Section 11: Design Hints and Guidelines

The Programming Contest Submission and Scoreboard is a GUI-based communication program between teams and judges of the contest. The teams will be able to use a number of GUI's to submit solutions to problems, as well as ask any questions they have about any of the problems. Teams will also be able to view a message board of questions asked by other teams, and the responses given by the judges, if the judges feel it is necessary to allow all teams to view the clarification of the questions. The judges will also have a number of GUI's, which will be used to reply to questions about the different problems or to respond to teams about the solutions they have submitted.

The teams and judges, as well as any outside user will be able to view the scoreboard of the current contest. The scoreboard will automatically update when judges respond to teams about their solutions they have submitted, and it will list teams based on their ranking, which is determined by the number of questions answered and how much time was used to answer the questions.

A website will also be set up, which will allow teams entered into the contest, and teams interested in entering, background on the contest, previously used problems,

and past winners of the contest. This website may also be viewed by anyone else in the world.

Appendices:
Appendix A: Glossary of Terms

Computer Science Programming Contest – A programming contest administered by the Computer Science Department for local high school teams to come to Siena and compete.

DFD – Data Flow Diagram – A graphical representation that depicts information flow and the transformations that occur as data moves from input to output.

Directories – Folders on Turing in which files will be stored, as well as being moved from and to them.

DreamWeaver – A program that will be used to develop all needed websites.

Gantt Chart - A chart that depicts progress in relation to time, often used in planning and tracking a project.

GUI – Graphical User Interface – The screens that users will see while using the Programming Contest Submission Software.

HTML - A markup language used to structure text and multimedia documents and to set up hypertext links between documents, used extensively on the World Wide Web.

Internet - An interconnected system of networks that connects computers around the world via the TCP/IP protocol.

Java- Programming language that our program will possibly be built in.

Linear Sequential Model / Classic Waterfall Model – A systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing, and support.

Scoreboard – A web-based list of teams, in ranked order, based on the number of problems each team has answered correctly, and the time used to answer the questions.

Turing – The Unix Server that every PC in the Computer Science Department Connects to. It is under the administration of Mr. Ken Swarner.

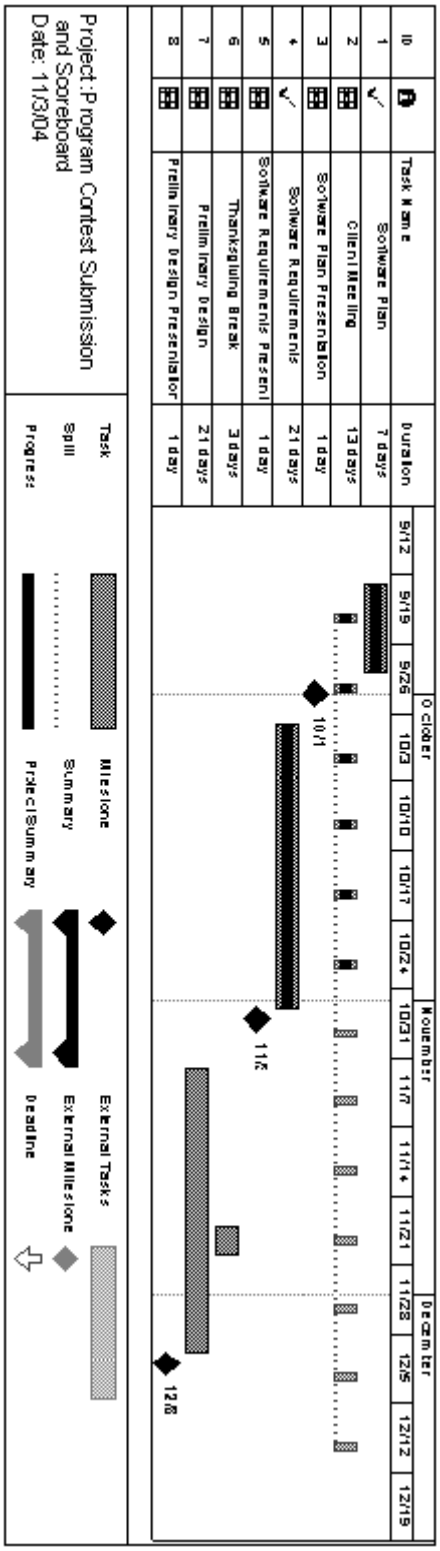
VB- Visual Basic - Programming language that our program will possibly be built in.

Web-based – A software that uses the World Wide Web on the Internet as a user interface

Appendix B: Sources of Information

The information for this section was obtained from meetings with our clients, Mr. James Matthews and Dr. Scott Vandenberg. Other sources of information include Dr. Lederman's class lectures, the Software Engineering class textbook *Software Engineering: A Practitioner's Approach* by Roger S. Pressman and previous Software Engineering teams' projects, located at http://www.cs.siena.edu/%7Elederman/csis410/csis410.html.

Appendix C: Gantt chart



Appendix D:

The following are the Data Flow Elements, which are incorporated with the DFD's above in Section 3.