

SLICE

Software **L**inked **I**nteractive **C**ompetitive **E**nvironment **P**reliminary **D**esign

Requested by:

Dr. Darren Lim

Associate Professor

Siena College

Loudonville, NY

Prepared by:

P=NP*solutions*

Zachary Fitzsimmons – Team Leader

Michael Pepe – Team Lieutenant

Anthony Parente – Systems Administrator

Matthew Ferritto – Webmaster

Renee Solheim – Document Analyst

December 6th, 2011

Contents

- 1 SLICE Preliminary Design 1**
- 1.1 Product Overview 1
- 1.2 User Case Narratives 1
 - 1.2.1 Participant User Case Narrative 1
 - 1.2.2 Team Advisor User Case Narrative 2
 - 1.2.3 Judges User Case Narrative 2
 - 1.2.4 Administrative User Case Narrative 3
- 1.3 UML Diagrams 4
 - 1.3.1 UML – Use Case Diagram 4
 - 1.3.2 Diagram Legend 4
 - 1.3.3 UML Use Case Diagram for SLICE 5
 - 1.3.4 UML – Deployment Diagram 6
 - 1.3.5 UML – Activity Diagram 7
- 1.4 SLICE Website Map 10
- 1.5 Functional Requirements Inventory 11
 - 1.5.1 Participant User 11
 - 1.5.2 Team Advisor User 11
 - 1.5.3 Judge User 12
 - 1.5.4 Administrative User 12
- 1.6 Non-Functional Requirements 13
- 1.7 Data Flow Diagrams 14
 - 1.7.1 Diagram Legend 14
 - 1.7.2 Context Diagram 16
 - 1.7.3 Level 0 17
 - 1.7.4 Level 1 Diagrams 18
 - 1.7.5 Level 2 Diagrams 25
 - 1.7.6 Level 3 Diagrams 28
- 1.8 Logical Data Dictionary 29
- 1.9 Prototype Screens 30
 - 1.9.1 General View 30
 - 1.9.2 Administrative User Views 31
 - 1.9.3 Judge User Views 37

1.9.4	Participant User Views	41
1.10	Testing Plan	46
1.10.1	Overview & Strategy	46
1.10.2	Acceptance Criteria	46
1.10.3	Unit Tests	46
1.10.4	Test Cases	47
1.11	Development and Production Environment	47
Appendix		48
A Glossary of Terms and Gantt Chart		49

Chapter 1

SLICE Preliminary Design

1.1 Product Overview

Competitive programming contests are held at all different levels and scales, with a few teams to up to many participants and from single network sessions to across the world through the Internet. Currently Dr. Darren Lim holds a programming contest at Siena College for local area high school students. In the Spring of 2013, Siena College will be holding the Consortium for Computer Science in Small Colleges Northeast (CCSCNE), which includes a programming contest. The current software is not user friendly and consists of several different parts. The goal of our software, SLICE, is to allow fast, both manual and automatic judging during a competitive programming contest and to integrate problem submission, judging, and scoreboard of a contest into a single application.

1.2 User Case Narratives

1.2.1 Participant User Case Narrative

Participants will each be able to log into SLICE with a provided username and password generated by the Administrative User and provided in a hardcopy sealed envelope. After a Participant logs in, the Participant will be able to view the contest scoreboard, submit problem-solutions or submit clarification-requests. For a problem-solution submission, a Participant will select one of the contest's problems as the associated problem and one of a list of programming languages for the language. For clarification-requests a Participant can make a general request or associate it with a problem. Participants receive a response from a Judge concerning the submission, stating one of the following: Correct Solution, Compiler Error, Running Time Error, Wrong Answer or Time-Limit Exceeded. Participants will also be able to receive both individual responses to clarification-requests from any Judge as well as publically broadcasted messages from any Judge or the Administrative User.

1.2.2 Team Advisor User Case Narrative

Team Advisors will be able to log into SLICE to view the problems and scoreboard. Team Advisors accounts will be created by the Administrative User in the same way that Participant Users are created. A Team Advisor's username and password to access SLICE and is provided to each Team Advisor in a hardcopy sealed envelope. The username and password for the Team Advisor account will be generated by the Administrative User. In addition to viewing the problems and scoreboard, Team Advisors will receive all publically broadcasted contest clarifications.

1.2.3 Judges User Case Narrative

Judges will each have the ability to log onto SLICE with a username and password generated by the Administrative User and provided to each Judge in a hardcopy sealed envelope. Once logged in, a Judge will be able to view Participant submitted solutions that the system will compile. If a submission compiles, a Judge will then run the submitted programs using test input provided in SLICE that corresponds to the associated question. A Judge then submits a response to the Participant stating one of the following: Correct Solution, Compiler Error, Running Time Error, Wrong Answer or Time-Limit Exceeded. If the submission was a correct solution to the corresponding problem, a Judge with extra permissions in SLICE to update the scoreboard, if there are errors present. A Judge will be able to either respond (in general or concerning a specific problem) to a specific participant or all Participants and Team Advisors by using a messaging prompt.

1.2.4 Administrative User Case Narrative

The Administrative User will be able to log into SLICE with a unique username and password. Once logged in, the Administrative User will be able to create and manage the Judge, Participant, and Team Advisor accounts. The permissions for one Judge to have Scoreboard Judge ability is also managed by the Administrative User. The Administrative user can view the scoreboard, but only the Scoreboard Judge can edit the scoreboard. The usernames and passwords for each individual account is put into a hardcopy sealed envelope and distributed to the associated Judges, Participant or Team advisors. The Administrative User will be able to send broadcast messages to all users, or any subset i.e. all Judges, the Scoring Judge, all Participants or all Team Advisors. The Administrative User will manage the set of problems, the test input, and correct output for each contest problem. The Administrative User will also be in charge of the length of the test contest, contest start and stop times and the allowed programming languages.

1.3 UML Diagrams

1.3.1 UML – Use Case Diagram

UML (Unified Modeling Language) Use Case Diagrams present a graphical overview of the functionality for a system. It uses Actors (human and non-human) to represent the users, and their goals, represented by Uses, as well as any dependencies or inheritance between them, represented by specific lines or arrows. The diagram shows what system functions are the performed by each actor.

1.3.2 Diagram Legend

We adhere to the UML standard for our UML Use Case Diagrams, but to better explain our diagram we describe each symbol below:



System Boundary: Where the users, human and nonhuman, located on the outside of the boundary, interact with the uses, located on the inside of the boundary, within the system

Left Side Actor: Human users which interact with the uses within the system

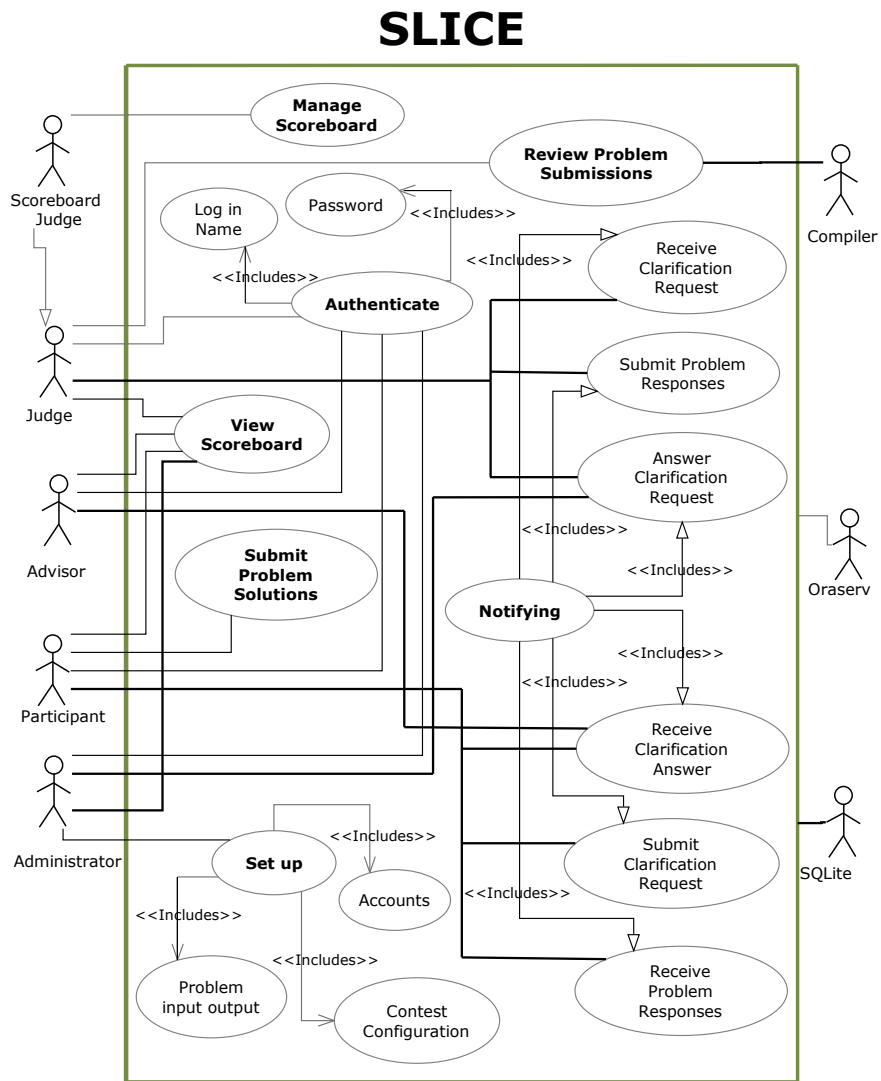
Right Side Actor: Non-Human users which interact with the uses within the system

Uses: The Activities that interact with the actors outside of the system

Includes Arrow: Specific items that are included within the connected

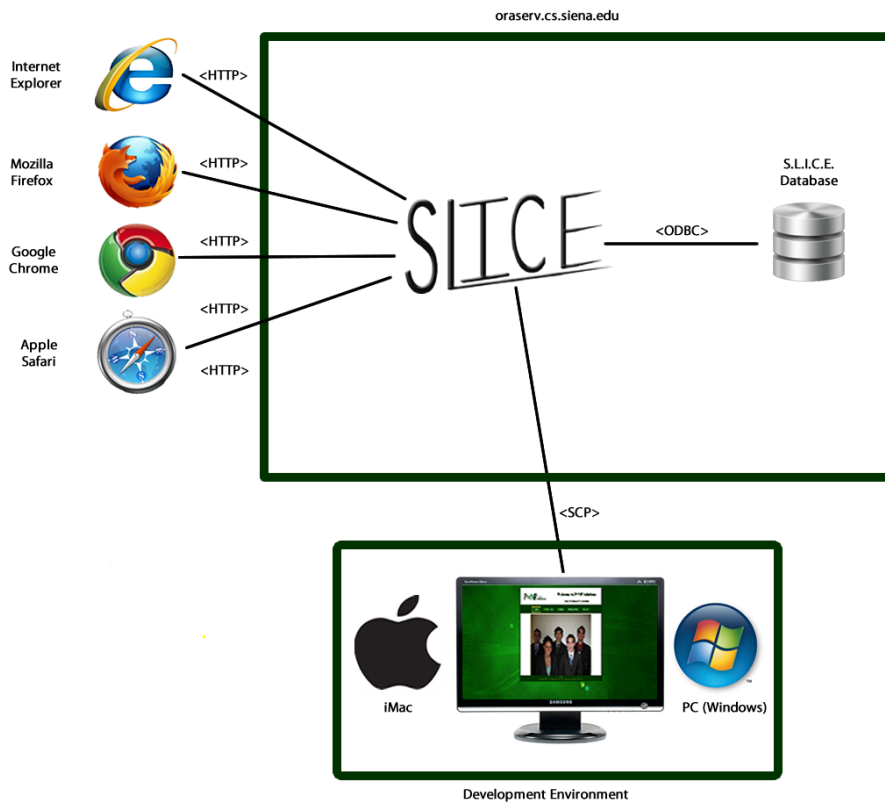
Inherits Arrow: Shows subuses of the given uses or actors which are not necessarily accessed when the parent is accessed

1.3.3 UML Use Case Diagram for SLICE



1.3.4 UML – Deployment Diagram

Deployment Diagrams are a Unified Modeling Language (UML) diagram used to show how different development and execution environment interact in a system. The following diagram shows the physical layout of SLICE.



1.3.5 UML – Activity Diagram

Activity Diagrams are a type of flowchart that represents the workflow of the components in our system, SLICE. It shows the overall flow of control and sequences of actions.

Legend

We adhere to the UML standard for our UML Activity Diagrams, but to better explain our diagrams we describe each symbol below:



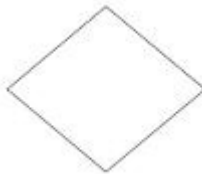
Start Node: The starting point for the described activity



End Node: The end point for the described activity



Activity Node: An activity at a certain step in the described activity



Decision: A decision point for the described activity



Object: An object used in the activity



Entity Connector: The connection used between nodes, decisions, objects and joins



Join: Joins several connections together

Diagrams

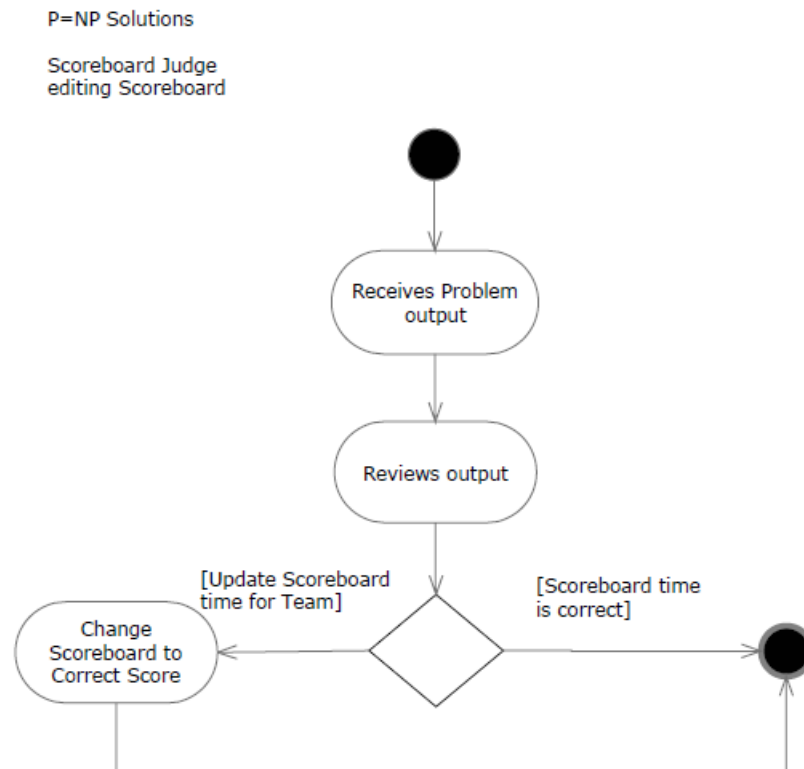


Figure 1.1: Activity Diagram for Editing the Scoreboard

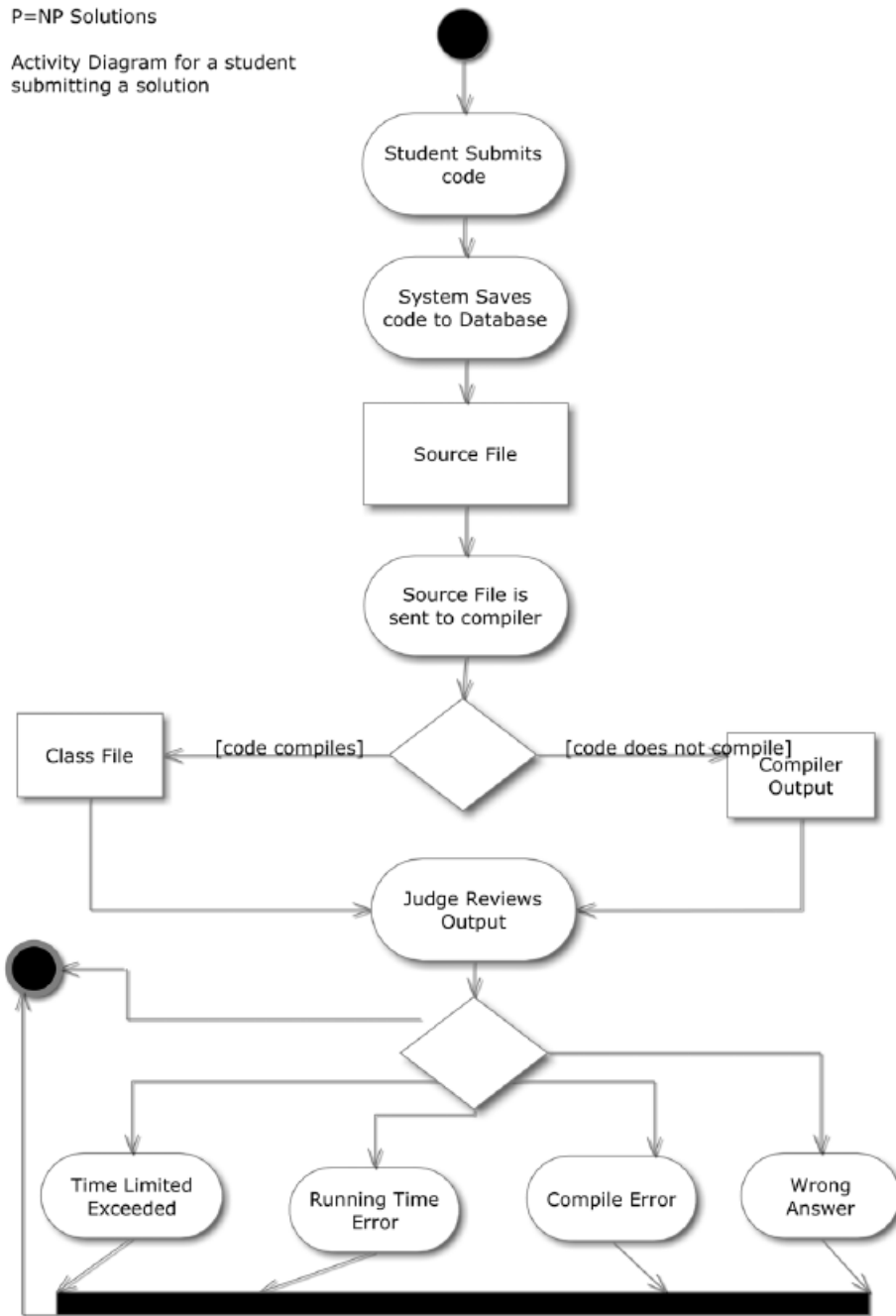
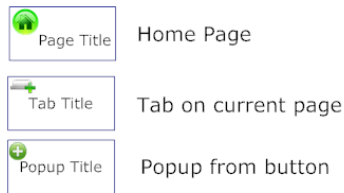


Figure 1.2: Activity Diagram for Problem Submissions

1.4 SLICE Website Map

The Website Map illustrates how each user navigates the different pages and views of the website in SLICE.

Legend:



→ Directed Relation between pages/views

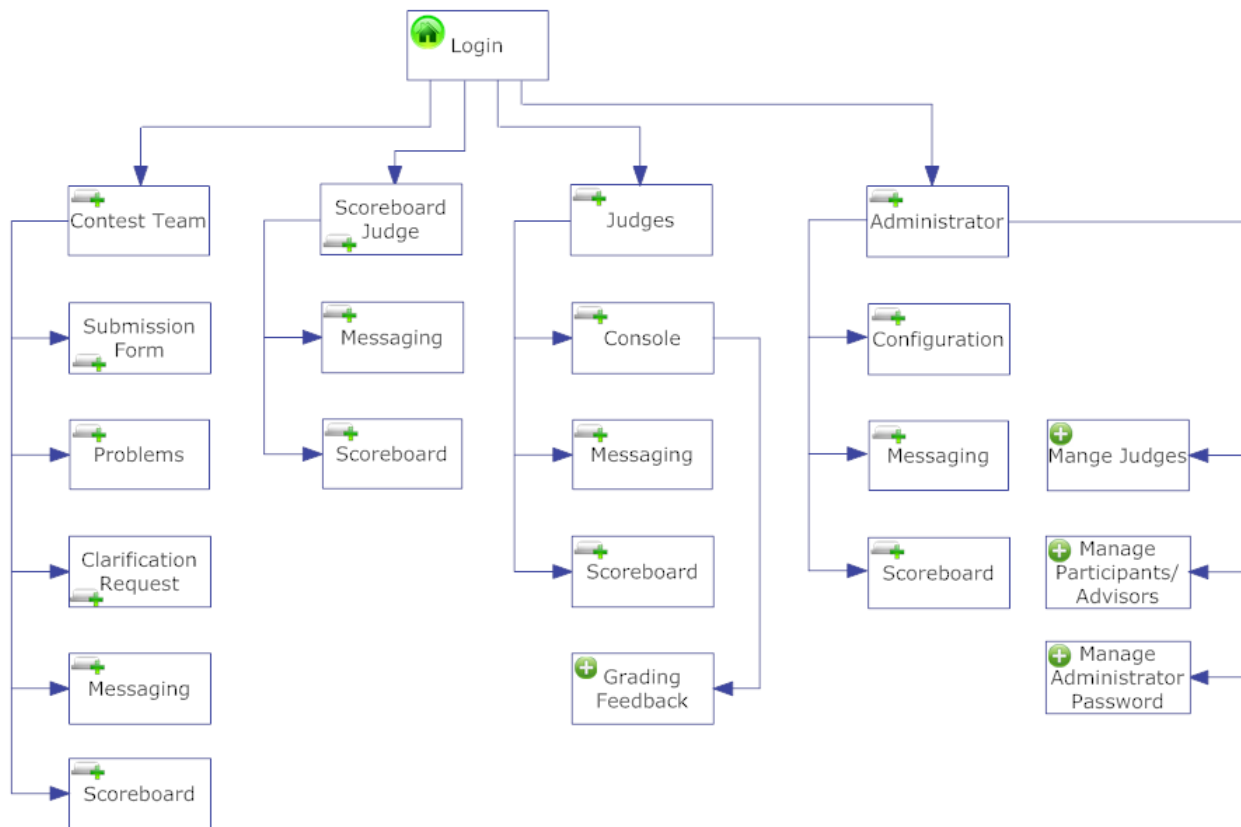


Figure 1.3: SLICE Website Map

1.5 Functional Requirements Inventory

For a general functional requirement, SLICE will be able to run on all popular browsers including, Internet Explorer, Mozilla Firefox, Google Chrome and Safari.

Below is a list of functional requirements for each human user that interacts with the system. One major functional requirement is that SLICE is scalable to up to 30 Participant users.

1.5.1 Participant User

- Will be able to log onto SLICE with a provided username and password while the contest is active
 - An incorrect username and/or password or invalid time period will result in an appropriate error message
- Will be able to log off of SLICE while the contest is active
- Will be automatically logged off of SLICE if the contest becomes inactive
- Will be able to view the contest scoreboard while active
- Will be able to submit problem submission sourcecode
 - Will be able to associate a contest problem with their problem submission
- Will be able to participate in messaging capabilities with the Judges
 - Will be able to send clarification-requests
 - * Will be able to associate a problem or send general clarification-requests
 - Will receive results from problem submissions
- Will be able to receive publically broadcasted messages

1.5.2 Team Advisor User

- Will be able to log onto SLICE with a provided username and password
 - An incorrect username and/or password will result in an appropriate error message
- Will be able to view the contest scoreboard while active
- Will be able to receive all publically broadcasted messages

1.5.3 Judge User

- Will be able to log onto SLICE with a provided username and password
 - An incorrect username and/or password will result in an appropriate error message
- Will be able to log off of SLICE at any point
- Will be able to view the contest scoreboard at any point
- Will be able to review Participant problem submissions
 - Will be able to compile problem submission sourcecode on a central server
 - Will receive compile-time feedback
 - Will be able to execute problem submissions with provided test input
 - Will be able to compare the output to provided test output
- Will be able to participate in messaging capabilities with the Participants
 - Will be able to answer both general and contest problem specific clarification-requests
 - Will be able to give results to the Participant concerning problem submissions
- Will be able to send publically broadcasted messages
- A Judge with extra permissions called the Scoreboard Judge will be able to update the contest scoreboard
- The Scoreboard Judge will be able to receive data consisting of correct submissions with corresponding, problem number, timestamp and Participant

1.5.4 Administrative User

- Will be able to log onto SLICE with a provided username and password
 - An incorrect username and/or password will result in an appropriate error message
- Will be able to log off of SLICE at any point
- Will be able to participate in messaging capabilities with all users or any subset of users

- Will be able to complete the initial contest set-up
 - Will be able to create all other user accounts (Participants, Team Advisors, Judges)
 - Will be able to grant one Judge User, Scoreboard Judge permissions
 - Will be able to list contest problems and associate test input and output
 - Will be able to set contest start and stop times
 - Will be able to set the practice time for the contest
 - Will be able to choose the list of allowed programming languages
 - Will be able to set the time that the contest scoreboard will be active during the contest
 - Will be able to view the scoreboard at any time.

1.6 Non-Functional Requirements

The non-functional requirement inventory is a list of non-functional system requirements. This list is composed of requirements that specify how the system should be. As more information is gathered about the project, this list may be subject to change:

- The system will be easily maintained.
- The system will be stable.
- The system will be viewable on multiple browsers.
- The system will run efficiently.
- The system will be user friendly.

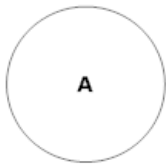
1.7 Data Flow Diagrams

The Data Flow Diagram will show how data flows and is transformed in SLICE. The following diagrams are used as a tool to better illustrate where data is being created, modified and going in SLICE. We will first describe the symbols used in our diagrams and how they are representing the data.

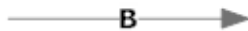
The Context Diagram is then shown, which will give a general overview of each of our data entities that interacts with the system. The Level Diagram shows more details about each process and the entities that it interacts with. Directed arrows illustrate the flow of data in this diagram. We also have a Level 1 Diagram that describes one of the major processes of the system in a greater detail than the Level 0 Diagram and is labeled to a smaller granularity of detail.

1.7.1 Diagram Legend

Data Flow diagrams are a means of representing a system at any level with a graphic network of symbols. Showing processes, data flows, data stores, and external entities. A key and definition for each symbol has been provided below:



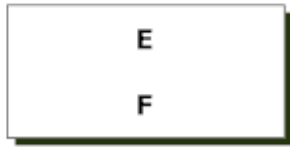
Process: The process symbol represents an activity that transforms or manipulates data. The name of the process is within the circle and represented by the letter A.



Data Flow: The data flow symbol represents the movement of data. Data flows are labeled with the description of the data that is being moved through it. The is represented by the letter B.



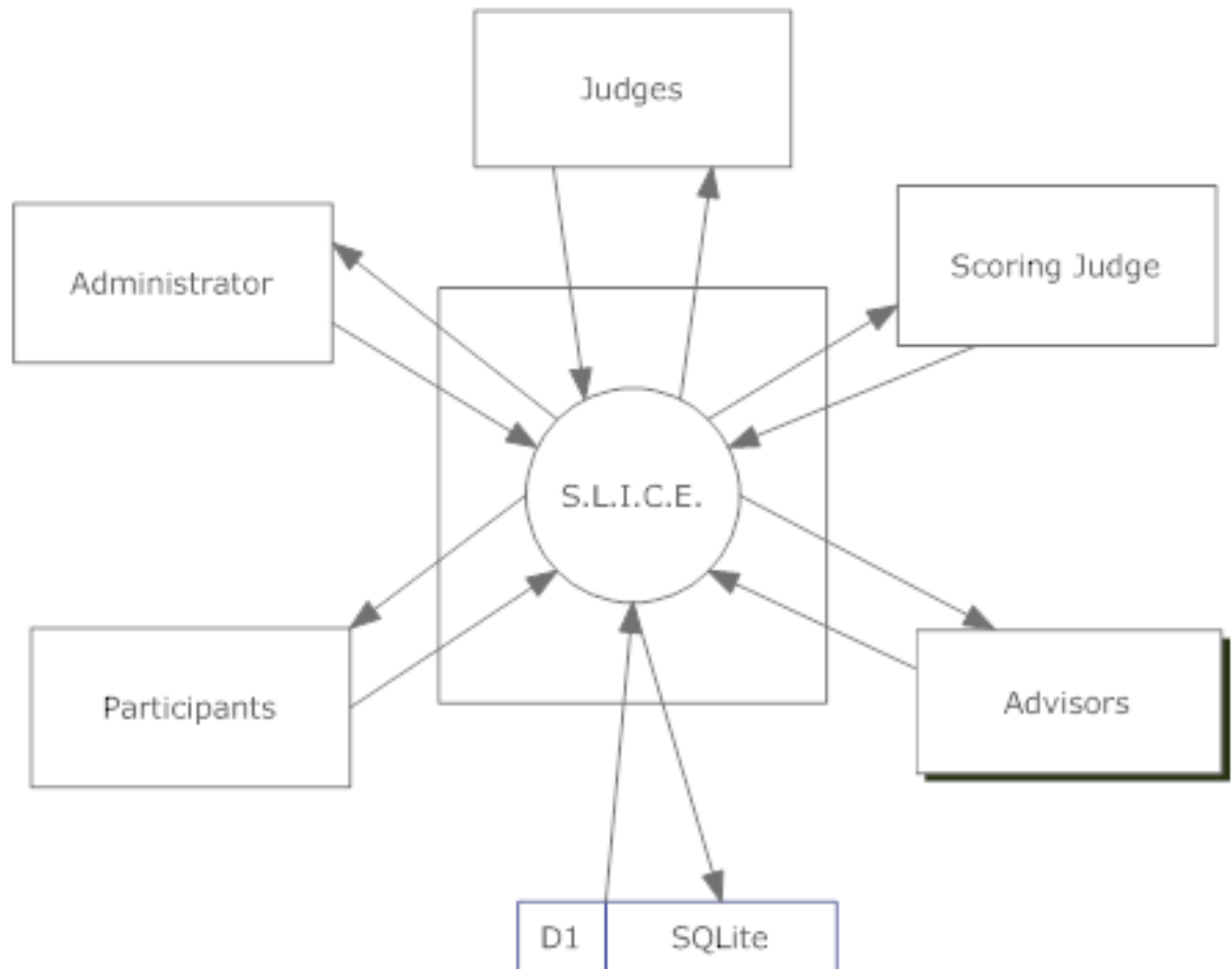
Data Stores: The data store symbol represents some location where data is temporarily held. The letter C represents the data store type and the letter D represents the data store name.



External Entities: External entities are sources/sinks which contribute data or information to the system or which receive data/information from it. The letter E represents the unique identifier for the entity and the letter F represents the name of the entity.

1.7.2 Context Diagram

Figure 1.4: Context Diagram for major data entities



1.7.4 Level 1 Diagrams

Figure 1.6: Level 1 Diagram of Authenticate

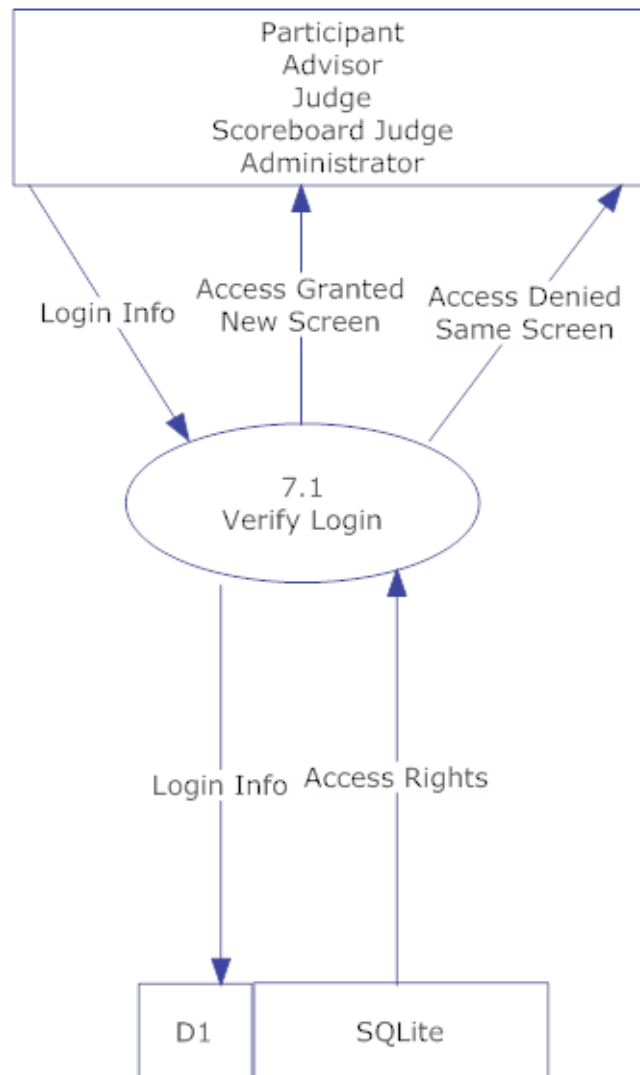


Figure 1.7: Level 1 Diagram of Manage Scoreboard

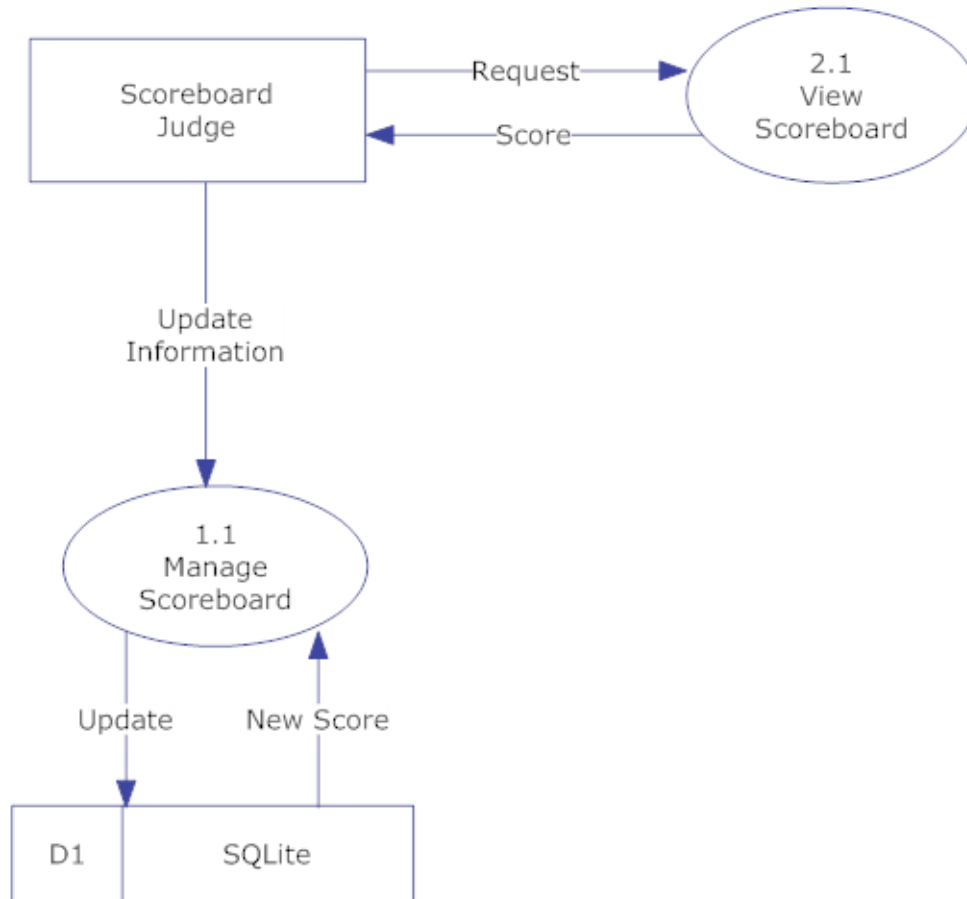


Figure 1.8: Level 1 Diagram of View Scoreboard

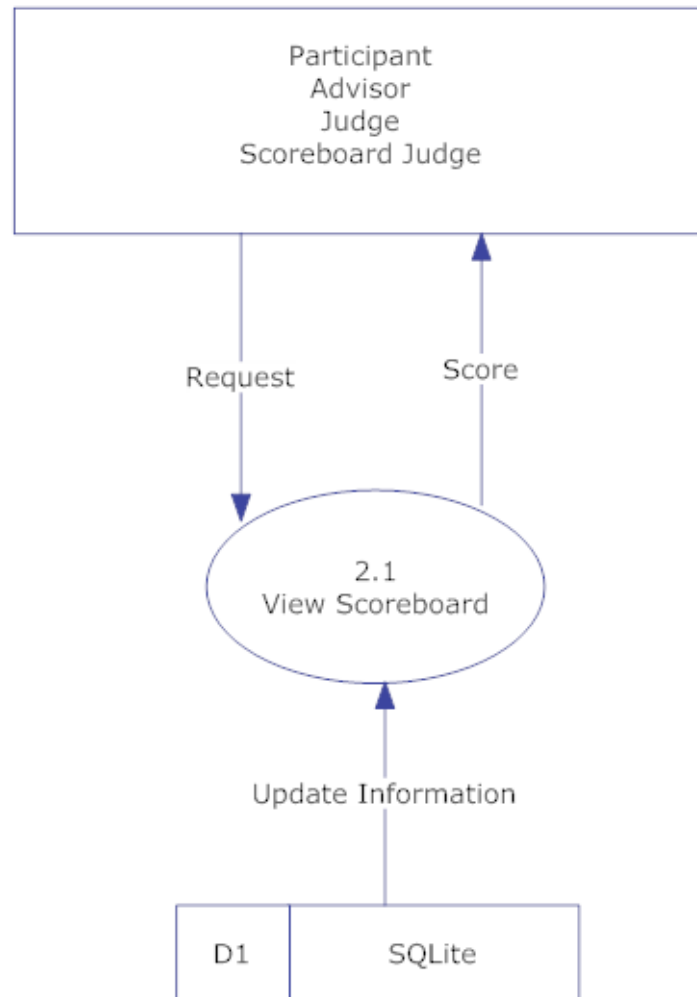


Figure 1.9: Level 1 Diagram of Notify

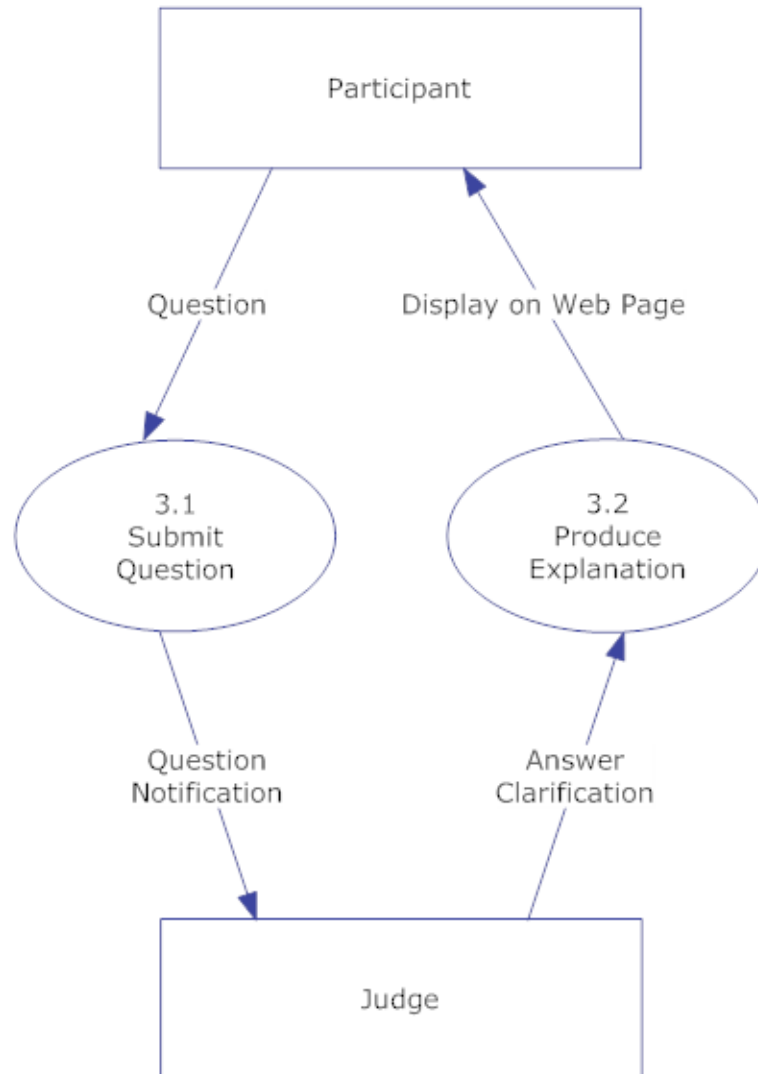


Figure 1.10: Level 1 Diagram of Set Up

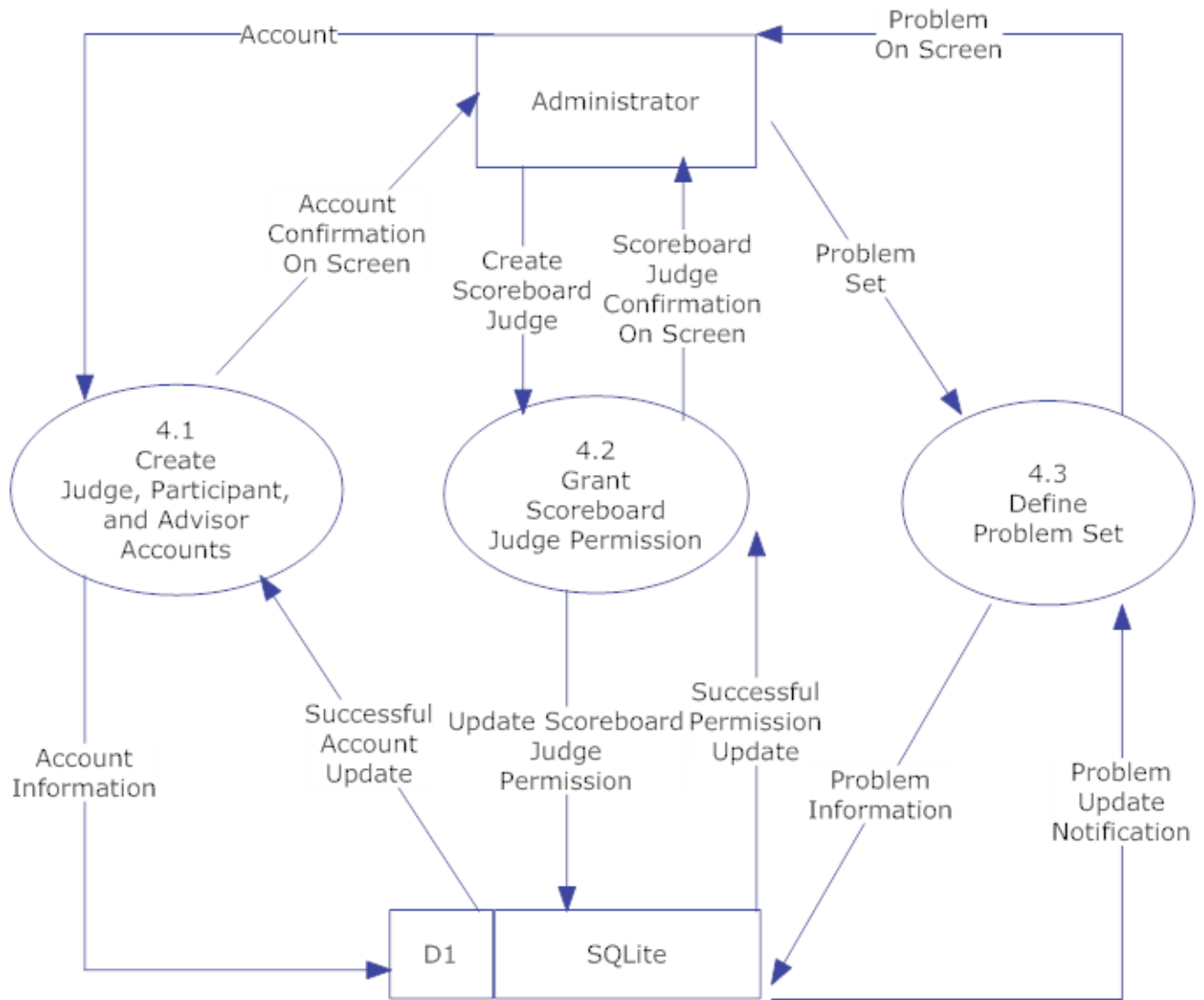


Figure 1.11: Level 1 Diagram of Review Problems

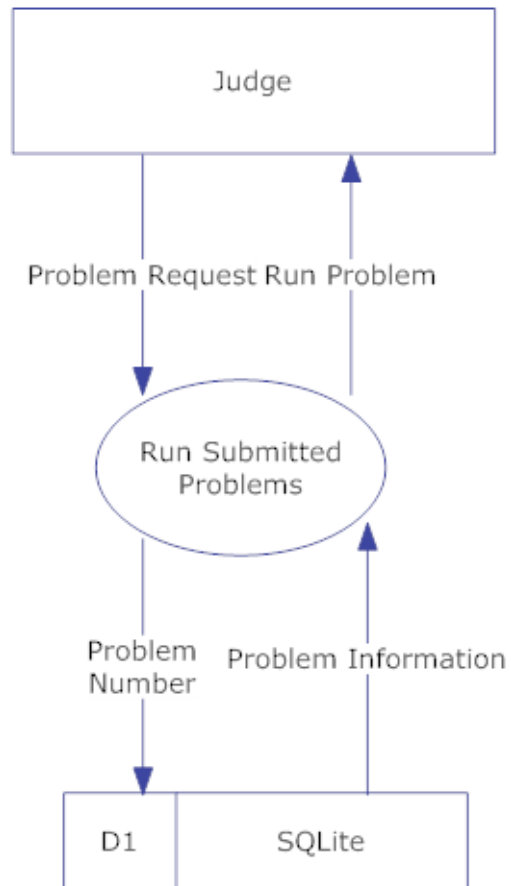
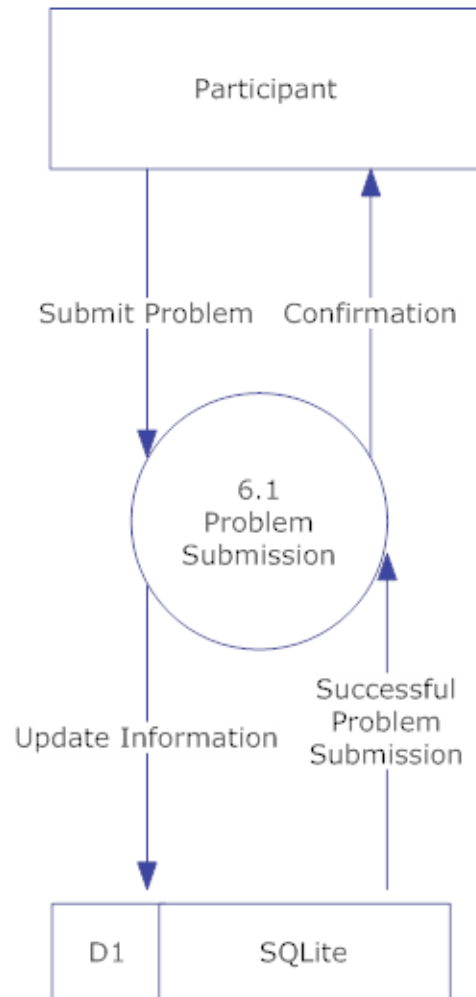


Figure 1.12: Level 1 Diagram of Submit Problems



1.7.5 Level 2 Diagrams

Figure 1.13: Level 2 Diagram of Manage Scoreboard

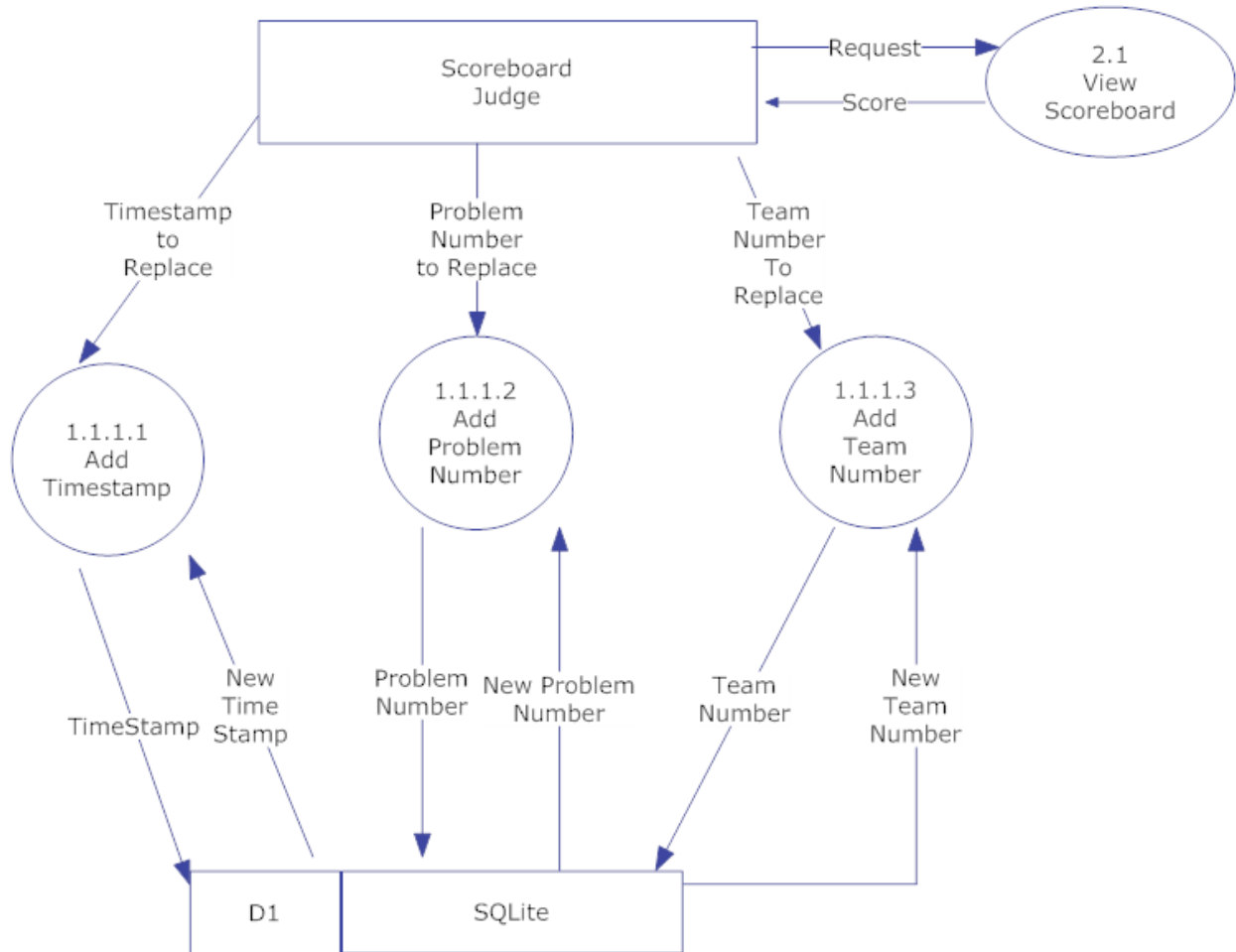


Figure 1.14: Level 2 Diagram of Notify

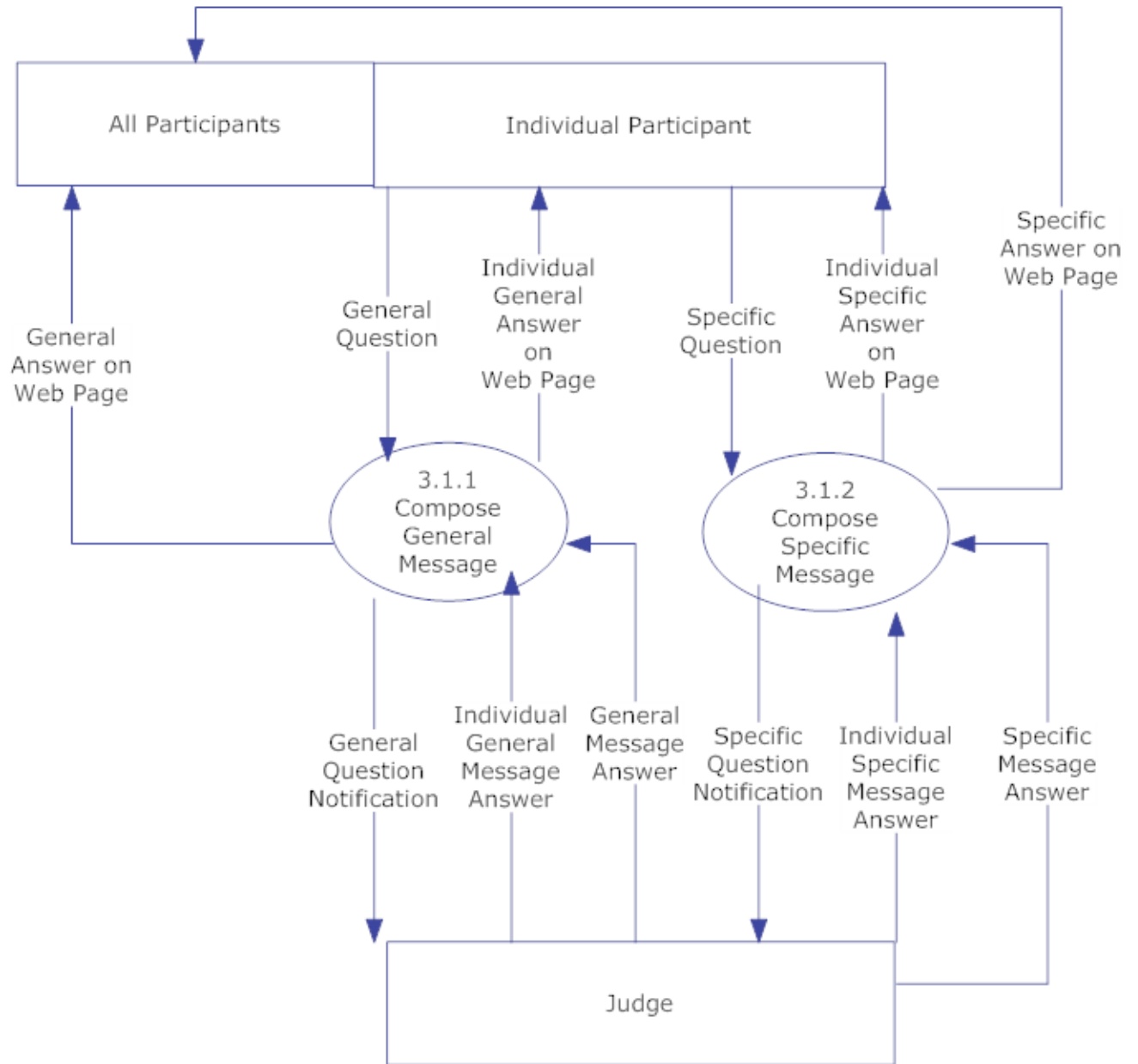
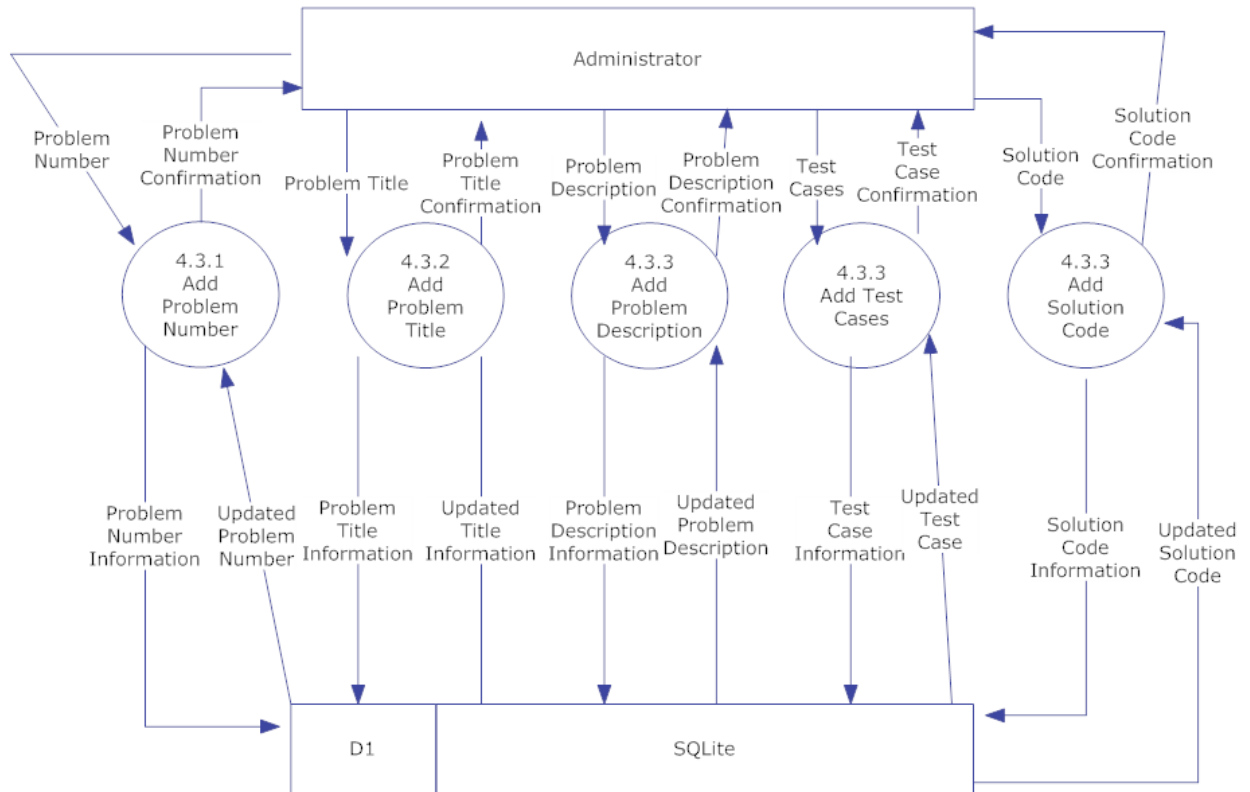
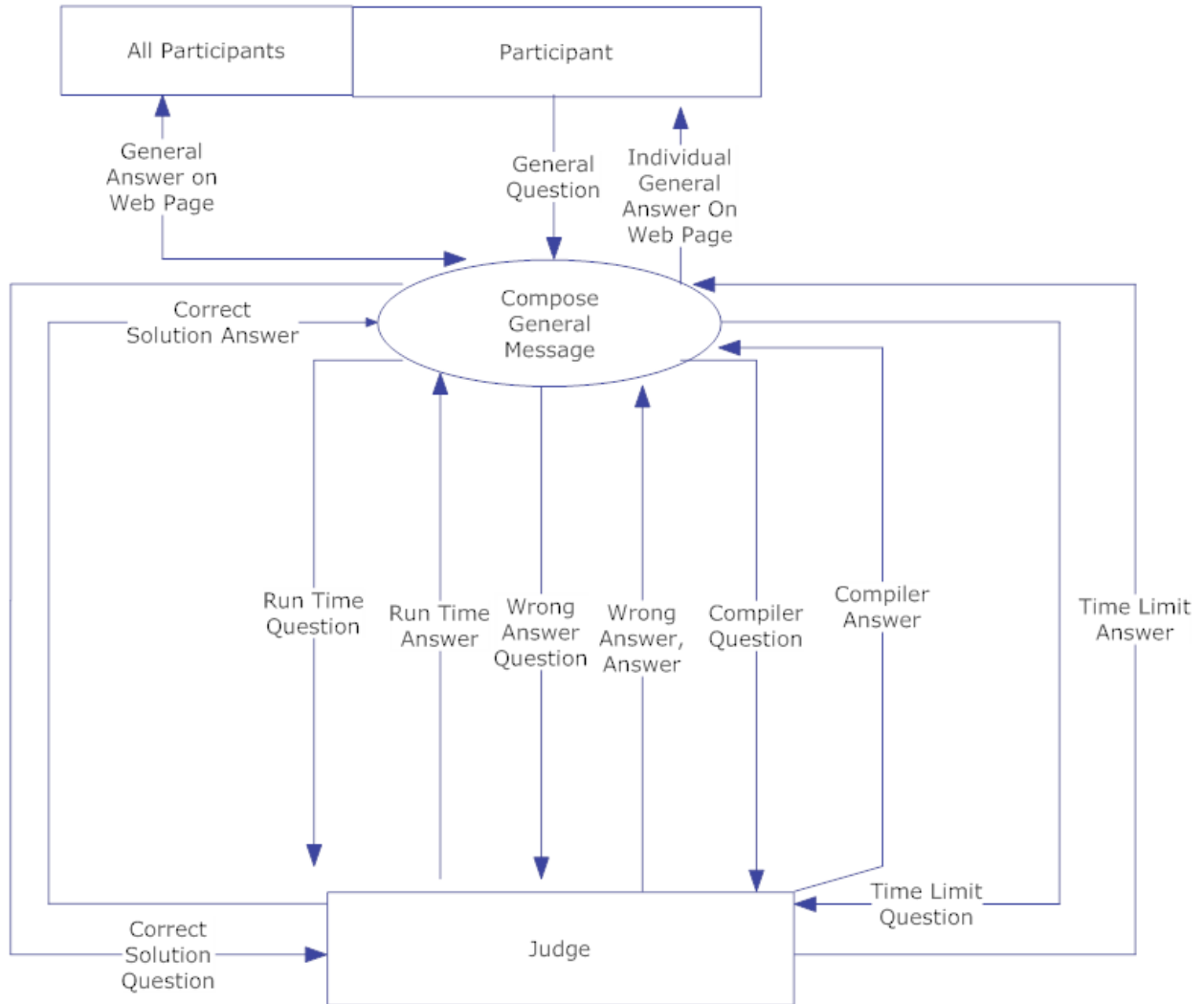


Figure 1.15: Level 2 Diagram of Define Problem Set



1.7.6 Level 3 Diagrams

Figure 1.16: Level 3 Diagram of Notify



1.8 Logical Data Dictionary

A Logical Data Dictionary is an inventory of all data entities and variables used within a system. All data entered, created, changed or stored by the system is included in the Logical Data Dictionary. Our data dictionary contains the following columns for each data entity described below:

- Data Name - The name of the data entity
- Applicable to - The views that the data is used or displayed in
- Data Type - How the data entity will be stored.
- Data Size - How large that data entity can be
- Description - What the data entity is used for
- Acceptable Input - A descriptor of allowed input
- Good Example of Input - An acceptable example input
- Notes - Additional comments on an entity

The data dictionary will be available on the $P = NP_{solutions}$ website as a Microsoft Excel File.

1.9 Prototype Screens

The following images were used for feature discovery and outline what SLICE will look like aesthetically.

1.9.1 General View

Figure 1.17: Login Screen for SLICE



1.9.2 Administrative User Views

Figure 1.18: Contest Configuration Page

SLICE 0h 00m 12s

Configuration
Messaging
Scoreboard

Contest Title:

Practice Time: Start Stop

Contest Time:

Scoreboard Time:

Manage Judges

Manage Participants/Advisors

Change Administrator Password

Reset

Save

Copyright 2011 - P=NP Solutions

Figure 1.19: Pop-Up for the Manage Judges Button



Figure 1.20: Pop-Up for the Manage Participants/Advisors Button

The screenshot displays a web-based interface for managing participants and advisors. At the top, the word "SLICE" is prominently displayed in a large, stylized font. Below this, a central panel is divided into several sections:

- List of participants/advisors...:** A scrollable list box on the left side of the panel, currently empty.
- Edit Form:** A form on the right side of the panel with three input fields labeled "Name:", "Username:", and "Password:". Below these fields are two buttons: "Edit" and "Delete".
- Add Form:** A second form below the edit form, also with three input fields labeled "Name:", "Username:", and "Password:". Below these fields is a checkbox labeled "Team Advisor" and an "Add" button.

At the bottom of the interface, a green footer bar contains the text "Copyright 2011 - P=NP Solutions".

Figure 1.21: Pop-Up for the Change Password Button

SLICE

Current Password:

New Password:

Confirm Password:

Submit

Copyright 2011 - P=NP Solutions

Figure 1.22: Administrative Messaging View



Figure 1.23: Administrative Scoreboard View

SLICE

Logout

Configuration

Messaging

Scoreboard

Contest Title

Time: 5:00 PM - 8:00 PM
Current Time: 6:15 PM

Team 1	Q1	Q2	Q3
Team 2	Q1	Q2	Q3
Team 3	Q1	Q2	Q3

Copyright 2011 - P=NP Solutions

1.9.3 Judge User Views

Figure 1.24: Judge Console and Scoring View



Figure 1.25: Pop-Up for Problem Grading

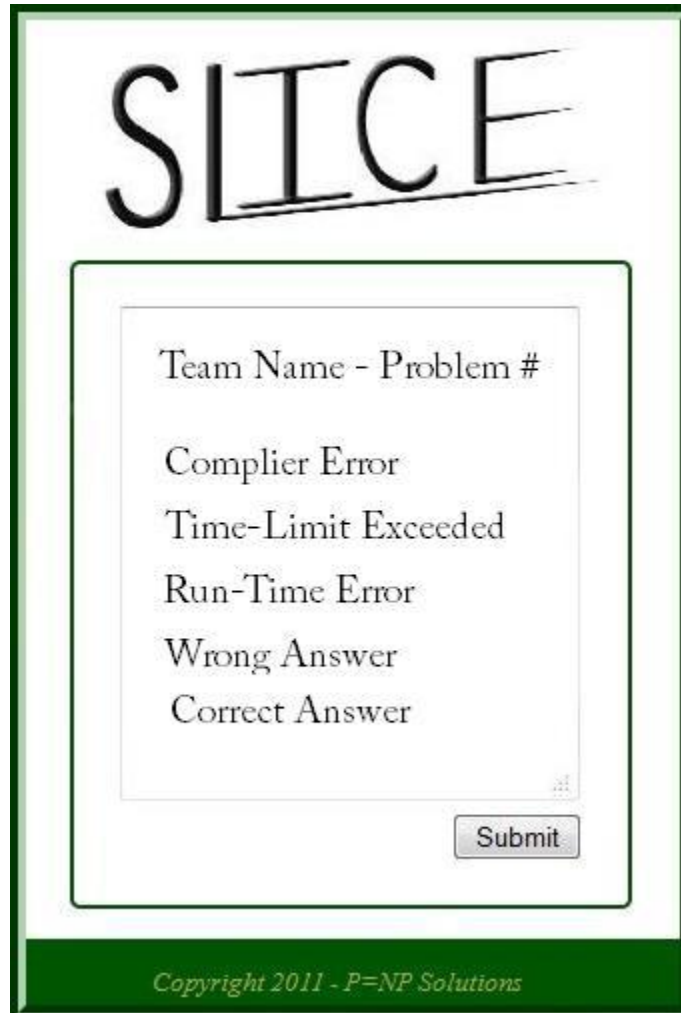


Figure 1.26: Judge Messaging View



Figure 1.27: Scoreboard Judge Scoreboard View (Judge View is missing the edit button)

SLICE

Edit Logout

Console

Messaging

Scoreboard

Contest Title

Time: 5:00 PM - 8:00 PM
Current Time: 6:15 PM

Team 1	Q1	Q2	Q3
Team 2	Q1	Q2	Q3
Team 3	Q1	Q2	Q3

Copyright 2011 - P=NP Solutions

1.9.4 Participant User Views

Figure 1.28: Participant Submission View

SLICE 0h 00m 07s

Submission Form

Problems

Clarification Request

Messaging

Scoreboard

Submission Forms

File Name: Browse...

Submit

Copyright 2011 - P=NP Solutions

Figure 1.29: Participant Problem View

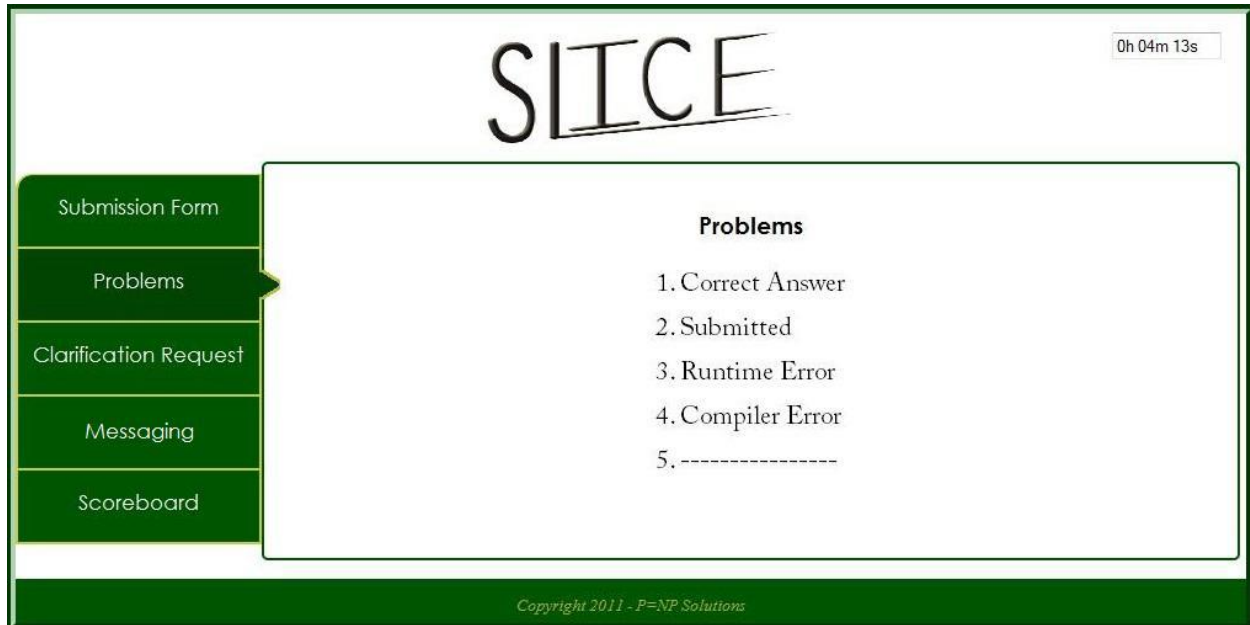


Figure 1.30: Participant Clarification Request View



Figure 1.31: Participant Messaging View



Figure 1.32: Participant Scoreboard View

Submission Form

Problems

Clarification Request

Messaging

Scoreboard

SLICE

0h 10m 40s

Scoreboard

Team 1	Q1	Q2	Q3
Team 2	Q1	Q2	Q3
Team 3	Q1	Q2	Q3

Copyright 2011 - P=NP Solutions

1.10 Testing Plan

1.10.1 Overview & Strategy

SLICE will be tested on the four major browsers, namely Internet Explorer, Mozilla Firefox, Google Chrome, and Safari. However, testing may not be limited to those browsers. Testing will be based upon the functional requirements, and will be divided into different modules. Each separate module will have its own unit, and each of these units will be tested separately from the other units. If each unit test works on its own, then tests will be conducted on each module. This will be done for each module, and once all the modules have been tested individually, they will be then integrated and tested within the system as a whole. The Detailed Design will elaborate on all of the testing modules and units. The results of the tests will be presented in the Acceptance Test document, at which point $P = NP_{solutions}$ will determine if all of the functional requirements were met efficiently.

1.10.2 Acceptance Criteria

The acceptance criteria for the system will be defined by the functional requirements inventory, listed in the Requirements Specification document. Additionally, the non-functional requirements inventory, also listed previously, will be used. The functional requirements, by definition, include the abilities of the system and their ability to be tested. Non-functional requirements, on the other hand, define how the system should behave, and are not able to be tested. After the completion and testing of SLICE, $P = NP_{solutions}$ will determine which functional requirements were fulfilled.

1.10.3 Unit Tests

Each unit test will be tested separately at first. Once each unit test passes, a full system test will be conducted on all units to ensure accuracy and efficiency. The following is a list of all separate units that will be tested:

- User Login
- Problem Submission
- Judging

The Unit Test document will be available on the $P = NP_{solutions}$ website as a Microsoft Excel File.

1.10.4 Test Cases

Each test case will be numbered and contain a description of what the test case is. The action to be performed, or the input entered by the user, the state before the test, and the expected result are described for the three units below. There are cases described for the correct actions of users as well as incorrect actions. Results will be recorded once testing for each unit commences.

1.11 Development and Production Environment

Our PC is an Optiplex 760 (name: seb2)

- Windows Vista Enterprise (32 bit) with service pack 1 installed
- An Intel Core2 Duo CPU processor (E7500 @ 2.93GHz)
- 4 GB of Memory

Our Mac is an iMac (model identifier: iMac5,1)

- Mac OS X, version 10.6.4
- An Intel Core 2 Duo Processor (667 MHz)
- 1 GB of Memory

Our server is an x86_64 PC

- Hostname: oraserv.cs.siena.edu
- CentOS 5.2 (final)
- Kernal: 2.6.18-92.el5
- Intel Xeon 2.66 GHz CPU
- 8 GB of Memory
- Java SE Runtime Environment (build 1.6.0.10-rc-b28)
- GCC Version 4.1.2 20071124 (Red Hat 4.1.2-42)
- Python 2.4.3

The following installs and upgrades are in-progress on our server:

- Upgrade Python from 2.4.3 to 2.7.1
- Install the mod_wsgi Apache module
- Install Python setuptools 0.6
- Install Django 1.3.1

Our operating environment is a web based application that is only to be used within the SoS (Siena School of Science) network. The maintenance environment includes all of the hardware and software that is used to modify SLICE both in programming code and in visual appearance. This includes the computers in the Software Engineering Lab (stated above) and the software Adobe Dreamweaver, Adobe Fireworks, Adobe Photoshop, Internet Explorer, Mozilla Firefox, Google Chrome, Safari, etc

Appendix A

Glossary of Terms and Gantt Chart

- Actor: An entity in UML Use Case Diagrams and UML Activity Diagrams. It represents the human and non-human external entities that interact with the system
- Activity Diagram: A diagram based on the Unified Model Language (UML). This represents the processes that comprise a certain activity within the system. These diagrams are generally created with the perspective of an actor in mind.
- Compiler: A program that reads in source code and generates an executable.
- Data Flow Diagrams: Used to show how data is moved and processed within a system. There are various levels, each providing more detail than the next.
- Data Flows: A component of a Data Flow Diagram that represents the movement of data.
- Data Stores: A component of a Data Flow Diagram that represents a location in which information or data is stored.
- Django: A Python Web framework
- External Entities: A component of a Data Flow Diagram that represents any human or non-human user of a Software System.
- Functional Requirements Inventory: Define what the system will be able to do and what is testable about the system.
- Hardware: The physical parts of a computer, such as the hard drive and the CPU.
- HTML: Hypertext Markup Language is the scripting language used to describe the information contained on a website. HTML utilizes Cascading Style Sheets (CSS) to generate the style of the page. HTML and CSS are parsed by web browsers, such as Internet Explorer and Firefox, to render the websites for users Software: The intangible components of a computer and server. It is a set of machine-level instructions that

is run from within the memory, and is used to perform a specific set of functions. Examples include Microsoft Word, Adobe Photoshop, and Mozilla Firefox.

- Microsoft Internet Explorer a web-browser developed by Microsoft
- Non-Functional Requirements: Specifies how a product is supposed to be in relationship to the functional requirements.
- Process: A component of a Data Flow Diagram that represents an activity that transforms or manipulates data.
- UML: Unified Modeling Language is the industry-standard language for the specification. Visualization, construction, and documentation of the components of software systems.
- Unit Testing: a method by which individual units of source code are tested to determine if they are fit for use. A unit is the smallest testable part of an application
- Use Case Diagram: Represents the high-level functions of the system. Also depicts how actors interact with each of those functions.
- User Case Narrative: an explanation of the functions and abilities users have for a specific Software System.

