

# Requirements Specification

---

TRACE Technologies

**Produced By: Chris Small, Ryan Kennedy, Alyssa Nghiem, Eduardo Pinto Barbosa, Tyler Vorpahl**

**For: Dr. Darren Lim, Dr. Tim Lederman**

**10/29/2012**



# 1. Table of Contents

1. Table of Contents .....	1
2. Product Overview and Summary .....	2
3. Development and Operating Environment .....	2
4. User Case Narratives .....	3
5. Use Case Diagram .....	6
6. Data Flow Diagrams .....	8
6.1 WORMS Context Diagram	
6.2 WORMS Level 0 Diagram	
6.3 WORMS Level 1 Diagram: Create Account	
6.4 WORMS Level 1 Diagram: Login	
6.5 WORMS Level 1 Diagram: Review	
6.6 WORMS Level 2 Diagram: Check Reviews	
6.7 WORMS Level 1 Diagram: Submit	
6.8 WORMS Level 1 Diagram: Maintain System	
6.9 WORMS Level 1 Diagram: Configure Conference	
6.10 WORMS Level 1 Diagram: Register for Conference	
7. Functional Requirements Inventory .....	19
8. Non-functional Requirements .....	23
9. Exception Handling .....	23
10. Implementation Priorities .....	23
11. Foreseeable Modifications and Enhancements.	23
12. Testing Requirements .....	23
13. Acceptance Criteria .....	24



## 2. Product Overview and Summary

Dr. Lim will be holding the Consortium for Computing Sciences in Colleges Northeast (CCSCNE) Conference in April 2013. The organization of this conference is very complex and there is a need for software that can assist in the processes of handling attendee registration, scholarly work submission, and scholarly work review. The WORMS web application will provide these services.

## 3. Development and Operating Environment

WORMS will be developed using the resources in the software engineering lab. Below are the available resources that will be used for development and maintenance of the system.

### Server

- Web Server: Apache
- Database: MySQL
- Operating System: Linux

### SE Lab hardware

- Dell ACP x86-based PC
  - Operating System: Windows Vista Enterprise
  - RAM: 4 GB
  - Processor: Intel Core 2 Duo 2.93GHz
- Apple iMac 21.5 inch
  - Operating System: 10.7.4 Lion
  - RAM: 4 GB
  - Processor: Intel Core i5



## 4. User Case Narratives

### **System Administrator**

The System Administrator is the person responsible for configuring the details of the system and monitoring the system. The System Administrator will log into a built-in account on the web site. The System Administrator will be able to configure a list of the email addresses of Conference Chairs. The System Administrator will also be able to configure a list of the email addresses of Submission Chairs and the type of scholarly work submission each Submission Chair is responsible for. The System Administrator will be able to configure a list of questions that users are asked when registering for the conference. The System Administrator will be able to view a list of all submitted scholarly works. The System Administrator will be able to view and remove accounts from the system. The System Administrator will be able to configure the list of topics that scholarly works can be categorized into.

### **Conference Chair**

A Conference Chair is one of the people responsible for organizing the conference. A Conference Chair will create an account in the WORMS web application. In order to have the necessary permissions, a Conference Chair must have created the account using an email address on the list of Conference Chair emails configured by the System Administrator. A Conference Chair will be able to log into the WORMS web application. A Conference Chair will be able to schedule the times and locations of events at the conference. A Conference Chair will be able to edit other conference information such as parking information, nearby hotels, and nearby attractions. A Conference Chair will be able to configure banquet meal selections.

### **Submission Chair**

A Submission Chair is one of the people responsible for organizing the submission and review of a certain type of scholarly work to the conference. A Submission Chair will create an account in the WORMS web application. In order to have the necessary permissions, a Submission Chair must have created the account using an email address on the list of Submission Chair emails configured by the System Administrator. Each Submission Chair will be associated with a specific type of scholarly work. A Submission Chair will log into the WORMS web application. A Submission Chair will specify deadlines for submissions of scholarly works. A Submission Chair will be able to view a list of scholarly works submitted and the person that submitted each scholarly work. A Submission Chair will be able to view the reviews given to scholarly works. A Submission Chair will choose whether or not a scholarly work should be accepted to the conference. A Submission Chair will confirm the credentials of each Reviewer.



## **Reviewer**

A Reviewer is a person with sufficient credentials who wishes to review submitted scholarly works. A Reviewer will create an account in the WORMS web application. A Reviewer will log into the WORMS web application. The Reviewer will enter credentials into a form in the WORMS web application and apply to become an accepted reviewer. After a Reviewer has been accepted, the Reviewer will be able to view scholarly works that have been submitted and assigned to that Reviewer. An accepted Reviewer will be able to submit a review for each scholarly work that was assigned to that Reviewer.

## **Submitter**

A Submitter is a person who wishes to submit a scholarly work to the conference. A Submitter will create an account in the WORMS web application. A Submitter will log into the WORMS web application. A Submitter will select the type of scholarly work that the Submitter wants to submit. A Submitter will select the topic of the scholarly work that the submitter will submit. A Submitter will upload a PDF (Portable Document Format) file containing a paper or an abstract for a scholarly work. A Submitter will be able to check whether or not the scholarly work that the Submitter submitted has been reviewed.

## **Programming Contest Team Advisor**

A Programming Contest Team Advisor is a faculty member responsible for organizing a programming team that will compete at the programming competition. A Programming Contest Team Advisor will create an account in the WORMS web application. A Programming Contest Team Advisor will register a programming contest team. A Programming Contest Team Advisor will provide the email addresses of the students on the programming contest team. A Programming Contest Team Advisor will pay the fee necessary to participate in the contest.

## **Programming Contest Participant**

A Programming Contest Participant will create an account in the WORMS web application. A Programming Contest Participant will register to attend the conference. A Programming Contest Participant will be able to view a confirmation that the Programming Contest Team Advisor has added the Programming Contest Participant to a team. A Programming Contest Participant will be able to view details about the programming contest.

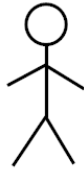


### **Conference Attendee**

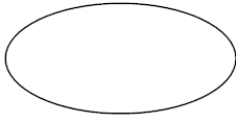
A Conference Attendee will create an account in the WORMS web application. A Conference Attendee will log into the WORMS web application. If a Conference Attendee wishes to attend the banquet, the Conference Attendee will make a meal selection. A Conference Attendee will pay the fee for attending the conference. A Conference Attendee may choose to purchase extra banquet tickets. A Conference Attendee will be able to view parking and hotel information. A Conference Attendee will be able to view the schedule of events during the conference. A Conference Attendee will be able to view conference sponsor information.

## 5. Use Case Diagram

### UML Use Case Diagram Legend



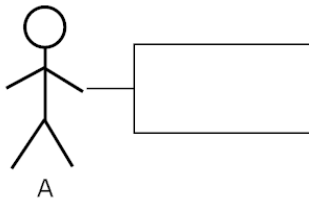
Actor: an external entity that interacts with the system. This may or may not be a human user



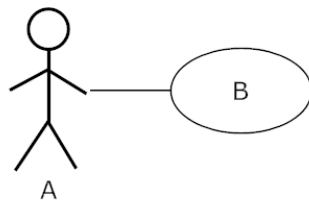
Use Case: a process within the system that actors may interact with



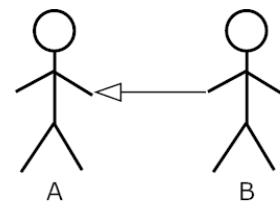
System Boundary: the boundary between the system internals and external Actors



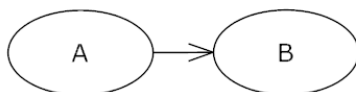
Actor A interacts with all Use Cases in the system



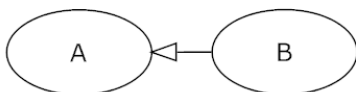
Actor A interacts with Use Case B



Actor B interacts with call Use Cases that Actor A is able to interact with.



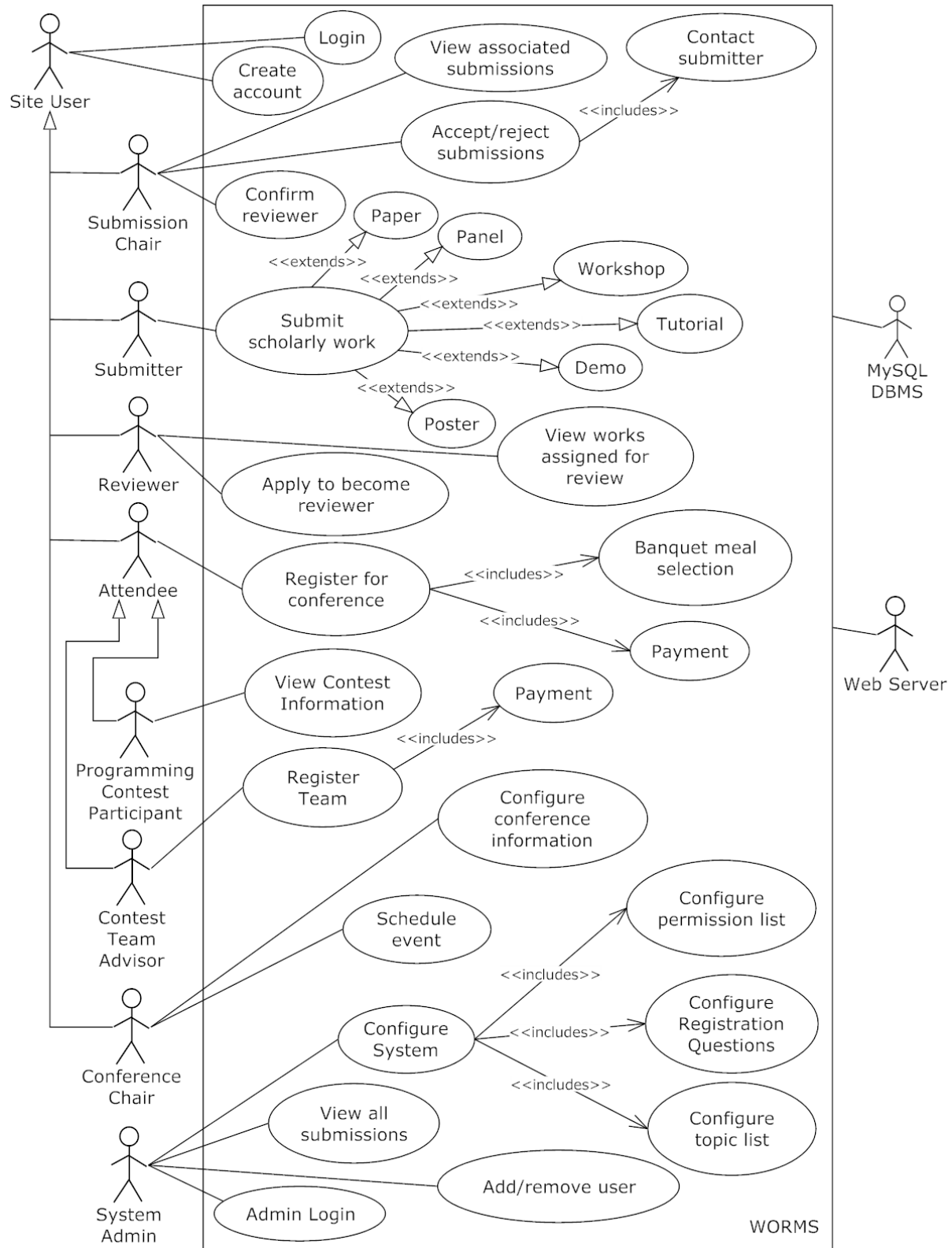
Use Case A includes Use Case B



Use Case B is a more specific type of Use Case A



### UML Use Case Diagram



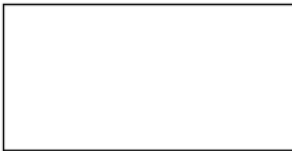




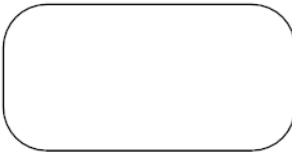
## 6. Data Flow Diagrams

This section will contain data flow diagrams for the system. These diagrams represent the movement of data between processes in the system as well as the movement of data between processes and external entities outside the system. The diagrams are a tool for analyzing the structure of the system and the ways in which data will be stored and retrieved by different processes. These diagrams model data flows at different levels of detail in the system. The symbols that are used in the diagrams are defined as follows.

### Data Flow Diagram Legend



Entity - either a source or destination of data for the system



Process - System functions that can receive data, modify data, and output data.



Data Store - A storage location for persistent data in the system.



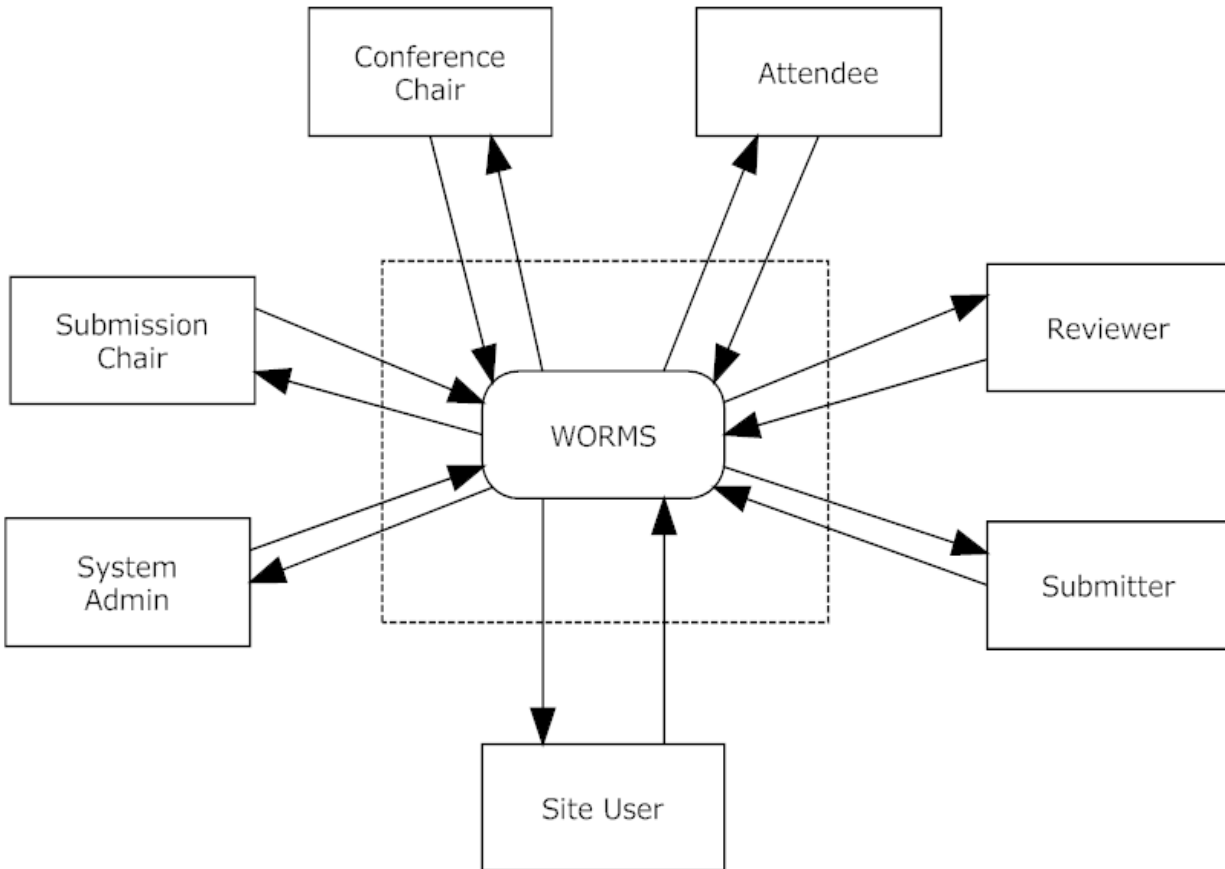
Data Flow - movement of a data entity.



System boundary - the boundary between the system and external entities.

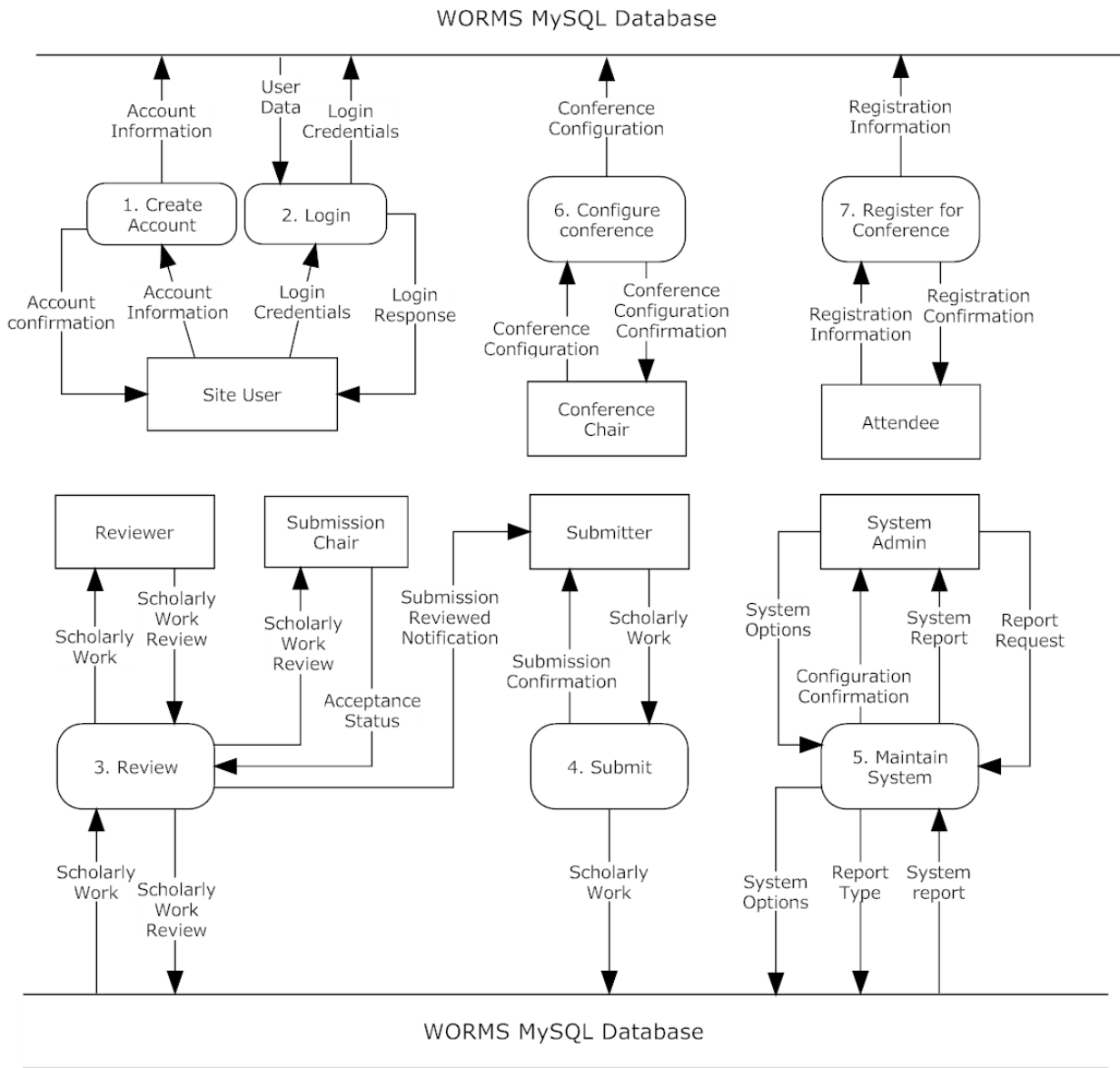


### 6.1 Context Diagram





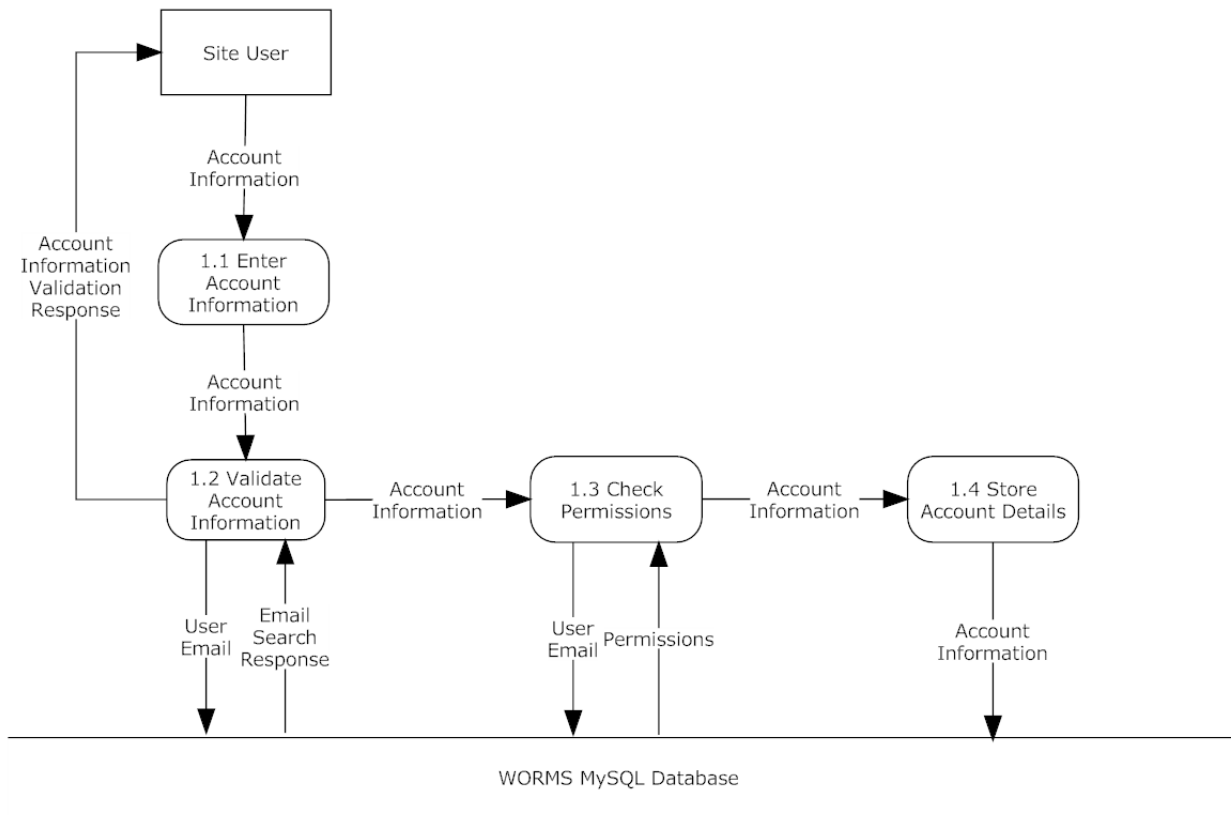
## 6.2 WORMS Level 0 Diagram





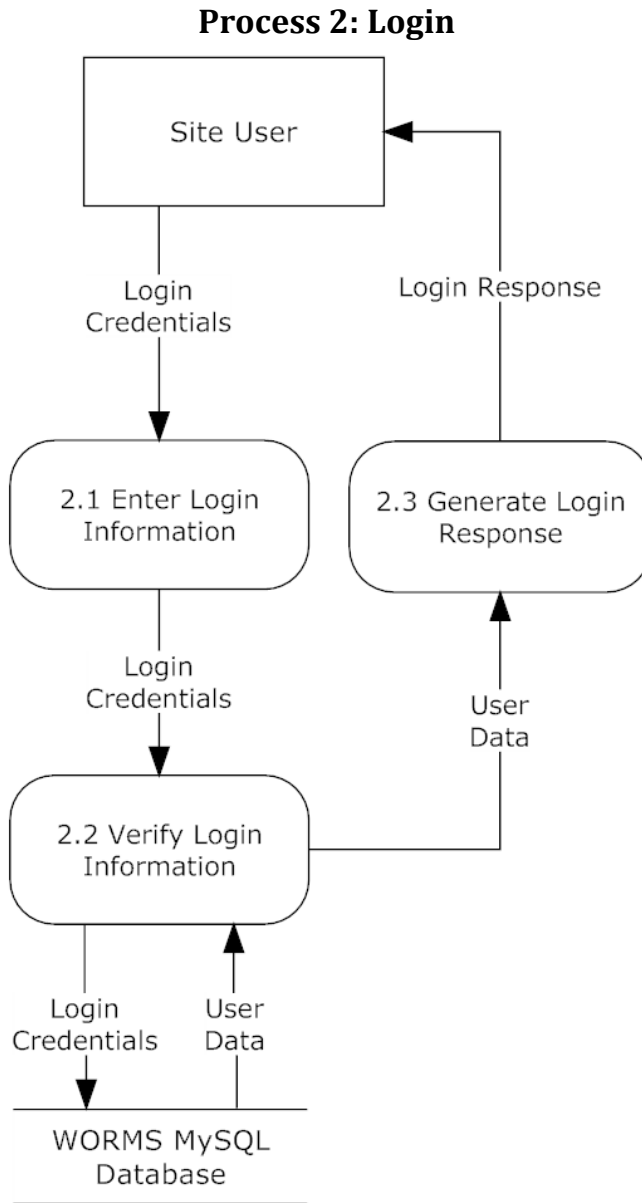
### 6.3 WORMS Level 1 Diagram

#### Process 1: Create Account



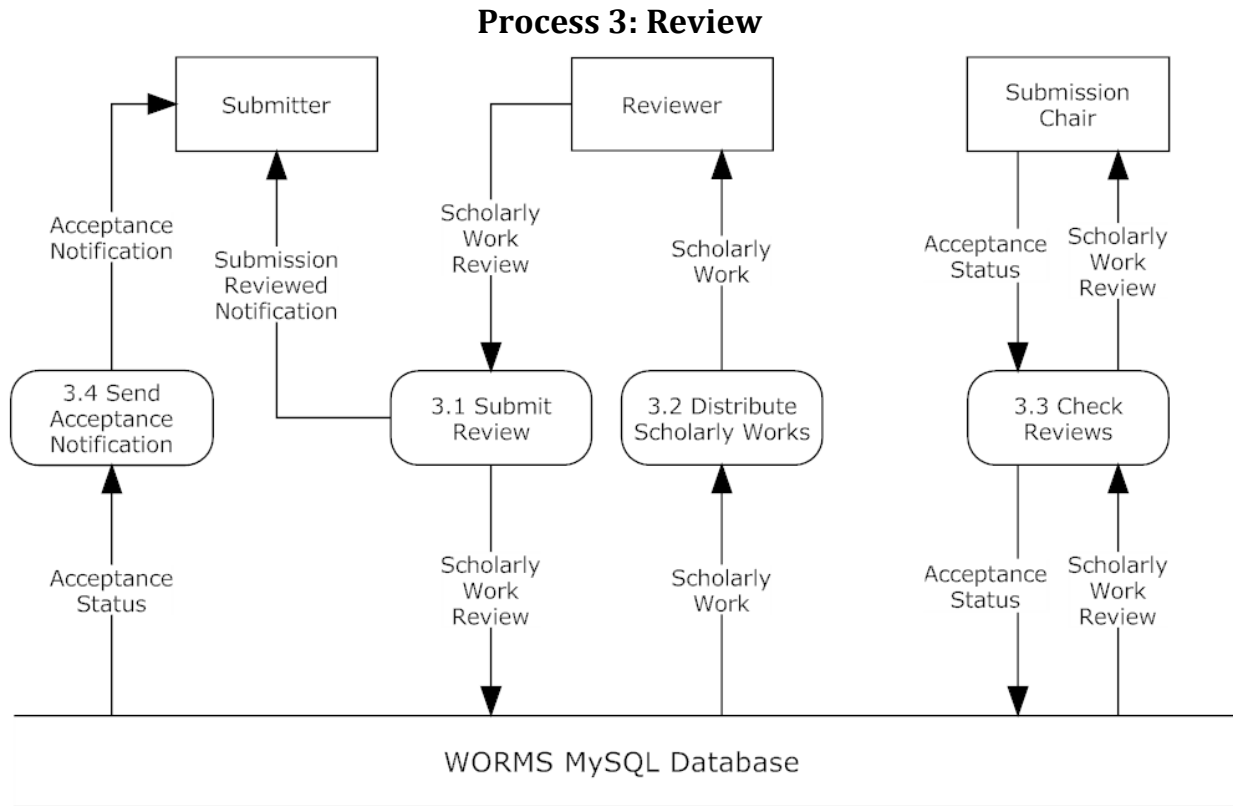


### 6.3 WORMS Level 1 Diagram





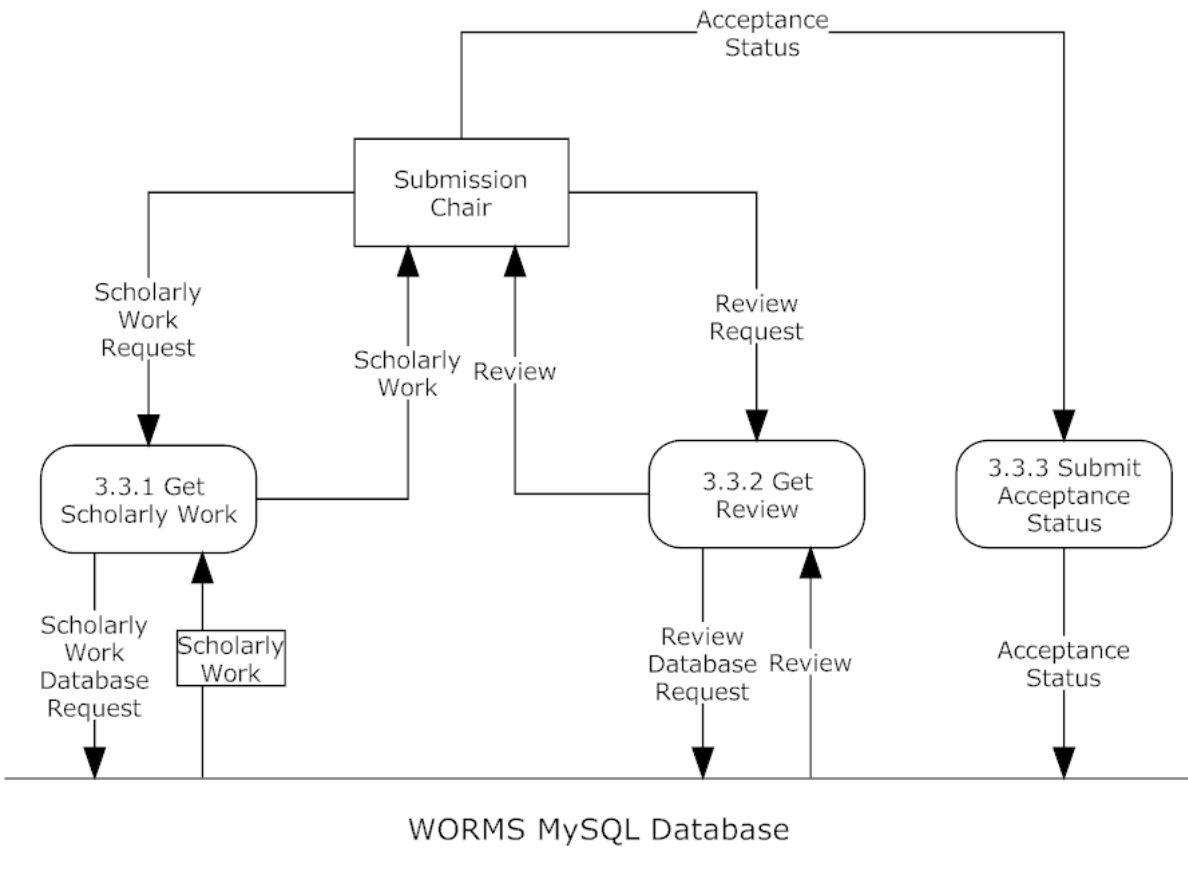
### 6.4 WORMS Level 1 Diagram





## 6.5 WORMS Level 2 Diagram

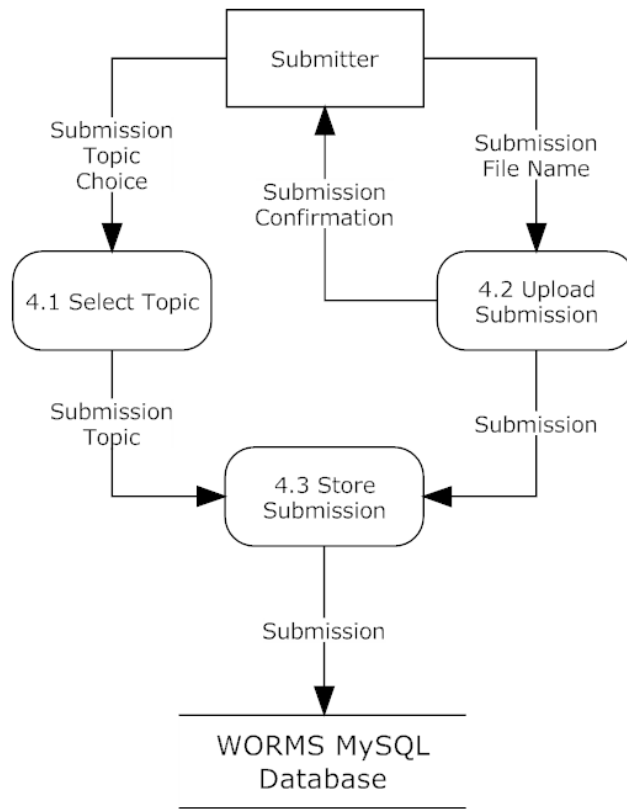
### Process 3.3: Check Reviews





### 6.6 WORMS Level 1 Diagram

#### Process 4: Submit

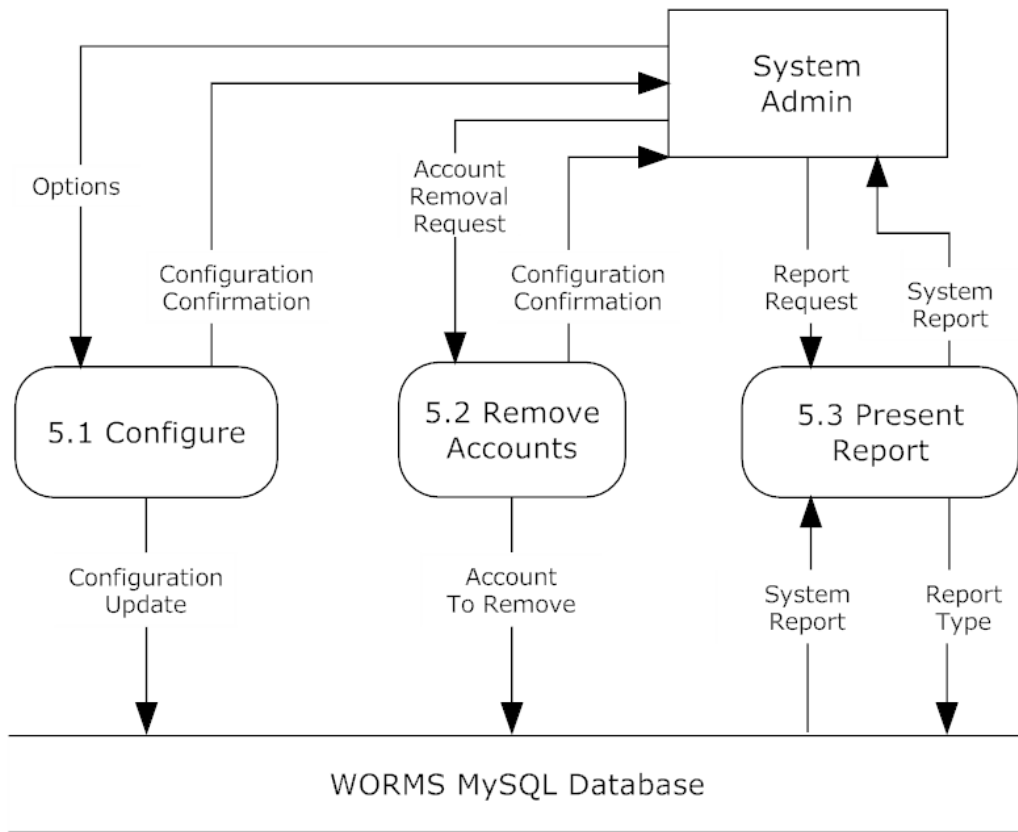






## 6.7 WORMS Level 1 Diagram

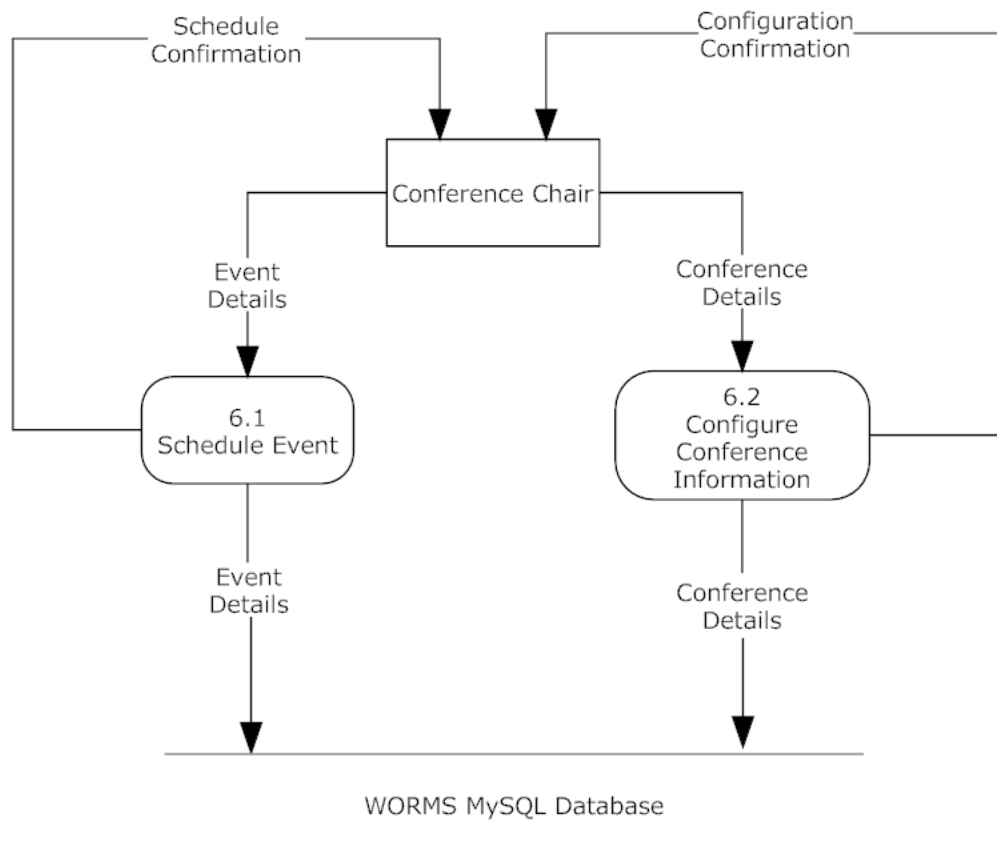
### Process 5: Maintain System





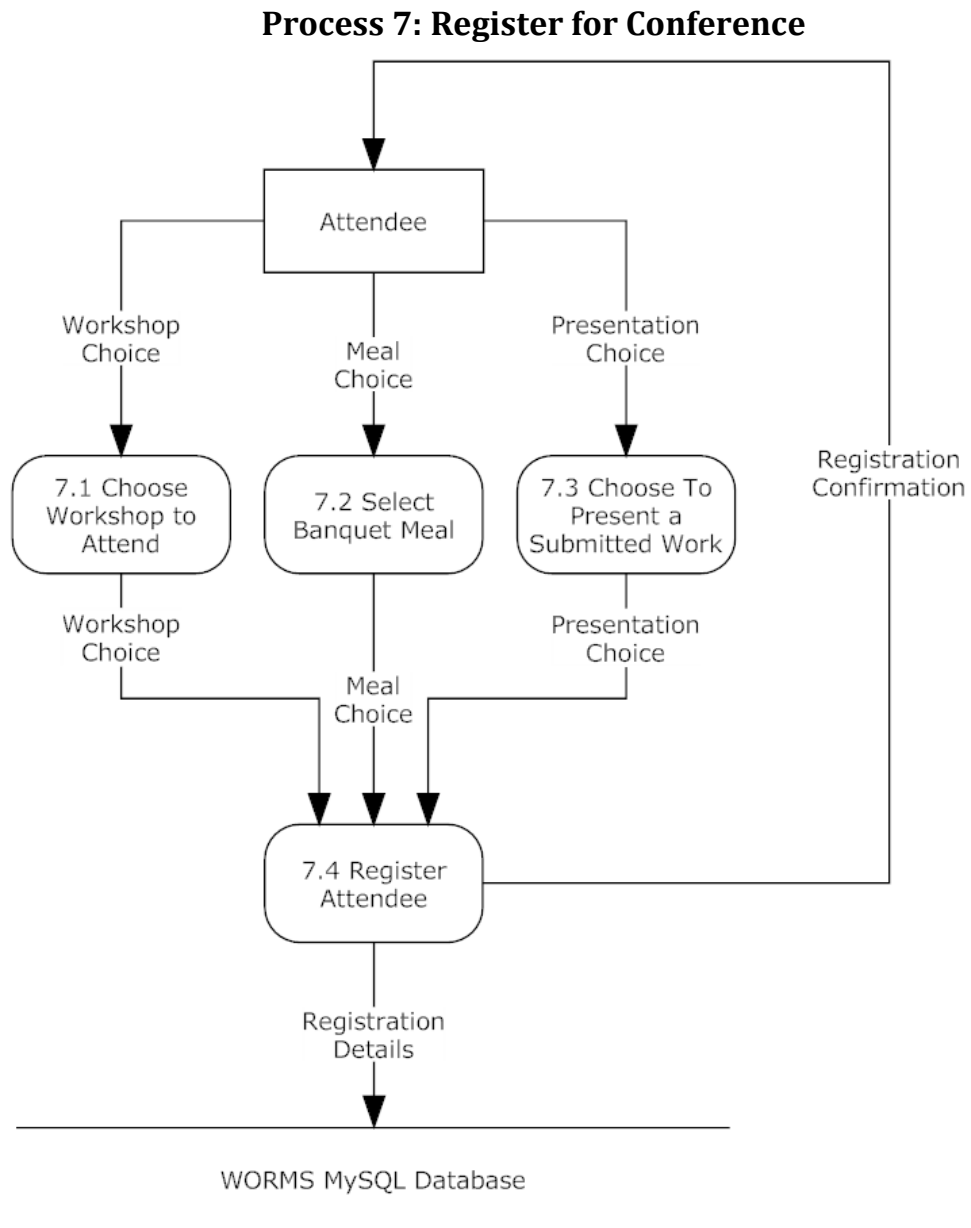
## 6.8 WORMS Level 1 Diagram

### Process 6: Configure Conference





## 6.9 WORMS Level 1 Diagram





## 7. Functional Requirements Inventory

### System Administrator

- Will be able to log onto WORMS with a provided username and password that is stored on the database
  - An incorrect username and/or password will result in appropriate error message
- Will be able to log off WORMS at any point
- Will be able to configure a list of email addresses stored on the database of the conference chairs
- Will be able to configure a list of email addresses stored on the database of the submission chairs
  - Will be able to configure the types of scholarly works each submission chair is responsible for
- Will be able to configure a list of questions that users are asked while registering stored on the database
- Will be able to view a list of all submitted scholarly works stored on the database
- Will be able to view and/or remove accounts from the system
- Will be able to configure the topics scholarly works can be categorized into

### Conference Chair

- Will be able to create an account using a valid email address confirmed by the system administrator
- Will be able to log onto WORMS with a provided username and password stored on the database
  - An incorrect username and/or password will result in appropriate error message
- Will be able to log off WORMS at any point
- Will be able to schedule times and locations of events to be stored on the database
- Will be able to edit information such as parking information, nearby hotels, and nearby attractions stored on the database
- Will be able to configure banquet meal selections stored on the database



## Submission Chair

- Will be able to create an account using a valid email address confirmed by the system administrator
- Will be able to log onto WORMS with created username and password stored on the database
  - An incorrect username and/or password will result in appropriate error message
- Will be able to log off WORMS at any point
- Will be associated with a specific type of scholarly work
- Will be able to specify deadlines for submissions of scholarly works
- Will be able to view a list of scholarly works submitted stored on the database
  - Will be able to view each scholarly works and submitter who submitted the scholarly work stored on the database
- Will be able to view the reviews given to scholarly works stored on the database
- Will be able to choose whether or not a scholarly work should be accepted to the conference
- Will be able to confirm the credentials of each reviewer stored on the database

## Reviewer

- Will be able to create an account using a valid email address
- Will be able to log onto WORMS with created username and password stored on the database
  - An incorrect username and/or password will result in appropriate error message
- Will be able to log off WORMS at any point
- Will be able to register to be a submitter, chair, programming contest participant, and/or programming contest advisor
- Will be able to register to become an accepted reviewer
- Will be able to view scholarly works that have been submitted stored on the database
- Will be able to review scholarly works assigned to the reviewer stored on the database
  - Will be able to fill out review form for scholarly work
  - Will be able to submit review form of scholarly work to be stored on the database



## Submitter

- Will be able to create an account using a valid email address
- Will be able to log onto WORMS with created username and password stored on the database
  - An incorrect username and/or password will result in appropriate error message
- Will be able to log off WORMS at any point
- Will be able to register to be a reviewer, chair, programming contest participant, and/or programming contest advisor
- Will be able to select the type of scholarly work the submitter wants to submit
  - Will be able to select the topic of scholarly work the submitter will submit
  - Will be able to upload a PDF file containing a paper or abstract for a scholarly work to be stored on the database
  - Will be able to check whether or not the scholarly work has been reviewed

## Programming Contest Team Advisor

- Will be able to create an account using a valid email address
- Will be able to log onto WORMS with created username and password stored on the database
  - An invalid username and/or password will result in appropriate error message
- Will be able to log off WORMS at any point
- Will be able to register to be a reviewer, submitter, chair, and/or programming contest participant
- Will be able to register a programming team
  - Will be able to provide email addresses of the students participating in the programming contest to be stored on the database
- Will be able to pay the fee necessary to participate in the programming contest



## **Programming Contest Participant**

- Will be able to create an account using a valid email address
- Will be able to log onto WORMS with created username and password stored on the database
  - An invalid username and/or password will result in appropriate error message
- Will be able to log off WORMS at any point
- Will be able to register to be a reviewer, submitter, chair, and/or programming contest advisor
- Will be able to view a confirmation that the programming contest team advisor has added the programming contest participant to a team stored on the database
- Will be able to register as members of a programming contest team that has already been created by the programming contest advisor to be stored on the database
- Will be able to view details about the programming contest stored on the database

## **Conference Attendee**

- Will be able to create an account using a valid email address
- Will be able to log onto WORMS with created username and password stored on the database
  - An invalid username and/or password will result in appropriate error message
- Will be able to register to be a submitter, chair, programming contest participant, and/or programming contest advisor
- Will be able to log off WORMS at any point
- Will be able to participate in banquets
  - Will be able to make a meal selection for the banquet stored on the database
  - Will be able to purchase extra banquet tickets
- Will be able to pay the fee necessary to attend the conference
- Will be able to view information about the conference stored on the database
  - Will be able to view hotel selections
- Will be able to view the schedule of events during the conference stored on the database
- Will be able to view conference sponsor information stored on the database



## 8. Non-functional Requirements

The following is a list of non-functional system requirements that specify properties of the system that are not necessarily specific features that exist in the software.

- The system will be easily maintained.
- The web application will function properly and consistently in all modern web browsers.
- The system's processes will be responsive and run efficiently.
- The system will be user friendly.

## 9. Exception Handling

All user input from the web application will be validated in order to avoid inconsistent or problematic data in the system. If problematic input is detected, the user will be notified and whatever process is being performed will either recover or terminate as necessary.

## 10. Implementation Priorities

The main implementation priorities for the WORMS system will be: submission of scholarly works, distribution and review of scholarly works, and event scheduling. These features are the most critical for providing a platform handling the organization of an academic conference. The processing of submitted works and the distribution of works to reviewers are processes that are difficult to handle without automation, which means the inclusion of these features in the software is important for improving the experience of conference organizers.

## 11. Foreseeable Modifications and Enhancements

The system is not currently required to handle payments from the users. In the future this functionality could be implemented within the system that is currently specified. The system could potentially be integrated with the software used to run the programming contest, allowing programming contest contestants and programming contest team advisors to check the score board on the contest day and at later times through the WORMS application.

## 12. Testing Requirements

Each subsystem of WORMS will be tested throughout its development to ensure that it is functioning properly and that it interacts properly with the other subsystems. The user interface will be tested for consistency, functionality, and usability as it is developed.





## **13. Acceptance Criteria**

In order for the system to be accepted, software that meets as many of the requirements specified in this document as possible must be provided to the client. If the final software delivered to the client does not meet some number of the requirements given here, there must be reasonable explanation as to why these requirements could not be met.

## **14. Appendix A: Glossary of Terms**

Apache – web server software

CCSCNE – Consortium for Computing Sciences in Colleges Northeast

Linux – an open source operating system kernel

MySQL – a database management system

Scholarly Work – one of the following: poster, paper, workshop, poster, tutorial, panel

UML – unified modeling language

WORMS – Web Organization Registration Management System



# 15. Appendix B: Project Timeline

