



# Detailed Design



Web Organization and Registration Management System

---

**For:**

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

**Produced By:**



**Alyssa Nghiem – Team Leader / QA Lead**

**Eduardo Pinto Barbosa – Team Leader / QA Lead**

**Ryan Kennedy – Lead Programmer**

**Chris Small – Assistant Programmer**

**Tyler Vorpahl – Web Developer**

3/6/2013



# 1. Table of Contents

- 1. Table of Contents ..... 1
- 2. Product Overview ..... 4
- 3. User Case Narratives..... 4
  - System Administrator ..... 4
  - Conference Chair ..... 4
  - Submission Chair ..... 5
  - Reviewer ..... 5
  - Submitter ..... 5
  - Programming Contest Team Advisor ..... 5
  - Programming Contest Participant ..... 6
  - Conference Attendee ..... 6
  - Vendor..... 6
- 4. Use Case Diagram..... 7
  - UML Use Case Diagram Legend..... 7
  - UML Use Case Diagram..... 8
- 5. Data Flow Diagrams ..... 9
  - Data Flow Diagram Legend..... 9
  - Context Diagram ..... 10
  - WORMS Level 0 Diagram..... 11
  - Process 1: Create Account..... 12
  - Process 1.2: Validate Account Information ..... 13
  - Process 2: Login ..... 14
  - Process 3: Review ..... 15
  - Process 3.3: Check Reviews..... 16
  - Process 3.3.1: Get Scholarly Work ..... 17
  - Process 4: Submit ..... 18
  - Process 4.3: Store Submission ..... 19
  - Process 4.3.1: Generate File Name..... 20
  - Process 5: Maintain System ..... 21



Process 5.1: Configure ..... 22

Process 6: Configure Conference ..... 23

Process 7: Register for Conference ..... 24

6. Structure Diagram ..... 25

7. Functional Requirements Inventory ..... 26

    System Administrator ..... 26

    Conference Chair ..... 26

    Submission Chair ..... 27

    Reviewer ..... 27

    Submitter ..... 28

    Programming Contest Team Advisor ..... 28

    Programming Contest Participant ..... 29

    Conference Attendee ..... 29

    Other ..... 29

8. Non-functional Requirements ..... 29

9. Entity-Relationship Diagram and Table Relations ..... 30

    ERD Legend ..... 30

    Entity-Relationship Diagram ..... 31

    Table Relations ..... 32

10. Logical Data Dictionary ..... 32

11. Prototype Screens ..... 32

12. Testing Plan ..... 33

    Overview and Strategy ..... 33

    Unit Tests and Test Cases ..... 33

    Integration Test ..... 33

    System Test ..... 34

    Acceptance Criteria ..... 35

    Non-functional Requirements ..... 37

14. Appendix A: Glossary of Terms ..... 38

15. Appendix B: Project Timeline ..... 39



16. Appendix C: Data Dictionary .....	40
17. Appendix D: Unit Tests.....	42
18. Appendix E: Database Schema .....	52
19. Appendix F: Prototype Screens.....	58



## 2. Product Overview

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 handling attendee registration, scholarly work submission, scholarly work review, as well as scheduling and general information dissemination. The WORMS web application will provide these services.

## 3. 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 that each Submission Chair is responsible for. The System Administrator will be able to configure a list of questions that users will be asking 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.

### **Vendor**

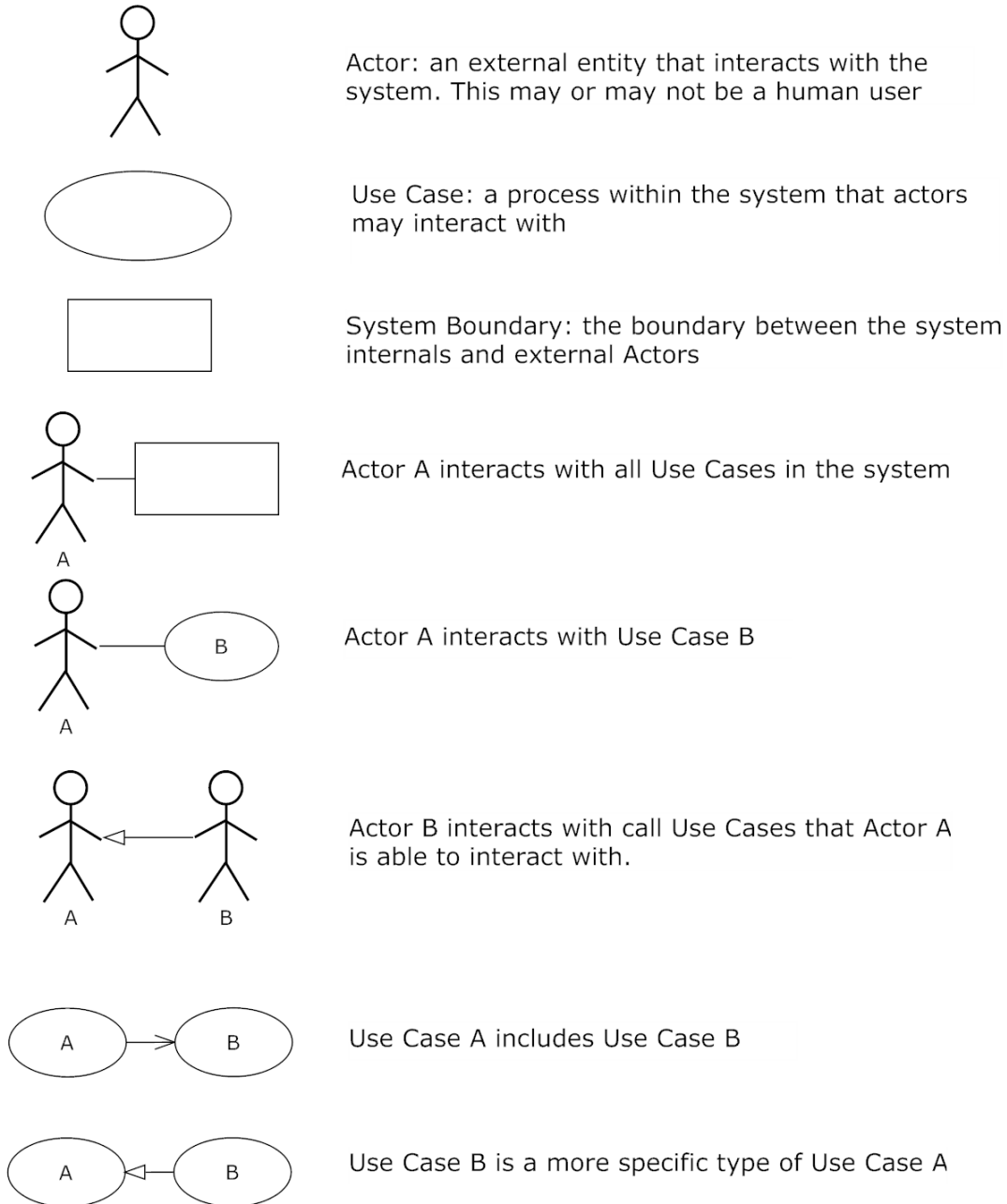
The conference allows Vendors to appear at the conference and set up displays for products. Vendors will use the WORMS application in the same way as a Conference Attendee. Vendors will pay a registration fee to attend the conference. Any organization required for the Vendors' displays will be handled outside of the WORMS application.



## 4. Use Case Diagram

The UML Use Case Diagram shows all the different users of the system, including both human and non-human. These are called actors. The diagram also shows the functions each user can perform called uses. Each user of WORMS is connected to different functions in the diagram by lines. Below is the legend of the WORMS UML diagram, followed by the actual diagram.

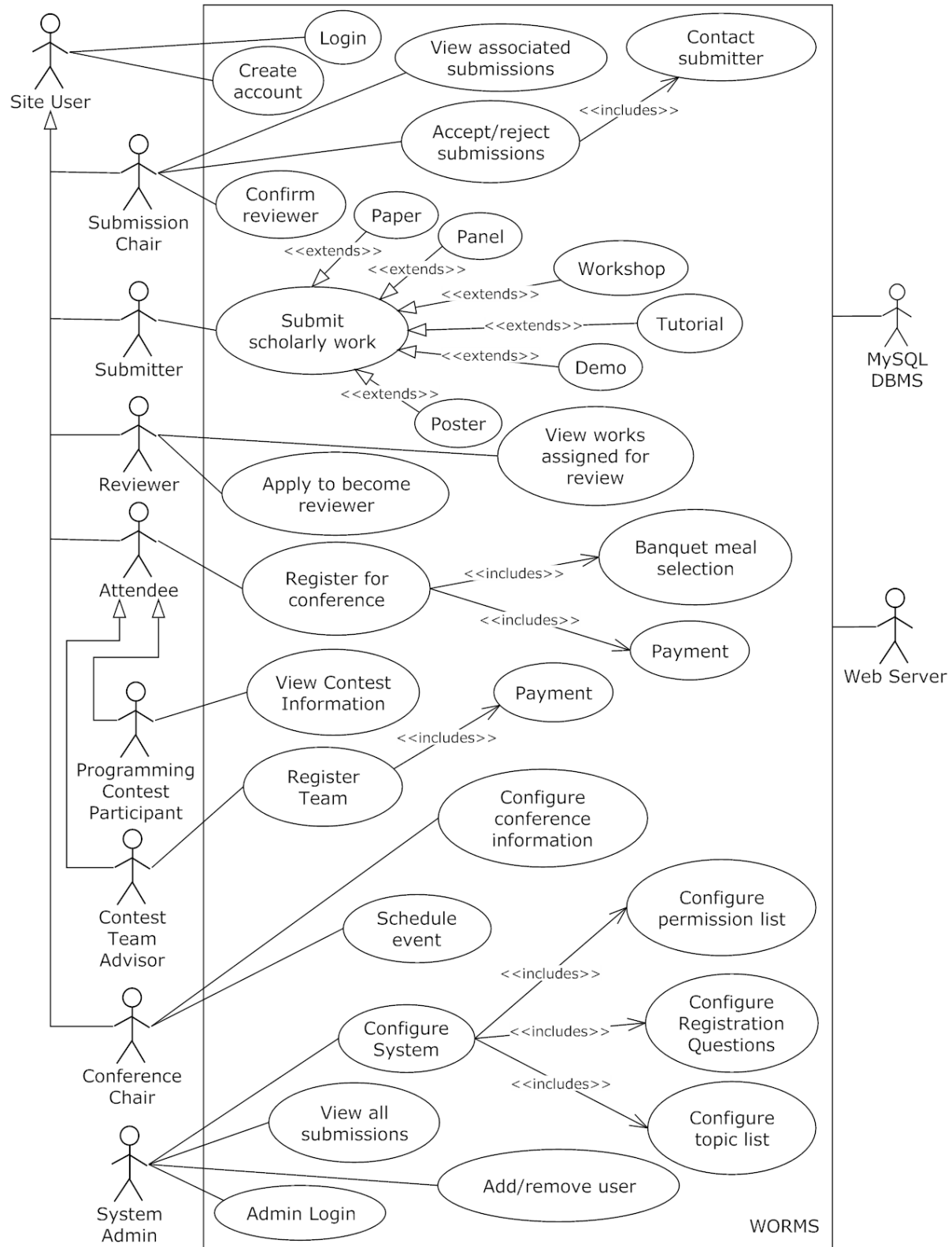
### UML Use Case Diagram Legend







UML Use Case Diagram

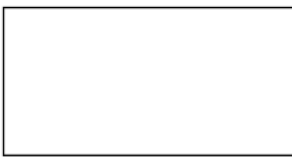




## 5. Data Flow Diagrams

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.

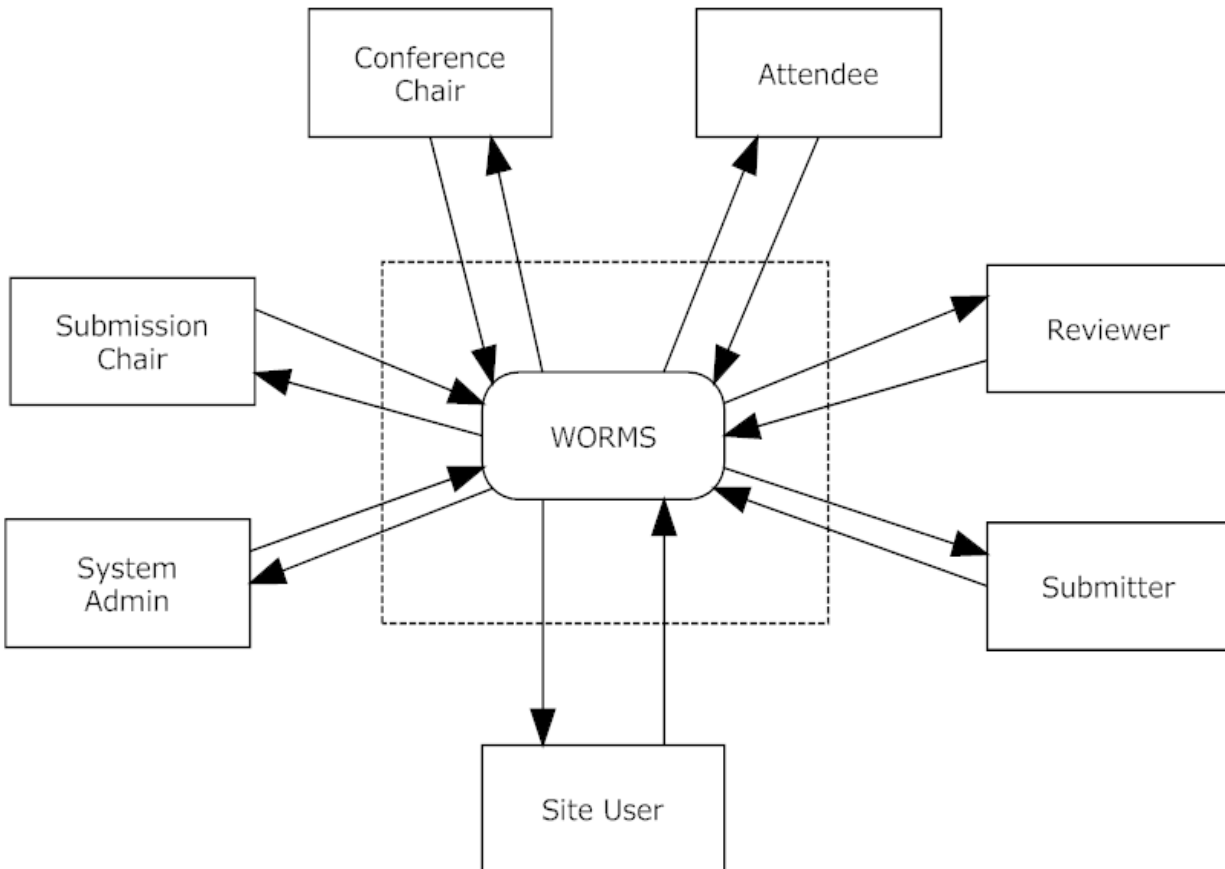


System boundary - the boundary between the system and external entities.



### Context Diagram

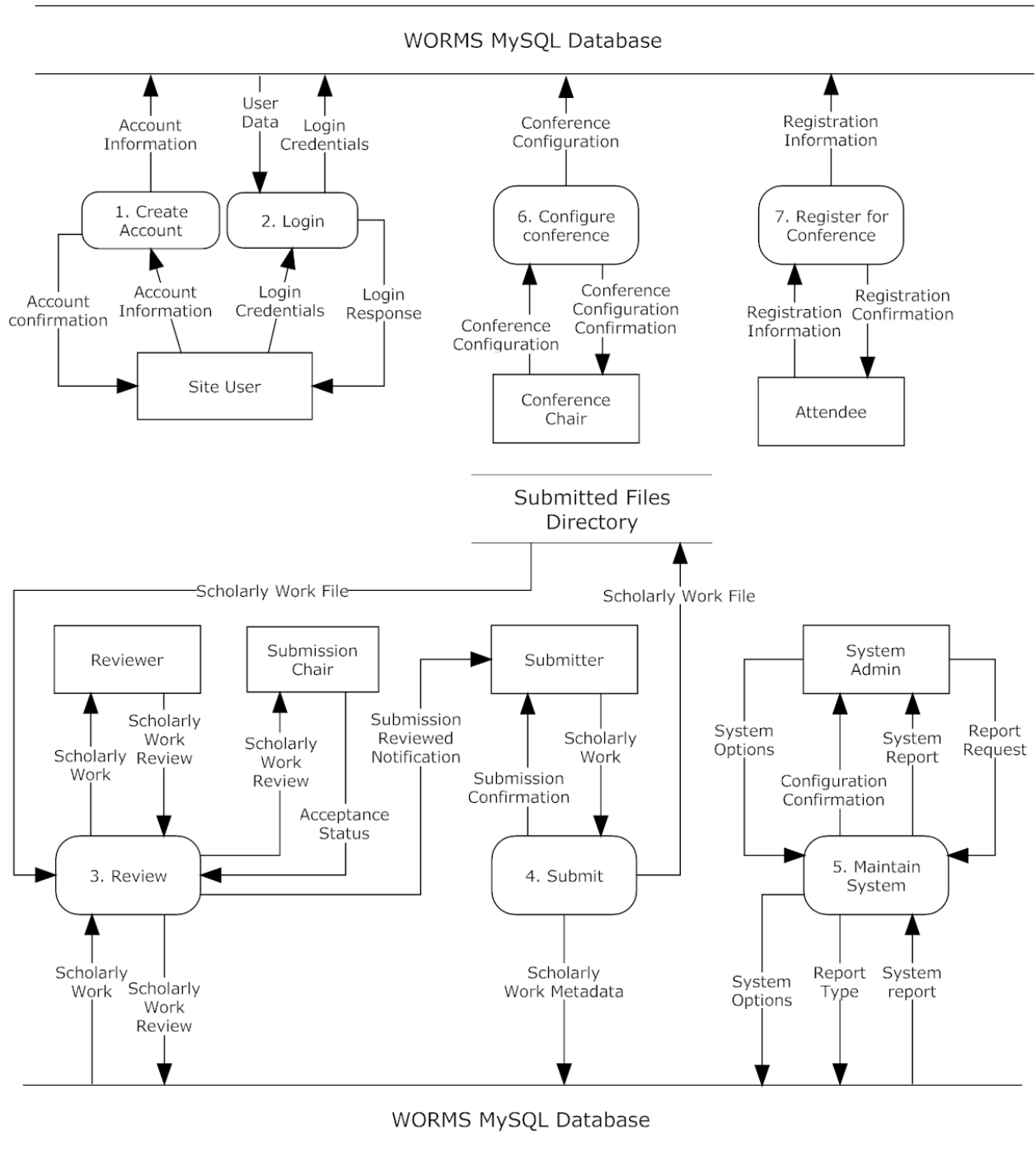
A Context Diagram is shown below. This is the general overview of each of the different entities interacting with the system.





### WORMS Level 0 Diagram

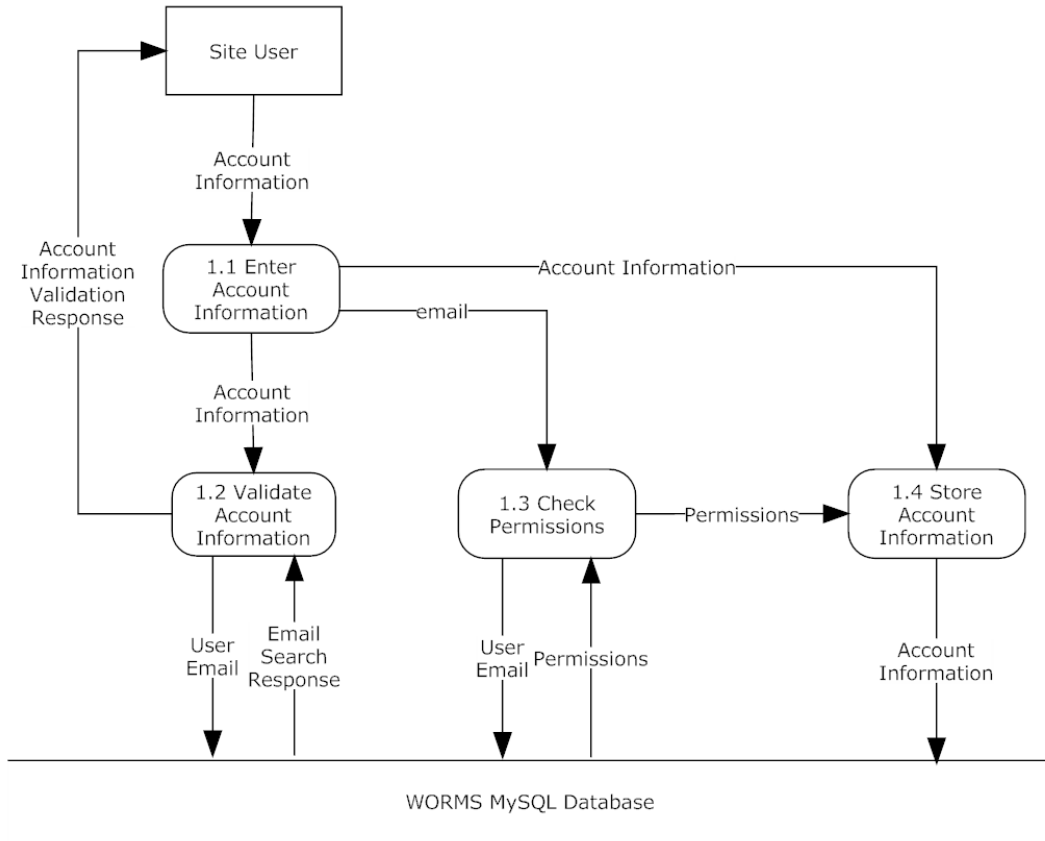
The Level 0 Diagram shows the major processes of the system and how they interact. Arrows are drawn to show the flow of data between the entities and processes.





**Process 1: Create Account**

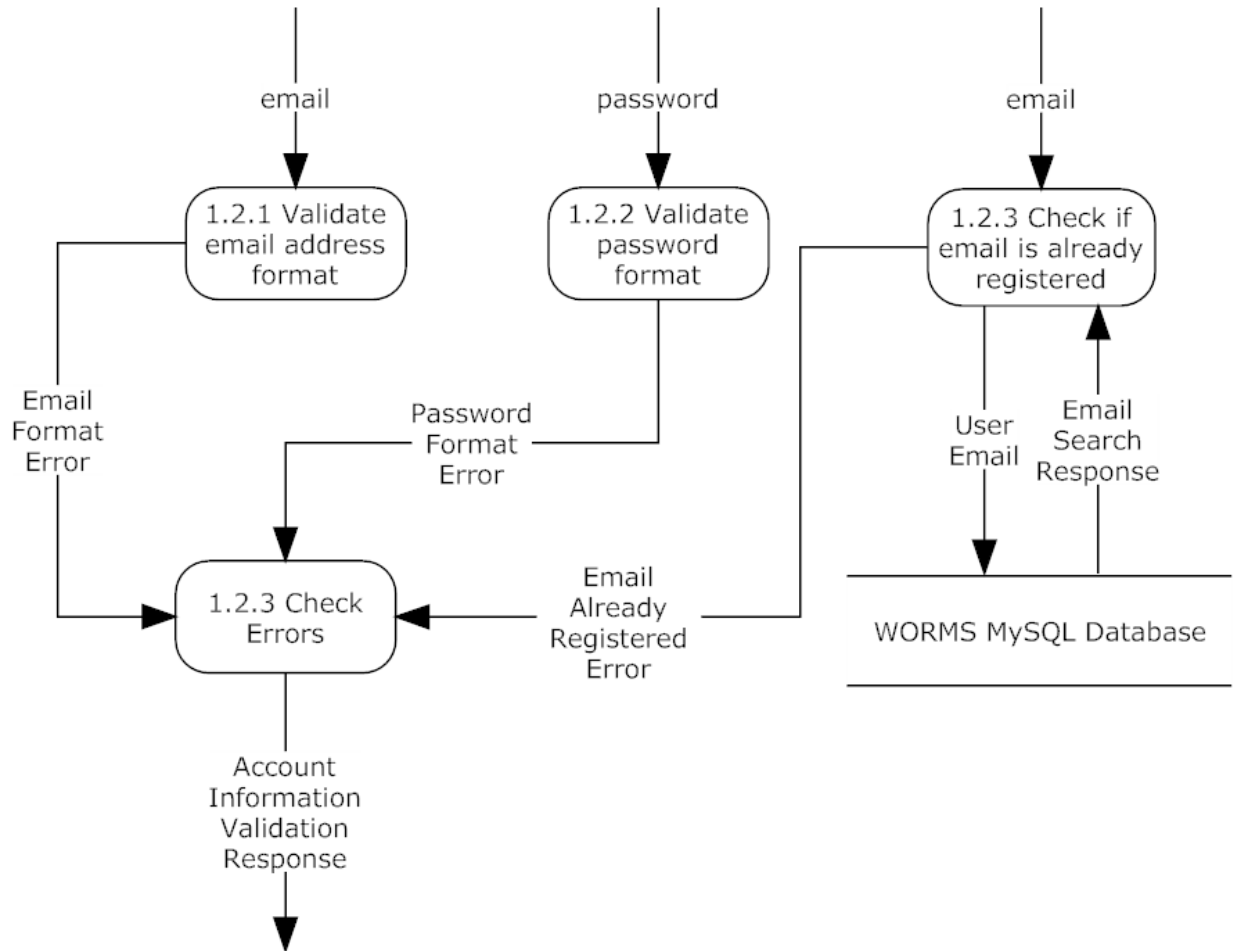
This Level 1 Diagram shows the Process 1: Create Account. This process is used to set up an account for a new user of the system.





**Process 1.2: Validate Account Information**

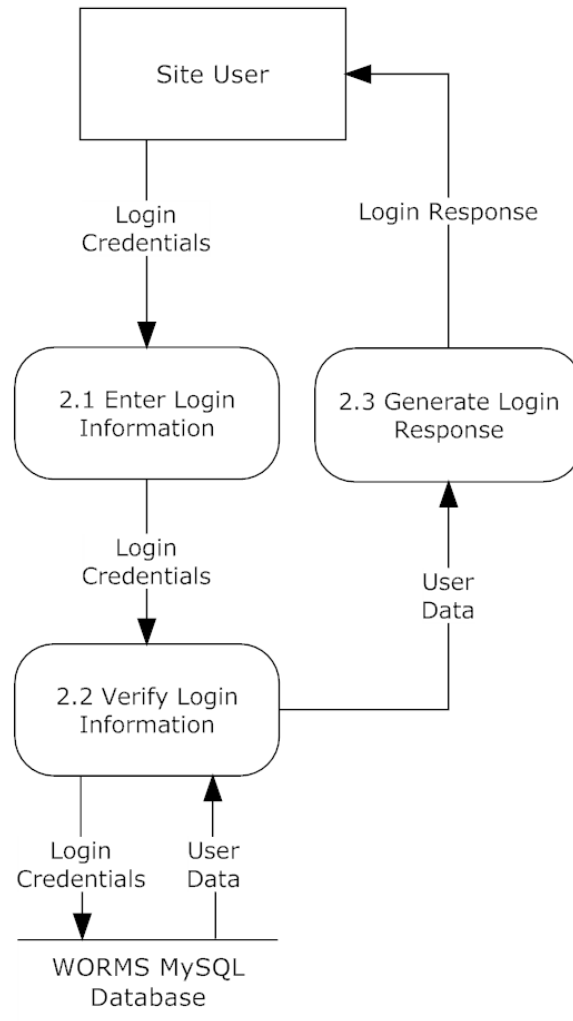
This Level 2 Diagram shows the Process 1.2: Validate Account information. This process is used each time an account is created to ensure that the information provided by the user is valid.





**Process 2: Login**

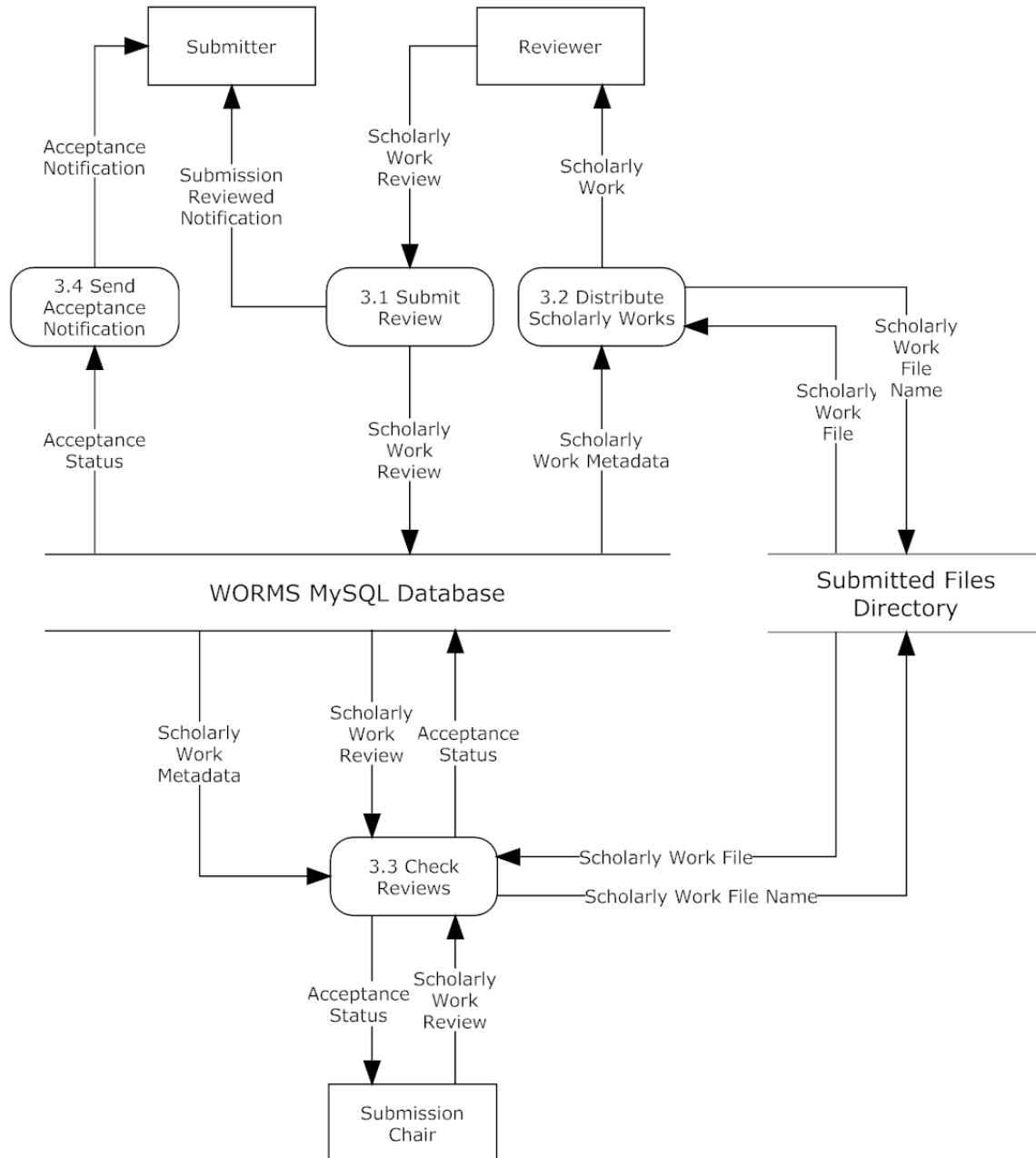
This Level 1 Diagram shows Process 2: Login. Each user interacts with this process in order to gain access to the main functionality of the system.





**Process 3: Review**

This Level 1 Diagram shows the Process 3: Review. This process involves the creation of reviews for each Scholarly Work and the determination of which Scholarly Works will be accepted to the conference.

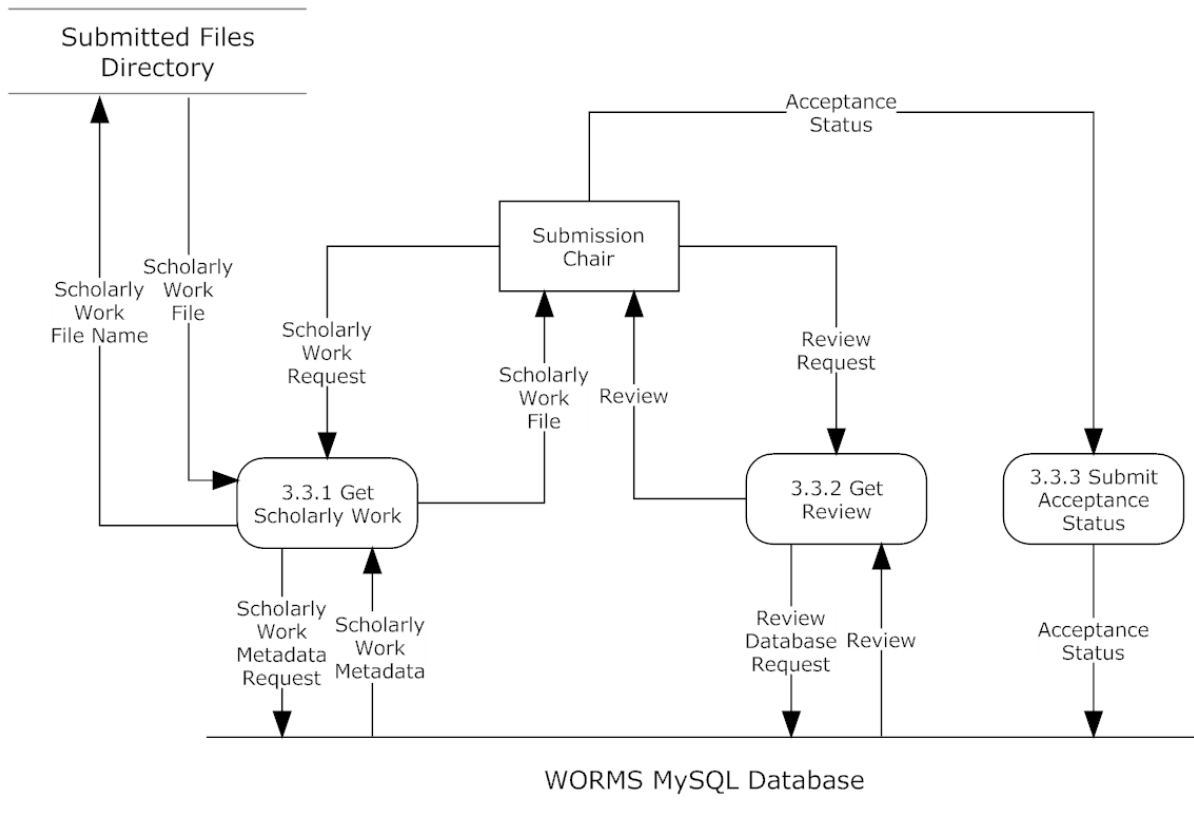






### Process 3.3: Check Reviews

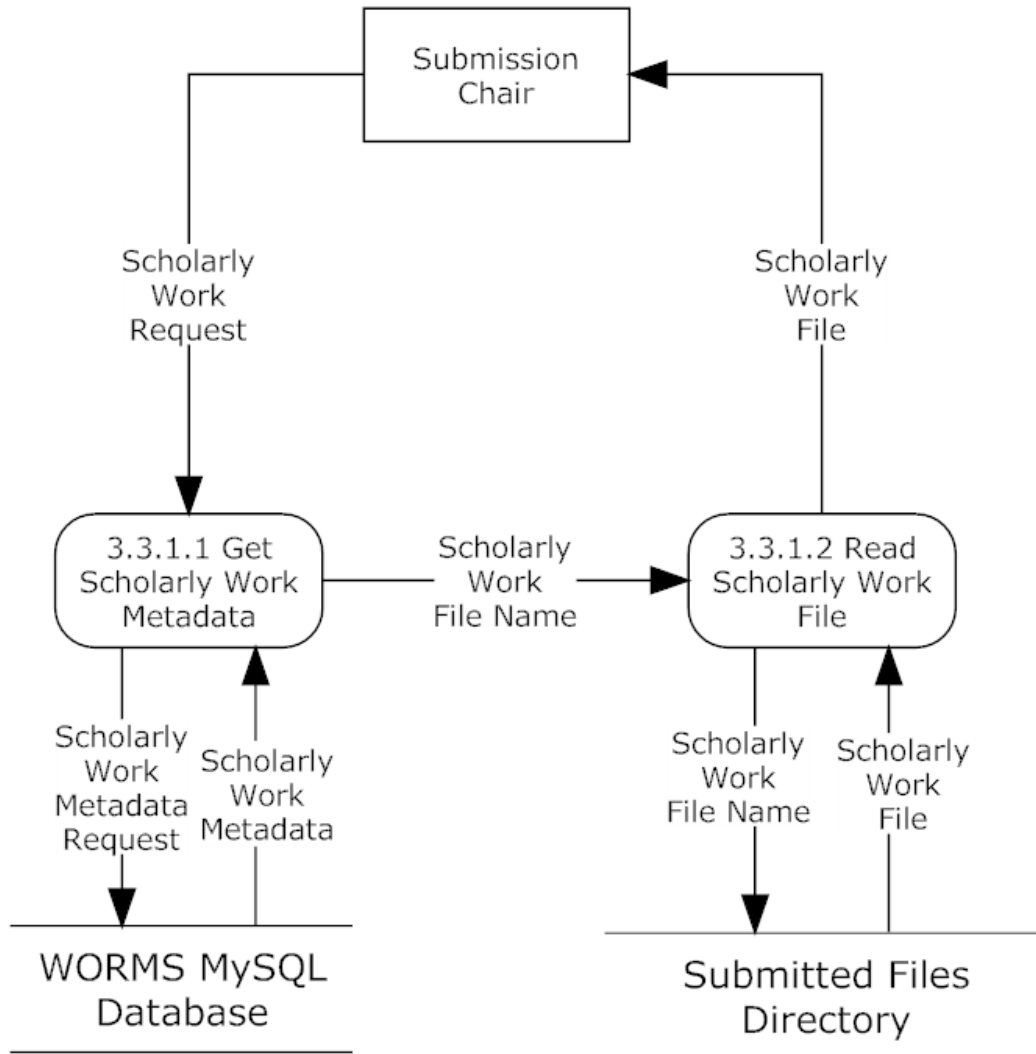
This Level 2 Diagram shows the Process 3.3: Check Reviews. This diagram shows the process of Check Reviews within the Review process. This process is used to determine which submitted scholarly works will be accepted into the conference.





**Process 3.3.1: Get Scholarly Work**

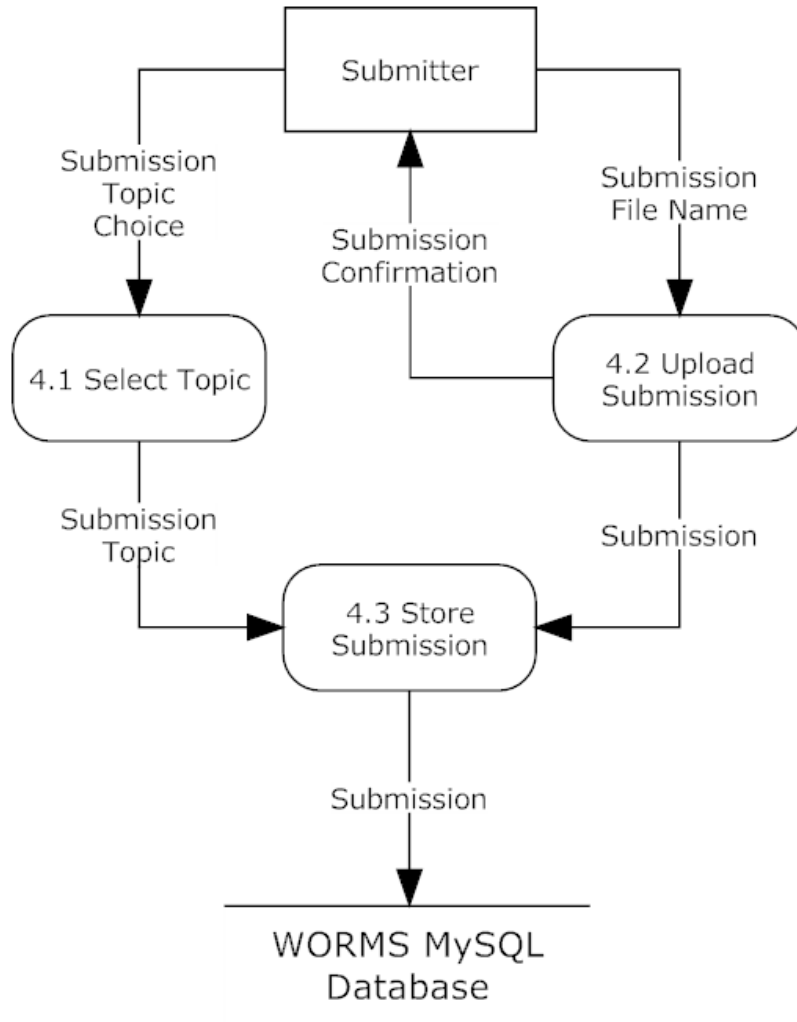
This Level 2 Diagram shows Process 3.3.1: Get Scholarly Work. This process is used to retrieve a Scholarly Work for examination by a Submission Chair.





**Process 4: Submit**

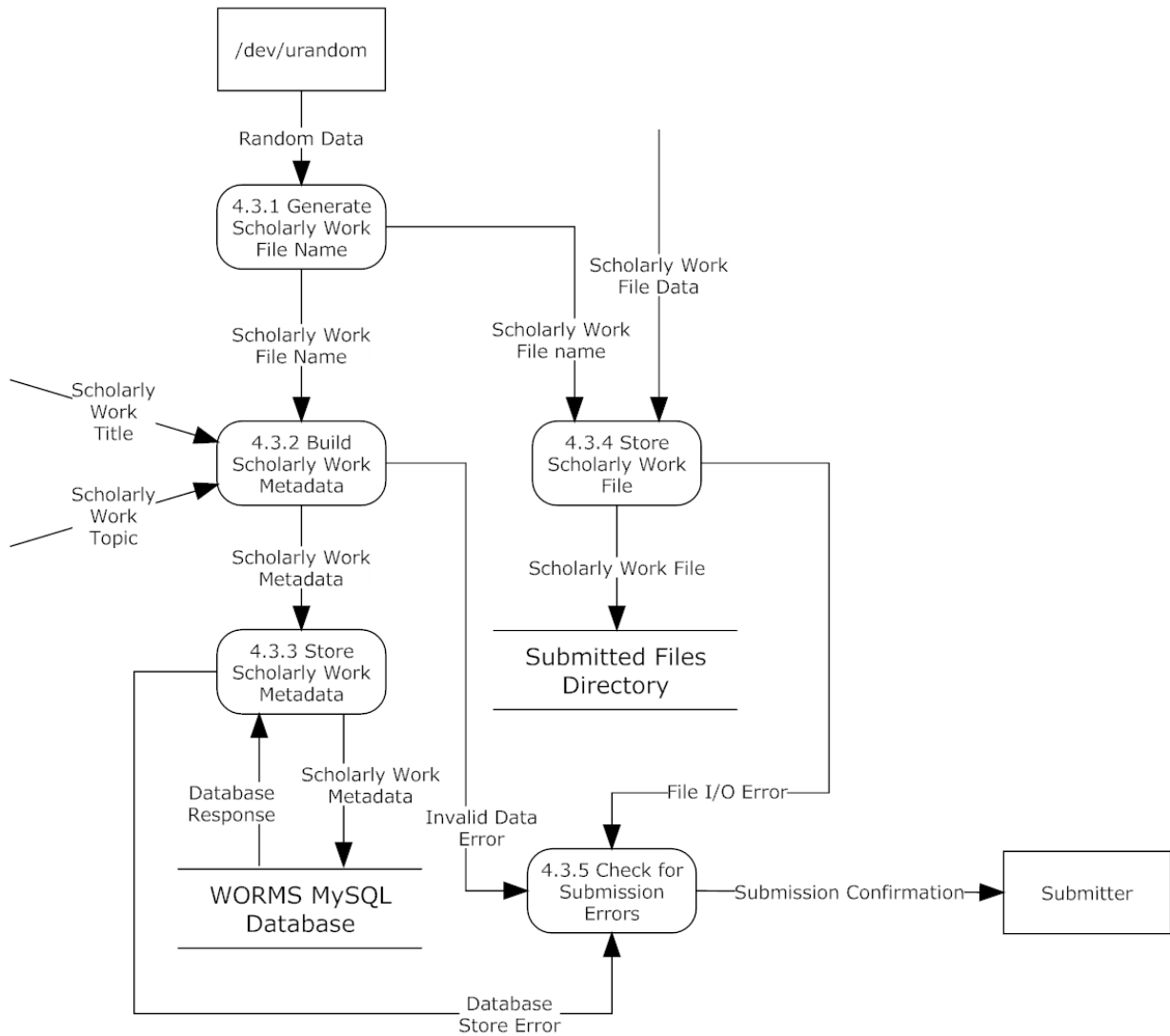
This Level 1 Diagram shows Process 4: Submit. This is the process of Scholarly Works being uploaded to the WORMS server and registered into the system.





### Process 4.3: Store Submission

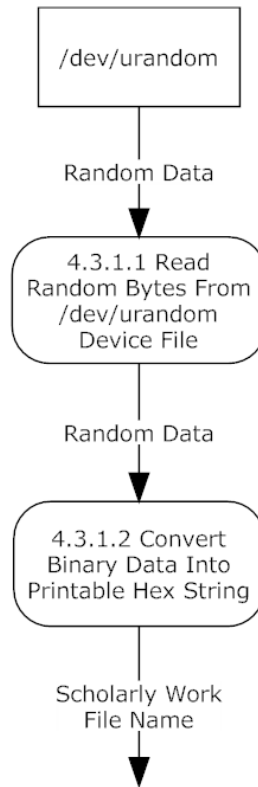
This Level 2 Diagram show Process 4.3: Store Submission. This process is used to store a Scholarly Work and the necessary metadata upon submission by a Submitter.





### Process 4.3.1: Generate File Name

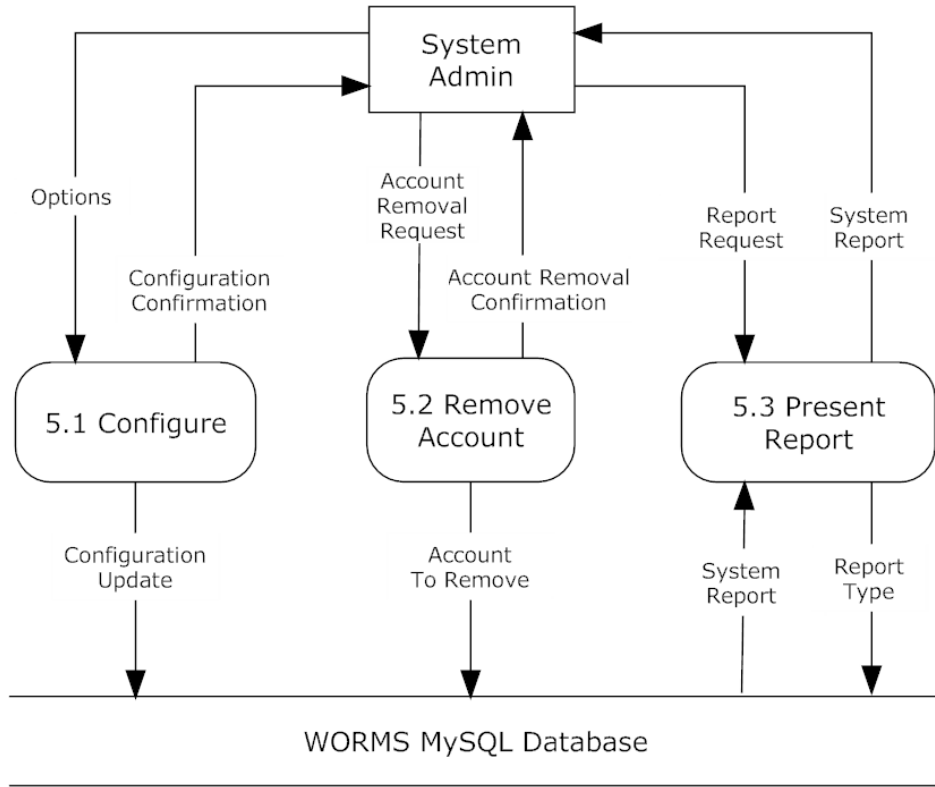
This Level 3 Diagram shows Process 4.3.1: Generate File Name. This process is used to generate a file name for the file where a submitted Scholarly Work will be stored.





### Process 5: Maintain System

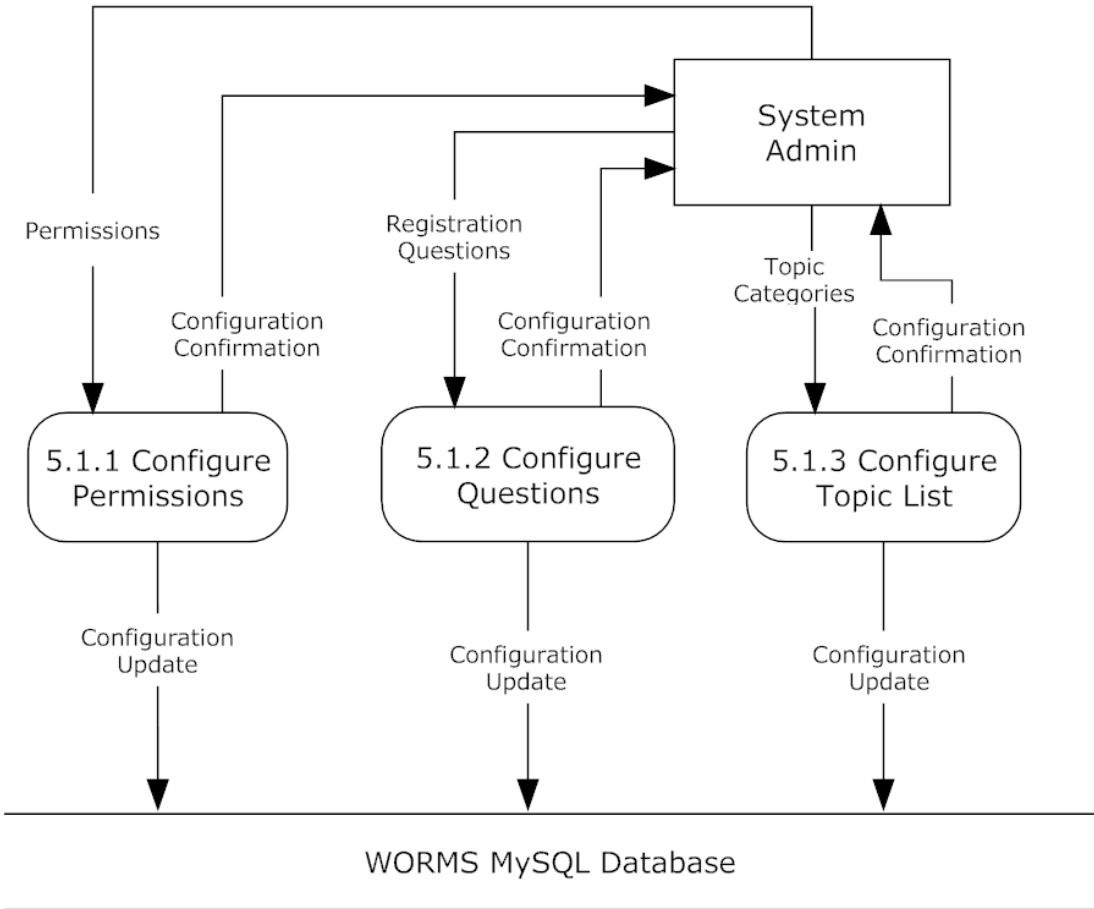
This Level 1 Diagram shows the Process 5: Maintain System. This is the process the System Administrator interacts with in order to initially set up the system and to keep it functioning properly.





### Process 5.1: Configure

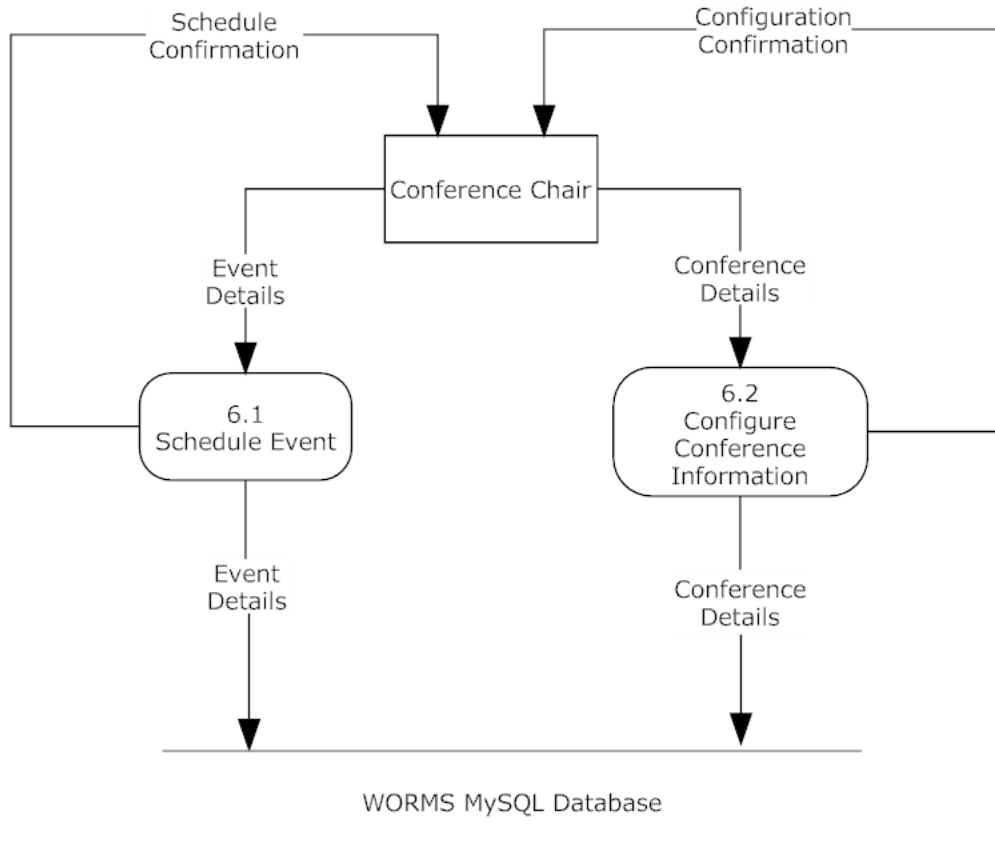
This Level 1 Diagram shows Process 5.1: Configure. This process is used when the System Administrator wishes to change system options and perform other administrative tasks.





### Process 6: Configure Conference

This Level 1 Diagram shows the Process 6: Configure Conference. This process allows the Conference Chair to schedule events and edit the conference information.

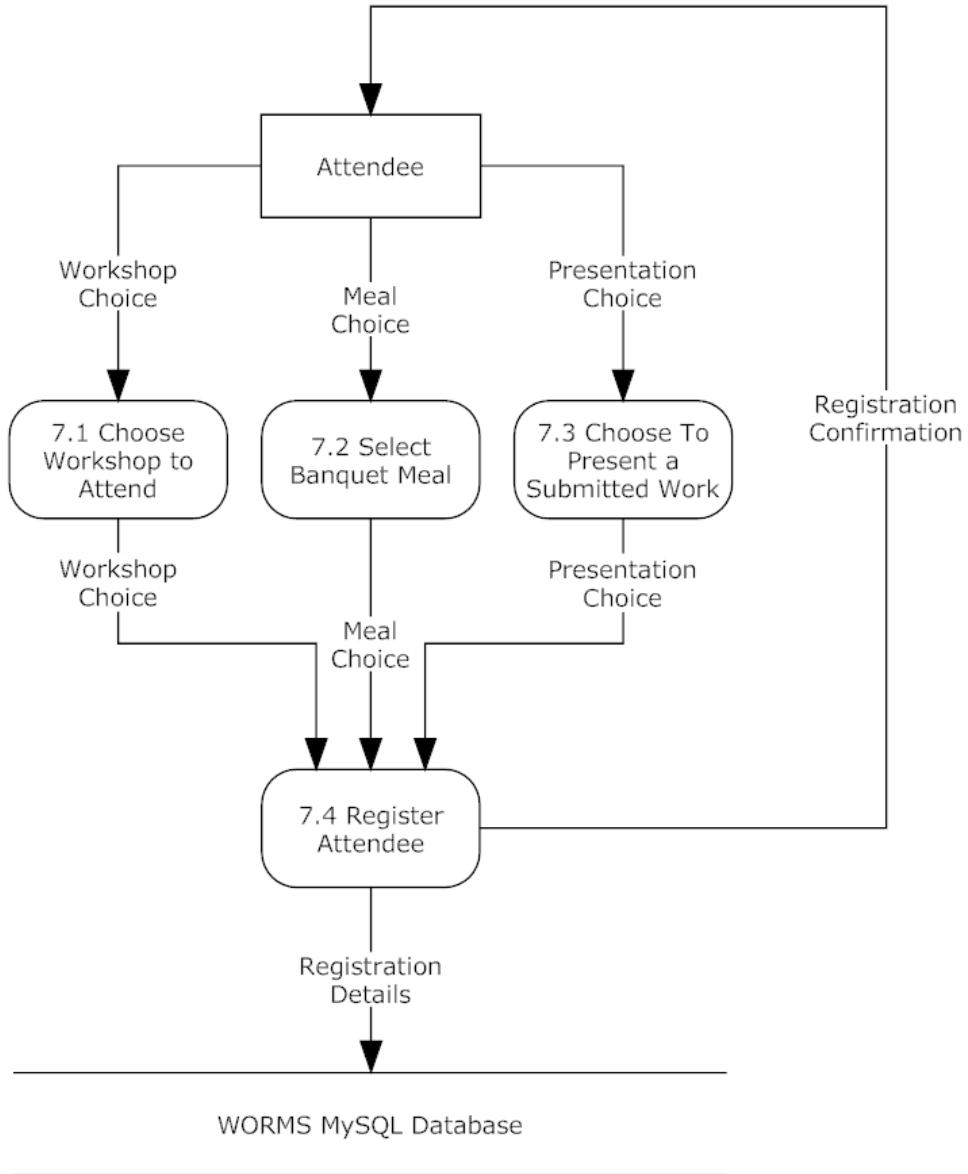






### Process 7: Register for Conference

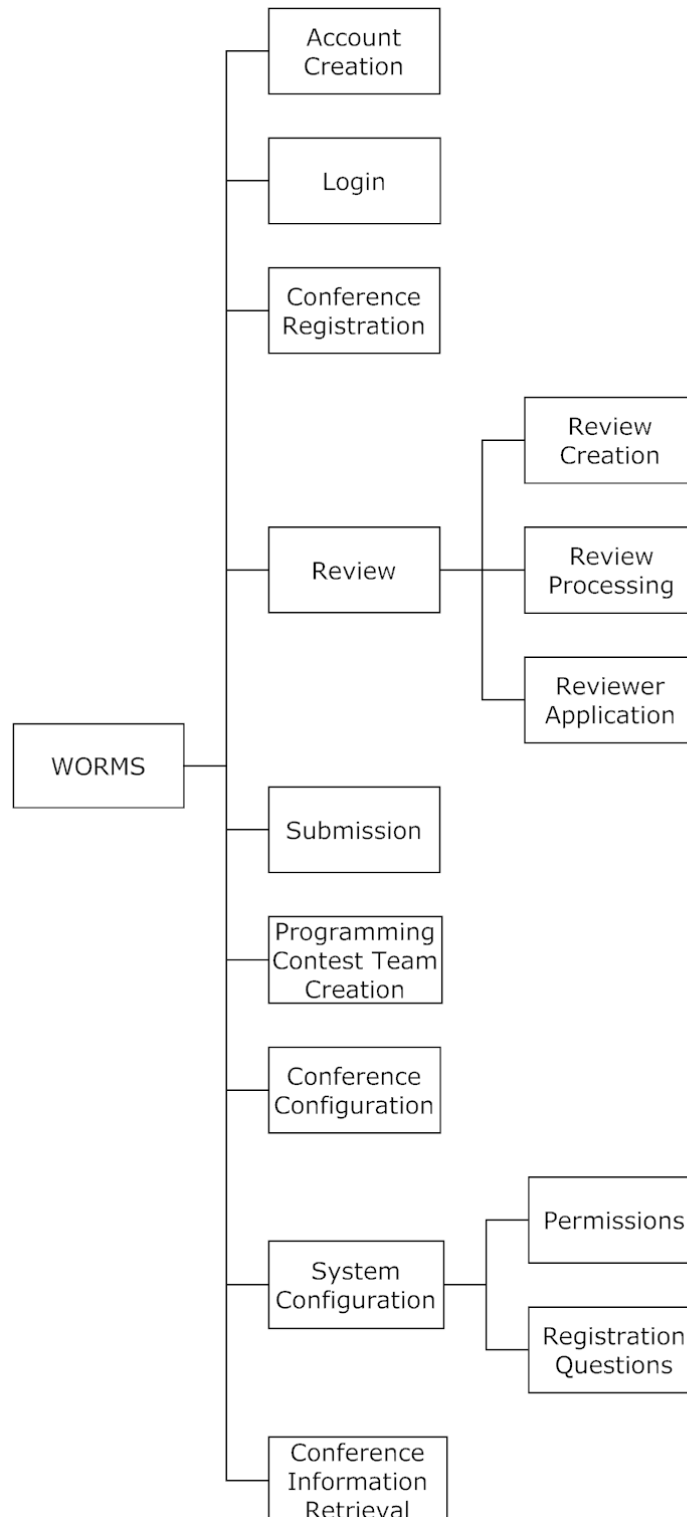
This Level 1 Diagram shows the Process 7: Register for Conference. This process allows a conference attendee to register for the conference.





## 6. Structure Diagram

This diagram models the hierarchical structure of the major components of the WORMS software.





## 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 view the final appearance of this information as a normal user will see it
  - This editing will take place in the same location on the web site where the final information will be presented to the user
- 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 download all submitted works
- 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
  - A confirmation email will be used to validate the 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 register to become an accepted Reviewer
  - Will be able to check whether or not credentials have been accepted yet
- Will be able to view scholarly works that have been submitted and assigned 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 a 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
  - A confirmation email will be used to validate the 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 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 in a file on the server
  - The name of the file containing the Scholarly Work will be stored in a specific folder with a randomly generated name. This file name will be associated with the scholarly work in the database.
  - Will be able to check whether or not the scholarly work has been reviewed
- Will receive a notification via email of submission acceptance or rejection

### **Programming Contest Team Advisor**

- Will be able to create an account using a valid email address
  - A confirmation email will be used to validate the 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 a programming team
  - Will be able to provide email addresses of the students participating in the programming contest to be stored on the database
  - Each account created with an email address specified by a Programming Contest Team Advisor will be associated with the team created by that Programming Contest Team Advisor
  - Will pay the fee necessary for registering a programming contest team



### **Programming Contest Participant**

- Will be able to create an account using a valid email address as specified by a Programming Contest Team Advisor on a team email list
  - A confirmation email will be used to validate the 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 view information about the programming contest such as dates, times, and contest environment information

### **Conference Attendee**

- Will be able to create an account using a valid email address
  - A confirmation email will be used to validate the 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 select a workshop to attend at the conference
- Will be able to view information about hotels and nearby attractions
- Will be able to register for the conference
- Will be able to view a schedule of events at the conference

### **Other**

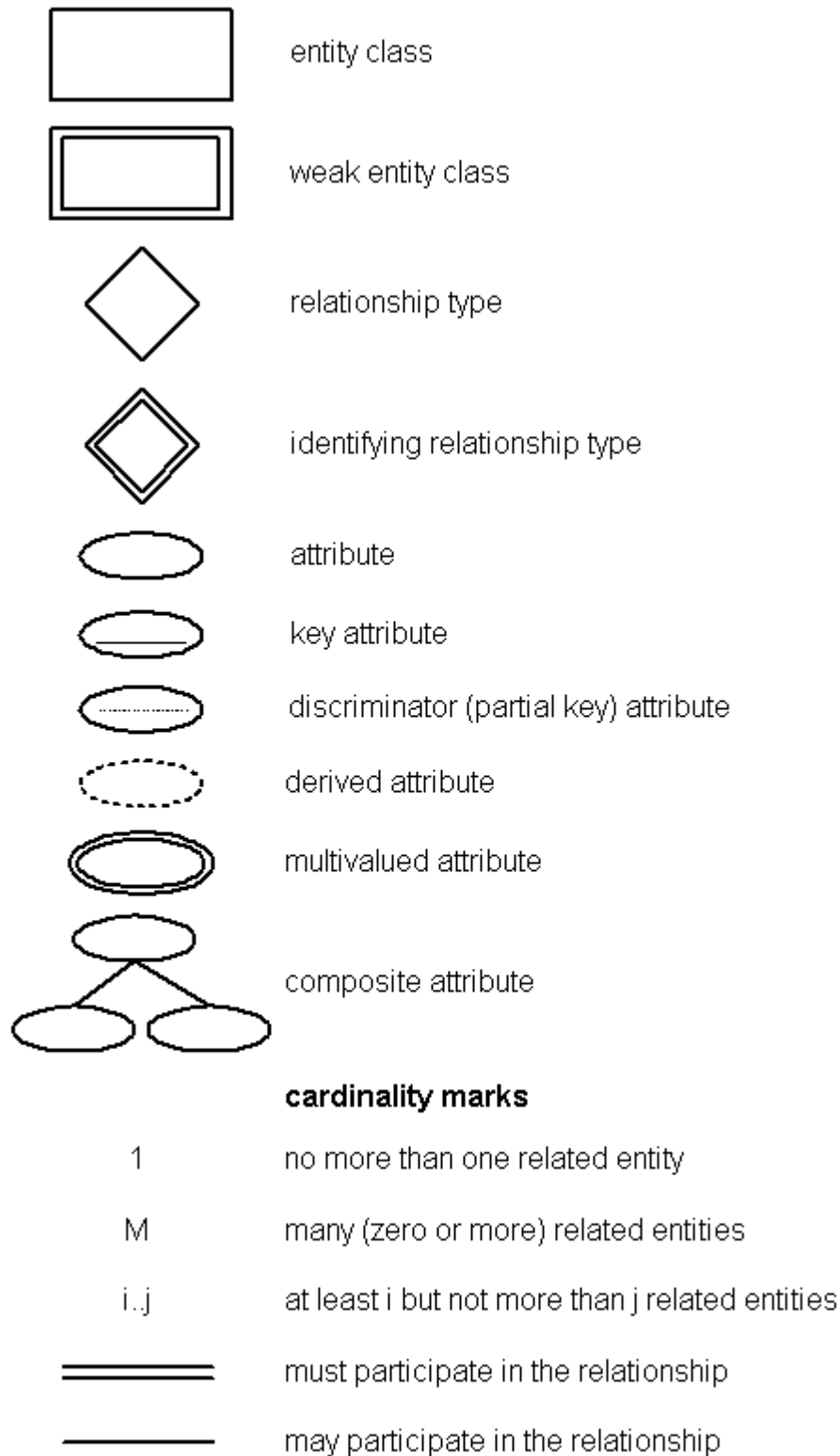
- Users who are not Conference Chairs, Submission Chairs, or the System Administrator will be able to apply to become a Reviewer or Submit a Scholarly Work
- WORMS must function properly in the latest versions of Google Chrome, Mozilla Firefox, and Safari

## **8. Non-functional Requirements**

Non-functional requirements specify the properties of the system that are not necessarily specific features to be implemented. The only non-functional requirement for the WORMS software is that the software must be user friendly.

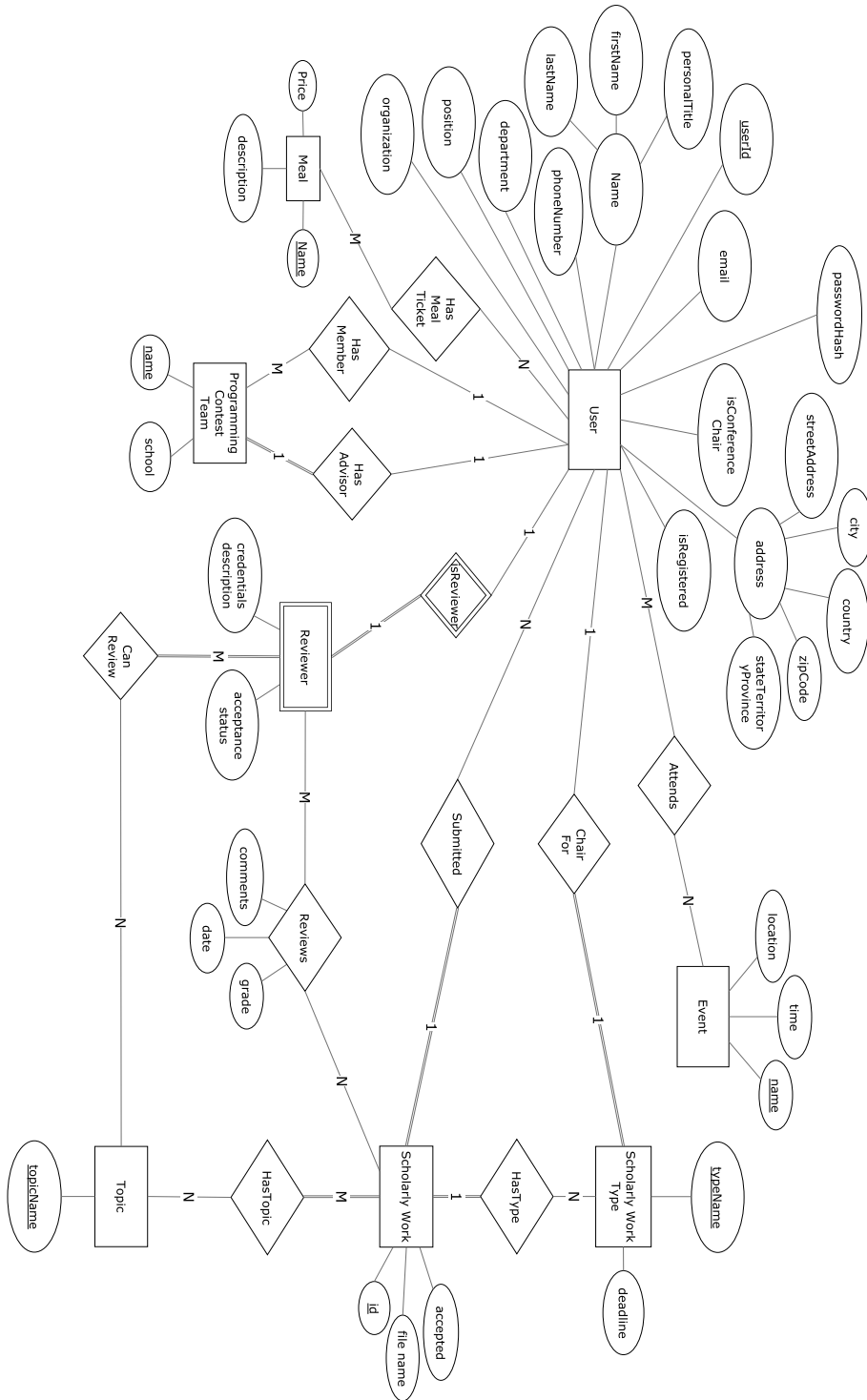
## 9. Entity-Relationship Diagram and Table Relations

### ERD Legend





### Entity-Relationship Diagram







## Table Relations

The full specification of the schema for all tables in MySQL Data Description Language is available in Appendix E.

## 10. Logical Data Dictionary

The WORMS Data Dictionary is in the appendix C. Following are the Columns and its descriptions for the Data Dictionary.

Columns	Description
Data Name	Name of the Data Entity
Can be NULL	If the data can be a NULL type or not in the database
Data Type	Database type used to store the data entity if it is stored in the database
Data Format	Size or other restrictions on the storage of a Data Entity
Example Acceptable Input	Examples of acceptable input to be stored in the Data Entity
Bad Input	Examples of invalid input for the Data Entity
Applicable to	Processes which the data are applicable to
Description	Textual description of the data

## 11. Prototype Screens

Appendix F contains images showing the preliminary layout of several web pages in the WORMS application.



## 12. Testing Plan

### Overview and Strategy

The testing procedures specified here will be used to determine whether or not the WORMS application meets each functional requirement. The WORMS web application will be tested in the latest versions of the following web browsers: Google Chrome, Mozilla Firefox, and Safari. Unit Tests will test individual components of the system. Integration tests will be used to ensure that subsets of the system components work together properly. System tests will be performed as integration tests of the entire set of system components. System tests will verify that the functional requirements are met. Nonfunctional requirements will be verified as much as possible through regular interaction with the system.

### Unit Tests and Test Cases

A unit test is used to determine whether or not an individual component of a software system is functioning correctly. Each unit test is made up of a set of test cases. Test Cases have input actions, steps to be performed, the state before test, and expected results for the test case. If the result observed for a test case is the expected result, then the test passes. If all test cases pass, the software component being tested is considered to be functioning properly.

Unit tests for these WORMS modules are given in Appendix D.

- Login
- Create Account
- Submit
- Review
- Conference Configuration
- Programming Contest Team Creation
- Scheduling
- System Admin Options
- Conference Registration

### Integration Test

An integration test is used to determine whether or not a subset of the components, or units, making up a software system interact with each other properly. When a unit is implemented or modified it must be unit tested to ensure that it functions properly. Once the unit has been determined to be functioning correctly, the related units must also be tested. The units related to a given unit are specified in the unit test directory.



### **System Test**

A system test is an integration test of all components of the software system. A system test determines whether or not the functionality specified in the Functional Requirements Inventory exists in the software system.



### Acceptance Criteria

The acceptance criteria for the WORMS software include the following functional requirements:

	<b>System Administrator</b>
Yes / No	Able to add email addresses to a list of verified conference chairs
Yes / No	Able to add email addresses to a list of verified submission chairs
Yes / No	Able to specify the types of submissions a submission chair is responsible for
Yes / No	Able to view a list of all submitted works
Yes / No	Able to add topics to the list of topics a scholarly work may be categorized into
	<b>Conference Chair</b>
Yes / No	Able to create an account using an email address that has been specified by the System Administrator
Yes / No	Able to add and remove events from a schedule
Yes / No	Able to edit conference information including hotels, parking information, and sponsors
Yes / No	Able to configure the available banquet meal selections
	<b>Submission Chair</b>
Yes / No	Able to create an account using an email address that has been specified by the System Administrator
Yes / No	Able to specify the deadline for submission of scholarly works
Yes / No	Able to view a list of submitted scholarly works
Yes / No	Able to download each submitted scholarly work
Yes / No	Able to view reviews given to each scholarly work
Yes / No	Able to specify whether or not each scholarly work is accepted
Yes / No	Able to view credentials of users who have applied to become reviewers
Yes / No	Able to specify whether or not a user is allowed to review scholarly works



	<b>Reviewer</b>
Yes / No	Able to create an account
Yes / No	Able to submit credentials for review by a Submission Chair
Yes / No	Able to check whether or not his/her credentials have been accepted by a Submission Chair
Yes / No	Able to view scholarly works that have been assigned to him/her for review
Yes / No	Able to fill submit a review form for each assigned scholarly work
	<b>Submitter</b>
Yes / No	Able to create an account
Yes / No	Able to select the type of scholarly work he/she wishes to submit
Yes / No	Able to upload a PDF file with a paper or abstract for a submission
Yes / No	Able to upload an anonymous version of the PDF submission in addition to the normal version
Yes / No	If valid submission options are provided, the Submitter's PDF files will be uploaded and stored on the WORMS server
Yes / No	A notification email of acceptance or rejection will be received for each scholarly work
	<b>Programming Contest Team Advisor</b>
Yes / No	Able to create an account
Yes / No	Able to create a programming contest team
Yes / No	Able to create a list of email addresses indicating the programming contest team members
	<b>Programming Contest Participant</b>
Yes / No	Able to create an account
Yes / No	After his/her advisor has added he/she to the team, the participant will see an indication on the programming contest team that he/she is a member of a team
Yes / No	Able to view information about the programming contest such as dates, times, and contest environment information



	<b>Conference Attendee</b>
Yes / No	Able to create an account
Yes / No	Able to choose workshops to attend while at the conference
Yes / No	Able to select a meal choice for each meal occurring during the conference
Yes / No	Able to choose to present an accepted scholarly work
Yes / No	Able to register to attend the conference
	<b>Other</b>
Yes / No	Any user who visits the WORMS web site will be able to view general conference information and the schedule of events
Yes / No	Users who are not Conference Chairs, Submission Chairs, or the System Administrator will be able to apply to become a Reviewer or submit a Scholarly work
Yes / No	All users are able to log in and log out of the system
Yes / No	Upon account creation email addresses will be validated using a validation email
Yes / No	The software functions properly in the latest versions of Google Chrome, Mozilla Firefox, and Safari

### Non-functional Requirements

The only non-functional requirement for the WORMS software is that the software must be user friendly. This requirement will be verified through constantly using the software throughout the development process and ensuring that the software's functionality is clear and easily accessible. At various stages of development the software will also be demonstrated to the client. After interacting with the software, the client will provide feedback about the user friendliness of the software and changes will be made where necessary.



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

CCSCNE – Consortium for Computing Sciences in Colleges Northeast

CSS – Cascading Style Sheets, language used to style web pages

DBMS – Database Management System

Go – A compiled, statically and strongly typed programming language developed by Google

I/O – Input and output

HTML – Hypertext Markup Language, language used to structure web pages

JavaScript – browser side scripting language

Linux – an open source operating system kernel

MySQL – a database management system

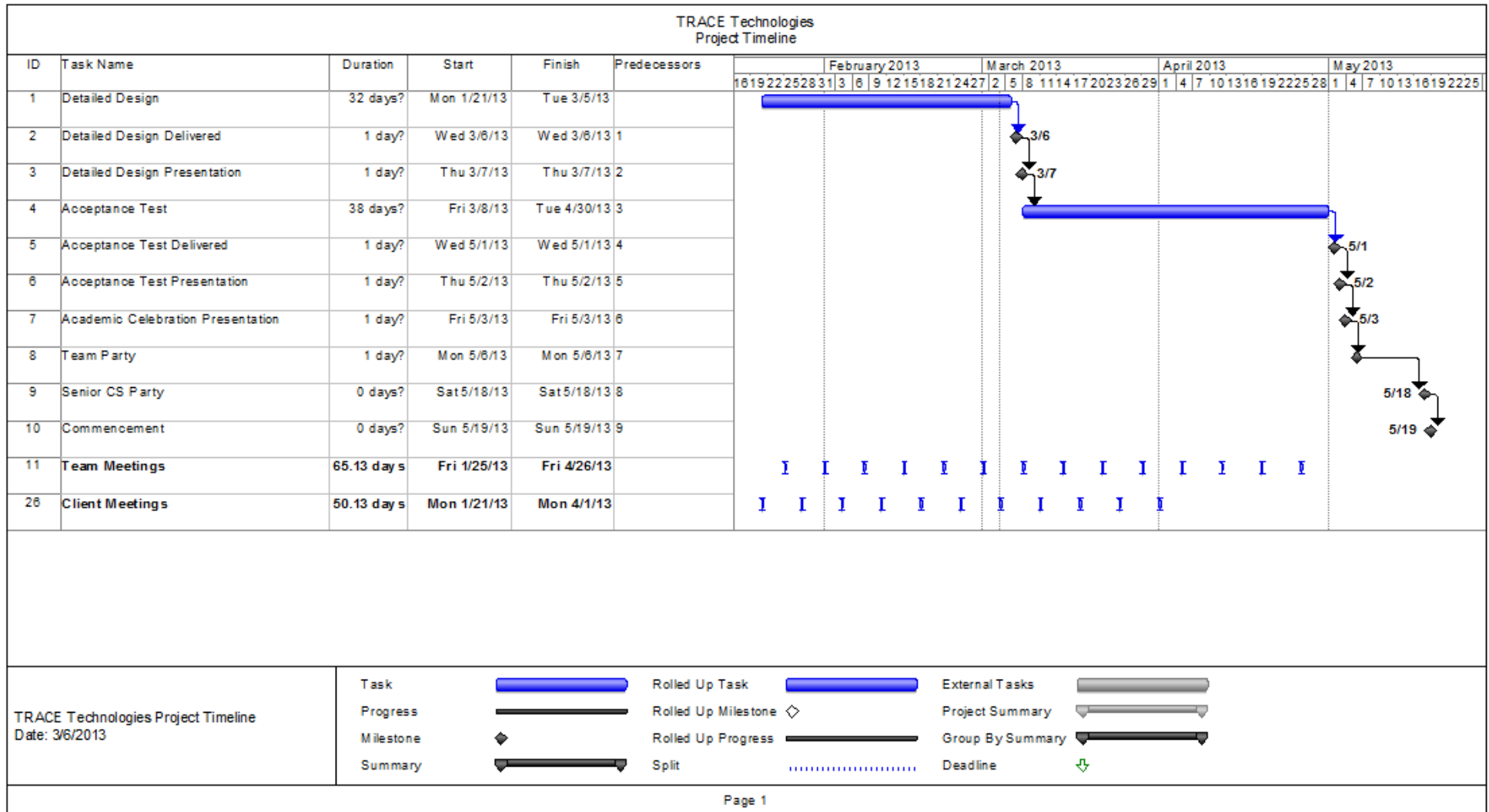
Scholarly Work – one of the following: poster, paper, workshop, tutorial, panel

UML – Unified Modeling Language

WORMS – Web Organization Registration Management System



## 15. Appendix B: Project Timeline







## 16. Appendix C: Data Dictionary

WORMS DATA DICTIONARY							
Data Name	Can be NU	Data type	Data Form	Example Acceptable In	Bad Input	Applicable to	Description
address	No	VARCHAR(100)	1 - 100 chars	515 Loudon Road	empty input, >100 chars	Create account	The account ow ner's address
city	No	VARCHAR(100)	1 - 100 chars	Loudonville	empty input, >100 chars	Create account	The account ow ner's city
countryIndex	No	INT	0-195 chars			Create account	The index of the account ow ner's country
department	Yes	VARCHAR(100)	1 - 100 chars	Computer Science	empty input, >100 chars	Create account	The account ow ner's school department
email	No	VARCHAR(100)	3 - 100 chars	siena@siena.edu	sienasiena.edu, @	Create account	The account ow ner's email address
eventLocation	No	VARCHAR(100)	1 - 100 chars	Roger Bacon 202	empty input, >100 chars	Register	Location of the scheduled event
eventTime	No	DATETIME		Time during the conference	Time not during the conference	Register	Time of the scheduled event
eventTitle	No	VARCHAR(100)	1 - 100 chars	Welcome speech	empty input, >100 chars	Register	Name of the scheduled event
firstName	No	VARCHAR(100)	1 - 100 chars	Darren, Joe	empty input, Darren123	Create account	The account ow ner's first name
isConferenceChair	No	BOOL		True/False	N/A	Create account, Configure	Is the user a conference chair
isPresenting	No	BOOL		True/False	N/A	Register, Submit	If the user has submitted a w ork that w as accepted, w ill that user present the w ork
isProrammingContest TeamLeader	No	BOOL		True/False	N/A	Register	Is the user registered as a programming contest team leader
isRegistered	No	BOOL		True/False	N/A	Register	Is the user registered to attend the conference
lastName	No	VARCHAR(100)	1 - 100 chars	Lim, Lederman	empty input, Lim123	Create account	The account ow ner's last name
mealSelection	No	INT	0 - (number of choices -1)	1	-1	Register	An identifier indicating the meal selection
normalFee	No	DECIMAL(10,2)	> 0	30,2	0,-1	Configure Conference,	Normal conference attendance fee
organization	Yes	VARCHAR(100)	1 - 100 chars	Siena College	empty input	Create account	The account ow ner's organization or company
passw ord	No	string	8 - 32 chars	mypassw ord123	pass, empty input	Create account, Login	User's passw ord
passw ordHash	No	BINARY(32)	32 bytes	any 32 bytes	N/A	Create account, Login	The account's hashed passw ord (SHA256)
personalTitle	No	VARCHAR(10)	1 - 10 chars	Dr., Mr., Mrs.	longer than 10 chars, Dr1234&	Create account	The account ow ner's personal title



Data Name	Can be NULL	Data type	Data Form	Example Acceptable Input	Bad Input	Applicable to	Description
phoneNumber	Yes	VARCHAR(20)	10 - 20 chars	5183219999, 518-321-9999	321-9999, 1234567	Create account	The account owner's phone number
position	No	VARCHAR(100)	1 - 100 chars	Associate Professor	\$***-#/	Create account	The account owner's academic title or professional position
programmingContestFee	No	DECIMAL(10,2)	> 0	50	0,-1	Configure Conference, Register	Programming contest attendance fee
programmingContestTeamId	No	VARCHAR(100)	1 - 100 chars	Team 1		Register	Name of program contest team
review	No	TEXT	Text data max 1 mb	Any ASCII text data < 1mb		Review	Text data for a review written by a reviewer for a scholarly work
reviewerId	Yes	INT	0 - (number of reviewers - 1)	0, 1, 2, NULL	number greater than the number of reviewers	Review	The ID of the reviewer data associated with this user if the user is a reviewer
scholarlyWorkAccepted	No	BOOL		True/False	N/A	Submit, Review	Whether or not a scholarly work has been accepted
scholarlyWorkDeadline	No	DATETIME		12/07/2012	12/07/2011	Submit, Review	A specified deadline for submission
scholarlyWorkFile	N/A	File on disk	max 20 mb			Submit, Review	A file uploaded by a submitter
scholarlyWorkFileName	No	CHAR(64)	64 hex digits	7d666e82dc53680ff3cc050071a49a00286af4f6ee	mission.pdf, Submission	Submit, Review	Name of the file used to store a scholarly work. Automatically generated.
scholarlyWorkTitle	No	VARCHAR(50)	1 - 50 chars	Some Title	empty input, >50 chars, contains nonprintable	Submit, Review	Title of a submitted work
scholarlyWorkTopicId	No	INT	0 - (number of topics - 1)	0,1,2	#####	Submit, Review	An identifier indicating the topic of a work
scholarlyWorkTypeId	No	INT	0 - 6	1,2,3,4	-1,10	Submit, Review	An identifier indicating the type of a work
stateTerritoryProvince	No	VARCHAR(100)	1 - 100 chars	New York	empty input, > 100 chars	Create account	The account owner's state
studentFee	No	DECIMAL(10,2)	> 0	20,5	0,-1	Configure Conference, Register	Student attendance fee
submissionChairId	Yes	INT	0 - (number of submission chairs - 1)	0,1,2, NULL	number greater than the number of submission chairs	Create account, Review	The ID of the submission chair data for this user if the user is a submission chair
userId	No	INT	0 - (number of users - 1)			All	Unique ID linked to a user. User ID 0 is the system admin.
zipCode	Yes	CHAR(5)	5 chars	12211	empty, 1234, 1	Create account	The account owner's zip code
workshopSelection	Yes	INT	0 - (number of workshops - 1)	0,1,2,NULL	number > number of workshops	Register	The ID of the workshop a user registering for the conference wishes to attend
workshopTitle	No	VARCHAR(100)	1 - 100 chars	Computer Science Education	empty input, >100 chars	Register	The title of a workshop that attendees may register for.





**WORMS**

**Login**

To allow a user to login to the system

Test Cases								
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments
F	1,001	Empty email field	Leave email field empty	Fill in password field and click login button	Empty Form	Error Message: "Please enter your email"		
F	1,002	Empty password field	Leave password field empty	Fill in email field and click login button	Empty Form	Error Message: "Please enter your password"		
F	1,003	Both Correct	Enter an email that does exist in the database	Enter the correct password for that user.	Empty Form	Navigate to user home page		
F	1,004	Wrong email	Enter an email that does not exist in the database	Enter something in the password field and click login button.	Empty Form	Error Message: "Incorrect email"		
F	1,005	Wrong password	Enter an email that exists in the database	Enter a password that does not correspond to the email provided and click the login button	Empty Form	Error Message: "Incorrect password"		
<b>F</b>	<b>= Unit Summary</b>		<b>0% passing</b>		<b>0 passed</b>		<b>Date of last</b>	
	<b>5</b>	<b>tests</b>			<b>5 failed</b>			



WORMS								
Create Account								
To allow a user to create a new account								
Test Cases								
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments
F	2,001	Valid email address	Enter email address which does not have an @ symbol designating an email provider	Enter an invalid email and fill out all other fields and click submit button	Empty form	Error Message: "Invalid email address"		
F	2,002	Valid password	Enter a invalid password	Enter an invalid password and fill out all other fields and click the submit button	Empty form	Error Message: "Invalid password"		
F	2,003	Matching passwords	Enter two passwords which do not match	Enter the two non matching passwords and all other fields and click the submit button	Empty form	Error Message: "Passwords do not match"		
F	2,004	Verification email	Check to see if account received verification email in their email's inbox	Check account's email inbox and click verification link. Then manually check user database for the account to see if it is now verified	Account created, verification email not received	Account in user database is listed as verified		
F	2,005	Empty required fields	Fill out the form leaving a required field empty	Fill out all other fields except one required field and click the submit button	Empty form	Error Message: "Please fill out all required fields"		
F	<b>= Unit Summary</b>		<b>0% passing</b>		<b>0 passed</b>			<b>Date of last</b>
	<b>5</b>	<b>tests</b>			<b>5 failed</b>			



**WORMS**

**Submit**

To allow a user to submit their scholarly work

Test Cases									
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By
F	3,001	Valid Format	Enter a file that does not end in .pdf indicating a pdf file format	Enter an invalid file format and fill out all other fields and click submit button	Empty form	Error Message: "Invalid format, PDF required"			
F	3,002	Empty required fields	Fill out the form leaving a required field empty	Fill out all other fields except one required field and click the submit button	Empty form	Error Message: "Please fill out all required fields"			
F	3,003	Successful storage into the database	Submit a scholarly work in a valid .pdf format with all required fields completed	With a valid .pdf format and all fields required, click the submit button and manually view the database to ensure a successful submission	Valid database, submittable .pdf, and all required fields completed	Error Message: "Failure: Did not successfully insert into the database"			
F	3,004	Title selected	Submit a scholarly work in a valid .pdf format without any text in the title field	Select a valid scholarly work in the correct format without the title field filled out and click the submit button	Valid scholarly work format, empty title field	Error Message: "Please designate a title for your submission"			
F	3,005								
F	<b>= Unit Summary</b>		<b>0% passing</b>		<b>0 passed</b>		<b>Date of last test =</b>		
	<b>4</b>	<b>tests</b>			<b>4 failed</b>				



**WORMS**

**Review**

To allow a user to view and review submitted works and submit their evaluation

Test Cases									
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By
F	4,001	Empty required fields	Fill out the form leaving a required field empty	Fill out all other fields except one required field and click the submit button	Empty form	Error Message: "Please fill out all required fields"			
F	4,002	Successful storage of review into the database	Completely fill out the review form	With the form completely filled out, hit the submit button and manually check the database to ensure that the review was successfully submitted	Valid scholarly work in the database waiting to be reviewed	Error Message: "Failure: Did not successfully insert into the database"			
F	4,003	Ensure reviewer is given an even number of randomly issued papers not from their host school	Click the button to issue out all of the papers to be reviewed	Click the button to issue out all the papers using our automated algorithm and ensure that no person receives an incorrect number of papers and/or papers from their host school	10 submitted papers, 3 from RPI, Siena, 4 from Union College. There will also be 2 reviewers, 1 from Siena and 1 from RPI.	Siena reviewer grading 3 papers from RPI, 2 papers from Union. RPI reviewer grading 3 papers from Siena, 1 from Union.			
F	4,004								
F	4,005								
<b>F</b>	<b>= Unit Summary</b>		<b>0% passing</b>		<b>0 passed</b>		<b>Date of last test =</b>		
	<b>3</b>	<b>tests</b>			<b>3 failed</b>				
	<a href="#">Directory Page</a>								



**WORMS**  
**Conference Configuration**

Configuration of conference options by conference chairs

Test Cases										
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
F	5.001	Add Hotel	Add a hotel on the information page	Log in as conference chair. Navigate to information page. Add a hotel	Empty form	Users may view the newly added hotel on the information page				
F	5.002	Add Meal	Configure a new meal choice	Log in as conference chair. Add a new meal choice under the configuration page	Empty form	Users may select the newly added meal choice upon registration				
F	5.003	Edit general information	Edit general information	Log in as a conference chair. Navigate to the general information page. Edit the general information section	Empty form	Users who navigate to the site can view the newly edited information.				
F	<b>= Unit Summary</b>			<b>0% passing</b>		<b>0 passed</b>		<b>Date of last test =</b>		<b>1/0/00</b>
	<b>3</b>	<b>tests</b>				<b>3 failed</b>				
	<a href="#">Directory Page</a>									





**WORMS**  
**Programming Contest Team Creation**

Creation of programming contest teams by a team advisor

Pass/Fail Status	Test Cases		Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
	Test Number	Description								
F	6.001	View Team Members	Create a contest team, view team page	Create a contest team by filling out the form with valid data. Go to the programming contest page.	Empty form	Page displays team member emails and indicates which ones are registered in the system				
F	6.002	Creation of programming contest team.	Program contest advisor fills out required forms for email addresses of contest team participants.	Fill out all forms required for each participant on the programming contest team and hit the submit button.	Empty forms	Successful creation of programming contest team with a list of participants.				
F	6.003	Valid email addresses of participants.	Register accounts with the email addresses provided by an advisor	Register accounts with email addresses provided by an advisor. Log in with those accounts and confirm they are added to the team.	Programming contest team created	Accounts that log in with emails provided by the advisor can view team information				
F	6.004	At least one team member	Fill out the team creation form without adding any members	Fill out the team creation form without adding any members. Attempt to submit the form.	Empty form	Error: team must have at least one person				
F	<b>= Unit Summary</b>									<b>Date of last test = 1/0/00</b>
	<b>4</b>	<b>tests</b>		<b>0% passing</b>		<b>0 passed</b>				
						<b>4 failed</b>				
	<a href="#">Directory Page</a>									



**WORMS**

**Conference Event Scheduling**

To allow an event to be scheduled by the conference chair

Test Cases										
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
F	7.001	Scheduling an event	Schedule an event in the calendar	Choose a date, submit the information to be displayed in the calendar, and hit submit	Empty calander, with no other dates	The event should be displaying on the website in the calendar under the date submitted				
F	7.002	Deleting an event	Delete an event from the calendar	Choose a random date in the month where an event is scheduled, choose the event, and delete it on the web form	Five random events scheduled on different days in the calendar	The event that was chosen to be deleted should be removed from the calendar without affecting any of the other events				
F	7.003	Scheduling the same event	Attempt to schedule an event with a duplicate name	Submit the event in the schedule twice and see if the event appears twice on the calendar	Empty calander, with no other dates	Error: "Events must have a unique name."				
F	7.004	Submitting an empty event	Submit empty data into the event creation form	Put nothing into the form for calendar event and hit submit	Empty calander, with no other dates	Error: "You must fill in all required fields"				
F	<b>= Unit Summary</b>			<b>0% passing</b>		<b>0 passed</b>			<b>Date of last test =</b>	<b>1/0/00</b>
	<b>4</b>	<b>tests</b>				<b>4 failed</b>				

[Directory Page](#)



WORMS										
System Administration										
Initial configuration of the system										
Test Cases										
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
F	8.001	Add Conference Chair	Add a conference chair on the admin page	Add a conference chair email, register with that email and ensure that appropriate permissions are given	No conference chair with the test email exists in the system	Registering with the test email will allow the user to access conference configuration options				
F	8.002	Add Submission Chair	Add a submission chair on the admin page	Add a submission chair email, register with that email and ensure that the user is able to view submissions	No submission chair with the test email exists in the system	Registering with the test email will allow the user to view submissions				
F	8.003	Configure Topic List	Add a topic to the topic list	Add a topic to the topic list on the admin page. Log in the system and attempt to select that newly added topic for a submission	Topic list does not contain the new test topic	Users may select the newly added topic when making a submission				
F	8.004	View submissions	Navigate to the admin's submissions page	Navigate to the admin's submissions page, ensure that all submissions are viewable	Submissions have been made in the system.	System Admin can view all submissions on the admin page				
F	8.004	View reviews	Navigate to the admin's reviews page	Navigate to the admin's reviews page, ensure that all submissions are viewable	Reviews have been made in the system.	System Admin can view all reviews on the admin page				
F	8.005	Registration Questions	Configure Registration Questions	Enter questions in to the proper form to be submitted to the system	Empty form	New registration questions are shown to a user in the registration form.				
F	= Unit Summary		0% passing		0 passed		Date of last test = 1/0/00			
	6 tests				6 failed					
<a href="#">Directory Page</a>										



**WORMS**

**Conference Registration**

To allow a user to register to attend a conference

Test Cases										
Pass/Fail Status	Test Number	Description	Action to perform test (input)	Steps to be Executed	State Before Test	Expected result	Observed result	Comments	Tested By	Test Date
F	9.001	Successful registration	Register for the conference	Fill in all the fields and click register	Status Unregistered	Status: Registered for the Conference				
F	9.002	Conflicting Workshop Choices	Attempt to register for two simultaneous workshops	Choose 2 or more different simultaneous workshops	Registration form is filled out except for workshops	Error Message: "Wokshop selections are in conflicting times."				
F	9.003	Empty Required Fields	Partially fill out registration form	Fill out the registration form except for some required field	Empty Form	Errors Message: "Please fill in all required fields."				
F	9.004	Additional Meal Tickets	Request Additional Meal Tickets	Navigate to the conference registration page. Request additional meal tickets	Registered successfully for conference	Additional meal tickets show up on the registration page				
F	<b>= Unit Summary</b>			<b>0% passing</b>		<b>0 passed</b>		<b>Date of last test =</b>		<b>1/0/00</b>
	<b>4</b>	<b>tests</b>				<b>4 failed</b>				

[Directory Page](#)



## 18. Appendix E: Database Schema

```

-----
-- Table `perm_trace`.`user`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`user` (
  `id` INT NOT NULL AUTO_INCREMENT ,
  `first_name` VARCHAR(100) NOT NULL ,
  `last_name` VARCHAR(100) NOT NULL ,
  `email` VARCHAR(100) NOT NULL ,
  `password_hash` BINARY(32) NOT NULL ,
  `personal_title` VARCHAR(10) NULL ,
  `phone_number` VARCHAR(20) NULL ,
  `department` VARCHAR(100) NULL ,
  `position` VARCHAR(100) NULL ,
  `city` VARCHAR(100) NOT NULL ,
  `country` VARCHAR(100) NOT NULL ,
  `zip_code` CHAR(5) NULL ,
  `state_territory_province` VARCHAR(100) NULL ,
  `street_address` VARCHAR(400) NULL ,
  `is_conference_chair` TINYINT(1) NOT NULL ,
  `is_registered` TINYINT(1) NOT NULL ,
  `is_accepted_reviewer` TINYINT(1) NOT NULL ,
  `reviewer_credentials` TEXT NULL ,
  `programming_contest_team_name` VARCHAR(30) NULL ,
  `school_name` VARCHAR(80) NOT NULL ,
  PRIMARY KEY (`id`, `school_name`),
  INDEX `fk_user_programming_contest_team1_idx`
  (`programming_contest_team_name` ASC) ,
  UNIQUE INDEX `email_UNIQUE` (`email` ASC) ,
  INDEX `fk_user_school1_idx` (`school_name` ASC) ,
  CONSTRAINT `fk_user_programming_contest_team1`
    FOREIGN KEY (`programming_contest_team_name` )
    REFERENCES `perm_trace`.`programming_contest_team` (`name` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_school1`
    FOREIGN KEY (`school_name` )
    REFERENCES `perm_trace`.`school` (`name` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```



```

-----
-- Table `perm_trace`.`scholarly_work_type`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`scholarly_work_type` (
  `name` VARCHAR(50) NOT NULL ,
  `chair_id` INT NOT NULL ,
  `submission_deadline` DATE NULL ,
  PRIMARY KEY (`name`) ,
  INDEX `fk_scholarly_work_type_user1_idx` (`chair_id` ASC) ,
  CONSTRAINT `fk_scholarly_work_type_user1`
    FOREIGN KEY (`chair_id`)
      REFERENCES `perm_trace`.`user` (`id` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `perm_trace`.`scholarly_work`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`scholarly_work` (
  `id` INT NOT NULL ,
  `submiter_id` INT NOT NULL ,
  `type_name` VARCHAR(50) NOT NULL ,
  `submission_date` DATE NOT NULL ,
  `file_name` CHAR(64) NOT NULL ,
  `accepted` TINYINT(1) NOT NULL ,
  PRIMARY KEY (`id`) ,
  INDEX `fk_scholarly_works_users_idx` (`submiter_id` ASC) ,
  INDEX `fk_scholarly_work_scholarly_work_type1_idx` (`type_name` ASC)
,
  CONSTRAINT `fk_scholarly_works_users`
    FOREIGN KEY (`submiter_id`)
      REFERENCES `perm_trace`.`user` (`id` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_scholarly_work_scholarly_work_type1`
    FOREIGN KEY (`type_name`)
      REFERENCES `perm_trace`.`scholarly_work_type` (`name` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```



```
-----
-- Table `perm_trace`.`review`
-----
```

```
CREATE TABLE IF NOT EXISTS `perm_trace`.`review` (
  `scholarly_work_id` INT NOT NULL ,
  `reviewer_id` INT NOT NULL ,
  `comments` TEXT NULL ,
  `date` DATE NULL ,
  `grade` INT NULL ,
  PRIMARY KEY (`scholarly_work_id`, `reviewer_id`) ,
  INDEX `fk_scholarly_work_has_user_user1_idx` (`reviewer_id` ASC) ,
  INDEX `fk_scholarly_work_has_user_scholarly_work1_idx`
(`scholarly_work_id` ASC) ,
  CONSTRAINT `fk_scholarly_work_has_user_scholarly_work1`
  FOREIGN KEY (`scholarly_work_id`)
  REFERENCES `perm_trace`.`scholarly_work` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_scholarly_work_has_user_user1`
  FOREIGN KEY (`reviewer_id`)
  REFERENCES `perm_trace`.`user` (`id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-----
-- Table `perm_trace`.`topic`
-----
```

```
CREATE TABLE IF NOT EXISTS `perm_trace`.`topic` (
  `name` VARCHAR(30) NOT NULL ,
  `description` VARCHAR(200) NOT NULL ,
  PRIMARY KEY (`name`) )
ENGINE = InnoDB;
```

```
-----
-- Table `perm_trace`.`meal`
-----
```

```
CREATE TABLE IF NOT EXISTS `perm_trace`.`meal` (
  `name` VARCHAR(30) NOT NULL ,
  `description` VARCHAR(300) NOT NULL ,
  `price` DECIMAL(10,2) NOT NULL ,
  PRIMARY KEY (`name`) )
ENGINE = InnoDB;
```



```

-----
-- Table `perm_trace`.`reviewer_has_interest`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`reviewer_has_interest` (
  `reviewer_id` INT NOT NULL ,
  `topic_name` VARCHAR(30) NOT NULL ,
  PRIMARY KEY (`reviewer_id`, `topic_name`) ,
  INDEX `fk_user_has_topic_topic1_idx` (`topic_name` ASC) ,
  INDEX `fk_user_has_topic_user1_idx` (`reviewer_id` ASC) ,
  CONSTRAINT `fk_user_has_topic_user1`
    FOREIGN KEY (`reviewer_id`)
      REFERENCES `perm_trace`.`user` (`id` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_has_topic_topic1`
    FOREIGN KEY (`topic_name`)
      REFERENCES `perm_trace`.`topic` (`name` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `perm_trace`.`scholarly_work_has_topic`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`scholarly_work_has_topic` (
  `scholarly_work_id` INT NOT NULL ,
  `topic_name` VARCHAR(30) NOT NULL ,
  PRIMARY KEY (`scholarly_work_id`, `topic_name`) ,
  INDEX `fk_scholarly_work_has_topic_topic1_idx` (`topic_name` ASC) ,
  INDEX `fk_scholarly_work_has_topic_scholarly_work1_idx`
  (`scholarly_work_id` ASC) ,
  CONSTRAINT `fk_scholarly_work_has_topic_scholarly_work1`
    FOREIGN KEY (`scholarly_work_id`)
      REFERENCES `perm_trace`.`scholarly_work` (`id` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_scholarly_work_has_topic_topic1`
    FOREIGN KEY (`topic_name`)
      REFERENCES `perm_trace`.`topic` (`name` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```





```

-----
-- Table `perm_trace`.`meal_ticket`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`meal_ticket` (
  `user_id` INT NOT NULL ,
  `meal_name` VARCHAR(30) NOT NULL ,
  PRIMARY KEY (`user_id`, `meal_name`) ,
  INDEX `fk_user_has_meal_meal1_idx` (`meal_name` ASC) ,
  INDEX `fk_user_has_meal_user1_idx` (`user_id` ASC) ,
  CONSTRAINT `fk_user_has_meal_user1`
    FOREIGN KEY (`user_id`)
      REFERENCES `perm_trace`.`user` (`id` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_has_meal_meal1`
    FOREIGN KEY (`meal_name`)
      REFERENCES `perm_trace`.`meal` (`name` )
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `perm_trace`.`event`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`event` (
  `name` VARCHAR(50) NOT NULL ,
  `description` VARCHAR(400) NULL ,
  `start_time` DATETIME NOT NULL ,
  `end_time` DATETIME NOT NULL ,
  `location` VARCHAR(50) NOT NULL ,
  PRIMARY KEY (`name`) ,
  INDEX `time_idx` (`start_time` ASC) )
ENGINE = InnoDB;

```

```

-----
-- Table `perm_trace`.`school`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`school` (
  `name` VARCHAR(80) NOT NULL ,
  PRIMARY KEY (`name`) )
ENGINE = InnoDB;

```



```

-----
-- Table `perm_trace`.`user_attends`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`user_attends` (
  `user_id` INT NOT NULL ,
  `event_name` VARCHAR(50) NOT NULL ,
  PRIMARY KEY (`user_id`, `event_name`) ,
  INDEX `fk_user_has_event_event1_idx` (`event_name` ASC) ,
  INDEX `fk_user_has_event_user1_idx` (`user_id` ASC) ,
  CONSTRAINT `fk_user_has_event_user1`
    FOREIGN KEY (`user_id`)
      REFERENCES `perm_trace`.`user` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_has_event_event1`
    FOREIGN KEY (`event_name`)
      REFERENCES `perm_trace`.`event` (`name`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `perm_trace`.`programming_contest_team`
-----
CREATE TABLE IF NOT EXISTS `perm_trace`.`programming_contest_team` (
  `name` VARCHAR(30) NOT NULL ,
  `school` VARCHAR(100) NOT NULL ,
  `advisor_id` INT NOT NULL ,
  PRIMARY KEY (`name`) ,
  INDEX `fk_programming_contest_team_user1_idx` (`advisor_id` ASC) ,
  CONSTRAINT `fk_programming_contest_team_user1`
    FOREIGN KEY (`advisor_id`)
      REFERENCES `perm_trace`.`user` (`id`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```



## 19. Appendix F: Prototype Screens

The first page a user will see before logging in or creating an account:

# TRACE Technologies Prototype

Username

Password

[New User? Create Account](#)

Menu

---

Home

General Information
Schedule

CCSCNE 2013 is the 18th Annual Conference of the Northeast region of the Consortium for Computing Sciences in Colleges (CCSC). The conference is held in cooperation with the ACM SIGCSE and Upsilon Pi Epsilon Honors Society and is hosted by Siena College in Loudonville, NY.

CCSCNE is the northeastern region of the national Consortium for Computing Sciences in Colleges (CCSC). Started in 1996, CCSCNE is one of the largest regions of CCSC.

CCSCNE brings together faculty, staff, and students from academic institutions throughout the Northeast for exchange of ideas and information concerning undergraduate computing curricula. This conference provides a regional forum for the exchange of information and ideas pertaining to the concerns of computing and computing curricula in a smaller academic environment.

CCSCNE holds a refereed conference every Spring, also called CCSCNE. This conference draws participants from all over the northeast and eastern United States. The proceedings of the conference are published as an issue of the Journal of Computing Sciences in Colleges.

<
Mar
>
2013

	Su	Mo	Tu	We	Th	Fr	Sa
8	24	25	26	27	28	1	2
9	3	4	5	6	7	8	9
10	10	11	12	13	14	15	16
11	17	18	19	20	21	22	23
12	24	25	26	27	28	29	30
13	31	1	2	3	4	5	6

[TRACE Technologies](#)



The page a user will see while creating an account:

## TRACE Technologies Prototype

### Register Account

University / Company:	<input type="text"/>
Department:	<input type="text"/>
Title:	<input style="width: 90%;" type="text" value="Mrs."/>
Full Academic / Job Title:	<input type="text"/>
First Name:	<input type="text"/>
Last Name:	<input type="text"/>
Address Line 1:	<input type="text"/>
Address Line 2:	<input type="text"/>
ZIP Code:	<input type="text"/>
City:	<input type="text"/>
State:	<input type="text" value="Select State"/>
Country:	<input type="text" value="Select Country"/>
Phone:	<input type="text"/>
Email:	<input type="text"/>
Homepage:	<input type="text"/>
Username:	<input type="text"/>
Password:	<input type="text"/>
Re-type Password:	<input type="text"/>



The home page most normal users will see after logging into the system:

## TRACE Technologies Prototype

Welcome, **testuser**

**Menu**

- Home
- Submit
- Review
- Conference Registration

ProfileGeneral InformationSchedule

You **are** registered for the conference

Meal selection: **Chicken**

Workshop selection: **How to teach Software Engineering**

You **have not** submitted any works

< Mar 2013 >

	Su	Mo	Tu	We	Th	Fr	Sa
8	24	25	26	27	28	1	2
9	3	4	5	6	7	8	9
10	10	11	12	13	14	15	16
11	17	18	19	20	21	22	23
12	24	25	26	27	28	29	30
13	31	1	2	3	4	5	6

Clear

TRACE Technologies



The page a submitter will see before attempting to submit a Scholarly Work:

## TRACE Technologies Prototype

Welcome, **testuser**

**Menu**

Home

**Submit**

Review

Conference Registration

Submit a work
View my submissions

Submission Type:

Submission Title:

Regular submission:  
 No file chosen

Anonymous submission:  
 No file chosen

Mar  2013

	Su	Mo	Tu	We	Th	Fr	Sa
8	24	25	26	27	28	1	2
9	3	4	5	6	7	8	9
10	10	11	12	13	14	15	16
11	17	18	19	20	21	22	23
12	24	25	26	27	28	29	30
13	31	1	2	3	4	5	6

[TRACE Technologies](#)



The page a Submitter may go to in order to see the status of different submissions:

## TRACE Technologies Prototype

Welcome, **testuser**

**Menu**

Home

**Submit**

Review

Conference Registration

Submit a work

View my submissions

Title	Status
Object Oriented Programming	Reviewed
CS Education	Accepted
Parallel Computing	Rejected
Software Engineering	Under Review

<
Mar
>
2013

	Su	Mo	Tu	We	Th	Fr	Sa
8	24	25	26	27	28	1	2
9	3	4	5	6	7	8	9
10	10	11	12	13	14	15	16
11	17	18	19	20	21	22	23
12	24	25	26	27	28	29	30
13	31	1	2	3	4	5	6

[TRACE Technologies](#)



The page a reviewer will visit to view works assigned for review and download them:

## TRACE Technologies Prototype

Welcome, **testuser**

**Menu**

- Home
- Submit
- Review**
- Conference Registration

**Assigned Works**

Title		
Introductory Curriculum	<input type="button" value="Download"/>	<input type="button" value="Submit Review"/>
Object Oriented Programming	<input type="button" value="Download"/>	<input type="button" value="Submit Review"/>
Graph Theory	<input type="button" value="Download"/>	<input type="button" value="Submit Review"/>
Software Engineering	<input type="button" value="Download"/>	<input type="button" value="Submit Review"/>

< Mar 2013 >

	Su	Mo	Tu	We	Th	Fr	Sa
8	24	25	26	27	28	1	2
9	3	4	5	6	7	8	9
10	10	11	12	13	14	15	16
11	17	18	19	20	21	22	23
12	24	25	26	27	28	29	30
13	31	1	2	3	4	5	6

TRACE Technologies





The page where a Submission Chair may view submissions, view reviews, and accept or reject submissions:

## TRACE Technologies Prototype

Welcome, **testuser**

**Menu**

Home

Submit

Review

**Submission Chair**

Conference Registration

**Submissions**
Reviewer Applications

Sort By Date Submitted Ascending  Descending

	Title	Author	Date	Grade
<input checked="" type="radio"/>	Software Engineering	Lederman	02/01/2013	1
<input type="radio"/>	CS Education	Joe	02/11/2013	0.5
<input type="radio"/>	Ticket to Ride	Lim	02/12/2013	1.2

Title: Software Engineering  
 Author: Lederman  
 Topics: Software Engineering, Education

**Submission PDF**

Reviewers: **John Doe**

Accept
Reject

TRACE Technologies

< Mar 2013 >

	Su	Mo	Tu	We	Th	Fr	Sa
8	24	25	26	27	28	1	2
9	3	4	5	6	7	8	9
10	10	11	12	13	14	15	16
11	17	18	19	20	21	22	23
12	24	25	26	27	28	29	30
13	31	1	2	3	4	5	6

Clear



The page where a Submission Chair may accept Reviewer applications:

## TRACE Technologies Prototype

Welcome, **testuser**

**Menu**

- Home
- Submit
- Review
- Submission Chair**
- Conference Registration

**Submissions** | **Reviewer Applications**

Name	
Jim Matthews	<input type="button" value="View Credentials"/> <input type="button" value="Accept"/>
Darren Lim	<input type="button" value="View Credentials"/> <input type="button" value="Accept"/>
Tim Lederman	<input type="button" value="View Credentials"/> <input type="button" value="Accept"/>

< Mar 2013 >

	Su	Mo	Tu	We	Th	Fr	Sa
8	24	25	26	27	28	1	2
9	3	4	5	6	7	8	9
10	10	11	12	13	14	15	16
11	17	18	19	20	21	22	23
12	24	25	26	27	28	29	30
13	31	1	2	3	4	5	6

TRACE Technologies