

# P.R.I.S.M.

## PROJECT RECORDING INFORMATION SYSTEM MANAGEMENT

### REQUIREMENTS SPECIFICATION

**Presented By:**



**Shannon Pfohl: Team Leader**

**Paul Cherrier: Web Master**

**Ryan Egan: Information Specialist**

**Jordan Holoboski: Database Administrator**

**Kathleen O'Hara: System Administrator**

**Julian Thomas: Development Director**

## Table of Contents

1. Product Overview and Summary .....	4
2. Development and Production Environments .....	4
2.1 Development Environment.....	4
2.2 Operating Environment .....	5
2.3 Maintenance .....	5
3. User Case Narratives.....	5
3.1 Employee .....	5
3.2 Team Leader.....	5
3.3 Supervisor .....	6
3.4 System Administrator .....	6
3.5 User Permission Chart.....	7
4. UML Use Case Diagram .....	8
4.1 UML Use Case Legend.....	8
4.2 UML Use Case Diagram.....	9
5. Data Flow Diagrams .....	10
5.1 Data Flow Legend.....	10
5.2 Context Diagram .....	11
5.3 Level 0 Diagram.....	12
5.4 Level 1 Diagrams .....	13
5.4.1 Log In.....	13
5.4.2 Average Hours.....	14
5.4.3 Create Evaluation.....	15
5.4.4 Record Hours.....	16
5.4.5 Confirm Team Hours .....	17
5.4.6 Create Team Leader Evaluation .....	18
5.4.7 Record Attendance .....	19
5.4.8 Send Email Notification.....	20
5.4.9 Set Up Projects.....	21

6. Prototypes for Discovery.....	22
6.1 Prototype for Team Evaluation .....	22
6.2 Prototype for Team Leader Evaluation .....	23
7. Functional Requirements Inventory .....	24
7.1 Employee .....	24
7.2 Team Leader.....	24
7.3 Supervisor .....	24
7.4 System Administrator .....	25
8. Non-Functional Requirements.....	25
9. Exception Handling .....	25
10. Implementation Priorities .....	26
11. Foreseeable Modifications and Enhancements.....	26
12. Testing Requirements .....	26
13. Acceptance Criteria.....	26
14. Appendices.....	27
14.1 Appendix A: Cross Reference Index .....	27
14.2 Appendix B: Sources of Information .....	27
14.3 Appendix C: Glossary of Terms .....	28
14.4 Appendix D: Timeline .....	30

## 1. Product Overview and Summary

The client, Dr. Fryling, needs an easy and convenient way to keep track of all hours clocked by Employees and evaluations the Employees complete. PRISM will be a web application capable of recording and saving the hours and evaluations into a database. Each Employee's hours will be accessible by the Employee, the Employee's Supervisor, Team Leader, and the System Administrator. Each Employee's evaluations can be seen by the Employee's Supervisor and the System Administrator. PRISM will also be a way for Team Leaders to record attendance for team members and create evaluations of the team members. The goal of PRISM is to keep hours worked, evaluations, and attendance all in one easy to access place for all employees.

## 2. Development and Production Environments

### 2.1 Development Environment

#### Computer #1 – Mac

- iMac
- Model Identifier: 12,1
- Mac OS X Lion 10.7.5 (11G3b)
- 21.4 inch (1920 x 1080 display)
- AMD Radeon HD 6750M 512 MB graphics
- Intel Core i5 (2.5GHz)
- 4 GB Memory
- 500 GB SATA Disk

#### Computer #2 – PC

- OptiPlex 760
- Windows Vista Enterprise
- Dual Dell Screens
  - Dell 2208WFP (1680 x 1050 display)
  - Dell 1908FP (1280 x 1024 display)
- Intel Core Duo CPU (2.93 GHz)
- 4 GB Memory
- 300 GB SATA Disk

#### Server

- Hostname: oraserv.cs.siena.edu
- CentOS 5.2 (final)
- Kernel: 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

## **2.2 Operating Environment**

Quantum Technologies will be using a web based application located on Siena College's oraserv database server. PRISM will consist of an Oracle database with an Apache Web server.

## **2.3 Maintenance**

A bulk of the maintenance will be done in the security of the Software Engineering lab with the appropriate applications. These applications include, but are not limited to programs such as Adobe Photoshop, Dreamweaver and Notepad++ as well as internet browsers such as Google Chrome, Internet Explorer, Safari and Mozilla Firefox.

## **3. User Case Narratives**

### **3.1 Employee**

The Employee will log into PRISM with a username and password. All users are Employees. The Employee can be a System Administrator, a Team Leader, or a Supervisor. The Employee will log into the system and record the hours worked on a project. Comments on completed work during the hours recorded can be made by the Employee. The Employee can work on multiple projects. The Employee can view the average hours recorded of other teams if the other team's average hours are public. At the end of each phase of a project, the Employee must complete evaluations on every Employee that worked on the project as well as the Employee's Team Leader. The Employee can view the average score the Employee received from evaluations.

### **3.2 Team Leader**

The Team Leader will log into PRISM with a username and password. The Team Leader can lead multiple teams and can be an Employee on other teams. The Team Leader has the same privileges as the Employee. The Team Leader can also be a System Administrator or a Supervisor. The Team Leader has the capability of approving the hours completed by the Employees on the project that the team leader is responsible for. Team Leaders can create evaluations on the members of the Team Leader's team. The Team Leader can only see the hours that the Employees record for the project that the Team Leader is responsible for.

### **3.3 Supervisor**

The Supervisor will log into PRISM with a username and password. There can be multiple levels of Supervisors. The Supervisor can have other Supervisors or Employees below them in the system. The Supervisor has the same capabilities as the Employee. On top of these capabilities, the Supervisor can see the hours that the Supervisor's subordinates record. The Supervisor cannot see team project hours unless the Team Leader has made the project hours public. The Supervisor also has the capability to create an evaluation on the Supervisor's Employees. The Supervisor can see all evaluations made on the Supervisor's Employees. The Supervisor can take on multiple roles such as Team Leader or System Administrator.

### **3.4 System Administrator**

The System Administrator will log into PRISM with a username and password. The System Administrator will have the responsibility of maintaining and configuring PRISM. The System Administrator will have access to adding and manipulating all accounts within PRISM. The System Administrator has the capability to add projects to the systems as well as adding all employees and team leaders to the projects. The System Administrator also has the capability to remove any employees and team leaders from projects. There can be multiple System Administrators for PRISM. System Administrators can also be Supervisors, Team Leaders, and Employees. System Administrators has access to all privileges and information recorded in PRISM. Email notifications about items due, such as recorded hours and evaluations, will be generated by the System Administrator and distributed to the appropriate Employees.

### 3.5 User Permission Chart

#### EVALUATIONS

	Submit Evaluation(s) Of Subordinate(s)	Submit Evaluation(s) For Other Member(s) On the Same Team	View Average Score On Own Evaluation(s)	View Complete Evaluation(s) Of Subordinate(s)
System Administrator	X	X	X	X
Supervisor	X	X	X	X
Team Leader	X	X	X	X
Employee		X	X	

#### HOURS

	Record Hours	View Own Recorded Hours	View Hours Recorded By Subordinate(s)	Confirm Hours Recorded By Subordinate(s)	View Other Team(s) Average Hours
System Administrator	X	X	X	X	X
Supervisor	X	X	X	X	X
Team Leader	X	X	X	X	X
Employee	X	X			X

#### ATTENDANCE

	Record Attendance	View Attendance Of Subordinate(s)
System Administrator	X	X
Supervisor		X
Team Leader	X	X
Employee		

<b>LEGEND</b>	
X	-Enabled Permission

#### MISC.

	Log In	Set Up System	Send Email Notifications To Employees	Add Users To System	Delete Users From System	Manage User Permissions
System Administrator	X	X	X	X	X	X
Supervisor	X		X			
Team Leader	X		X			
Employee	X					

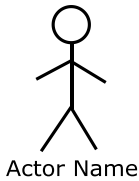
## 4. UML Use Case Diagram

### 4.1 UML Use Case Legend

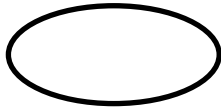
UML Use Case Diagrams are used to give an overview of the functionality of a system. The diagram is based off of the use case narratives. The diagram is a brief snapshot of the use case narratives. The UML Use Case Diagram shows how each user interacts with the system and other users. The diagram also displays what actions each user can perform.



**System Boundary:** This is where all the interaction occurs. Represents what is within the system and outside of it. Uses go on the inside and actors go on the outside.

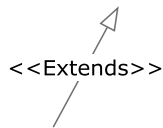


**Actor:** Actors interact with the system through uses. Actors can be human or non-human. Human actors go on the left side of the system boundary. Non-human actors go on the right side. Actor name gets displayed below the actor.

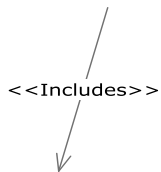


**Scenarios:** The actions that occur within a system and how the user interacts with the system.

**Participation Line:** Shows what scenarios an actor can interact with.



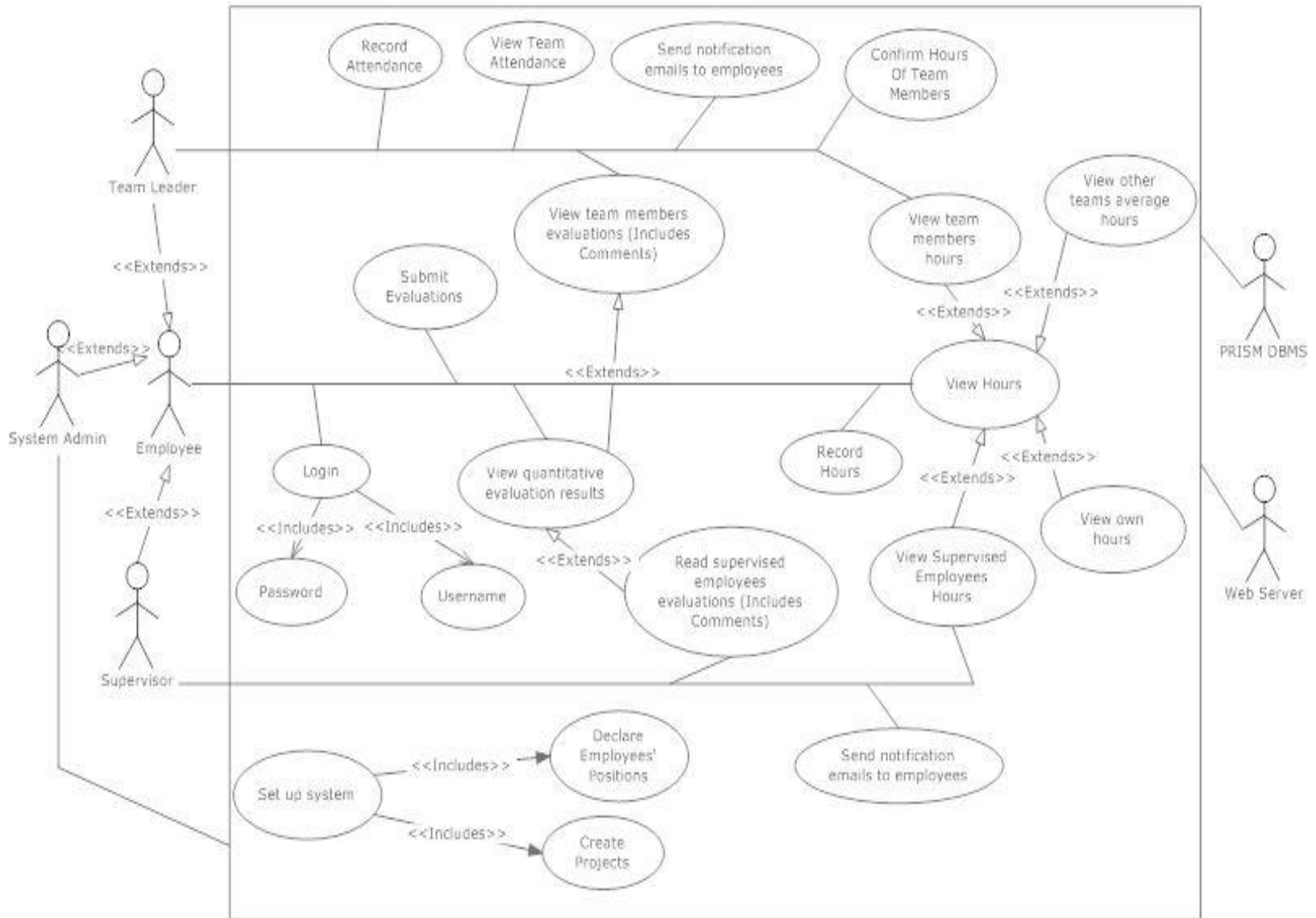
**Inheritance Arrow :** An arrow that points from one use to another. The use being pointed at is the parent and the other is the sub.



**Inclusion Arrow :** An arrow that points from a scenario to another scenario to show that something must be included for the scenario.



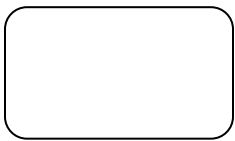
### 4.2 UML Use Case Diagram



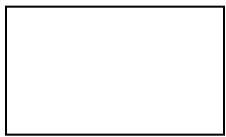
## 5. Data Flow Diagrams

### 5.1 Data Flow Legend

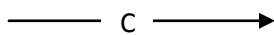
Data Flow Diagrams represent the movement of data between processes in the system as well as the movement of data between processes and external entities outside the system. The diagrams are a tool for analyzing the structure of the system and the ways in which data will be stored and retrieved by different processes. These diagrams model data flows at different levels of detail in the system. The following symbols will be used within the Data Flow Diagram:



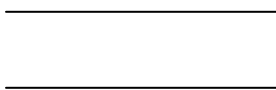
**Process:** Transforms or manipulates data.



**External Entity:** Contributes data or information to the system or which receive data/information from it.



**Data Flow:** Data/information flowing to or from a process where C is the data/information.

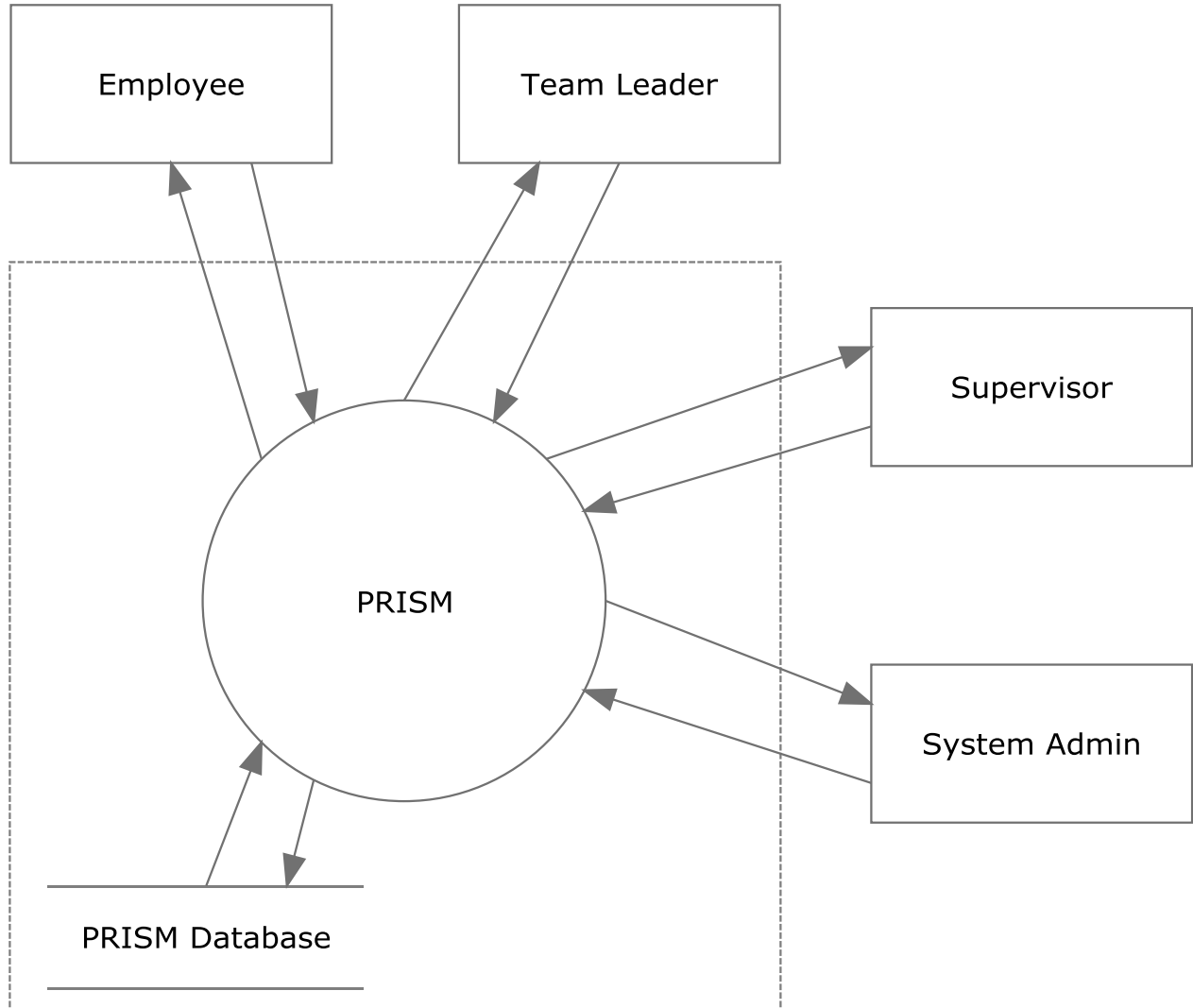


**Data Store:** Location where data is held temporarily or permanently.

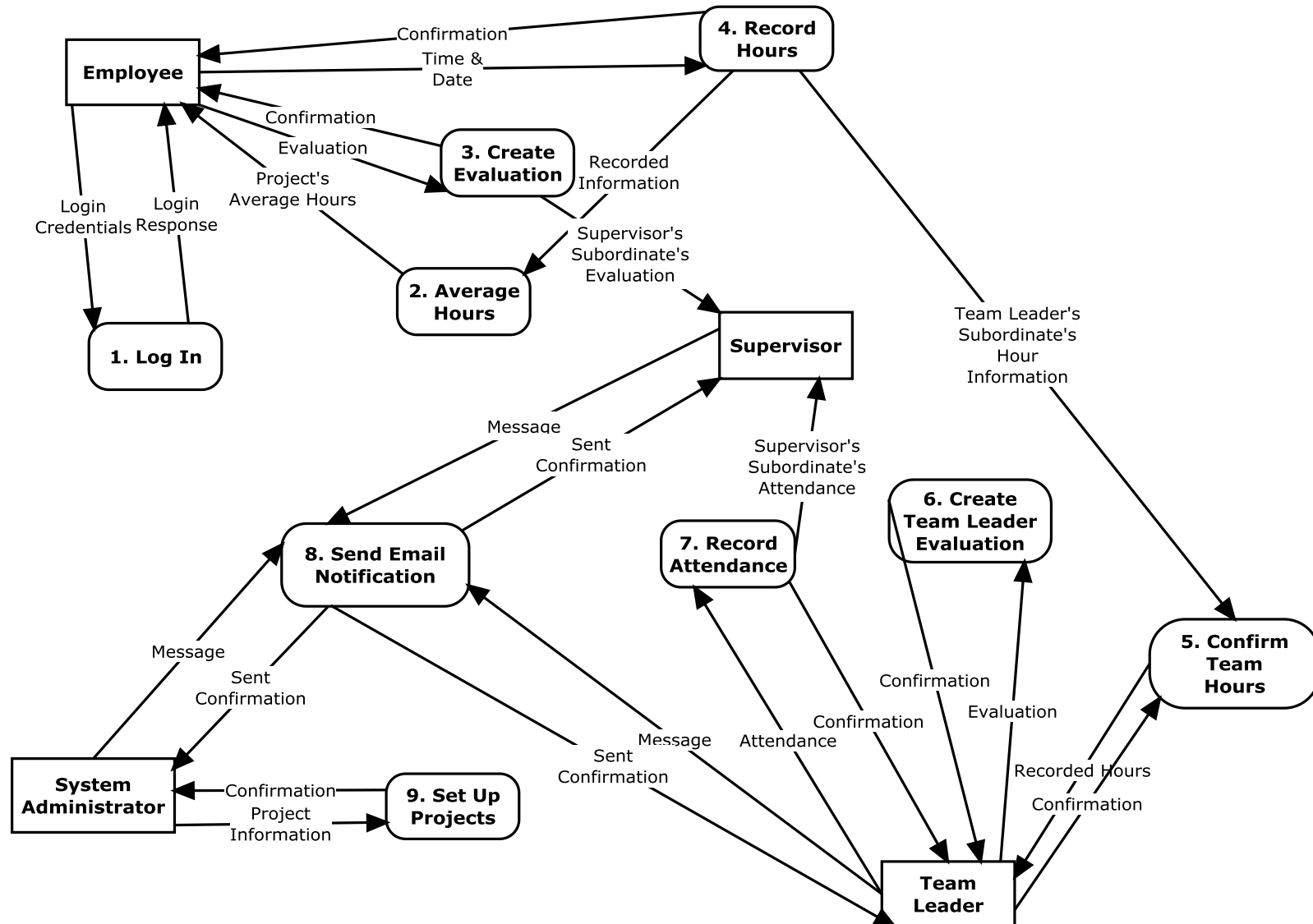


**System Boundary:** This is where all the interaction occurs. Represents what is within the system and outside of it. Uses go on the inside and actors go on the outside.

### 5.2 Context Diagram

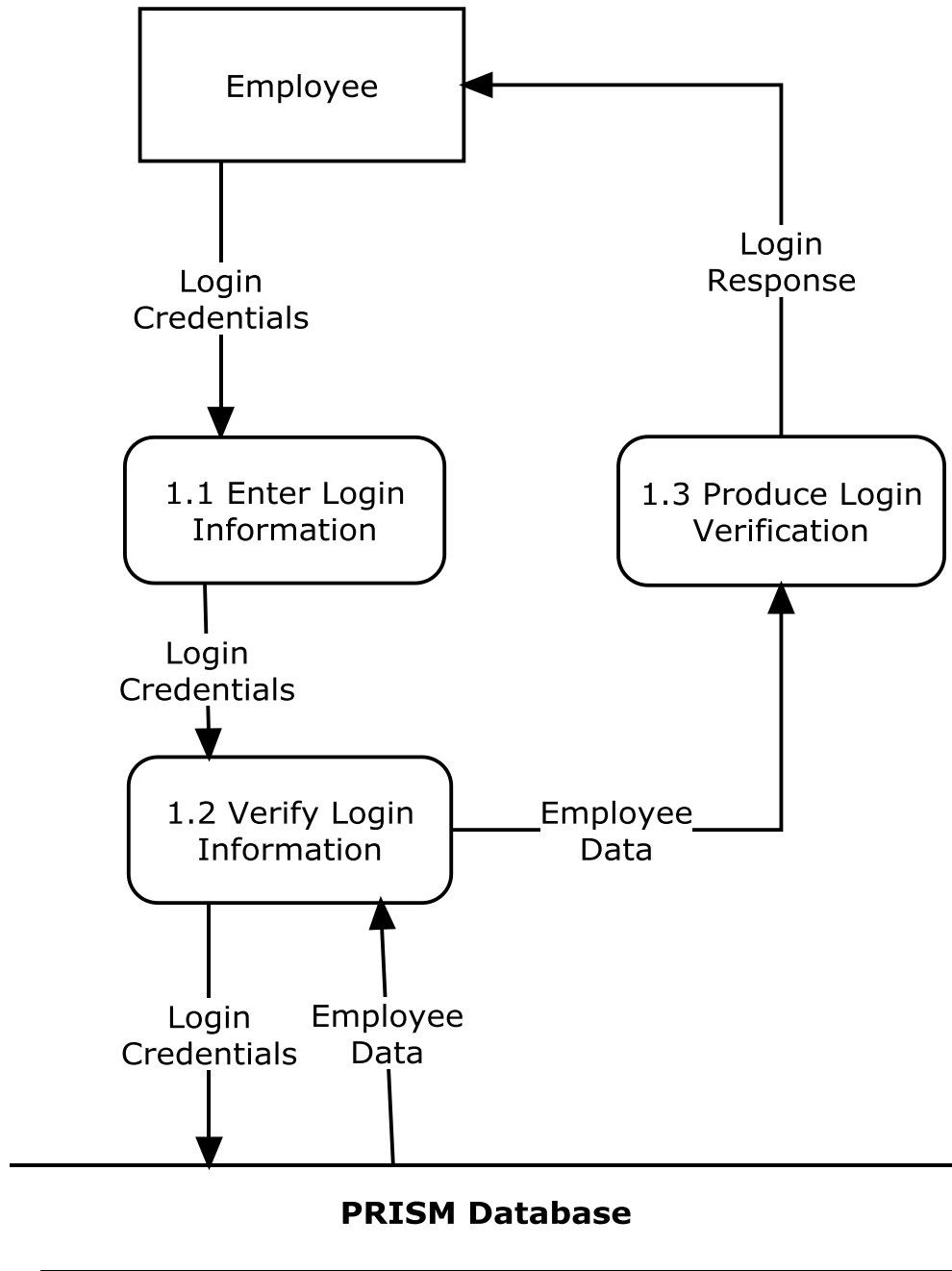


### 5.3 Level 0 Diagram

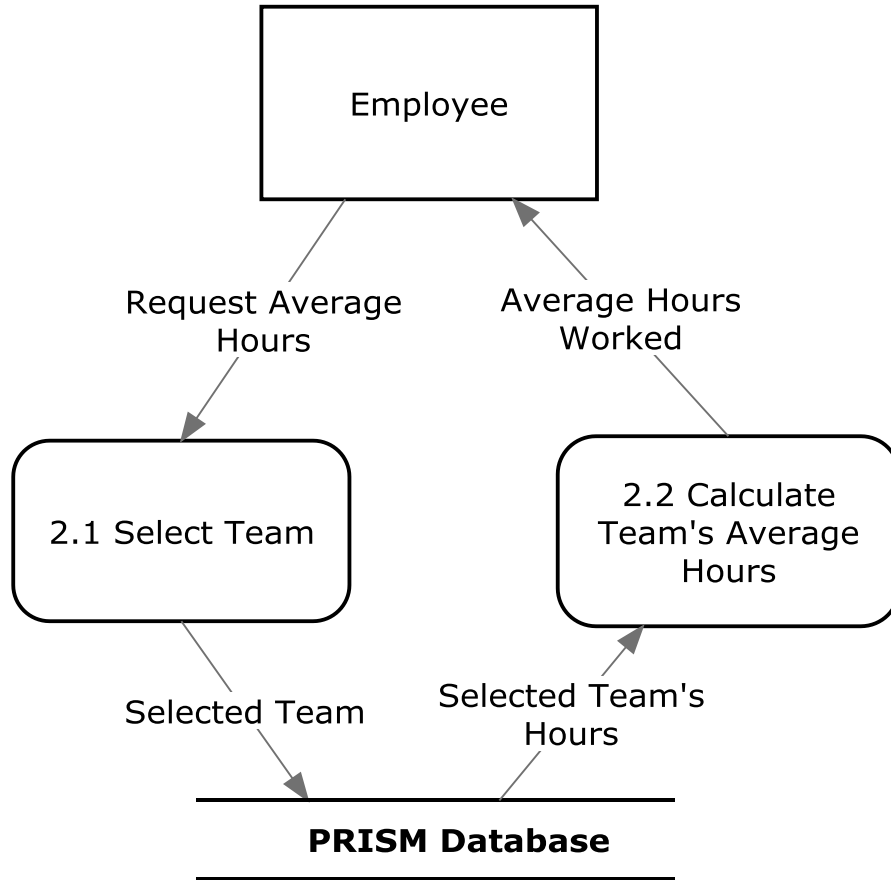


### 5.4 Level 1 Diagrams

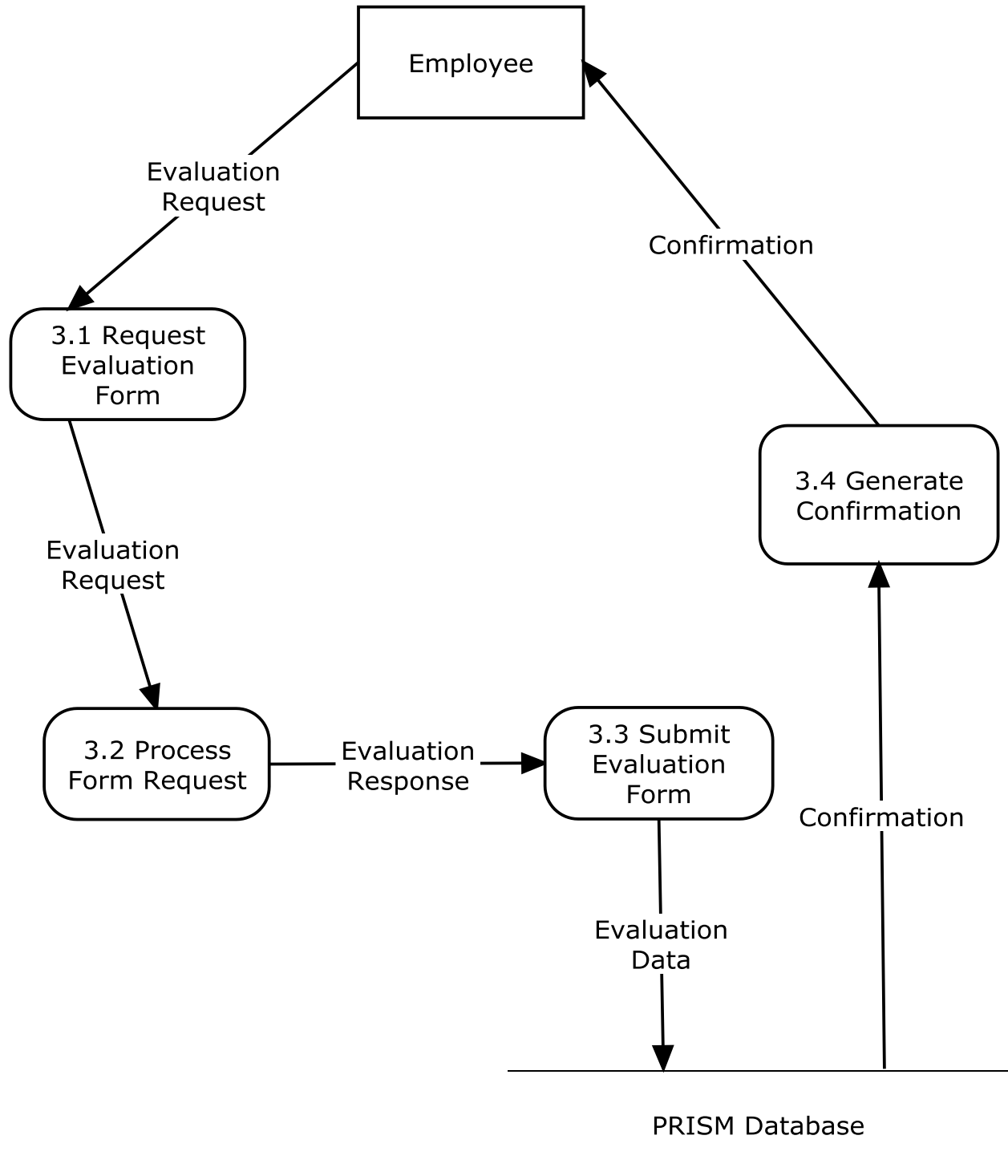
#### 5.4.1 Log In



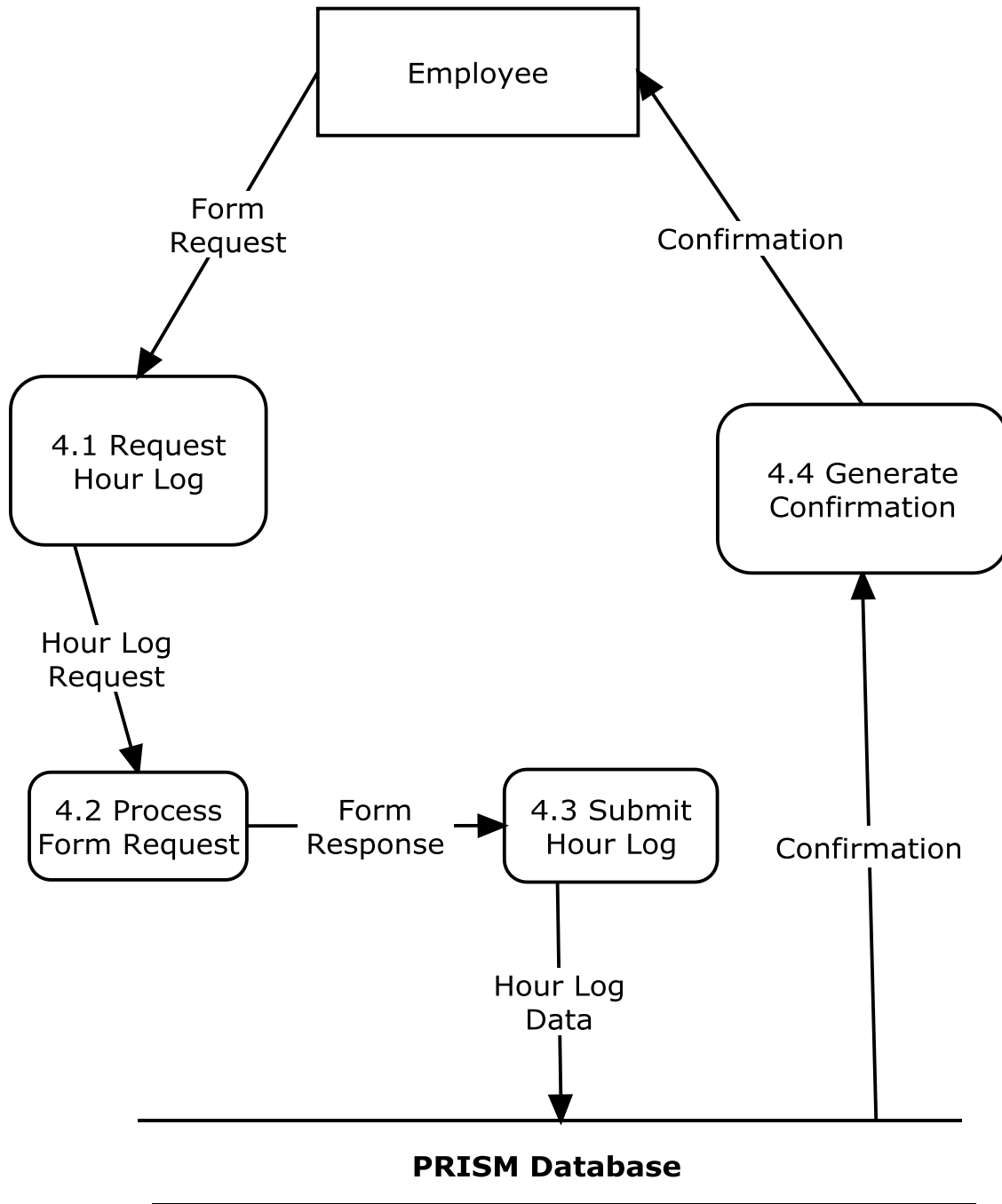
### 5.4.2 Average Hours



### 5.4.3 Create Evaluation

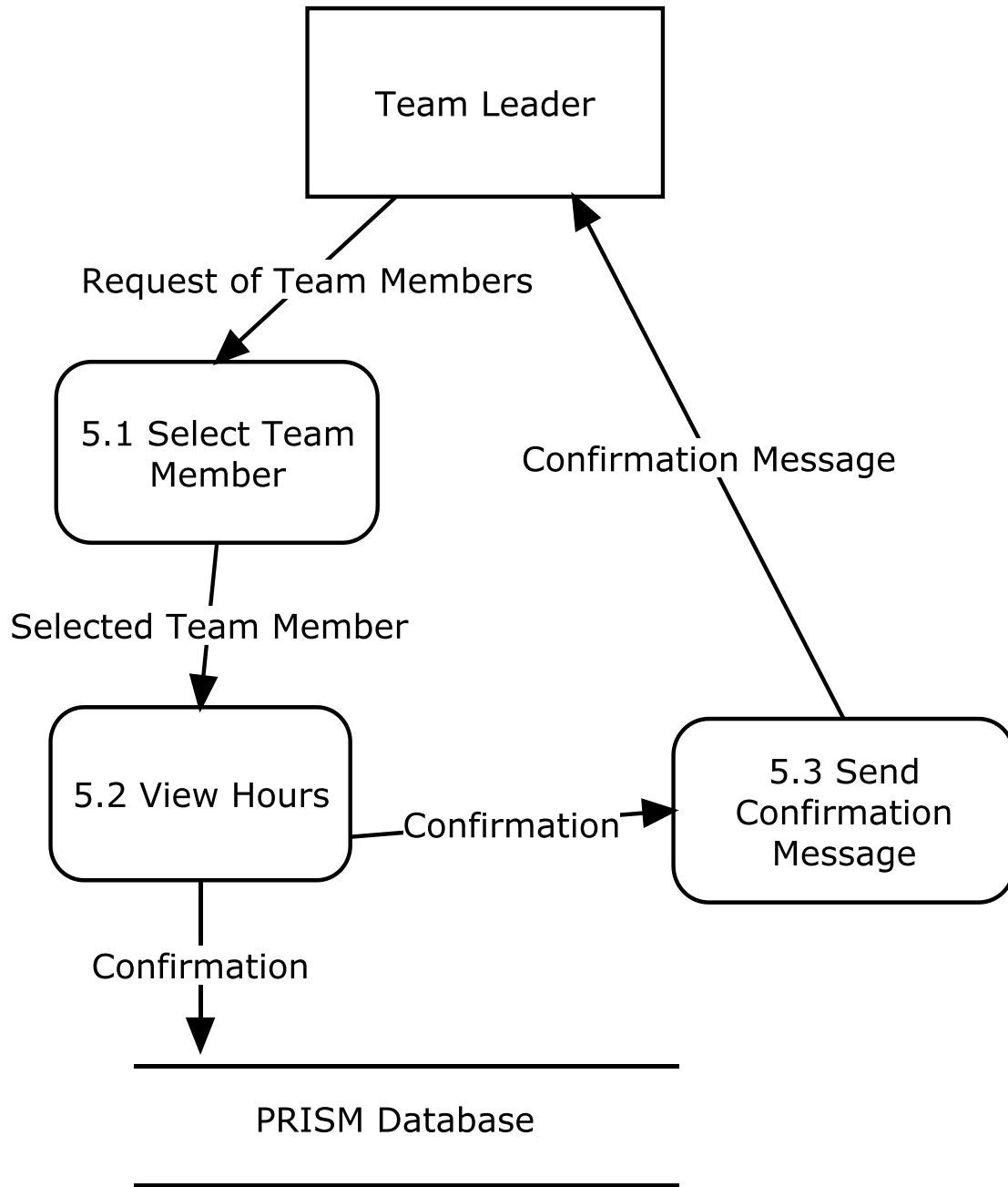


5.4.4 Record Hours

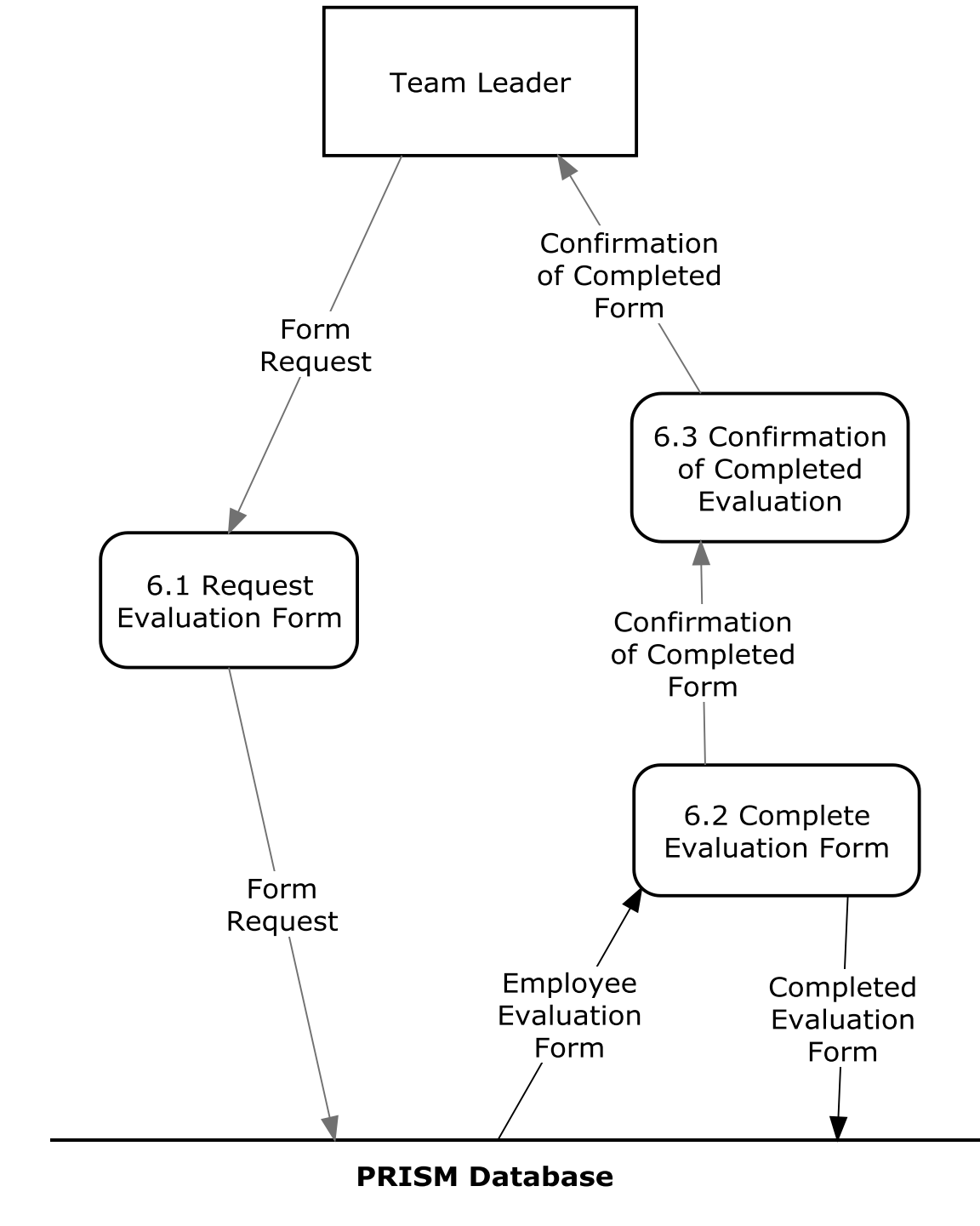




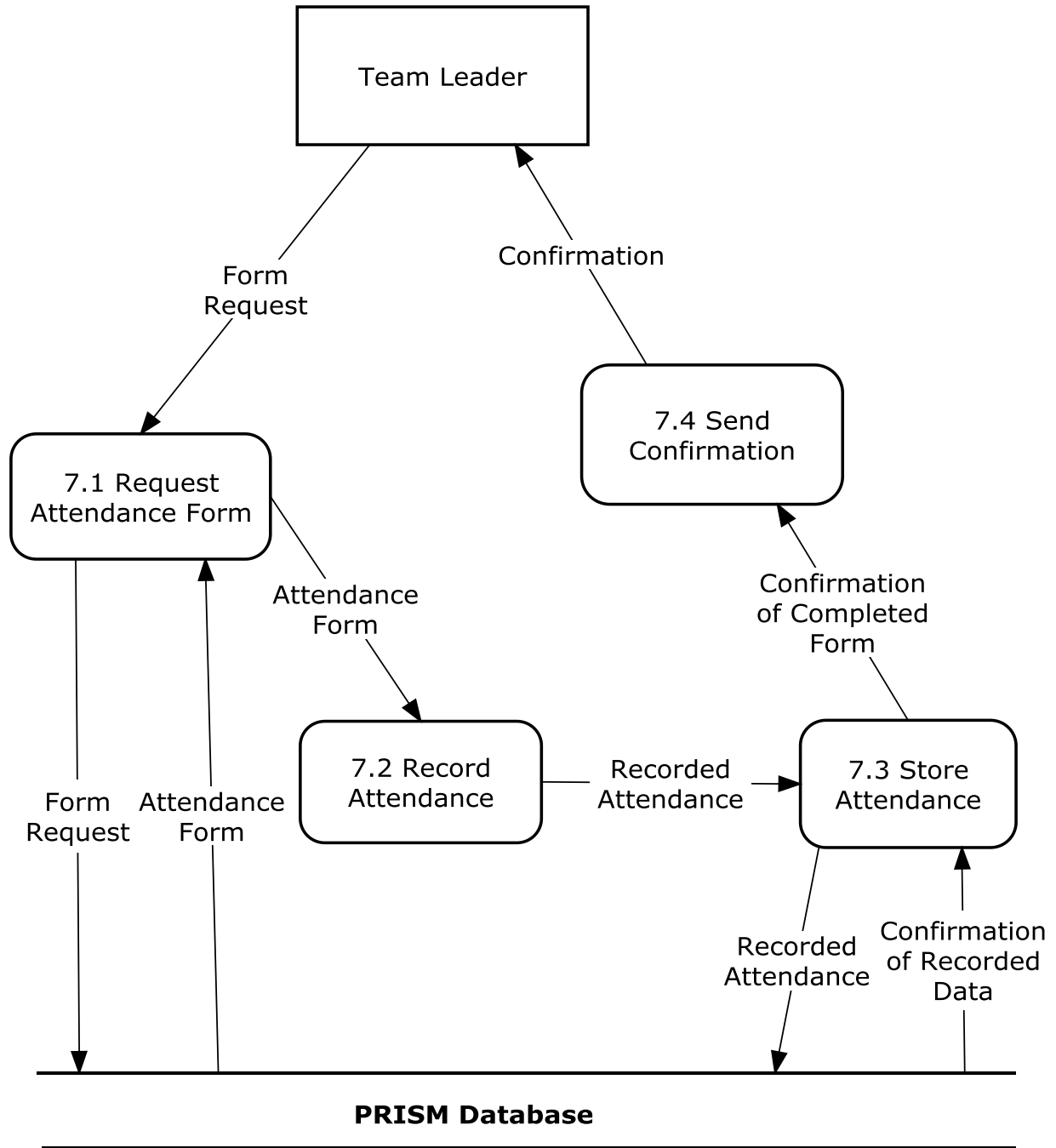
5.4.5 Confirm Team Hours



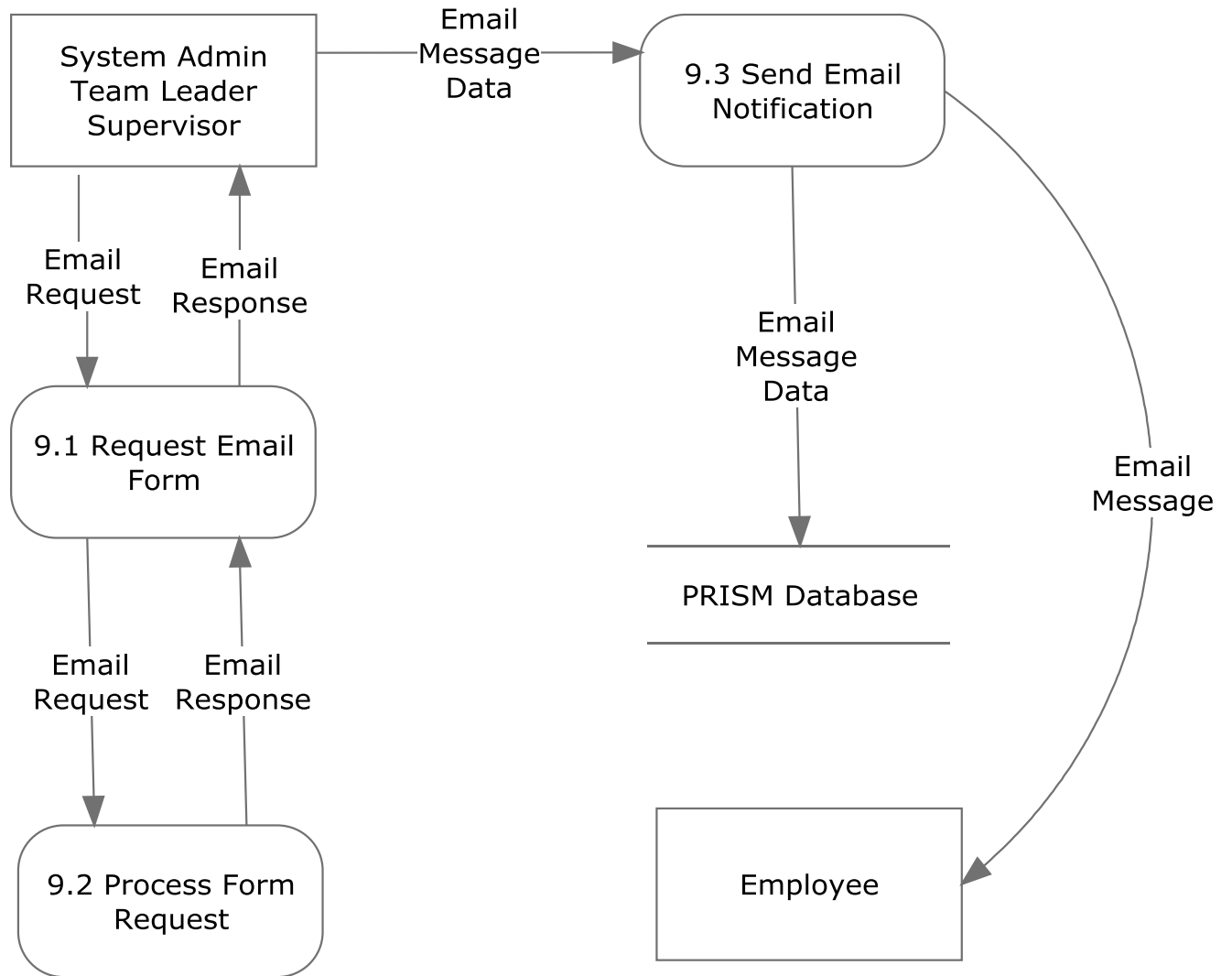
### 5.4.6 Create Team Leader Evaluation



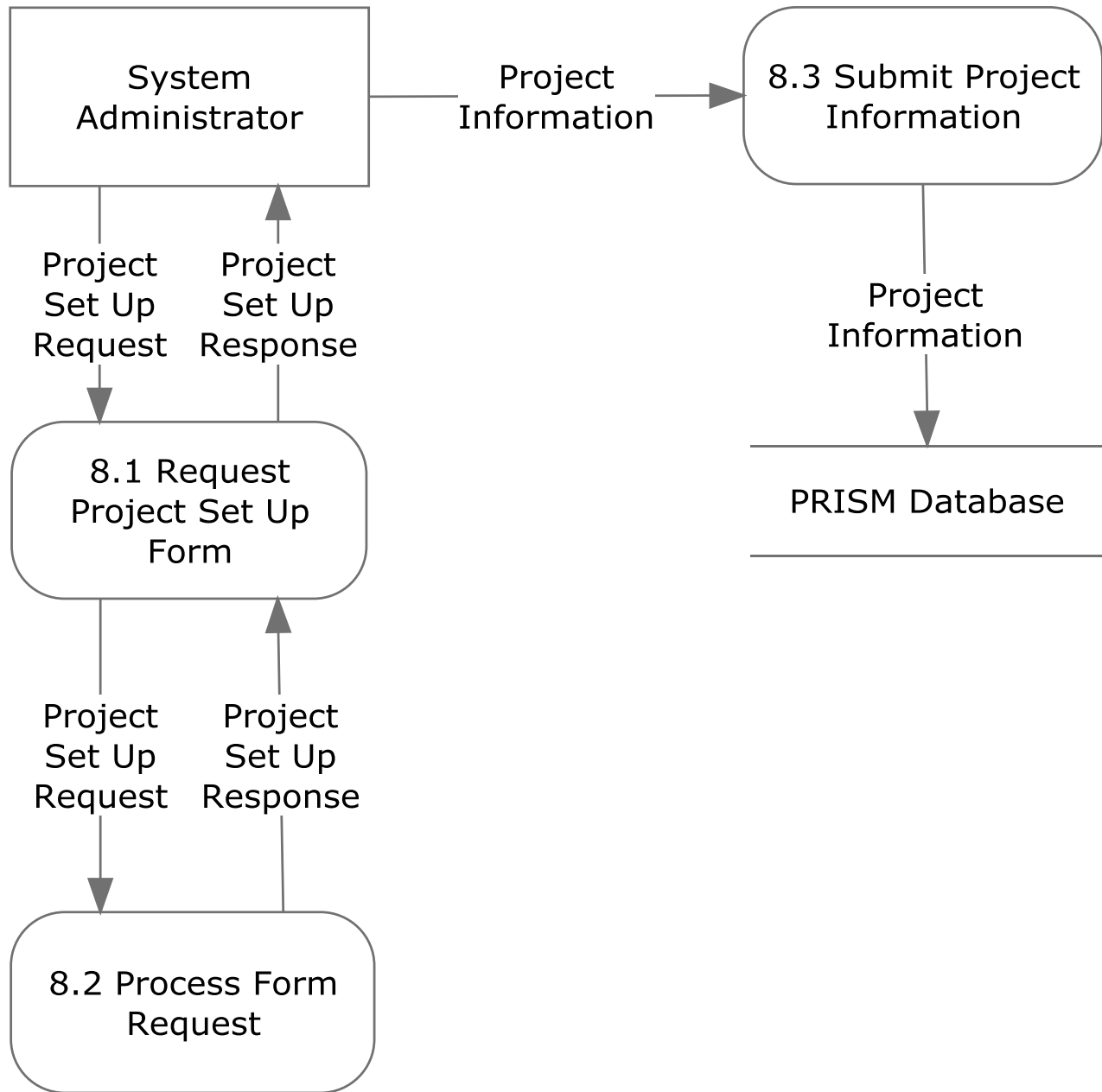
### 5.4.7 Record Attendance



**5.4.8 Send Email Notification**



5.4.9 Set Up Projects



## 6. Prototypes for Discovery

### 6.1 Prototype for Team Evaluation

# Team Member Evaluations

Client(s): Automated

Team Name: Automated

Your Name: Automated

Project Phase: Automated

Name	% of Work (0-100%)	Quality of Work (0-100%)	Comments
<i>Automated</i>	0	0	
<i>Automated</i>	0	0	
<i>Automated</i>	0	0	
<i>Automated</i>	0	0	
<i>Automated</i>	0	0	
<i>Automated</i>	0	0	
	Adds Up to 100% Automatically	Team Average Calculated Automatically	



## 7. Functional Requirements Inventory

The list below outlines the required functionality PRISM will have once the final system is complete. PRISM will be a web application that will be viewable and useable on all major Internet browsers, such as Internet Explorer, Google Chrome, Mozilla Firefox, and Apple Safari.

The following lists detail the requirements of each human user that will interact with PRISM.

### 7.1 Employee

- Will be able to login to the system
  - A login attempt that fails will display an error message
- Will be able to record hours spent on projects
  - Includes hours spent working on projects and hours spent project related meetings
- Will be able to view own recorded hours
- Will be able to view other teams' average hours worked on a project
  - Only true if the other team(s) allow for average hours to be displayed publicly
- Will be able to submit evaluations of other team members
- Will be able to view own average score of evaluations submitted by other team members
- Will be able to log out

### 7.2 Team Leader

- Will inherit all functional requirements specified for the Employee
- Will be able to record attendance of self and team members
- Will be able to view recorded attendance for own team
- Will be able to view hours recorded by team members
  - Will be able to confirm hours if the hours recorded by team members are valid; which will lock the hours from any future changes by the Employee
- Will be able to view evaluations made by other team members that are subordinate to the team leader
  - Will be able to view all scores and comments made on evaluations
- Will be able to send emails to Employees on the team
- Will be able to send periodic emails to Employees on the team

### 7.3 Supervisor

- Will inherit all functional requirements specified for the Employee
- Will be able to view the hours of the Employees they supervise
- Will be able to view attendance of Employees they supervise
- Will be able create an evaluation on the Employees they supervise
- Will be able to view all evaluations made on the Employees, Team Leaders, and Supervisors they supervise
- Will be able to send emails to all Employees that they supervise
- Will be able to send periodic emails to Employees that they supervise



## 7.4 System Administrator

- Will inherit all functional requirements specified for the Employee, Team Leader, and Supervisor
- Will be able to create individual accounts
- Will be able to manipulate any user's account
- Will be able to assign the role of a user
- Will be able to add projects to the system
- Will be able to assign specific Employees to projects
- Will be able to assign Team Leaders to projects
- Will be able to assign Supervisors to oversee Employees
- Will have complete access to all information recorded in the system by all Employees
- Will be able to send emails to all Employees
- Will be able to send periodic emails to all Employees

## 8. Non-Functional Requirements

The non-functional requirements will specify the properties of PRISM. The following requirements are not necessarily specific features that exist within the system, but what the system is intended to do. These features cannot be objectively measured. PRISM's non-functional requirements are as follows:

- PRISM will be efficient
- PRISM will be user friendly
- PRISM will have compatibility on multiple browsers
- PRISM will be easily maintained
- PRISM will be stable
- PRISM will be easy to access

## 9. Exception Handling

PRISM will be able to handle all exceptions that could possibly occur so that the system will run properly even in the event of an error. All input will be validated before the input is used or stored in PRISM. If input is not valid or required information fields are not filled out, PRISM will not accept the input and the user will receive an error message. As development of PRISM begins, Quantum Technologies will handle system and software exceptions when Quantum Technologies decides what system level software will be used.

## 10. Implementation Priorities

Every effort will be made to satisfy each functional requirement, but the most important components are:

- The ability for users to login
- The ability for every employee to record hours worked
- The ability to create evaluations
- The ability for team leaders to take attendance
- The system should be easy to access for all employees
- The System Administrator should have access to everything within PRISM

## 11. Foreseeable Modifications and Enhancements

At this time, Quantum Technologies does not know of any additional modifications or enhancements for PRISM. If any new enhancements or modifications are needed, the enhancements will be added accordingly. If Dr. Fryling believes that a modification or enhancement needs to be made, then Quantum Technologies will make note of the modification. Quantum Technologies will only make the modification if Quantum Technologies is hired to do future work after the acceptance test.

## 12. Testing Requirements

PRISM will be tested on Internet Explorer, Google Chrome, Mozilla Firefox, and Apple Safari. Each functional requirement detailed in the functional requirements inventory will be tested to ensure that each functional requirement is correctly implemented within the system. The tests PRISM will undergo will be specified in more detail as the project reaches the preliminary design phase.

## 13. Acceptance Criteria

Each component of PRISM will have individual tests that will be performed on the system. These tests will be done in order to make sure the individual parts of the system are working properly. Once the individual tests have been completed, an acceptance test will be executed to check if the functional requirements have been met. Once the acceptance test is over, Quantum Technologies and the client, Dr. Fryling will decide whether all of the requirements have been met. Quantum Technologies will design the testing plan using the functional requirements inventory.

## **14. Appendices**

### **14.1 Appendix A: Cross Reference Index**

Project Recording Information System Management Context Diagram

Project Recording Information System Management Level 0 Diagram

Project Recording Information System Management 1 Login Level 1 Diagram

Project Recording Information System Management 2 Average Hours Level 1 Diagram

Project Recording Information System Management 3 Create Evaluation Level 1 Diagram

Project Recording Information System Management 4 Record Hours Level 1 Diagram

Project Recording Information System Management 5 Confirm Team Hours Level 1 Diagram

Project Recording Information System Management 6 Create Team Leader Evaluation Level 1 Diagram

Project Recording Information System Management 7 Record Attendance Level 1 Diagram

Project Recording Information System Management 8 Send Email Notification Level 1 Diagram

Project Recording Information System Management 9 Set Up Projects Level 1 Diagram

### **14.2 Appendix B: Sources of Information**

The primary source of information necessary for PRISM will come from Quantum Technologies' client, Dr. Fryling. The supervisor, Dr. Lederman, will provide extra information and help in class, while the supervisor, Dr. Lim, will give Quantum Technologies instructions through labs. Quantum Technologies will also utilize the information provided through credible sources on the World Wide Web.

### 14.3 Appendix C: Glossary of Terms

**Actor:** Actors that interact with the system through uses. Actors can be human or non human.

**Adobe Dreamweaver:** Tool used for web application development

**Adobe Photoshop:** Graphic editing application

**Apache HTTP Server:** Apache HyperText Transfer Protocol Server, Web server application

**Apple Safari:** Web browser designed by Apple

**Classic Waterfall Model:** Software development process where each phase flows down into the next. This process makes it difficult to go back up the process

**Data Stores:** A component of a Data Flow Diagram that represents a location in which information or data is stored

**Database:** Organizes data, typically through a computer, so that the data is easily accessible

**Data Flow:** Data/information flowing to or from a process in a Data Flow Diagram

**Data Flow Diagram:** A graphical representation of the "flow" of data through an information system

**Data Store:** Location where data is held temporarily or permanently in a Data Flow Diagram

**External Entities:** A component of a Data Flow Diagram that represents any human or non-human user of a Software System

**Functional Requirements Inventory:** Defines what the system will be able to do and what is testable about the system

**Gantt Chart:** Bar chart typically used to project scheduling

**Google Chrome:** Web browser designed by Google

**Inclusion Arrow:** An arrow that points from a scenario to another scenario to show that something must be included for the scenario

**Inheritance Arrow:** An arrow that points from one use to another; the use of being pointed at is the parent and the other is the sub

**Internet Explorer:** Web browser designed by Microsoft

**Level-0 Diagram:** A data flow diagram that represents a system's major processes, data flows, and data stores at a high level of detail

**Level-1 Diagram:** Provides an overview of the major functional areas of the undertaking

**Mozilla Firefox:** Web browser designed by Mozilla Foundation and the Mozilla Corporation

**Non-Functional Requirements Inventory:** Requirements that are not necessarily specific features that exist in a system, but what the system is intended to do

**Notepad++:** Text editor specializing in syntactic highlighting of various programming languages

**Oracle Database:** An object-relational database management system produced and marketed by Oracle Corporation

**Oraserv Database:** Siena College's database server

**Participation Line:** Shows what scenarios an actor can interact with in a UML Use Case Diagram

**PRISM:** Project Recording Information System Management

**Process:** Transforms or manipulates data in a Data Flow Diagram

**Prototype:** An early sample, model or release of a product built to test a concept

**Requirements Specification:** Further defining the client's problem to meet the specifications and requirements

**Scenarios:** The actions that occur within a system and how the user interacts with the system

**SQL:** Structured Query Language, language used to query databases

**System Boundary:** The boundary between the system and the external entities in a Data Flow Diagram

**UML Use Case Diagram:** A type of behavioral diagram to present a graphical overview of the functionality provided by a system

**UML (Unified Modeling Language):** A specification language used in software engineering

**UPC (User Permission Chart):** Chart that demonstrates the permissions of the different users in PRISM

### 14.4 Appendix D: Timeline

