|  |  |
| --- | --- |
|  | Prepared by:  Emma Bostian  Kevin Conner  Justin Rose  David Hall  Sara Pinti |

Detailed design: Iteration 2

**Client: Dr. Erik Eddy**

**Teammate Evaluation Software**

Dr. Meg Fryling

Dr. Darren Lim

Siena College

Department of Computer Science

March 9, 2015

Table of contents

|  |  |
| --- | --- |
| 1. External Design Specifications |  |
| 1.1 Product Overview |  |
| 1.2. Development and Production  Environment |  |
| 1.2.1 Development Environment |  |
| 1.2.2 Operating Environment |  |
| 1.3 Detailed Data Flow Diagrams | D |
| 1.3.1 Data Flow Diagram Legend | D |
| 1.3.2 Context Diagram | D |
| 1.3.3 Level 0 Diagram | D |
| 1.3.4 Level 1 Diagrams | D |
| 1.3.4.1 Create Account | D |
| 1.3.4.2 Login | D |
| 1.3.4.3 Evaluate Teammate | D |
| 1.3.4.4 View Individual Report | D |
| 1.3.4.5 View Team Report | D |
| 1.3.4.6 Establish Class Roster/Team  Profile | D |
| 1.3.4.7 Manage Team Evaluations | D |
| 1.3.4.8 Manage User Permissions | D |
| 1.4 UML Activity Diagram |  |
| 1.4.1 Activity Diagram Legend |  |
| 1.4.2 Login Activity Diagram |  |
| 1.5 UML Deployment Diagram |  |
| 1.5.1 Deployment Diagram Legend |  |
| 1.5.2 Deployment Diagram |  |
| 1.6 Logical data stores\*x |  |
| 1.7 Logical data dictionary |  |
| 2. Logical Data Model (E/R Diagram) |  |
| 2.1 E/R Diagram Legend |  |
| 2.2 E/R Diagram |  |
| 3. Phsyical Data Model (Relational Schema) |  |
| 3.1 Relational Schema Legend |  |
| 3.2 Relational Schema |  |
| 4. Architectural Design Specification |  |
| 4.1 Website Map Legend | D |
| 4.1.1 Website Map | D |
| 4.2 Parameter specifications\*x |  |
| 4.3 Logical data structures\*x?? |  |
| 4.4 Functional descriptions\*?? |  |
| 4.5 Packaging Specifications |  |
| 5. Test Plans |  |
| 5.1 Test Directory |  |
| 5.2 Unit Test 1 |  |
| 5.3 Unit Test 2 |  |
| 6. Pseudocode Routines |  |
| 6.1 Module 1 |  |
| 6.2 Module 2 |  |
| 6.3 Web Application |  |
| 7. Prototypes |  |
| 7.1 TEAMS 101 |  |
| 7.2 Team Evaluation Login |  |
| 7.3 Team Evaluation Home |  |
| 8. Appendix/Glossary |  |

External Design Specifications

**1.1 Product Overview**

Teammate Evaluation will be a core web application on the TEAMS 101 (Team Evaluation and Management System) dashboard, and it will give students an easily-accessible interface to provide teammate evaluations periodically during group projects. Dr. Erik Eddy realizes the importance that team evaluations hold within a group dynamic, not only in regards to the final grade for the assignment, but for the overall cohesiveness and harmony of the group. Team Evaluation will not only provide Siena College students with the ability to communicate more productively, it will provide professors and administrators the ability to be better acquainted with the team member’s participation and performance.

**1.2 Development and Production Environment**

**1.2.1 Development Environment:**

Window’s Computer (Software Lab):

Model: Dell OptiPlex 760

Operating System: Windows Vista

Proc: Intel Core 2 Duo E7500 @2.93GHz

RAM: 4GB

HDD: 500GB

Software:

Adobe Dreamweaver, Google Chrome, Internet Explorer, Mozilla Firefox, MySQL, Notepad ++

NOVA Tech will also be using personal laptops throughout the development process.

**1.2.2 Production Environment:**

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)

NOVA Tech will be using a web based application located on a server provided by Dr. Eddy. Team Evaluations will utilize an Oracle database with an Apache Web server.

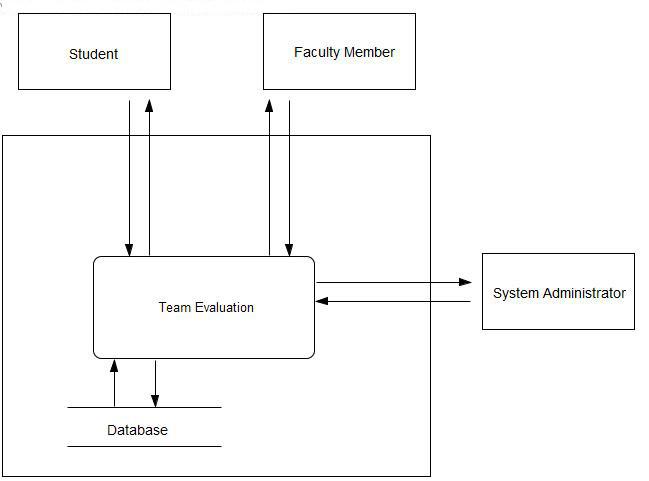
**1.3 Detailed Data Flow Diagrams**

The data flow diagrams will contain the context diagram, a level 0 diagram, and multiple level 1 diagrams. These diagrams visually depict the movement of data between both internal processes and external entities. From these diagrams, the structure of the system can be analyzed as well as the ways in which data moves throughout the system, outside of the system, and is stored and retrieved. The following symbols will be used in the data flow diagrams:

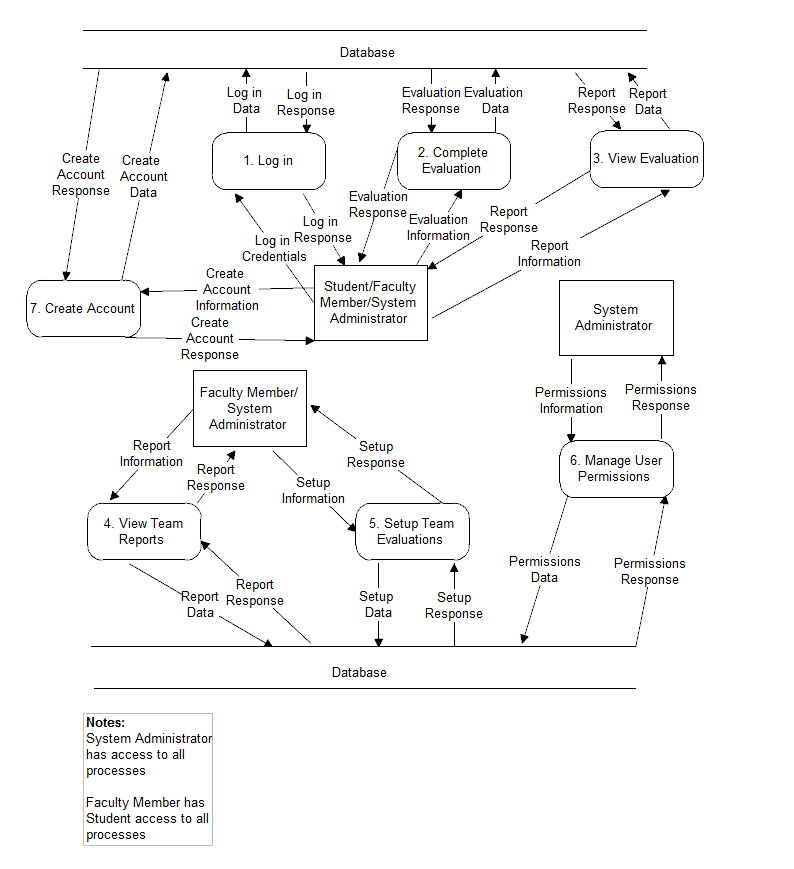
**1.3.1 Data Flow Diagram Legend**

|  |  |
| --- | --- |
|  | **Process:** System components that can receive, modify, and output data. |
|  | **Entity:** Contributes data and information to system. Entities can also receive information from the system. |
|  | **Data Flow:** Indicates the movement of data to or from a process. |
|  | **Data Store:** The location where data is held either temporarily or permanently. |
|  | **System Boundary:** The definition between internal processes and external entities. |

**1.3.2 Context Diagram**

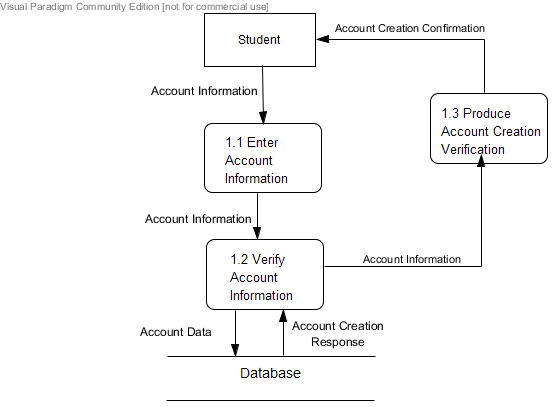


**1.3.3 Level 0 Diagram**

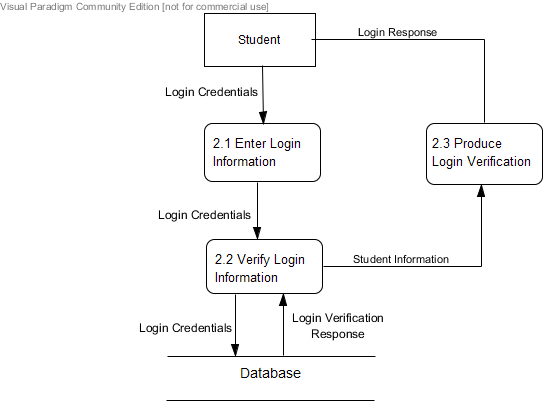


**1.3.4 Level 1 Diagrams**

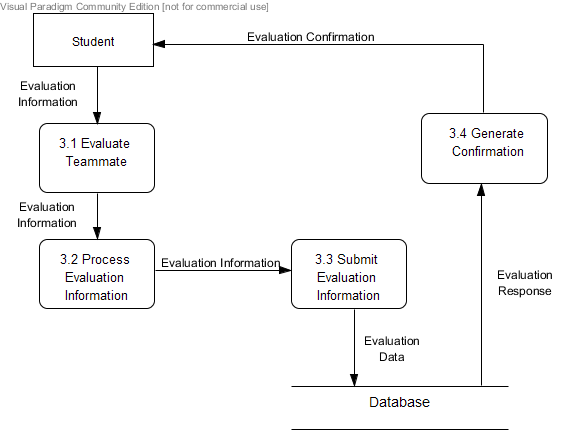
**1.3.4.1 Create Account**



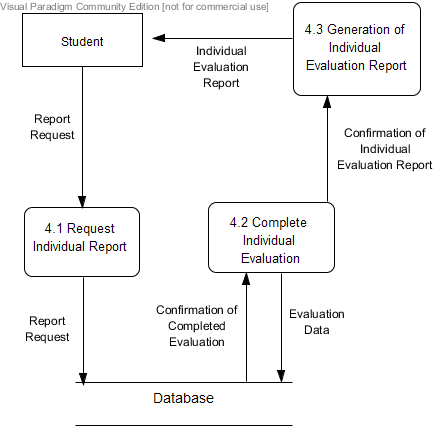
**1.3.4.2 Login**



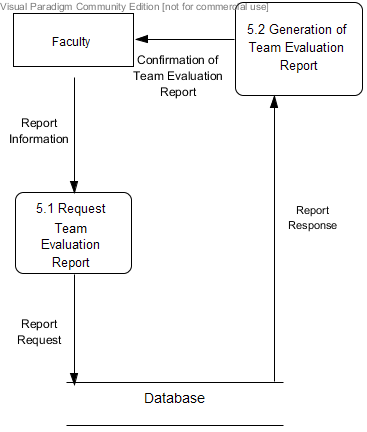
**1.3.4.3 Evaluate Teammate**



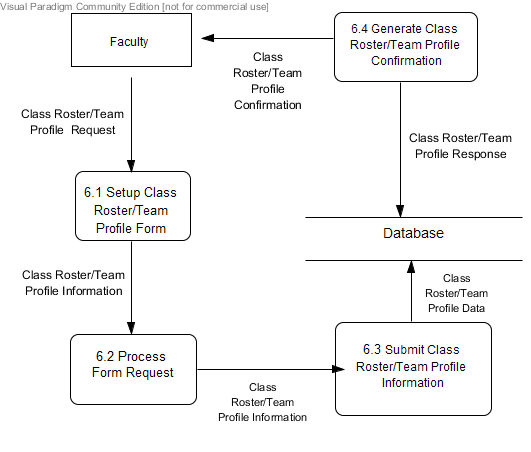
**1.3.4.4 View Individual Report**



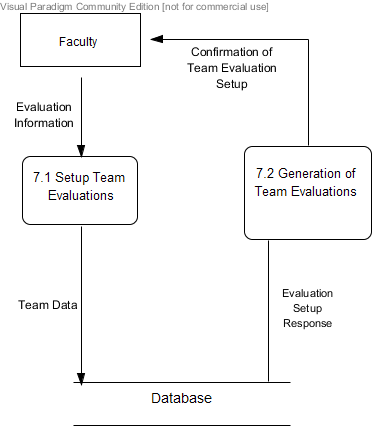
**1.3.4.5 View Team Report**



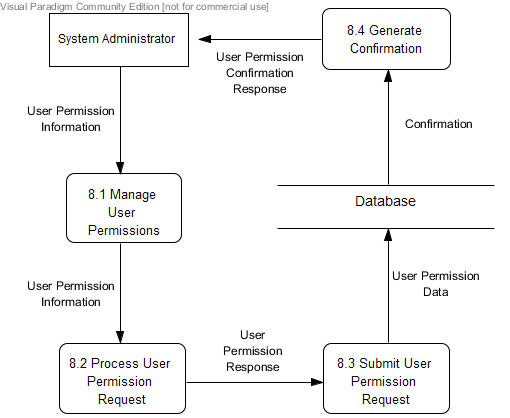
**1.3.4.6 Establish Class Roster/Team Profile**



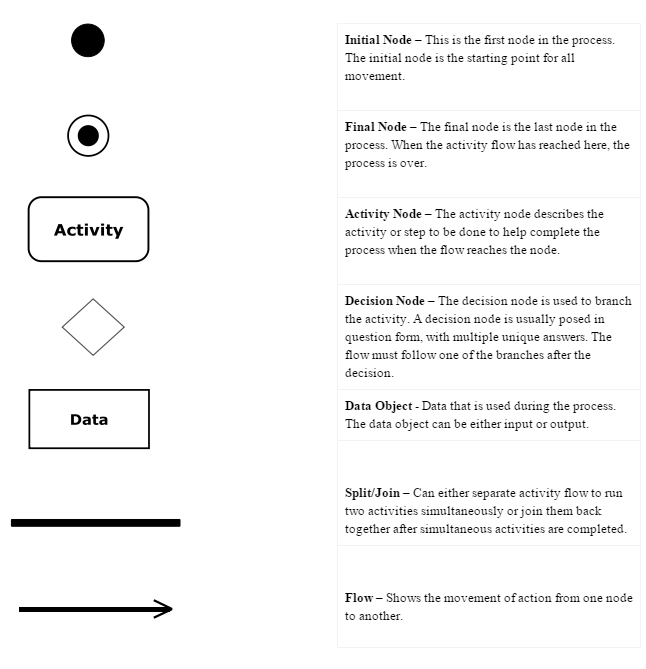
**1.3.4.7 Manage Team Evaluations**



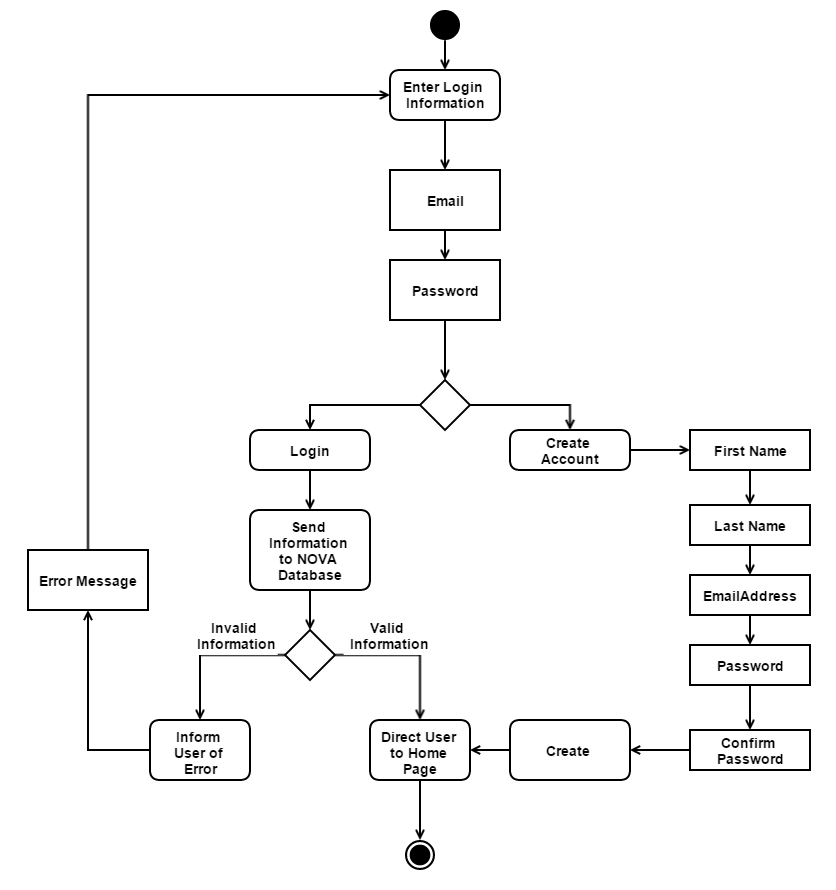
**1.3.4.8 Manage User Permissions**



ACTIVITY DIAGRAMS

**1.4.1 Activity Diagram Legend**

**1.4.2 Activity Diagram**

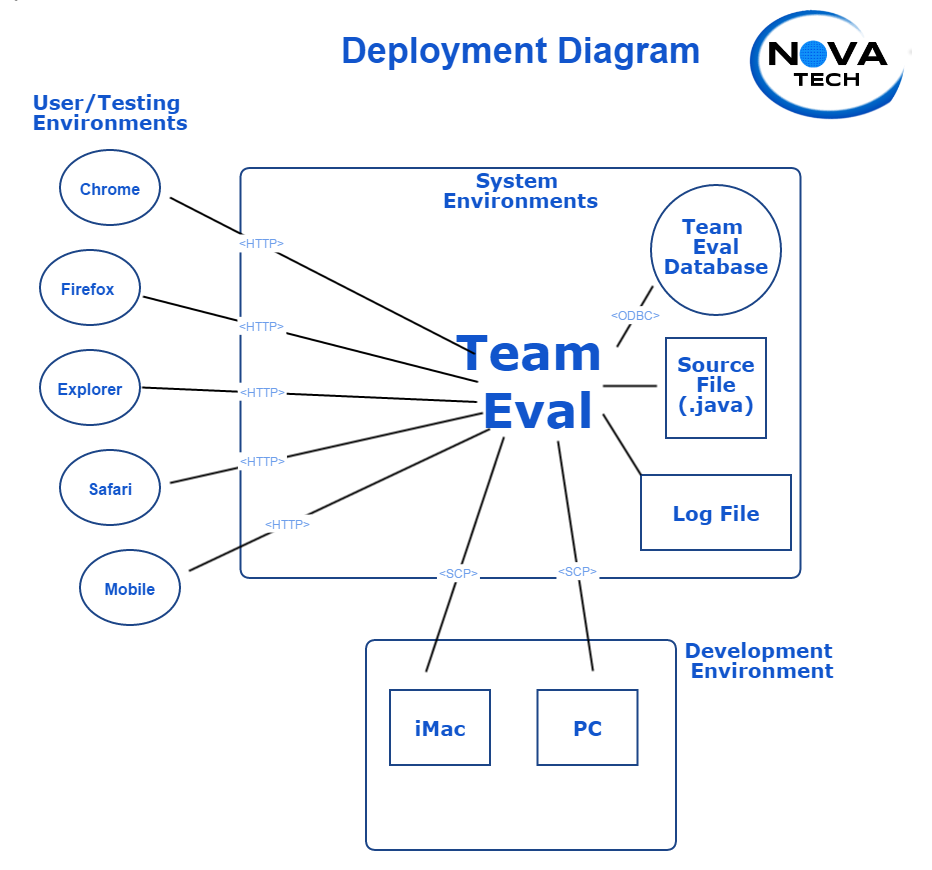


UML Deployment DIAGRAM

**1.5.1 UML Deployment Diagram Legend**

|  |  |
| --- | --- |
| **<HTTP>** | **HTTP - Hypertext Transfer Protocol defines how messages are formatted and transmitted, and what actions web servers and browsers should take in response to various commands.** |
| **<SCP>** | **SCP - Securely transfers computer files between a local host and a remote host** |
| **<ODBC>** | **ODBC - Open Database Connectivity is a standard programming language middleware for accessing database management systems.** |
|  | **System Boundary - This is where all the interactions occur. Represents what is within the system and outside of it.** |
|  | **Connection - Displays a relationship between boundaries.** |

**1.5.1 UML Deployment Diagram**



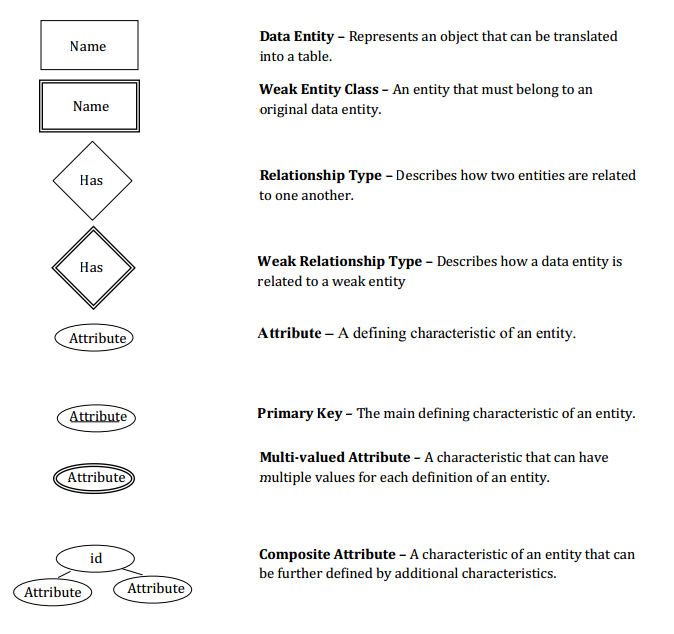
**1.6 Logical Data Stores**

**1.7 Logical Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Data Size** | **Description** | **Acceptable Input** | **Good Example of Input** | **Notes** |
| Username | String | 6-15 Chars | Username | A-Z, a-z, 0-9 | Dj04ferr |  |
| User\_pass | String | 6-15 Chars | Password | A-Z, a-z, 0-9, ASCII 33-47 | Pdj901584480 |  |
| UserID | String | 6-15 Chars | Student ID | A-Z, a-z, 0-9 | 901445531 | Unique |
| User\_Fname | String | 1-30  Chars | Username | A-Z, a-z | Jonny |  |
| User\_Lname | String | 1-30  Chars | Username | A-Z, a-z | Doe |  |
| User\_email | String | 6-50  Chars | Email | A-Z, a-z, 0-9 | Dj04ferr@siena.edu |  |
| PW\_reset | Boolean | 4-5 Chars | Password Reset Button | TRUE, FALSE | true |  |
| User\_Level | Integer | 1 Integer | User Permission Ranking | 0,1,2 | 2 | 0 = Student  1 = Faculty  2 = Admin |
| TeamID | String | 6-15 Chars | Team Id | A-Z, a-z, 0-9 | 113 | Unique |
| Number\_on\_team | Integer | 1-2 Integers | Number of team members | 1-10 | 4 |  |
| ClassID | Integer | 6-15 Chars | Class Id | A-Z, a-z, 0-9 | 17A | Unique |
| Member\_ID | String | 6-15 Chars | ID for a team member | A-Z, a-z, 0-9 | 901445531 | Unique |
| Member\_FN | String | 1-30 Chars | User First Name | A-Z, a-z, ‘, - | Hannah |  |
| Member\_LN | String | 1-30 Chars | User Last Name | A-Z, a-z, ‘, - | Cooper |  |
| Team\_Number | Integer | 1-2 Integers | Team Number | 0-20 | 9 |  |
| Faculty\_Member | String | 1-15 Chars | Faculty member name | A-Z, a-z, ‘, - | Lim |  |
| Faculty\_ID | String | 6-15 Chars | Faculty member ID number | A-Z, a-z, 0-9 | 901584480 | Unique |

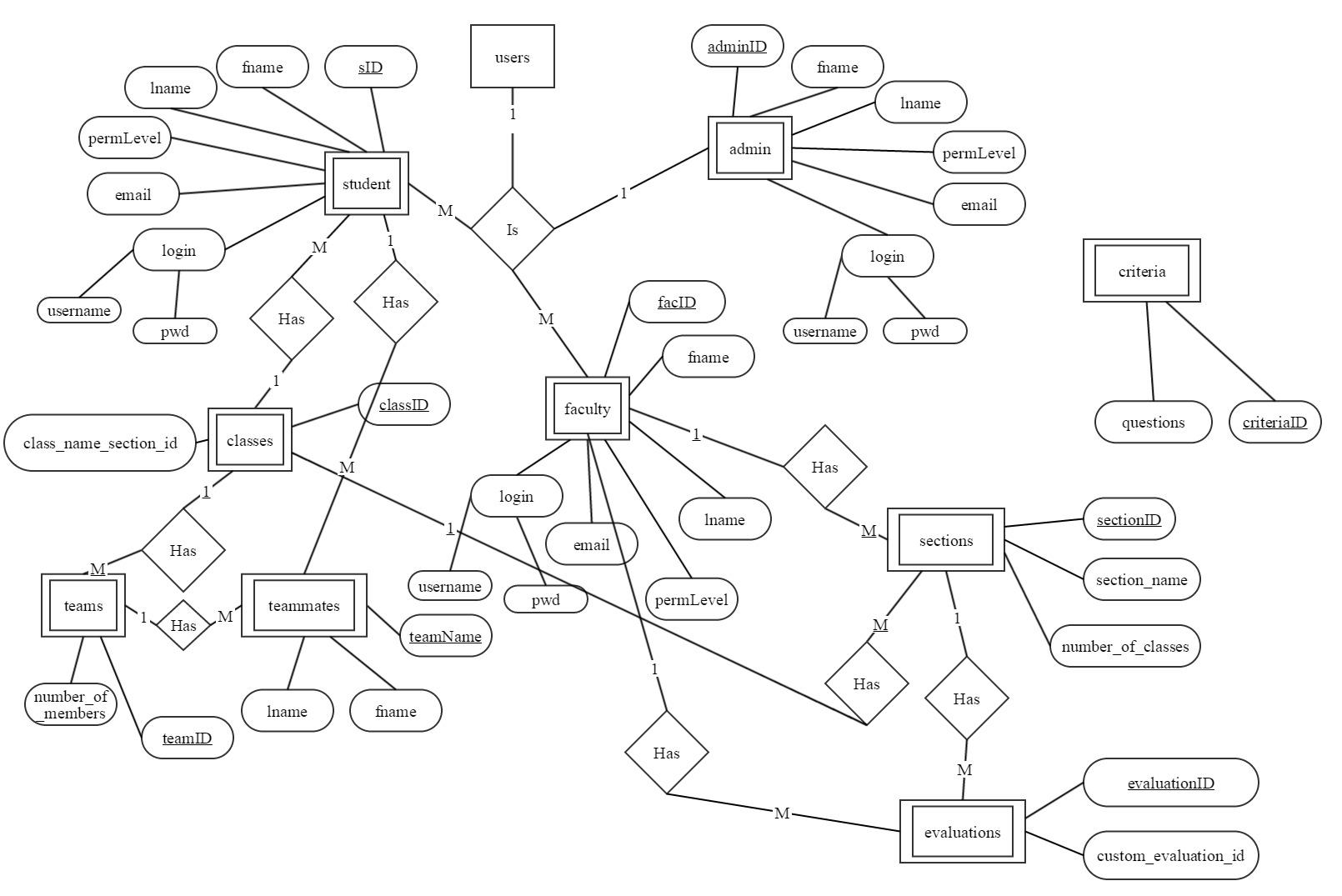
Logical Data model (E/R Diagram)

**2.1 E/R Diagram Legend**

**An ER Diagram is an entity-relationship model that abstractly describes a database.** 

**2.2 E/R Diagram**

**The following image is the ER Diagram for Teams101 - Team Evaluation.**



Physical Data model (Relational SChema)

**3.1 Relational Schema**

**From the ER Diagram, a relational schema can be made. The relational schema is another way to describe a database.**

users

student (sID , fname, lname, login, pwd, email, permLevel)

faculty (facId, login, pwd, fname, lname, email, level)

admin (adminID, login, pwd, fname, lname, email, level)

faculty processes

sections (sectionID , section\_name, number\_of\_classes, faculty\_ID (fk references faculty table), custom\_evaluation\_id (fk references evaluations)

\*\*custom\_eval\_id is for if faculty users want to uniquely generate evaluations and questions on those evaluations for each specific section

classes ( classID, class\_name, section\_id (fk references sections) )

teams ( teamID, classID (fk references classes), number\_of\_members )

teammates ( sID (fk references student) , teamID (fk references teams), fname, lname, )

criteria table - used to generate tests, we would query this table, grab all the information from it, and use it to generate some type of evaluation template for students to fill out

criteria ( criteria\_ID, criteria\_language, parent\_1, parent\_2)

parent\_1 would be the column (if you look at the criteria spreadsheet that Sara shared with us)

parent\_2 would be the row that it is in

criteria\_language is the actual language of the criteria i.e. “is ready for work”

criteria\_ID was thinking we just start at 1 and go up from there, just so we can more easily access certain criterias

evaluations (evaluationID , facID (fk references faculty) )

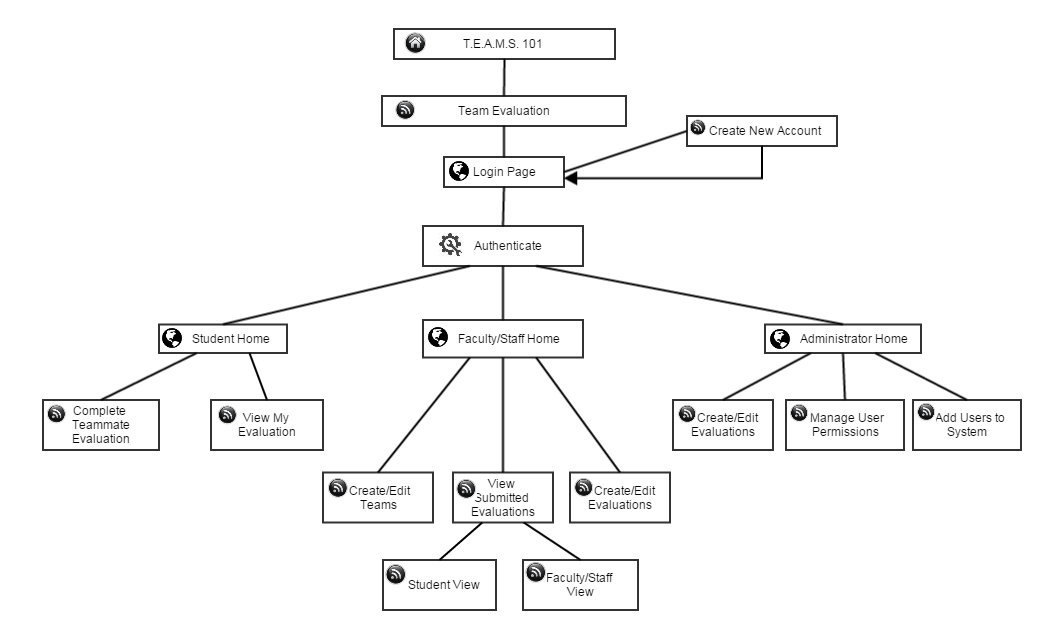
This table will have at least one row, it will be with an evaluation ID of 0 that represents the default evaluation for if faculty members do not want or need to customize

Architectural Design Specification

**4.1 Website Map Legend**

|  |  |
| --- | --- |
|  | **Home Page:** The first page a user accesses when navigating to T.E.A.M.S. 101 |
|  | **System Interaction:** Option visible on current web page for user to interact with |
|  | **Web Page:** Name of web page user is currently accessing |
|  | **System Action:** Action being carried out by system |
|  | **Page Redirect:** Relocates a user to another web page |
|  | **Link:** Connection between web pages and system interactions |

**4.1.1 Login Website Map**



**4.2 Parameter Specifications**

**4.3 Logical Data Structures**

**4.4 Functional Descriptions**

**4.5 Packaging Specification**

Dr. Eddy will receive the Teammate Evaluation software as well as a Web Application of T.E.A.M.S 101 electronically with all of the documentation after the Acceptance Test is completed. The software will be hosted on the server/domain given to Nova Tech, however, Dr. Eddy may choose to switch this to a personal server/domain name to use T.E.A.M.S 101 for team based projects. Nova will provide every component necessary to run the system to evaluate teammates for projects.

Test Plans

**5.1 Test Directory**

**5.2 Unit Test 1**

**5.3 Unit Test 2**

Pseudocode Routines

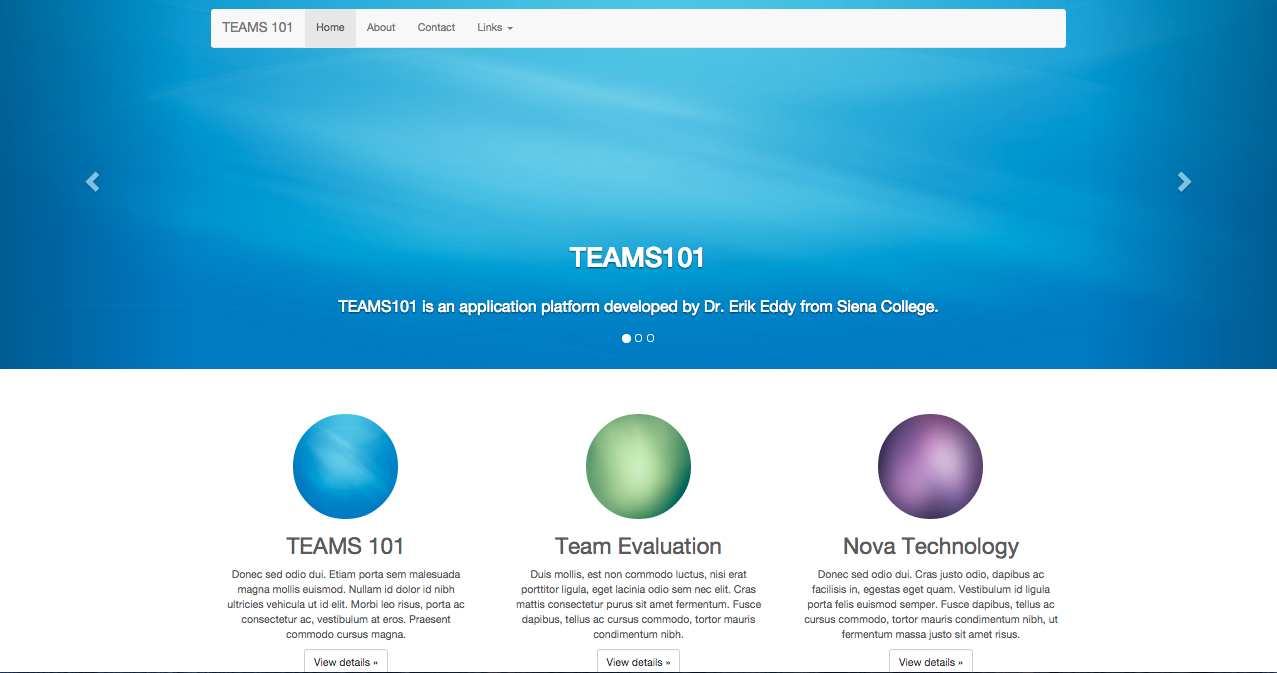
**6.1 Module 1**

**6.2 Module 2**

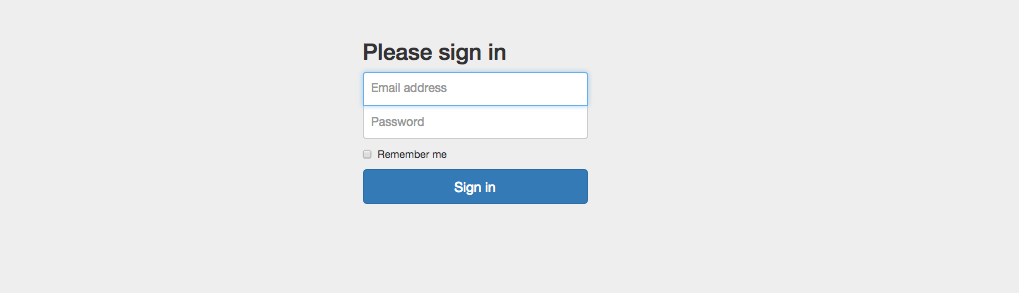
**6.3 Web Application**

Prototypes

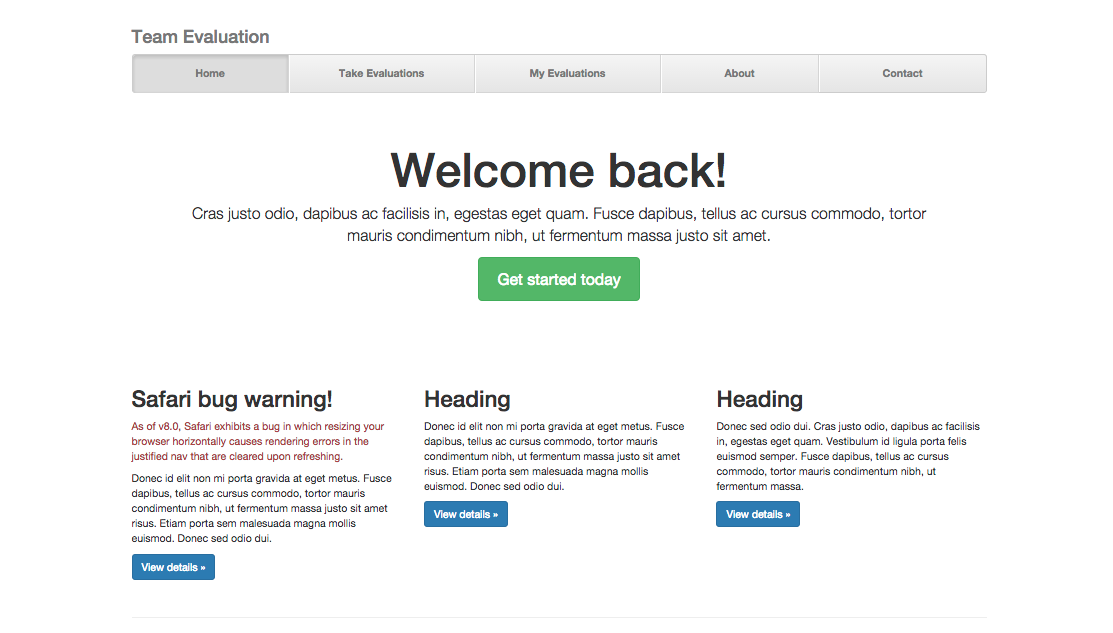
**7.1 TEAMS 101**



**7.2 Team Evaluation Login**



**7.3 Team Evaluation Home**



Appendix/Glossary

**Glossary of Terms**

**Actor**: Actors that interact with the system through sues/actors can be human or non human

**Agile method:** Agile software development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional teams

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

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

**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

**GIMP (GNU Image Manipulation Program):** Image retouching and editing tool released as free and open-source software by creators Spencer Kimball and Peter Mattis

**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

**mySQL (Structured Query Language):** Programming language designed to manage data and develop databases

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

**Nova Tech:** Team name

**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

**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

**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

**SQL Developer:** Program used to create and modify database

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

Diagram

**TEAMS 101 - Team Evaluation:** Project name

**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

**Unit Te**

**sting:** A testing method where the system is broken down into units and each unit is tested

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

**Visual Paradigm:** a UML CASE Tool supporting UML 2, SysML and Business Process Modeling Notation (BPMN) from the Object Management Group (OMG). In addition to modeling support, it provides report generation and code engineering capabilities including code generation. It can reverse engineer diagrams from code, and provide round-trip engineering for various programming languages.

**Website Map:** A list of pages of a website accessible to users