**Design Document**

**Graduate Teaching Assistantship Management System**

**Team 18**

Tubach Niko, Tyukina Iryna, Urquhart Kyle, Verbano Anthony

**Contents page**

**Section Page Number**

1. Introduction

1.1 Background 3

1.2 Objective 3

1.3 Technology Used 3

2. System Description

2.1 System Overview 4

2.2 User Types and Features 4

2.3 Process Flow 4

3. System Design

3.1 Conceptual Design (ER Diagram) 6

3.2 Logical Design (Relational Database Schema) 7

3.3 Normalization 11

3.3 Physical database Design 11

**1. Introduction**

**1.1 Background**

Each spring semester, faculty members of the department nominate Ph.D. students for a GTA (Graduate Teaching Assistantship) for the next school year. These nominations are evaluated by the Graduate Committee (GC) consisting of several professors in the department. The best nominees are selected based on their academic performance.

GTA Management System is designed to maintain GTA nominees in a suitable way. GTAMS is a web-application which avoids manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to Graduate Committee Members. It is very easy to manage historical data in database. Faculty members can easily initiate a nomination. Nominees can conveniently access and fill out an online information form if they are interested in GTA. Using this system, Graduate Committee Members can efficiently review the performance about a nominees and score them.

**1.2 Objective**

The objective of the project is to develop GTAMS to centrally handle the information of all the Graduate Teaching Assistantship nominations and to provide access to this information with an easy to use web-based interface that can be accessed from any device with basic html rendering capabilities.

GTAMS allows:

* the system administrator to set up a new nomination session
* the faculty members to nominate Ph.D. students for a GTA (Graduate Teaching Assistantship) for the next school year
* the nominee to conveniently access an online application form
* the Graduate Committee (GC) member to review the performance about a nominee and to score him/her

GTA Management System can be used by education institutes to maintain the records of GTA candidates easily. Achieving this objective is difficult using a manual system as the information is scattered, can be redundant and collecting relevant information may be very time consuming.

**1.3 Technology Used**

* Programming language: MySQL, PHP
* Front-End: HTML, CSS, JavaScript
* Hardware interface: 4Gb RAM, WINDOWS 7
* Database: MySQL Server
* Web Server: Apache

**2. System Description**

**2.1 System Overview**

The GTAMS website manages nominee’s applications and selections. The website should be user-friendly and reliable. GTAMS is intended to be a stand-alone product andshould not depend on the availability of other website/s. The system will also have an administrator who can perform all actions related to control and management of the website.

**2.2 User Types and Features**

There are four different type of users, supplied with their own GTAMS interfaces:

* Administrator.
* Nominator
* Nominee
* GC member and GC Chair

The features that are available to the Administrator are:

* Can log in into the system and perform any of the available operations
* Can set up a new nomination session
* Can administrate and support the system
* Can close the session
* Can save all the information in the database for future reference.

The features that are available to the nominator are:

* Can log in to use the system
* Can initiate a nomination an existing or an incoming Ph.D. student for a GTA
* Can verify the nominee’s information and complete the nomination process

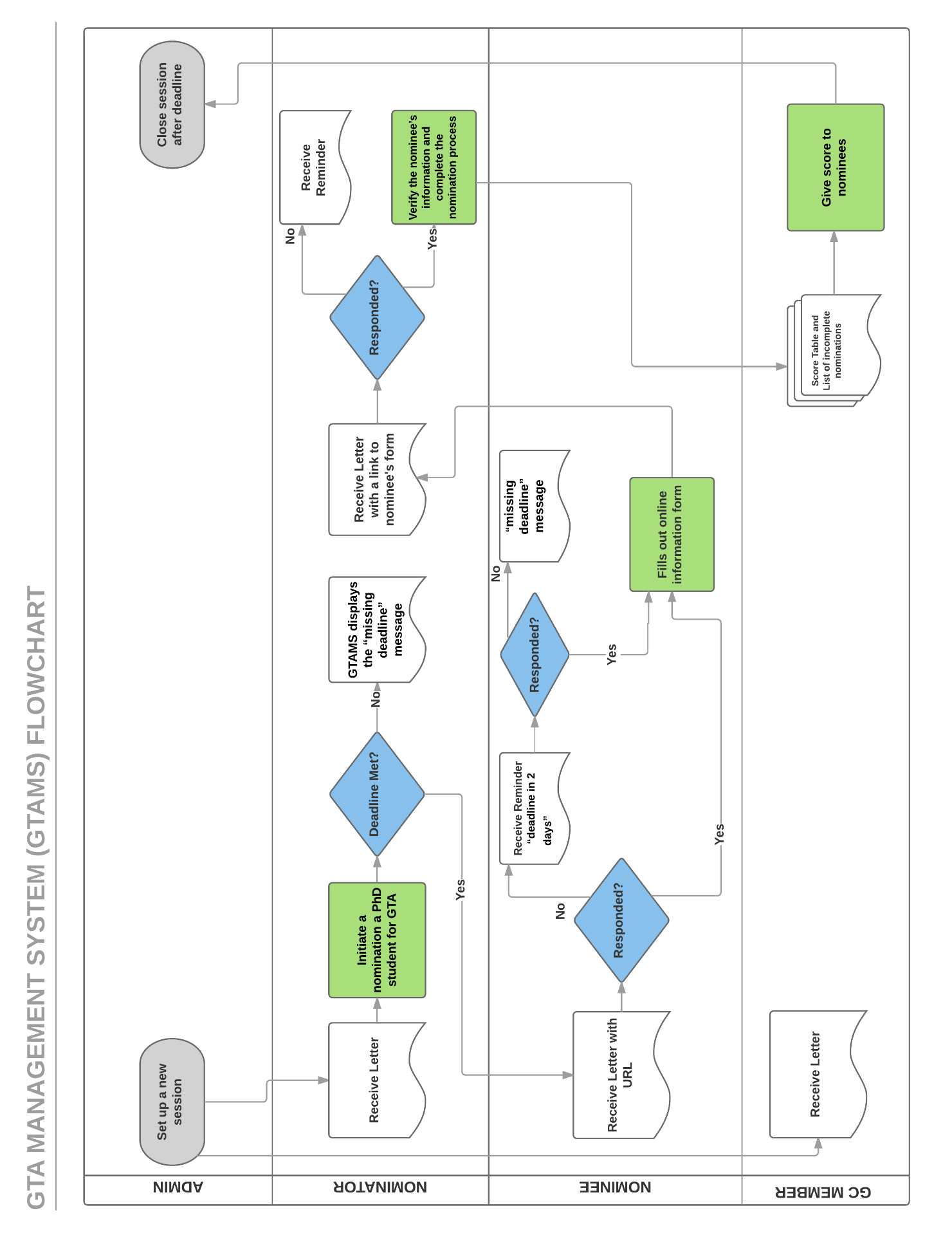
The features that are available to the nominee are:

* Can access and edit application form through a URL, which is provided in the email, sent out by GTAMS

The features that are available to the GC member and GC Chair:

* Can log in to use the system
* Can review the performance of a nominee
* Can score a nominee

**2.3 Process Flow**



**3. System Design**

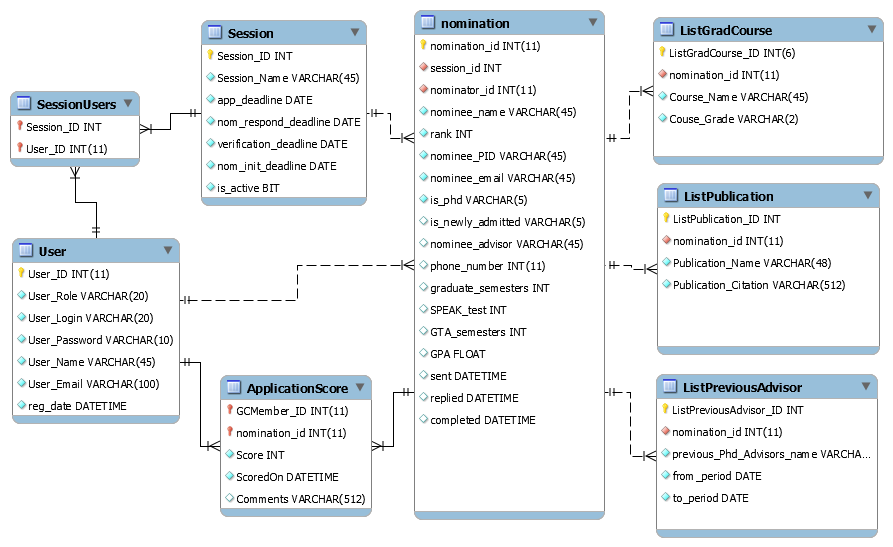
There are three phases of database system design model:

* Conceptual design
* Logical design
* Physical model in which data has been placed in database.

All three phases have been designed and described below.

**3.1 Conceptual Design (ER Diagram)**

Main entities of GTAMS are Session, User (administrator, GC members, GC Chair, nominators), Nomination and ApplicationScore which store data regarding GTA nominees e.g. name, email, scores, their nominators and PhD advisors etc. Possible entities and their relationship with each other are shown and also entities attributes.

****

**Cardinality of the relationship is described as:**

A session can have many nominations (1:\*)

A nomination can have many previous Ph.D. advisors (1:\*)

A nomination can have many completed graduate-level courses (1:\*)

A nomination can have many publications (1:\*)

A session can have many users (1:\*)

A user can participate in many sessions (1:\*)

A user (nominator) can nominee many nominees (1 :\*)

A nomination can have many scores (each from different GC Member) (1 :\*)

A user (GC Member) can give many scores (one for different nominees) (1 :\*)

**Identifying relationship** is when child object is dependent on parent object.

Shown as dotted line: - - - - - - - - -

**Non**-**identifying relationship** is when the primary key attributes of the parent must not become primary key attributes of the child

Defined as continuous line:

The diagram shows Nomination entity is dependent on Session and User entities. ListGradCourse, ListPublication and ListPreviousAdvisor entities are dependent on Nomination entity because for every graduate course, publication and previous PhD advisers belong to a nomination. This diagram shows relations dependencies and helps to reduce redundancy.

**3.2 Logical Design (Relational Database Schema)**

Relations between entities and try to summarize them into proper relations (tables):

**Entities & their Relationships Detail**

**Table: session**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Primary key** | **Data type** | **Not NULL** | **Comment** |
| Session\_ID | Yes | int | Yes |  |
| Session\_Name | No | varchar | Yes |  |
| app\_deadline | No | date | Yes |  |
| nom\_respond\_deadline | No | date | Yes | Deadline for a nominee to respond to a nomination |
| verification\_deadline | No | date | Yes | Deadline for the nominator to verify nominee’s information and complete the nomination |
| nom\_init\_deadline | No | date | Yes | Deadline for a faculty member to initiate a nomination |
| is\_active | No | bit | Yes |  |

**Relationships:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Relationship type** | **Parent** | **Child** |
| session\_sessionusers | Identifying | session | sessionusers |
| session\_nomination | Non Identifying | session | nomination |

**Table: sessionusers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Primary key** | **Data type** | **Not NULL** | **Comment** |
| Session\_ID | Yes | int | Yes |  |
| User\_ID | Yes | int | Yes |  |

**Relationships:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Relationship type** | **Parent** | **Child** |
| User\_sessionusers | Identifying | User | sessionusers |
| session\_sessionusers | Identifying | session | sessionusers |

**Table: User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Primary key** | **Data type** | **Not NULL** | **Comment** |
| User\_ID | Yes | int | Yes |  |
| User\_Role | No | varchar | No | User Role can be ADMIN, GCMEMBER, GCCHAIR, NOMINATOR |
| User\_Login | No | date | No | User Login |
| User\_Password | No | date | No | User Password |
| User\_Name | No | date | No |  |
| User\_Email | No | date | No | Email |
| reg\_date | No | bit | No |  |

**Relationships:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Relationship type** | **Parent** | **Child** |
| User\_sessionusers | Identifying | User | sessionusers |
| User\_nomination | Non Identifying | User | nomination |
| User\_ApplicationScore \_ | Identifying | User | ApplicationScore |

**Table: nomination**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Primary key** | **Data type** | **Not NULL** | **Comment** |
| nomination\_id | Yes | int(11) unsigned | Yes |  |
| session\_id | No | int(11) | Yes | the id for the session the Admin set up |
| nominator\_id | No | int(11) unsigned | Yes | it is User\_id with User\_Role="Nominator" |
| nominee\_name | No | varchar(45) | Yes |  |
| rank | No | int(11) | Yes |  |
| nominee\_PID | No | varchar(45) | Yes |  |
| nominee\_email | No | varchar(45) | Yes |  |
| is\_phd | No | varchar(5) | Yes | Is the nominee currently a Ph.D. student in the Department of Computer Science? |
| is\_newly\_admitted | No | varchar(5) | No |  |
| nominee\_advisor | No | varchar(45) | No |  |
| phone\_number | No | int(11) | No |  |
| graduate\_semesters | No | int(11) | No |  |
| SPEAK\_test | No | int(11) | No | 1, 2, 3 = yes, no, graduated from US school |
| GTA\_semesters | No | int(11) | No |  |
| GPA | No | float | No |  |
| sent | No | datetime | No | when the nomination was sent to the user |
| replied | No | datetime | No | when the nomination was replied to by the nominee |
| completed | No | datetime | No | when the nominee form was confirmed by the nominator |

**Relationships:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Relationship type** | **Parent** | **Child** |
| session\_nomination | Non Identifying | session | nomination |
| User\_nomination | Non Identifying | User | nomination |
| nomination\_ListGradCourse | Non Identifying | nomination | ListGradCourse |
| nomination\_ListPublication | Non Identifying | nomination | ListPublication |
| nomination\_ListPreviousAdvisor | Non Identifying | nomination | ListPreviousAdvisor |
| nomination\_ApplicationScore | Identifying | nomination | ApplicationScore |

**Table: ListGradCourse**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Primary key** | **Data type** | **Not NULL** | **Comment** |
| ListGradCourse\_ID | Yes | int(6) unsigned | Yes |  |
| nomination\_id | No | int(11) unsigned | Yes |  |
| Course\_Name | No | varchar(45) | Yes |  |
| Couse\_Grade | No | varchar(2) | Yes |  |

**Relationships:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Relationship type** | **Parent** | **Child** |
| nomination\_ListGradCourse | Non Identifying | nomination | ListGradCourse |

**Table** **ListPublication**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Primary key** | **Data type** | **Not NULL** | **Comment** |
| ListPublication\_ID | Yes | int(10) unsigned | Yes |  |
| nomination\_id | No | int(11) unsigned | Yes |  |
| Publication\_Name | No | varchar(48) | Yes |  |
| Publication\_Citation | No | varchar(512) | Yes |  |

**Relationships:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Relationship type** | **Parent** | **Child** |
| nomination\_ListPublication | Non Identifying | nomination | ListPublication |

**Table:** **ListPreviousAdvisor**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Primary key** | **Data type** | **Not NULL** | **Comment** |
| previous\_Phd\_Advisors\_ID | Yes | int(10) unsigned | Yes |  |
| nomination\_id | No | int(11) unsigned | Yes |  |
| previous\_Phd\_Advisors\_name | No | varchar(45) | Yes |  |
| from\_period | No | date | Yes |  |
| to\_period | No | date | Yes |  |

**Relationships:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Relationship type** | **Parent** | **Child** |
| nomination\_ListPreviousAdvisor | Non Identifying | nomination | ListPreviousAdvisor |

**Table** **ApplicationScore**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column name** | **Primary key** | **Data type** | **Not NULL** | **Comment** |
| GCMember\_ID | Yes | int(11) unsigned | Yes |  |
| nomination\_id | Yes | int(11) unsigned | Yes |  |
| Score | No | int(11) | Yes |  |
| ScoredOn | No | datetime | Yes |  |
| Comments | No | varchar(512) | No |  |

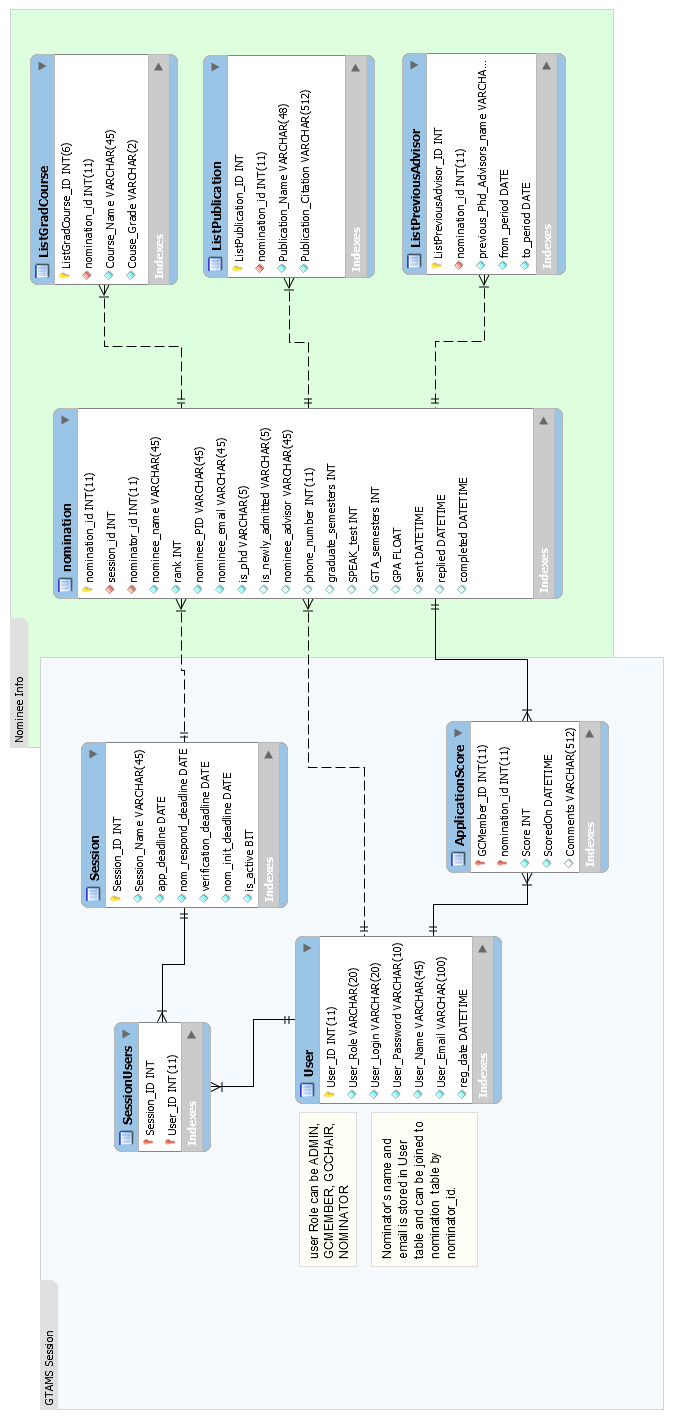
**Relationships:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Relationship type** | **Parent** | **Child** | **Cardinality** |
| nomination\_ApplicationScore | Identifying | nomination | ApplicationScore |  |
| User\_ApplicationScore \_ | Identifying | User | ApplicationScore | Zero Or More |

**3.3 Normalization**

Database tables are designed in the third normal form (3NF). In order to support third normal form, all tables must satisfy the requirements of the first normal form (1NF) and the second normal form (2NF). Tables have been designed to keep optimization in mind. Normalization helps to enforce data integrity. The ApplicationScore is separated from nomination table to avoid data duplication. ListGradCourse, ListPublication ,ListPreviousAdvisor tables are split from nomination table to avoid data duplication and to support one-to-many relationship.

**3.4 Physical database Design**



After logical design of database, we put data in database to create table and their attributes, we run following queries in MYSQL console:

CREATE TABLE IF NOT EXISTS `gtass`.`User` (

`User\_ID` INT(11) UNSIGNED NOT NULL,

`User\_Role` VARCHAR(20) NOT NULL COMMENT 'User Role can be ADMIN, GCMEMBER, GCCHAIR, NOMINATOR',

`User\_Login` VARCHAR(20) NOT NULL COMMENT 'User Login',

`User\_Password` VARCHAR(10) NOT NULL COMMENT 'User Password',

`User\_Name` VARCHAR(45) NOT NULL,

`User\_Email` VARCHAR(100) NOT NULL COMMENT 'Email',

`reg\_date` DATETIME NOT NULL,

PRIMARY KEY (`User\_ID`),

INDEX `ix\_userrole` (`User\_Role` ASC),

UNIQUE INDEX `User\_Login\_UNIQUE` (`User\_Login` ASC))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `gtass`.`Session`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `gtass`.`Session` (

`Session\_ID` INT NOT NULL AUTO\_INCREMENT,

`Session\_Name` VARCHAR(45) NOT NULL,

`app\_deadline` DATE NOT NULL,

`nom\_respond\_deadline` DATE NOT NULL,

`verification\_deadline` DATE NOT NULL,

`nom\_init\_deadline` DATE NOT NULL,

`is\_active` BIT NOT NULL,

PRIMARY KEY (`Session\_ID`),

UNIQUE INDEX `GTASession\_Name\_UNIQUE` (`Session\_Name` ASC))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `gtass`.`SessionUsers`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `gtass`.`SessionUsers` (

`Session\_ID` INT NOT NULL,

`User\_ID` INT(11) UNSIGNED NOT NULL,

PRIMARY KEY (`Session\_ID`, `User\_ID`),

INDEX `fk\_SessionGCMember\_User1\_idx` (`User\_ID` ASC),

CONSTRAINT `fk\_SessionGCMember\_Session1`

FOREIGN KEY (`Session\_ID`)

REFERENCES `gtass`.`Session` (`Session\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_SessionGCMember\_User1`

FOREIGN KEY (`User\_ID`)

REFERENCES `gtass`.`User` (`User\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `gtass`.`nomination`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `gtass`.`nomination` (

`nomination\_id` INT(11) UNSIGNED NOT NULL,

`session\_id` INT NOT NULL COMMENT 'the id for the session the Admin set up',

`nominator\_id` INT(11) UNSIGNED NOT NULL COMMENT 'it is User\_id with User\_Role=\"Nominator\"',

`nominee\_name` VARCHAR(45) NOT NULL,

`rank` INT NOT NULL,

`nominee\_PID` VARCHAR(45) NOT NULL,

`nominee\_email` VARCHAR(45) NOT NULL,

`is\_phd` VARCHAR(5) NOT NULL COMMENT 'Is the nominee currently a Ph.D. student in the Department of Computer Science? The nominator\ncan click to select “Yes” or “No”',

`is\_newly\_admitted` VARCHAR(5) NULL,

`nominee\_advisor` VARCHAR(45) NULL,

`phone\_number` INT(11) NULL,

`graduate\_semesters` INT NULL,

`SPEAK\_test` INT NULL COMMENT '1, 2, 3 = yes, no, graduated from US school',

`GTA\_semesters` INT NULL,

`GPA` FLOAT NULL,

`sent` DATETIME NULL COMMENT 'when the nomination was sent to the user',

`replied` DATETIME NULL COMMENT 'when the nomination was replied to by the nominee',

`completed` DATETIME NULL COMMENT 'when the nominee form was confirmed by the nominator',

PRIMARY KEY (`nomination\_id`),

INDEX `fk\_ApplicationForm\_Session1\_idx` (`session\_id` ASC),

INDEX `fk\_nomination\_User1\_idx` (`nominator\_id` ASC),

CONSTRAINT `fk\_ApplicationForm\_Session1`

FOREIGN KEY (`session\_id`)

REFERENCES `gtass`.`Session` (`Session\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_nomination\_User1`

FOREIGN KEY (`nominator\_id`)

REFERENCES `gtass`.`User` (`User\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `gtass`.`ListPreviousAdvisor`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `gtass`.`ListPreviousAdvisor` (

`ListPreviousAdvisor\_ID` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`nomination\_id` INT(11) UNSIGNED NOT NULL,

`previous\_Phd\_Advisors\_name` VARCHAR(45) NOT NULL,

`from\_period` DATE NOT NULL,

`to\_period` DATE NOT NULL,

PRIMARY KEY (`ListPreviousAdvisor\_ID`),

INDEX `fk\_Previous\_Phd\_Advisors\_nomination1\_idx` (`nomination\_id` ASC),

CONSTRAINT `fk\_Previous\_Phd\_Advisors\_nomination1`

FOREIGN KEY (`nomination\_id`)

REFERENCES `gtass`.`nomination` (`nomination\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `gtass`.`ListPublication`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `gtass`.`ListPublication` (

`ListPublication\_ID` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`nomination\_id` INT(11) UNSIGNED NOT NULL,

`Publication\_Name` VARCHAR(48) NOT NULL,

`Publication\_Citation` VARCHAR(512) NOT NULL,

PRIMARY KEY (`ListPublication\_ID`),

INDEX `fk\_ListPublication\_nomination1\_idx` (`nomination\_id` ASC),

CONSTRAINT `fk\_ListPublication\_nomination1`

FOREIGN KEY (`nomination\_id`)

REFERENCES `gtass`.`nomination` (`nomination\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `gtass`.`ApplicationScore`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `gtass`.`ApplicationScore` (

`GCMember\_ID` INT(11) UNSIGNED NOT NULL,

`nomination\_id` INT(11) UNSIGNED NOT NULL,

`Score` INT NOT NULL DEFAULT 0,

`ScoredOn` DATETIME NOT NULL,

`Comments` VARCHAR(512) NULL,

PRIMARY KEY (`GCMember\_ID`, `nomination\_id`),

INDEX `fk\_ApplicationScore\_User1\_idx` (`GCMember\_ID` ASC),

INDEX `fk\_ApplicationScore\_nomination1\_idx` (`nomination\_id` ASC),

CONSTRAINT `fk\_ApplicationScore\_User1`

FOREIGN KEY (`GCMember\_ID`)

REFERENCES `gtass`.`User` (`User\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_ApplicationScore\_nomination1`

FOREIGN KEY (`nomination\_id`)

REFERENCES `gtass`.`nomination` (`nomination\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `gtass`.`ListGradCourse`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `gtass`.`ListGradCourse` (

`ListGradCourse\_ID` INT(6) UNSIGNED NOT NULL AUTO\_INCREMENT,

`nomination\_id` INT(11) UNSIGNED NOT NULL,

`Course\_Name` VARCHAR(45) NOT NULL,

`Couse\_Grade` VARCHAR(2) NOT NULL,

PRIMARY KEY (`ListGradCourse\_ID`),

INDEX `fk\_ListGradCourse\_nomination1\_idx` (`nomination\_id` ASC),

CONSTRAINT `fk\_ListGradCourse\_nomination1`

FOREIGN KEY (`nomination\_id`)

REFERENCES `gtass`.`nomination` (`nomination\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;