**Part I - Project Information**

**Introduction**

My team has created the Database Academy database system for our project. This system has been designed to replicate an internal search tool for a high school (inspired by the UVA Internal People Search website). In order to create an effective search tool that could accurately represent a high school ecosystem, we had to be able to accurately show the

In order to accurately represent the high school environment, we first created several entity sets to represent the numerous elements of a high school. These included

Introduction (2%)

- Describe the world you are trying to model. Include everything that you feel is relevant. This should be around 1/2 to a full page.

**Requirements**

* If the user is a student attending Database Academy, a teacher teaching at the academy, or a parent of a student at the academy, they should be able to log in using their email and a password (we assume people logging in already know a username/password combination for a student/teacher/parent they are trying to log in as).
* Users should be able to log out.
* Any user should be able to log in and go to the home screen of the website, which features a menu bar (which is consistent across all screens of the website), and a search bar.
* From the home screen, any user should be able to search for a specific query in the search bar and be presented with a page that lists all of the names of people (students/teachers/parents) whose first or last name match the search query.
* Any user should be able to click on any one of these names and be presented with details of that person, which is unique to their role at the academy.
* \*\*\*Students should only be able to view their own GPAs.
* \*\*\*\*Parents should only be able to view the GPAs of the students associated with them.
* \*\*\*Only teachers should be able to view their own salaries.
* Any user should be able to navigate to the search page for Clubs and search for specific clubs at the academy. Once they are presented with the list of clubs, they should be able to click on any club name to view the specific details (this same functionality should also exist for courses).
* Teachers should be able to update, delete, and add new information to the website for any type of person, club, and course (except salaries of other teachers, which they cannot see).
* \*\*\*All of the data in the database should be able to be exported by a Teacher as a \_\_\_\_ and printed. The salaries of other professors should not show up.
* \*\*\*\*Special Functionality

**Part II - The Design Process**

• Explanation of design decisions that you made in creating your application (i.e. why you choose the type of app you did, why that language, if your database should be secure, did you secure it and how, anything interesting or relevant, etc.) (2%)

- Describe your thinking as to why you did certain things (around 1/2 a page)

• An E-R diagram (5%)

- Make sure you model EVERYTHING in your database! Don't leave out stuff!

• Database schema including integrity constraints (2%)

- Just give the schemas straight from your database with things like primary key, unique, not null, etc. Easy 2% here.

• Proof that your database is in Third Normal Form (5%)

- Take your tables and perform the tests we went over in class. Be thorough. Do NOT skip steps.

**Part III - Evaluation of Product**

• A description of the testing procedures for the database and the application (2%) - If you didn't test, you didn't do good design. You don't have to go into unit tests, etc., but just give a general idea of what you did to make sure that everything works.

• Sample data and sample queries from your application (2%)

- Give an idea of what your database does without having to load it.

• Grammar, spelling, etc. (2%)

- Your document should be proofread and free from error.