CSCI-3432-C

Database Design Report

Project Details:

The name of our project is Instrument Rentals, and our team members are Charles Garman, Tim White, and Maguire Mahalak.

Introduction:

This project involves creating a database and an accompanying web based application that help a variety of users handle instrument rental transactions safely, conveniently, and intelligently.

Problem Statement:

One thing that is often consistent among the different types of instruments is that they are expensive. This can be a major barrier for people who would otherwise want to learn how to play one, especially if they are weary about making such an investment in something they are not sure they will enjoy. Moving to instrument owners, many possess more than they use, and instead of letting their collection sit there they could instead be making money. A solution that would benefit both parties would be to rent instruments for a price lower than the full purchase cost. The customers can afford to try an instrument and so the renters can make money they otherwise would not be able to.

However, rentals cannot be unstructured. Without a system in place it could be tedious for customers to find someone who would want to rent out a specific instrument for a reasonable price. Additionally, customers would have no way to know how trustworthy potential renters are. Without this knowledge there could be various scams abound. As such, a rental service is required to solve this problem while handling the aforementioned issues. An excel file would not suit the needs of this service because data must meet specific constraints, and using an excel file to hold all details would make accomplishing this inconvenient. For example, in a database only a specific type of data can be stored for a certain attribute, limiting the chances of error, while an excel file would require additional checks. (365 Data Science, n.d.) Several people could be using the service at the same time as well, making robust multi-user support a necessity that an excel file cannot satisfy. (365 Data Science, n.d.) A database does not have these shortcomings, and provides a flexible, secure, and convenient way to maintain all the data the rental service needs.

Project Scope:

Our database will hold enough information to represent pertinent aspects of renters, the instruments they offer, and customers as well as automatically keep track of who is renting what, how long they have been doing so, how much money they owe, and to who. Actual payments and renting interactions will be simulated, but should an actual business use this database it would be sufficient.

When a user creates an account with the service they will need to supply a username, password, email, address, and credit card (stored as an aggregate of the card’s sixteen-digit number, security code, and expiration date). All available instruments will be viewable by a user who selects to use the service as a customer after login, organized by recency by default although more search options will be usable. Each instrument will list its name, date of posting, asking price (in dollars a day), owner, and the rating of its owner. Should a customer purchase an instrument the owner will be able to see this on their page, and the database will keep track of how long the instrument has been rented to the customer. All shipping will be high priority and only take one day, providing a consistent offset for the duration of the rental. In the event it takes longer for the instrument to arrive, customers can have this reflected in the rating they give the owner once they decide to send the instrument back. The money they owe to the owner will be recorded by the database until they pay it off.

When a user chooses to use the service to offer their instruments they will be able to see all the instruments they have registered and their status (available, hidden, or being rented). They can click on an individual instrument for more information, such as who is renting it and how long they have been doing so, as well as the daily price of the instrument. They will also be able to see their current rating as well as notices of pending payments.

The same account can be used for both renting instruments (as a customer) and renting them out (as a seller), and so, all users will have values that keep track of how many times they have been reviewed and the total number of points they have received (each rating is based on a five point system, where five is exceptional and 1 is atrocious). Their average score will be calculated by the database as it is needed, but not explicitly stored within it. This rating system is not used to evaluate customers, however, so it will only appear to other users in situations where said user is renting out instruments.

A specific password is stored to indicate that when someone uses it without entering a username they are an administrator, so they will be able to see all stored information about users, instruments, and payments. They will also have the ability to delete users, instruments, and payments, but due to relational integrity constraints a user may only be deleted if they have no relationships with any instruments or payments.

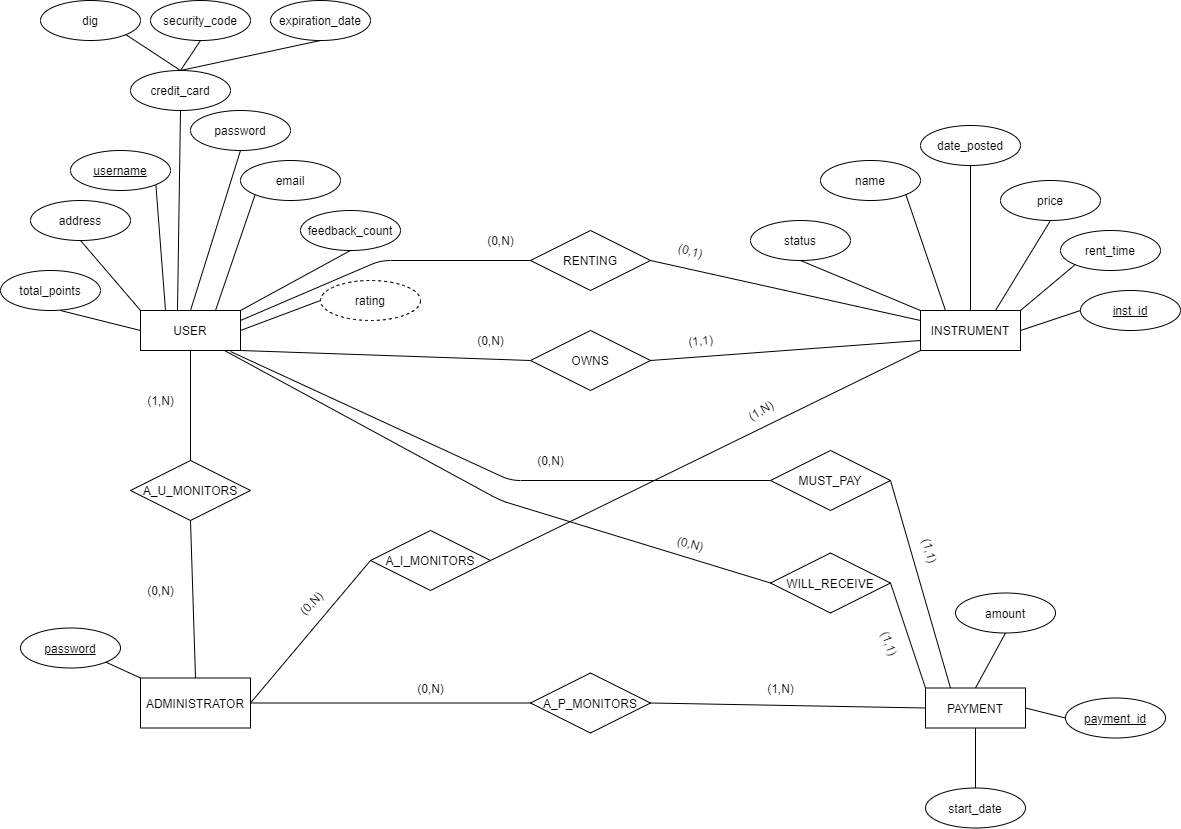
Target User:

Here is a real-life scenario that could transpire with the use of the instrument rental service. An individual has several instruments they no longer use, and they have heard of the instrument rental service and think that it is a good way to make money without having to part with their collection. They list their instruments in the application and set them to be visible. After a few days of not checking the website they log on to find that someone has requested to rent one of their instruments, a ukulele. They then ship it to them.

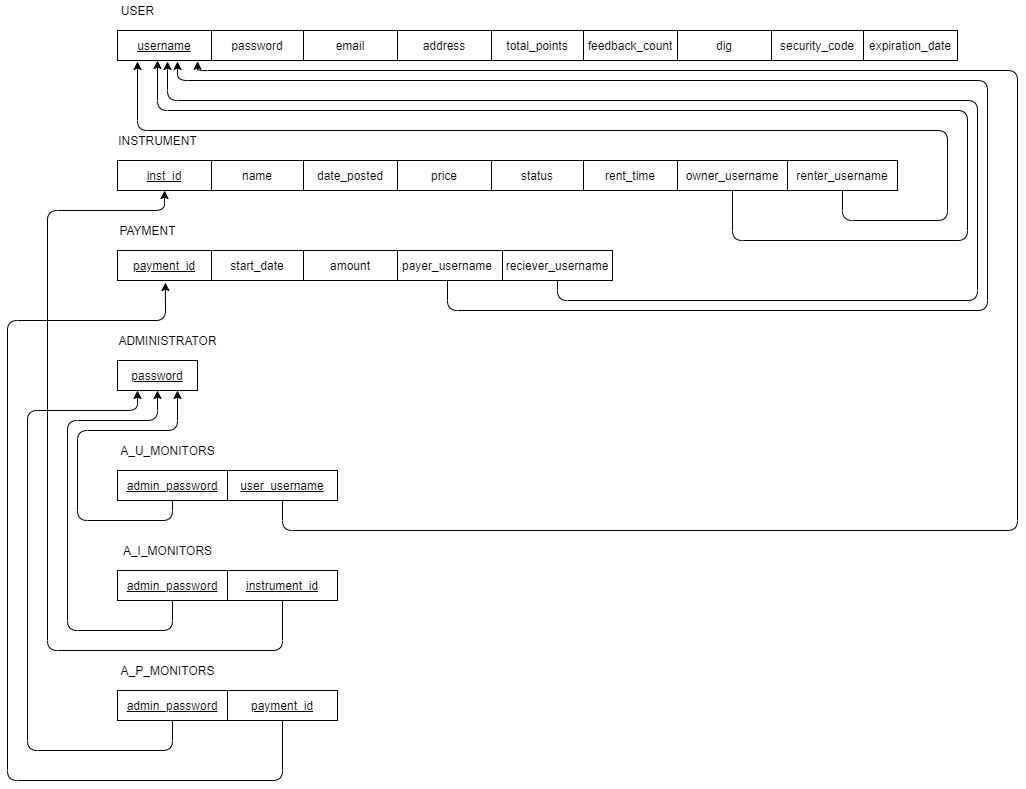
A customer who has been looking to rent a ukulele waits three days for the instrument to arrive, as opposed to the appropriate one. Additionally, it appears that the seller has not opened the case of their instrument in years. Strings are missing, and the ones that are intact are irreversibly out of tune. Upset, they elect to immediately send the ukulele back and leave a rating of an abysmal 0.5.

A service employee is managing the web application, and can see all information regarding registered users, their payments, and instruments after entering the administrator password. They then realize that a specific user has been rated poorly. Wanting to keep the integrity of the service in check, the administrator then contacts the user. After some discussion the user agrees to refund their client and that they will have their account deleted if they do not make significant improvements.

ER Diagram:



Relations:



Relations Mapping:

USER[username, password, email, address, total\_points, feedback\_count, dig, security\_code, expiration\_date]

INSTRUMENT[inst\_id, name, date\_posted, price, status, rent\_time, owner\_username, renter\_username]

PAYMENT[payment\_id, start\_date, amount, payer\_username, reciever\_username]

ADMINISTRATOR[password]

A\_U\_MONITORS[admin\_password, user\_username]

A\_I\_MONITORS[admin\_password, instrument\_id]

A\_P\_MONITORS[admin\_password, payment\_id]

Suggested Deliverables:

The application that uses our database will be a package of PHP files. It will create several user interfaces. The first will be for a general login. If a user needs to create an account they can click on a link and they will be sent to another page where they can enter their information. If the password entered at the login screen is logged in the database as an administrator password and they have not entered a username, the user will be brought to a page listing all the information stored in the database regarding users, their instruments, and payments. If the password is to a normal account a page asking if the user wishes to buy or sell will be displayed. If buy is selected they will be brought to an interface where they can view available instruments, rent them, send instruments back, and check what payments they owe. If sell is selected the interface will only list the instruments that the user owns. Clicking on an instrument will bring them to a new page listing all of its details and give them options for changing the instrument’s status or possibly to delete it. The main seller page will also have a button that leads to another page that allows the user to add a new instrument.

A simple instruction manual will be created to explain the functions of each page to any potential users. This manual will include documents that illustrate and explain the database’s design, to help clear up any potential ambiguities concerning how the application is working. Code will also have comments to aid understanding and possibility for future expansion.

Definitions/Glossary:

Attribute- “a database field. Attributes describe the instances in the column of a database”

(Techopedia, n.d.e)

Composite attribute- “an attribute composed of several other simple attributes”

(Techopedia, n.d.e)

Database- “any collection of data, or [information](https://www.britannica.com/science/information-science), that is specially organized for rapid search and retrieval by a [computer](https://www.britannica.com/technology/computer)”

(The Editors of Encyclopaedia Britannica, 2020)

Derived attribute- “these are derived from other attributes, either directly or through specific formula results”

(Techopedia, n.d.e)

ER Diagram- “a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system”

(Lucidchart, n.d.)

Entity- “a thing, person, place, unit, object or any item about which the data should be captured and stored”

(Centriqs, n.d.)

Foreign key- “a column or group of columns in a relational database table that provides a link between data in two tables”

(Techopedia, n.d.c)

Key Attribute- “an ID, key, letter or number that uniquely identifies that item”

(Techopedia, n.d.e)

Min,Max Notation- “a pair of numbers(m, n) that appear on the connecting line between the entities and their relationships. The minimum number of times an entity can appear in a relation is represented by m whereas, the maximum time it is available is denoted by n”

(supriya\_saxena, 2020)

Multivalued attribute- “an attribute where more than one description can be provided”

(Techopedia, n.d.e)

Relation- “a table in a relational database”

(Techopedia, n.d.a)

Relational Database- “organizes data into tables which can be linked—or *related—*based on data common to each”

(IBM Cloud Education, 2019)

Relationship- “a situation that exists between two relational database tables when one table has a foreign key that references the primary key of the other table”

(Techopedia, n.d.d)

Table- “a named relational database data set that is organized by rows and columns”

(Techopedia, n.d.b)

User Interface- “the means in which a person controls a software [application](https://techterms.com/definition/application) or hardware device” (Christensson, 2009)

References:

365 Data Science. (n.d.). *Database vs Spreadsheet- Advantages and Disadvantages*. Retrieved September 28, 2020, from <https://365datascience.com/explainer-video/database-vs-spreadsheet/>

Centriqs. (n.d.). *Database Entity: Definition, Relationship, Attributes, and Settings*. Retrieved September 27, 2020, from <http://www.centriqs.com/database-software/database-entity.php>

Christensson, P. (2009, March 31). *User Interface Definition*. Tech Terms. Retrieved 2020, Sep 27, from <https://techterms.com/definition/user_interface>

IBM Cloud Education. (2019, August 6). *Relational Databases*. IBM. Retrieved September 27, 2020, from <https://www.ibm.com/cloud/learn/relational-databases>

Lucidchart. (n.d.). *What is an Entity Relationship Diagram (ERD)?* Retrieved September 27, 2020, from <https://www.lucidchart.com/pages/er-diagrams>

Techopedia. (n.d.). *What is Relation? - Definition from Techopedia*. Retrieved September 27, 2020, from <https://www.techopedia.com/definition/21677/relation>

Techopedia. (n.d.). *What is Table? - Definition from Techopedia*. Retrieved September 27, 2020, from <https://www.techopedia.com/definition/1247/table>

Techopedia. (n.d.). *What is a Foreign Key? - Definition from Techopedia*. Retrieved September 27, 2020, from <https://www.techopedia.com/definition/7272/foreign-key>

Techopedia. (n.d.). *What is a Relationship? - Definition from Techopedia*. Retrieved September 27, 2020, from <https://www.techopedia.com/definition/24438/relationship-databases>

Techopedia. (n.d.). *What is an Attribute in Database Systems? - Definition from Techopedia*.Retrieved September 27, 2020, from <https://www.techopedia.com/definition/1164/attribute-database-systems>

The Editors of Encyclopaedia Britannica. (2020, May 18). *Database*. Encyclopaedia Britannica. Retrieved September 27, 2020, from <https://www.britannica.com/technology/database>

supriya\_saxena. (2020, June 12). *Structural Constraints of Relationships in ER Model*. GeeksforGeeks. Retrieved September 27, 2020, from <https://www.geeksforgeeks.org/structural-constraints-of-relationships-in-er-model/>