Team S4G1 - Rapture

DOCUMENT TITLE

Team Signatures:

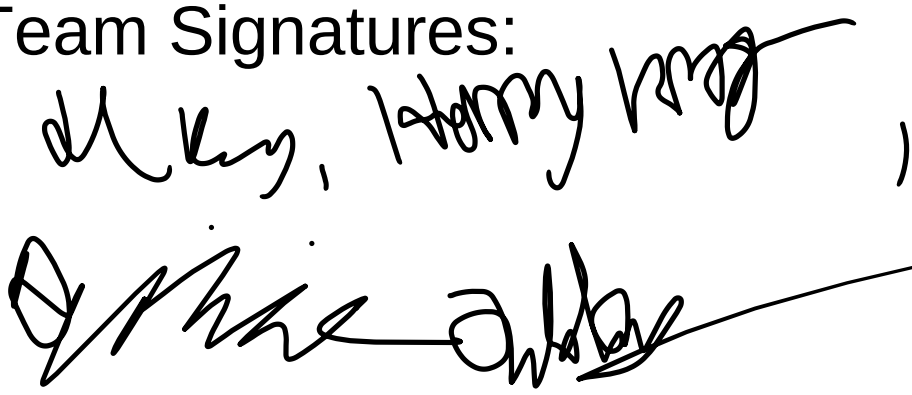

Table of Contents

Executive Summary

This document is to clearly outline the privacy and security parameters we seek to make our database with. This document will leave no vulnerability unaddressed and will clearly make point of all referential and entity integrity constraints.

Introduction

The purpose of this document is to give users of our database a brief and concise understanding of which of their personal data will and won't be securely protected.

Main Content

1. Privacy Analysis

   Within our database we will be storing different personal information for each of our users. This information consists of a user's name, password, ID, and the location they live at. A user has other information pertaining to it such as the locations they experience disasters and their username. These attributes are available to be viewed by other users, and do not need to be protected as severely. General rules for our database are as follows: a user can view the usernames of other people, and a user can also view the location at which a person experienced a disaster at. This location will be a one-time location at the time of the disaster so as not to compromise the other user's current location. Other users will be allowed to view the real name of other users, and each person will be known solely by their respected names. Username is primarily for login purposes.

2. Security Analysis

   The main data that can be breached in our database that is of any substance would be the user information. This includes the information mentioned in the privacy analysis. So, the password of the user, their location, and the person's username. One way that this will be dealt with is through each user only having access to their data and only being allowed to access their own data. Another way to deal with this problem is by protecting

passwords sufficiently. This will be done by salting and hashing each password so that the passwords are encrypted and cannot be accessed or hacked by password tables. The salt used will be sufficiently long as to not have more data risks. The last security risk that will be dealt with is injection attacks which would mess with the information of the users by possibly deleting tables that hold important data. Injection attacks will be dealt with by using prepared statements and callable statements and not just normal statement as well as limiting non string inputs to not accept stings.

3. Entity Integrity Analysis

    1. For the Person table, ID is a unique non null integer that is a foreign key to the HasItem, HasSkill, and Expierienced tables. Username is a unique string.
    2. For the Asset Table, ID is a unique non null integer that is a foreign key to the HasItem, and HasSkill tables, Name is a non null string of some small length and the description is a much larger length.
    3. For the CloseTo table, Location1ID and Location2ID are two different non null integer values that together comprise a primary key.
    4. For the Location table, LocationID is a unique integer integer that is the primary key. Name is a string of max length 20.
    5. For the Disaster Table, ID is a unique non null integer that is the primary key. Name is a string of max length 20, Description is a string of max length 280, and Range is a positive integer.
    6. For the Experienced Table DisasterId, PersonID, and LocationID are all foreign keys that comprise the primary key, taken from the Disaster, Person, and Location tables respectively. DangerLevel is a character, TimeStamp is a datetime.
    7. For the HasItem Table, PersonID is a foreign key that references Person, ItemID is a foreign key that references Asset. Together PersonID and ItemID comprise the primary key.
    8. For the Item Table, ID is a foreign key that acts as the primary key and references the Asset Table.
    9. For the Skill Table, ID is a foreign key that acts as the primary key and references the Asset Table.
    10. For the UsefulFor Table, DisasterID is a foreign key that references Disaster and AssetID is a foreign key that references Asset. Together DisasterID and AssetID make up the primary key.
    11. For the Skill Table, ID is a foreign key that acts as the primary key and references the Asset Table.
    12. For the UsefulFor Table, DisasterID is a foreign key that references Disaster and AssetID is a foreign key that references Asset. Together DisasterID and AssetID make up the primary key.

4. Referential Integrity Analysis

On delete we would usually like to set null because if we are deleting an item or a skill that doesn't mean we want to remove any of the people who have that item or skill. Additionally, we will not worry about deleting or updating location because it will be a static variable that is manually entered by us.

On update we would like to cascade because if the name of an attribute or skill changes then we would like it to become the new name for all foreign keys referencing it. Also, the same goes for updating a person because all referencing foreign keys would need to be updated to the appropriate cascaded value.

5. Business Rule Integrity Analysis

   Because we are not a business and do not have any transactions there will be very few if any business rule constraints. The system is designed solely with information sharing purposes in mind, and not for the profit of anyone.

References

https://www.ifrc.org/our-work/disasters-climate-and-crises/what-disaster

Appendix

DISCLAIMER: Although from a glance it seems that this document and project that it pertains to is of serious matter and importance to the daily life of the Rose-Hulman students and faculty, let me assure you this is not the case. This project is composed of purely fictional data pertaining to an event, and anyone who chooses to place an actual crisis or disaster into our system is doing so of their own volition. As a team we will attempt to make it abundantly clear that although names and locations might be real, any actual event that takes place as described within our project is purely coincidental. As a team we understand the possibility of irreverence that could come from descriptions of deadly and dangerous events, along with how such similarities between a real event and one described in our project could induce anxiety and undue misinformation about the health and wellbeing of all who use the Rose-Hulman campus. For this reason, our team will strive to make the user experience one of lighthearted toying with unknown threats and stay away from serious and relevant real-world events. The UI created will hopefully reinforce the mood that we intend to portray, and we will encourage our users to follow the guidelines that we hold while creating our system.

Index

Glossary

Entity Integrity Constraint – To Identify each row in a table, the table must have a primary key. The primary key is a unique value that identifies each row. This requirement is called the entity integrity constraint.

Referential Integrity Constraint – A referential integrity constraint is defined as part of an association between two entity types. The definition for a referential integrity constraint specifies the following information: The principal end of the constraint. (An entity type whose entity key is referenced by the dependent end.)