Security and Data analysis
CookChain

Ashlee Boyer
Emma Rogge
Lujia Zhang

01/19/2016

Table of Contents

1 Privacy Analysis

2 Security Analysis

The main security concern is that personal information getting leak since each CookChain user has so much personal such as daily grocery list, ingredients list, eating preference and food allergies. All these information is highly sensitive since CookChain account is so closely associated with users' personal life. So the biggest security restrictions for CookChain would be users can only access their own account. Each users account should be highly secured. There will be two level securities to prevent hacking and account leaking. Users' username and password will both be encrypted and linked to each other with a complicated hashing relation. Each account has an account ID. So whenever users added or change data of their account, database will confirm the account ID first then add/change data into the related account. For privacy reason, no one can look at sepecific account data. Even developer and system administrator can only look at the big database table with encrypted usersname and other sensitive personal information.

3 Entity Integrity Analysis

a. User Table: The UserID is the primary key for this table and an nvarchar with the maximum length of 15. Name is a 30 character nvarchar. Password is a 10 character nvarchar. No attributes of this table can be null.

b. Credit Card Table: The Number attribute is the unique primary key of this table which is an integer value. The UserID is a foreign key referenced from the User table. The Code attribute is an integer value for the security code located on the credit card. Type is an nvarchar which holds values such as "MasterCard", "Visa", or "American Express." ExpDate is an nvarchar type which represents the month and year of when the credit card will expire. No attributes in this table can be null.

c. Recipe Table: RecipeID is a unique integer value used to identify every Recipe. The Author attribute is nvarchar with a maximum length of 30 characters. Time is of type time which stores when the recipe was created by the author. The FilePath is an nvarchar used to store the location of the recipe. Only the Author and Time attributes may be null in this table.

d. Ingredient Table: IngredientID is a unique integer value used to identify different Ingredients. The Name is an nvarchar of maximum length 20 characters. Provenance is also a 20 character nvarchar to store the origination of the ingredient. Quantity is an integer value which keeps track of how many of the ingredient a user has in their storage. Only the Provenance attribute may be null in this table.

e. IsAllergicTo, Stores, and Uses Tables: UserID and IngredientID are both foreign keys referenced from the User and Ingredient tables, respectively. IsAllergicTo describes the relationship where a User is allergic to an Ingredient.

Stores describes when a User has add an Ingredient to their storage. Uses describes when a User has used or removed an Ingredient from their storage. No attributes of these tables may be null in any instance.

f. Saves Table: UserID and RecipeID are foreign keys referenced from the User and Recipe tables, respectively. The Saves relationship describes when a User saves a Recipe to their personal collection. No attributes of these tables may be null in any instance.

g. Contains Table: RecipeID and IngredientID are foreign keys referenced from the Recipe and Ingredient tables, respectively. The Contains relationship describes when an Ingredient is contained by some Recipe. No attributes of these tables may be null in any instance.

4 Referential Integrity Analysis

Upon delete, all operations will cascade. For example, if a User is removed from the database, all relationships having to do with a UserID must also be deleted. These relationships include the isAllergicTo, Stores, Uses, Saves and Credit Card Tables. If the relationships are not deleted, there will be null values for UserID which is not allowed.

Updates to tables will cascade and also reject, but only if the desired operation creates null values where they are not allowed. For example, if a UserID was updated in the User table, then all the instanaces of UserID in other tables would be null, which is never allowed in those tables.

5 Business Rule Integrity Analysis

When a User consumes or disposes of any Ingredients, they will need to specify so using their profile. This will provide the most accurate results when searching for Recipes in the database. For the same reason, a User should update their storage details when they purchase or add Ingredients to their storage. A User is also responsible for specifying what Ingredients they are allergic to. Cook Chain does not claim any responsibility to incidents that follow a User selecting a Recipe that may contain an Ingredient they are allergic to if the User has not provided the proper information.

References