Class Diagram:

A diagram with a white square with black text

Description automatically generated with medium confidence

1. Class Diagram Descriptions

Provides descriptions for each class in the diagram and their responsibilities:

User Class

* Description: Represents the user of the application. Each user has a unique email, a master password, and answers to three security questions for authentication and recovery purposes.
* Attributes:
  + email: The user's unique email.
  + master\_password: The hashed password for logging in.
  + security\_q1, security\_q2, security\_q3: Security questions.
  + security\_a1, security\_a2, security\_a3: Answers to the security questions.
* Relationships:
  + Has a one-to-many relationship with VaultItem (a user can have many vault items).
  + Has a one-to-many relationship with Notification (a user can have many notifications).

VaultItem Class

* Description: Represents an item stored in the user's vault, such as a login, credit card, identity, or secure note. Each item can store specific details depending on its type.
* Attributes:
  + General: title, item\_type, secure\_note, masked.
  + Login-specific: username, password, url.
  + Credit Card-specific: credit\_card\_number, cvv, credit\_card\_expiration.
  + Identity-specific: full\_name, address, phone\_number, passport\_number, passport\_expiration, driver\_license, license\_expiration, ssn.
* Methods:
  + notify\_observers: Notifies the user of relevant issues (e.g., expiration dates or weak passwords).
  + check\_credit\_card\_expiration: Checks if a credit card is expired.
  + check\_sensitive\_identity\_fields: Checks for missing or expired identity-related fields.

Notification Class

* Description: Represents a message to the user about events, such as weak passwords or expired items.
* Attributes:
  + message: The notification text.
  + read: Boolean to track if the notification has been viewed.
* Methods:
  + mark\_as\_read: Marks the notification as read.

2. Database Schema and Descriptions

Provides the schema and a short description for each table:

User Table

* Description: Stores user information, including authentication details and security questions.
* Schema:

id INTEGER PRIMARY KEY  
email TEXT NOT NULL UNIQUE  
master\_password TEXT NOT NULL  
security\_q1 TEXT NOT NULL  
security\_a1 TEXT NOT NULL  
security\_q2 TEXT NOT NULL

security\_a2 TEXT NOT NULL  
security\_q3 TEXT NOT NULL  
security\_a3 TEXT NOT NULL

VaultItem Table

* Description: Stores information about vault items, which can be logins, credit cards, identities, or secure notes.
* Schema:

id INTEGER PRIMARY KEY  
user\_id INTEGER FOREIGN KEY REFERENCES user(id)  
item\_type TEXT NOT NULL  
title TEXT NOT NULL  
username TEXT  
password TEXT  
url TEXT  
credit\_card\_number TEXT  
cvv TEXT  
credit\_card\_expiration TEXT  
full\_name TEXT  
address TEXT  
phone\_number TEXT  
passport\_number TEXT  
passport\_expiration TEXT  
driver\_license TEXT  
license\_expiration TEXT  
ssn TEXT  
secure\_note TEXT  
masked BOOLEAN DEFAULT TRUE

Notification Table

* Description: Stores messages notifying users of various events.
* Schema:

id INTEGER PRIMARY KEY  
user\_id INTEGER FOREIGN KEY REFERENCES user(id)  
message TEXT NOT NULL  
read BOOLEAN DEFAULT FALSE

3. User Interface Screenshots and Descriptions

Includes the following screenshots with descriptions:

Login Page

* Description: Allows users to log in using their email and master password. Includes links for registration and password recovery.

Dashboard

* Description: Displays a list of vault items and unread notifications. Provides options to add, edit, delete, or mask/unmask items.

Add Vault Item

* Description: Allows users to add a new item to the vault, selecting its type (login, credit card, identity, or secure note) and entering the corresponding details.

Edit Vault Item

* Description: Allows users to modify an existing vault item, with fields displayed dynamically based on the item's type.

Notifications

* Description: Displays a list of notifications for the user, including options to mark notifications as read.

4. References

* Flask Documentation: https://flask.palletsprojects.com/
* SQLite Documentation: https://www.sqlite.org/docs.html
* SQLAlchemy Documentation: https://docs.sqlalchemy.org/
* Flask-Migrate: <https://flask-migrate.readthedocs.io/>