

## Project Proposal: Personal Data Vault

### Project Description

This project pursues to develop a personal data vault application that is both safe and easy to use. A data vault, like the name implies, holds data related to personal events. This includes the entries of the user's diary and data related that might be useful to be retrieved later for analysis by others. An example of data that can be store is bodyweight values, body height, and glucose levels.

This application seeks to address the demands of those who currently keep a physical journal. It will record the entries from the user's actions while being capable of adding functionalities proper of a database. This helps to keep related data around and in an easy place to use at any time.

### Design Overview

The application will use a relational database to store the journal entries. In addition to that, categories of records can be created and data about those categories can be added at any time. An example to demonstrate how this will work is the following: the user adds an entry about his visit to a doctor. He wants to add a new vault bucket where he will record his weight. He can add a tag that will be a reference to that record, so it can be retrieved later from the entry or directly by entering the vault bucket section.

The weight vault buckner will hold the date and weight measurement. Internally, the record will store the user's reference so only he or a specific group can retrieve the results. The application will be capable of plotting data from the buckets if desired by the user. Similarly, the user should keep the reference to files that are related to his journal entries from each vault bucket.

### Database

To implement this project, a database with at least five tables should be used. These five tables represent the same amount of entities in the system. Table 1 shows a preliminary and simplified relationship between the entities considered. These are as follows:

1. The user entity represents the user of the system. It mainly stores the data required to log in. It records the user id which becomes a foreign key to the other entities.
2. Journals have to be assigned to users. The user ID key is required to create a new journal.
3. Every journal entry belongs to a journal. Hence the journal id is a foreign key required for each entry.

4. The vault bucket entity includes the buckets created in the system. They require a user to be used.
5. Vault bucket entries need to be part of a bucket. A vault ID is used as a foreign key.
6. The bucket-to-journal entity keeps records of any bucket referenced by a journal entry, therefore requiring the identification numbers from each one.

Table 1 - Simplified entity definition and relationships  
Same colors represent foreign-primary key relationships

User Entity

Field name	Type
UserID	Primary Key
Username	Varchar
Password	Varchar
CreatedOn	Timestamp
LastLoggedOn	Timestamp

Journal

Field name	Type
JournalID	Primary Key
UserID	Foreign Key
JournalName	Varchar
CreatedOn	Timestamp
LastModifiedOn	Timestamp

Journal Entry

Field name	Type
JournalEntryID	Primary Key
JournalID	Foreign Key
EntryTitle	Varchar
EntryText	Text
CreatedOn	Timestamp
LastModifiedOn	Timestamp

Vault Buckets

Field name	Type
VaultID	Primary Key
UserID	Foreign Key
VaultName	Varchar
Category	Varchar
CreatedOn	Timestamp
ModifiedOn	Timestamp

Vault Buckets entries

Field name	Type
BucketEntryID	Primary Key
VaultID	Foreign Key
RecordDate	Date
Value	Text
Comment	Text
CreatedOn	Timestamp
ModifiedOn	Timestamp

Bucket to Journal Entry

Fieldname	Type
VaultID	Foreign Key
JournalEntryId	Foreign Key

## Database Retrieval

For this application the following data needs to be retrieved and presented to the user:

1. User entity data to authenticate and allow login to the system. Data will be presented for configuration purposes in the users' account section.
2. The journals created will be retrieved and presented to the user. These journals will be presented using HTML tables. Each row in those tables will present the name of the journal.
3. The journal entries will be presented similarly to the journals' titles. The only difference is that the entries will include an extract of the journal text below the title. This is to differentiate entries having the same name but with different content.
4. The data vaults will be presented to the user using HTML tables. A link from the title of each vault will direct the user to a view where the vault entries are entered and can be edited.

## Database Update

This application will allow adding/updating the following records:

1. From the user data available, only the password and the last logged on a date can be modified. All remaining details would only be removed on account deletion.
2. The journal details' will be modifiable only by the authenticated user that owns the data. To enforce this, the user id is required for each journal.
3. Similarly, the journal entries will only be added/updated by the authenticated user owning the data. A reference from the user and the journal is needed to find a journal entry. These entries will be modifiable only from the users' dashboard page.
4. Both the vault buckets and their entries will be added/updated by the authenticated user who owns the data. The user id is needed to retrieve the vault buckets and the entry data. It serves as a method to control who has access and could compromise the integrity of the data.