

Planning Document: Sprint 2

Team 19 – Sentinel Data Vault

Jiho Choi, Zhaoji Jiang, Adam Petty, Dingfu Sun, Thomas Worns

October 17, 2015

SPRINT OVERVIEW

In this sprint, we will focus on finishing the main components of our project including data entry/user account creation and storage. Our main goals for this sprint are:

1. Finish the database setup and integration with the Java application.
2. Enable users to create and store data entries.
3. Enable users to create an account (or multiple accounts).
4. Securely encrypt all private user account and data entry information.
5. Complete user account functions including account deletion, password reset, and system settings.

Scrum Master: Adam Petty

Scrum meeting time: MWF 12:30pm

Risks/Challenges: SQL database setup is still a huge challenge for us, specifically integrating it with the Java side of the data vault. This is the most important task of this sprint, as it will allow us to perform complete program flow from account creation, entry creation, and entry retrieval. Data encryption/decryption implementation will be another major issue during this sprint, ensuring that the two-way cryptography is secure and efficient.

CURRENT SPRINT DETAIL

User Story (Database):

- As a developer, I want to have a database for storing user and data entry information.

Task Description	Owner	Estimated Time
Choose and set up a Relational Database Management System (RDBMS) to work with for this project.	Team	2
Design the database to store user and data entry information.	Team	5
Integrate the chosen RDBMS with the Sentinel Data Vault Java application.	Team	5
Adjust the abstract DataEntry class and its subclasses that represents data entries and their required information to best fit with the database design.	Team	2
Adjust the User class that represents a data vault user and their account-specific information to best fit with the database design.	Team	1

Acceptance Criteria:

- Must have chosen and setup an RDBMS to use with this project.
- Must have designed a database that will allow for the storage of user and data entry information.
- Must have integrated the chosen RDBMS for use with Java and the data vault.
- Must have implemented SingleFieldEntry, DoubleFieldEntry, and MultiFieldEntry classes to extend the abstract DataEntry class to represent data entries and their required information within the data vault.
- Must have implemented the User class to represent a data vault user and their account-specific information within the data vault.

Test Case: Database correctly stores all user accounts and associated data entries

Instructions:

1. Launch the Sentinel Data Vault program.
2. Assume existing user and stored data entries have been created for that user.
3. Login as an existing user.
4. Create a new data entry for that user.
5. Log out and close program entirely.
6. Relaunch program, and login as the same user.

Expected Results

1. All of the user's pre-existing data entries should appear in the data entry browser along with the newly-created data entry.
2. All pre-existing and newly-created data entry information should also be the same as it was prior to logout and when the user logs back in.

User Stories (Account Creation):

- As a user, I want to create an account for the data vault.
- As a user, I want to set a password for my data vault account.
- As a user, I want to set an account password security question/answer to help remember the forgotten password.
- As a user, I want to create multiple data vault accounts each with its separate data.

Task Description	Owner	Estimated Time
Implement functionality to retrieve user-entered information from the SignupView.	Dingfu Sun	5
Implement functionality to create and store a user object in the database from the account information.	Dingfu Sun	5

Acceptance Criteria:

- When a user clicks "Sign Up" from the login window, they will be given an account creation window that will ask for the required information to create a new data vault account.
- Given a user has entered all required sign up information into the Sign Up window, when the user clicks "Create Account," they will see a success

message if valid information was entered, and they will be taken to the login window to sign in with the newly-created credentials.

- If the user enters invalid sign up information, they will see a failure message, the reason for failure, and be prompted to enter valid information.

Test Case: Valid account creation

Instructions:

1. At the login window, click “Sign Up” button.
2. At the Sign Up window, enter valid account information into all fields.
3. Click “Create Account”.
4. Click “OK” on the success message.
5. Repeat the process for the desired number of created accounts.

Expected Results:

1. The Sign Up window will open, showing account information text entry fields.
2. Text fields will show entered information.
3. A success message will confirm that the account has been successfully created.
4. The Sign Up window will close, and the Login window will prompt to login with new account credentials.
5. Logging in with any of the newly-created accounts will grant access to that account’s data.

Test Case: Invalid account creation

Instructions:

1. At the login window, click “Sign Up” button.
2. At the Sign Up window, enter invalid account information into all fields (in the form of invalid emails, missing fields, mismatched passwords, and/or invalid passwords.
3. Click “Create Account”.
4. Click “OK” on the invalid information error message.

Expected Results:

1. The Sign Up window will open, showing account information text entry fields.
2. Text fields will show entered information.
3. An invalid information error message will appear.
 - a. If an invalid email is entered, the error message will say so.
 - b. If a field had no information entered, the error message will say to enter all the fields.
 - c. If the password fields are mismatched, the error message will say so.
 - d. If the password is invalid (a strong account password meets the following criteria: at least eight characters, at least one uppercase and special

character), the error message will indicate invalid password and the reason.

4. The error message will disappear and the user will be returned to the Sign Up view in order to alter the previously entered invalid information.

User Story (Change Account Identification Details):

- As a user, I want to change my account password.
- As a user, I want to change my account security question and answer.

Task Description	Owner	Estimated Time
Create a “Change Password” button in SettingsView to initiate the password change process.	Thomas Worns	5
Implement PasswordChangeView to display a window for the user to enter their new desired password and security question/answer.	Thomas Worns	5
Implement functionality to change the stored user password within the database.	Adam Petty	3
Implement functionality to change the stored account security question/answer.	Adam Petty	3

Acceptance Criteria:

- When a user clicks on “Change Password” in the Settings window, the Change Password window will open.
- When a user enters their current password, a new password (twice for confirmation), and a new security question/answer into the text fields and clicks “Change Password”, their user account password and security question/answer will be changed to the newly-entered password followed by the appearance of a success message.
- If the entered new password is invalid, then a failure message will appear informing the user why it was invalid.
- If the new password fields mismatch, then a failure message will appear informing the user of the mismatch.

- If the password that the user entered as the current password is not correct, then a failure message will appear requesting that the user enter the correct current account password.
- If there are missing fields, then a failure message will appear informing the user of missing fields.

Test Case: Successfully changing a user account password and security question/answer

Instructions:

1. From the Settings window, click “Change Password.”
2. In the Change Password window, enter the current account password, enter the desired new password (twice for confirmation), select a new security question, and enter a new security answer into their respective text entry fields.
3. Click “Change Password”.
4. Click “OK” on the Password Change success message.

Expected Results:

1. The Change Password window will open, showing text entry fields for current password, new password, confirm new password, and new security answer.
2. Text fields will show entered information.
3. A success message will confirm that the user’s account password has been changed along with a new security question/answer.
4. The success message will disappear and the user will be returned to the Settings window.

Test Case: Unsuccessfully changing a user account password

Instructions:

1. From the Settings window, click “Change Password.”
2. In the Change Password window, enter invalid information into the entry fields (in the form of incorrect current password, mismatching new passwords, invalid new password, and/or missing fields).
3. Click “Change Password”.
4. Click “OK” on the invalid information error message.

Expected Results:

1. The Change Password window will open, showing text entry fields for current password, new password, confirm new password, and new security answer.
2. Text fields will show entered information.
3. An invalid information error message will appear.
 - a. If an incorrect current password is entered, the error message will say so.
 - b. If the new password fields are mismatched, the error message will say so.

- c. If the new password is invalid (a strong account password meets the following criteria: at least eight characters, at least one uppercase and special character), the error message will indicate invalid password and the reason.
 - d. If a field has no information entered, the error message will say to enter all the fields.
4. The error message will disappear and the user will be returned to the Change Password window in order to enter valid information.

User Stories (Deletion of Account & Associated Data Entries):

- As a user, I want to be able to completely delete my account and its associated data.
- As a user, I want my account and data to be securely wiped after a configurable number of failed login attempts, given that this feature is turned on.
- As a user, I want to be able to delete a selected data entry that is stored in the database.

Task Description	Owner	Estimated Time
Implement the functionality to detect and track of the number of failed consecutive login attempts in the LoginView.	Dingfu Sun	2
Implement the functionality to send out an email to notify user when the number of failed consecutive login attempts reaches half the user-configured value.	Dingfu Sun	4
Implement the functionality to automatically delete associated user data when the number of failed consecutive login attempts reaches user-configured value.	Dingfu Sun	5
Make the “Delete account” button functioning which will delete the currently logged-in user account info and all associated data entries from	Dingfu Sun	4

the database then return user to the LoginView menu.		
Make the “Delete entry” button fully functioning in the MainView which will delete the selected data entry from the database.	Zhaoji Jiang	8
Add and implement a toggle option for “Account wipe” in the SettingsView which will enable the user to turn on/off this feature.	Jiho Choi	1

Acceptance Criteria:

- When the number of failed consecutive login attempts reaches half the user-configured value, an email will be sent to the user’s login email address informing them that their account is unsuccessfully trying to be logged into. This email will also notify them of their account wipe settings.
- When the number of failed consecutive login attempts reaches the user-configured value, all associated data for that account will be automatically deleted and an email will be sent to notify the user that his account and data have been wiped due to fraudulent attempted account access.
- When user hits “Delete account” button in MainView, a confirmation window will be displayed to ask the user if he wants to continue. If user chooses “No” then he will be returned to MainView and no further actions or changes are needed. If the user chooses “Yes”, current user account and associated data will be removed from the database *only* after the user confirms their account password. After the deletion is completed, the user will be returned to LoginView.
- When user hits the “Delete Entry” button in MainView, a confirmation window will be displayed to ask the user if he wants to continue. If user chooses “No” then he will be returned to MainView and no further actions or changes are needed. If the user chooses “Yes”, the selected data entry will be removed from the database. After the deletion is completed, the user will be returned to MainView and can no longer see the deleted data entry.

Test Case: Successfully wipe the user account and associated data after a user-configured number of failed consecutive login attempts

Instructions:

1. From the login window of the data vault, enter a valid username, but with the wrong password repeatedly (3 times as default configuration).
2. From the login window of the data vault, enter a valid username, but with the wrong password repeatedly (3 additional times as default configuration, for a total of 6 failed attempts as default).

Expected Results:

1. The attempted username email address will receive an email informing the user that his account has reached half the failed login attempts, and that once the limit is reached, their account will be wiped completely.
2. A warning message will appear with the notification that this account has been wiped from the data vault. Typing in the correct username and password will no longer allow user to login (combination doesn't exist in database anymore and need to recreate the account).

Test Case: Successfully delete a user account and associated data

Instructions:

1. From the main window of the data vault, click "Delete Account" button.
2. Click "Yes" when the confirmation window asks if you wish to continue.
3. Enter the user password to initiate the account deletion and click "Delete Account".
4. Type in the just-deleted account login info into the Login window.

Expected Results:

1. A message window will appear to warn the user that his account and data will be deleted permanently, and asks if he wants to continue.
2. A window will ask the user to enter their account password to confirm the deletion.
3. The account will be deleted, the Main window closed, and the Login window shown.
4. Login failure message is appear because the user login combination doesn't exist in database anymore and requires the creation of a new account.

Test Case: Cancel the deletion of the user account and associated data in MainView

Instructions:

1. From the main window of the data vault, click “Delete Account” button.
2. Click “No” when the confirmation window asks if you wish to continue.

Expected Results:

1. The user will be brought back to the main window and can still see all his data.
2. The user will still be able to login with his current username and password

Test Case: Successfully delete a selected data entry from the data entry browser

Instructions:

1. From the main window of the data vault, select a desired data entry from the data entry browser and click “Delete Entry” button.
2. Click “Yes” when the confirmation window asks if you wish to continue

Expected Results:

1. A message window will appear and warn the user that the data entry will be deleted permanently.
2. The user can no longer see his deleted data entry from the main window and the data entry will be removed from the data vault database.

Test Case: Cancel the deletion of the selected data entry in MainView

Instructions:

1. From the main window of the data vault, select a desired data entry from the data entry browser and click “Delete Entry” button.
2. Click “No” when the confirmation window asks if you wish to continue.

Expected Results:

1. A message window will appear and warn the user that the data entry will be deleted permanently.
2. The user will be brought back to the main window and will still be able to see the data entry in the data entry browser.

User Stories (Data Entry Creation):

- As a user, I want to create a data entry that stores specialized, sensitive information.
- As a user, I want to create data entries of a wide variety of types.

Task Description	Owner	Estimated Time
Implement a New Data Entry GUI list element to appear after clicking the “New Data Entry” button on the main window of the data vault.	Zhaoji Jiang	2
Implement a NewDataEntryView UI class to display the data entry creation window.	Zhaoji Jiang	2
Implement functionality to retrieve user-entered information from the NewDataEntryView.	Zhaoji Jiang	2
Implement functionality to create and store a data entry object in the database from the entered New Data Entry information.	Zhaoji Jiang	3
Implement functionality to encrypt and store user data after data has been entered.	Thomas Worns/Adam Petty	6

Acceptance Criteria:

- When a user clicks on “New Data Entry” from the main window of the data vault, then a list will open with all the possible data entry types that a user can create.
- When a user clicks on the desired data entry type, then the Create New Data Entry window will open with all the required information text entry fields for that specific entry type.
- Given a user has entered all the required information for that specific entry type into the text entry fields, when they click “Create”, then they will be returned to the main window, and the new data entry will be listed among the user’s other entries with the newly-created entry’s data on display in the entry viewer.

- Given a user has not entered all the required information for that specific entry type into the text entry fields, when they click “Create”, then they will be shown an error message informing the user and asking that all fields be entered.
- When a user clicks “Cancel” while in the Create New Data Entry window, then no data entry or information will be saved, and the user will be returned to the main data vault window.
- When newly-created data entries are stored in the database, all sensitive information will be in an encrypted format.

Test Case: Successfully creating a new data entry

Instructions:

1. From the main window of the data vault, click “New Data Entry”.
2. From the list of data entry types, click on a data entry type.
3. In the Create New Data Entry window, enter information into all the required fields.
4. Click “Create”.

Expected Results:

1. A list view opens containing all the possible data entry types that a user can create.
2. The Create New Data Entry window opens, showing all required entry information text fields to be entered.
3. Text fields will show entered information.
4. The Create New Data Entry window closes, the newly-created data entry appears in the list of all user data entries, and the newly-created data entry’s information will display in the entry viewer.

Test Case: Unsuccessfully creating a new data entry

Instructions:

1. From the main window of the data vault, click “New Data Entry”.
2. From the list of data entry types, click on a data entry type.
3. In the Create New Data Entry window, enter information into the text entry fields, leaving one or more fields blank.
4. Click “Create”.
5. Click “OK” on error message.

Expected Results:

1. A list view opens containing all the possible data entry types that a user can create.
2. The Create New Data Entry window opens, showing all required entry information text fields to be entered.

3. Text fields will show entered information.
4. An error message informing the user of missing fields will appear.
5. The error message will disappear and the user will be returned to the Create New Data Entry window in order to complete all entry fields.

User Story (Data Entry Modification):

- As a user, I want to edit information in existing data entries and have the changes be saved.

Task Description	Owner	Estimated Time
Implement an EditDataEntryView UI class to display the data entry editing window.	Zhaoji Jiang	2
Implement functionality to retrieve the new information from the EditDataEntryView.	Zhaoji Jiang	2
Implement functionality to edit and save an existing data entry object in the database.	Zhaoji Jiang	4

Acceptance Criteria:

- When a user clicks on “Edit Data Entry” from the main window of the data vault, then the Edit Data Entry window will open and display all the existing data entry information in their respective text entry fields (able to edited by the user).
- Given a user has entered all the required information for that specific entry type into the text entry fields, when they click “Done”, then they will be returned to the main window, and the modified entry’s information will display in the entry viewer.
- Given a user has not entered all the required information for that specific entry type into the text entry fields, when they click “Done”, then they will be shown an error message informing the user and asking that all fields be entered.
- When a user clicks “Cancel” while in the Edit Data Entry window, then no data entry information changes will be saved, and the user will be returned to the main data vault window.

Test Case: Successfully editing an existing data entry**Instructions:**

1. From the main window of the data vault, click "Edit Data Entry".
2. In the Edit Data Entry window, alter information in one or more of the required fields without deleting a field and leaving it blank.
3. Click "Done".

Expected Results:

1. The Edit Data Entry window opens, showing all the existing data entry information in their respective text entry fields (able to be edited by the user).
2. Text fields will show existing and altered information.
3. The Edit Data Entry window closes, the user will be returned to the main window of the data vault, and the modified entry's information will display in the entry viewer.

Test Case: Unsuccessfully editing an existing data entry**Instructions:**

1. From the main window of the data vault, click "Edit Data Entry".
2. In the Edit Data Entry window, alter information in one or more of the required fields, leaving one or more fields blank.
3. Click "Done".
4. Click "OK" on the error message.

Expected Results:

1. The Edit Data Entry window opens, showing all the existing data entry information in their respective text entry fields (able to be edited by the user).
2. Text fields will show existing and altered information.
3. An error message informing the user of missing fields will appear.
4. The error message will disappear and the user will be returned to the Edit Data Entry window in order to complete all entry fields.

User Story (Accessing & Viewing Stored Data Entries):

- As a user, I want to see all existing Data Entries upon login via a file browser-type interface.

Task Description	Owner	Estimated Time
Implement a DataEntryBrowser UI class to display the data entries in the center panel of MainView frame.	Jiho Choi	3
Implement functionality to retrieve data entries from the database.	Jiho Choi	5

Acceptance Criteria:

- When a user logs into the data vault, the center panel of the main window will display the user's data entries in a file browser-type interface.
- When a user clicks on a desired data entry *type* (represented by a folder containing all data entries of that type), then the folder will expand, and its existing data entries will be listed out beneath it.
- When a user clicks on a desired data entry *type* (represented by a folder containing all data entries of that type) that has already been expanded, then the folder will collapse, and its existing data entries will not be visible in the browser.
- Data entry names that are displayed to the user will be in a human-readable format.

Test Case: Accessing and viewing stored data entries

Instructions:

1. From the Data Entry Browser of the main window of the data vault, click on any one of the data entry type folders.
2. Click again on the same data entry type folder.

Expected Results:

1. Within the Data Entry Browser, the clicked-on data entry type folder will expand and list out beneath it all the user's existing data entries of that specific type.
2. Within the Data Entry Browser, the clicked-on data entry type folder will collapse and hide all the data entries that were previously listed out beneath it.

User Story (Accessing & Viewing a Data Entry's Information):

- As a user, I want to easily and clearly view a data entry's information.
- As a user, I want to copy out data entry fields (logins, passwords, etc.) to the clipboard.

Task Description	Owner	Estimated Time
Implement all required data entry type UI viewers to display data entries of their respective type (e.g. Logins, Credit Cards, PINs, etc.) in the Data Entry Viewer of the main window.	Jiho Choi	10
Implement functionality to decrypt user data entries that has been retrieved from database.	Thomas Worns, Adam Petty	5
Implement functionality to display decrypted data entry information.	Jiho Choi	6

Acceptance Criteria:

- When a user selects a desired data entry from the Data Entry Browser of the main window, the data entry's information will be displayed in the Data Entry Viewer of the main window.
- When a user selects a desired data entry from the Data Entry Browser of the main window, the data entry's information will be displayed in the correct format and layout according to its data entry type.
- When a data entry is displayed in the Data Entry Viewer, its information is in human-readable format.
- When a user highlights and copies a data entry field, it will be copied to the OS clipboard and pasteable.

Test Case: Accessing and viewing a data entry's information

Instructions:

1. From the Data Entry Browser of the main window of the data vault, click on any one of the data entries.
2. Repeat with any number of data entries in the Data Entry Browser.

Expected Results:

1. For each clicked-on data entry, it's information will be displayed in the correct format and layout according to its data entry type in the Data Entry Viewer of the main window in human-readable format.

User Stories (Improve Existing Security Functionality):

- As a user, I want to check the strength of passwords and be notified of a password's security level.
- As a user, I want a sophisticated password generator with length, case, special characters, and character redundancy as parameters.

Task Description	Owner	Estimated Time
Improve the password strength checking algorithm to consider the following criteria: length, case, special characters, character redundancy, and more.	Adam Petty	4
Improve the password generation algorithm to take length, case, special characters, character redundancy as parameters while also incorporating true randomness.	Adam Petty	4
Implement a password strength checker class to allow for this functionality to be used across the data vault system.	Adam Petty	2
Improve the the password validation and SHA to incorporate true randomness and salt/pepper hashing.	Adam Petty	2

Acceptance Criteria:

- When a user checks the strength of a password with the Password Strength Checker in the Security window, they will be given a security rating of weak, adequate, or strong based on its length, case, special characters, character redundancy, and other criteria.
- When a user adjusts their desired password parameters and generates a password with the Password Generator in the Security window, they will be

given a password that is as secure as possible while still adhering to the set parameters.

Test Case: Checking the strength of a password using the Password Strength Checker

Instructions:

- 1) Enter the following password into the Password Strength Checker: **a8N\$xR7!z3Y@Dk9%s** and click “Check”.
- 2) Enter the following password into the Password Strength Checker: **a8N\$xR7!** and click “Check”
- 3) Enter the following password into the Password Strength Checker: **password** and click “Check”
- 4) Enter the following password into the Password Strength Checker: **thispasswordisstrongbecauseofitsabsurdlength** and click “Check”

Expected Results:

- 1) The resulting strength should be “**strong**”
- 2) The resulting strength should be “**adequate**”
- 3) The resulting strength should be “**weak**”
- 4) The resulting strength should be “**strong**”

Test Case: Generating a password that meets user-defined criteria

Instructions:

1. Within the Password Generator, adjust the parameters in any way.
2. Click “Generate”
3. Repeat as many times needed.

Expected Results:

1. All checked or entered parameters have been obeyed in the generated password.

User Story (Sign Out):

- As a user, I want to be able to sign out of my data vault account.

Task Description	Owner	Estimated Time
Make the “Sign Out” button fully functioning in the MainView, which will signout the current user.	Thomas Worns	4

Acceptance Criteria:

- When a user clicks the “logout” button in the main window of the data vault, a confirmation window will be displayed to ask the user if he wants to continue.
- If the user chooses “No”, then he will be returned to the main window and no further actions or changes are needed. If the user chooses “Yes”, the user will be returned to the login window.

Test Case: Successfully sign out of the current data vault user account**Instructions:**

1. From the main window of the data vault, click the “Sign Out” button.
2. Click “Yes” when the confirmation window asks if you wish to continue.
3. Log back into the program and repeat Step 1, but click “No” when the confirmation window asks if you wish to continue.

Expected Results:

1. A confirmation window will be displayed, asking if the user wishes to log out of their account.
2. The user is returned to the login window.
3. The user is returned to the main window of the data vault.

REMAINING BACKLOG

Functional:

1. As a user, I want to be able to sort my stored data by type or name.
2. As a user, I want to be able to hide/show passwords in text fields.
3. As a user, I want to be able to search for an information entry.
4. As a user, I want to be able to backup my account data.
5. As a user, I want to be able to specify a location for backups to be stored.
6. As a user, I want to be able to export all my passwords and logins from the application to an encrypted file for porting to a new machine.
7. As a user, I want to have a shared folder that will be available across multiple accounts.
8. As a user, I want to be able to share a data item securely with another user. (e.g. sharing a WiFi password)
9. As a user, I want my data to be securely deleted after a certain number of failed login attempts (number would be configurable).
10. As a user, I want to have two-factor authentication for account access. *(If time allows)*
11. As a user, I want to have a built-in virtual keyboard. *(If time allows)*
12. As a user, I want to have the ability to attach images to a data item. *(If time allows)*
13. As a user, I want to have an overall security rating based off total password strength. *(If time allows)*
14. As a user, I want to have automatic internet form filling option. *(if time allows)*
15. As a user, I want to have an avatar or picture associated with my account *(if time allows)*.

Non-functional:

1. As a user, I want the application to run smoothly and be highly responsive.
2. As a developer, I do not want to have limits on the number of passwords for one user.
3. As a developer, I want user data and backup files to be very strongly encrypted.
4. As a developer, I want to clear the clipboard after a copy and paste operation to increase security.
5. As a developer, I want the ability to change encryption in the future and upscale security.