|  |  |
| --- | --- |
| **Use Case** | Encrypt |
| **Primary Actor** | Database user |
| **Goal in Context** | To encrypt the content that is being written to the database file, whether via adding an account, creating, editing, or deleting. |
| **Preconditions** | 1. Portable program exists on client 2. User has program launched 3. User has initiated an option that uses Encrypt use case |
| **Trigger** | User has created, edited, or deleted an account or database and needs the contents to be encrypted as it is saved |
| **Scenario** |  |
| 1. User initiates add account, create database, edit account, delete account or save account or adds or edits security questions or properties 2. User successfully carries out adding, creating, editing, or a deleting operation. 3. Account/database details are ready to be written to database file 4. Details are encrypted by the System before being written to the file 5. Details are encrypted using the Blowfish algorithm, and using the master password as the key 6. Encrypted contents are successfully written to the file by the system | |
| **Exceptions** |  |
| 1. User decides to cancel database creation, or account adding, editing, deleting    1. User selects cancel option    2. User is taken back to the main menu    3. Contents are not encrypted | |
| **Priority** | Essential |
| **When Available** | Second increment |
| **Frequency of Use** | Many times a day |
| **Channel to Actor** | Via program interface |
| **Secondary Actors** | System |
| **Channels to Secondary Actors** | Via encryption method |
|  | |
| **Open Issues** | Is handling of the master password within the Account Manager completely secure?  Do all the prompts successfully prevent a blank or non existent password from being used. |