Password and Personal Information Management System (PAPIS)

# Overview

The explosion of e-Commerce, other online activities, computer system, networks, and electronic gadgets has dramatically increased the amount of personal identifying information an individual has to keep track of. For example, a typically adult may have dozens on accounts or identifiers, for everything from garage door keypads to dozens of online banking systems. Because of varying standards and security concerns, a person’s identifiers and passwords cannot be all the same. So, keeping track of an ever increasing number on user accounts and confidential notes has developed into a serious challenge. Even, though there are a number of commercial products and open source software systems that help people manage this account information, few meet all of a typical user’s needs.

This document outline an initial set of requirements for simple a *Password and Personal Information Management (PAPIS)* that addresses some of weakness in existing password management systems. Specifically, PAPIS will

* keep track of access information for all types of electronic systems, not just on-line accounts,
* help users management confidential personal notes (e.g. personal facts like SSN) that are not directly associated with account information, and
* allow a user to share certain accounts or notes with specific users or groups of users.
* help users generate passwords they can remember

# Functional Requirements

1. User Authentication and Access Controls
   1. An authorized PAPIS user should be able to login onto PAPIS using his/her own PAPIS username and password. For discussion propose, a PAPIS user who is logged into the system will be called an *Information Owner*.
   2. An Information Owner should be able to modify his/her own PAPIS profile, which includes a username, password, full name, and preferred email address.
   3. An Information Owner should be able to view and edit information about his/her own accounts
   4. An information Owner should be able to view and edit his/her own personal notes
2. Types of information
   1. The system must be able to manage access-related information for any type of electronic system, such as on-line applications, computer systems, Intranet domains, keypad-controlled rooms, etc.
      1. An *Account* is defined as the necessary information need to identify and access a specific electronic system
      2. Accounts may be one of the following:
         1. A web-based account
         2. A desktop-based application account
         3. Computer or network account
         4. Physical-access types of accounts
      3. All accounts must be able to include the following information:
         1. Account Label, specified by the Information Owner for the purpose of recognizing and organizing the accounts
         2. Username
         3. Account Identifier (optional)
         4. Password
         5. A set of reminder questions and answers
         6. Zero or more additional codes or properties, each of which consists of a code name and code value.
         7. Comments
         8. Created on date
         9. Last modified date
         10. Password expiration date (optional)
      4. An Web-based Account must include the following information:
         1. Website URL
         2. Last accessed on date (optional)
      5. A Desktop-based Account must include the following information:
         1. Application name or program name
      6. A computer or network-domain account must include the following information:
         1. A computer name (or IP address) or a network domain name
      7. A Physical-access Account must include the following information:
         1. Account Label
         2. Location
         3. Access Instruction
   2. The system must be able to manage personal notes, such as social security numbers, medical identifiers, credit card numbers, etc.
      1. The system can treat a personal note as a block of text.
      2. The system be able keep the following information about each note:
         1. Label
         2. The text of the note
         3. Create date
         4. Last modified date
3. Organization and Navigation
   1. The system should allow an information owner to organize accounts and notes into a hierarchy of folders
      1. The system should allow an information owner to create new folders
      2. The system should allow an information owner to move accounts and notes between folders
      3. The system should allow an information owner to delete existing folders
   2. The system should allow an information owner to browse his/her accounts and notes, using the folder hierarchy
   3. The system should allow an information owner to search for accounts or notes by label or any piece of text in the account or note.
4. Sharing information
   1. The system needs to allow an information owner to place other users into groups. For example, a user could put all family members into a “family” group, all co-workers into a “peers” group, or managers into a “bosses” group.
   2. An Information Owner should be able to grants another user or group of users the right to view an account.
   3. An information Owner should be able to grant another user or group of users the right to view a note.
   4. *Trusted Friends* can an information owner viewed selected accounts and notes.
      1. A *Trusted Friend* is a user to which an Information Owner has given view rights, either directly or by virtues of being part of group with view rights.
      2. A trusted friend can browse the accounts and notes for which he/she has view rights
      3. A trusted friend can view the details of any account or note for which he/she has view rights
5. Password Generation
   1. The system should be provide a simple tool for generating passwords that meets user-specified requirements and utilize user-specified building blocks
   2. The password requirements include:
      1. The minimum password length
      2. Whether the password has to include upper case characters
      3. Whether the password has to include numbers
      4. Whether the password has to include special characters and what those special characters are
      5. Characters that cannot exist in the password
      6. Whether the password has to start with a letter
      7. Whether the password can’t be a pervious password
   3. The password generator should allow the user to enter a list of password fragments that the generator can use to compose a more complex password.
6. Confidentiality
   1. The system must save all the data in an encrypted persistent store
7. System Administration
   1. A system administration should be able to add, edit, archive, and delete users

# Functional Requirements

1. The system must run on Windows XP
2. The system must be written in C# using the .NET Framework 2.0 or 2.1
3. The initial version of system must be a desktop application