Password Pattern Container

Problem

Password are literally required in every corner of digital citizen life in order to access various services. Online service providers, who mostly are independent from each other, force their users to create distinguished username and password. However, users are human being, memorizing passwords is cumbersome and non-realistic for human user. In the other hand, same username and password introduces major security risk and raises privacy concern if one of the service providers leaks their data.

Indeed, popular web browsers such as Chrome, Safari, and Firefox help their users memorize and fill in password. Yet, the problem is, they are close-circled. A Safari user has to type every password to every website service if the passwords are stored and managed by Chrome. Some would argue that Firefox which runs in all major computing platform is the pain-killer. But the pain-killer is more personal preferences oriented rather than function oriented. What's more, browser performance plays an vital role in user's browser decision.

To mitigate the security risk and privacy concern, as well as to bridge the gap between various web browser and computing platform, I propose that human user needs only to create a password pattern instead of password, and utilize the Password Pattern Container, which stores the combination of patterns, to retrieve and calculate a specific password they need.

For example, service A requires user to create strong password with following requirements:

- At least 8-character in length
- Include at least one digital number 0-9
- Include at least one uppercase letter A-Z
- Include at least one lowercase letter a-z
- Include at least one special character ~`!@#$%^&*()+=_-{}[]\|:;"'?/<>,.

So, strong password requirement like service A, user can create a password pattern:

- Four digits: 4321
- Two uppercase letter: QWER
- Two lowercase letter: asdf
- Four special character: !@#$
- One special letter for service provider: A

By matching the strong password pattern, we could generate password for service A "4321QWERasdf!@#$A" and store it to the container.

If another service B requires same strong password, we can create "4321QWERasdf!@#$B" and store the pattern requirement. In later occasion if we want to retrieve password for service B, search in the container and instantly remember the password if we know its requirement.

Another example, for debit card pin, the requirement normally is:

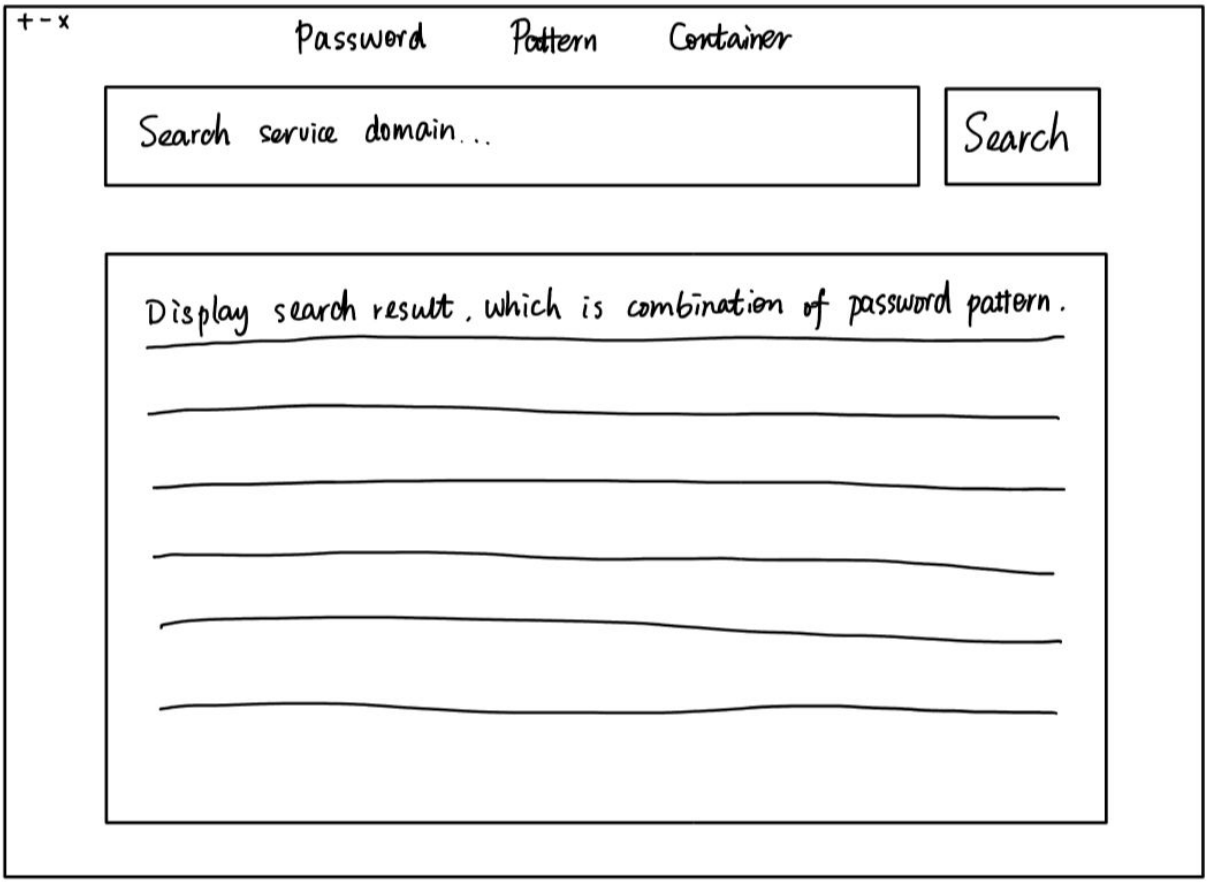
- At length 4
- Include digit number

In this case, we can store the pattern and retrieve the password if we know the password requirement.

Primary Stakeholder

Human users who need support for cross computing platform login to access same service from diverse web browser or applications in difference operating system.

Graphical User Interface



Data

Key = www.website-xxx.com
Integer length
Boolean digitNumber
Integer digitNumberLength
Boolean upperCase
Integer upperCaseLength
Boolean lowerCase
Integer lowerCaseLength

Boolean specialChar
Integer specialCharLength

Requirement reference

Length
Digit 0-9
Uppercase letter A-Z
Lowercase letter a-z
Special character ~`!@#$%^&*()+=_-{}[]\|:;”’?/<>,.