Product Requirements Document

PROJECT TITLE: CorgKey | **TEAM:** Fat Stacks

MEMBERS: Adam Yu (Lead), Arjun Singh, Hunter Massey (Scribe), Olivia Gillam, Simon Yu

BACKGROUND:

Traditional security products (keys, smart cards, passwords, fobs, etc.) are inconvenient for users and are associated with a number of security risks. Without proper security infrastructure, people deal with memorizing passwords and carrying around keys and smart cards wherever they go, increasing the risk of getting locked out of their computers, accounts, servers, doors, and even automobiles. Malicious actors can use these vulnerabilities to execute phishing attacks, stealing users' credentials and gaining access to their information and accounts. The risk posed by these attacks necessitates the use of additional security measures, including two-factor and biometric authentication.

Two-factor authentication is widely used online as another security measure in case one's password becomes compromised. Many companies have their own versions of 2FA in conjunction with either mobile phones (SMS or app) or with a separate email address. 2FA via SMS is vulnerable to SIM swapping attacks, making it less secure when compared to 2FA apps, such as Google Authenticator or Duo. Two-factor authentication still necessitates the use of a password, meaning that if a user forgets their password they must reset their password (usually via email). If given access to a user's email, hackers can exploit this vulnerability and change a user's password. This effectively bypasses 2FA, without relieving the user's burden of password management.

By creating an application to serve as the *primary* factor of authentication, users will not have to worry about managing passwords, nor will they be inconvenienced with 2FA apps or SMS codes. The application effectively replaces a user's password, eliminating the risk of phishing and allowing fast & convenient account access as long as a user carries their phone on their person. Users will also have easy access to

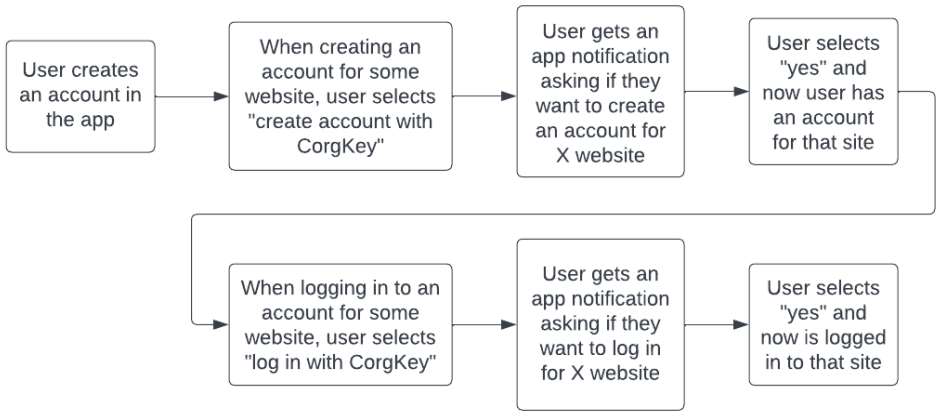
account management, allowing them to retain authorization to their accounts in the event that their phone is lost.

PROJECT GOALS:

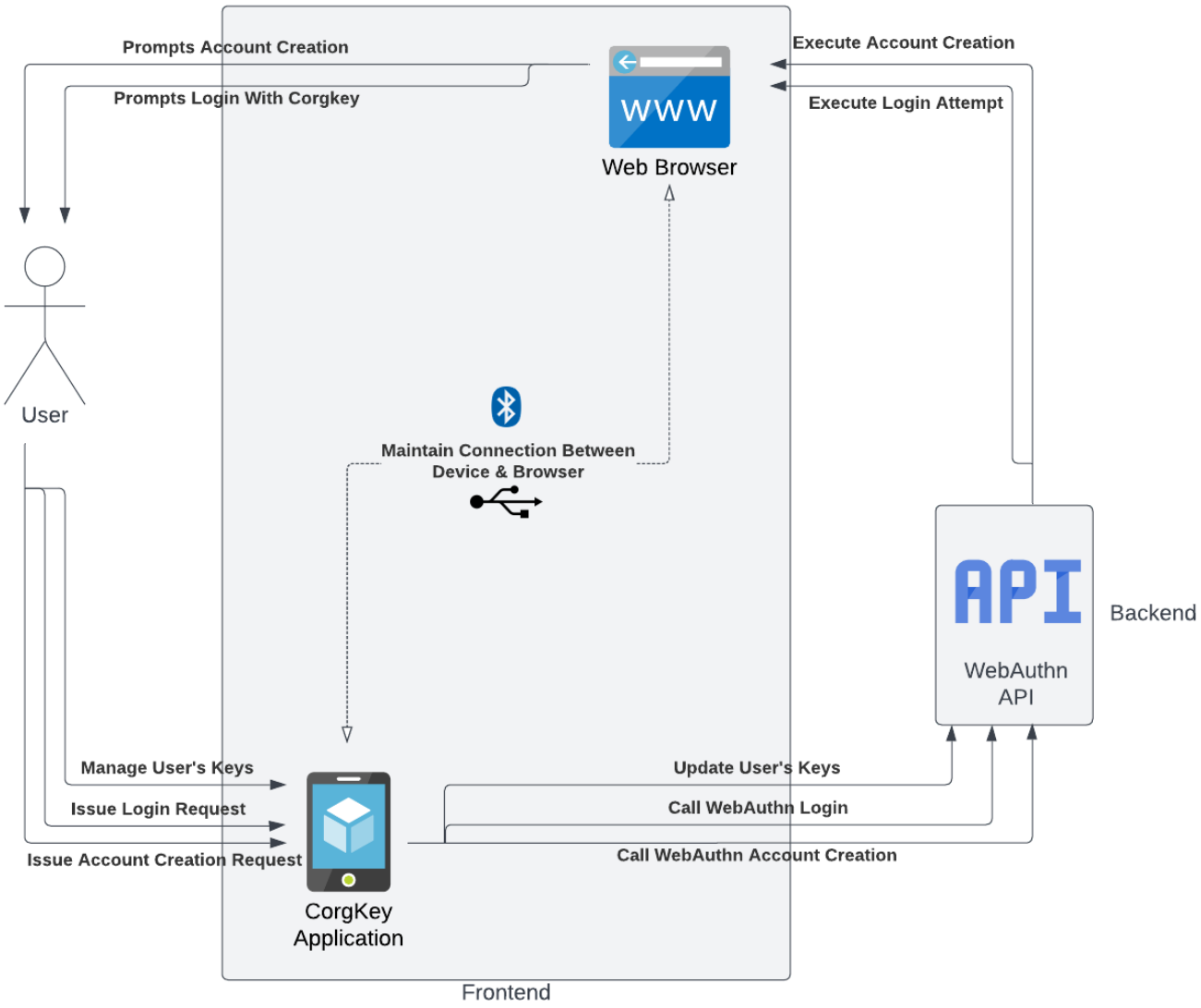
With this project, we will create a mobile application that talks to a browser, using the FIDO protocol to authenticate users. Upon their first visit to a website, users will be able to create an account for a website on the mobile application without inputting their own credentials. Upon future visits, users will be able to log into the website by opening the application, rather than manually entering their username and password.

SYSTEM ARCHITECTURE OVERVIEW:

➔ Basic user interaction & design



➔ High level diagram



FUNCTIONAL REQUIREMENTS

- **User Stories (& Associated Program Stories):**
  - User creates an account / User can log in
    - Users can easily create an account and log in using email and password or some other method (e.g. Google OAuth)
    - Program stores CorgKey account information in a database (?)
    - GitLab Issue: https://gitlab.com/allthenticate/integrations/fido/app/-/issues/12
  - User can create a new account on some site
    - Program should receive a request from the device that is trying to create an account (via bluetooth, wifi, cable???)
    - Program should send prompt to user asking for confirmation that the user wants to sign in using Corg-Key
    - Program should create new account associated with CorgKey identity and store necessary information in a database
    - GitLab Issue: https://gitlab.com/allthenticate/integrations/fido/app/-/issues/13
  - User logs into account on some site
    - Websites should show option to log in with CorgKey, and then our app should process the request (if phone is nearby / if phone is connected to laptop)
    - Program should receive request from device that is trying to log in to account (via bluetooth, wifi, cable???)
    - Program should send prompt to user asking for confirmation
    - Program should send request back to site, allowing log in
    - GitLab Issue: https://gitlab.com/allthenticate/integrations/fido/app/-/issues/14
  - User can view all sites that their authentication is connected to
    - Program should have a scrollable alphabetical view of all site names & link to the login page

        - GitLab Issue: https://gitlab.com/allthenticate/integrations/fido/app/-/issues/15
    - User can delete an account associated with a website
        - Program should ask for confirmation
        - Program should send request to website asking for account deletion
        - GitLab Issue: https://gitlab.com/allthenticate/integrations/fido/app/-/issues/16
    - User can disable their account if their phone is lost
        - Program should keep list of trusted external device(s); user can choose how many are required to disable, and can add/remove with verification
        - Program should ask series of previously set up personal questions
        - GitLab Issue: https://gitlab.com/allthenticate/integrations/fido/app/-/issues/17

NON-FUNCTIONAL REQUIREMENTS:

- The app should be intuitive to use
- The user should not be inconvenienced with frequently opening the app for authentication (e.g. Having the app run in the background)
- The authentication should be secure and fast
- There should be reliable methods for account recovery in the case of lost phone
- The application should be scalable to allow for many accounts

APPENDICES:

- Technologies employed
    - GitLab
    - Flutter (& Dart)

- Notion
- Figma
- VS Code
- Lucid Chart Creator
- WebAuthn API
- USB