

Cryptify

Crypto + Simplify

Sprint 1 Presentation
Team 20

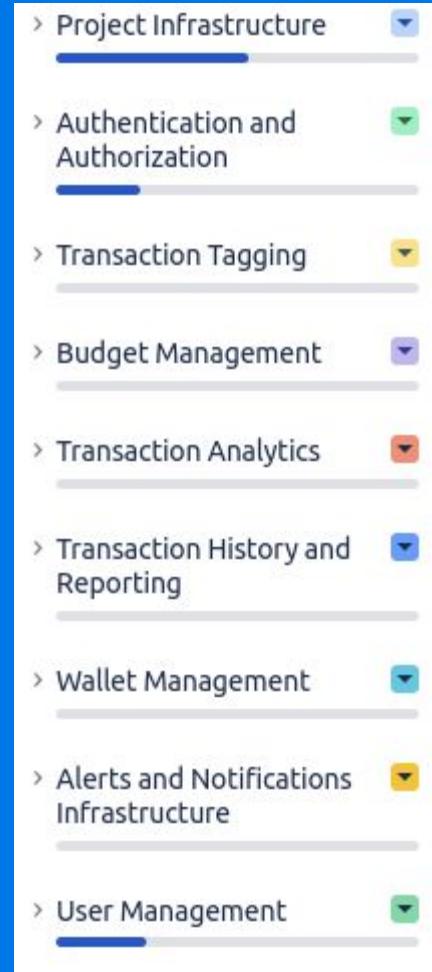
What is Cryptify?

- In a world where cryptocurrencies are used more and more for everyday transactions, people need a platform that brings the modern conveniences of fiat currencies to crypto
 - Aggregation of wallets / accounts
 - Viewing, filtering, and sorting of transactions across wallets, currencies, etc
 - Transaction tagging to simplify remember who you sent and received money from as opposed to a long encoded string
 - Budget management for those cryptocurrency wallets
 - Spending and budget history / reporting for personal and tax reasons

Backlog Overview

Current Epics

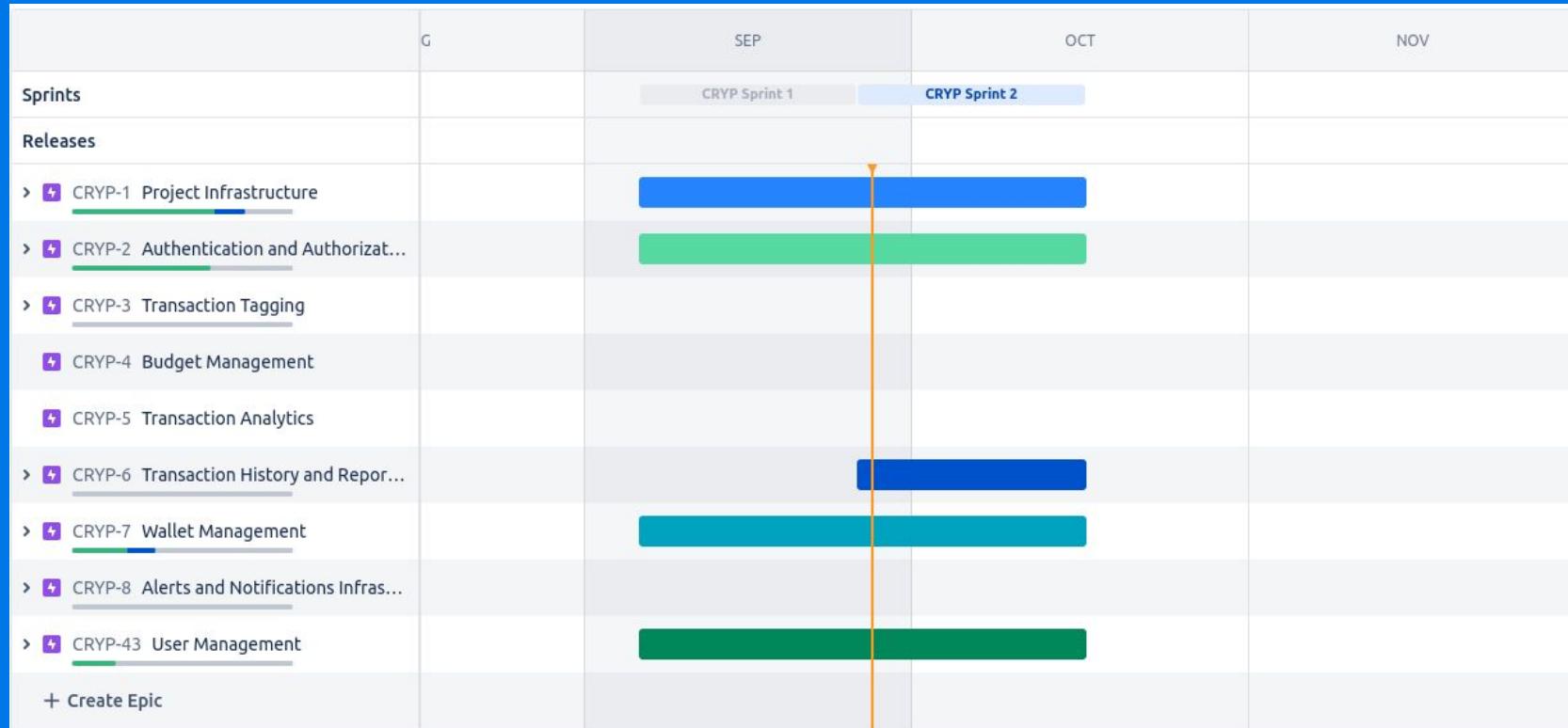
- Project Infrastructure
- Authentication and Authorization
- Transaction Tagging
- Budget Management
- Transaction Analytics
- Transaction History and Reporting
- Wallet Management
- Alerts and Notifications Infrastructure
- User Management



Backlog To Dos

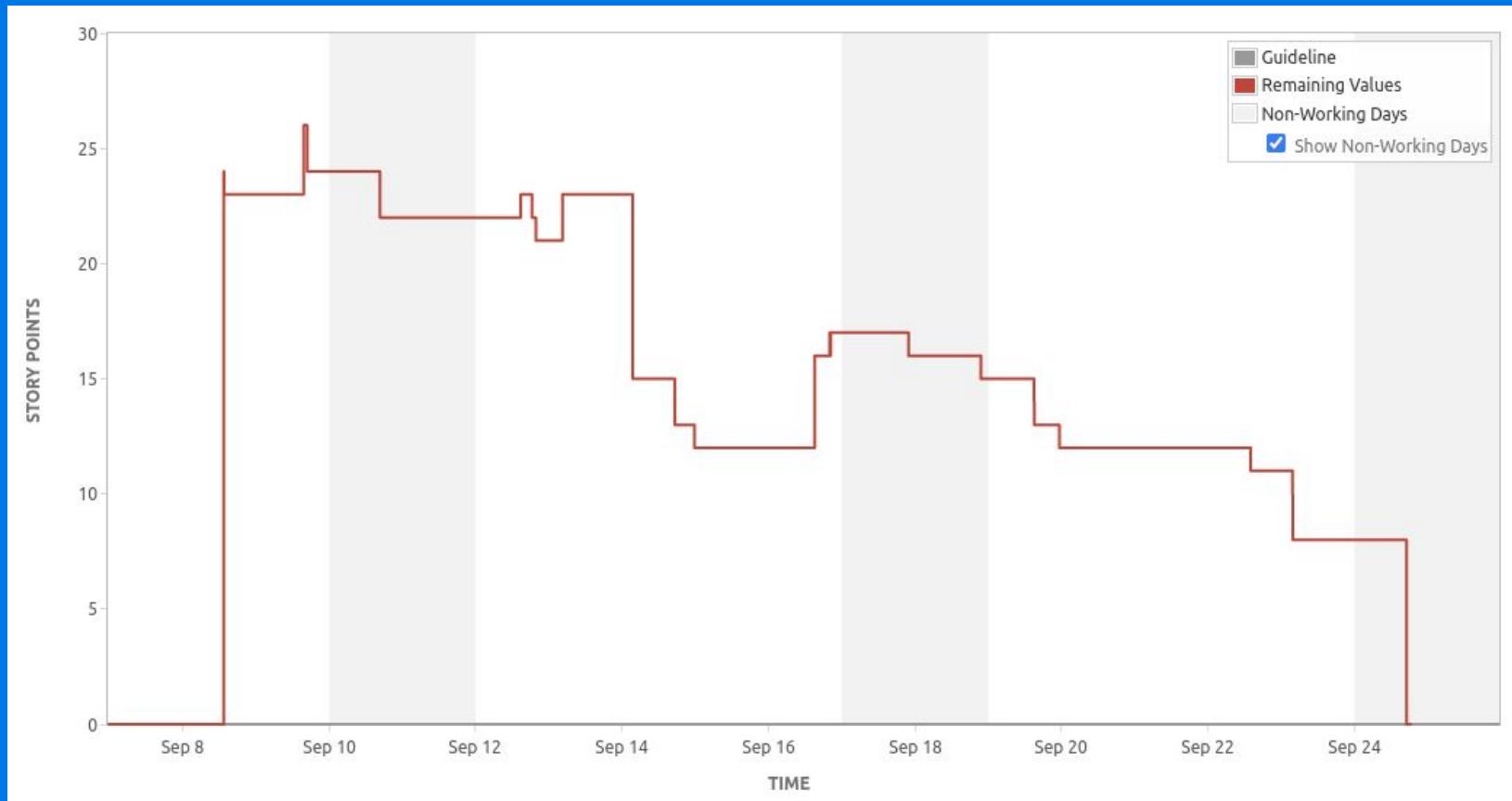
- Backlog is missing user stories for these epics
 - Budget Management
 - Transaction Analytics
- These will be added later on since they are both epics scheduled for release 3

Roadmap Progress



Sprint Retrospective and Burndown

Sprint Burndown Chart



Sprint Retrospective

Went Well	To Improve	Action Items
Rearranging tasks to make sure no one is overloaded -- Project management -- Well organized tasks on Jira	We should have a clear deadline on when the code should be ready. -- Having clear internal deadlines for each task	Create a code freeze deadline 3 days before the end of the sprint, all pull requests must be merged by EOD on the code freeze day
Infrastructure structure -- Tech stack choice -- Good work on setting up the basic project and architecture, this will allow us to work much quicker in the next sprints -- Project setup was done quickly!	Improving knowledge of our tech stack. e.g implement more components on frontend -- Understanding nativebase component library	@Dom to give quick presentation on internal components and figma specifications
Stakeholder communication	Handoff between UI mockups and front end dev -- Respecting the UI prototypes -- Focus on attention to detail for the frontend, make sure they are covering all the detailed parts of the frontend (error messages, colors, spacing, etc)	Make sure everyone on the team is able to run the app with the backend and database on either an emulator or a physical device
	Communication between frontend and backend	

Plan For Next Sprint

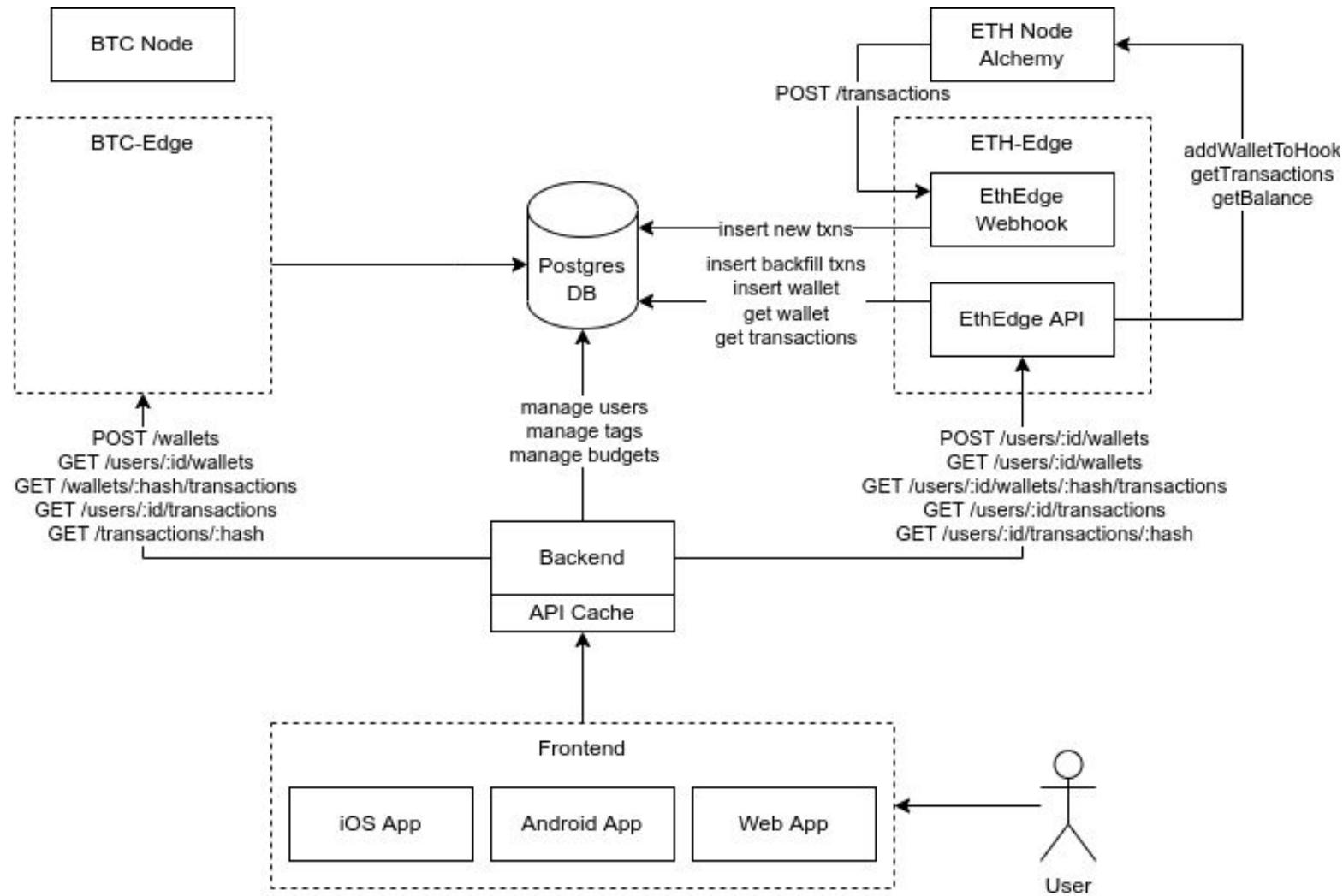
Goals of next sprint

- Build and integrate CI pipeline
- Setup EthEdge package
- Account settings page + sign out
- Basic app / site infra (navigation tab bars)
- Add and view Bitcoin and Ethereum wallets in account
- Implement ethereum transaction management
 - Backfill previous transactions
 - Store all new transactions

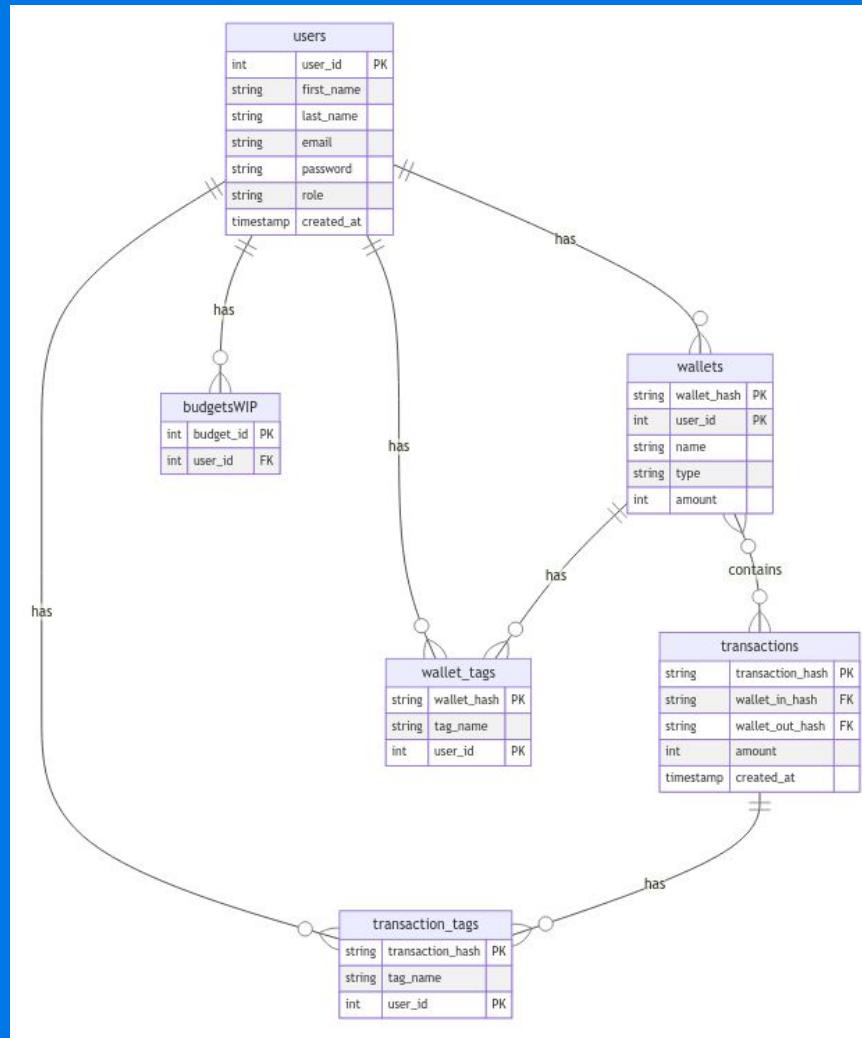
System Architecture and Diagrams

High Level Architecture

- Cross platform frontend client written in React Native, builds for
 - iOS
 - Android
 - Web
- Backend API / frontend gateway
- Edge services for each of our crypto blockchain integrations
 - EthEdge
 - BtcEdge
- PostgreSQL DB handling all persistent storage
- Everything is running in Docker
- Deployed on AWS



ERD



Personas

The Beginner



Alex Roy

AGE 18
EDUCATION High School
STATUS Single
OCCUPATION Sandwich Artist
LOCATION New York City
CRYPTO LITERATE Beginner

"I wish I learned about crypto in school."

Personality
Adventurous Risk-taker
Studious

Bio
He currently lives in New York City. He is about to finish high school and is currently planning for university. He bought a small amount of Bitcoin in 2020 after being introduced to it by friends. He considers his crypto literacy as beginner because he was only recently introduced to the topic.

Core needs
• A way to specify the owner of a crypto wallet address
• A simple way to see and understand crypto transactions
• An application that breaks down crypto verbiage to something more familiar

Frustrations
• Difficult to remember someone's crypto wallet address
• When looking at transaction history on the blockchain, it is nearly impossible to know the owner of a crypto wallet address that transacted with his wallet address
• The language used in crypto to describe similar banking terms is somewhat verbose

Brands
    

Payment medium


Platform
 

The Intermediate



Alexandra Shmidtt

AGE 28
EDUCATION MBA
STATUS In a relationship
OCCUPATION Sales Manager
LOCATION Brazil
CRYPTO LITERATE Intermediate

Bio
She lives in Sao Paulo, Brazil with her boyfriend. She spends most of her time reading books and exploring new topics. She got interested in crypto a few years ago and is well versed in it from an economic perspective after reading many books. She considers her crypto literacy as intermediate because she is very comfortable discussing various crypto topics and dabbles with various cryptocurrencies.

Core needs

- A way to categorize crypto transactions
- An app that provides basic crypto asset management capabilities
- Wants to be notified when a crypto transaction occurs with one of her wallets

Frustrations

- Being unable to categorize crypto transactions makes asset management difficult
- Most crypto apps focus on buying and selling crypto
- Has to look at the blockchain to see if her crypto wallets have any new transactions

Personality

Adaptable Thinker
Empathy

Brands

Scholastic Penguin Pearson

Payment medium

Cash/Cheque Digital Payment

Platform

Website Mobile App

The Expert

Samira Saeed



AGE 24
EDUCATION Software Engineering, BE
STATUS Single
OCCUPATION UI/UX Designer
LOCATION Egypt
CRYPTO LITERATE Expert

Not your keys, not your coins.

Personality
Passionate Curious
Creative

Bio
She works as a UI/UX freelancer. She has been a creative person since a young age. She was introduced to crypto at 16 years old and has been enamoured with it ever since. She owns many cryptocurrencies. She is extremely knowledgeable in crypto and is the person all her friends and family go to with crypto-related questions as she is the most knowledgeable. She has even convinced many of them to start buying Bitcoin over the years. She despises not having access to her wallet's private keys and refuses to use apps that restrict access to such keys. She considers her crypto literacy as an expert.

Core needs

- A crypto related app that focuses on privacy and security
- An app that centralizes crypto holdings located across different platforms and products
- An app that provides advanced crypto asset management capabilities

Frustrations

- All crypto apps that allow you to create/have a wallet do not give you access to the wallets private key
- Crypto apps either poorly integrate or refuse to integrate third-party crypto wallets
- Financial management apps consider crypto as another type of investment asset class and therefore do not provide asset management capabilities tailored to crypto

Brands

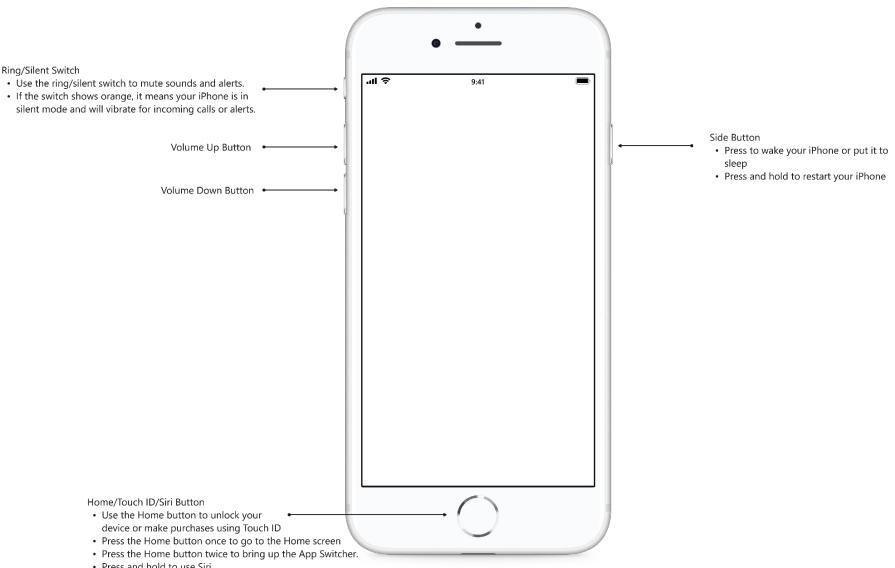


Payment medium
 Cash/Cheque  Digital Payment

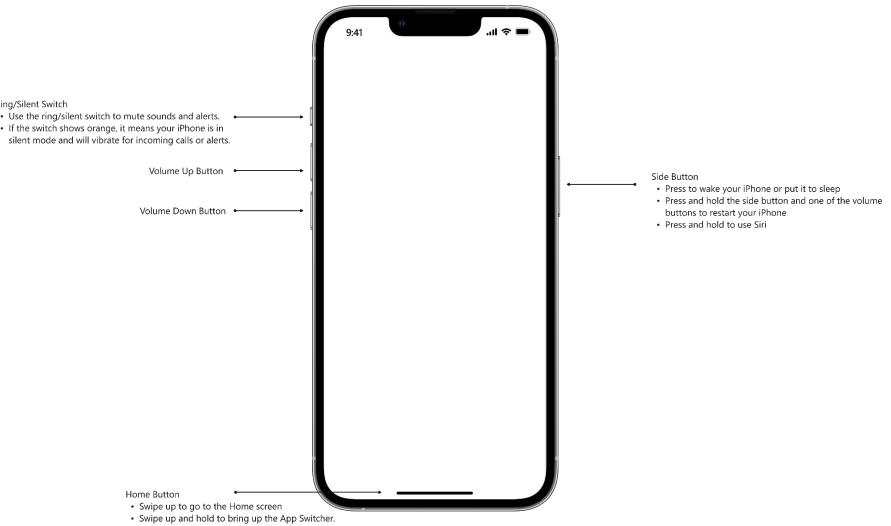
Platform
 Website  Mobile App

Supported Devices

Apple iPhone 8



Apple iPhone 13 / 13 Pro



Google Pixel 6 Pro



Style Guide

Typography

Aa

Segoe UI
Regular

Aa

Segoe UI
Semibold

Aa

Segoe UI
Bold

Title

Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Sed ac ipsum in ante consequat
aliquam accumsan ac lectus.

Colors



white
#FFFFFF



text.200
#E5E5E5 text.300
#D4D4D4 text.400
#A3A3A3 text.700
#404040



darkBlue.500
#0077E6



error.50
#FEF2F2 error.500
#EF4444 error.600
#DC2626



Rose/rose.100
#FFE4E6



success.50
#22C55E success.500
#22C55E

UI Mockups

Sprint 1

Sign Up

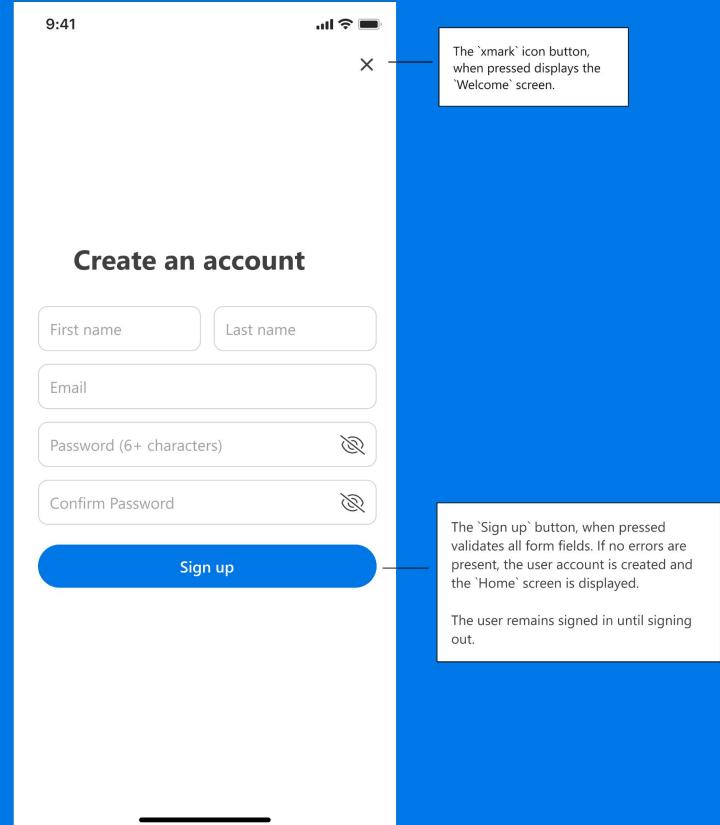
User Stories:

- CRYP-21: As a User, I want to sign up for an account, so that I can access the application's features

Personas:

- The Beginner
- The Intermediate
- The Expert

"The sign up page is up to standard."
- Stakeholder



Sign Up cont.

The image displays two screenshots of a mobile application's sign-up screen, illustrating different states of form fields and associated error messages.

Screenshot 1 (Left): Shows the initial state of the sign-up form. Fields include First Name (John), Last Name (Doe), Email (johndoe@hotmail.com), and Password (Password1). The password field contains black dots. A note indicates that the 'eye' icon button hides the text. Below the form is a blue 'Sign up' button.

Screenshot 2 (Right): Shows the form after entering the same information. The password field now shows the text 'Password1'. A note indicates that the 'eye-slash' icon button removes the black dots. The 'Sign up' button is now blue and appears to be in a pressed state.

Annotations:

- 'First name' error messages:**
 - "Enter a first name."
 - "First name must only contain alphabetic characters."
 - "First name must be 32 characters or less."
- 'Email' error messages:**
 - "Enter a valid email!"
 - "This email address is already associated with another account."
 - "Email must be 50 characters or less."
- 'Password' error messages:**
 - "Enter a password."
 - "Password must be between 6 and 20 characters"
 - "Password must contain at least 1 upper case, 1 lower case letter, 1 number, and 1 special character."
- 'Confirm Password' error messages:**
 - "Confirm your password."
 - "The passwords you entered do not match."
- 'Last name' error messages:**
 - "Enter a last name."
 - "Last name must only contain alphabetic characters."
 - "Last name must be 32 characters or less."

Sign In

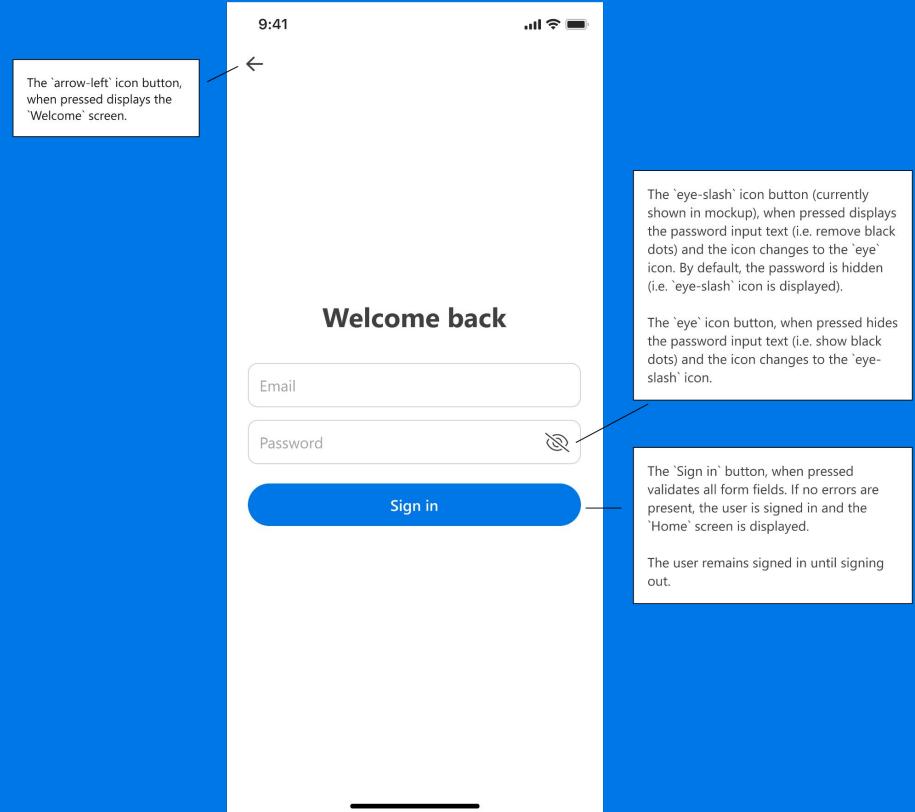
User Stories:

- CRYP-22: As a User, I want to be able to sign in, so that I can access my account

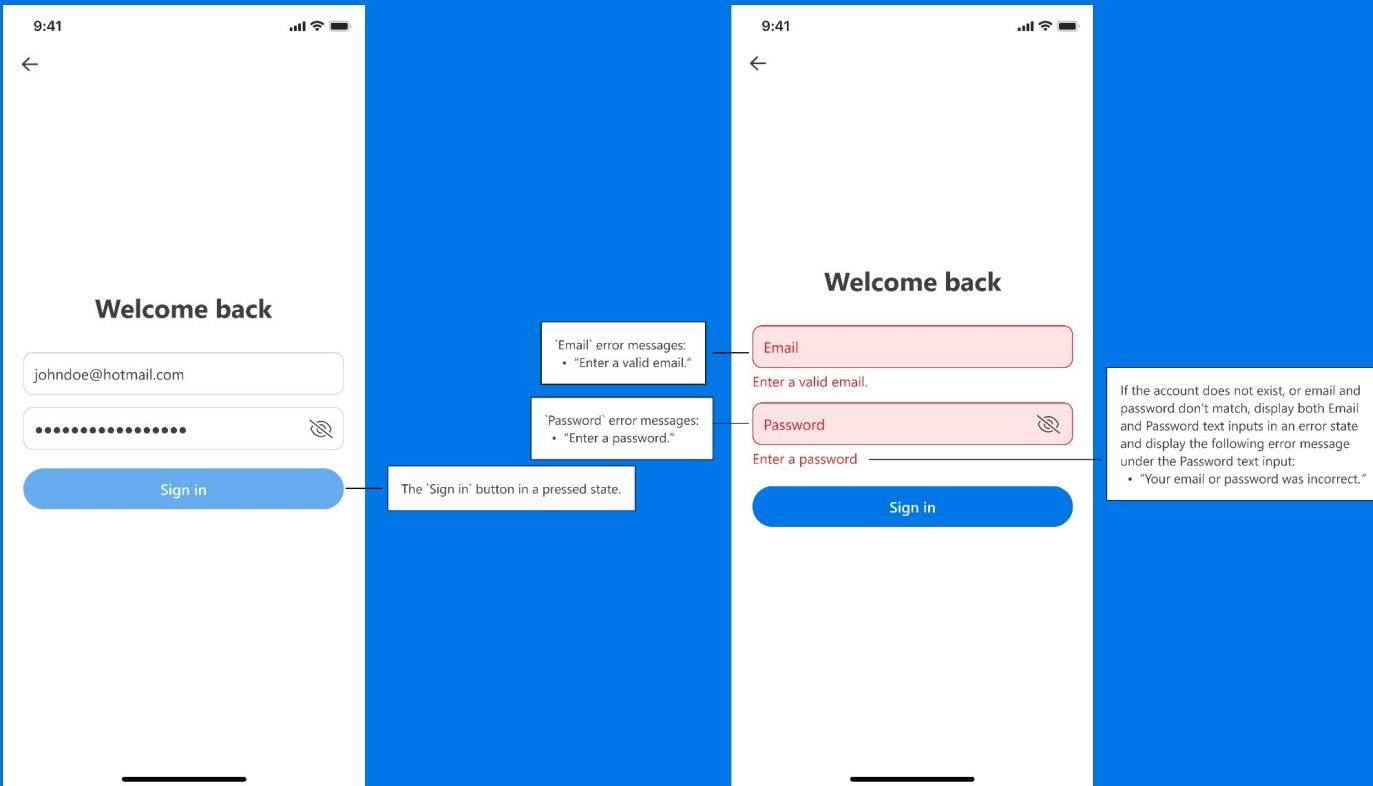
Personas:

- The Beginner
- The Intermediate
- The Expert

"The sign in page is up to standard."
- Stakeholder



Sign In cont.



UI Mockups

Sprint 2

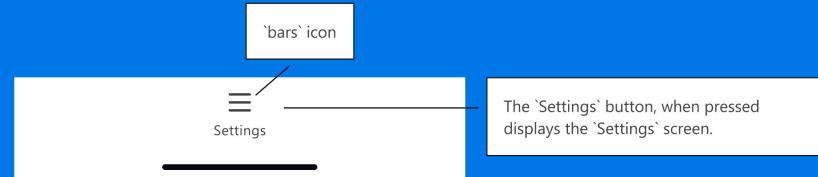
Navigation Tab Bars

User Stories:

- CRYP-94: As a User, I want a navigation tab bars, so that I can navigate around the app
- CRYP-95: As a User, I want an account settings page, so I make changes to my account or personalize my app experience

Personas:

- The Beginner
- The Intermediate
- The Expert



“Looks good, coding can start.”
- Product Owner

Settings

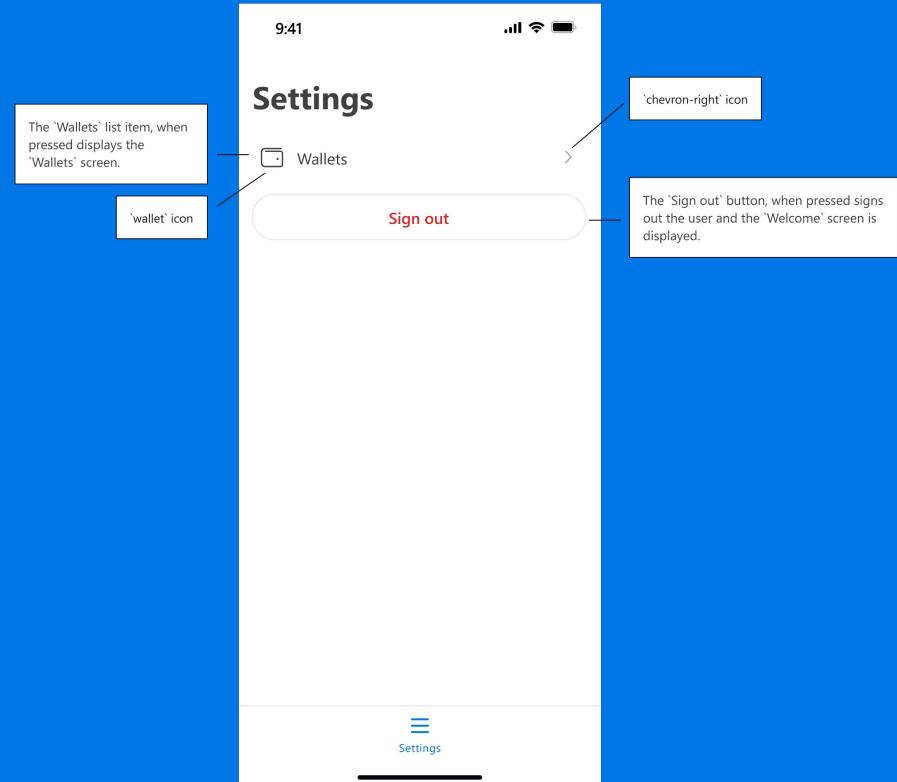
User Stories:

- CRYP-23: As a User, I want to sign out of my account, so that I can delete my session
- CRYP-27: As a User, I want to view a list of all my Bitcoin and Ethereum wallets, so I can see their details in a single place
- CRYP-95: As a User, I want an account settings page, so I make changes to my account or personalize my app experience

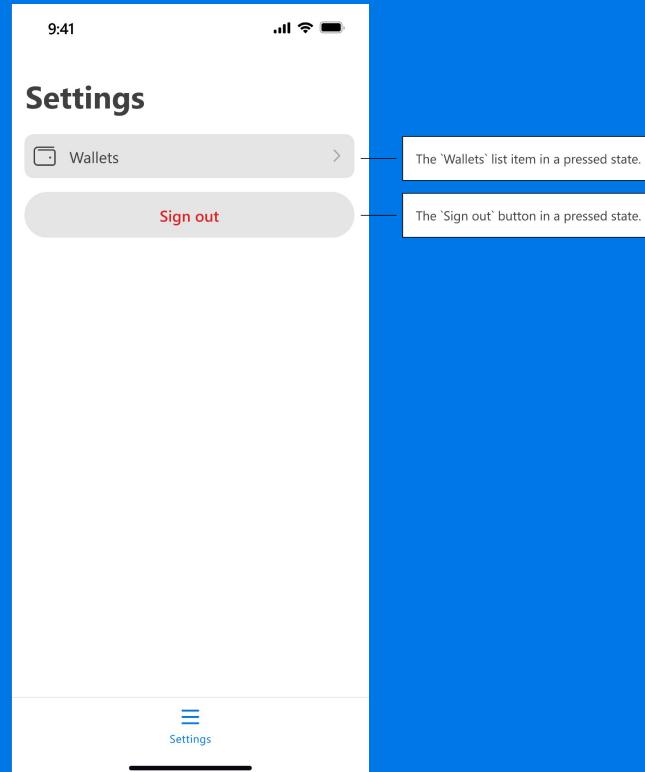
Personas:

- The Beginner
- The Intermediate
- The Expert

"Looks good, coding can start."
- Product Owner



Settings cont.



Settings / Wallets

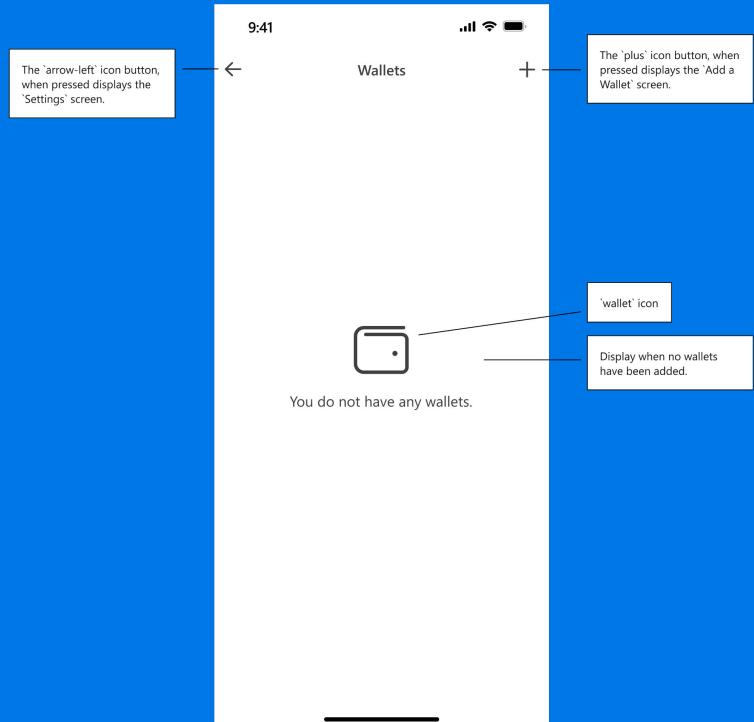
User Stories:

- CRYP-27: As a User, I want to view a list of all my Bitcoin and Ethereum wallets, so I can see their details in a single place

Personas:

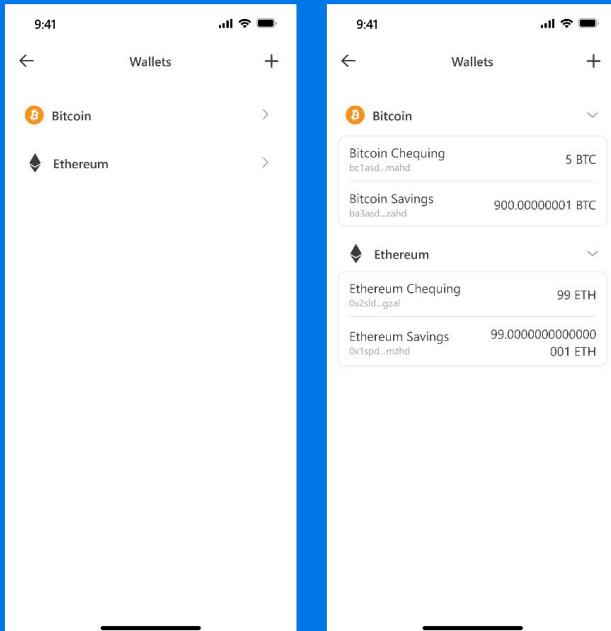
- The Beginner
- The Intermediate
- The Expert

"Very detailed, I like it. Approved for development."
- Product Owner

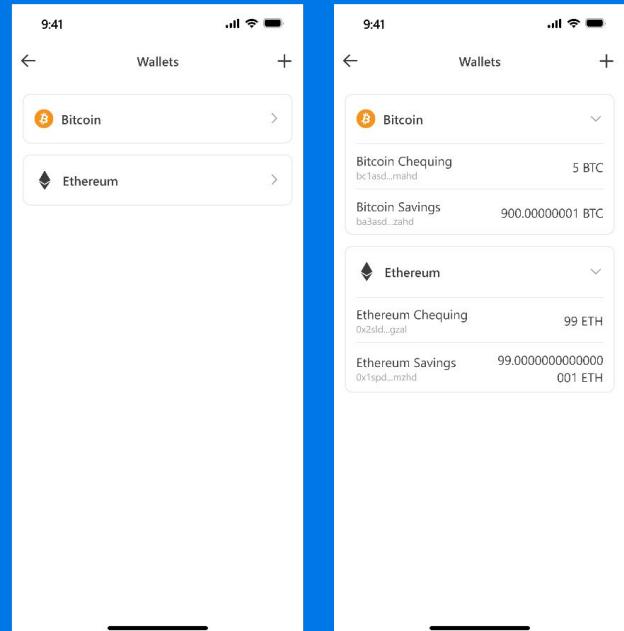


Settings / Wallets cont.

Option 1



Option 2



VS.

Product owner chose option 2.

Settings / Wallets cont.

The 'arrow-left' icon button, when pressed displays the 'Settings' screen.

Display only the wallet accordion that has at least 1 wallet added (i.e. do not display Ethereum if only a Bitcoin wallet has been added).

When the 'Wallets' screen is initially displayed, accordions are collapsed.

The 'plus' icon button, when pressed displays the 'Add a Wallet' screen.

A wallet accordion, when pressed expands and displays all associated wallets. The associated 'chevron-right' icon is also replaced with a 'chevron-down' icon.

Order the wallets in alphabetical order by wallet name.

Wallet address format:

- <first 6 digits><ellipsis><last 4 digits>

9:41

Wallets

+

Bitcoin

Ethereum

9:41

Wallets

+

Bitcoin

Bitcoin Chequing
bc1asd...maha

5 BTC

Bitcoin Savings
ba3asd...zahd

900.00000001 BTC

Ethereum

Ethereum Chequing
0x2sld...gzal

99 ETH

Ethereum Savings
0x1spd...mzhd

001 ETH

Wallet name

Wallet balance (latest)

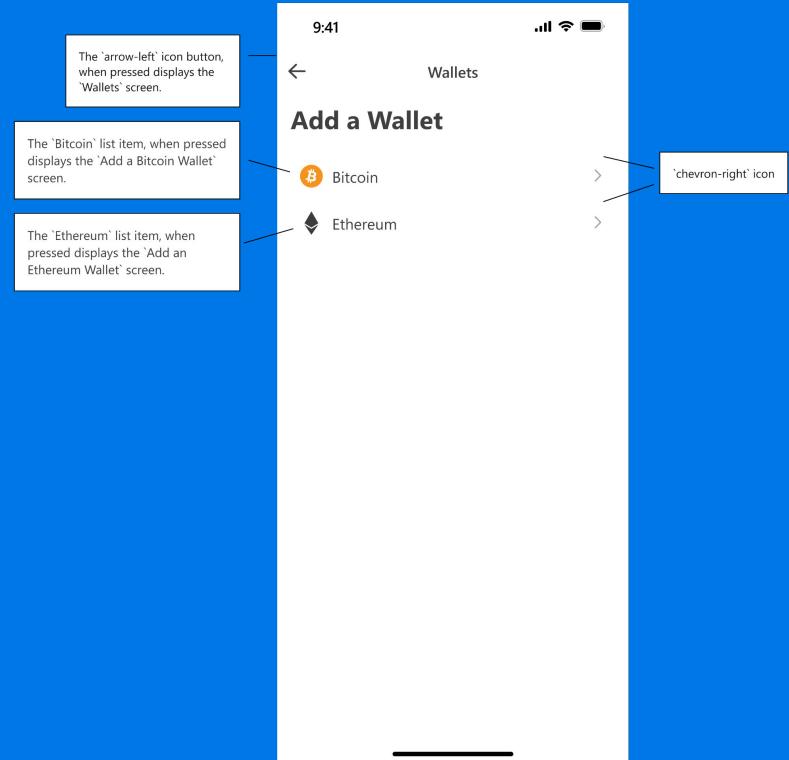
Settings / Wallets / Add a Wallet

User Stories:

- CRYP-24: As a User, I want to add a Bitcoin or Ethereum wallet to my account, so I can keep track of them in a single place

Personas:

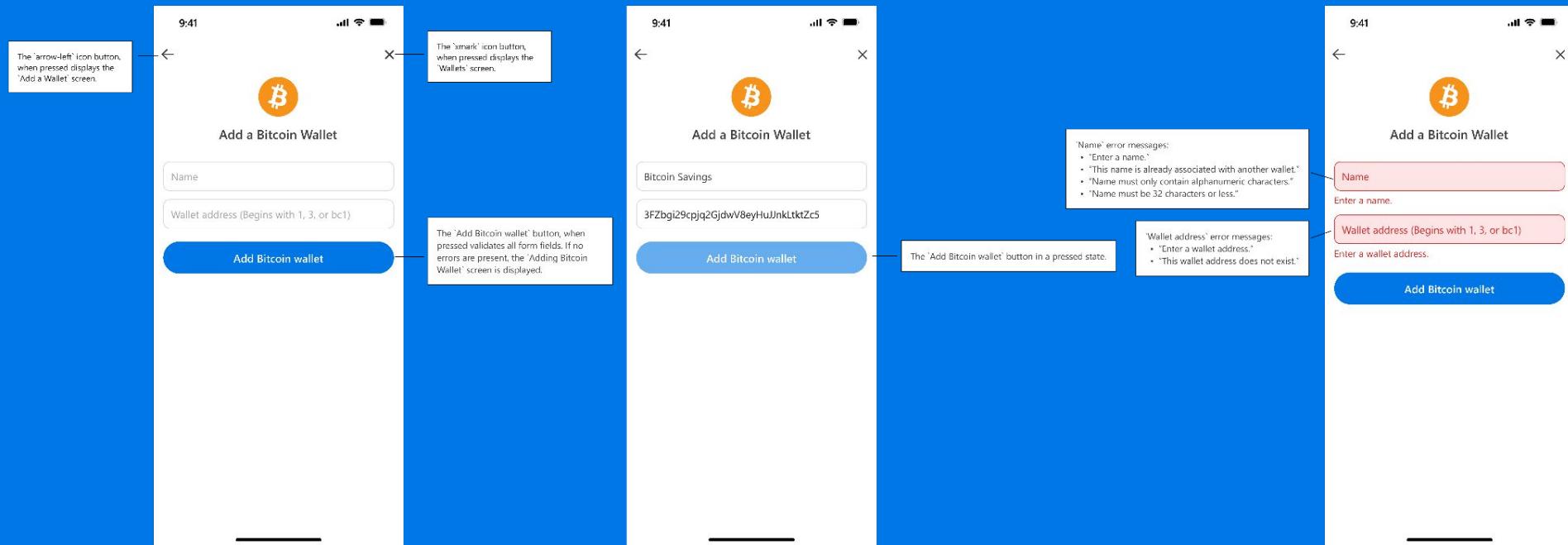
- The Beginner
- The Intermediate
- The Expert



"Approved for development."

- Product Owner

Settings / Wallets / Add a Wallet cont.



Settings / Wallets / Add a Wallet cont.

The image consists of three screenshots of a mobile application interface, likely for iOS, illustrating the process of adding a Bitcoin wallet. The screenshots are arranged horizontally and show different stages of the operation.

Screenshot 1: Initial State

The screen shows a circular spinner icon at the top left and the text "Adding Bitcoin Wallet" below it. A callout box provides instructions: "The spinner represents the current progress for getting all information from the blockchain for the associated Bitcoin wallet (i.e. transactions, wallet details, etc.). If no errors occur during this process then display the 'Added Bitcoin Wallet' screen upon completion. If an error occurs during this process then display the 'Could Not Add Bitcoin Wallet' screen. Even if all information is obtained extremely quickly (i.e. in ms), rotate the spinner for at least 2 seconds. This artificial time slowing is required to provide users feedback on backend processes."

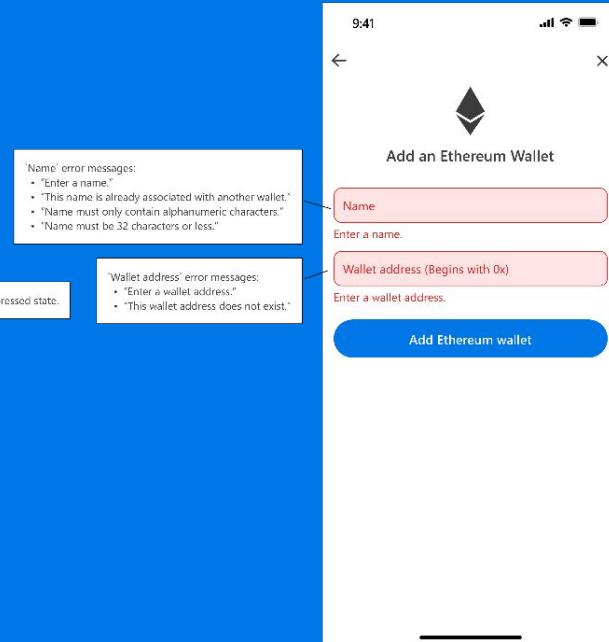
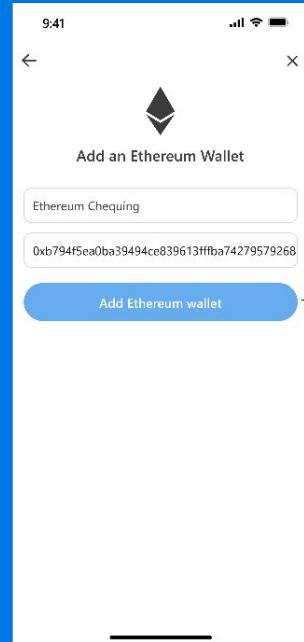
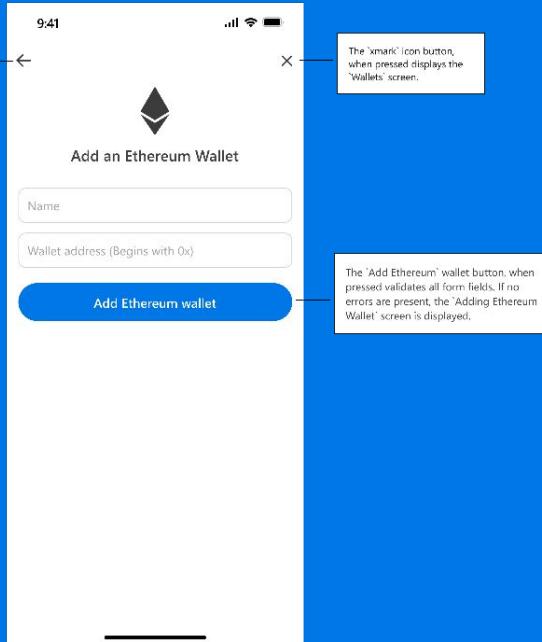
Screenshot 2: Success State

The screen displays a green checkmark icon and the text "Added Bitcoin Wallet" followed by the name "Bitcoin Savings". Below this, there is a "Back to wallets" link and a "Try again" button. Callouts explain the "circle-check" icon and the "Back to wallets" link. A note indicates that the "Add another Bitcoin wallet" button, when pressed, displays the "Add a Bitcoin Wallet" screen.

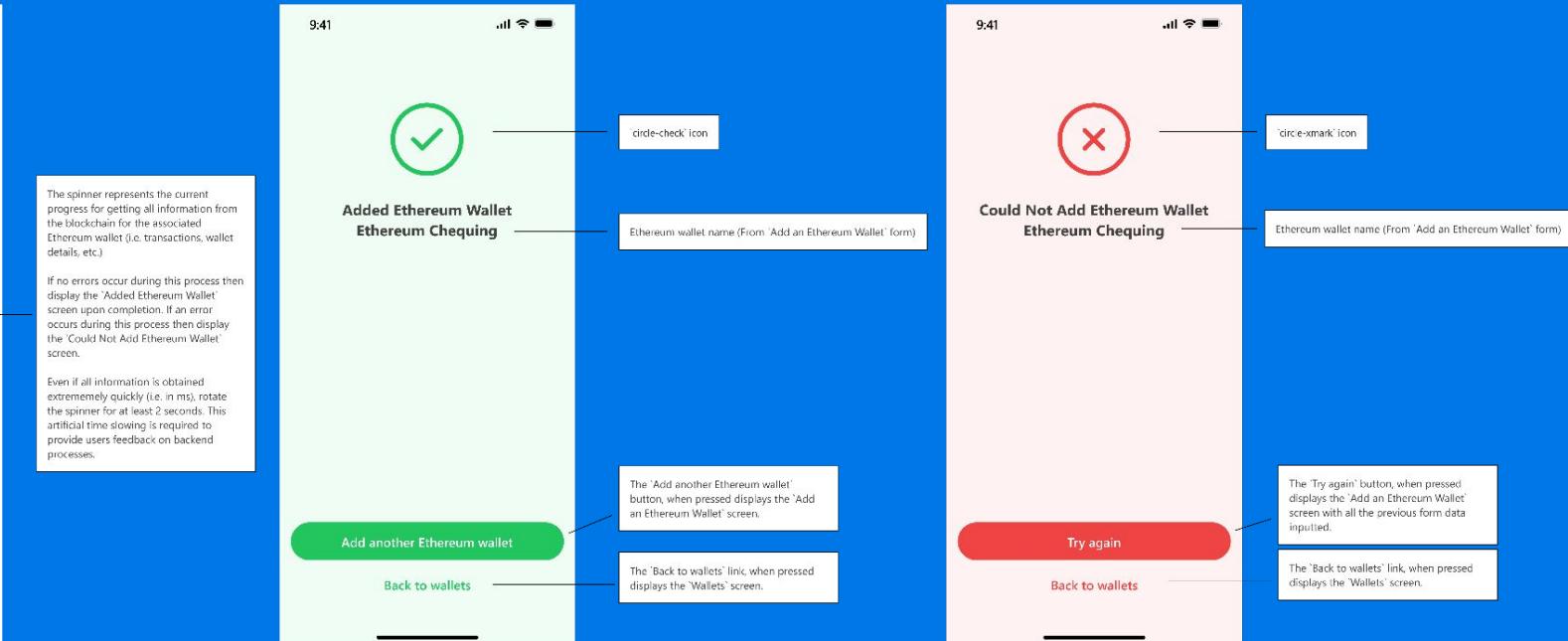
Screenshot 3: Error State

The screen shows a red X icon and the text "Could Not Add Bitcoin Wallet" followed by the name "Bitcoin Savings". Below this, there is a "Try again" button and a "Back to wallets" link. Callouts explain the "circle-smash" icon and the "Back to wallets" link. A note indicates that the "Back to wallets" link, when pressed, displays the "Wallets" screen.

Settings / Wallets / Add a Wallet cont.



Settings / Wallets / Add a Wallet cont.



The first screenshot shows a spinner icon indicating progress. A note states: "The spinner represents the current progress for getting all information from the blockchain for the associated Ethereum wallet (i.e. transactions, wallet details, etc.)." Another note specifies: "If no errors occur during this process then display the 'Added Ethereum Wallet' screen upon completion. If an error occurs during this process then display the 'Could Not Add Ethereum Wallet' screen." A third note indicates: "Even if all information is obtained extremely quickly (i.e. in ms), rotate the spinner for at least 2 seconds. This artificial time slowing is required to provide users feedback on backend processes."

The second screenshot shows a success message: "Added Ethereum Wallet Ethereum Chequing". It includes a green checkmark icon labeled "circle-check" icon, the wallet name "Ethereum wallet name (From 'Add an Ethereum Wallet' form)", and a "Back to wallets" link. A note describes the "Add another Ethereum wallet" button as "The 'Add another Ethereum wallet' button, when pressed displays the 'Add an Ethereum Wallet' screen."

The third screenshot shows an error message: "Could Not Add Ethereum Wallet Ethereum Chequing". It includes a red X icon labeled "circle-xmark" icon, the wallet name "Ethereum wallet name (From 'Add an Ethereum Wallet' form)", and a "Try again" button. A note describes the "Try again" button as "The 'Try again' button, when pressed displays the 'Add an Ethereum Wallet' screen with all the previous form data inputted." It also includes a "Back to wallets" link.

Testing Infrastructure

Unit Tests

Backend

- Unit tests are written using the Jest testing and mocking framework

Frontend

- Unit tests are done through Jest snapshot tests which store the rendered html of the components as files and “expect” against them when those files change

Integration Tests

- Integration tests are written for the backend which test against the integration of the API and the backend as well as the backend with the database as they are run using a live pre-seeded instance of our production database running in docker

System Tests

- Since our platform is mobile app focused we will do our automated system tests through the app
- Make use of a framework call Detox which will give us selenium like control over the application and assertions / expectations
- There will be 1 system test per path per user story
- Each of them will be recorded and saved to the repository

Acceptance Tests

- Acceptance tests will be manually run with a list of acceptance criteria with the stakeholder
- The test is successful once the stakeholder has seen the run and checked off all the ACs
- Done in the last sprint per release for all the user stories of that release
- Each user story will have a sub-task for the acceptance test where the stakeholder can either comment that it is approved or if changes need to be made

Demo Time!

Questions?