

# Cryptify

Crypto asset management  
made simple

Sprint 2 Presentation  
Team 20

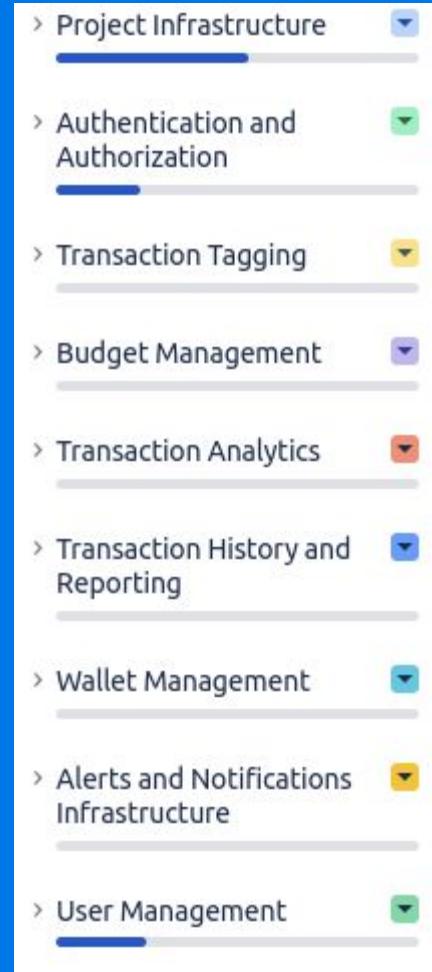
# What is Cryptify?

- 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



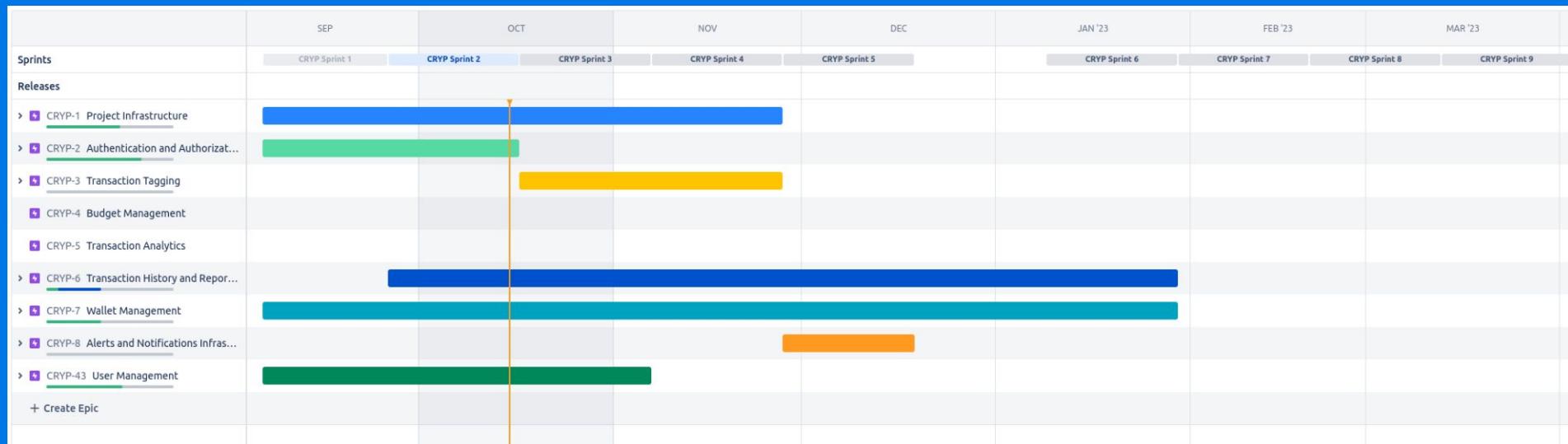
# Backlog To Dos

- 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

# Updates in the backlog

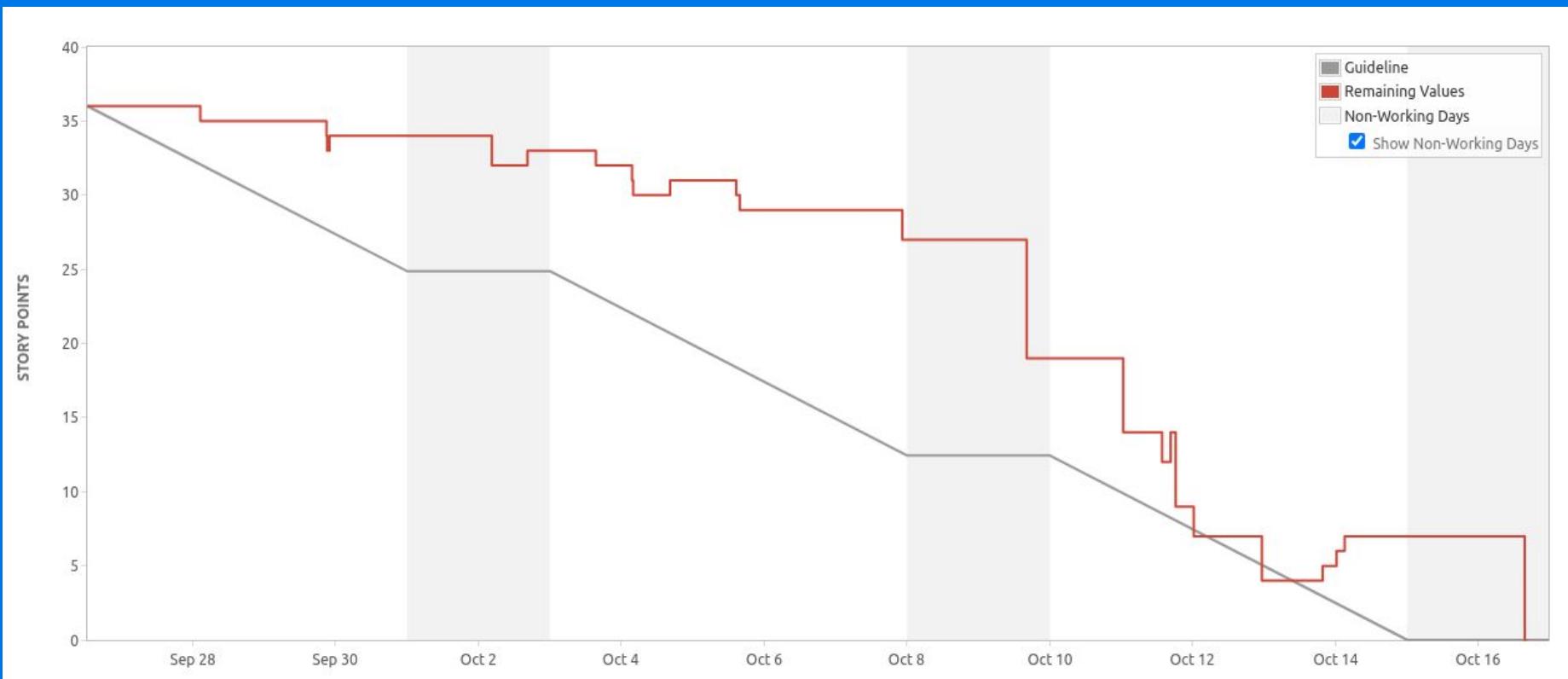
- We split many user stories from “Ethereum and Bitcoin” to a separate user story for each, this allows us to reduce some of the implementation complexity because we only have to focus on 1 at a time
- Created some refactoring tasks for areas in the code that certain design patterns could be applied to in order to improve maintainability and readability

# Roadmap Progress



# Sprint Retrospective and Burndown

# Sprint Burndown Chart



# Sprint Retrospective

Went Well	To Improve	Action Items
<p>Good work meeting code freeze deadline as per the action item from last retrospective</p> <p>--</p> <p>Respecting internal deadlines</p>	<p>even though we only have one deadline individual tasks should be done faster to avoid blocking other people or doing things at the last minute</p> <p>--</p> <p>Knowing who is doing what tasks</p>	<p>@everyone communicate with UI/UX design team to coordinate mid development spec changes as they occur so everyone is on the same page, and mockups / specs can be adjusted accordingly</p>
<p>refactoring tasks where good</p>	<p>Keep learning about the stack</p>	<p>@everyone moving forward 1 person per user story / task / bug / spike and the number of pull requests is up to the person</p>
	<p>Communication between backend, frontend and design when specs change mid development</p> <p>--</p> <p>Communication about the roadmap and overall plan with the team</p> <p>--</p> <p>we need to ask more questions if there is something that is blocking</p>	

# Plan For Next Sprint

# Goals of next sprint

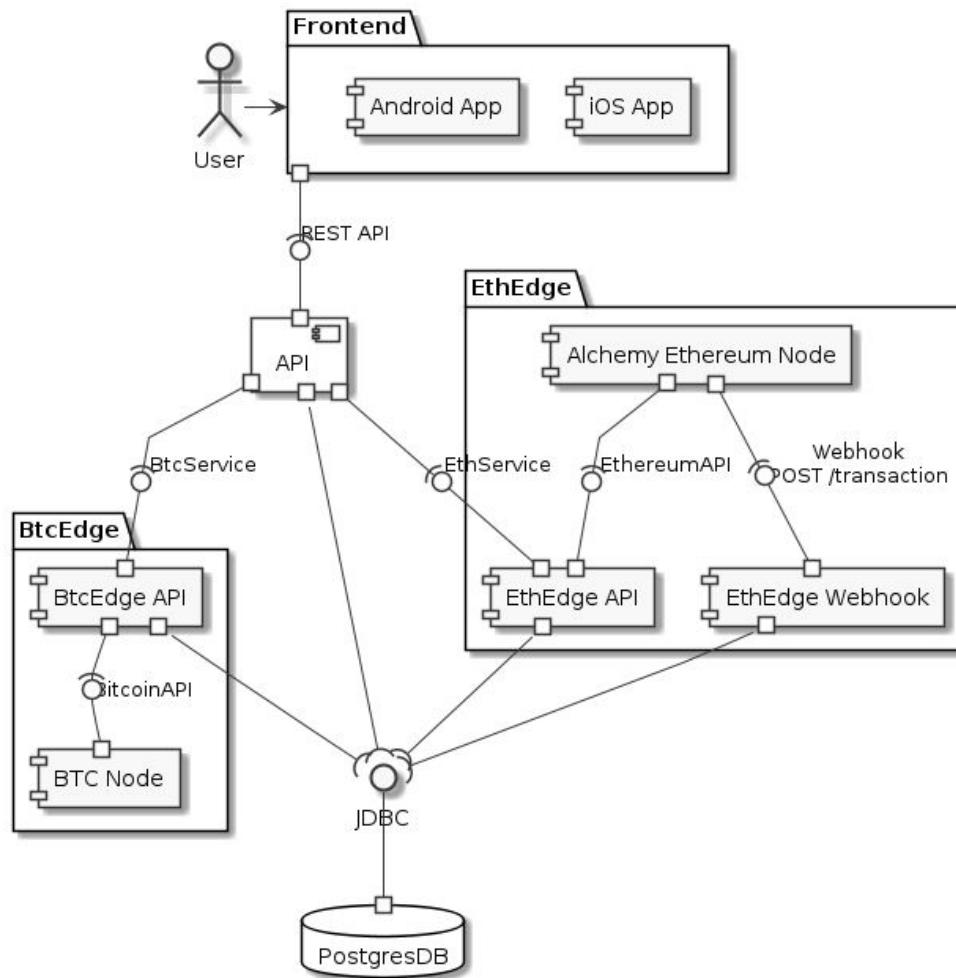
- Implement transaction history
  - Transactions list
  - Details pages
  - Sorting
  - Filtering
- Finish sign up / sign in flow with welcome page and app home screen
- Wallets details page
- Wallets sharing by QR code
- Refactoring tasks
- Setting up and doing the system tests
- Code quality with SonarQube
- Investigation of the BtcEdge architecture for Bitcoin integration

# System Architecture and Diagrams

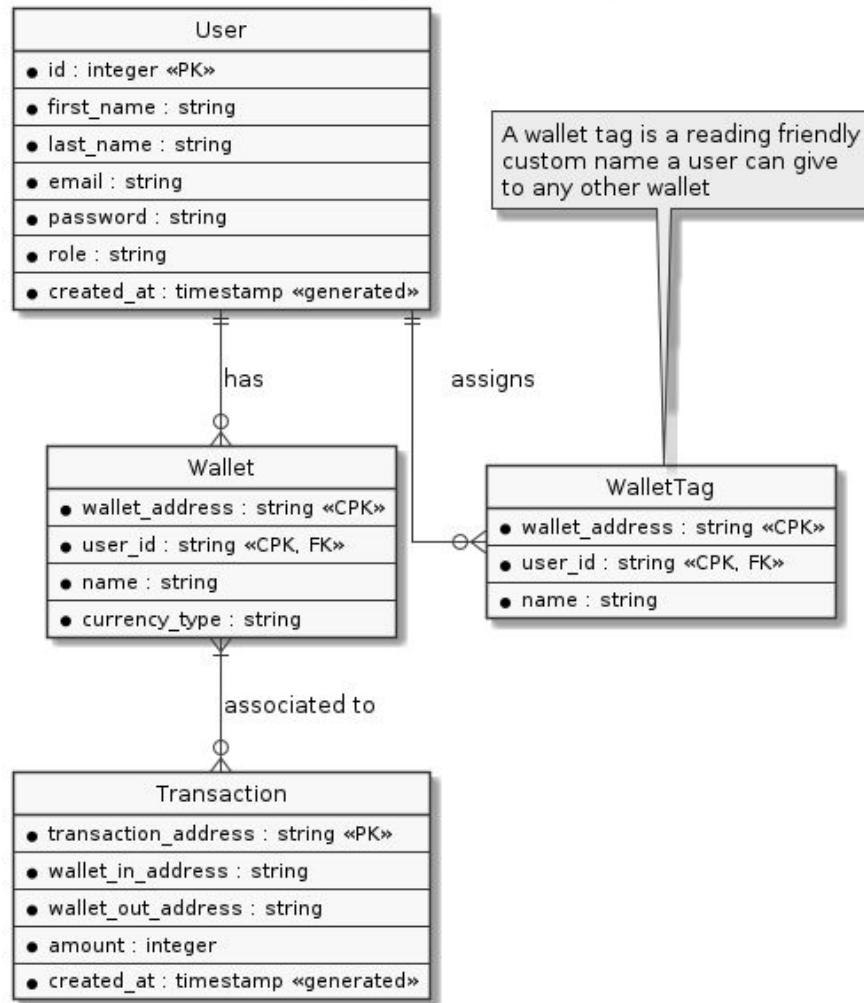
# High Level Architecture

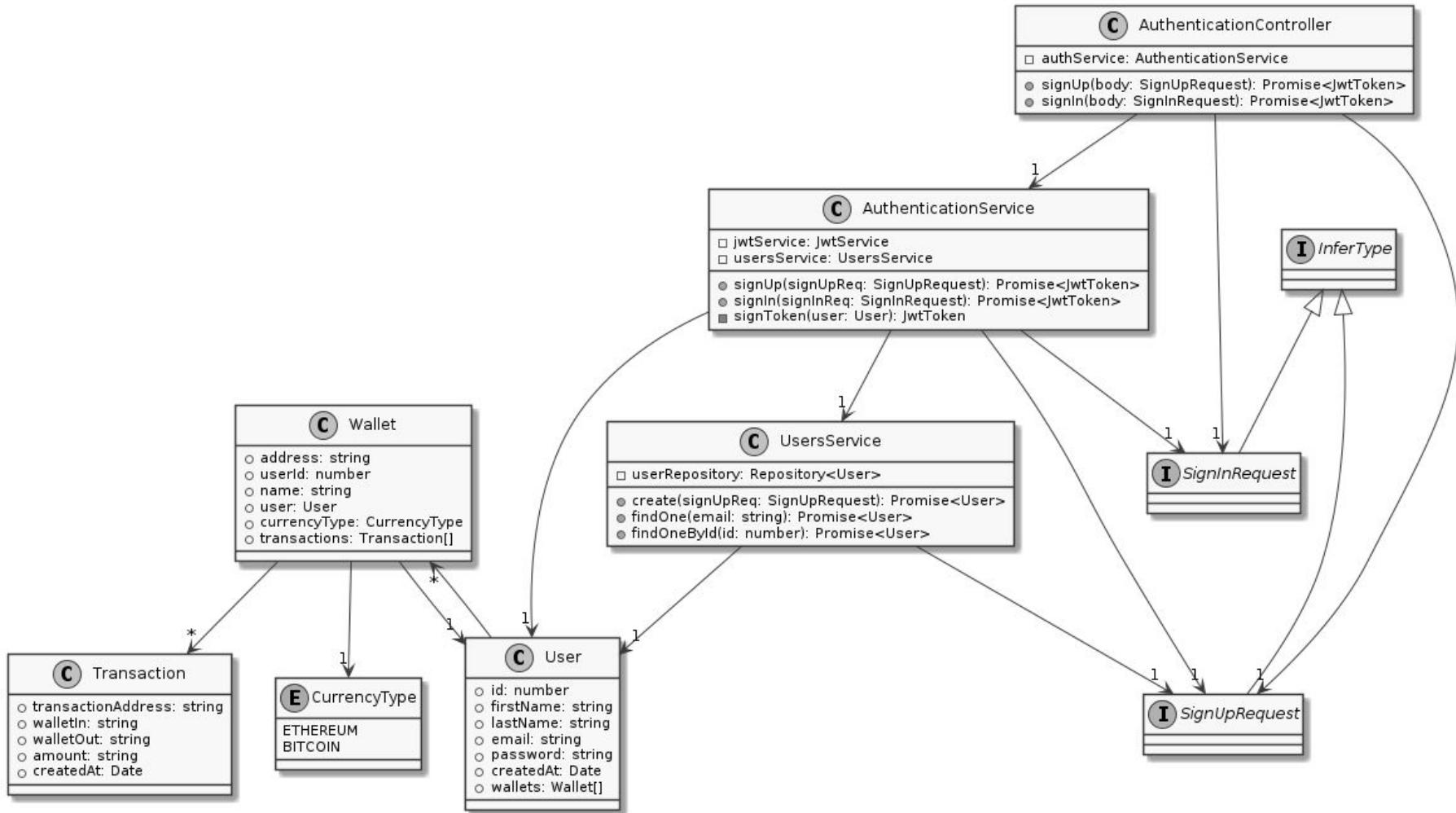
- Cross platform frontend client written in React Native, builds for
  - iOS
  - Android
  - Web
- Backend API / frontend gateway
- Edge services for each of our crypto blockchain integrations
  - EthEdge
  - BtcEdge
- PostgreSQL DB handling all persistent storage
- Everything is running in Docker
-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

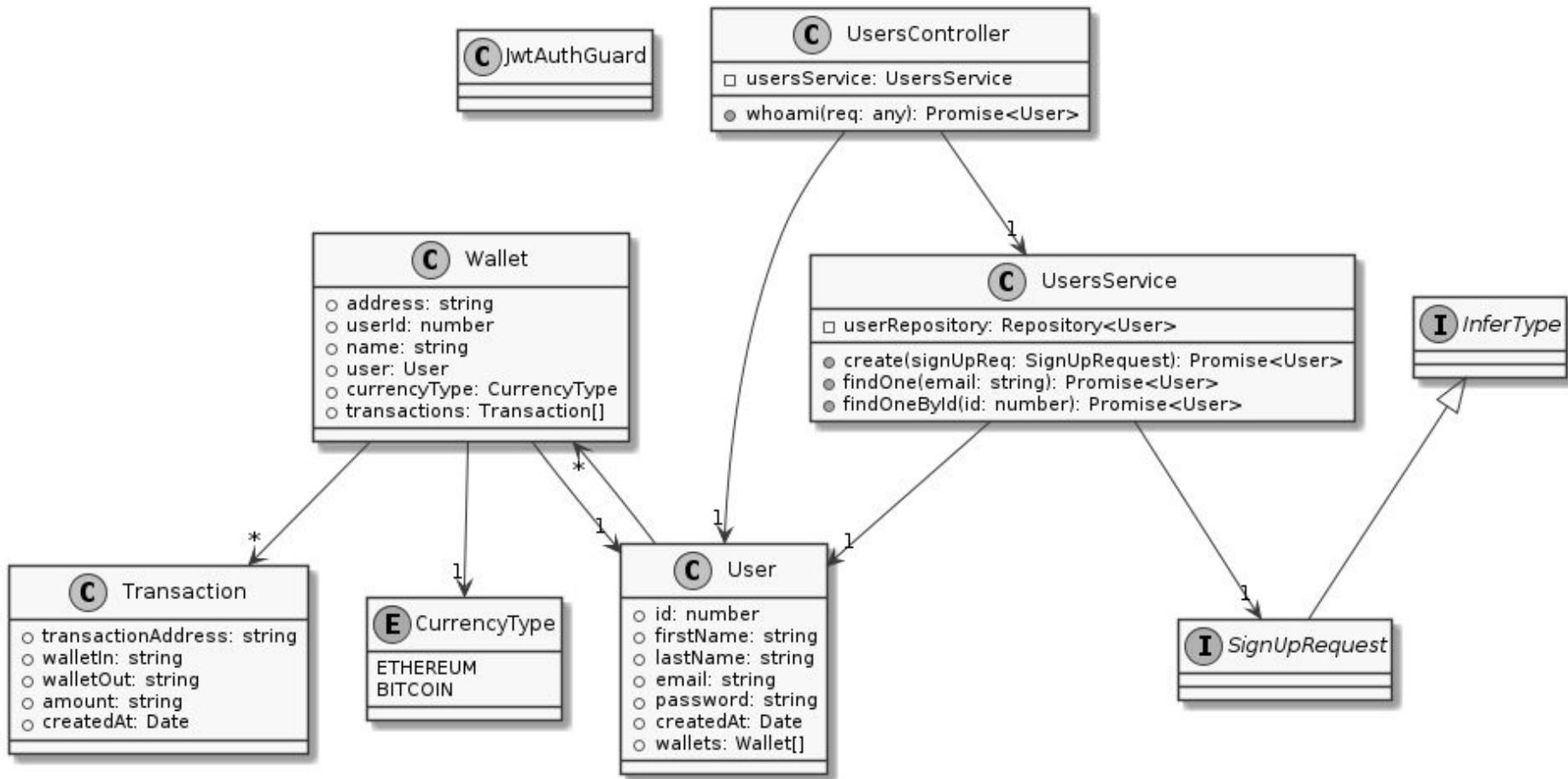
# Cryptify Component Diagram

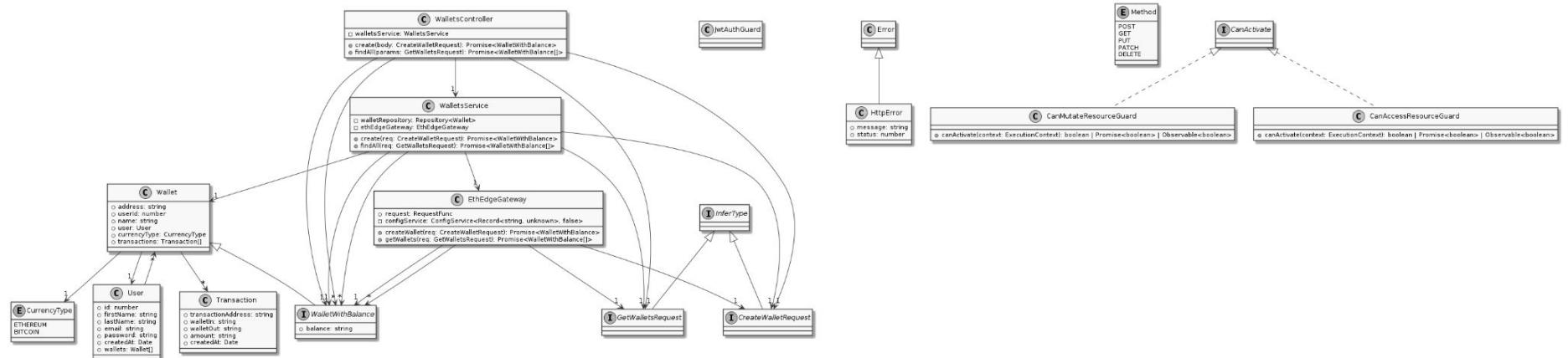


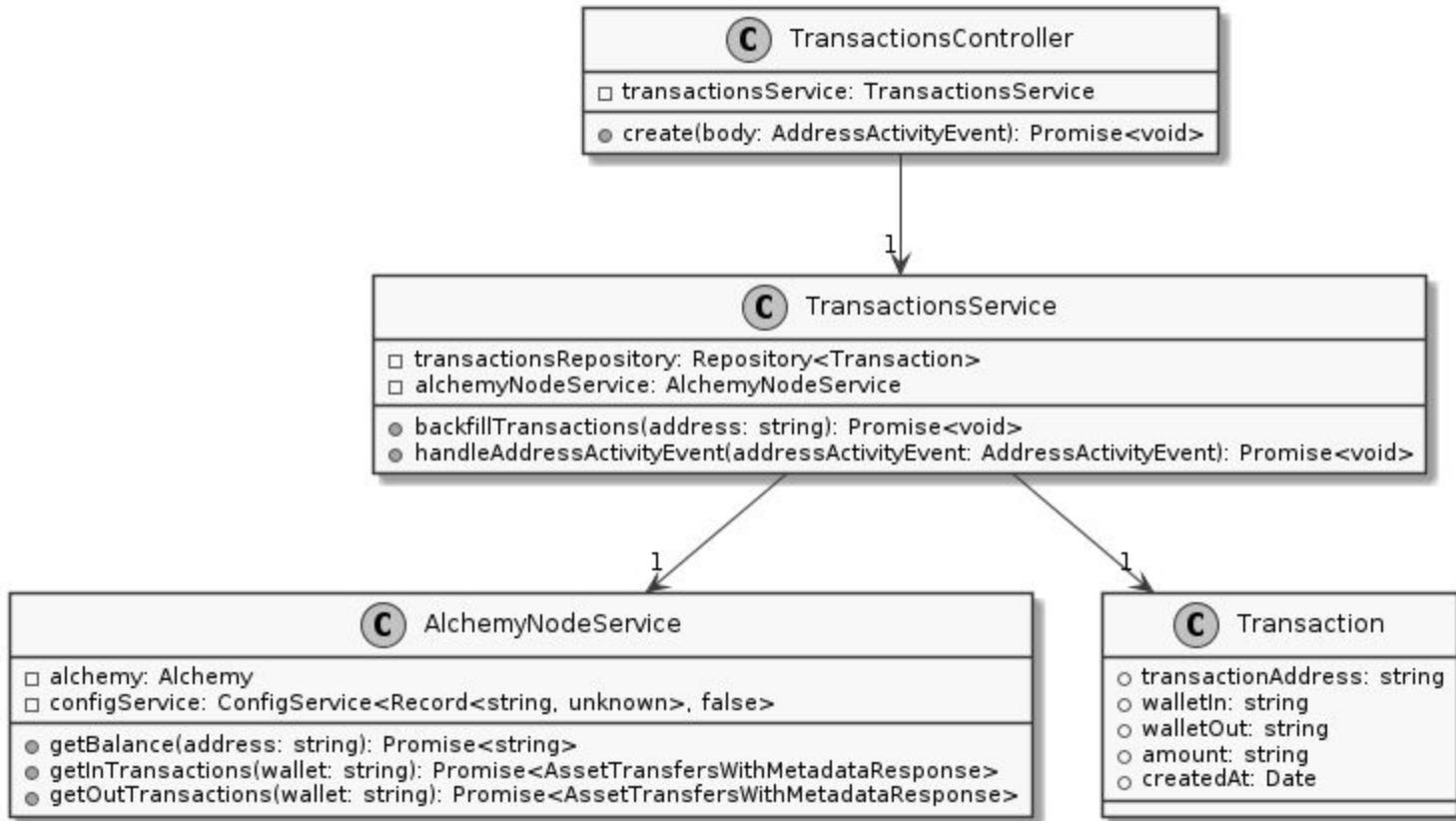
# Cryptify ER Diagram

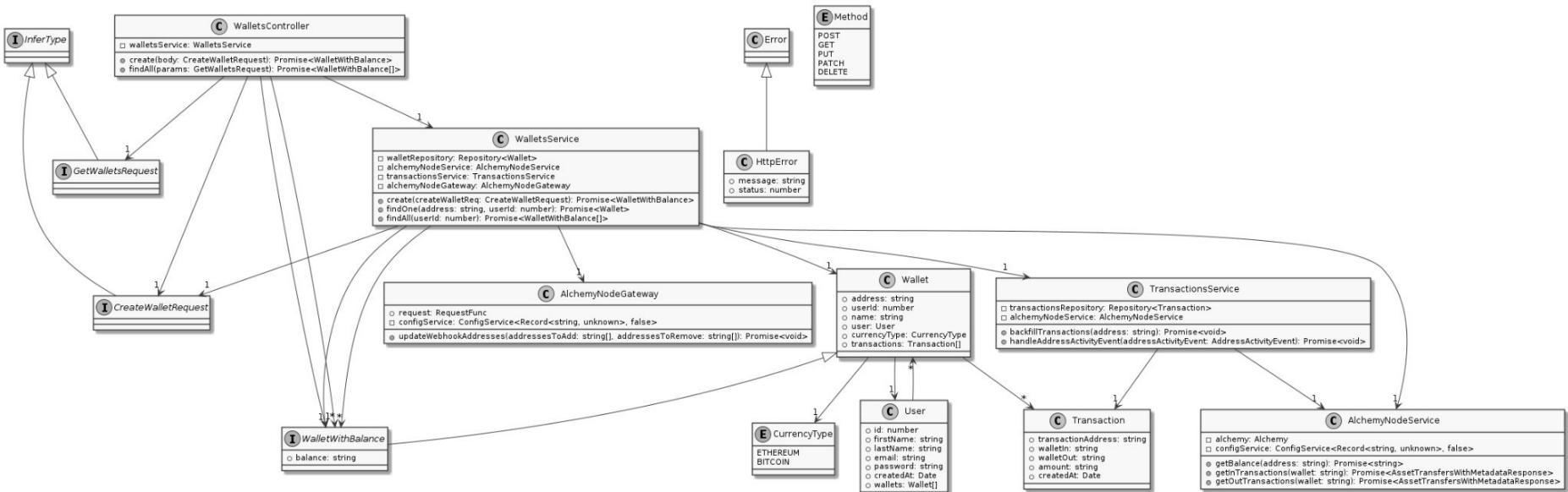




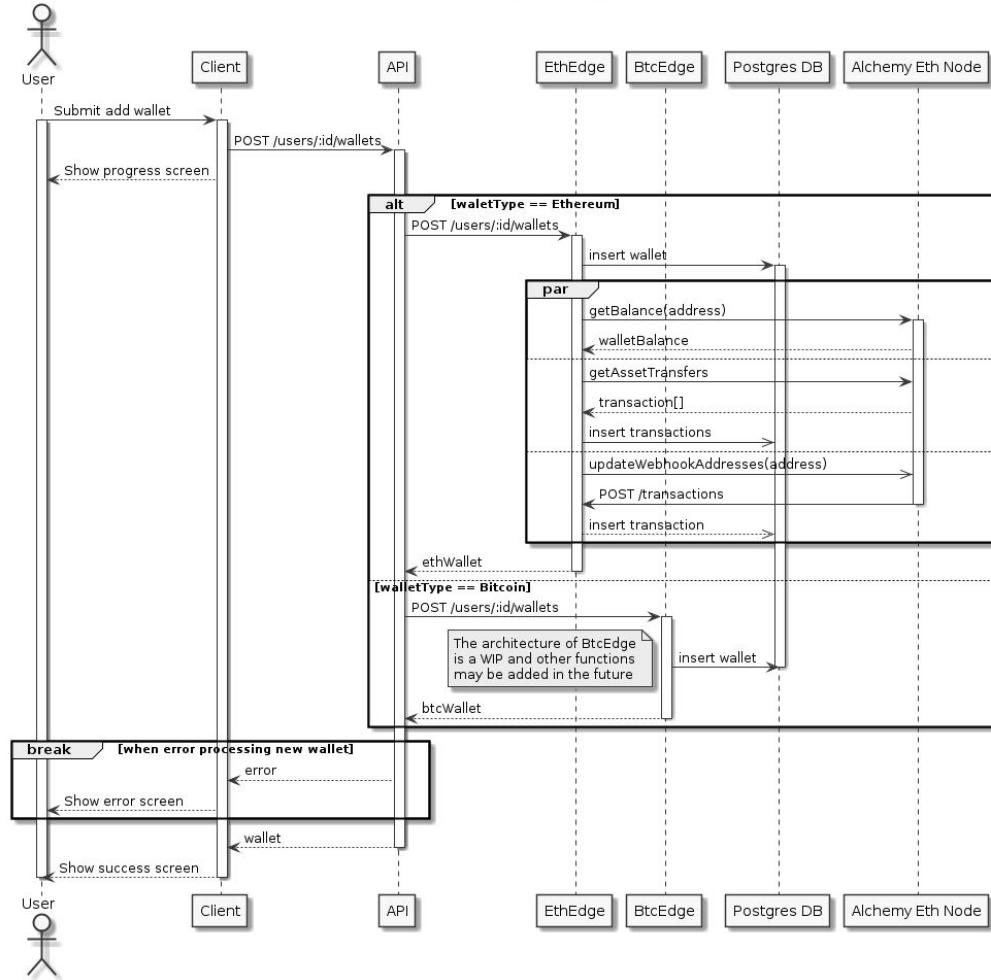




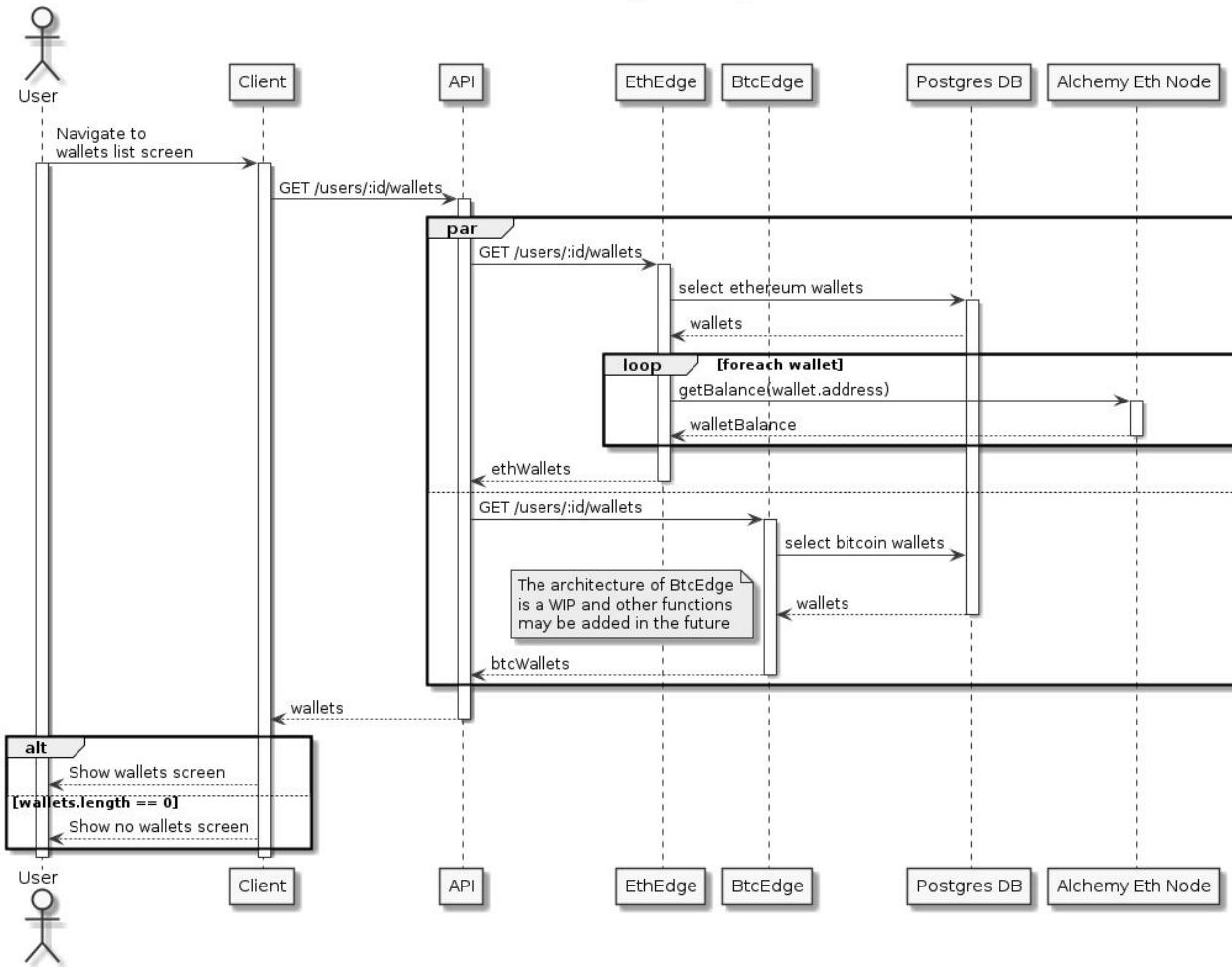




### Add Wallet Sequence Diagram



### View Wallets Sequence Diagram



# Design Pattern - Controller

- Handles incoming request
  - Guards is a service provided by Nestjs, determines if request will be handled
- Validates data
  - Yup schema with inferred type
- Calls the service

Example Controller Method:

```
@UseGuards(JwtAuthGuard, CanAccessResourceGuard)
@Get("/users/:id/wallets")
async findAll(@Param() params: GetWalletsRequest): Promise<WalletWithBalance[]> {
    const getWalletsReq = await useValidate(getWalletsSchema, params);

    return this.walletsService.findAll(getWalletsReq);
}
```

# Design Pattern - Service

- Handles business logic
- Calls the other services, repositories, gateways, etc.

Example Service Method:

```
async findAll(userId: number): Promise<WalletWithBalance[]> {
  const wallets = await this.walletRepository.find({ where: { currencyType: CurrencyType.ETHEREUM, userId } });
  // For each wallet owned by the user get the wallets balance from Alchemy, we can parallelize these requests
  // to speed up the processing time
  const balances = await Promise.all(
    wallets.map(async (wallet) => this.alchemyNodeService.getBalance(wallet.address)),
  );
  // Once all the balances have been retrieved zip the lists together and map through them to construct the final
  // object, Promise.all will return the values in the same order we inputted them meaning the wallets and balances
  // will line up when we zip them
  return zip(wallets, balances).map(([wallet, balance]) => ({
    ...wallet,
    balance,
  }));
}
```

# Design Pattern - Repository / Data Mapper

- typeORM
- For each entity we have a repository
  - Logic to communicate with the database
- Follows the Data Mapper pattern
  - Create
  - Read
  - Update
  - delete

```
async findOne(email: string): Promise<User> {
    return this.userRepository.findOne({ where: { email } });
}
```

```
@Entity()
export class User {
    @PrimaryGeneratedColumn()
    id: number;

    @Column({ nullable: false })
    firstName: string;

    @Column({ nullable: false })
    lastName: string;

    @Column({ nullable: false, unique: true })
    email: string;

    @Column({ nullable: false })
    password: string;

    @CreateDateColumn({ type: "timestamp" })
    createdAt: Date;

    @OneToMany(() => Wallet, (wallet) => wallet.user)
    wallets: Wallet[];
}
```

# Design Pattern - Gateway

- Used for outgoing http request
  - Builds request url
  - Sends the request
  - Returns the response
- Client has an Api Gateway
- Api has a Eth-Edge Gateway

```
@Injectable()
export class EthEdgeGateway {
    request: RequestFunc;

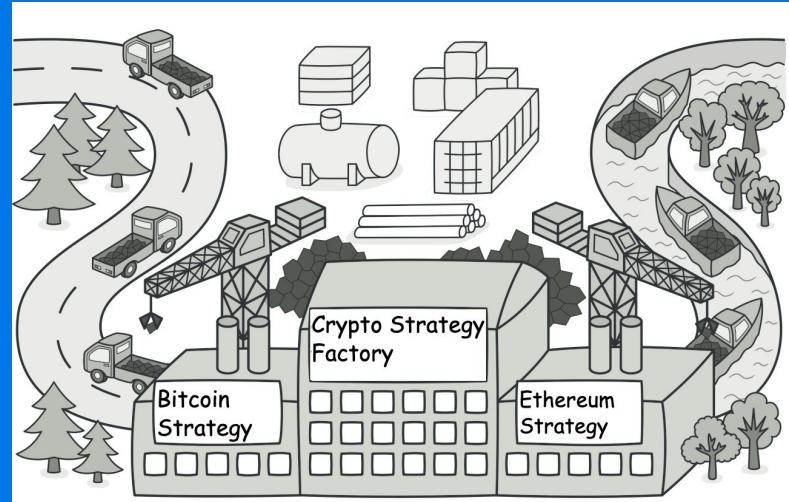
    constructor(private configService: ConfigService) {
        const host = configService.get<string>("ETH_EDGE_HOST");
        const port = configService.get<string>("ETH_EDGE_PORT");
        this.request = request(`http://${host}:${port}`);
    }

    async createWallet(req: CreateWalletRequest): Promise<WalletWithBalance> {
        const path = `users/${req.userId}/wallets`;
        return this.request<WalletWithBalance>(Method.POST, {}, path, req);
    }

    async getWallets(req: GetWalletsRequest): Promise<WalletWithBalance[]> {
        const path = `users/${req.id}/wallets`;
        return this.request<WalletWithBalance[]>(Method.GET, {}, path, null);
    }
}
```

# Design Pattern - Factory / Strategy (to be implemented)

- Since we will support different cryptocurrencies we need a strategy for each one because the logic will be different
- And each currency will have its own node service
- We will have a factory that will decide what strategy should be used based on the wallet address



# UI Mockups

## Sprint 2 Updates

# Components, Spacing, Styling, and Typography Standardization

## Components (w/ states):

- Form
  - Input
  - Radio button & Radio group
  - Button
- List
- Accordion
- Link
- Icon
- Toast
- ...and many more

## Typography (Regular & Semibold)

- Title
- Subheadline
- Body
- Footnote

## Spacing

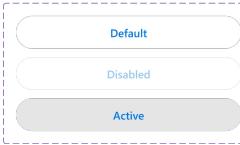
- Margin
- Padding

## Button

Solid



Outline



## Input

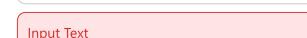
Focused



Entered



Error (w/ Placeholder)



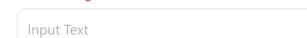
Error message.

Error (w/ Entered Text)



Error message.

Disabled



## Typography

### Regular

Default / Regular / Caption1

Default / Regular / Footnote2

Default / Regular / Footnote1

Default / Regular / Subheadline

Default / Regular / Body2

Default / Regular / Body

Default / Regular / Title3

Default / Regular / Title2

Default / Regular / Title1

### Semibold

Default / Semibold / Footnote2

Default / Semibold / Footnote1

Default / Semibold / Subheadline

Default / Semibold / Body

Default / Semibold / Title3

Default / Semibold / Title2

Default / Semibold / Title1

# Settings / Wallets / Add a Wallet

**'Name' error messages:**

- "Enter a name."
- "Name must only contain alphanumeric characters and spaces."
- "Name must be 32 characters or less."

Display the following error message if the wallet name is found to be already associated with another wallet during the 'Adding Bitcoin Wallet' server side Bitcoin wallet validation and the 'Add a Bitcoin Wallet' screen is re-displayed

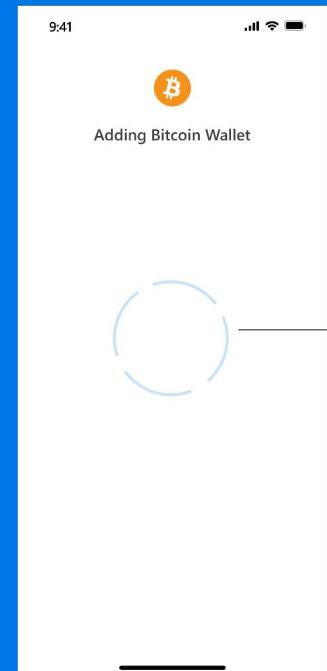
- "This name is already associated with another wallet."

**'Wallet address' error messages:**

- "Enter a wallet address."
- "Enter a valid Bitcoin wallet address."

Display the following error message if the wallet address is found to be already associated with another Bitcoin wallet during the 'Adding Bitcoin Wallet' server side Bitcoin wallet validation and the 'Add a Bitcoin Wallet' screen is re-displayed

- "This wallet address is already associated with another Bitcoin wallet."



The spinner represents the current progress for performing server side Bitcoin wallet validation and getting all information from the blockchain (i.e. transactions, wallet details, etc.)

- If the wallet name is already associated with another wallet, display the 'Add a Bitcoin Wallet' screen with the following error message under the 'Name' input field
  - "This name is already associated with another wallet."
- If the wallet address is already associated with another Bitcoin wallet, display the 'Add a Bitcoin Wallet' screen with the following error message under the 'Wallet address' input field
  - "This wallet address is already associated with another Bitcoin wallet."
- If a wallet address is valid but not posted on the blockchain, assume it exists and give it a balance of 0.

If no fatal errors (i.e. issue with third-party API, network error, etc.) occur during this process then display the 'Added Bitcoin Wallet' screen upon completion else 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.

# Settings / Wallets / Add a Wallet cont.

'Name' error messages:

- "Enter a name."
- "Name must only contain alphanumeric characters and spaces."
- "Name must be 32 characters or less."

Display the following error message if the wallet name is found to be already associated with another wallet during the 'Adding Ethereum Wallet' server side Ethereum wallet validation and the 'Add an Ethereum Wallet' screen is re-displayed

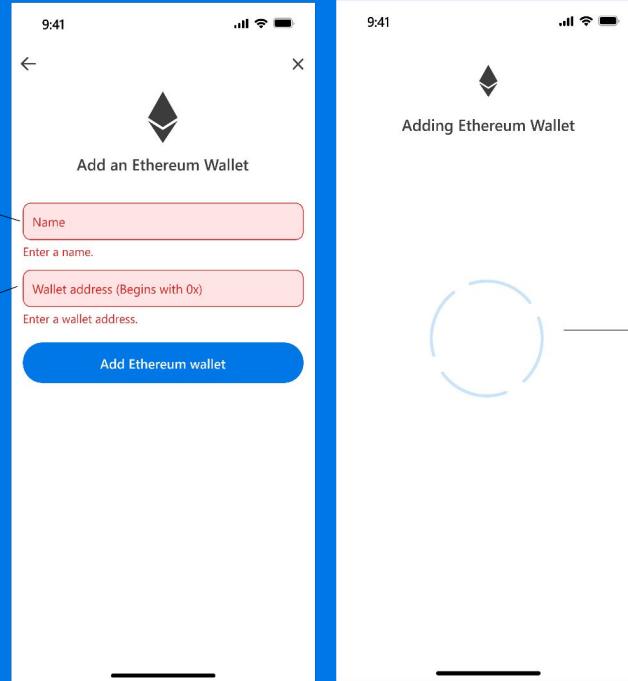
- "This name is already associated with another wallet."

'Wallet address' error messages:

- "Enter a wallet address."
- "Enter a valid Ethereum wallet address."

Display the following error message if the wallet address is found to be already associated with another Ethereum wallet during the 'Adding Ethereum Wallet' server side Ethereum wallet validation and the 'Add an Ethereum Wallet' screen is re-displayed

- "This wallet address is already associated with another Ethereum wallet."



The spinner represents the current progress for performing server side Ethereum wallet validation and getting all information from the blockchain (i.e. transactions, wallet details, etc.)

- If the wallet name is already associated with another wallet, display the 'Add an Ethereum Wallet' screen with the following error message under the 'Name' input field
  - "This name is already associated with another wallet."
- If the wallet address is already associated with another Ethereum wallet; display the 'Add an Ethereum Wallet' screen with the following error message under the 'Wallet address' input field
  - "This wallet address is already associated with another Ethereum wallet."
- If a wallet address is valid but not posted on the blockchain, assume it exists and give it a balance of 0.

If no fatal errors (i.e. issue with third-party API, network error, etc.) occur during this process then display the 'Added Ethereum Wallet' screen upon completion else display the 'Could Not Add Ethereum 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.

# UI Mockups

## Sprint 3

# Welcome

## User Stories:

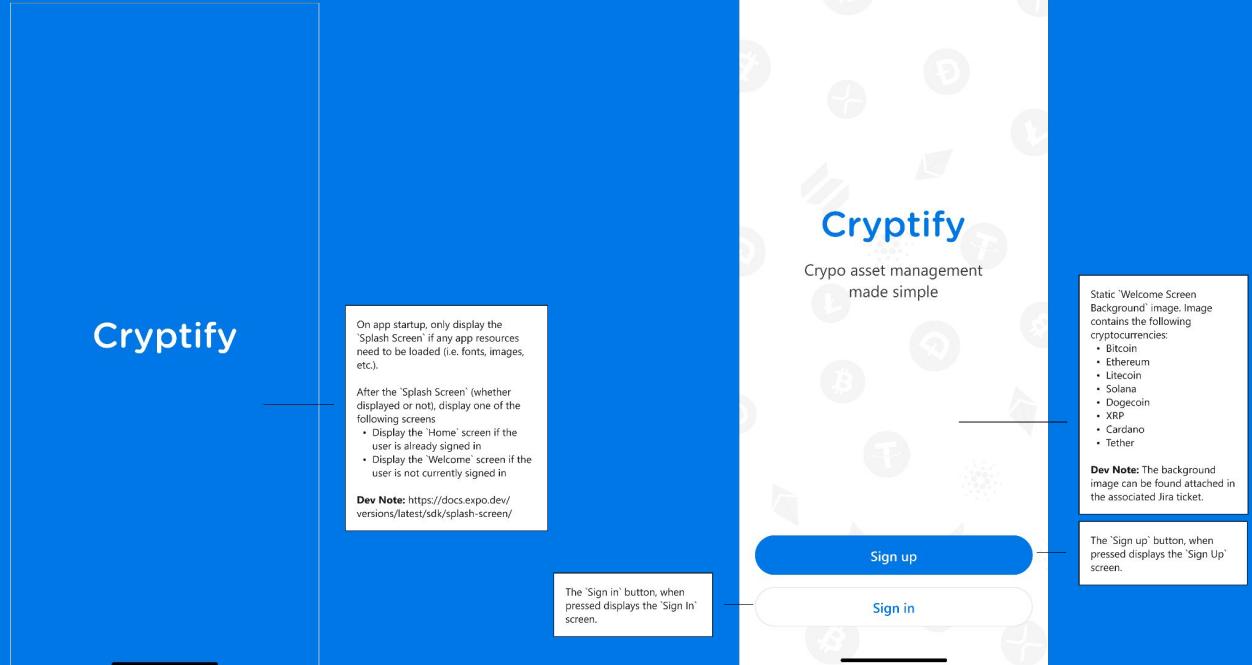
- CRYP-100: As a User, I want a welcome page when I open the app and I'm not signed in, so I can either sign up for an account or sign in to my account

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

"Love the design of the main page. The subtle logos in the background are a nice touch."

- Stakeholder



# Home

## User Stories:

- CRYP-101: As a User, I want a home page, so I can view all my wallets immediately after signing in

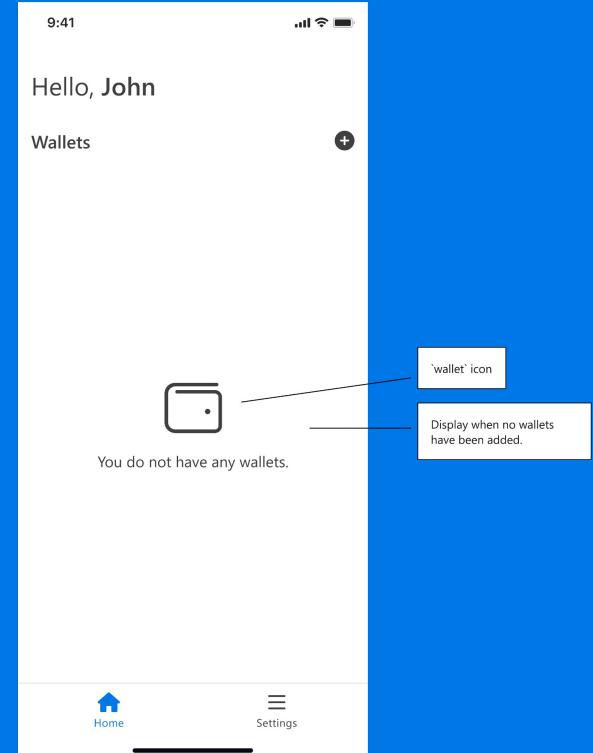
## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

"The blue to indicate it is the home page is a nice touch. I also like the simplicity of the home page where you can see your wallets.

Are there any plans to add other things to the home page in future releases?"

- Stakeholder



# Home cont.

Account first name. If first name is too long then it should be on a new line  
**Dev Note:** Figma should provide the proper styling for this component behavior under the 'Inspect' sidebar menu.

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 'Home' screen is initially displayed (on sign in), accordions are collapsed. Accordions must maintain their current state (collapsed or expanded) until the user signs out. (i.e. if one of the accordions is open and the user either leaves the app or changes screens, once returned the same accordion that was previously manipulated must remain open).

The 'circle-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.

If the wallet name is too long to fit on a single line or the balance overlaps it, wrap the wallet name on a new line.

Order the wallets in alphabetical order by wallet name.

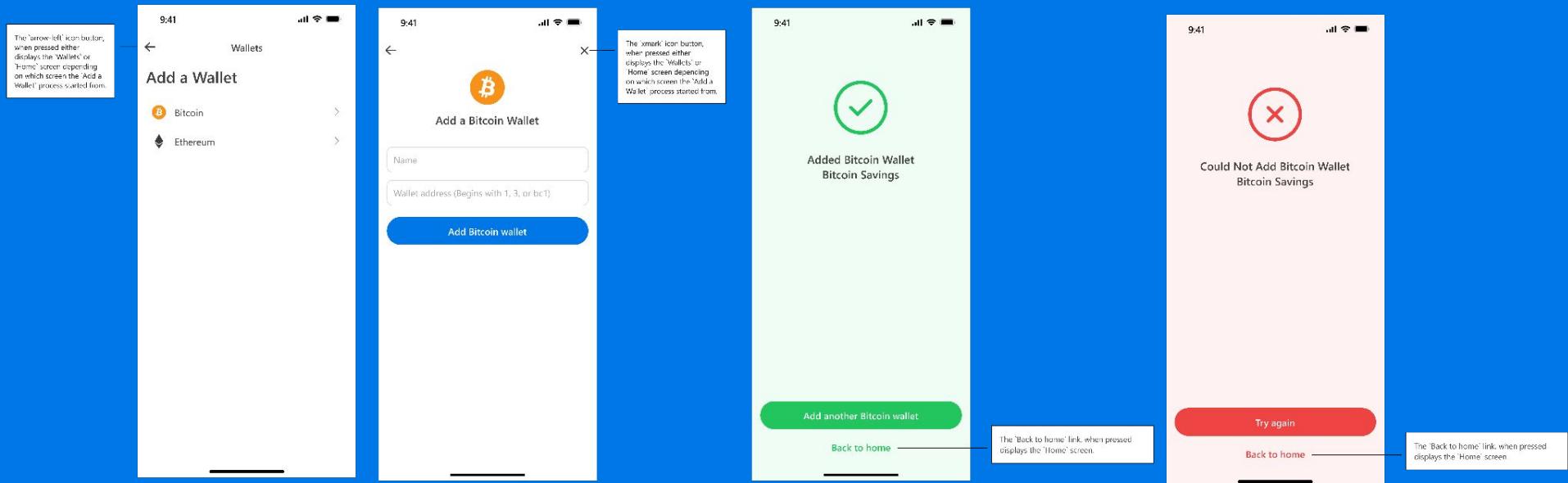
A Bitcoin wallet accordion item, when pressed displays the 'Bitcoin Wallet Overview'.

An Ethereum wallet accordion item, when pressed displays the 'Ethereum Wallet Overview'.

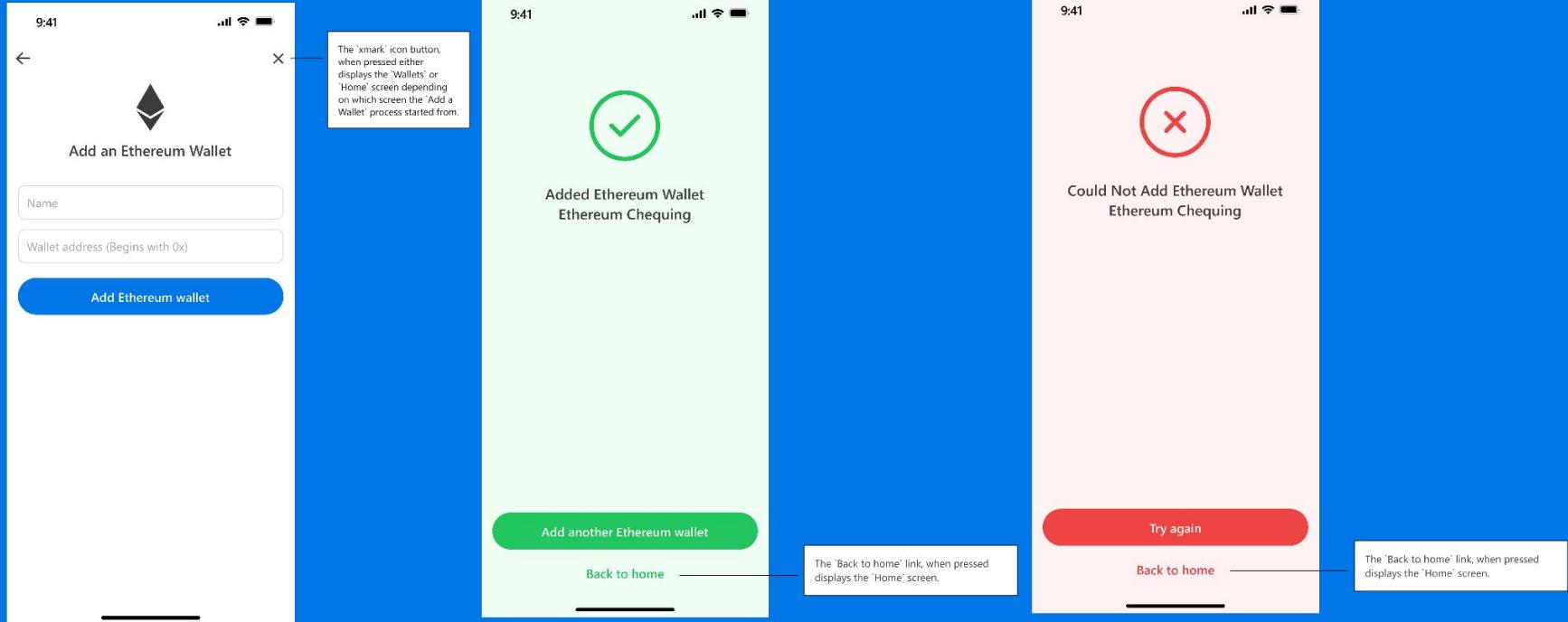
Wallet address format:  
• <first 6 digits><ellipsis><last 4 digits>

Wallet balance (latest) format:  
• Always display all digits before the decimal point  
• Remove any trailing 0 digits after the decimal point  
• Example:  
    • 1,000.0000 -> 1,000  
    • 1,000.0001 -> 1,000.0001  
• If the balance does not display within a single line do the following:  
    • Add an ellipsis at the end of the balance  
    • Keep at least 1 digit after the decimal point and remove as many trailing digits (after the decimal point) necessary until the balance, ellipsis and currency type remain on a single line  
    • Example:  
        • 1,000.000.04832 ETH -> 1,000.000...ETH

# Settings / Wallets / Add Wallet updates



# Settings / Wallets / Add Wallet updates cont.



# Home / Wallets / Ethereum Wallet Overview

## User Stories:

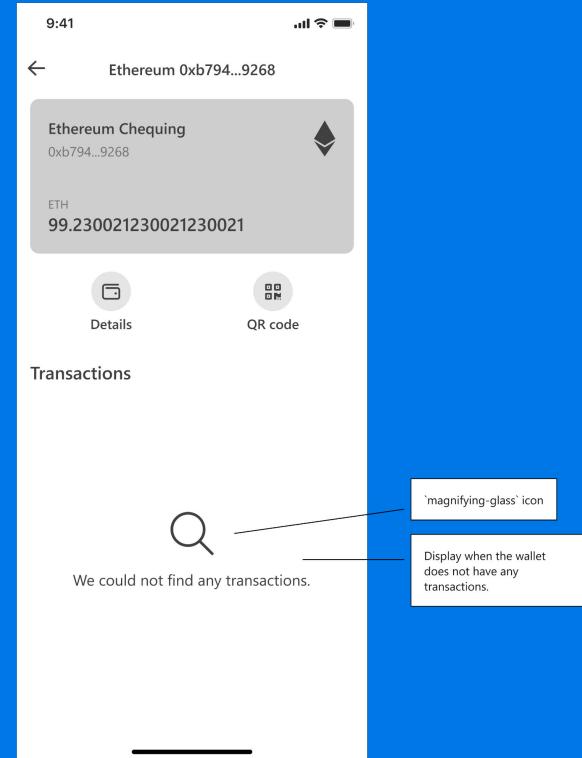
- CRYP-26: As a User, I want to view the details for each of my Ethereum wallets, so that I can see all associated information available on the blockchain
- CRYP-28: As a User, I want to view a list of all transactions associated to each of my Ethereum wallets, so that I can see who I have transacted with
- CRYP-108: As a User, I want to share my Ethereum wallet address, so that others can easily send me crypto without inputting my wallet address incorrectly

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

"The wallet looks good."

- Stakeholder



# Home / Wallets / Ethereum Wallet Overview cont.

## Transactions

### Requirements:

- Display the following
  - Wallet address (sender/receiver)
  - Transaction date
  - Transaction amount w/ currency type
  - Indicate transaction type (send or receive)
- Wallet address (sender/received), transaction amount and type are easily identifiable
- Extendible to include transaction category and sender/receiver name in a future sprint

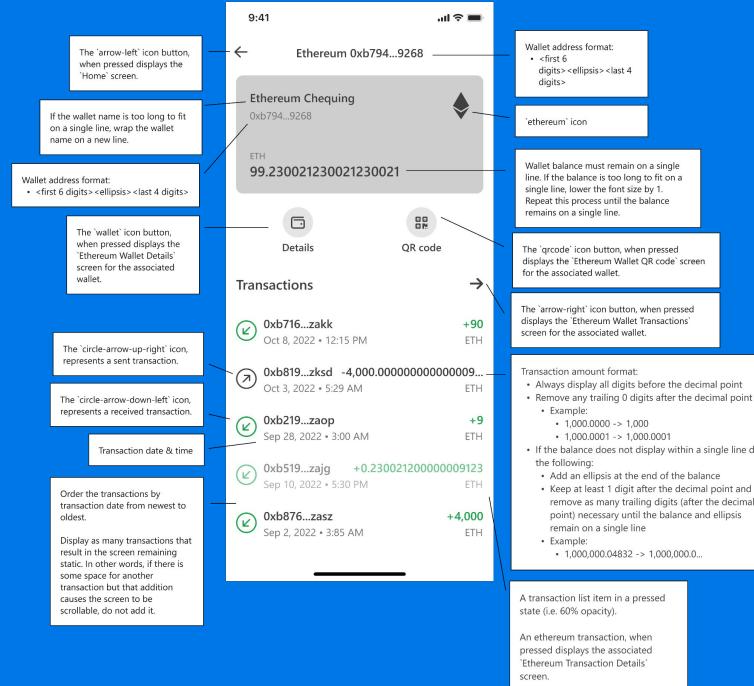
### Challenges:

1. 120+ million Ethereum in circulation
2. Ethereum has 18 decimal places
3. Transaction amount must remain on a single line
4. Mobile phones have smaller screen sizes than desktop
5. Wallet address format cannot be changed

[Home](#) / [Wallets](#) / Ethereum Wallet Overview cont.

## 20+ transaction list item designs tested

## Home / Wallets / Ethereum Wallet Overview cont.



## Final Design

# Home / Wallets / Ethereum Wallet Overview cont.

## Stakeholder Feedback

"ETH is moving away from the grey color scheme towards a more blue tone. Could we try and represent that in the eth logos and the backdrop color? Let me know your thought."

- Stakeholder

After research and discussions with stakeholder,  
no changes made

Reasons for not changing current color scheme

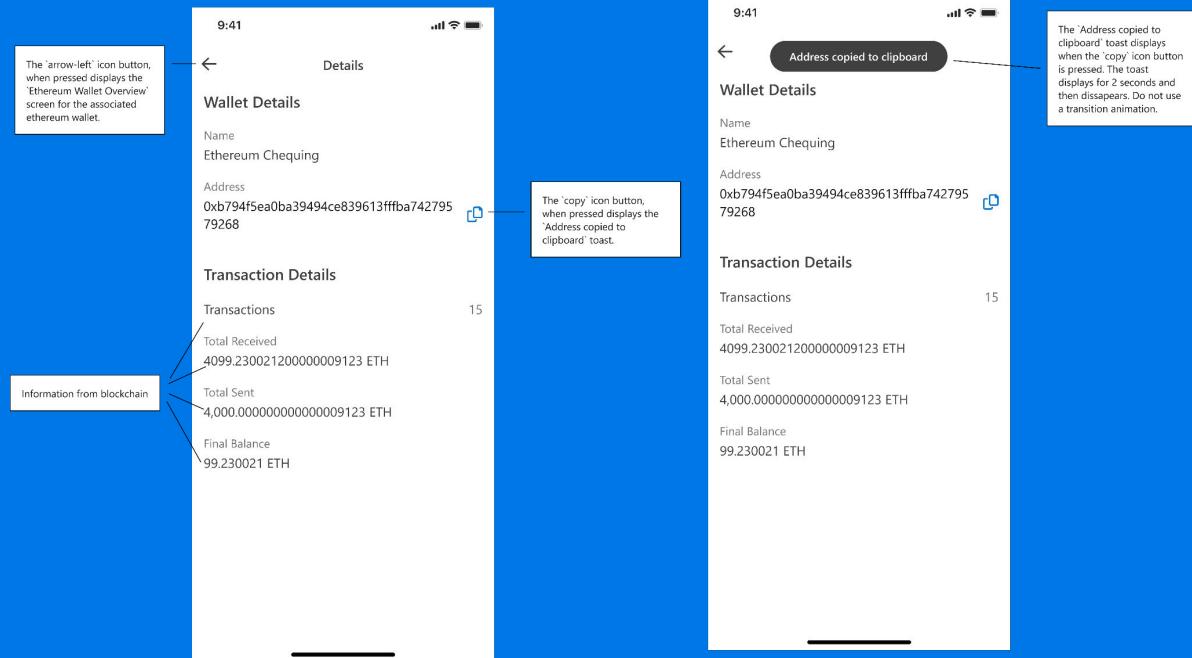
1. Most crypto apps still use the black & grey color scheme
2. Maintain sense of familiarity and cohesiveness for users when switching between Cryptify and other crypto apps (i.e. Coinbase, Binance, Crypto.com)



# Home / Wallets / Ethereum Wallet Details

## User Stories:

- CRYP-26: As a User, I want to view the details for each of my Ethereum wallets, so that I can see all associated information available on the blockchain
  - CRYP-108: As a User, I want to share my Ethereum wallet address, so that others can easily send me crypto without inputting my wallet address incorrectly



“Good to go for coding.”

## - Stakeholder

# Home / Wallets / Ethereum Wallet QR Code

## User Stories:

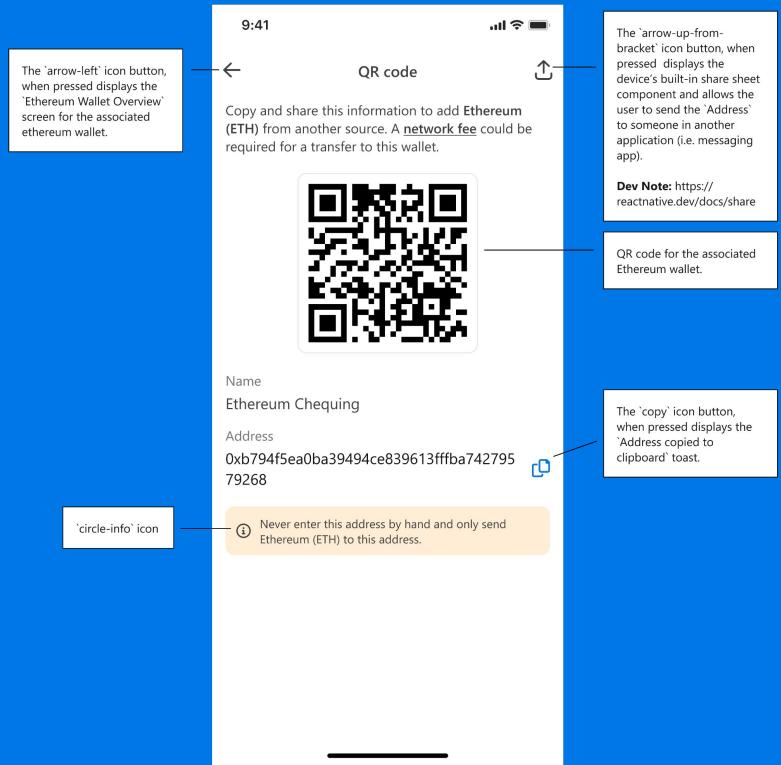
- CRYP-108: As a User, I want to share my Ethereum wallet address, so that others can easily send me crypto without inputting my wallet address incorrectly

## Personas:

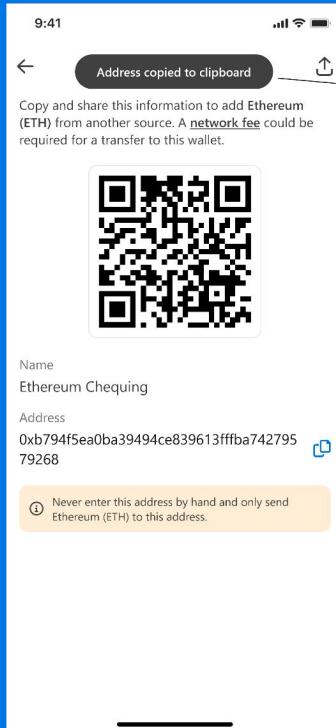
- User
  - The Beginner
  - The Intermediate
  - The Expert

“Really good idea with the qr code and the simplicity of sharing them.”

- Stakeholder

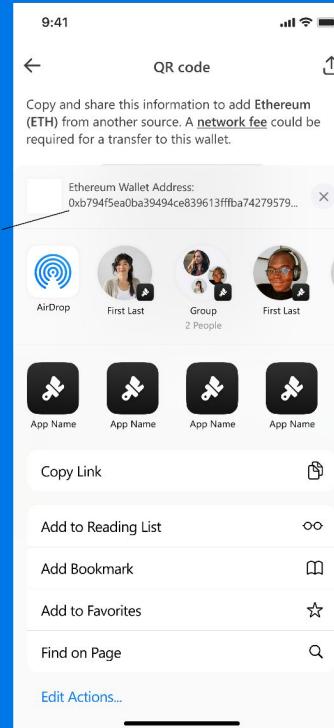


# Home / Wallets / Ethereum Wallet QR Code cont.



The 'Address copied to clipboard' toast displays when the 'copy' icon button is pressed. The toast displays for 2 seconds and then disappears. Do not use a transition animation.

Message format:  
• Ethereum Wallet  
Address: \r\n<Address>



The device's built-in share sheet component is displayed when the 'arrow-up-from-bracket' icon button is pressed.

Dev Note: <https://reactnative.dev/docs/share>

# Home / Wallets / Ethereum Wallet Transactions

## User Stories:

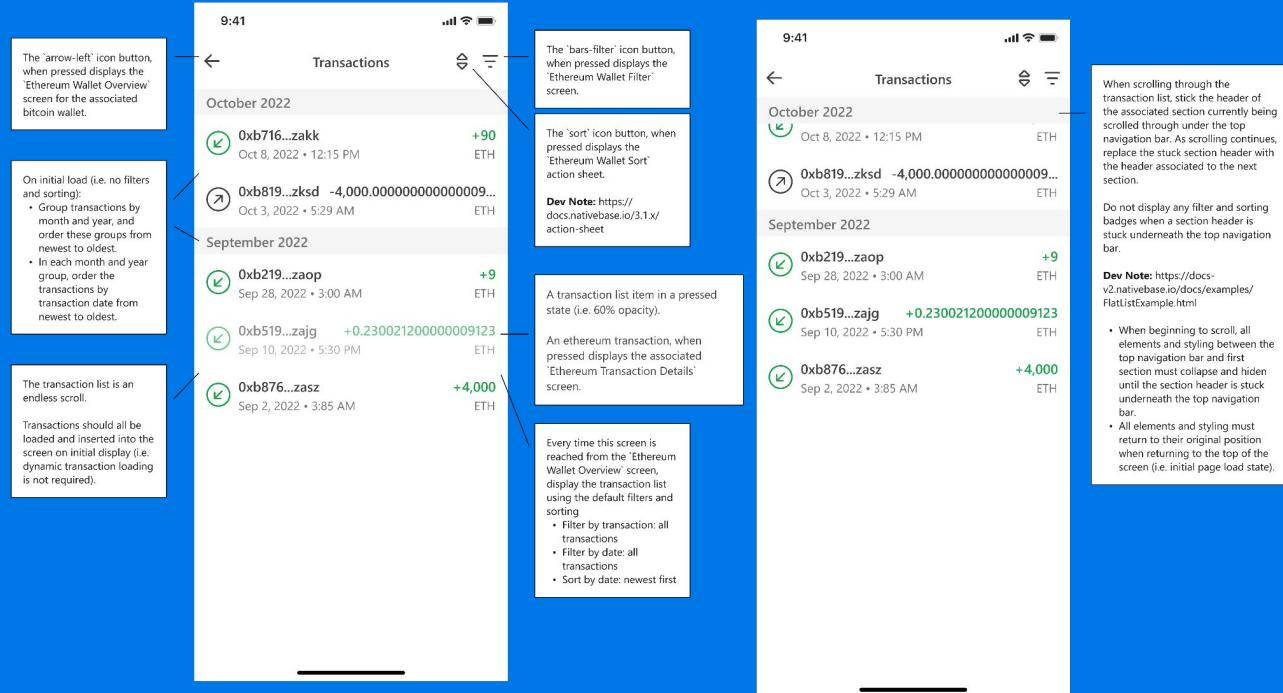
- CRYP-28: As a User, I want to view a list of all transactions associated to each of my Ethereum wallets, so that I can see who I have transacted with

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

"Good to go for coding."

- Stakeholder



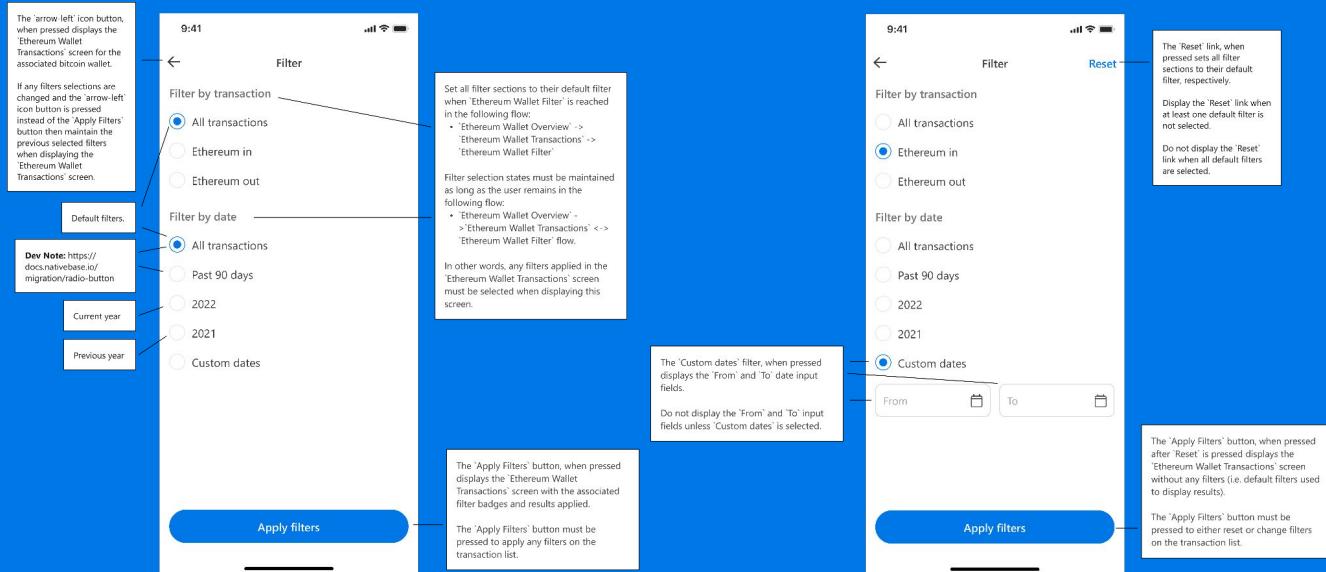
# Home / Wallets / Ethereum Wallet Transactions / Filter

## User Stories:

- CRYP-29: As a User, I want to filter my Ethereum transactions list, so I can search for specific transaction characteristics

## Personas:

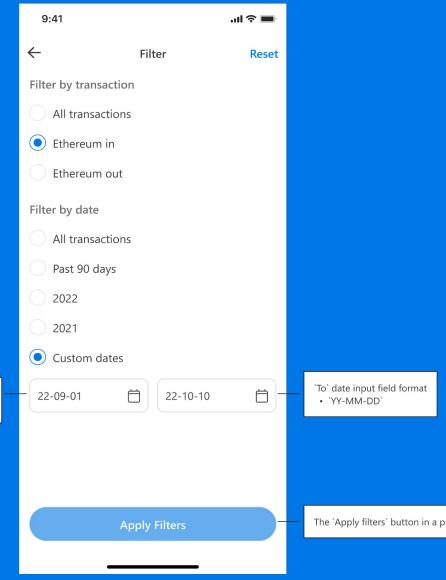
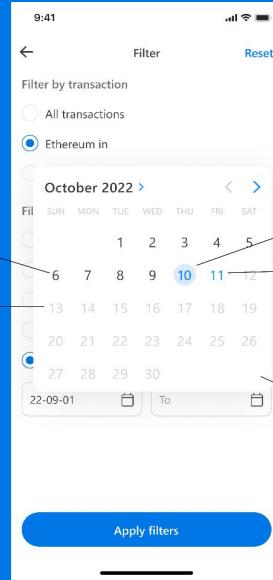
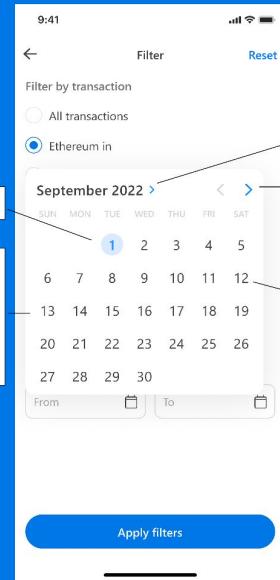
- User
  - The Beginner
  - The Intermediate
  - The Expert



"Good to go for coding."

- Stakeholder

# Home / Wallets / Ethereum Wallet Transactions / Filter cont.



# Home / Wallets / Ethereum Wallet Transactions / Filter cont.

The image consists of three screenshots of a mobile application interface for Ethereum wallet transactions, arranged horizontally. Each screenshot shows a list of transactions with various filters applied and annotations explaining the UI elements.

**Screenshot 1: Applied Filters**

Annotations:

- 'Filter by transaction -> Ethereum in' filter applied and associated badge
- The 'circle-xmark' icon button, when pressed removes the associated filter from the transaction list. As such, the transaction list must display the transaction list without said filter applied.
- Dev Note: <https://docs.nativebase.io/react/migration/badge&page-title>
- Filter by date -> Custom dates' filter applied and associated badge
  - Format: 'From' - 'To'
- Badge area is horizontally scrollable when there is not enough space to display all the badges (i.e. go past the device screen).

**Screenshot 2: No Results**

Annotations:

- 'Filter by transaction -> Ethereum out' filter applied and associated badge
- October 2022
- 0xb716...zakk Oct 8, 2022 • 12:15 PM +90 ETH
- 0xb219...zaop Sep 28, 2022 • 3:00 AM +9 ETH
- 0xb519...zajg Sep 10, 2022 • 5:30 PM +0.230021200000009123 ETH
- 0xb876...zasz Sep 2, 2022 • 3:35 AM +4,000 ETH

**Screenshot 3: Choose Another Filter**

Annotations:

- 'magnifying-glass' icon
- Display when there aren't any transactions matching your filters.
- Choose another filter
- The 'Choose another filter' link, when pressed displayed the Bitcoin Wallet Filter screen.

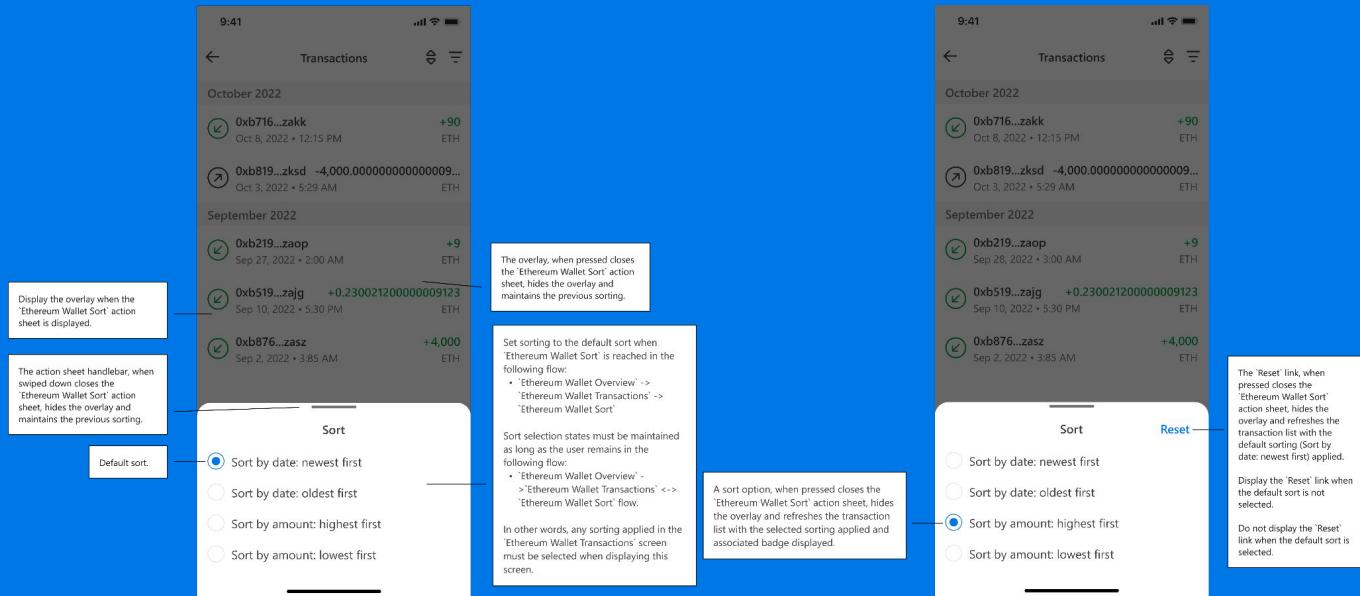
# Home / Wallets / Ethereum Wallet Transactions / Sort

## User Stories:

- CRYP-31: As a User, I want to sort my Ethereum transactions list, so I can order my transactions differently

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert



"Good to go for coding."

- Stakeholder

# Home / Wallets / Ethereum Wallet Transactions / Sort cont.

The image displays three screenshots of a mobile application interface for managing Ethereum wallet transactions. The interface includes a header with navigation icons, a title bar, and a main content area for displaying transaction history.

**Left Screenshot (Sort by Date: Oldest First):**

- Header: "Transactions" with sorting options: "Date: oldest first" and "Ethereum in".
- Content: Transaction list for September 2022 and October 2022. Each transaction row shows the address, timestamp, amount (+4,000 ETH, +900 ETH, etc.), and currency (ETH).
- Annotations:
  - "Sort by date: oldest first" sorting applied and associated badge.
  - Order the transactions by transaction date from oldest to newest.
  - The 'circle-mark' icon button, when pressed removes the associated filter from the transactions list. As such, the transaction list must display the transaction list without said filter applied.
  - Dev Note: <https://docs.nativebase.io/react/migration/badge#page-title>

**Middle Screenshot (Sort by Amount: Highest First):**

- Header: "Transactions" with sorting options: "Amount: highest first".
- Content: Transaction list showing the top five largest transactions. The first transaction is for +4,000 ETH.
- Annotations:
  - "Sort by amount: highest first" sorting applied and associated badge.
  - Order the transactions by largest transaction amount to smallest.

**Right Screenshot (Sort by Amount: Lowest First):**

- Header: "Transactions" with sorting options: "Amount: lowest first".
- Content: Transaction list showing the bottom five smallest transactions. The last transaction is for -4,000 ETH.
- Annotations:
  - "Sort by amount: lowest first" sorting applied and associated badge.
  - Order the transactions by smallest transaction amount to largest.

# Home / Wallets / Ethereum Transaction Details

## User Stories:

- CRYP-30: As a User, I want to view a details page for each Ethereum transaction, so I can get more information about it

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

The 'arrow-left' icon button, when pressed either displays the 'Ethereum Wallet Overview' or Ethereum Wallet Transaction screen depending on which screen the 'Ethereum Transaction Details' started from.

(From) Wallet address format:  
• <first 6 digits><ellipsis><last 4 digits>

Transaction hash

9:41 Oct 10, 2022 • 5:30 PM

Transaction date & time

+000,000.230021200000009123 ETH

0xb519...zajg

Transaction ID  
0x0894a2e3488da4d601bdee1c49f34752eb  
c1bc1adc6504e250eb828dc539678d

Status  
Confirmed

From  
0xb519f5ea0ba39494ce839613ffba742795  
7zajg

To  
Ethereum Chequing

Transaction Fee

Gas Price  
0.000000017704372255 ETH

Gas Limit  
100,000

Other Details

Block Number  
15

Position in Block  
19

Nonce  
109,764

The 'circle-arrow-down-left' icon, represents a received transaction.

The 'copy' icon button, when pressed displays the 'Transaction ID copied to clipboard' toast.

Dev Note: <https://docs.ethereum.org/need/migration/badge#page-title>

The 'copy' icon button, when pressed displays the 'Address copied to clipboard' toast.

9:41 Oct 3, 2022 • 5:29 AM

-4,000.000000000000000000009123 ETH

0xb819...zksd

Transaction ID  
0x9e62ca314b06720c3a0155a42983ec853b  
a472cc1d81b9071759f10d28af1d38

Status  
Confirmed

From  
Ethereum Chequing

To  
0xb519f5ea0ba39494ce839613ffba742795  
7zajg

Transaction Fee

Gas Price  
0.000000018917215377 ETH

Gas Limit  
730,010

Other Details

Block Number  
15,736,513

Position in Block  
15

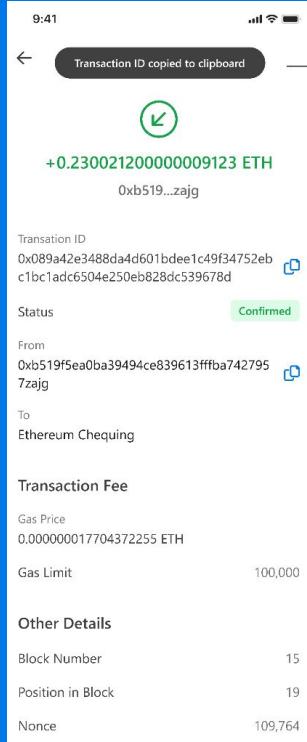
Nonce  
78

The 'copy' icon button, when pressed displays the 'Address copied to clipboard' toast.

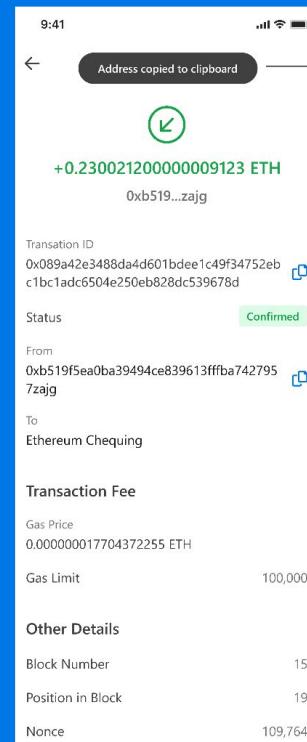
"Good to go for coding."

- Stakeholder

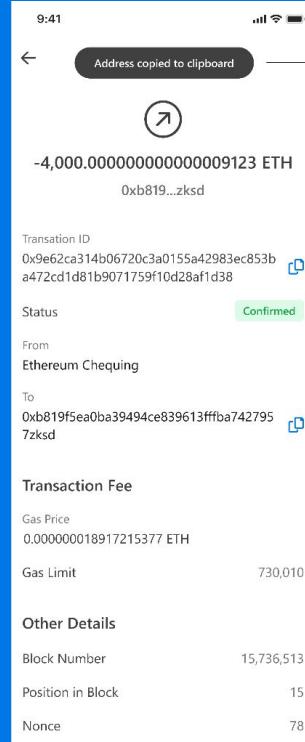
## Home / Wallets / Ethereum Transaction Details cont.



The 'Transaction ID copied to clipboard' toast displays when the 'copy' icon button associated to 'Transaction ID' is pressed. The toast displays for 2 seconds and then disappears. Do not use a transition animation.



The Address copied to clipboard toast displays when the 'copy' icon button associated to 'From' is pressed. The toast displays for 2 seconds and then disappears. Do not use a transition animation.



**Address copied to clipboard**

The 'Address copied to clipboard' toast displays when the 'copy' icon button associated to 'From' is pressed. The toast displays for 2 seconds and then disappears. Do not use a transition animation.

# UI Mockups

## Backlog

# Home / Wallets / Bitcoin Wallet Overview

## User Stories:

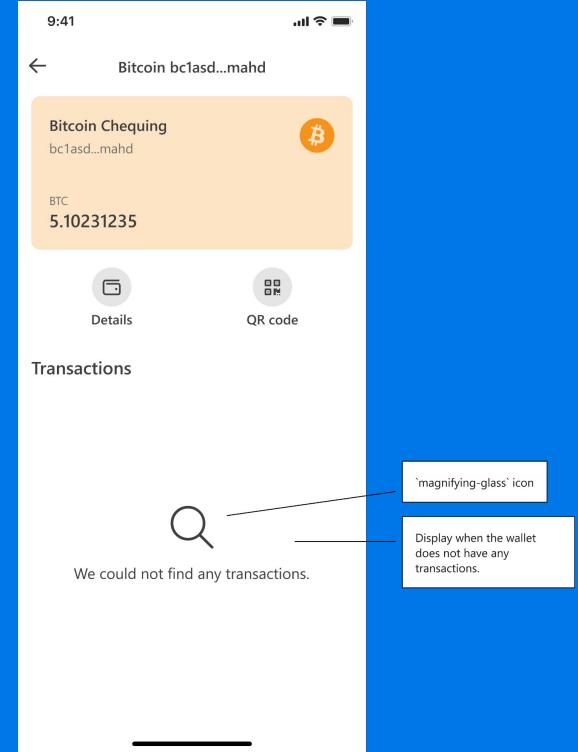
- CRYP-160: As a User, I want to view a list of all transactions associated to each of my Bitcoin wallets, so that I can see who I have transacted with
- CRYP-168: As a User, I want to share my Bitcoin wallet address, so that others can easily send me crypto without inputting my wallet address incorrectly
- CRYP-169: As a User, I want to view the details for each of my Bitcoin wallets, so that I can see all associated information available on the blockchain

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

"The wallet looks good."

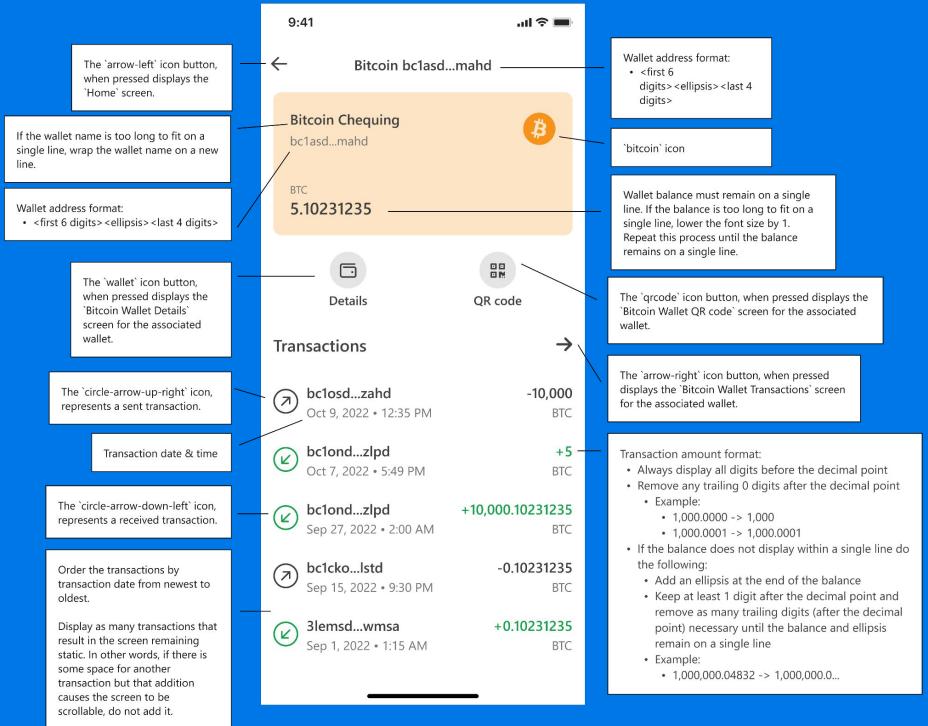
- Stakeholder



# Home / Wallets / Bitcoin Wallet Overview cont.

## Transactions

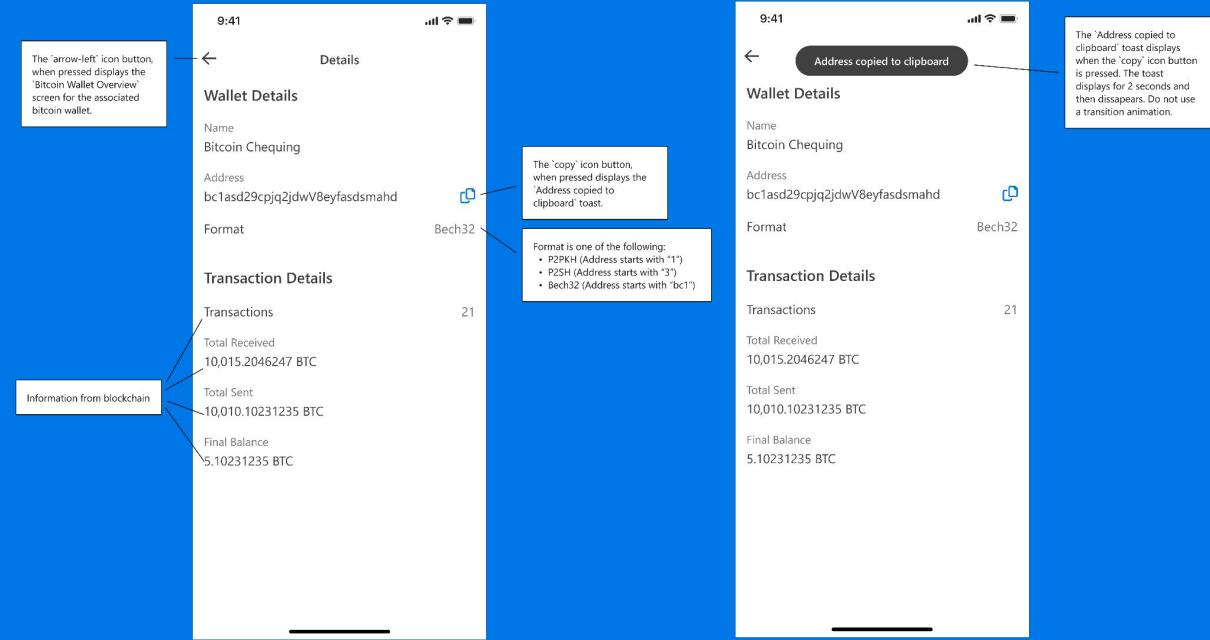
- Same requirements as Ethereum
- Similar challenges
  - 19+ million Bitcoin in circulation (out of 21 million)
  - Bitcoin has 8 decimal places
  - Rest of the challenges are the same as Ethereum



# Home / Wallets / Bitcoin Wallet Details

## User Stories:

- CRYP-168: As a User, I want to share my Bitcoin wallet address, so that others can easily send me crypto without inputting my wallet address incorrectly
- CRYP-169: As a User, I want to view the details for each of my Bitcoin wallets, so that I can see all associated information available on the blockchain



## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

"God to go for coding."

- Stakeholder

# Home / Wallets / Bitcoin Wallet QR Code

## User Stories:

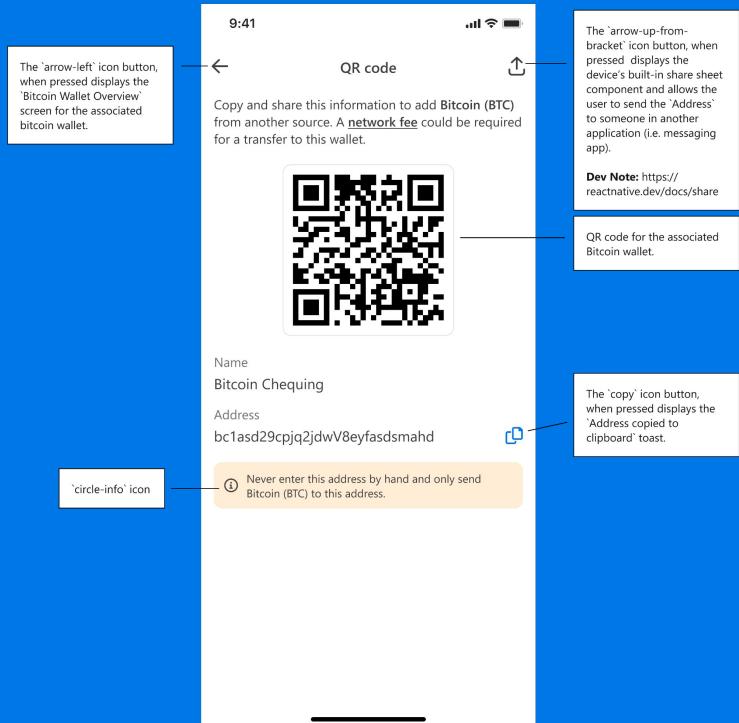
- CRYP-168: As a User, I want to share my Bitcoin wallet address, so that others can easily send me crypto without inputting my wallet address incorrectly

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

"God to go for coding."

- Stakeholder

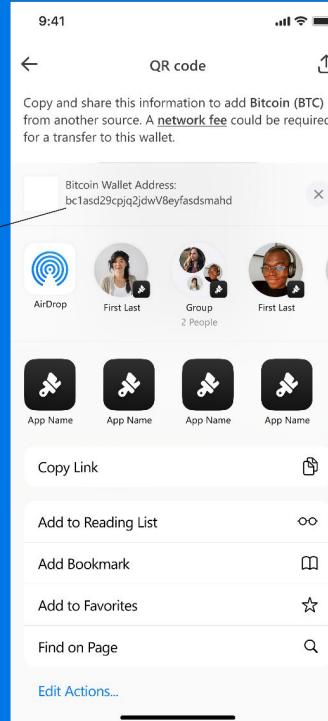


# Home / Wallets / Bitcoin Wallet QR Code cont.



The 'Address copied to clipboard' toast displays when the 'copy' icon button is pressed. The toast displays for 2 seconds and then disappears. Do not use a transition animation.

Message format:  
• Bitcoin Wallet Address:  
\\n<Address>



The device's built-in share sheet component is displayed when the 'arrow-up-from-bracket' icon button is pressed.

**Dev Note:** <https://reactnative.dev/docs/share>

# Home / Wallets / Bitcoin Wallet Transactions

## User Stories:

- CRYP-160: As a User, I want to view a list of all transactions associated to each of my Bitcoin wallets, so that I can see who I have transacted with

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert

The 'arrow-left' icon button, when pressed displays the 'Bitcoin Wallet Overview' screen for the associated bitcoin wallet.

On initial load (i.e. no filters and sorting):

- Group transactions by month and year, and order these groups from newest to oldest.
- In each month and year group, order the transactions by transaction date from newest to oldest.

The transaction list is an endless scroll.  
Transactions should all be loaded and inserted into the screen on initial display (i.e. dynamic transaction loading is not required).

The 'bars-filter' icon button, when pressed displays the 'Bitcoin Wallet Filter' screen.

The 'sort' icon button, when pressed displays the 'Bitcoin Wallet Sort' action sheet.  
**Dev Note:** <https://docs.nativebase.io/3.1.x/action-sheet>

Every time this screen is reached from the 'Bitcoin Wallet Overview' screen, display the transaction list using the default filters and sorting

- Filter by transaction: all transactions
- Filter by date: all transactions
- Sort by date: newest first

When scrolling through the transaction list, stick the header of the associated section currently being scrolled through under the top navigation bar. As scrolling continues, replace the stuck section header with the header associated to the next section.  
**Dev Note:** <https://docs-v2.nativebase.io/docs/examples/FlatListExample.html>

Do not display any filter and sorting badges when a section header is stuck underneath the top navigation bar.

When beginning to scroll, all elements and styling between the top navigation bar and first section must collapse and hide until the section header is stuck underneath the top navigation bar.  
All elements and styling must return to their original position when returning to the top of the screen (i.e. initial page load state).

Date	Transaction ID	Time	Change	Currency
Oct 9, 2022	bc1osd...zahd	12:35 PM	-10,000	BTC
Oct 7, 2022	bc1ond...zlpd	5:49 PM	+5	BTC
Sep 27, 2022	bc1ond...zlpd	2:00 AM	+10,000.10231235	BTC
Sep 15, 2022	bc1cko...lstd	9:30 PM	-0.10231235	BTC
Sep 1, 2022	3lemsd...wmsa	1:15 AM	+0.10231235	BTC
Aug 9, 2022	3lpmsd...wmsa	11:01 PM	-10	BTC
Aug 9, 2022	3lqmsd...wmsa	8:01 PM	+10	BTC
Oct 9, 2022	bc1ond...zlpd	5:49 PM	+5	BTC
Sep 27, 2022	bc1cko...lstd	2:00 AM	-0.10231235	BTC
Sep 1, 2022	3lemsd...wmsa	1:15 AM	+0.10231235	BTC
Aug 9, 2022	3lpmsd...wmsa	11:01 PM	-10	BTC
Aug 9, 2022	3lqmsd...wmsa	8:01 PM	+10	BTC

"Good to go for coding."

- Stakeholder

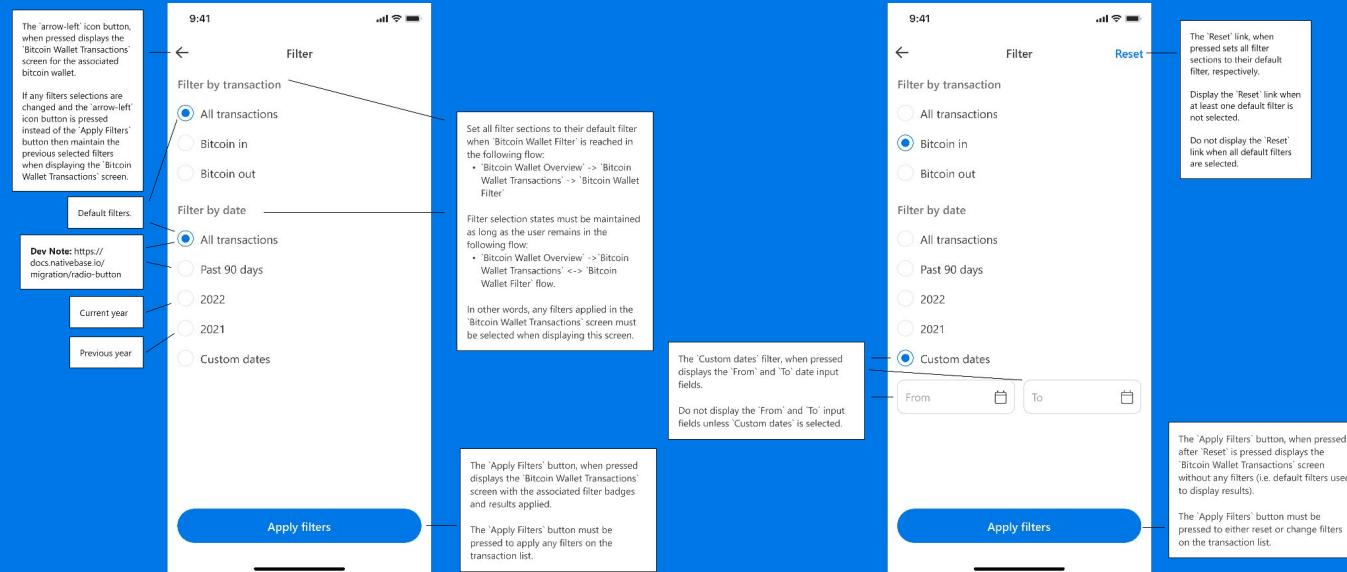
# Home / Wallets / Bitcoin Wallet Transactions / Filter

## User Stories:

- CRYP-164: As a User, I want to filter my Bitcoin transactions list, so I can search for specific transaction characteristics

## Personas:

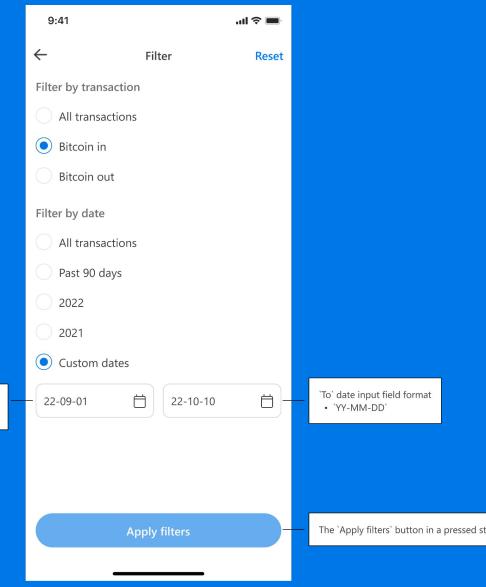
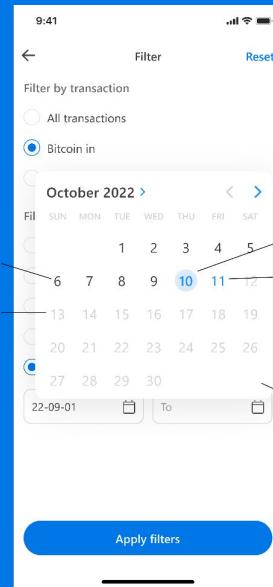
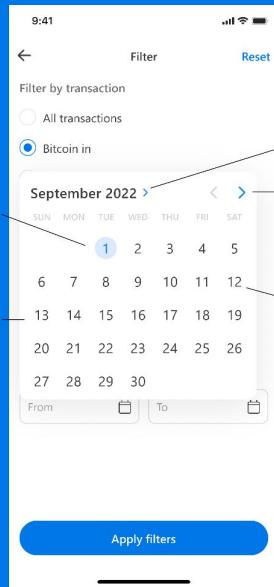
- User
  - The Beginner
  - The Intermediate
  - The Expert



"Good to go for coding."

- Stakeholder

# Home / Wallets / Bitcoin Wallet Transactions / Filter cont.



# Home / Wallets / Bitcoin Wallet Transactions / Filter cont.

The image displays three screenshots of a mobile application's 'Transactions' screen, illustrating various filtering options and their results.

**Screenshot 1: Bitcoin in filter applied**

This screenshot shows transactions from September and October 2022. A 'Bitcoin in' filter badge is applied, and a date range filter 'Sep 1, 2022 - Oct 10, 2022' is also applied. The transaction list includes:

- bc1ond...zlpd (Oct 7, 2022) +5 BTC
- bc1ond...zlpd (Sep 27, 2022) +10,000.10231235 BTC
- 3lemsd...wmsa (Sep 1, 2022) +0.10231235 BTC

**Screenshot 2: Bitcoin out filter applied**

This screenshot shows transactions from August, September, and October 2022. A 'Bitcoin out' filter badge is applied. The transaction list includes:

- bc1osd...zahd (Oct 9, 2022) -10,000 BTC
- bc1cko...lstd (Sep 15, 2022) -0.10231235 BTC
- 3lpmsd...wmsa (Aug 9, 2022) -10 BTC

**Screenshot 3: No results found**

This screenshot shows a search result where no transactions match the filters applied ('Bitcoin in' and date range 'Jan 1, 2022 - Feb 1, 2022'). The message 'We could not find any transactions matching your filters.' is displayed, along with a magnifying glass icon and a 'Choose another filter' link.

**Annotations and Notes:**

- 'Filter by transaction -> Bitcoin in' filter applied and associated badge
- The 'circle-xmark' icon button, when pressed removes the associated filter from the transaction list. As such, the transaction list must display the transaction list without said filter applied.
- Dev Note: <https://docs.nativebase.io/next/migration/badge#page-title>
- 'Filter by date -> Custom dates' filter applied and associated badge
  - Format: 'from' - 'to'
- 'Filter by transaction -> Bitcoin out' filter applied and associated badge
- Badge area is horizontally scrollable when there is not enough space to display all the badges (i.e. go past the device screen).
- Badge order is as follows: Sorting first, filters second
- 'magnifying glass' icon
- Display when there aren't any transactions that match the filters applied.
- The 'Choose another filter' link, when pressed displayed the 'Bitcoin Wallet Filter' screen.

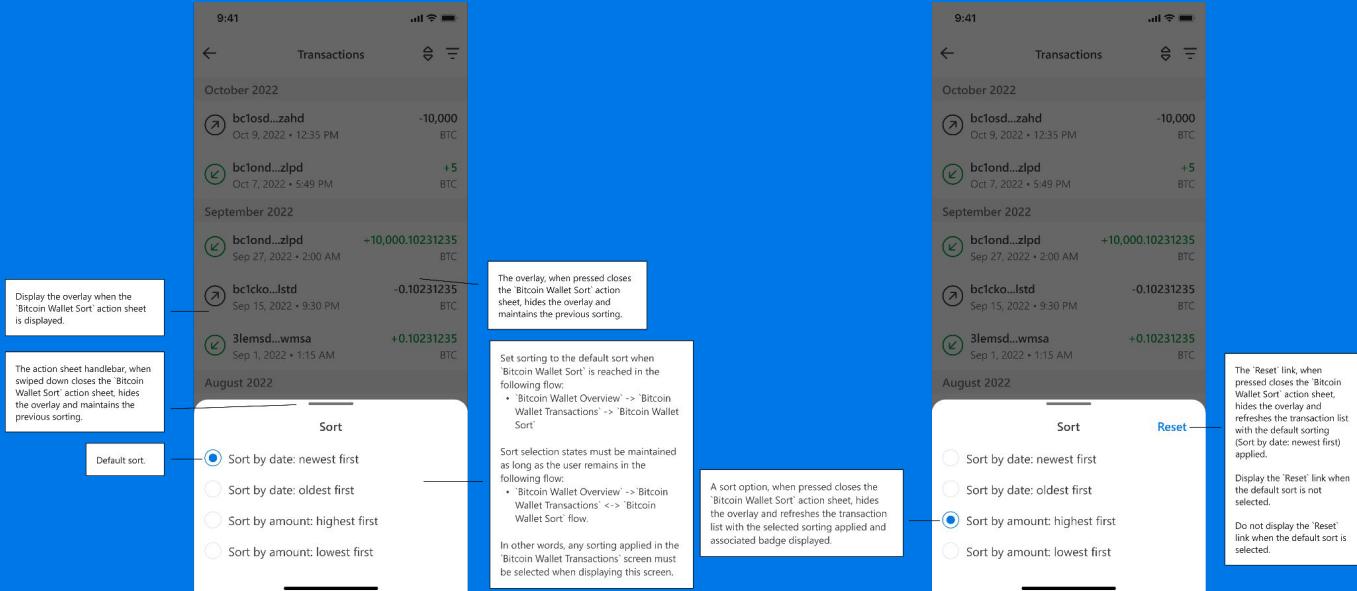
# Home / Wallets / Bitcoin Wallet Transactions / Sort

## User Stories:

- CRYP-166: As a User, I want to sort my Bitcoin transactions list, so I can order my transactions differently

## Personas:

- User
  - The Beginner
  - The Intermediate
  - The Expert



"Good to go for coding."

- Stakeholder

# Home / Wallets / Bitcoin Wallet Transactions / Sort cont.

The image displays three screenshots of a mobile application interface for managing Bitcoin transactions. Each screenshot shows a list of transactions with various filters and sorting applied.

**Screenshot 1: Date: oldest first**

'Sort by date: oldest first' sorting applied and associated badge  
• Order the transactions by transaction date from oldest to newest.

The 'circle-xmark' icon button, when pressed removes the associated filter from the transaction list. As such, the transaction list must display the transaction list without said filter applied.

**Dev Note:** <https://docs.nativebase.io/react/migration/badge/page-title>

**Badge area is horizontally scrollable when there is not enough space to display all the badges (i.e. go past the device screen).**

**Badge order is as follows: Sorting first, filters second.**

Date	Address	Time	Amount	Currency
August 2022	3lqmsd...wmsa	Aug 9, 2022 • 8:01 PM	+10	BTC
September 2022	3lemsd...wmsa	Sep 1, 2022 • 1:15 AM	+0.10231235	BTC
September 2022	bc1ond...zlpd	Sep 27, 2022 • 2:00 AM	+10,000.10231235	BTC
October 2022	bc1ond...zlpd	Oct 7, 2022 • 5:49 PM	+5	BTC

**Screenshot 2: Amount: highest first**

'Sort by amount: highest first' sorting applied and associated badge  
• Order the transactions by largest transaction amount to smallest.

Address	Date	Time	Amount	Currency
bc1ond...zlpd	Sep 27, 2022	2:00 AM	+10,000.10231235	BTC
3lqmsd...wmsa	Aug 9, 2022	8:01 PM	+10	BTC
bc1ond...zlpd	Oct 7, 2022	5:49 PM	+5	BTC
3lemsd...wmsa	Sep 1, 2022	1:15 AM	+0.10231235	BTC
bc1cko...lstd	Sep 15, 2022	9:30 PM	-0.10231235	BTC
3lpmzd...wmsa	Aug 9, 2022	11:01 PM	-10	BTC
bc1osd...zahd	Oct 9, 2022	12:35 PM	-10,000	BTC

**Screenshot 3: Amount: lowest first**

'Sort by amount: lowest first' sorting applied and associated badge  
• Order the transactions by smallest transaction amount to largest.

Address	Date	Time	Amount	Currency
bc1osd...zahd	Oct 9, 2022	12:35 PM	-10,000	BTC
3lpmzd...wmsa	Aug 9, 2022	8:01 PM	-10	BTC
bc1cko...lstd	Sep 15, 2022	9:30 PM	-0.10231235	BTC
3lemsd...wmsa	Sep 1, 2022	1:15 AM	+0.10231235	BTC
bc1ond...zlpd	Oct 7, 2022	5:49 PM	+5	BTC
3lqmsd...wmsa	Aug 9, 2022	8:01 PM	+10	BTC
bc1ond...zlpd	Sep 27, 2022	2:00 AM	+10,000.10231235	BTC

# Testing Infrastructure and QA Tools

# 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

# Measuring Code Quality

- We will be running SonarQube in order to measure various quality attributes of our system and services
  - Cyclomatic Complexity
  - Cognitive Complexity
  - Duplicated blocks, Duplicated files, Duplicated lines
  - Code Smells
  - Maintainability Rating
  - Vulnerabilities
  - Security Hotspots
  - Classes, Comment lines, Lines of code, Functions, Statements
  - Coverage, Condition coverage, Line coverage

# Demo Time!

# Questions?