Squirtle Voucher Vault

Blockchain Voucher Platform

Agenda

01

The Problem. Why Blockchain?

This system has been designed before, why does implementing this on the Blockchain provide an improved and worthwhile solution?

02

System Overview

We have designed a secure voucher management system, which includes a user-friendly front-end interface for seamless interaction. We will provide an overview of the technical stack and architecture.

03

Demo

A user-centric walkthrough of the application's key features and functionality. Showcasing the application's core capabilities in addressing the identified business requirements.

01

The Problem Case

Why Blockchain?

“Impact.com often connects 3 parties together, who do not necessarily know each other and have no formal relationship, but need to do trusted business with each other.”

What is a blockchain?

Imagine you and your friends are playing a game where you keep track of everyone's scores. Instead of writing the scores on a piece of paper, you decide to write them in a special notebook that everyone can see. This notebook is special because once you write something in it, you can't erase it or change it.

Every time someone scores points, you write it in the notebook, and then everyone has to agree that the score is correct. After everyone agrees, you add that score to the notebook and close that page forever. The next time someone scores, you open a new page and do the same thing.

This special notebook is kind of like a blockchain. It's a way of keeping track of things (like scores, or money, or anything important) in a way that everyone can see, no one can change, and everyone has to agree on. It's safe and trustworthy because everyone can check the notebook to make sure everything is right.

What Are We?

Voucher Vault is a platform that uses **blockchain technology** to create, store, and manage digital vouchers securely and efficiently.

To define who we are, let’s start of with a scenario to paint a picture:
You know how gift cards or coupons work, right? They are like special tickets that you can use to buy things or get discounts. Sometimes, people can lose them, they can get stolen, or someone might try to make fake ones.

Here’s where we come in:

- **Secure Storage:** Voucher Vault stores these vouchers digitally on a blockchain, which is like a super-secure and unchangeable digital notebook that everyone agrees is correct. This means your vouchers can't be lost or tampered with.
- **Easy Verification:** Because the blockchain records every detail about the voucher, stores and people can easily check to make sure the voucher is real and see its history. This helps prevent fraud and makes everyone trust the system more.
- **Quick and Convenient:** You can access your vouchers anytime using a computer or smartphone, and use them easily without needing to carry physical copies.
- **Transparent Tracking:** You can see when and where your vouchers were issued and used, making it easy to keep track of everything.

This has been done before?

Blockchain and smart contracts are not just innovations; they're revolutions. They bring unparalleled security, transparency, and efficiency to your business. By adopting blockchain and smart contracts, you're not just keeping up with the future—you're leading it.

- **Unbreakable Security**
  Blockchain decentralized nature means that there's no single point of failure. Traditional systems can be hacked, altered, or manipulated by bad actors. But with blockchain, every transaction is recorded across multiple computers (nodes), making it nearly impossible to change or fake. Your data is safe, your transactions are secure, and your business is protected.
- **Cost-Effective**
  By eliminating middlemen and reducing the need for reconciliation, blockchain slashes operational costs. You're saving money while gaining security and speed—a win-win!
- **Automation Equals Efficiency**
  Smart contracts automatically execute when the conditions are met. No more waiting for a third party to review, approve, or process transactions. This reduces the time and effort needed to manage agreements, freeing you up to focus on growth.
- **Trustless Transactions**
  With smart contracts, trust is built into the system. You don't need to rely on the other party to fulfill their end of the deal—the contract enforces it automatically. This is especially powerful for international transactions or deals between strangers

02

System Overview

Voucher System, Tech Stack, User Experience

High Level Design

Back-End:
Solana Native
Rust

Front End:
React
Typescript
Solita

Development :
Git Flow

Voucher System (Gideon)

2 Instructions:

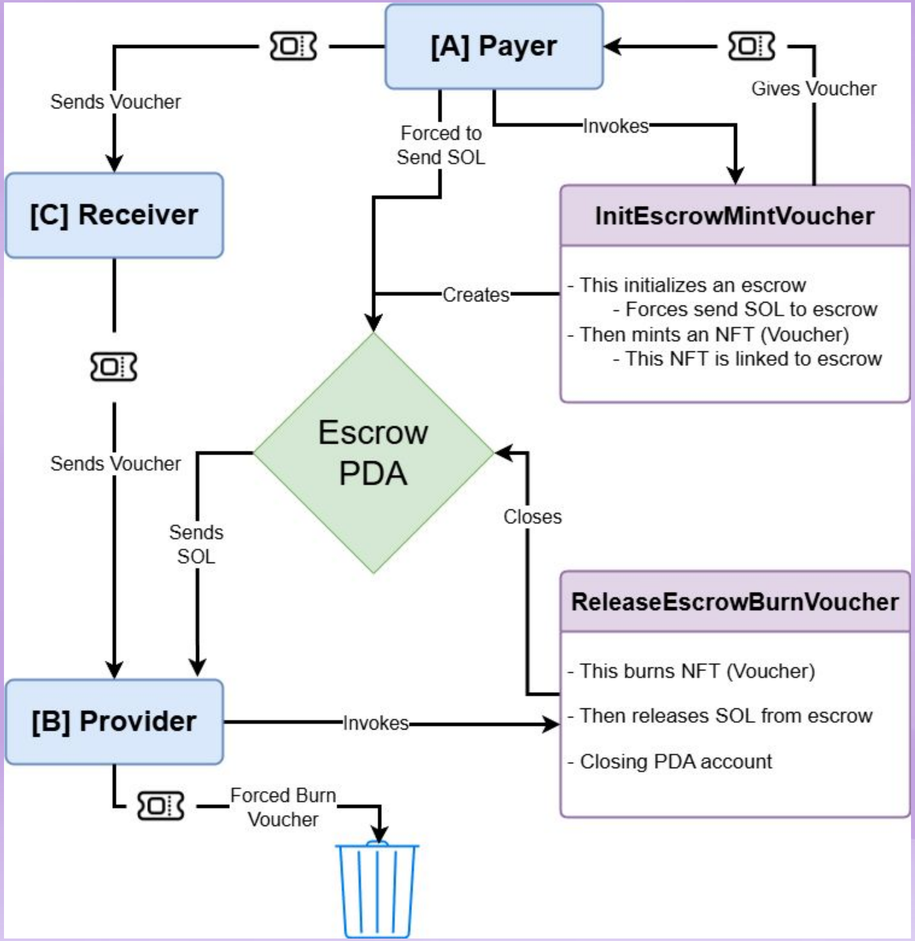
InitEcrowAndMintVoucher

- Initializes the escrow and deposits funds from the invoking party into it.
- Then mints the voucher and links it to the escrow.
- The escrow and voucher are both linked to the recipient so only they can redeem the voucher.

ReleaseEscrowAndBurnVoucher

- Releases the escrow funds to the recipient if the recipient link is valid.
- Then burns the voucher.

Voucher System (Gideon)



Voucher System Tech Stack

Solana Native - The Blockchain Interaction:

- Solana native was chosen as opposed to the Anchor framework since it allows more control over the solana program as we are not restricted by the rules of the framework. Solana native offers everything we need except an easy way to integrate with the frontend, and that is where shank comes in.

Shank - The IDL Creator:

- Shank allows us to declare the arguments and accounts needed by each instruction in a structured manner. This declaration is then used to generate an IDL which is used by solita on our frontend to generate a TypeScript SDK for easily invoking the solana program.

Front End (Viper)

3 Users:

- the payer (manufacturer)
- the provider (issuer)
- and the receiver (influencer).

The interface is crafted to be dynamic and intuitive, allowing users to navigate and interact with the platform effortlessly. To meet the unique needs of each role, we created three separate dashboards:

Payer Dashboard:

- Allows issuers to create vouchers and determine what receiver can obtain voucher

Provider Dashboard:

- Facilitates voucher validation and processing.

- Ensures smooth handling of voucher claims.

Receiver Dashboard:

- Enables easy management and redemption of vouchers.

Front End (Viper)

To meet the unique needs of each role, we created three separate dashboards:

Payer	Create vouchers, transfer vouchers to influencers.
Provider	Voucher validation and processing, claim escrow money
Receiver	Manage vouchers; and redeem vouchers by transfer.

Front End Tech Stack

Our front-end is built to be powerful, reliable, and user-friendly

Next.js - The Performance Booster:

- The power of Next.js. It optimizes our platform for speed and performance, ensuring fast load times and smooth navigation.

React - The interface:

- React powers the user experience, allowing us to build modular, responsive components that work together seamlessly.

Typescript - the safety net:

- TypeScript acts like a safety feature, catching errors in our code early and ensuring everything runs smoothly and reliably.

Solita - The blockchain bridge:

- Solita connects our front-end to the blockchain, translating complex blockchain interactions into simple, user-friendly actions.

MUI - The design framework:

- MUI gives our platform a polished, professional look. It provides pre-built, customizable components that make our interface both attractive and user-friendly.

Interacting with the Blockchain: Methods and Techniques

We developed our own **custom library** to interact with the blockchain from our platform, enabling the fetching and decoding of essential token data.

Connecting to the Blockchain:

Cluster Connection:

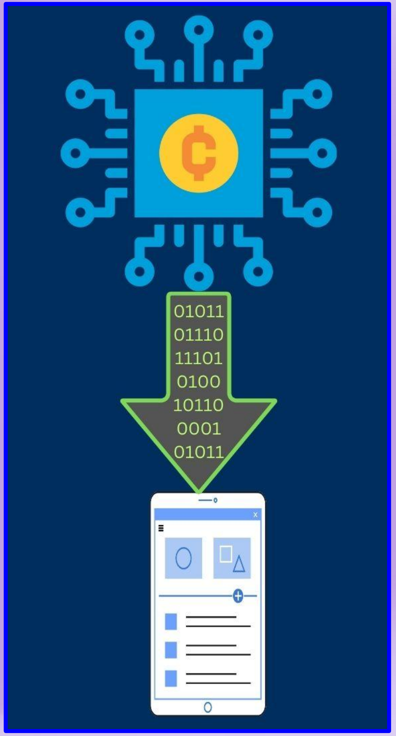
- Utilizing *clusterApiURL* to connect our application to the Solana blockchain.
- Targeting the specific NFT token program ID to identify relevant tokens accounts.

Wallet and Token Address Retrieval:

- Querying a user's wallet public key to identify associated token addresses.
- Retrieving the mint addresses for tokens (vouchers) from these token addresses.

Decoding Data and Accessing Off-Chain Information

After retrieving token data from the blockchain, we decoded the raw data and accessed additional information stored off-chain to fully understand and utilize the vouchers.



Data Decoding:

- Converting raw binary data from the blockchain into meaningful information.
- Extracting key elements such as token symbol, name, escrow address, and URI from the mint addresses.

Accessing Off-Chain Data:

- Using the URI to access a JSON file that stores additional off-chain data.
- Parsing metadata stored in the URI, such as the token image via IPFS links referenced in the JSON file.

Optimizing Data Storage:

- Only essential data is stored on-chain and off-chain data is referenced by the URI, efficiently managing storage without overloading the blockchain.

03

Live Demo

User-centric walkthrough

04

SE Principles

Agile Software Engineering, SCRUM

Agile Manifesto

Individuals

and interactions over processes and tools

Working Software

over comprehensive documentation

Collaboration

over contract negotiation

Respond to Change

over following a plan

Sub-Team Approach

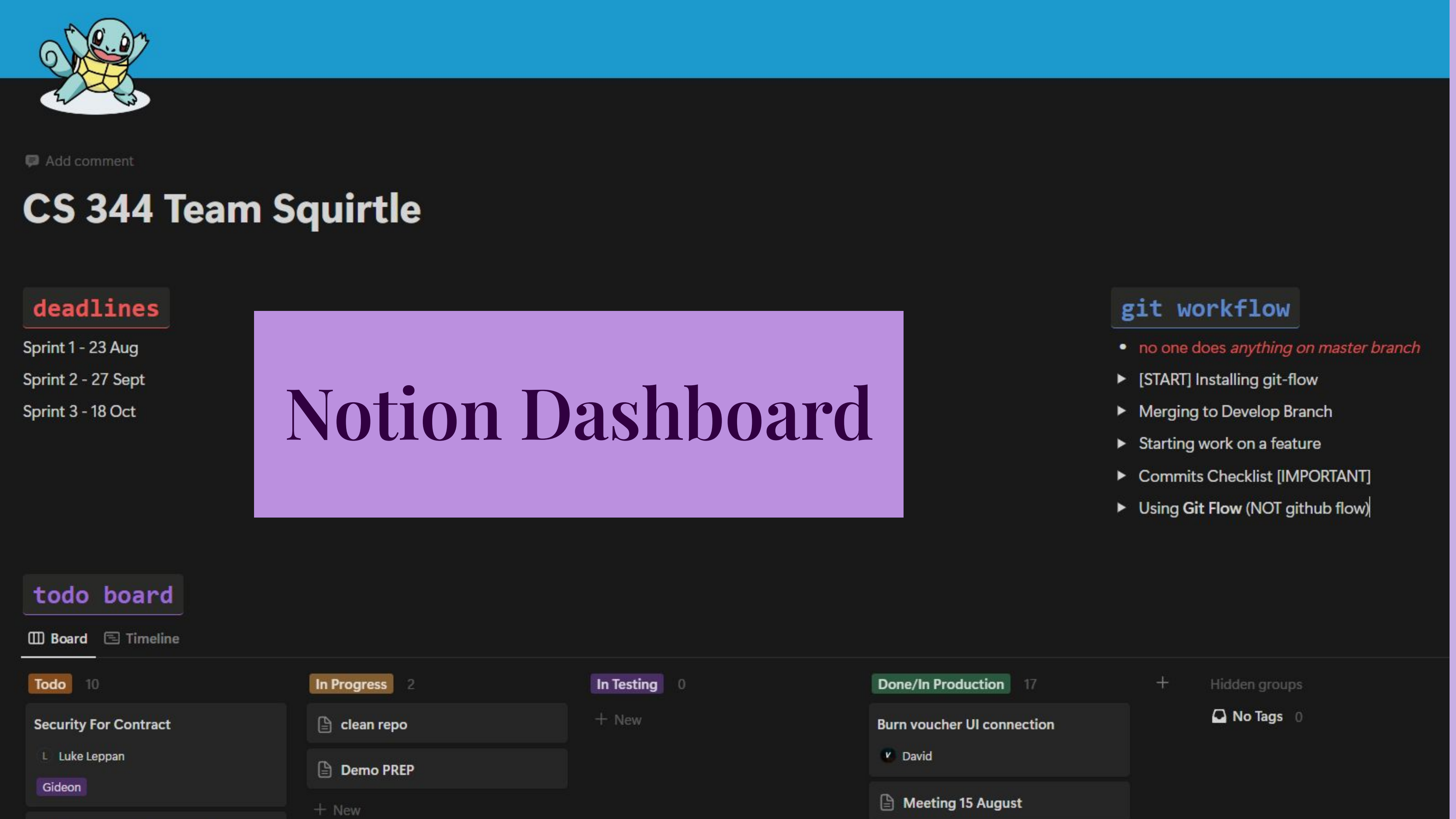
Viper - Front End

Yedi
Caeden
Izak

Gideon - Back End

Luke
David

Notion Dashboard



Add comment

CS 344 Team Squirtle

deadlines

Sprint 1 - 23 Aug

Sprint 2 - 27 Sept

Sprint 3 - 18 Oct

git workflow

- no one does *anything on master branch*
- ▶ [START] Installing git-flow
- ▶ Merging to Develop Branch
- ▶ Starting work on a feature
- ▶ Commits Checklist [IMPORTANT]
- ▶ Using **Git Flow** (NOT github flow)

todo board

Board | Timeline

Todo 10	In Progress 2	In Testing 0	Done/In Production 17	Hidden groups
Security For Contract Luke Leppan Gideon	clean repo	+ New	Burn voucher UI connection David	No Tags 0
	Demo PREP		Meeting 15 August	
	+ New			

Sprint Breakdown

Blockchain voucher system, basic UI

Sprint 1

Sprint 2

Transferring through QR codes, transaction history,

Admin system, stats tracking and visualization, UX

Sprint 3

"The problem with troubleshooting is that trouble shoots back."

—Unknown