Team 9

Kai Widman, 2953974
Jonathan Downs, 2927444
Tristan Gant, 2846312
Lucas Brakenridge, 2988068
Javier Barea, 3001667

Project Name: BlockPoll

Project Synopsis: A real-time, public-facing, block-chain solution for casual polling scenarios.

Project Description

A relatively new technology has begun picking up steam, blockchain. It's something almost every tech savvy person has heard of, but few properly understand. The most familiar use for blockchain is of course cryptocurrencies like Bitcoin and Ethereum. However, there are many more wide reaching applications for blockchain. With this revolutionary technology on the horizon, our team wanted to build something that would expose us to the world of blockchain to better understand the details of these increasingly important systems. We have chosen to build a polling application that stores all of it's vital information on the blockchain. This application will be similar to online polling applications like strawpoll or google forms. It will exist to allow any user to create and share a poll. These polls could be for things like deciding where a group should eat dinner or voting for leadership for clubs.

The goal of this project isn't marketability or problem solving, it's primarily exploratory. However, there is some potential down the road to turn this into something with more serious applications. As either a stretch goal or for development after graduation, this project could serve as the baseline to a much more robust polling system. Depending on the security and scalability, it would be possible to try to create a product that could be used in legitimate elections of any size.

Project Milestones:

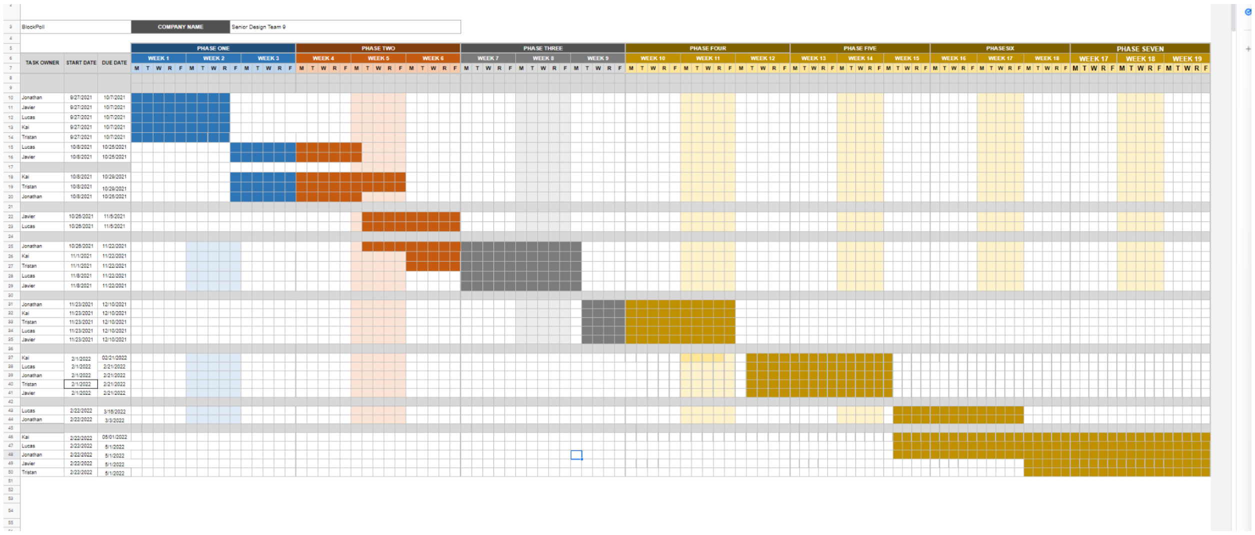
Semester one:

- September 27th - Initial project description
- October 8th - Reach out to Ripple/KU Blockchain Institute/Perry Alexander about funding/support/learning options available through KU donation
- October 29th - Finish researching blockchain technology and the XRPL infrastructure
- November 5th - Mock up designs for the products UI
- November 22nd - Complete first draft for front-end implementation
- December 10th - Finalize implementation planning and documentation of the full project stack

Semester two:

- February 4th - Begin MVP development

- March 4th - Integrate front-end and back-end into MVP
- March 11th - Handle double voting issues
- March 18th - Add user accounts
- April 1st - Finish extra features/stretch goals
- April 8th - Integrate all loose ends in the codebase
- April 15th - Polish and test finished project
- April 22nd - Resolve all outstanding bugs
- May 27th - Finalize project and present



BlockPoll | COMPANY NAME | Senior Design Team 9

TASK OWNER	START DATE	DUE DATE
Jonathan	9/27/2021	10/7/2021
Javier	9/27/2021	10/7/2021
Lucas	9/27/2021	10/7/2021
Kai	9/27/2021	10/7/2021
Tristan	9/27/2021	10/7/2021
Lucas	10/8/2021	10/25/2021
Javier	10/8/2021	10/25/2021
Kai	10/8/2021	10/29/2021
Tristan	10/8/2021	10/29/2021
Jonathan	10/8/2021	10/25/2021
Javier	10/26/2021	11/5/2021
Lucas	10/26/2021	11/5/2021
Jonathan	10/26/2021	11/22/2021
Kai	11/1/2021	11/22/2021
Tristan	11/1/2021	11/22/2021
Lucas	11/8/2021	11/22/2021
Javier	11/8/2021	11/22/2021
Jonathan	11/23/2021	12/10/2021
Kai	11/23/2021	12/10/2021
Tristan	11/23/2021	12/10/2021
Lucas	11/23/2021	12/10/2021
Javier	11/23/2021	12/10/2021
Kai	2/1/2022	02/21/2022
Lucas	2/1/2022	2/21/2022
Jonathan	2/1/2022	2/21/2022
Tristan	2/1/2022	2/21/2022
Javier	2/1/2022	2/21/2022
Lucas	2/22/2022	3/16/2022
Jonathan	2/22/2022	3/3/2022
Kai	2/22/2022	05/01/2022
Lucas	2/22/2022	5/1/2022
Jonathan	2/22/2022	5/1/2022
Javier	2/22/2022	5/1/2022
Tristan	2/22/2022	5/1/2022

https://docs.google.com/spreadsheets/d/1Su21BNsDHBZ108M-xBgJ9TmzNWfD4IE3MOwDRvuV_yU/edit?usp=sharing

Project Budget:

- **No** hardware requirements. We will be using XRPL as a blockchain framework, and this is open source (and thus free). No computing resources will be needed.
- Estimated Cost: $0
- Vendor: None
- Special Training: None
- When they will be required: N/A

Final Project Design:

We will be developing a polling web-application that allows users to create their own private/public customized polls and vote on the XRP blockchain. Voting on the blockchain for increased security and voter confidence. Our user experience will consist of 3 core actions, voting, creating polls, and viewing results.

Voting:

If a user wishes to vote in an active poll, they will need a link to that polling page. After navigating to the link in a browser, the user will then be met with a poll title, an optional description, and a list of questions and their choices. The user can then select their desired choices using a radio button or a checkbox depending on what setting the poll-maker has chosen. After they have submitted their votes by clicking on the respective submit button. However, if the user has already voted (which will be verified with the block-chain back-end), they will be notified that their second voting attempt was unsuccessful. Regardless of success, the user will then be directed to the results page for that poll.

Creating Polls:

When a poll organizer decides to create a poll, they can visit the main landing page of our site. From here, the user can fill out the information for the poll detailed in the “voting” action. The user can then choose to make the poll public or private. If the poll is private, the poll will not appear in any of the browsing sections, and it will require an additional auto-generated passphrase. After creating a poll, the user will be shown a link and if relevant, the passphrase. The user will also be able to customize their own polls and decide the format in which these will appear.

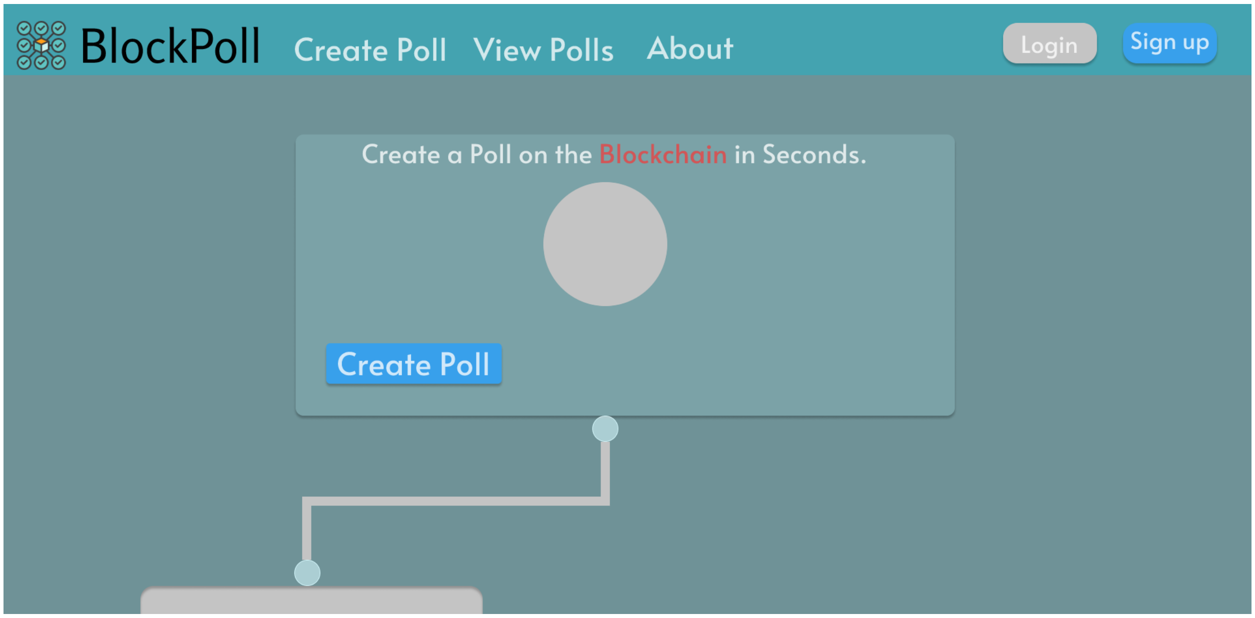
Viewing Results:

After a user has voted in a poll, they will be directed to this results view. This page will display the contents of the poll and list the values of votes cast for each option and question. This data will be displayed in a visual manner to improve the user experience.

Front End Design:

Our front-end is going to be a lightweight, web-based, reactive application. The future of applications is web, not hard-coded desktop apps, and one of the emergent technologies when it comes to web, is using a Reactive approach. This means our web app will be self-contained and with a more minimalist UX and UI approach than most other web apps. We plan on using several JS frameworks and libraries to build the front-end of the web app.

React is a pretty well-known front-end JS library that is used to build UI interfaces and components. The beauty about React is that it is open-source, it is very well maintained and has hundreds of open-source components readily available for us to build and deploy quick web apps. Apart from being lightweight and using minimal CPU resources, React is meant to be user-friendly, and avoid bloatware and the excess of information when it comes to using single webapps. Therefore our aim is for our UI and UX to be as straightforward as possible, meaning doing the most actions with the least quantity of clicks from the user.

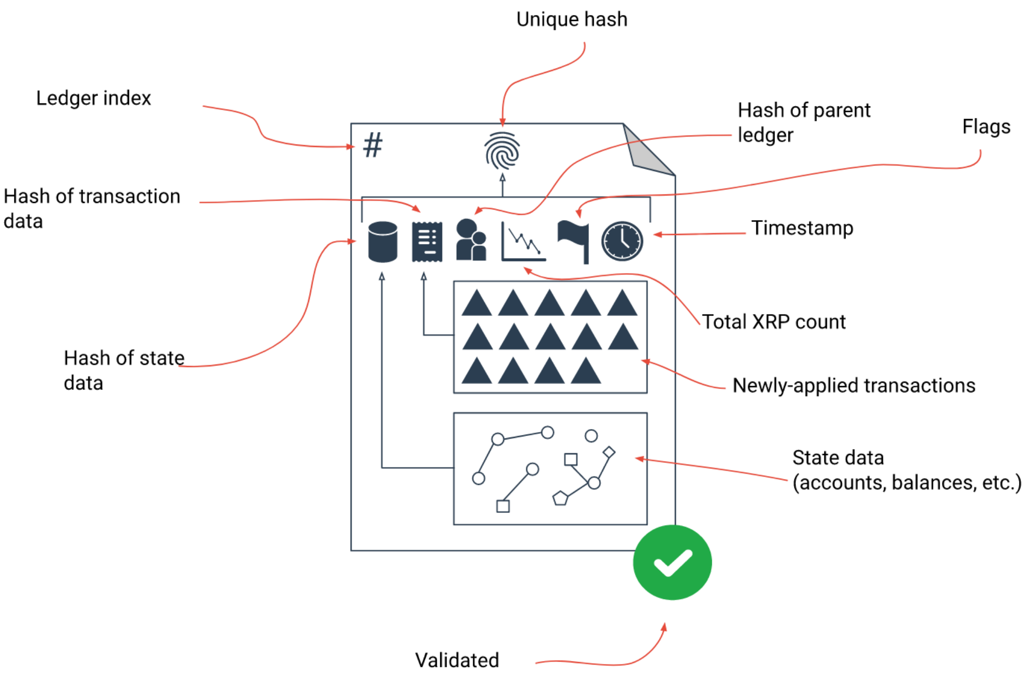


Snippet of the UI Mockup for the Home Page of our Web Application

<u>Back End Design:</u>

The backend will take poll votes from the front-end, given from the users voting on a particular poll via the React App. Those polls come in from the front-end and send them to an XRPL server. This server will update the blockchain for a particular poll with the new information (the polling results), and do any processing required to update the result. The front-end, when it requests the poll information again, will receive the updated result and will show the new information.

In order to post a vote to the system, the blockchain needs to be updated to reflect the new data. This is done by adding a new ledger to the ledger history. This ledger history is where all transactional blockchain data is stored. Below is a breakdown of the elements that compose a ledger.



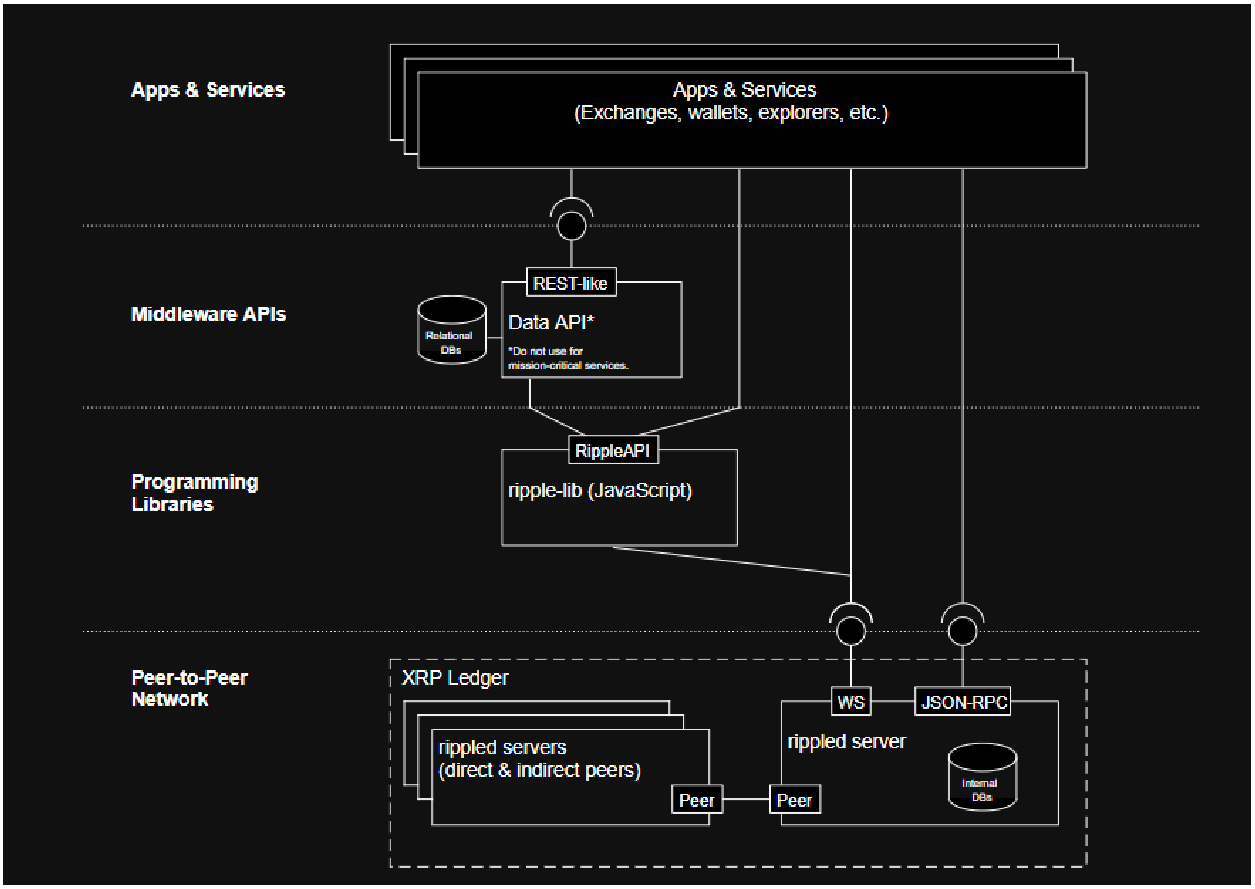
XRP Ledge Elements (Source: XRPL documentation)

Polling Process in App:

Our polling process will happen as follows: Users, upon entering the application, will enter a link if they wish to cast a vote to a particular poll. This link will then be sent to the back end, and the backend will search for this link, finding which poll the link corresponds to.This poll will then be processed for relevant information for the user. For instance, the user does not need the full poll blockchain to be sent to them, only information about what answers are present and what results have already been tallied.

Once this information has been sent back to the user, they will receive options to vote from. The user will select an option and then the relevant information (what option they selected, along with any credentials we need in order to verify the user and prevent any duplication) will be sent back to the server. The server will then update the poll's blockchain to report the change to the poll. At the same time, the information pulled from the poll on the server will be updated and the user will have "voted".

The poll itself will be stored as a collection of data structures on the server. There will be the poll blockchain, the poll link, a boolean determining whether it is enabled, the poll answers and their associated values (the amount of votes each one has). The poll answers and votes will provide an easy way to pull data from the server to the front-end. The poll link provides a way for users to access a particular poll. The boolean will be used to make sure no votes are cast on a closed poll.



Stack Levels of the XRP Ledger and how it communicates with our application (Source: XRPL documentation)

<u>Design Constraints:</u>

Since this project is self-directed, almost all the design constraints are self-imposed. The only design constraint that isn't ultimately decided by us is our time schedule for this project. In order to complete this course, we must, of course, meet our deliverable deadlines put in place by our professor. The only other non-technical constraint is the team we are working with. As of now, our teams are set in stone.

As for technical constraints, all of these are more flexible. Our goal is to implement a polling system on the blockchain. From here, we have to decide a front-end stack and a back-end stack. Our constraints are whatever stacks we choose.

As previously mentioned, for the front-end, we have chosen to use the Javascript web framework React. For the back-end is that we are using the XRPL framework. Our back-end will then of course need to be written in a language supported by XRPL framework. As of now, XRPL supports access via HTML, Javascript, Python, and Java. Our team is going to be using Python to implement the backend using the xrpl-py library.

Ethical Issues:

Privacy: Our project, inherently, has privacy as one of its main concerns, and at the same time, its main feature, given we are building a poll and voting system. We believe that privacy, especially in the Information Era, is and needs to be of utmost importance, that is why we are building this using the blockchain as the cornerstone of our system.

Voting and polls will have the ability to be exclusively private and anonymous, where the voter/user has complete anonymity and his vote is both secured, inmutable, and at the same time, anonymous. The way the network of a blockchain works is pretty interesting, as each transaction is saved on the ledger, there are no “names” embedded to each transaction, only IDs, or metadata about each transaction. This makes privacy and anonymity one of the main features of blockchain’s core uses and features. However, we will also give the user the feature to create polls in which voting can be public, given our scale will range initially from small polls, to eventually larger polls.

Fraud, double-spending, immutability and efficiency of Votes: One of the main issues when it comes to voting and polling systems, is its security features and whether fraud and the manipulation of votes can happen, which leads to corruption, and therefore to the fall of Countries. Votes will need to be efficient, and immutable. Being immutable means there is no way the votes can be modified or altered at all, therefore, there can’t be double-spending either. You get one vote, you get one opportunity to vote. Each vote will be registered with a key ID on the network, and the blockchain makes sure that ID can’t be altered, or spent again.

This is what separates a distributed ledger from a centralized ledger.The efficiency of our votes will be our top priority, as it may very well be our hardest challenge in this project, but that’s why the blockchain exists, to solve problems like this.

Intellectual Property Issues

As part of our development stack, we plan to use two projects that are open to the public for use. The web-development framework React and the Ripple ledger protocol, XRPL. Both of these projects are freely available. In fact, both actively support new developers utilizing their platforms. As long as we cite these libraries and adhere to their licensing agreements, we will face no intellectual property issues. The only violations of these agreements are attempts to engage in malicious activity. It goes without saying, we will ensure this will not be an issue for our project which will remain ethically sound.

Change Log

- The project milestones were very slightly modified to better fit our semester’s time schedule.
- Some minor grammatical errors and typos were fixed.
- The project description was updated to better reflect our new vision for the project.