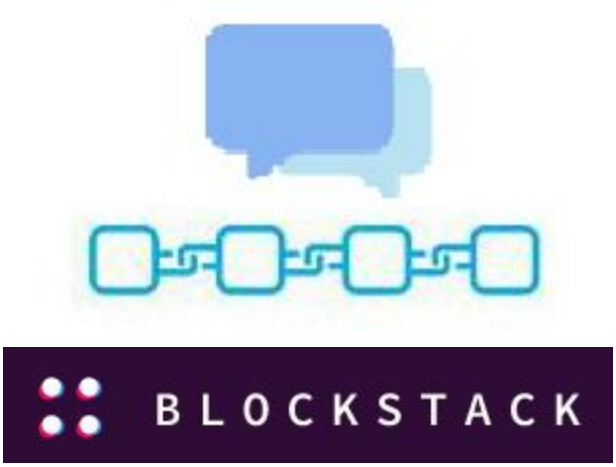
I. Cover Page



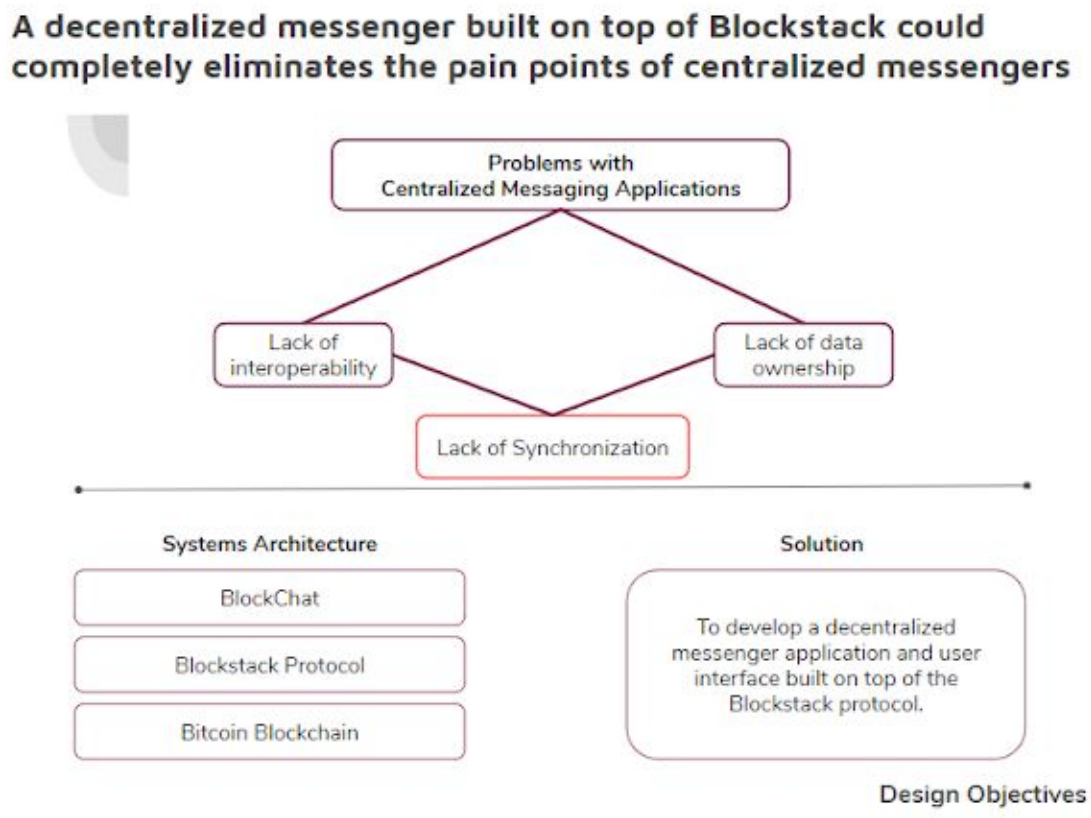
BlockChat: A Decentralized Messenger on the Blockchain

Andrew Zhang (SSE)
Sahil Shah (SSE)
Yassine Elmandjra (SSE)
Mark DeCesare (SSE)
Matthew Hanna (EE)

Advisor: Scott Snyder

II. Executive Summary

Our group developed a decentralized messaging framework and app, BlockChat, because we want to solve the numerous problems with the centralized messaging services that are prevalent in the current day. These problems include a lack of interoperability between services and a lack of data ownership. We developed our messaging framework on Blockstack, a decentralized internet, because it provides the decentralized identity and storage functions that are key to our project. Since Blockstack is programmed using Node.JS, a part of Javascript, we are using this, along with React.JS and Vue.JS for the user interface, to build our messaging framework and application on top of Blockstack. Furthermore, the new multi-player storage feature on Blockstack was especially important in implementing BlockChat, as it allows users to view files/messages in another user's storage (i.e., the person whom they are messaging). By the conclusion of senior design, we were successfully able to build an app that allows users to sign in using their Blockstack ID, manage and add users to their list of messaging contacts, and send messages to one another, which are stored as encrypted files on a user's local drive or cloud storage service. In addition, our app allows for group messaging between users, and it is created in a standardized way such that BlockChat users could potentially message with someone using a different decentralized messaging app on Blockstack.



III. Overview of Project

The objective of our project is to design a messaging framework, BlockChat, that allows for innovation, scalability, and interoperability between applications, while enabling users to own and control their data and storage. We aim to accomplish this through Blockstack, a new internet for decentralized applications. Blockstack users register IDs and are then able to run decentralized apps on the Blockstack browser, where their information is encrypted and stored on either the user's personal device or a cloud service (such as Google Drive), which acts as a dumb terminal in this case. As an open source project, Blockstack encourages developers to create new apps and services on top of the existing Blockstack browser. BlockChat leverages the power of the Blockstack protocol to enable cross-compatibility and interoperability amongst messaging services. Furthermore, using a Blockstack ID, users possess full ownership of their messaging data with BlockChat.

Today, most means of communication are dominated by centralized messaging services, including iMessage, Facebook Messenger, WhatsApp, and GroupMe. With BlockChat, we hope to solve several problems with these centralized messaging services. The increasingly common issues that arise from these centralized services include a lack of interoperability and synchronization in communication with the same individual across different platforms, along with a lack of data ownership around the privacy and content of the messages. Users are forced to create a separate account with each centralized messaging service (i.e., Facebook, Apple, Google, etc.), which means that messaging services ultimately have control over the user's data.

We believe our project is very interesting, as it builds on top of emerging blockchain technologies and aims to solve some important issues with messaging applications in the current day. Our project gives individuals a means of sending and receiving all their messages through only one service, and they gain ownership of their messaging data as well. With concerns about data privacy increasing, especially due to the pending removal of net neutrality, a decentralized messaging service that provides complete data ownership to users is extremely beneficial to individuals, and it will help to push forward Blockstack's breakthrough of a decentralized internet.

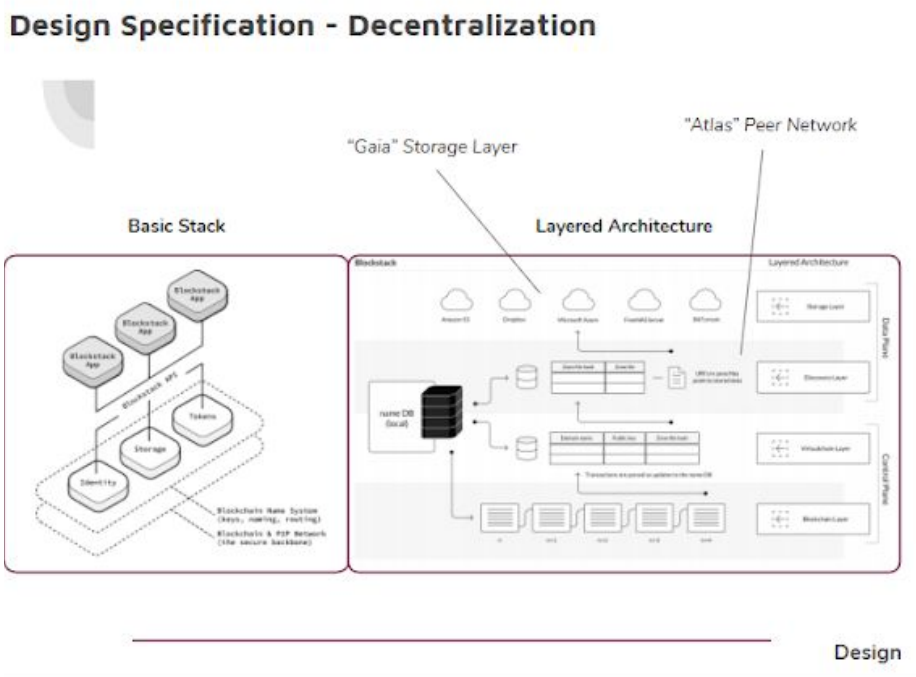
IV. Method of Solution

Specification and Requirements:

From a qualitative standpoint, the design had to solve three main problems: lack of interoperability (can't send messages from Whatsapp to GroupMe, and there is no sync of messages with the same contacts across applications), a lack of data ownership (Facebook owns all Messenger data, and users cannot migrate to other applications with this data) and a lack of ownership (due to the oligopoly structure of today's messaging applications, firms and new developers are not incentivized to innovate - i.e WeChat has so many features that iMessage does not).

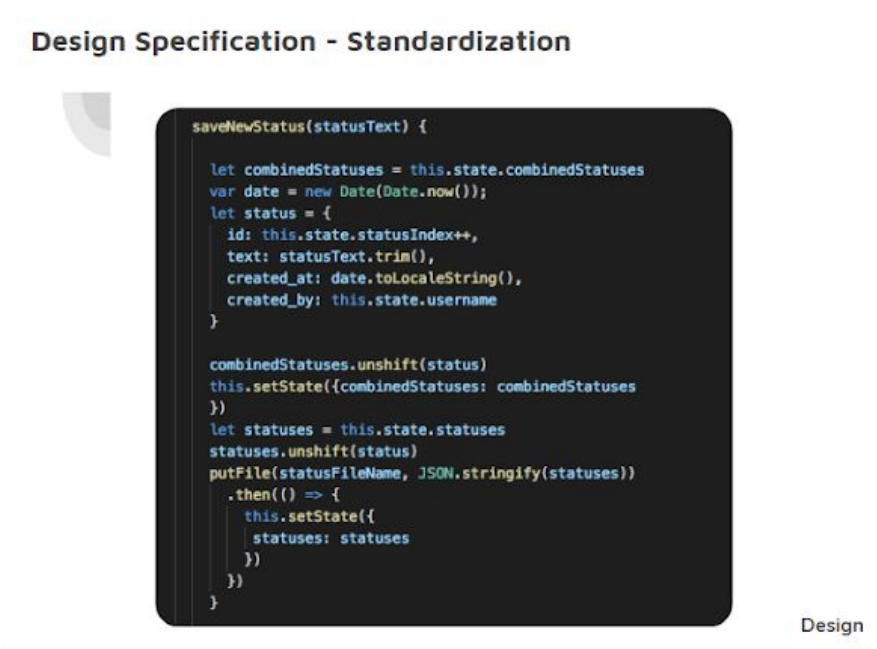
Therefore, from a technical standpoint, our solution was to create an infrastructure that is both ‘decentralized’ and ‘standardized.’ These terms are summarized as:

- **Decentralization using the Blockstack Protocol**
    - In the same way that Bitcoin stores a public record of all transactions, Blockstack stores identity.
    - Users own all their encrypted data, including Messages. All storage management is taken care of by Blockstack, and users choose their preferred current storage providers as “dumb terminals.”
    - One can look at the publicly available “Address Book” stored on Blockstack and directly establish communication.
    - All the identities and storage pointers are secured by the Bitcoin Blockchain, the most secure blockchain that would require the computing power of billions of computers to hack.
    - By owning your data, you are not bound to a specific application, which solves the primary issue with current centralized messaging apps.



- **Standardization**
    - We designed a JSON object such that all future developers can build on top of this framework.
    - The JSON object contains all the necessary components of Messaging including: Sender Blockstack ID, Receiver(s) Blockstack ID, Message Data, Multimedia, Time of Message, etc.

- This allows for true synchronization and interoperability within apps built on top of this framework.



Moreover, we also wanted to create a visual manner in which to interact with the framework, and, therefore, we wanted to create an instance of an application built atop of our framework. This included creating a front-end application using Vue.JS in which one can test the 'Decentralized Identity Systems' and 'Encrypted and Controlled Storage' that Blockstack offers.

Classes and Knowledge Required:

In order to implement this method, our project requires knowledge in the following critical fields:

- Blockstack API - We need to interact with the local instance of the Blockstack node in order to authenticate identity and access personal storage with a private key.
- React.JS - In order to build a simple, dynamic single page application, knowledge of React or a similar library is critical. We chose React due to its prevalence in the current ecosystem.
- JSON - We needed to understand how to build, encrypt, and use JSON objects in order to create a standardized messaging framework and allow our applications to seamlessly integrate with the storage (there is no back-end due to decentralization).
- Networks (ESE407) - This project depends on a deep understanding of Internet Architecture to understand how Blockstack uses TCP/IP infrastructure to create a scalable, decentralized network.

Alternative Designs:

There are various other alternatives that we could have considered in addressing the problems with today's messaging:

- Ethereum/Lisk - Instead of using Blockstack, we could have built our application on top of another protocol such as Ethereum and Lisk. The reason we chose Blockstack was that it provided us with decentralized identity, storage and payment. These are the three pillars of any application, and therefore this ecosystem was the best option for us.
- Create our own Blockchain - We could have also created our own protocol (this would also allow us to immediately monetize). However, we believe that being part of another ecosystem is more powerful because it allows us to: 1) embed our messaging in other applications, 2) create a developer ecosystem, and 3) use our identities for other applications.
- Centralized Database - We could have created an open-source centralized DB. However, the economics would not work out and whoever is the steward of this database would have the power to corrupt the system.

V. Self-learning

Due to the nature of the project, there was a lot of self-learning involved. Initially, all team members read as much as possible about blockchain and how the technology is completely transforming the Internet Stack. Then, we read about the Blockstack solution to create a Decentralized Identity System (with pointers to different storage) and how we could design something truly valuable using the Blockstack architecture. Specifically, we read the Blockstack whitepaper, which provided us with a lot of very useful information on the inner-workings of Blockstack. Many of the videos on Blockstack's YouTube channel were also helpful in learning more about the technology. In addition to these readings and videos, all team members completed several tutorials on Blockstack to develop a better understanding of how Blockstack works and how to develop apps on top of it. First, we completed the Hello Blockstack tutorial, which taught us how decentralized identity and identity authentication/login works on Blockstack. Next, we completed the Blockstack Todo List tutorial, learning how storage works on Blockstack (called Gaia storage) and more about Blockstack's identity authentication. Finally, we completed the Multi-Player Storage tutorial, which was extremely helpful in completing our decentralized messaging app. This multi-player storage feature allows users to make files stored on Gaia visible to other users, which is necessary to create a messaging app without a centralized system in place.

Subsequently, we needed to understand the technical requirements for the project. Therefore, we started participating in Khan Academy and other YouTube courses on: React.JS, Node.JS, Javascript in general (HTML was known before the project started), JSON Objects, and the Blockstack API.

Overall, this self-learning enabled us to successfully create BlockChat and achieve our goal of creating a decentralized messenger. Additionally, it was enriching for us to learn about Blockstack and the blockchain more broadly, as these are emerging and very relevant technologies in the modern world.

VI. Design and Iteration

Our initial problem statement, as stated above, was to resolve many problems associated with Messaging. The initial goal was to create a new, underlying blockchain to decentralize messaging by creating a tokenized protocol to incentivize developers. However, we quickly realized that we need some sort of decentralized identity management, as all current identities are owned by some centralized entity (i.e., Social Security is owned by the U.S. government, Snapchat ID is owned by Snap Inc, etc.).

Therefore, in the next step of our design decisioning, we decided to use Blockstack, the only known and scalable decentralized identity system with storage pointers. Storage pointers are key, as we need users to 'connect' their identity to the storage provider of their choice. We also need users to see where other users' files are stored and whether they can access a small subset of the files. This is critical to create a two-way messaging platform.

After deciding on Blockstack, we decided to build a simple "Hello World!" to familiarize ourselves with the development environment native to BlockStack, which largely involved using React.JS along with blockstack.JS, both of which were unfamiliar to team members.

At the end of the fall semester and beginning of the spring semester, we iterated upon our Hello World app by building a simple microblogging app using Blockstack's Multi-Player Storage, a Blockstack data-sharing standard that enables users to share data with other users. In this app, users can post new statuses, as well as lookup other users to see their posted statuses. This app was a necessary precursor to our final iteration since it was easier for us to implement non-private data sharing.

Finally, we arrived upon our final build. In this final build, users can send private messages to specific users using that other user's Blockstack id (ex. sahil.id). Within this design, we created a standardized JSON object that allows for interoperability on top of BlockChat. For example, users can plug their conversation JSON objects into a disappearing chat app that deletes chats after a certain amount of time. This final design also involved a front-end application, as we needed this for testing using Vue.JS. This iterative process allowed us to refine our problem statement and learn about various types of solutions. This made us truly understand the problem at hand, along with the advantages and disadvantages of our solution.

VII. Societal, Global, and/or Economic Impact

The History of Computing:

There are without a doubt global, economic, and societal impacts of the project we have taken on, especially as of recently. To provide context, it helps to look at the history of computing and where we stand in this timeline. Computing ultimately started in the 1930s with the introduction of the Turing Machine. The transition from the Turing machine came in the 1960s and 1970s, where most of computing was done by huge mainframes that terminals would plug into. None of the computing was actually done on the terminal and the terminals were therefore coined as "dumb" ones. In the 1980s, with the advent of Apple and the desktop computer, computing saw a huge shift in that users could own a personal computer and download their own software. This transition then led to cloud computing in the 2000s, where most of the data was pushed back to the cloud and managed by huge data centers. And so, what people are beginning to realize is that we are slowly being reduced back to these dumb terminals.

The Internet Today:

If we think about our computer today, there is very little point to it without the cloud. We do not actually own any of our data; rather, we rent our data from the Facebooks, Googles, and Amazons of the world. Not only do we have zero ownership of our data, but having these huge data silos results in a much greater probability of compromise and breach (just think Equifax). If we take a look at the current web stack, it makes sense why the Internet has evolved in the way that it has. Simply put, today's stack consists of thin protocols and fat applications. Very little value is captured at the protocol level (HTTP and TCP/IP) because of its stateless nature, and much of the data layer has to be built on top of these protocols, hence why the value in today's internet is at the application layer where massive companies leverage user data. The introduction of Blockstack, a new internet for decentralized applications, completely reverses this notion, where protocols are fat and applications are thin. This ultimately undermines today's centralized incumbents and creates new global, societal, and economic incentives to innovate and adopt.

The Ethical Implications:

With a system like this, current ethical issues are resolved, while new ones are raised. For instance, there has been much controversy on a lack of censorship resistance and large incumbents exploiting user data for profit. This results in a manipulation of information spread to specific users and poses very deep ethical issues. With Blockstack, users would have complete ownership of not only their data, but also their identity. None of the data would be stored at the application level, and all information would be encrypted and distributed with no single point of failure. This however, brings out a new ethical question as to whether or not users would be

ready to bear the responsibility of being the sole owner of their data. Moreover, there are issues such as terrorism and cyber-bullying that would arise in a decentralized DB.

The Use of Bots:

However, we would be able to address this with “bots.” These are AI-enables protocols that can be embedded within our standardized protocols to ensure that no messaging is harmful in anyway. There could be an international committee created to “draw the lines” on what harmful messaging entails and to ensure that these AI algorithms are not biased and do not discriminate. In fact, we think these bots can essentially replace all the benefits of centralized messaging companies.

VIII. Business Plan

Executive Summary:

Developing an application for a decentralized ecosystem sheds light on very interesting notions around where value is captured (and hence where the business model can be focused). A decentralized application on top of a decentralized protocol completely reverses the business models that we know today. In the internet of today, most if not all of the value is captured at the application level (largely in the form of proprietary databases). That is, the large centralized incumbents like Google, Amazon, and Facebook have been able to generate massive amounts of value in large part due to the data collected from users and the ads generated as a result. In a decentralized messenger, this business model is no longer valid, as none of the data is actually stored at the application level. Many involved in the space realize this notion and its implications. Decentralized applications bring about a new form of open competition where users are no longer siloed to any specific platform and platforms can no longer leverage the data that users provide as effectively. For that reason, there has been much thought on where value can be captured in this ecosystem. Let us look at value generation that might occur by building on top of Blockstack.

Monetizing on top of Blockstack:

For one, Blockstack recently raised a 25 million dollar fund called the “Blockstack Signature Fund” that is solely focused on encouraging developers to build on top of the ecosystem. Developers are rewarded with bounties for building specific applications that can be used by Blockstack users. This also includes building out a decentralized messenger on top of Blockstack. While this is certainly not a sustainable or recurring model, bounties can definitely provide a runway for the company to execute on future plans that provide a more sustainable business model.

Monetization of blockchain protocol:

In Web 3.0, the dynamics of provisioning a resource or service are far different. Most decentralized applications are likely in the long run to be powered by a native asset, a cryptoasset, that will enable the resource or service to be provisioned. One might think of this asset as the unit economics of the protocol. For a social network or messenger app like BlockChat, we can think of the following analogy and translate this to see how BlockChat as a company may become more valuable. Let us think of a decentralized application with its accompanying cryptoasset as a county fair with a limited number of tickets. In this fair, there are tons of rides and entertaining games that can be played, but the only way to participate in these rides or fairs is by having a ticket. If there are more people who want to partake in the fair than there are tickets, then basic supply and demand tells us that the ticket will be costlier. If the organizers of the fair generate revenue based on how costly the ticket are, then, as demand for the ticket increases, their share of revenue also increases. This opens the conversation for how a tokenized network might generate value. In the case of BlockChat, as of now there is no native asset that powers the network. However, increasingly we are seeing a trend in messaging applications that tokenize their networks. This includes both Telegram and Kik, two of the largest messaging platforms in the world.

Other purposes for tokenizing a messenger network would be the democratization of ads as well. An economy can be created for selling and reading ads in which users get paid for reading ads and advertisers are able to display ads based on the number of tokens they have. The price would fluctuate from supply and demand. If developers and the team allocate a portion of these tokens to themselves, they will accrue a lot of value as ad usage increases. If BlockChat does not have a native asset powered, there are certainly other ways in which revenue could be generated. With the openness of the messaging application, users will have the opportunity to interact with anyone who accepts their request. What if a user wants to interact with someone famous or otherwise hard to reach? Is there some sort of way to align incentives to respond for the person to whom the user is reaching out? With something like BlockChat, a new model can be introduced whereby users pay (in a cryptocurrency) another user to read their message (similar to earn.com). In return for what has been received, the recipient must now respond to the message. From this transaction, BlockChat would take a transaction fee.

Competitors and Adoption Risks:

Decentralized applications on a decentralized internet as a whole are still very far from competing with centralized incumbents. This can be particularly highlighted with the recent Cambridge Analytica controversy. Besides continuing to shed light on data privacy concerns and revealing how information extracted from these honeypots can be exploited, the recent Facebook controversy brings about an interesting realization on user preferences. And that is, at the end of the day, the average platform user still doesn't care enough to leave or search/push for a better alternative. For instance, the "delete Facebook" campaign resulted in a

measly 1/100th of a percentage of users removing their account and, even in casual conversation, it is accepted that most people really do not care that their data is out there. And this makes sense: There still is no widely adopted alternative to compare/switch (something like Blockstack is still in its infancy stages), the new generation of internet users has always been accustomed to sharing data online, and most importantly, the implications of data exploitation are predominantly second order. To elaborate on the last point, yes, it is scary that these platforms can control user data and potentially influence behavior as a result. Yes, it is scary that there could be a data breach that risks leaking user info (as we saw in the Equifax Data Breach). Yes, it is unfair that these centralized platforms are generating millions of dollars of revenue by exploiting user data without having the user be even remotely financially compensated. But, through a deeper analysis, one comes to the realization that a user is not really able to tangibly and instantly measure the ultimate impact the lack of data control would have on him or her. Because there is no way to make a quantitative, immediate, and tangible assessment of the impact, users might subconsciously marginalize the situation; the human psychology is inefficient in taking actions on second order effects. This could partially explain why Equifax, a negligent business model that requires all data in one place, is up 380% since 2011 and continues to recover since its data leak fiasco (up 25% since crashing) and Facebook, a direct competitor to BlockChat and co., had record earnings in Q1 2018. While there certainly is value in decentralized platforms and providing data back to the user, before irrationally going about discussing a business model, it is important to take a step back and analyze the potential lack of adoption given the era that we are in currently.

IX. Summary of Meetings

Our team met regularly during the semester to discuss major milestones as they approached, and to give updates on progress. Typically, these meetings happened anywhere between Sunday and Tuesday of each week. Between January and early February, the meetings centered on learning about Blockstack's new multi-player storage implementation by working through the demo available on Blockstack's website. These meetings also focused on incorporating updates into our application structure. From February until April, work occurred both remotely and in meetings, as during the second half of the semester, other coursework increased, as expected. Since our project was not hardware-centric, we had flexibility with where we worked. Because of the new nature of this space, meeting with stakeholders or consultants was neither required nor very practical. However, we did meet with our advisor, professor Scott Snyder, who was able to provide us with helpful advice and suggestions when needed.

X. Final Schedule with Milestones

In the spring semester, our team successfully met its goal of developing a communication system for users of BlockChat. Once Blockstack released its multi-player

storage system API update, our team began working and using the updated technology. The schedule we followed is detailed in Appendix A. The following table details the milestones and how each member contributed:

Milestone	Member Involved
Implement two-person communication	All (Ideation & Implementation)
User Interface Update (graphics, view)	Sahil, Yassine, Matthew
Business plan development	Andrew, Mark
Multimedia messaging and other features	All (Ideation), not implemented

As described in later sections of this report, group messaging was a difficult feature to implement because it was unclear how to update permissions for groups under Blockstack's update. Additionally, as mentioned in the last report, Blockstack is not supported on Windows, and so our team member Mark was somewhat limited by this. Nonetheless, the team persisted, and each member still contributed to designing the app. For the most part, we met our milestones and course deadlines.

XI. Discussion of Teamwork

The team used Facebook Messenger to communicate (unfortunately, BlockChat was still being developed at this point) and Google Drive for file storage and maintenance. We used this platform to discuss dates for meetings, which we held regularly. Where in the first semester our focus was learning about Blockstack and the technologies involved, the second semester involved implementation and iteration on said knowledge. As such, tasks were split based on technical expertise and familiarity with the technologies. All team members were involved in the ideation process; we discussed how we wanted the application to look and feel, and what potential issues could arise upon implementation. Each team member also completed Blockstack's Multi-Player Storage and Todo List demos to gain insight into Blockstack's functionality and updates. While each member became familiar with Blockstack during the first semester session, Sahil and Yassine had much more experience with Blockstack due to prior study and projects, and so they took on leading roles for the team, explaining technologies and considering how our ideas could function plausibly in Blockstack's ecosystem. From here, we split: Yassine and Sahil worked on the backend of development, while Andrew, Mark, and Matthew work on front-end development. Mark and Andrew also focused on business aspects of the project, such as budgeting and developing a business plan for the project.

XII. Budget

Because we developed an application, our budget was smaller than most other projects. The only cost we incurred over the two semesters was the registration cost for Blockstack ID's for Andrew, Mark, and Matthew. The Blockstack ID's were critical for our development, as they provided us access to Blockstack's platform. We anticipated a $50 registration fee for these ID's in our fall semester and report, but this price decreased to $30 in spring semester, therein saving money. This price reflects the cost of Bitcoin, which experienced significant fluctuation between December and March. Although we expected app registration costs to be a constant expense, they did not end up coming into play with Blockstack. Therefore, in total, our team spent approximately $90. Other than these changes, we made no other adjustments to our fall semester budget.

XIII. Work for Second Semester

Second semester's work focused heavily on implementing the ideas developed from the first semester session. While delayed briefly in the beginning of spring semester, development became possible once Blockstack released its updates which allowed multi-player storage on the Blockstack platform. Because of this update, our application could function, as we use files to send messages. This was not available in the first semester.

We planned our growth in stages. Our first milestone was a simple messaging platform where two users could communicate with each other. Stylistically, this communication functioned more like a blog and less like an app, but it was important because it was the first implementation attempt, and it was functional. This work was finished by mid-semester demo-day. Our second milestone centered on solidifying the first milestone's foundation, cleaning it up and adding new designs and user interface. Our last milestone desired to expand this individual communication to a group-messaging platform, and also aimed to incorporate multimedia messaging (such as pictures and stickers). We anticipated this third milestone to be simple, but it was more challenging because Blockstack's multi-player system is complex. The challenge was not just about connecting people, but in making the communication accessible for all users involved.

In the second semester, the team planned more consistently and adjusted as it discovered which milestones were more plausible than others. From here, we pivoted from initial goals established in the first semester. We anticipated our initial goal of finishing by February 28 to be aggressive, but despite adjustments and some delays, we met our goal of finishing by April.

XIV. Standards and Compliance

Our standards can be organized into two layers. The first layer is our project, BlockChat, plus the browser it is built on top of, Blockstack. The second layer is the Blockchain that BlockStack is built on top of (see Appendix B).

These layer are additionally divided into two parts, (1) technological and (2) legal/ethical.

At the first layer in the technological section, BlockChat leverages a technical protocol called Multi-Player Storage that is native to the Blockstack browser. This protocol allows the user to not only own and control access to her own data, but also share with her friends without the need for a third party. This piece is critical to our chat application, since it allows users to share chat files without the need for a middleman.

Within this same section, the Blockstack Naming System (BNS) standardizes issues of identification registry and domain access. This naming system provides human-readable ID's (ex. Paul.id) to end users in a decentralized manner. Users can then send a chat to another user by addressing his BNS-enabled ID.

Shifting to the legal and ethical side of the first layer, EU Data Privacy Laws are coming into effect in 2018. Among the numerous provisions, one includes the right to ask to have a company erase your data. These regulations do not apply to BlockChat because users store their data on their own personal storage, not some "BlockChat data center". This subverts the complication that arises in traditional chat applications of storing private user data.

Again in this quadrant, individuals may use BlockChat for malicious reasons such as cyberbullying. However, this is a problem that is native to all messenger applications. If BlockChat were to implement a mechanism to monitor messages, it would eliminate the benefits of personal data storage that makes the application compelling to use in the first place.

Shifting to the lower layer of blockchain standards, on the technological side, Blockstack is critically dependent on the consensus standards of the underlying blockchain (in this case, Bitcoin). In the same way that one's Bitcoin wallet is reliant on the security of the blockchain, so too is the security of our messaging application.

On the legal and ethical side, we see that Bitcoin stands in a gray area right now. Some countries, like Japan and the EU, have fully legalized Bitcoin. Others, like Algeria, have banned the use of it. BlockChat is reliant on the existence of Bitcoin miners verifying transactions, and miners are incentivized by the value of Bitcoin. If governments continue to ban the use of Bitcoin, the future of Blockchat could also be threatened.

XV. Conclusion

With respect to the decentralized internet and communication among players in such a network, there is still much work to be done in deciding how to enable connections. Our application was an attempt to explore methods of communication under the assumption that BlockChat and Blockstack were widely adopted. By no means was the final product robust, but it was effective in using this new technology and exploring its use cases. The application utilized Blockstack's multi-player storage system, and allowed users to send messages to each other; in

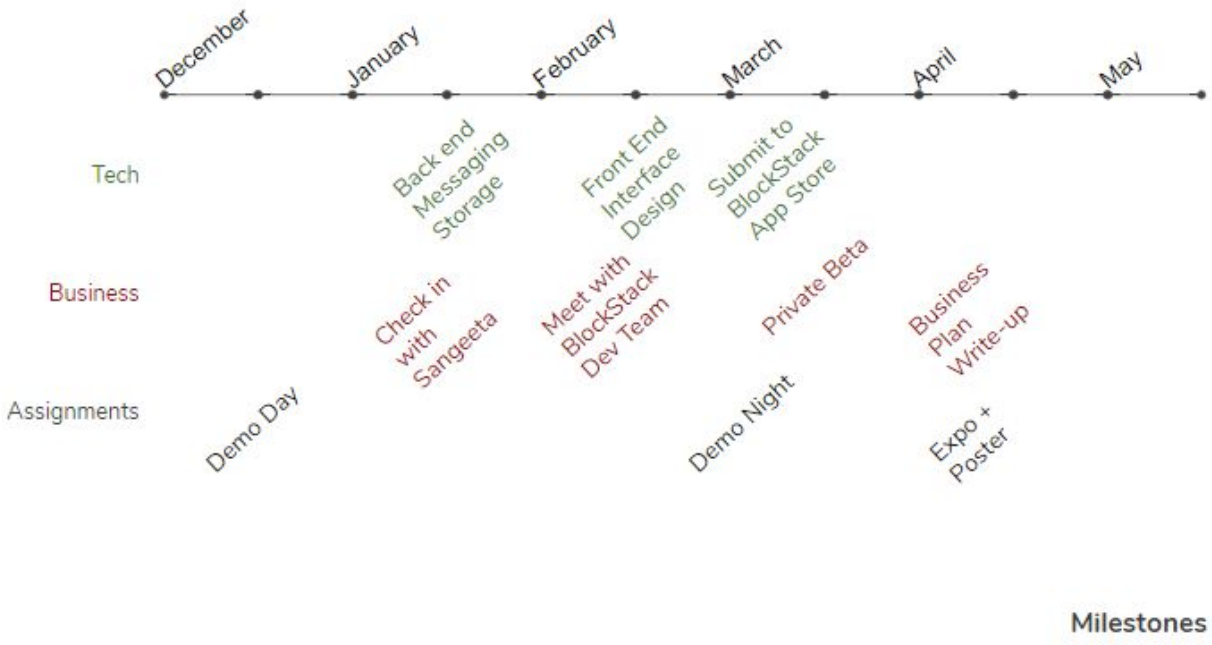
these ways, we met our desired goal. In this lies one of the most valuable lessons we learned, as we applied our collective knowledge from coursework at Penn to solve a problem.

It may take years, decades, or even lifetimes (if ever) for BlockChat to be adopted simply because this adoption requires Blockstack's decentralized internet layer to function. Furthermore, the app exists in a market dominated by larger, more established players like Facebook, WeChat, and countless others, and in all likelihood, these companies do not wish to lose market share. Overcoming this challenge is not easy, and requires a substantial amount of time and financial resources in addition to a sound, robust final product. Future iterations would include other forms of messaging, such as multimedia messaging, and updates to the user interface; significant attention to detail must be given to privacy maintenance and monitoring of message threads, if possible. Plus, Blockstack is always updating its features, and so development is certainly possible in months and years to come.

This project was a valuable learning opportunity, as it required us to use multiple aspects of our knowledge. We learned how to communicate as a team, and to allocate tasks effectively for productivity. Fundamentally, this project deepened our technical knowledge as well, giving us insight into app development.

XVI. Appendices

Appendix A:



Appendix B:

	Technological	**Legal & Ethical**
BlockChat	Data Sharing Standard (Multi Player Storage)	EU Data Privacy Laws?
BLOCKSTACK	BlockStack Naming System (BNS) for IDs, Domains, Routing	Malicious Use of Technology (bullying, terrorism)
Blockchain	Consensus (Proof of Work)	*Legality of Bitcoin* Legal: Japan, EU Illegal: Algeria, Bolivia Gray Area: China