Sharify

Buy Less, Borrow More

ESE Senior Design 21-22, Team 17
04/29/2022

Adviser: Rakesh Vohra rvohra@seas.upenn.edu
Team email: teamsharify@gmail.com

Team:
Angela Lin, CIS & Finance, angela5@wharton.upenn.edu
Phoebe Weiser, SSE & Business Analytics, pweis@wharton.upenn.edu
Maanasi Garg, CIS & Finance, maanasig@wharton.upenn.edu
Cole Hersowitz, CIS & Finance, chersowi@wharton.upenn.edu
Ken Schumacher, CIS & Management, kenschu@wharton.upenn.edu

II. Executive Summary

Sharify is a peer-to-peer, on-demand item sharing platform that connects borrowers and lenders for items of interest in a closed community setting (a school campus, a workplace, a neighborhood). Whether it's computer chargers, office supplies, electronic devices, kitchen supplies, or other day to day items, every member can be both a borrower and lender, posting items they are willing to lend or give away while also requesting items they are seeking. Sharify seeks to combine what works from the on-demand economy and shared economy models into a platform deployed in closed communities to ultimately reduce waste, increase sustainable consumption behaviors and encourage people to buy less by borrowing more.

Our model and application setup is based around a set of limited inventory and predefined categories consisting of the most needed and portable items. With market research surveys, we discovered immense commonality among the items that potential users specified they would want to borrow or lend, such as chargers and medicine like Advil. Our team built an MVP consisting of a front-end react native app, a firebase backend, a flask-based REST API, in addition to our core matching algorithm which was developed along with a Python test environment. The Sharify algorithm focuses on quickly and efficiently dispatching borrow requests that are either accepted or rejected by the user to make a supply and demand match while also not burdening potential lenders with excess notifications.

III. Overview and Motivation

Each year, the world generates about 2.12 billion tons of waste and 99% of the stuff we buy is put in the trash 6 months later[1]. We believe this societal issue does not stem from a lack of intention or awareness to reduce waste and reuse items. Especially among younger generations, there is wide awareness of issues concerning waste and the destruction of our planet. Rather, there is often an economic mismatch between people who need or want to borrow goods and those who can supply it immediately. This mismatch stems from frictions within the discovery process; potential borrowers are unable to tap into the supply around them.

From market research, we find this mismatch presents itself primarily through two scenarios. In the first, people find themselves needing items urgently but are unable to access them due to distance, lack of ownership, or other situational constraints. These items are generally everyday goods, such as a calculator for an exam tomorrow, some Advil for a headache mid-grind when CVS is closed, or even a mask to enter a public place. In the other scenario, people often struggle to access niche items that are hard to find or costly to buy, but could easily exist in a peer network. Examples of this need include a $100 textbook that a student only uses for a semester, or some nice but inexpensive furniture that a college senior will only get to use for 10 months. In these cases, there's often an individual nearby - in the same building, neighborhood,

1 https://www.theworldcounts.com/challenges/planet-earth/state-of-the-planet/world-waste-facts/story

somewhere else on campus - who owns the item and is willing to lend it out, or even donate, for some financial reward. There currently does not exist a solution for efficiently matching demand and supply of easily shareable items.

Our solution is to efficiently match requesters and lenders of demanded items to help busy individuals (students, young professionals) obtain items they need by helping them tap into the existing supply around them, thereby reducing overall waste. Sharify is an on-demand item sharing platform that connects borrowers and lenders for items of interest in a closed community setting based on location, willingness to lend, and usage. At any point in time, when a borrow request is made for an item, a matching algorithm ensures an optimal lender is found and matched with the borrower based on distance, availability, and reliability within an optimal matching time.

There is a significant amount of discussion around how young individuals are destroying the planet through waste, but relatively few effective actions have been taken in response. By identifying a platform that has the potential to significantly reduce this waste, our team was motivated to build and test its effect. Sharify's proposed structure allows the app's benefits to easily scale; successful implementation within one community implies success elsewhere using the same technology. We believed a platform with these levels of impact was worth our time and worth building.

IV. Technical Description

A. Specifications and Requirements

The technical specifications we outlined for ourselves included implementing a matching algorithm that ran in <10 seconds, ensuring a database latency of no greater than 10 milliseconds, and allowing a user to submit a borrowing request in no greater than 3 clicks. The key constraint we had to work around was a limited amount of lending and borrowing data. Since item lending and borrowing is a novel concept, there aren't any existing datasets capturing which items users might lend or borrow most frequently, how long a lender might take on average to respond to a lending request push notification, how the lender population might be distributed in terms of rare lenders versus superlenders, and more. We therefore had to use educated assumptions to create mock lender and borrower data, and design a simulation to test our algorithm based on this mock data.

We also had to design the algorithm to adhere to the engineering standards that applied to our project. This meant making sure that we weren't storing historical location data or demographic data, meaning that we could not use either as a part of our algorithm optimization. Finally, our biggest constraint was the timeline of this project. Due to our project requiring us to build out both a front end React application and a full simulation in order to test our backend algorithm, we did not get time to test our integrated app on a large number of live users. Testing out our application on live users would have allowed us to refine our algorithm weights based on live user data.

B. Iterations

There are several alternatives we considered for our platform solution, requiring us to make key design decisions. First of all, we decided to go with a mobile app rather than a web app since Sharify is heavily location based and would utilize features like push notifications which are more conducive to a mobile experience. Second, we decided to use dynamic pricing for items on our platform rather than fixed pricing, since dynamic pricing would allow us to better address any supply and demand imbalances for specific items, similar to Uber's surge pricing model.

Third, we decided to use a points based system to price the items rather than a USD based pricing system. This will allow us to reduce friction for single use and low value items, since there would probably be high resistance to paying dollar fees on these items. It will also allow us to reduce user friction in entering payment information. Finally, we decided to start off Sharify with a predefined set of items, rather than allowing users to borrow and lend any item of their choice. This would allow us to concentrate the initial set of lenders on our platform into fewer item categories, allowing for faster matches and increased borrower satisfaction. This decision may be reversed in the longer term.

C. Societal, Environmental, or Economic Considerations

There were key social considerations that drove our design to offer Sharify only to closed communities. We realized that it could be difficult for lenders to trust borrowers on the platform who they do not personally know. However, if the borrowers are from a closed community that they have regular interaction with, and that they respect and trust, they might be more willing to lend. A closed community would also create more social consequences for damaging or losing items, hopefully maximizing the number of safety returned items. In addition, we realize that members of closed communities generally have a desire to meet other people in the community and earn social currency, which is something lending on Sharify would allow them to do.

Finally, as we described in the previous section, low willingness to pay transaction fees on low value items led us to designing a points based system rather than dollar transaction fee based system.

D. Technical description and approach

The technical development of our product was broken into distinct stages; the construction of the algorithm and construction of the platform (occurring in parallel), and the integration of the two. We outline each stage below.

Sharify's matching algorithm is the key innovation of our project that drives the efficiency of the platform (Figure 1). We see user engagement, satisfaction, growth, and other KPIs as primarily stemming from our algorithm. There were several metrics of user experience that we wanted to meet; for example, we wanted to minimize user matching time, and encourage first-time lending. While it was difficult to create an algorithm that accomplishes all of these goals, there was also a challenge along the way in judging how well our algorithm met these requirements.

To develop and test our algorithm, as well as see how well it met our KPI's, we created a simulation environment (Figure 2) to mock situations for our algorithm to help us get a match in under 10 seconds. More specifically, our team constructed a python module that allowed us to create mock lender and borrower objects with mock attributes, and then simulate the arrival of these lenders and borrowers over time. We then could observe how our algorithm managed the supply and demand over time, as well as how it met the performance goals that we outlined. For example, in initial iterations first-time lenders weren't being selected for as many matches, as the history of lenders was considered, causing veteran lenders to be selected more often than not. Our group saw this as a sub-optimal outcome as we want first-time lenders to be welcomed into the community quickly. Thus, we edited our algorithm to give "the benefit of the doubt" to initial lenders, giving them a quantitative boost when judged by the algorithm and boosting their utilization rates. After several iterations in the environment, we created an algorithm capable of meeting almost all performance metrics we outlined in the fall.

To provide the most accessibility possible in terms of a user base (an important social factor), we built a cross-platform React native app that could be deployed to both iOS and Android while also taking advantage of a device's native functionality like location services. We added a Firebase backend that handles both secure user authentication (an important privacy and trust factor) in addition to offering a cloud database service. To integrate a backend matching algorithm, we exposed it via a custom API we built using the Flask framework in Python. We were able to run this api locally for testing and then could also deploy it on a Python virtual machine and bind the IP publicly so our app could make http requests to it. Refer to Figure 3 for a more detailed service diagram.

E. Final status of the project and test results

Our final product met the three key quantitative requirements we had set out for ourselves. We were able to design a matching algorithm that ran in ~3.1 seconds on hundreds of mock lender and borrower profiles stored in our database. This was thanks to our low database latency at 9.8 milliseconds, although latency will be hard to maintain as the number of lender and borrower accounts on our platform exponentially increase. Finally, we were able to design the React native app to allow for a basic borrowing request submission within three clicks.

F. Overall evaluation

Since we did not get to test in a real life community, we are not certain if our matching time and latency goals would be met, but we do have some validation that it is at least possible via the simulations conducted. Additionally, we designed the UI/UX so that a borrow request can be made in less than 3 clicks, but if the inventory becomes more complicated such that we need to account for variations or add features like a requested pickup location, then it is likely that making a request will require additional clicks.

G. Conclusion

Given the lack of existing data related to item lending and borrowing, our tight timeline, and our limited resources, we were happy to end with an integrated, deployable application, and a matching algorithm that meets our initial criteria based off of mock data. Our next steps would definitely be live testing, in order to confirm the various educated assumptions we made about lender and borrower behavior, and to flesh out more features on the React native app, such as live tracking of your matched borrower/lender on a map.

V. Self-learning

A. Self-learning

We all learned Figma to prototype our idea and develop the first iteration of our front-end UI. We used Figma for extensive design work, from storyboarding, to building our app pages, flows, segues, and features, to connecting the views together to animate the borrowing and lending scenario. The software itself was very comprehensive with its own development environment so we spent time teaching ourselves the different design and programming functionalities and how to best utilize them to achieve our application goals.

On the backend, we did extensive research on online spatial matching algorithms and demand/supply distributions, understanding both the algorithmic theory as well as practical techniques we will use to build our backend model. We used elements of our research to then design our own greedy algorithm based on state of the art frameworks and methodologies being done in tangential shared economy spaces (e.g., ridesharing).

We learned about building REST APIs and needed to do this for the integration between the front end of our app and the matching algorithm running on the backend. To do this, we learned how to use Flask and we used clients like Postman to generate mock requests to test the API output.

We learned about real time database operations, specifically using Firebase, and needed to have streams to listen to new Message objects that were added to a data table so that messaging communication between a lender and borrower could occur in real-time like it would with Apple’s iMessage app.

For the front-end, we utilized React Native, and several packages for front-end components, which required reading lots of documentation and getting familiar with cross-platform mobile development.

B. Specific classes and knowledge

STAT 430 and 431 helped with the understanding of stochastic processes, generating supply and demand models through relevant probability distributions, and drawing inferences of overall population data from sample sizes.
Algorithmic design/programming classes such as CIS 121 and CIS 320 taught a good theoretical graph theory base to model our lenders and borrowers and therefore come up with an optimal matching.

Machine learning classes such as CIS 520 and CIS 522 gave us the neural network and deep learning knowledge to design our model to be non-linear, complex, trianable, and generalizable.

Finally, general mobile app development experience helped us enforce best practices for our mobile UI.

VI. Ethical and Professional Responsibilities

A. Professional Responsibility

A primary motivator of our project was that the platform can create significant contributions to the closed communities in which we operate. If implemented successfully, we hope that Sharify will promote the sharing of otherwise unused items, reducing waste and contributing to the circular economy. The effects of the increase of sharing are twofold. First, the utilization of items within our preselected categories will significantly increase, as our platform will facilitate easy access for borrowers to these unused items. Sharify's mission of reducing waste comes at the right time, as there is a new worldwide focus on keeping the Earth clean, significant investment in new ESG initiatives, and growingly alarming environmental challenges around the world.

Second, we hope that our facilitation of item sharing will increase cohesiveness within the communities we support. Every sharing transaction on our platform relies on trust; while we will provide insurance policies and support in case of issues, each lender trusts that their corresponding borrower will use their property responsibly and return the item on time and without damage. When this transaction occurs successfully (and we predict the overwhelming majority of transitions will be), the involved individuals will receive positive feedback for their given trust. Over time, we hope that this process will build cohesiveness and closeness between community members.

B. Ethical Issues

Building any user-facing services comes with a set of duties, but Sharify was unique in the number of considerations necessary to responsibly launch our platform. Every user on Sharify trusts that we will protect their personal information and location data while utilizing them when appropriate. Further, we must ensure that all users are treated fairly on our site, and that no single demographic, location or other group benefits or suffers from unfair treatment (e.g. longer wait times) on our product. To meet these standards and minimize potential for bias, we had to constantly evaluate the ethics of our product, throughout ideation and in all areas of our tech stack.

One significant effort to reduce bias in our product was derived from the simulation module for our algorithm, built above. While the environment was primarily created to house and develop our matching algorithm through the mocking of borrower and lending data, the same setup allowed our team to observe and tune how different groups of users were being treated by our product. For example, in an early iteration of our product, the location of previous matches influenced the probability for a future match to occur in the same location. While this boosted performance among a set of variables we considered, this could lead to problematic circumstances when deployed publicly. For example, if Huntsman Hall became a “hotspot” through this algorithmic process, business students may unfairly receive faster response times than the broader university, violating the ethical guidelines we set for ourselves coming into the project. Running several simulations with our final algorithm gave us confidence in fair treatment of our users. The recording of real world matches (and associated metrics) will allow us to backtest and confirm that no group is being unfairly treated when Sharify is publicly deployed.

Sharify is prepared to be compliant with Apple’s App Store guidelines as we prepare for general launch. Sharify does collect sensitive personal data, such as a user’s lending history, items on the site, and users’ current location when lending items. We are taking necessary steps to protect user’s data, such as encrypting communications with our Firebase database, deleting unnecessary or old data, and requiring consent from users when location data is being utilized.

VII. Meetings and Outside Advisement

Our primary advisor for this project was Dr. Rakesh Vohra, a faculty member in the ESE Department and the Economics Department at Penn. Our team had a limited number of zoom calls with him during the first semester when we needed his input on how to approach our platform and algorithm design. For the second semester, we primarily communicated with him via email bi-weekly or monthly updates that we sent out. During the first semester, we also met with Simone Marenisi via zoom, an OIDD professor in Wharton, to discuss the incentive structure for borrowers and lenders on Sharify. Additionally, before mocking data, our team met with Professor Venkatesh, an ESE professor with research interests in machine learning and applied mathematics. When we were first ideating early in this course, we arranged supplemental meetings with Sid and Jan to get preliminary feedback. Additionally, we met with Sangeeta Vohra, our M&T director, who provided early guidance.

VIII. Proposed Schedule with Milestones

Milestones

1	Create mock inventory and listing templates database objects
2	Test user authentication with at least 2 different accounts
3	Setup and test real time messaging between 2 different accounts
4	Create a POST API endpoint that calls Sharify matching algorithm
5	Tune algorithm on simulations of 300+ users
6	Compile react native app on ios and android
7	Deploy an app build for a closed beta test

As depicted in the table above, we were able to meet many milestones but had to scale back on numbers 6 and 7. When it came to compiling the app for native deployment, we encountered a series of React related issues pertaining to the newer package version we were using. Since React native is an open source, we were able to review github tickets and forums of many people encountering similar issues. After days of debugging, we realized that we needed to spend time on more critical parts of the project, so we scrapped this initiative which then in turn affected our ability to deploy the app for a closed beta test.

IX. Discussion of Teamwork

A. Coordination of Work

From the onset of the project, we recognized the importance of teamwork within our project to produce maximal results. Our collaboration was strong from the start, meeting in a hybrid of small groups as well as team-wide throughout the fall. Since the majority of our fall work was non-technical, we found it easier to assign work individually and meet as a team regularly to share progress and work on shared deliverables. As all five members are in M&T and have busy schedules, we elected to have two recurring meetings a week on Sunday and Wednesday. This ensured that all team members blocked off meeting time and we're able to participate regularly.

Our strategy changed in the spring semester as we began the technical work of Sharify. We spent several hours in January brainstorming workflows, as we felt that clarity was important to move the building process as quickly as possible. We decided to split the group into two sub-teams: Ken, Maanasi and Angela would work on the algorithm and accompanying environment, while Phoebe and Cole worked on the construction of the project.

After the definition of workflows, each team frequently met at informal times over the course of the semester, and we had one recurring weekly meeting to share progress. After the completion of each work stream, we worked as a team to integrate the two products and prepare demo materials.

X. Budget and Justification

Original Budget

Part	Source of funding	Unit cost/Subtotal
AWS Cloud storage/server platform	ESE	$100/month for EC2 and DB usage
Jira- ticketing/PM software	ESE	$7.50/user/month so $37.50/month
Figma/Balsamiq wireframe software	ESE	$89 license
User Promotions	ESE	$200 pool of gift cards to join app initially
Advertising (next semester)	ESE	TBD
UI/UX Template	ESE	$50
	Total	>$475

Actual Spending

Part	Source of funding	Unit cost/Subtotal
Firebase Costs	ESE	$50
Figma/Balsamiq wireframe software	ESE	$89
UI/UX Template	ESE	$50
	Total	$189

Our actual spending was significantly less than our budget. This stems from the fact that the original scope of our project was much larger, so we were anticipating publicly launching a completed app and having some operating expenses associated with doing so such as some limited user promotions. Given the original scope was very large and we needed to focus on the core application functionality without a public launch, many of these expense line items such as advertising were either eliminated or significantly reduced. We ended up using Firebase services for our backend instead of AWS.

XI. Standards and Compliance

Our project complies with the IEEE P7002 standard, a personal data privacy standard, and other US privacy laws, by ensuring that historical location data is not stored, and that user login authorization data is securely stored. It also complies with the IEEE P7003 standard, a standard that requires platforms like Sharify to consider algorithmic bias, by ensuring that we will not store or use demographic data in order to inform our matching algorithm. Finally, it will also comply with the IEEE STD 23026 standard, which dictates rules on the engineering and management of websites for systems, software, and services information, and the RFC 4122 standard by generating unique UUID identifiers for users.

XII. Work Done Since Last Semester

By the end of last semester, we had defined the specifications of our platform, drafted what technology stack we would deploy, and created a Figma mock user interface for the Sharify platform. This semester was focused on actual implementation and deployment. Half of our team worked on actually implementing the back end algorithm that we had designed, and on the mock data and simulation environment we would need to test this algorithm. The other half of us worked on the React native mobile application. Finally, we did basic integration of the front and back end and tested this integrated app on a few live users. Outside of the technical work, we further refined our business plans as we applied to various funding competitions, including the Venture Innovation Prize and the President's Innovation Prize.

XIII. Discussion and Conclusion

By the end of April, we ended with an integrated, deployable mobile application, and a backend matching algorithm that performed reasonably on our mock lender and borrower profiles. Our next step would be to conduct live testing, in order to confirm the various educated assumptions we made about lender and borrower behavior, and to flesh out more features on the React native app, such as live tracking of your matched borrower/lender on a map. The live testing will most likely lead to challenges with redesigning of the algorithm, and its factors/weights.

Going through the full process of ideation to production of an integrated application taught us various lessons around teamwork and full stack development. We learned how difficult it can be to decide on a single project idea that satisfies the personal interests and motivations of all our team members. We learned how to confidently dive into frameworks that we've never used before, such as SimPy and React Native. Finally, we learned that the integration process is always more complicated and time consuming than it might initially seem. These are only a few of the many things we've learned through this project, and we're grateful for this senior design experience!

Business Report (M&T)

Stakeholders and Value Proposition

Sharify has three key stakeholder groups: 1) our customers, the closed community members who will serve as borrowers and lenders on our platform, 2) the organizations behind the closed communities that Sharify will operate in, and 3) our investors. The value proposition of the Sharify platform to its users is two fold: on one hand it provide convenience, and immediate, effective access to daily items needed most urgently for the borrower; on the other, it provides an opportunity to generate greater item utilization out of infrequently used items for lenders, whether it's for direct monetary value or some currency on our platform. At the same time, the organizations behind Sharify's selected closed communities - schools, company offices, HOAs (Homeowners Associations), and more - will benefit from a more social, unified community that is bound to improve the organization's culture. Finally, we will ensure a large ROI to our investors by implementing the planned revenue streams of retail partnerships, paid subscription offerings, and take rates on high value items.

Customer Segments

Our initial target customer segment is the typical college student: forgetful, on-the-go, studying and meeting deadlines, frequently looking to save money, and always valuing convenience. They stay up late into the night, when stores and restaurants on campus aren't open. As an informed, environmentally-conscious Gen Z student, they are also seeking to achieve more sustainable consumption habits. The counterpart of this target customer has a similar Gen Z profile, and is looking to earn value off of unused in their college apartments (textbooks, cameras, kitchen appliances).

Once we have captured the college student market, we plan on targeting other closed communities with similar stakeholder pain points and interests: 1) corporate offices with young, busy professionals on the go, and 2) suburban neighborhoods with young parents or general residents, and 3) rural neighborhoods and senior living centers.

Market Opportunity

We estimate our serviceable addressable market, which are college campuses, to be around 20 million users. According to US News, in 2020, there were 3,982 colleges and universities in the US. On average, there are 6,354 students per college campus, yielding a total serviceable addressable market of 25.3 million users. Our total addressable market consists of these students in addition to corporate office settings, and homeowner associations in the United States. Given a U.S. civilian labor force of 164 million people, and a WSJ calculation that 36.2% of US employees work at a large or very large organization (2.5K employees or more), corporate office settings should add another 59M users. Finally, HOA-USA identifies 40 million households, so assuming one Sharify account per household, that yields a total addressable value of 40 million users from HOAs. While these calculations aren't perfect, as they don't account for overlap between the labor force and HOA residents, they still prove a massive potential TAM. We estimate our market growth rate to be in the healthy double digits. There are not many CAGR estimates for the item rental market, since it's a market that Sharify will be creating. Therefore, given other reference points - the online clothing rental market's 25% CAGR, GoPuff's 60% user growth rate, and more - we expect a healthy double digit growth rate.

As another reference point, the Buy Nothing Project, which is a global network of community-based groups founded in the US that encourages giving of consumer goods and services in preference to conventional commerce, already has 4.3 million users just across its Facebook groups. Given Sharify's similar value proposition and emphasis on sustainability and reusability, this is a good benchmark on how many users we can feasibly achieve in the first few years of operation.

Competition

We are confident that Sharify will be able to capture these users thanks to its unique positioning relative to existing services. There are other online rental marketplaces and shared economy platforms that facilitate rental. Uber and Airbnb are the most apparent ones that have pioneered this field: finding economic mismatch between supply and demand of certain common utilities–cars and houses–and efficiently matching the two to close the gap. Rental marketplaces for specific item categories like cars (Turn), dresses (Rent the Runway), appliances (Direct Appliance Rental), furniture (Cort), or high-end equipment/electronics (Fat Llama) do exist as well. However, while their main value proposition is to facilitate the transaction the renting or buying of specific items, a core value proposition of Sharify is to be able to service on-demand item sharing and help borrowers specifically tap into the supply right around them, to maximize reusability. Sharify not only helps users find the item they desire, but in a way that maximizes sustainable consumption for the community as a whole.

Costs and Revenue Model

Building out Sharify's offering will require various upfront costs: R&D costs to develop the Sharify algorithm (which will be Sharify's key IP), cloud infrastructure costs to keep up with user growth, marketing costs for targeted advertising, community organization partnerships, referral programs. Our revenue model should ensure that we generate a high return on these costs.

We are currently considering three main revenue models. In the first, Sharify offers users targeted purchase recommendations based on frequently borrowed items and partners with third party retailers to match users to those suppliers, facilitating the transaction on Sharify and taking a share of the revenue generated per transaction. After all, if a user borrows an item enough, perhaps it does become more economical and sustainable to buy. The second is a subscription to the service itself where users pay a monthly fee to access the platform. Every transaction is free and insurance is guaranteed up to a certain monetary value. Under this model, we would offer a free trial period to capture users. In a third model, retailers pay to act as business lenders on the platform in order to introduce their product or service to the Sharify community as a trial (i.e cannot redeem from this merchant multiple times). For example, iRobot could be launching a new floor cleaning robot, and they advertise via Sharify for users to check out such a robot for 24 hours and then return it.

We currently find the third model to be the most attractive for the initial deployment of Sharify. Currently, retailers of high-value physical goods (e.g Bose, NIkon) have no feasible method of offering low-friction free trials to college age students. We view this as an issue, as there is a large percentage of the market that is somewhat intrigued by these retail offerings, but does not have enough conviction in the product to make an expensive purchase. Retailers, especially those with technology products, have been increasingly targeting younger demographics, making college-aged students especially important to their marketing efforts. We believe that a free-trial offering through Sharify would be incredibly attractive to these retailers if we were able to amass a sufficient user base. Of course, beyond the monetary contribution, adding retailers to Sharify also widens the set of product offerings on the platform, bettering customer experience and attracting more users to the product.

Appendix

Figure 1: Matching Algorithm

When a borrower submits a request for an item, lenders who own that item are assigned a score based on their distance from the borrower, how frequently they lend on the Sharify platform, how quickly they normally respond to Sharify push notifications, and other factors. The lenders are ordered from highest to lowest composite score and split into batches. The first batch is pinged with the borrowing request, and if no one from the batch replies within a few minutes, the next batch is pinged, etc.

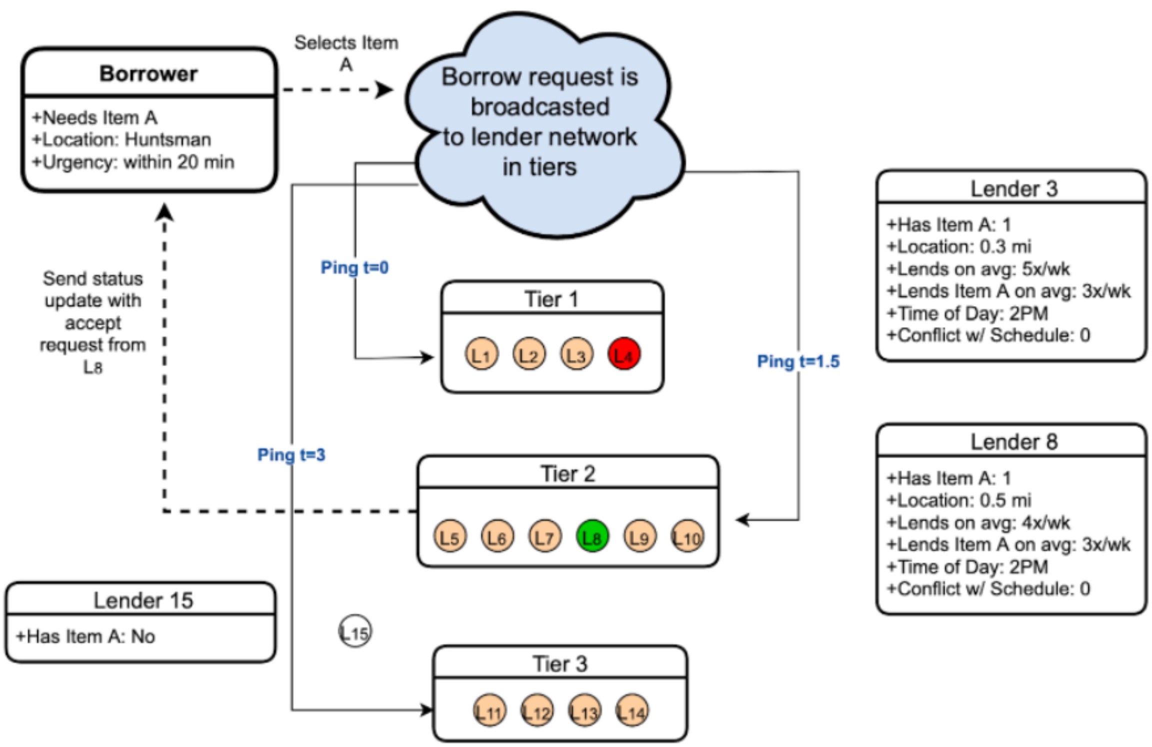

Figure 2: Simulation Results
The following graphs show the supply and demand over time (left) and surplus (right) of three sample items over the course of a simulation run.

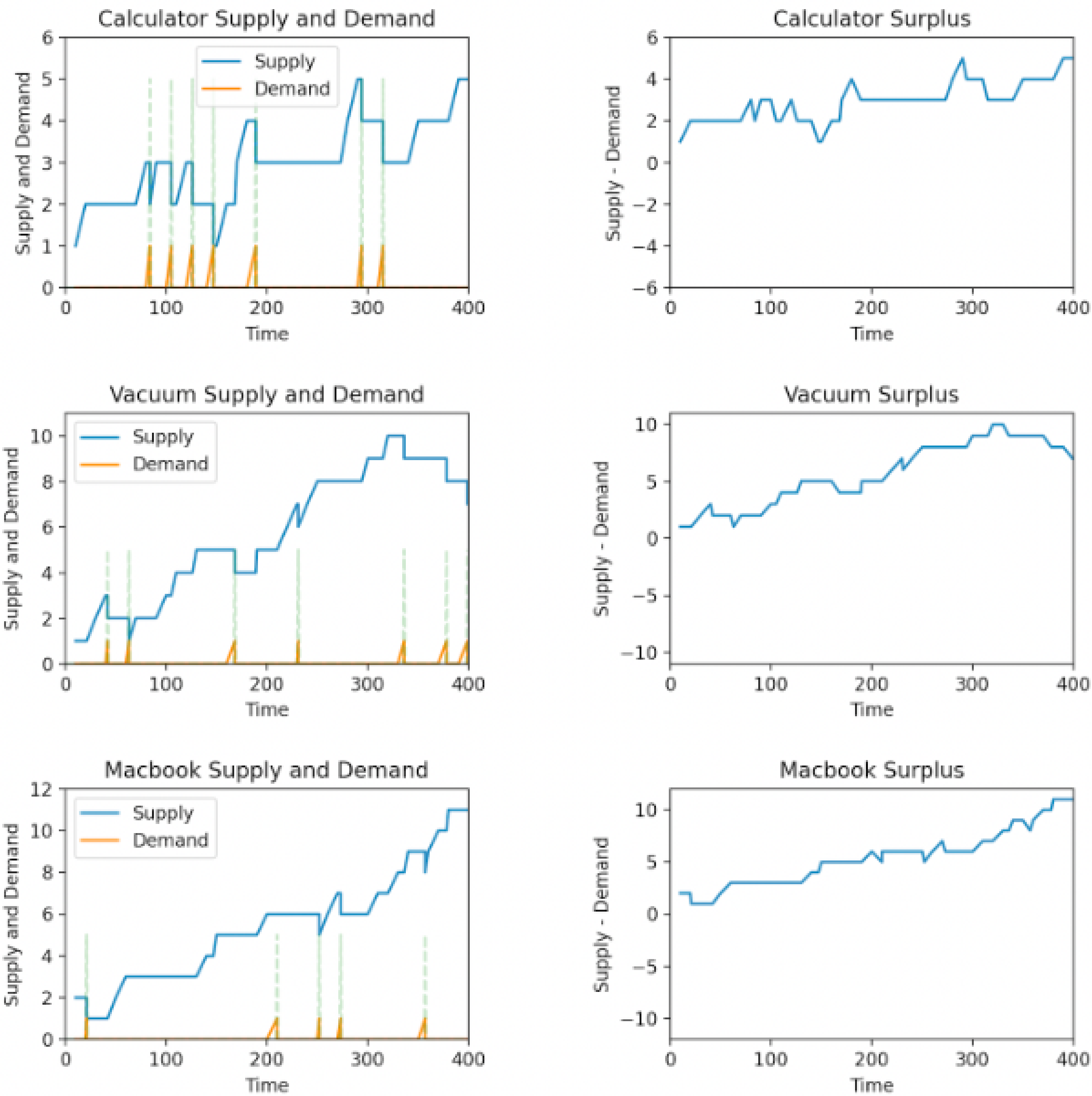

Figure 3: Service diagram

A flow chart demonstrating the back end services that our React Native application will connect to. For now, the application uses Firebase user authentication, and Flask REST APIs to connect with our Firestore database and the deployed matching algorithm.