Goal-laborator: A Socially Collaborative Wellness Platform

Wellington Lee
wtnlee@seas

Nishita Jain
nisjain@seas

Pranjal Goel
gpranjal@seas

Shirali Shah
shirali@seas

Abstract

Goal-laborator is a web-application designed to motivate people to achieve their own goals as well as goals they choose to collaborate on with their friends by using social accountability as a motivational tool. The app focuses on college students to help students manage their time, achieve the goals they set for themselves and find new interests. The application improves the overall wellness and mental health of students, a well-documented issue across college campuses across the nation. The application was carefully analyzed to explore and mitigate potential negative social impact. The quantitative performance of the application in both the back-end and front-end was empirically analyzed to optimize the user experience while qualitative feedback was directly obtained from the users.

1 Introduction

1.1 Product Motivation and Value Proposition

The main motivation for creating Goal-laborator was derived from a clear need here at Penn where many students including ourselves have a large variety of goals they hope to achieve, ranging from academic, social, professional and to creative. We observed that these goals were often ignored or discarded due to fundamental issues such as over-packed schedules or a work-hard school culture. We hoped to create a product that would allow high-functioning and busy individuals to carve out time to focus on **creating** and **maintaining** their own forms of wellness. A link to the deployed Goal-laborator application can be found at https://goal-laborator.herokuapp.com. Also, a video demo is linked here too https://youtu.be/CEemC7cZgZg.

1.2 Stakeholders

Our application is primarily meant to be used by college students. We chose this group because we understand this cohort's problems and needs personally. A thorough description of all relevant stakeholders can be found below.

- College-aged Students: The most important stakeholders of our product are college-aged students (the primary users). After analyzing the impact of a college lifestyle, we realized that students often have a tough time remaining motivated for their goals due to academic, professional, and social stresses. College students tend to have goals within the following areas: fitness, learning, hobbies, and creativity. These are goals that help students focus on their well-being. We think that this application helps address all these goal categories and helps college-students achieve a healthier life-style by collaborating with their friends.

- University administrators: In the future, we imagine that university administrators should be involved in helping work with their student bodies. Universities care about the mental health of their students; therefore, admin could encourage the use of this application because it might benefit a large number of students.

- Businesses: Companies and brands on a local, national, and international scale can interact with the Goal-laborator. The business incentive is that they will be able to sponsor global goals. For instance, Nike could sponsor a global goal of everyone running every week. These sorts of goals serve as Corporate Social Responsibility which is important for a company's public image.

1.3 Functionality

At a very high level, our app provides the functionality for users to create goals that can be **public** or **private**. Users are able to add friends on the platform and see their friends' public goals and collaborate on them.

A user can create an account and login to the web-app. After login, there are four main tabs that the user interacts with:

1. **For You:** The landing page of our post-login app shows the goals that the user has created or has chosen to collaborate on. They can also see any friend requests they have received or sent on the same page. They are able to check-in to their goals to signify they have performed the action that day and are able to mark goals completed once they are fully achieved. Every check-in in on a goal gives points to the user (2 for collaborations, 1 for private).

2. **Profile:** The user is able to see their basic profile information on this page along with their friend list and goals they have created. There is also a search bar to search for and add other users as friends. The point system on this page is used to help users stay motivated.

3. **Create Goal:** This page is a form that allows users to input and categorize their goals. The categories available are: Creativity, Diet, General, Hobby, Physical, and Wellness. The user is able to customize the privacy settings, descriptive information and weekly frequency regarding this goal through this interface.

4. **Explore:** This page displays a user's potential collaborations. On this interface, friends' public goals are displayed by category. Users choose the goals they wish to collaborate on and this page is designed to be the safe space where people broaden their horizons.

2 Related Work

2.1 Gamification as a Motivator

A 24-week study took 602 overweight adults and used social incentives to encourage activity by gamifying exercise. One group of the participants were involved in a supportive game where a sponsor would see if they met their goals each week. Another group was involved in a collaborative game where teams were allocated points based on the performance of a randomly chosen member of each team. A third group was involved in a competitive game where teams compare performances weekly through rankings. The last group was a control group that did not gamify exercising. The study found that the collaborative group increased their activity by 637 steps per day when compared to the control group, the supportive group increased by 689 steps per day, and the competitive group increased by 920 steps per day (Mitesh S. Patel, 2019). The competitive gamification notably increased physical activity in the participants compared to the other forms of gamification. Additionally, after the researchers ended the games but encouraged continued exercise, only the competitive group was able to noticeably maintain an improved level of physical activity over the next 12-weeks of monitoring (Mitesh S. Patel, 2019). This study shows that not only did competition result in the highest growth of physical activity, it also had the most effective lasting impact, supporting how well competition operates as a motivator. Our application is going to create, encourage, and make permanent habits and goals by developing a point system to induce competition.

2.2 Accountability Through a Network

A study conducted at the University of Pennsylvania tried to evaluate the role social media can play in motivating people to exercise more. In the experiment, a group of students were enrolled in exercise classes. Some of these students received promotional messages and another group was placed in a social network with six other peers that could track their network's progress on a website. The study found that the promotional messages resulted in a spike of exercise in the beginning, but the behavior did not stick throughout the 13-week study. The social network participants, on the other hand, showed a substantial increase in enrollment as time passed, showing how network effects are able to motivate the students more effectively (Jingwen Zhang, 2015). Social influence has always had tremendous power over how people behave, but consistent positive reinforcement is also extremely important for motivation. Since the peer networks only shared positive behavior on its social platform, there is a consistent source of positive reinforcement, encouraging even more

exercise. Our application will leverage this phenomena by creating a network of friends that is inherently positive. Users will be able to see their friends' updates and accomplishments. This is a source of positive reinforcement that will increase the permanence of their goals and habits.

3 Business Fundamentals

3.1 Competition

Some of the direct competitors that app-users use for habit-tracking and goal-setting are:

HabitShare is a habit-tracking application that includes social features like messaging and sharing goals within a network of peers, but the social interface did not allow users to collaborate on public goals in their network. The application also tracks streaks on goals but in the case where streaks are broken, all prior progress is lost. (Hab, b)

Habitica is an application that gamifies accomplishing goals through assigning rewards for task completion. Habitica also supports collaborative goals, but it forces these goals to have the same users every time. Progress is nicely tracked through an avatar that levels up with more accomplishments, but there is no centralized news feed to view other users' progress. (Hab, a).

Streaks: is a task-list application that helps users form good habits. This application tracks personal user progress very well using streaks as motivation, but it does not quantify a user's overall performance across all goals. This application completely lacks a social element; there is no network of peers or public goals, forcing users to only hold themselves accountable. (Str).

Goal-laborator aims to take all the effective features of each of these applications and add missing features to create a consolidated application that meets the functional needs of all users. This application will have social features like a network of peers and publicly view-able goals, but it will also allow peers to collaborate on goals within their networks. Instead of tracking progress through streaks, Goal-laborator uses a point system so that previous progress remains view-able and to gamify the application, providing incentives to users to compete and collaborate. This application also includes an 'Explore' page that provides a centralized feed of friends' goals in order to stay updated and to encourage more collaboration.

3.2 Market Sizing and Growth

The specific market for social goal tracking is largely uncharted, but we can evaluate adjacent markets to provide important insights about what the relative size of this market might be. The two markets to consider are habit-tracking/self-care and fitness-tracking. MarketWatch (Pesce) found that in 2018 the largest growing categories in iOS was that of self-care. This included meditation and habit-tracking apps such as Fabulous - Motivate Me. More and more consumers are taking charge of their self-care and we can see this reflected in growth trends across the market.

Younger groups are notably more involved with self-care than older generations. NPR found that millennial generations and those that followed are twice as likely to be focused on self-care and wellness such as dieting and working out. Given the push that younger generations have in this segment, we can see the revenue growth of self-care apps, which is potentially as high as 170% Year-over-Year (Silva).

We also think that the fitness tracking industry is relevant, which has exhibited a CAGR growth of almost 17% (Intelligence). Most importantly, this market has seen a rapid growth of socialized-fitness primarily because it has recognized the network effect benefits of motivation. These sort of applications allow for a safe space where socialized goals can live separate from the fakeness and toxicity of existing social media sites such as Facebook or Instagram.

3.3 Consumer-Side Costs

This will be a completely free application for users. We stand by this decision because the application is meant for social wellness. We don't believe that anyone should be stopped from becoming mentally healthier because of cost-constraints. Therefore, there will only be one free-tier version of this application; everyone should have an equal right to fulfilling their goals!

3.4 Businesses and the Revenue Model

As we have mentioned before, our primary source of revenue in the future will be businesses posting public goals that all users can see. For example, Nike might have a goal related to running or Sweetgreen might have one for eating salads. Businesses will only be allowed to post positive goals; these will be thoroughly vetted before be-

coming publicly available. All users will be able to see immediately that these explore-page goals are Sponsored for full transparency.

In terms of the revenue model, we have decided to follow the gold-standard for social-media based advertisements by using an approach similar to Facebook. We'll employ an algorithmic ad-auction that allows a competitive rate for businesses to interact with the application. This auction will be based on factors such as geographic reach of the goal, wellness potential with the goal, and target customer segment. We believe this ad-auction method is important because as our product scales, we can't have a static price per impression. This auction should be implemented using an automated algorithmic and all details should be accessible through a businesses portal on our application. We do want to set a general range for impressions. We will on average hope for a cost per thousand impressions of $15. Additionally, we will also charge based on every engagement (collaboration) on the goal at an average of $1.

Of course, this has the potential to detract from the application's primary purpose. However, we're going to do everything in our control to limit any negative consequences of including business interactions. Specifically, these goals will only be shown on the Explore page which is not the first (default) page available to users. Additionally, at any given time, we'll show a maximum of 3 sponsored goals on the page. Lastly, by thoroughly vetting every single business goal, we hope that we'll better user mental health by providing new options to engage with CSR-motivated goals.

4 Technical approach

4.1 Technical Stack

During the beginning of the academic year, we had decided to pursue an iOS application built for iPhone and iPad. However, we got a significant amount of feedback that a mobile application would encourage people to spend more time on their phones and mobile notifications could be intimidating (or even distracting). So, in late January, we actually decided to pivot and build a web application; this primarily required changing the front-end with limited structural changes to the back-end.

After our pivot, we chose to use a MERN+G architecture, using MongoDB, Express, React, NodeJS, and GraphQL. Each decision had pros and cons that we outline below. For the code base, please reference https://github.com/wtnlee/goal-laborator-deploy.

4.2 Express and NodeJS

For most modern web applications, NodeJS is the web-framework of choice primarily because it allows event-driven I/O connections that are also single-threaded. This allows for increased efficiency and scalability. Additionally, the package manager that supports NodeJS development (npm) has many developer tools that made it easier for us to leverage the power of well-tested and optimized libraries. Express is the standard server library that most developers prefer to use with NodeJS. Finally, we also decided to use Express given our team's existing familiarity with Express features.

4.3 MongoDB

For the database, we considered multiple options, including DynamoDB and MongoDB. After looking into the features of both databases, we decided to go with MongoDB. DynamoDB is a deployed database, which would have made local- and unit-testing extremely difficult. Additionally, DynamoDB's primary querying mechanism is through simple key-value searches; however, given our GraphQL structure (described later), this is extremely cumbersome and we needed the ability to query based on a complex host of parameters. Additionally, given how interrelated our data is, DynamoDB's lack of foreign keys and links between tables was a clear disadvantage.

Given all of these problems with DynamoDB, we found that MongoDB addressed man of these concerns. MongoDB's collection and document structure is extremely flexible; documents can have different fields and attributes present without causing consistency issues because they are retrievable in a JSON format. Additionally, MongoDB allows for querying input parameters which would enable our application to search for very specific documents. Given that this is an application that could quickly grow to many users given network effects, it is also important that MongoDB is very scalable.

After deciding on MongoDB, we created 3 main models: User, Goal, and CheckIn. The User model keeps track of all of an individual user profile details, a user's friends, user created goals, and any collaborated goals. This structure allows for an almost relational structure where documents

point to other documents using an ObjectID. The MongoDB schemas are included in the Appendix.

4.4 GraphQL

The past 10 years have brought new changes to routing APIs and as such, we wanted to make the best decision about how we would set up our client-server routes. The two options we considered were REST and GraphQL. While REST is still far more popular in most applications, GraphQL has gained popularity because it provides a well-defined interface between the client and server. A GraphQL server is extremely transparent and implements exact formatting when sending back responses to the client, allowing the client to predict nested JSON structures. This was an important factor for us to consider because we query deeply nested and related data from the server on almost every client interaction.

The main performance indicators that we evaluated GraphQL on were the following:

- Request Round-trips: GraphQL allows for just one single request on page load by leveraging nested responses. For example, to load the public goals of a specific user's friends, REST would require that we implement and add individual routes for each part of that request.

- Overquerying: GraphQL lets clients specify exactly what information is needed from the database. For example, to load the titles of public goals of a specific user's friends, we don't need to grab the entire user object or goal object (only the relevant parameters).

We defined our GraphQL Schema early on, and iterating on it throughout. This Schema is available on exactly one endpoint, the GraphQL endpoint; we use the express-graphql library to assist in setting up this endpoint.

Clients are able to issue two types of requests, Query and Mutation (Appendix A1), through a JSON string in the request body. A Query consists of all possible requests that do not write or update any document in the database; for example, when the client requests for Dashboard goals, this is a Query. On the other hand, when the client wants to write or update data, they issue a Mutation request; for example, when the user wants to create a goal, the request would be a Mutation instance.

One major issue with GraphQL is that it potentially overextends the database. GraphQL resolvers execute independently; in other words, the nested structure means that a resolver executes for every single field in the query. This can often mean that a resolver runs duplicate queries to the database. Because we run our MongoDB instance on the cloud, every request to the database can increase latency. We utilized caching to help reduce this problem as much as possible (more on this in the Evaluation section)

4.5 React.js

For the front-end, we decided to use Facebook's open source library React. One of the major benefits of React is the JSX language. This made managing the top level code simpler without having to worry about deeply nested components. Moreover, React also promotes re-usability. A lot of our applications relies on the repeated rendering of certain components such as a Goal-Card used in different contexts. The JSX language and the system of creating a bunch of components that have implicitly individually managed states and user defined event handlers made our front-end development experience more painless.

React also increases the speed of rendering of the web-page, one of the critical measures of success for a web-based application. The open-source library essentially optimizes the rendering of the DOM and makes it so that the client browser minimizes un-necessary re-computations upon any updates.

5 Evaluation

5.1 Quantitative Evaluation

Note that for the following section, there is a wide distribution of times solely because networking relies on a lot of different components in the Internet. There are a lot of moving parts in each API call, from hitting our Heroku server to hitting an Azure MongoDB deployment. Any jitters in network connection, packets being dropped or TCP connection status will lead to natural variation which is what is shown in the Appendixes.

Throughout the development process, we have been focused on evaluating our technical decisions to make sure that we have an efficient application. There were two main areas we focused on: 1. Server-side caching and 2. Asynchronous frontend requests.

5.1.1 Backend

As mentioned previously, with GraphQL, there was a fear of overextending the database by issuing too many requests. This increases the time that it takes for the server to send a response back to the client after a request. So, we discussed the possible solutions for this and decided to use caching/batching to limit database over-extension. The main caching that we used using the Facebook open source Dataloader library. This library gives a simplistic approach to caching that reduces the time is takes for resolvers to execute. We decided to measure the time it takes for a GraphQL query request to complete and how long it takes with and without caching. We used the Apollo Tracing library to track the performance of our improvements.

We found that caching improves the average query time from 347.2 ms to 172.6 ms, a 50.2% speed up. A visualization of the distribution of run time pre- and post-caching can be found at A2 in the Appendix.

5.1.2 Front-end

On the front-end, we noticed that we were issuing large queries to the back-end on content-heavy pages such as the dashboard and the explore page. We specifically evaluated the front-end on the key metrics of full-render time and load-balancing. We measured load-balancing by calculating the time difference between the route that took the longest to return and the route that took the least time to return. The underlying theory behind this is that parts of the page loading quickly while others load slowly results in a displeasing and uneven user experience. Load-balancing also reduces the amount of time it takes for the web-page to fully populate. It has been shown that long render times negatively impacts a company's bottom line as well as consumer engagement.

To collect our data, we experimented with a bunch of different payloads (various sizes) that are used to populate our explore page. We created a baseline where we requested and received all of the necessary data in one API call. We then created a set of variations where we broke up the requests into several API calls. We empirically measured how long the application took to render using the Network waterfall graph that is built into inspect element.

The results of this experimentation are shown in Appendix A3. The end-result is that we managed to deconstruct the explore page into 4 load-balanced queries that balance the trade-off between too many sequential data-base queries in a route and routes that are too slow to return. Empirically, we found that this breakdown resulted in a 45.6% improvement over the baseline, from 400.78 ms 217.9 ms with an average load-balance spread of 47.8 ms.

We show a plot of the render time and load-balancing metric in Appendix A4. for the baseline of acquiring all the information in a single route and our best-effort optimized version of splitting the render into 4 concurrent queries.

5.2 Qualitative Evaluation

5.2.1 Beta Rollout Evaluation

Because of the COVID-19 situation, it was difficult to on-board a large number of users for testing and feedback, so we rolled out the Beta version to a limited group of people which included family members and Team 11: Spry. We surveyed these users for feedback on the user interface and experience of the web application. We also used the survey to get a self-assessment of how effective Goal-laborator was in motivating them to set and complete goals while keeping the focus around wellness. Of the users that tested our application, 91.7% collaborated on goals (Appendix A5) and 66.7% found that the collaboration increased their motivation, while the remaining felt neutral (Appendix A6). Every single one of these users saw an increase in their goal achievement since joining the application (Appendix A7). The social aspect of the application motivated about 66.7% of the users, while the remaining were either neutral or only slightly intimidated (Appendix A8). Of the users that did take advantage of the collaborative abilities of the goals, all of them expressed an improvement on their general wellness (Appendix A9). A majority of the feedback we received was to further expand the capabilities of the application by facilitating more collaboration through the user interface and validating collaboration by tracking friends' progress, but overall the survey indicates that Goal-laborator was able to improve user wellness and motivate users through collaboration and competition rather than intimidate. The user survey can be found at https://forms.gle/JzLnaPPpJJRpChMP6.

5.2.2 Detailed User Studies

Below is our plan for a detailed user study that would have run for about 2 months. We would have chosen a pool of around 100 volunteers across all schools here at Penn. We would have then randomized them across two groups. For one group, we will ask them to write down a few goals they want to accomplish at the beginning of the experimental period and send them periodic reminders to record down in a Google form how many times they accomplished the goal and when they did so. For the other group, we would have given them full access to Goal-laborator and have them use the features as intended.

The user study will likely allow us to get valuable feedback on the improvements offered by our application. The initial step of randomization across two experimental groups will help control for any heterogeneity or idiosyncratic differences between the two experimental groups.

After collecting the data, we can calculate sample averages on how frequently people are checking into their goals under a Poisson count model. We will calculate for each group, what the propensity and count is of a user accomplishing their goals on a weekly basis. We can then perform a hypothesis test under the null hypothesis that our application does not offer any significant benefits over a self-managed goal-tracking system. This will let us get potentially statistically significant results (at a 0.05 or lower confidence level).

Moreover, by collecting data on when people are checking into their goals, we will be able to glean on an individual level what drives people to complete or fail to complete their goals. We can analyze this data to make decisions on how we want to improve our application from a UI/UX perspective (eg. notification timing, layout). We already have this data collected in our application due to check ins being assigned time stamps. We can also collect data on engagement levels (both frequency and recency are interesting) to model the stickiness of our app and see how much people are actually using it.

We also planned on supplementing the efficacy measures with a more qualitative survey (similar to the one described in the previous section) sent out four or five times during the duration of the period. This feedback will give us valuable insights into what is working and what is not working for our application and let us iteratively perform updates to the feature set and user interface to make the application more acceptable to our target audience.

6 Societal impact

6.1 Positive Social Impact

As alluded to in the previous section on motivation,we realized that a major problem among our peers is that they do not have much time for themselves. Due to a myriad of reasons including Penn's culture, general workload and social expectations a lot of people are unable to enjoy their hobbies, achieve wellness goals and maintain their health.

By creating a safe-space where friends are able to collectively work on achieving goals, we hope to act as a major driver of improvement on student wellness and mental health. By creating a system that induces friendly competition and promotes social accountability we hope to ensure people can create, achieve and maintain both physically and mentally healthier and productive lifestyles.

We hope the design of our application will help create additional benefits of deepening friendships and allowing students to become more well-rounded as they explore the goals and interests of those around them. However, we truly believe the structure of our application with the closed networks, opt-in only goal collaboration and the bifurcated news feed with the user's first touch point being their own goals will truly allow people to focus on improving and de-stressing themselves primarily with a healthy supplement of other people's goals if they so choose.

6.2 Negative Social Impact and Mitigation Strategies

6.2.1 Privacy Concerns

With any social media-like application there are inherent risks of privacy. The data that users input into the goal creation forms on our application is potentially extremely personal and private. While many goals are innocuous such as playing an instrument or going on a run every day, there are more sensitive personal goals that are potentially entered on our website regarding important subjects such as mental and health conditions.

While our team has tried to mitigate this by collecting minimal amounts of personal information, the stark reality is that in order to identify individual users and allow their friends to find them,

names and emails are needed. As evidenced by numerous privacy case studies (Netflix leak) a little bit of self-identifying information can allow an intelligent adversary to link sensitive information to individuals in real-life. The information can also allow for the inference of factors that make up an individual's identity such as political stances or sexual orientation. It must be noted however,that because our system operates on the public and private goal system, an adversary will not be able to view the more sensitive goals a user has created, unless they manage to break encryption or gain unauthorized access to the database. This dichotomy of the two types of goals serves as another way for our application to help maintain user-level privacy.

There are also potential concerns regarding the next steps of our application, especially the revenue model. The most pressing concern is that if companies are choosing to sponsor our application, these companies will want metrics such as user demographics, interaction data and location data regarding engagement with their advertisements or sponsored goals. It will be hard to compete with other platforms that willingly collect and monetize such sensitive data. As a result, we are planning to be firm in our negotiations with potential partner companies in the interest with remaining compliant with consumer data protection laws as well as re-doubling on our commitment to protect the privacy of our users.

6.2.2 Vulnerable Groups

We believe that there are not any targeted groups that will be marginalized by our application. The underlying premise behind the application is a network of mutual support. However, we have identified certain groups of individuals that might be particularly affected by the potential stresses of a social network. Individuals with low self-esteem may feel slighted or helpless if they see a lot of seemingly confident peers accomplishing amazing goals. This could result in processes of self-destruction and a further loss of self-esteem. Moreover, socio-economic status could also play a role in this. Some users may create goals such as going to expensive tennis lessons that could re-highlight the drawbacks associated with an individual's status quo. Ultimately, these are concerns that will occur in any sort of social network. We specifically did not include large social networks into the mix (such as a Facebook or Instagram integration for friends); this means that users can deliberately become friends with people they think will motivate them. The audience and groups we envision targeting are likely to create a supportive as opposed to toxic environment, which will likely help mitigate this risk.

Moreover, another concern is that we may be excluding people who do not have access to the Internet or technology. We believe, however, is that the modern college student spends most of their time on their laptops and their phones and colleges provide financial support to those that cannot afford it. It is simply the reality that technology is a big part of our lives. Simply put, people who insist on not using any technology are definitely not part of our target audience.

6.2.3 Cybersecurity

We see the potential of certain cybersecurity risks. Any web-hosted application has the potential to be afflicted by risks such as the database being hacked, user-information being leaked, cloud-abuse (spamming the server and denial-of-service), data loss due to hardware failure and form-jacking. While many of these things are valid concerns, we do not see them as particularly large issues especially as we are in the testing stage of development. Deploying onto AWS or Azure, with native cloud security and load-balancing built in will help mitigate these issues naturally.

In addition, we plan on taking a few other necessary precautions. For instance, we made sure that the passwords are encrypted and salted on both client and server side to ensure security. This poses a widely known risk of crackers being able to use the same passwords for different user accounts as people tend to re-use passwords. We further made sure that when our site is hosted, it has an appropriate SSL certificate and that all communication between client and server is end-to-end encrypted. Moreover, the session management that is done on the server side as opposed to the client side will help with not allowing users to impersonate each other's logins- the tokens and certificates generated are not easily spoofed.

6.2.4 Deception and Manipulation

We see the possibility of manipulation and deception from legitimate users of the applications. The application we built does not have an actual way of verifying users actually completed a goal to its

entirety or that the check-ins and updates are legitimate. A truly airtight system could necessitate mutual verification systems between friends with pictures serving as proof or geo-location data as proof of an action being accomplished. However, this raises a myriad of other concerns both from a technical and privacy perspective. More importantly, such a system adds a large degree of complexity from the end user's perspective and creates additional barriers to usage that will negatively impact the stickiness of our app. As a result, we decided to rely on the fact that our social accountability from a close friend group will ensure users stay honest and that the target audience of the application is unlikely to engage in subterfuge.

Another source of potential manipulation is that people may make unwholesome goals on the application. We implemented a feature that buckets goals into categories such as Hobbies or Wellness to serve as a reminder to the user that the goals on the platform should be inherently positive. This, however, does not prevent or disallow for toxic and negative descriptions of the goals. A natural solution to this that we discuss in the next section is using Natural Language Processing to tag and disallow goals that do not fulfill a set of positivity criteria that we laid out.

6.2.5 Unintended Consequences

One of the long-term unintended consequences we envision could stem from Goal-laborator mirrors the fate of popular social media platforms today such as Instagram and TikTok. These platforms suffer from the creation and idolization of influencers (such as Charli D'Amelio on TikTok and Kendall Jenner of Instagram) which leads to a community based on a culture of toxicity and a never-ending goal of achieving a sort of "perfect life." While the core functionalities of Goal-laborator focus on the interaction between close friends, our planned features regarding more global penetration and sponsorship could inevitably lead to fan-out that could spiral into a culture of toxicity.

We have attempted to mitigate this issue through a number of features. Specifically, by using bi-directional friendships, we make it much more difficult for influencers to rise. A collaboration requires both individuals to press accept or add friend. Apps that have created cultures of idolization (TikTok and Instagram) use a unidirectional follow feature and we hope the bi-directional feature will help prevent this. Another feature that speaks to this problem is the creation of private goals. Theoretically, an individual could use the application to solely keep track of his or her own goals without worrying about people or viewing other people's global goals. This feature will also help slow down the proliferation of idolization.

Another unintended consequence we envision could occur is that Goal-laborator uses gamification and competition between friends to help achieve social accountability. This raises the potential problem competition could work in direct opposition of Goal-laborator's fundamental goal of wellness. Competition can brood ill-will and can induce significant amounts of stress in users. We mitigated this risk by minimizing the amount of notifications (as evidenced by our pivot to a web app as opposed to mobile app) to reduce stress and try to encourage users to only use this app with their close friends. The hope is that anyone using our app has wholesome motivations to empower their friend group to do more.

Third, another unintended consequence that could occur is that we could see goal homogeneity within friend groups. This could potentially be an issue as friends converge and collaborate on very similar goals which could work directly in contrast to our goal of helping people become more well-rounded. To help mitigate this potential pitfall, we have ensured that the landing point of the application is always the "For You" page. A user always sees his or her goals first before seeing any friends goals. This will force the user to focus on his or her existing commitments first and put their own interests and goals first.

To further mitigate this issue of homogeneity, our "Explore" page requires a physical opt-in in order for the goal to show up on the "For You" page. Pressing the button requires a sort of commitment, and by displaying all potential collaborations, we allow users to have an unbiased and holistic view of what is out there.

7 Final Thoughts

7.1 Discussion and Lessons Learned

7.1.1 Story-boarding and UX Design

This project helped us appreciate the importance of story-boarding in the process of UX design. Story-boarding allowed us to quickly narrow down the needs and use cases of our intended users and

figure out what was the overall objective of the app. Story-boarding is a cheap and easy way of eliciting requirements from the end users as well as establishing a general direction for development without requiting too much code overhead. We started out on paper and then moved towards prototyping in Figma. As a result, we believe we saved a massive amount of time and effort later on by going through the entire story-boarding and iteration cycle before finalizing and then creating a finalized user experience.

7.1.2 Database Design and Refinement

One of the biggest lessons we learned as a result of this project is the importance of good database schema planning and management. One of the issues we ran into early on was that we did not have good communication between the back-end and front-end in terms of what queries and information we need to extract and display as well as what frequency. This led to a constant reworking of schemas which would hinder our progress as we would have to go back and add things or change queries around. Additionally, a poor initial design of the schema also resulted in our having to implement joins in MongoDB when we needed to get the information out of multiple datastores which increased the computational complexity and increased the response time of our backend. A better schema, which we moved towards the end, helped reduce these overhead costs and resulted in a more seamless application.

7.1.3 Cross-Origin Resource Management and Session Management

Another issue we had was the interaction between Cross-Origin Resource Management and Session Management. With improper management, sessions would not take hold- a new session would essentially be created on every single API call which made server state impossible to maintain.

This issue was tricky to debug as it made us do a deep-dive into how different web browsers work. For instance, Firefox and Chrome have a laxer security policy- they let cookies be set from cross-origin sources. That means, if you make an api call to sampleclient.com and sampleclient.com re-routes you to sampleserver.com which sends a packet telling the browser to set a cookie, the two browsers will accept the cookie. As it turns out, this leaves the user open to a cybersecurity issue- cross-site scripting. Browsers such as Safari prevent that from happening. The original issue stemmed from the separation of the client and server on two separate Heroku deployments couple with HTTP header issues. Getting around this issue required a deep dive into managing HTTP headers on each API call we made, as well as the eventual decision to make static React files statically delivered from the same server as the database.

7.2 Moving Forwards

Our vision for the application was unfortunately cut short by the onset of the global COVID-19 Pandemic with everyone being sent home and unable to work together in person. As a result, there were a couple of features that we hoped to implement that are the logical next steps of the application we deployed.

7.2.1 Competition

In future iterations of this application, we would have liked to improve the competition aspects of Goal-laborator (according to user feedback) to increase the odds users complete their goals. It is important to note that we need to find the strict balance between friendly competition and causing undue stress on our users.

Specifically, we are hoping to implement a progress bar on collaborated goals that displays the progress of all of a user's friends on a goal. This progress bar would require a user-click to become visible as opposed to an immediate render, which should reduce the stress on the user.

We also plan on implementing badges that show up on a user's profile from a both a personal and public view. These badges can be broken into tiers and act as rewards for people checking in and earning points. By using badges as opposed to an explicit leader board, we are hoping to create a sense of competition and reward without explicitly ranking users in a stack ranking system. Such stack ranking systems used in workplaces have been shown to cause undue stress and reduce performance.

7.2.2 Natural Language Processing

Another issue that we were thinking about is what happens if an adversary decides to make negative goals on their profile. To help solve this issue, we were considering implementing a hybrid of rule-based and automatic sentiment analysis to flag adversarial goals.

The approach we first considered was training a neural net on open-source data-sets to build a Natural Language Classifier. In its most basic form, basic sentiment analysis (Stanford has an open source pre-tagged database) could give us a probabilistic measure on how negative or positive a goal is. Note that this could produce some false positives as some perfectly acceptable and wholesome goals geared towards self-improvement may have negative language or sentiment. To rectify this would require us to make our own labeled data. However, for a proof of concept, the pre-labeled data is sufficient.

We can couple this Neural Net with a rule-based system that flags some inherently negative sentiments and words. Goals that display a high proportion of negative words can be flagged and can catch any errors the neural net makes. A problem with this rule-based system is that it is likely to be incomplete. Constant fine-tuning and updates to the list of negative words or diction/syntax that define negative concepts is expensive.

7.2.3 Lightweight Notifications

We also hope to implement a lightweight notifications feature that will occasionally nudge users to complete goals if they are falling behind schedule. These nudges are psychologically important to help people stay on track of their goals. However, we want to minimize the presence notifications as to not cause stress. A good way to achieve this would be sending digests or goal-progress summaries every other day to keep users updated on their progress and remind them to check-in.

References

a. Habitica.

b. Habitshare.

Streaks.

Prescient Strategic Intelligence. Wearable fitness trackers market overview.

Sijia Yang Damon Centola Jingwen Zhang, Devon Brackbill. 2015. Efficacy and causal mechanism of an online social media intervention to increase physical activity: Results of a randomized controlled trial. *Preventive Medicine Reports*, 2:651–657.

Joseph D. Harrison Mitesh S. Patel, Dylan S. Small. 2019. Effectiveness of behaviorally designed gamification interventions with social incentives for increasing physical activity among overweight and obese adults across the united states. *JAMA Internal Medicine*, 179(12):1624–1632.

Nicole L. Pesce. This was the hottest app trend of the year.

Chrstianna Silva. The millennial obsession with self-care.

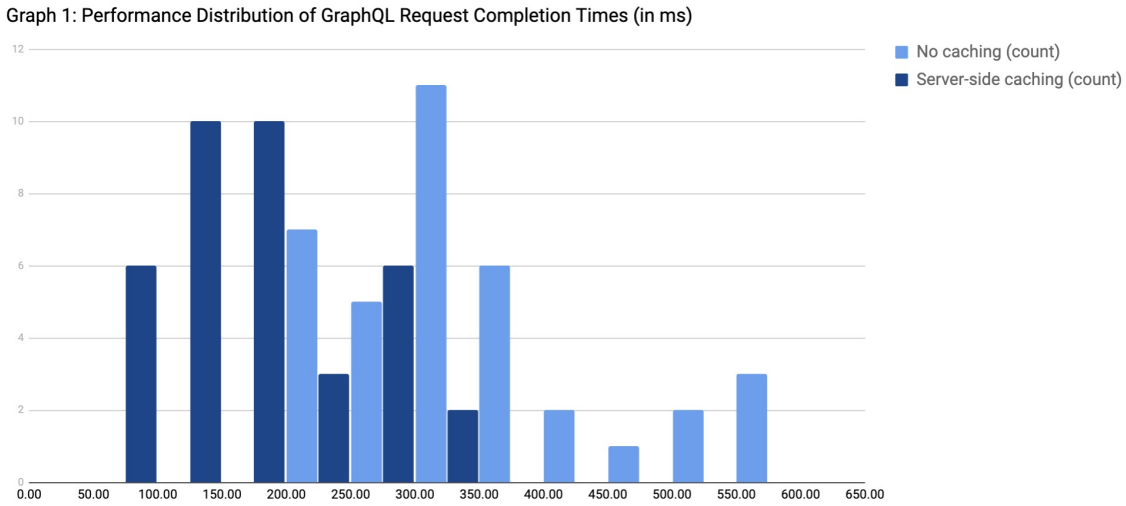
A Appendices

A.1 GraphQL Schema

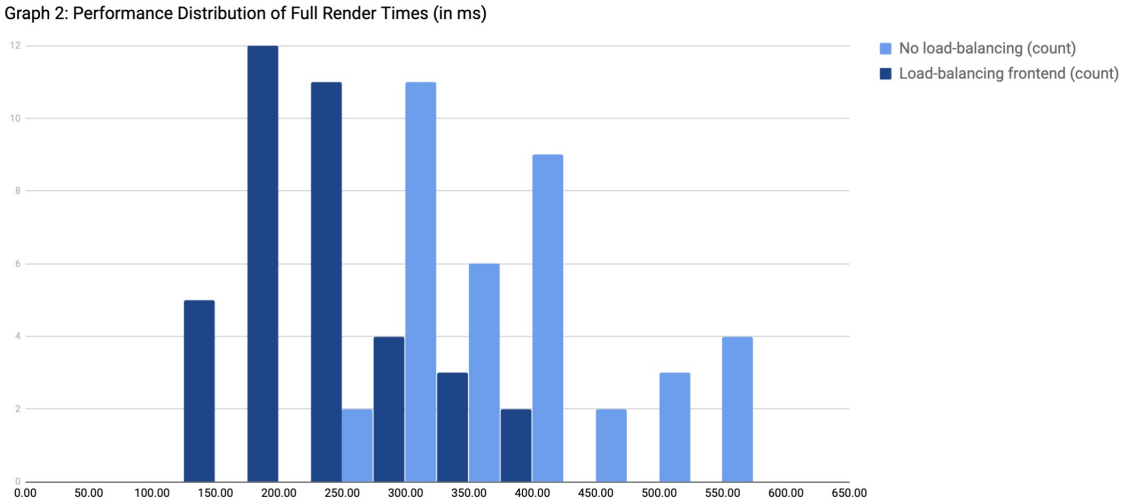
```
type Query {
  login(email:String!, password:String!): AuthData!
  me: User
  myDashboardCreatedGoals: [DashboardGoal!]
  myDashboardCollaboratedGoals: [DashboardGoal!]
  userByEmail(email:String!): User
  myExploreGeneralGoals: [Goal!]
  myExploreWellnessPhysicalGoals: [Goal!]
  myExploreHobbyCreativeGoals: [Goal!]
  getFriend(friendId: String!): User
}

type Mutation {
  createUser(userInput: UserInput): User
  createGoal(goalInput: GoalInput): Goal
  completeGoal(goalId: String!, completedGoal: Boolean): Goal
  collaborateOnGoal(goalId: String!): Goal
  goalCheckIn(goalId: String!): CheckIn
  likeGoal(goalId: String!): Goal
  sendFriendRequest(friendId: String!): Boolean
  updateFriendRequest(friendId: String!, accepted: Boolean): Boolean
  removeFriend (friendId: String!): Boolean
}
```

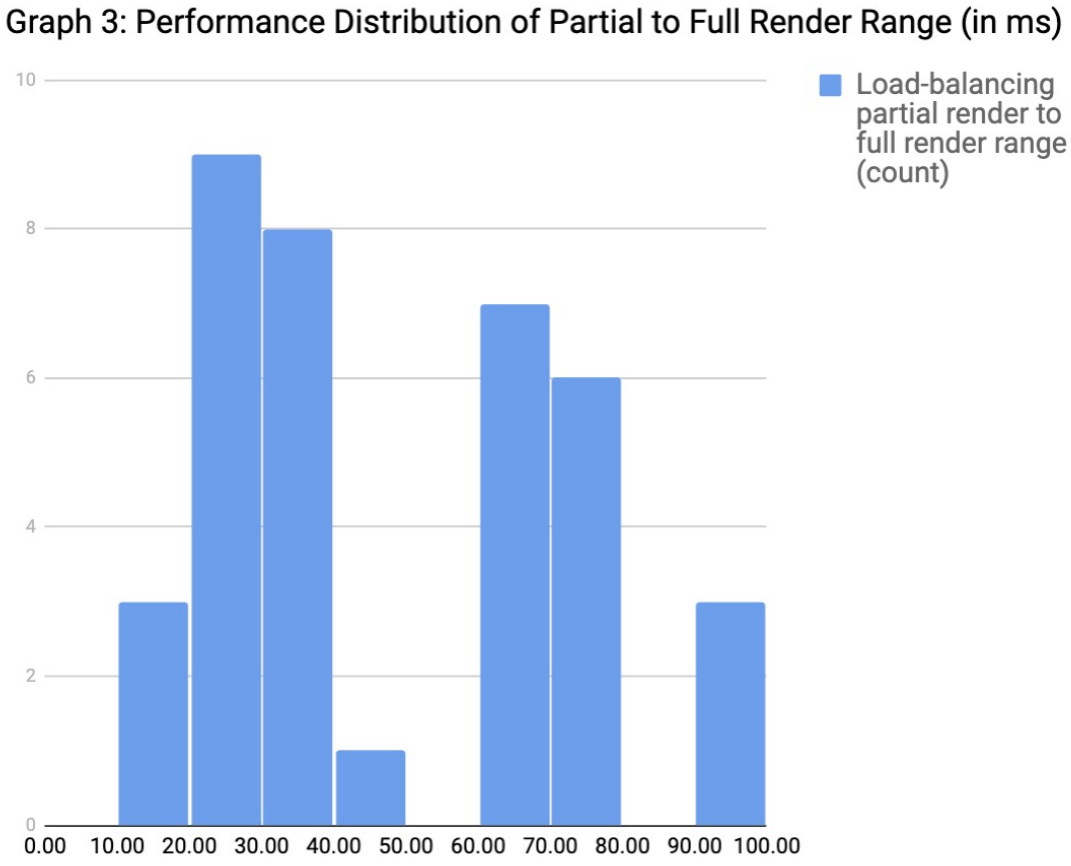
A.2 Quantitative Evaluation: Graph 1



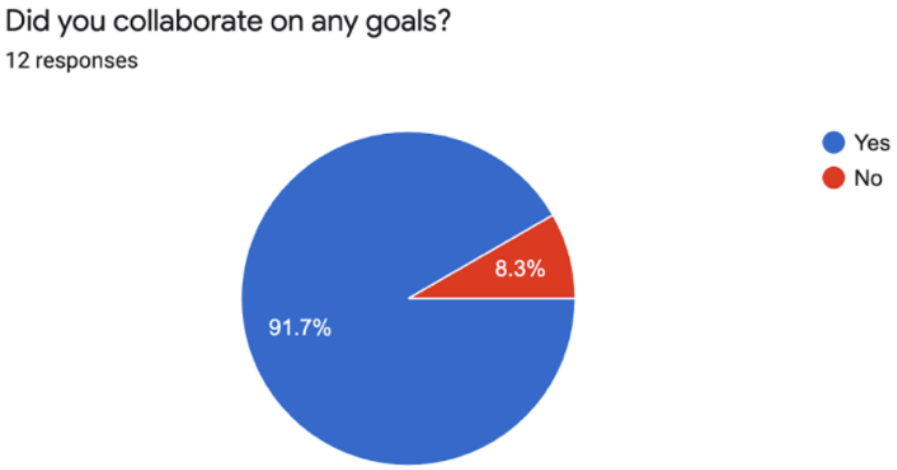
A.3 Quantitative Evaluation: Graph 2



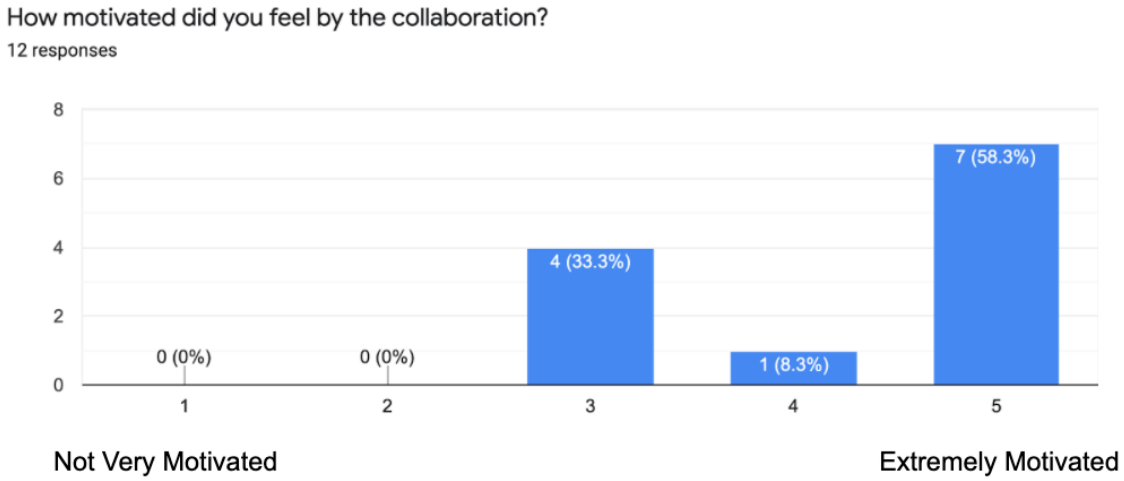
A.4 Quantitative Evaluation: Graph 3



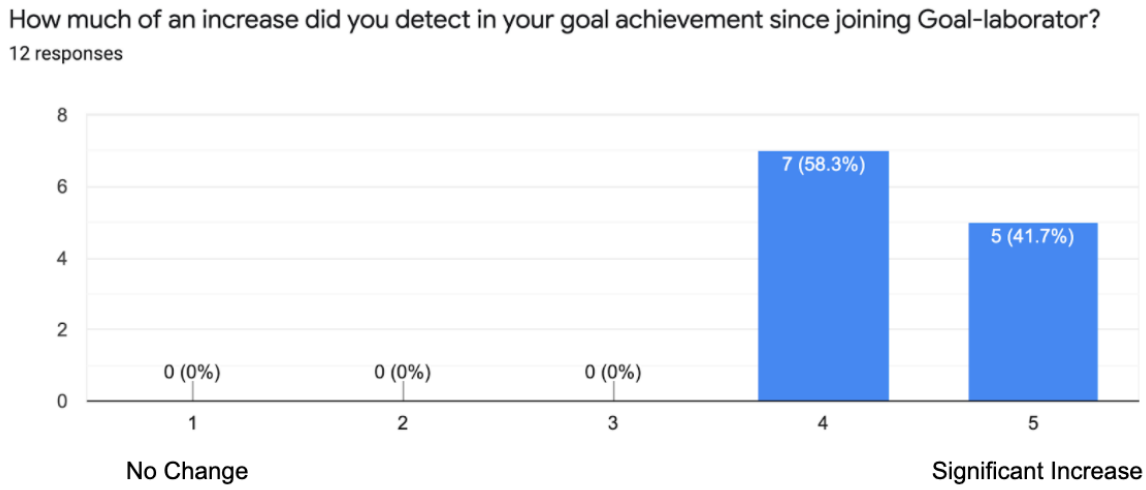
A.5 User Survey Results: Chart 1



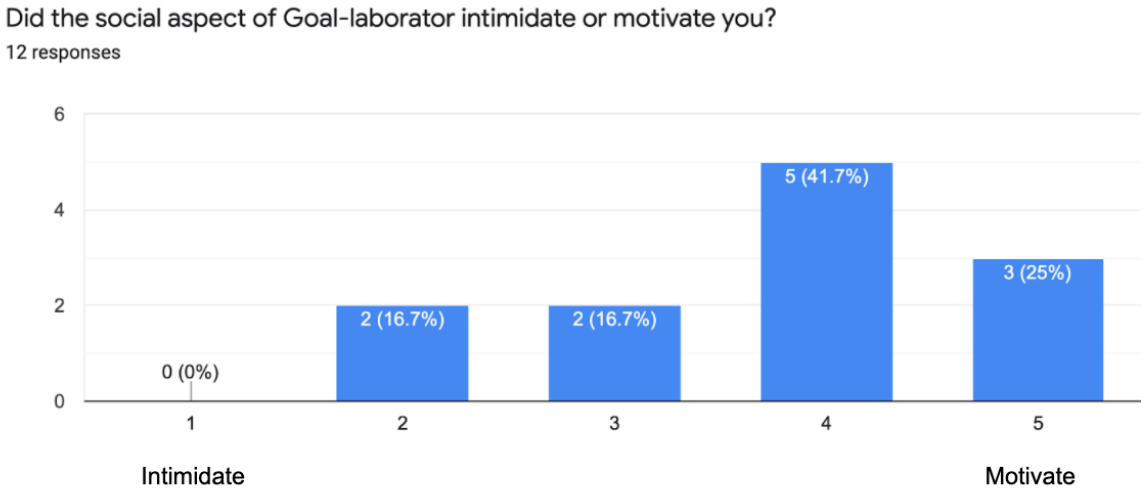
A.6 User Survey Results: Chart 2



A.7 User Survey Results: Chart 3



A.8 User Survey Results: Chart 4



A.9 User Survey Results: Chart 5