Nate Cadicamo, Parker Kasiewicz, Joshua Rizo, Hamed Hekmat

CS 278 Final Report

Spring 2025

# WishingWell: A Little Kindness, Delivered Daily

Frontend repo: https://github.com/njcad/WishingWellApp

Backend repo: https://github.com/njcad/WishingWellBack

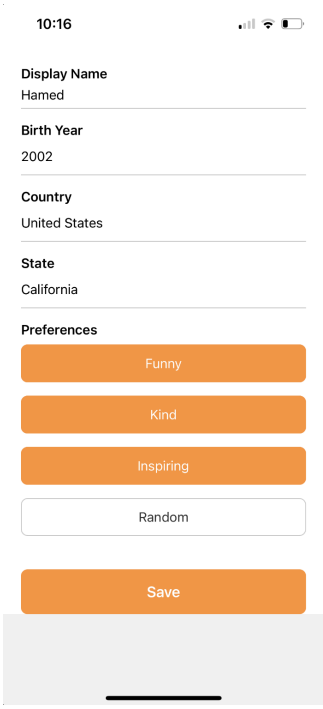
Video demo: wwDemoFinal.mp4

## I. Introduction

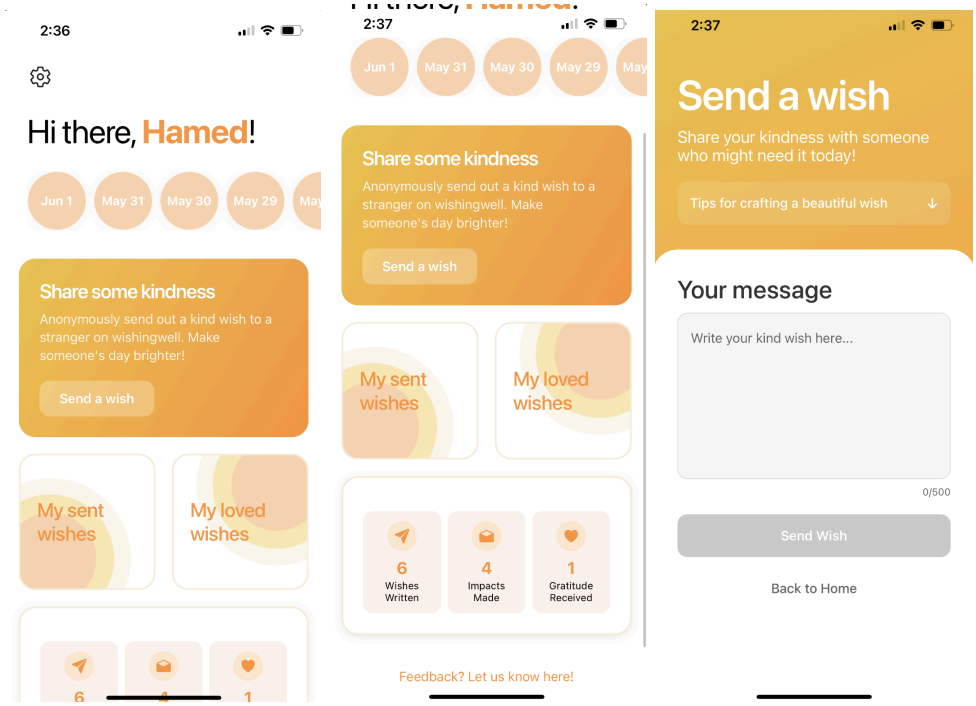
Inspired by universal compassion meditation, we've designed WishingWell, a sociotechnical system built on anonymous microinteractions to spread kindness. We believe everyone benefits from the opportunity to spread and receive positive energy, so we designed a mobile application where users receive a kind wish from a random stranger on a daily basis and are also provided with the opportunity to submit their own kind wish to a stranger. While we hope this app can be used by everyone, it's especially designed for those that desire to spread kindness or are simply hoping for a little boost of positivity. Our pilot revealed that we've successfully cultivated a space where users can spread kindness, and that users can be incredibly creative in the types of well wishes they offer to strangers. At the same time, we've realized our design also has significant room for improvement to maximize user engagement and retention, so we hope to explore some of these ideas and additional features outside of this course.

## II. Design

WishingWell is designed to evoke warmth and simplicity, centered around providing an avenue to extend a daily dose of kindness. On signup, the user is prompted to fill in basic information, including a **Preferences** portion to specify what "flavor" of kindness they're seeking to receive from the app.




On login, following suit with any social interaction, the user is greeted with a simple salutation message, upheld against a fuzzy orange gradient, that signals the warmth of endearing engagement. To focus users on the app's sole purpose of spreading and receiving a kind message, the home page encapsulates the entirety of the app's user flow. At the top of the page, users can access their past received wishes via clicking on individual circles corresponding to each day they received a wish. The intention here is to perpetuate the presence of kindness via these glowing orbs of light and positivity, kindness is meant to stick with the user, just as any thoughtful gesture lingers.




Below that, a simple call to action is made that appeals to an intrinsic motivation to lighten another user's day. With this being the primary feature of the app, we route the user to a separate page upon clicking the *Send a wish* button. In the same vein of simplicity, this page contains a brief prompting message and a text box, an empty canvas where the user gets to decide how they choose to convey their message of kindness for the day. Upon submission, the user's wish is added to a pool of pending wishes that are scheduled to be distributed the following day, completing the action item for the user for that day.

As a means to allow the users to "feel" as though their messages are reaching another individual – rather than going into an undefined mobile app void – we provide a dashboard of metrics, including how many wishes they've sent, received, and how many of their sent messages have received a "like". Feedback is a critical component of any social interaction, so we emphasize in our system the notion of expressing gratitude to messages that moved the receiving user, and we relay that response back to the user.


Lastly, we felt it was important for users to be able to revisit both the wishes they've sent and the ones they've favorited. Whether someone needs inspiration for a new message or simply wants to hold onto a particularly resonant wish, we provide the user with both a "My sent wishes" and "My loved wishes", giving users a dedicated space to reflect on both their own acts of compassion and the message that moved them. Day to day, the app asks for no more than a few minutes of a user's time: they open the app, read their daily assignment, and jot down a brief thought of kindness. By giving a view into the cumulative result of their activity, we hope to provide the user with a sense of lasting impact and connection, via a showcase that everyday gestures of goodwill can together build a beautiful wall of shared humanity.

## III. Implementation

**Zone 2**

WishingWell is built on a React Native front-end and Python/Flask backend, with database and user auth management via Supabase. To host our back-end, we use Render. We also

host a cron scheduler service to handle the daily assignment of pending wishes to users. Our technical implementation follows suit with Zone 1, where the entirety of the platform is built on a codebase created by our team, however, **we note the substantive use of AI platforms to aid in development, which is why we self-categorize as Zone 2, especially since we have enough users to do a substantive analysis of the social interactions 🙂.**

**The front-end**

The front-end of our mobile app is built with React Native. We structured our screens using React Navigation Tab/Stack navigators, defining the basic user flows from Sign Up/Log In to the home page and downstream sub-screens and tabs. Key design elements revolving around the "warm" aesthetic were achieved via Expo modules such as Linear Gradient and basic StyleSheets. We manage user logins and authentication via `supabase.auth`, using `AsyncStorage` to persist sessions. To connect our front-end to our back-end services, we use an Axios instance to hit the relevant back-end routes.

**The back-end**

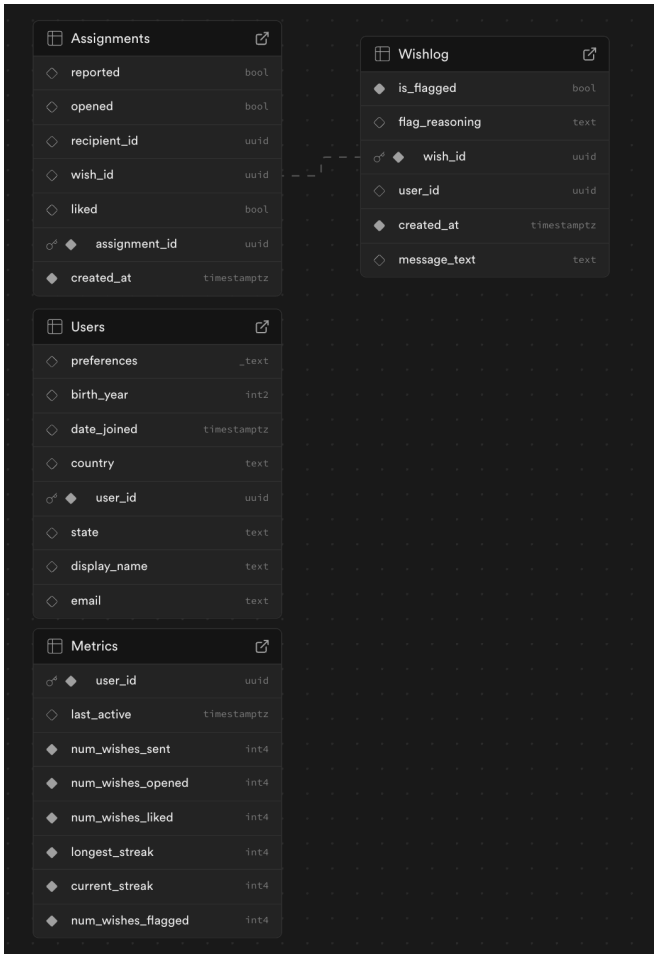
We organize our Python Flask backend into Blueprints under `app/routes/`, each of which delegates to a service layer under `app/services/` for business logic, which in turn calls repository classes under `app/repositories/` to perform the specific `.select()`, `.insert()` and other Supabase SQL calls. Specifically, each of these directories contains decompositioned files for our four main tables of `users, wishes, metrics,` and `assignments`.

To demonstrate the technical design, we'll illustrate the flow for a user creating a wish. From the front end, the Axios instance calls on our `"/create_wish"` route found in our backend `routes/assignment_routes.py`, with the user id and message passed via the

request's header. The route then calls on `create_wish()` in `services/wishes_services` – it's here that we encode any logic, such as looping, though we'll note here that no extra enveloping logic is necessary for this task. This service function then calls `create_wish()` in `routes/wishes_repo` to interface with Supabase, completing the task.

An important functionality of WishingWell is our moderation setup, put in place to ensure that users abide by an injunctive norm of only disseminating positive messages. To achieve this, we pass each submitted wish through a Llama LLM call, with an accompanying prompt instructing the model that it is to flag any messages that aren't explicitly kind and provide clear reasoning back to the user for flagged messages. The output is then processed and used to flag the wish in the database, at which point the wish will be ignored from any future assignments. We also have internal Supabase logic for dealing with users who are repeatedly flagged or have their wishes repeatedly reported.

Below is a visual breakdown of the database setup in Supabase. We contain any user information in our **Users** table. In our **Wishlog** table, all wishes submitted by users are stored, along with auxiliary data such as moderation flagging. In the **Metrics** table, we store up to date metric information for each user, such as daily usage streaks and number of wishes sent. Lastly, our **Assignments** table contains all assigned wishes, mapping wishes in **Wishlog** to the user id of the recipient. We also store wish interaction data here, such as whether a recipient opened or liked the message. We have been iterating with the table structure as we have recently been tinkering with security features and adding new app features outside the scope of this class, so the design is a little in-flux at this point!




## IV. Viral Usage

Once our initial prototype was approved on TestFlight and functioning as intended with the backend up and running on Render, we began to launch to users. We initially reached out to the 20 or so individuals we'd recruited for our earlier testing of our piggyback prototype, and soon after, we expanded to various subcommunities that at least one of us was a part of. For example, we sent the TestFlight invite to our families, friend groups, and other group chats (related to our other courses and extracurriculars) to simply widen our reach as much as possible. These distribution and outreach efforts were generally targeted towards people/groups of people that we believe align directly with our overarching ethos and vision of spreading kindness for the

sake of spreading kindness, essentially functioning as the common denominator for our intended atomic networks. For example, we invited members of Stanford Zen Society, as well as a group chat with one team member's classmates from an Arabic calligraphy course.

We did, however, intentionally avoid posting on Fizz or any similar platform where we'd have less control over who'd be joining WishingWell. At least in our current stage where we're still ironing out some technical wrinkles and testing certain flows and features, we decided it'd be best to have a relative grasp over who we welcome into the virtual space we've created, especially in terms of the intentions of those who join. We anticipated that expanding WishingWell to a large collection of people that we can't vet/vouch for would inevitably lead to an increased likelihood of rare events and antisocial behaviors such as trolling, which would be especially problematic as we still work on the specifics of our security features (e.g. row level security) and content moderation pipeline (e.g. prompt engineering and refinement).

Over the course of our 2-week pilot, we had 40 users sign up thus far. The vast majority of these users sent fewer than 5 wishes with 4 users sending at least 5 wishes. About ½ of users sent a total of one wish, meaning that they downloaded the app and likely never logged in again. About ¼ of users never sent a wish at all (could be explained by a variety of factors including sign-in issues, lack of interest, lurking). The remaining ¼ of users had streaks of at least a few days, which is certainly the behavior we're looking for. Because our app contains just micro-interactions, we imagine we would greatly benefit from post-notifications, for example, so we optimistically think that our churn might be lower in an environment with a fully-published app.

As far as user behavior, we for the most part simply witnessed a wonderful display of creativity when it came to people submitting their kind wishes. This didn't really come as a

surprise considering we'd already witnessed people's ability to think outside the box when we experimented with our piggyback prototype, but we still found it refreshing to see everything ranging from excerpts of poems to anecdotes involving kindness to prompts inspiring users to complete quests of kindness to many other kinds of messages. Thankfully, we didn't witness a lot of users simply mailing in their wishes in the form of short responses that clearly lacked thought and effort, a behavior we were initially concerned about. However, we did see instances where people opted to send silly or obscure messages (e.g. quoting irrelevant rap lyrics) instead of kind ones, oftentimes getting flagged by our content moderation LLM. We also noticed some cases in our database where individuals seemed to be really trying to push the limits of our content moderation strategy and see what could pass through/trick the system, which we found to actually be quite useful as it revealed minor issues with our prompt engineering. Despite all of these observations regarding user activity on WishingWell, perhaps the most surprising result for us was the fact that many of our users never sent a single wish. These users went through the process of creating and authenticating their accounts yet never truly used the app, prompting valuable reflections on our end regarding how our design choices influence user engagement.

## V. Design Reflections

Before further exploring why we didn't necessarily receive the maximal user engagement we desired, it's worth acknowledging the aspects of our design that worked well based on our users' responses through informal interviews and an embedded feedback form. We received quite a lot of positive feedback regarding the simplicity and elegance of our overall aesthetic as well as how intuitive our main flows were. In a similar vein, people also noted that our orange gradient color scheme certainly resonated with the overall theme of spreading kindness and positive

energy due to the warm atmosphere it creates. Some also expressed appreciation for the orb imagery (visualizing kind wishes as glowing orbs), as well as more subtle details such the dropdown suggestions for crafting a kind wish or the animated loading screen that reminds the user to, "take a few deep breaths."

From a technical design standpoint, we were exceptionally intentional and disciplined about how we structured and iterated upon our code, making it much easier to develop and test new features. Our repositories were extremely well organized, and we practiced very effective decomposition. Not only did this make our lives easier during the course of this project, but it reminded us just how valuable that extra initial effort can be in the long run. We also had several moments that helped us realize that essentially all logic should be handled in the backend to optimize frontend performance. Similarly, while we eventually realized that the runtime complexity of our wish assignment algorithm wouldn't matter since it's only run in the background once a day and late at night, we still collaboratively modified our initial design to make it significantly more efficient while cleverly meeting our needs of ensuring everyone's submitted wish is randomly distributed to at least one other user. We're quite proud of WishingWell as a piece of software, as we feel like we learned a lot about best practices and reinforced good collaborative software development habits in the process.

Revisiting the dilemma of user engagement and retention, we anticipated this being a potential challenge from the outset, especially considering the novelty may wear off and boredom may arise from the simplicity of the microinteractions our users would be engaging in. This inspired the addition of some of our current features, such as the inclusion of a panel that displays the user's metrics in terms of streak length, people impacted (how many messages sent from me were opened), and gratitude received (how many likes I've received on messages that

I'd sent) to try to keep users more invested in contributing to the WishingWell community. On another note, despite simplicity being a strength, we'd arguably made things too minimalist in some aspects, with one user pointing out that the button for opening a new wish should immediately be extremely prominent upon entering the app rather than simply being a small circle labeled, "New!"

Our primary response when reflecting on the overall effectiveness of our design was to brainstorm how future features could encourage users to stay engaged with WishingWell. For example, a simple addition would be push notifications, perhaps one in the morning reminding the user that they have received a new wish, and one in the evening as a final reminder if a user hasn't sent their kind message of the day quite yet. We didn't have a chance to test this hypothesis over the course of this quarter, but we do believe that having to open the application and develop a new habit of checking it daily may have added too much friction for some users, which could hopefully be remedied via gentle reminders in the form of push notifications. When considering other potential obstacles to increased user engagement, we also discussed the fact that providing users with such an open ended prompt/task ("Share some kindness - Anonymously send out a kind wish to a stranger on wishingwell.") could be daunting for some, and even for those that don't struggle to be creative, we imagined it could quickly become redundant (or even tedious, with one user suggesting we allow users to write a batch of wishes at once for convenience). We've posited that one solution for this could be to have a set of more specific prompts/themes for wishes and cycle through them, adding an element of intrigue and suspense for users and increasing the likelihood that one might open the app out of curiosity. For example, some days could be sharing a favorite uplifting quote, another could be providing a

positive mantra, another could be describing a self-care/meditation exercise for the recipient to try, and so on.

Another avenue for growth would be greater customization and personalization to improve a user's experience in a more targeted and directed manner. Early on, we floated the idea of designing a more tailored wish assignment algorithm that factors in each user's preferences, which is why we included preferences as fields in our user profiles. This next iteration of our algorithm would still randomly sample from the previous day's pool of LLM-approved wishes, but this sampling process would be weighted by the wish's compatibility with a given user, perhaps obtained by scoring each wish on certain dimensions users might care about.

On a similar note, many users expressed interest in having an ability to further connect with those from whom they're receiving wishes and/or to whom they're sending wishes. We considered displaying location and age-based information about the senders and recipients a given user interacts with to make their interactions feel more concrete and grounded, which is why we included these fields in our user profiles as well. For example, alongside the simple numerical summary of how many people have received, opened, and liked a wish from a given user, we envision that the user could have access to a map that includes pins in the general geographic locations of those with whom they've interacted to help them visualize the scope of their impact. Another user suggested we include some kind of discussion board or threaded messaging system as an additional offering. Overall, such considerations and concerns voiced by our users validated our initial instinct that while our app's simplicity may be a strength in some ways, it also poses challenges regarding user engagement and retention as motivated users may feel restricted and/or dissatisfied by their inability to connect more deeply with other users.

Lastly, we believe we designed well to mitigate anti-social behaviors, but the specific niche and intended target audience of WishingWell makes it especially important for us to ensure that we're cultivating and maintaining a safe and positive space for users. Our Terms of Service clarify behaviors that we don't tolerate, and the general atmosphere of the app (along with subtle details such as the dropdown with tips for crafting a kind wish) make it a conducive environment to stick to our desired behavior of spreading kindness. Additionally, we further protected ourselves by rolling out to groups that we believed would respect the space and use it as intended. This was especially crucial as even a single negative experience (e.g. an inappropriate or unkind wish slips through our content moderation system) can dissuade a user from returning to the app, so we still found ourselves proactively monitoring Supabase to keep track of patterns regarding what kinds of wishes were flagged and why, helping us refine our content moderation strategy.

## VI. Theory

Major existing social media platforms primarily incentivize quantity over quality of connections and interactions, oftentimes leading to disingenuous and dissatisfying user experiences. In response, we sought to design a sociotechnical system that stripped back all the added fluff to emphasize its core purpose and features. Human beings have a tendency to find fulfillment in doing kind deeds, so we hoped to lean into this quality and the innate contagiousness of positivity by providing users with the opportunity for an uplifting microinteraction on a daily basis. By focusing on mirroring the positive energy associated with universal compassion meditation, we strive to go "beyond being there". We never attempt to recreate the feeling of an in-person kind interaction, instead leaning on the unique advantages of

an anonymous platform that allows you to share and receive kindness from a complete stranger from literally anywhere.

Once again revisiting the dilemma of user engagement and retention, many of our observations regarding user behaviors can be explained by applying course concepts. We attempted to target atomic networks that would align with the ethos and values of WishingWell in hopes of avoiding becoming a ghost town, and we weren't concerned about the cold start problem because we only needed two wishes per day to be able to successfully assign a new wish to every user. With this initial wave of users, we believed they'd be predominantly driven by intrinsic motivators, a trait which we feel would be reflective of an archetypal user down the line. We had many discussions all throughout our design process pertaining to the question of intrinsic versus extrinsic motivators, especially determining the extent to which we could introduce extrinsic motivators or gamification without contradicting our mission or risking a crowding out effect. For example, we briefly debated creating a publicly visible leaderboard for most liked wishes, but we ultimately decided this would conflict with our primary purpose.

We also pondered how to effectively leverage reciprocity early on, initially wondering whether a user should have to submit a wish in order to receive their assigned wish for that day. We anticipated that requiring a submission might provide too big of a barrier and get in the way of regular use, as well as potentially conflicting with the main ethos of spreading kindness with no strings attached. However, we do believe that reciprocity has still played a role in encouraging users to participate and contribute wishes, simply because intrinsically motivated users won't want to only consume the content of others, hoping to return the favor and be active contributors to the community. We also feel as though our platform is not susceptible to Grudin's paradox for

similar reasons, as the people that most consistently contribute wishes all still benefit (as everyone is guaranteed to receive a wish) and are also likely the most intrinsically motivated.

As for explaining why user engagement seemed to dwindle over time and some users never sent wishes at all, the fact that some users did express a desire to be able to interact with those who send/receive their wishes does highlight a potential limitation of our app. Having users engage anonymously through microinteractions does inherently limit such connections to being strictly weak ties and restricts the ability to develop greater affinity and work toward developing strong ties with other users. Along with our aforementioned design reflections regarding how we could make WishingWell more engaging moving forward, this consideration may explain how the observed behaviors of our users certainly create a contribution pyramid, and our decision to track user metrics allows us to see this concretely. At the top of the pyramid (not including ourselves as moderators) are those users that maintained extended streaks and submitted wishes for the majority of the days, followed by users that sent the occasional wish but actively sent gratitude to many of the wishes they received. Beneath that, we have users that opened some of their wishes (essentially the lurkers), followed by users that created accounts but never contributed. We didn't have enough testers to see a significant narrowing down ascending the pyramid, but we certainly observed a clear tapering.

Lastly, course concepts played a prominent role in how we encourage desired behaviors on our platforms and strive to maintain an atmosphere that cultivates kindness. We leveraged injunctive norms by explicitly clarifying what content we consider to be inappropriate in our Terms of Use as well as providing guidance through our dropdown tips for crafting a kind message, and we simultaneously believe that descriptive norms developed over time as users got exposed to more wishes from others that could act as inspiration for future wishes. Additionally,

we tried to protect against people that would promote antisocial behaviors such as trolls by being selective about which communities/people we invited onto the app. We also consistently iterated on our content moderation, specifically by improving our prompt engineering for the LLM to only accept explicitly kind messages, creating a buffer where even messages that may slip past the LLM and avoid being flagged would be neutral/confusing at worst, significantly reducing the likelihood of any truly harmful wishes being distributed. Overall, we established a governance structure right off the bat where we'd alert users when their wish got flagged and why while also acting as BDFLs (benevolent dictators for life) that can terminate the accounts of repeat offenders, avoiding the problem discussed in lecture of no governance being bad governance.

## VII. Tarot Analysis

Thinking back on the Tarot card activity, we think the same cards still apply but in different and nuanced ways. For the "catalyst" card, we would like to think our app could have this sort of effect, and based on user responses and feedback we hope it has some, but obviously our app is still at a tiny scale. We think the "big bad wolf" is also still quite applicable, and while we had no obvious negative occurrences, we did have a few wishes that were off-topic and/or confusing slip through our moderation thanks to the "kind" redteaming of our friends. Interestingly, we experienced to some extent the "backstabber", where although we certainly didn't lose trust for ethical reasons or failures, and users were certainly primed and sympathetic because it was just an initial release, we did have intermittent periods of downtime where the app wasn't functional and just general UI issues, which we think contributed greatly to user disengagement. Throughout the project, we've had to think a lot about user trust both from an ethical standpoint and also a metrics/churn standpoint, which we anticipated being important, of

course, but didn't predict how it would manifest. And we had to adapt (improving the depth of the moderation pipeline as well as developing better methods for dropping stable releases and seeking feedback on UI issues) to accommodate these facets of our platform.

## VIII. Conclusion

Our journey in developing, iterating, and deploying WishingWell taught us an important lesson: embracing a simple app design centered around the quintessential human desire for sharing compassion *works*. At least, somewhat! We have a few things we are itching to do before we really go live: build post notifications, improve customization and add more outlets for creative expression, and improve our security and moderation systems. But, we hope that you see us on the app store someday soon, and thank you for your support and time with our project! 🙂