Hyper-Local Social Media: Herd-It

Nicholas Sulzbacher
Gabe Ewsuk

Clemente Izurieta
CSCI 483R
Montana State University

Abstract

The proposed application, Herd-It, outlines the development of a social media platform with a unique approach to content organization. This platform will feature dedicated hubs for various topics, fostering community engagement and discussions. A key feature is the platform's emphasis on location-driven content, allowing users to tailor their experiences by expanding or narrowing their searches based on geographical preferences. Additionally, users can access content in their area before creating an account promoting out-of-the-box usability without a cumbersome signup process. While the project acknowledges the importance of user interface design, it aims to maintain simplicity in its initial stages.

Introduction to Herd-It

Herd-It is a new social platform that uses geo-location to drive users to discover relevant topics and ideas. Large social media platforms, such as Instagram and X, lack methods to discover local activities and topics based on the user's location. Herd-It aims to solve this issue by allowing users to view and interact with posts that are within a certain range of them. Whether it's a short weekend trip or a full relocation due to a job, users are constantly moving and need a platform that moves with them. While some existing applications offer similar functions, they confine the user to one stagnant location and don't allow for a change of location. Herd-It will constantly adapt to the user's location, allowing discoverability in any area. This enables users not only to discover and meet new people who are interested in similar topics but also to interact with new people wherever they go.

The primary functionality of this application relies on geo-location. When users create a post, they'll have the option to set the post's radius, determining its visibility. The post will only be visible to other users within that specified radius on their main feed. This innovative feature sets our application apart from other mainstream social media platforms. We aim for users of Herd-It to effortlessly discover new people, topics, hobbies, and events. Setting post radius isn't the sole technological advancement; users will also have a dynamic radius that adjusts as they move. This dynamic radius powers our 'discovery' board. Unlike the main board, which reflects the user's immediate vicinity, the discovery board allows users to explore posts or events beyond their direct connections. This capability enables users to interact with content not limited to their immediate location.

Herd-It is an innovative social media platform that uses geo-location technology to foster more authentic social experiences among its users. Large social media platforms often facilitate interactions that significantly differ from in-person conversations. This distinction is evident in comment sections accompanying posts and through numerous direct messages exchanged between users. Our upcoming application, Herd-It, aims to address this challenge directly. By organizing user interactions within a defined radius set by the user, our platform enables individuals to explore posts and engage in conversations that significantly impact their social experience.

There are two primary benefits to this technology. The main boards of mainstream social media applications are primarily driven by the vast network that the user has built and don't allow for easy localized discoverability. Although other social media sites have 'discovery' feeds, they fall short in enabling users to explore effectively. The 'discovery' algorithms on these sites tend to trap users in a loop by providing more of the same topics that users are already viewing or engaging with regularly. Herd-It aims to build a network of users primarily driven by their proximity to each

other and their posts. Instead of simply regurgitating topics and posts users have already encountered, our algorithms will form the foundation for users to genuinely discover new content around them. This directly benefits users by enhancing their discovery of new posts, ideas, topics, hobbies, and events.

We have two primary goals we aim to achieve with our technology. The first goal is to encourage novel interactions among users. Herd-It serves as the bridge to close the gap, enabling the public to access what they desire, precisely when they desire it. By showcasing ongoing events in their vicinity, users can easily discover matters of interest around them, be it concerts, sporting events, educational topics, or casual conversations. Those utilizing Herd-It will find it easier than ever to discover these opportunities.

Team Qualifications

Team Members:

- Nicholas Sulzbacher: Bachelors in Interdisciplinary Computer Science
    - Minor: Business Administration
    - Resume: Appendix entry A.1

- Gabriel Ewsuk: Bachelors in Interdisciplinary Computer Science
    - Minor: Finance
    - Resume: Appendix entry A.2

Background Research

The background research for Herd-It included three different areas. The first area was research into other social media applications and their similarities. The second area was UI and UX design. The last area was diving into other social media applications' algorithms and the data that is tracked within them.

Similar Applications Research

Developing an app involves thorough research, and one crucial aspect is studying existing technology addressing similar issues. While many major social media apps incorporate location-based features, only a few make it a central part of their service. NextDoor is a prime example being described as, "a hyperlocal social networking service for neighborhoods." With over 1.5 million reviews and a 4.7-star rating on the app store, it has proven successful in the hyperlocal social tech space. It can be inferred that people are interested in interacting with locals using social media. However, NextDoor is targeting a very specific niche, homeowners. The goal of our app is to target younger people, more specifically college students.

But aren't there other social media apps that already target college students? YikYak, for example, stands out for its approach to targeting this demographic. The app is designed to foster local connections among users attending specific universities and offers an anonymous feature for sharing thoughts without judgment. Initially, it was widely acclaimed, often described as a "digital bulletin board." YikYak gained significant traction across U.S. campuses. However, its success was short-lived due to the toxic, inappropriate, and dangerous anonymous user activity, which led to threats, bullying, and hate speech, ultimately resulting in campus bans. Despite the issues related to anonymity, YikYak demonstrated that hyperlocal engagement and anonymous interaction are powerful features that set it apart from other players in the space. Had it not been for unchecked

toxicity, the app might have continued to thrive. These events suggest a demand for hyperlocal engagement on college campuses.

While these apps are popular in the local context, it's important to consider an app closer to the concept we're trying to create, such as Meetup.com. Meetup is a social media platform that focuses on organizing in-person activities, gatherings, and events for people with shared interests, hobbies, and professions. With 225k ratings and a 4.7-star rating on the app store, it's evident that Meetup is achieving success in the space. However, it primarily focuses on real events and lacks features for smaller, quick gatherings suitable for college students looking to connect. The app is somewhat cumbersome and has a high learning curve for new users.

Examining the success of these three applications reveals a discernible demand for connecting with individuals nearby. However, significant pitfalls, notably stemming from issues related to anonymity and navigating a convoluted user interface, have been identified. To address these challenges, our application should prioritize an unparalleled level of user-friendliness, emulating the instantaneous accessibility characteristic of Instagram.
Upon logging in, users should be met with a dynamic feed presenting local posts, eliminating the necessity for users to actively follow or specify interests, thereby streamlining their overall experience. By embracing a format reminiscent of Instagram, the primary objective is to reduce friction for users, ensuring a seamless transition into local content upon the app's installation. The focus should be on mitigating anonymity concerns and accentuating ease of use, enabling users to promptly access and engage with local events from the very onset of their participation on the platform.

UML/UX Research

Social media applications have become more prevalent in recent decades than anyone would have guessed. These applications are primarily used for social networking. When MySpace sparked the social networking concept, the goal was to help users connect with friends and family and meet new users with similar interests. This tech was revolutionary as users got content in real time. However, in this age, social apps are heavily used for advertising and branding along with the social connections that are created. These companies aim to keep users on the app for as long as possible. More interactions and purchases mean more money for the company owners and content creators. Users interact with these companies and influencers, which makes for a more personal experience. To accomplish this, developers create an easy-to-look-at application that has multiple different components to make sure users do not lose interest.

Component management and capabilities across different social media sites share the same outline. As stated above, the goal is to keep users on the app for as long as possible. Since MySpace, hundreds and thousands of social media applications have been developed. Some catch on and have a significant impact for multiple decades (Instagram and Facebook), while others have a less pleasant impact (Slidebean). With the amount of users on large social media applications, companies run the risk of having unpleasant users on their software. As stated earlier, YikYak's downfall was mainly due to a component that the developers made to help the app stand out. Users could post anonymously and communicate with other users in the same vicinity. However, the app's lack of moderation and accountability created a more “toxic” environment. This deters users from the app as their experience tends to be more negative.

No matter the moderation level of the apps, the main drawing point for users is the ease of using the application. Over time, social media platforms have learned ways to keep users on the app and create an optimal environment for the users. These apps allow users to follow and interact with

other people. These followers might be close friends or strangers, depending on the platform. Users interact with posts and read comments from other viewers. Most platforms also let users customize their profiles. Apps are now becoming more customizable to increase user interaction. This includes filtering what type of information they are seeing. For example, location-based applications let users view content within a certain region depending on what they are looking to find. As social media platforms continue thriving, developers will continue finding new ways to better the user experience.

Algorithms Research

Each social media application has a wide variety of information tracking, but most track the same information. Four categories of information—format, user, activity, and interaction history—are tracked on posts. The format includes information on the post type, whether in video, photo, or text. The user information tracks the relationship between users. This includes whether both users follow each other and if not, who follows who. Activity is tracked through different means. Included in it are the types of posts a user typically engages with and whether or not videos are commonly engaged with. Interaction history mainly tracks whether or not you like, comment, share, or repost another user's content.

Engagement algorithms in various other social media applications share a common objective: to retain users on the site for as long and as frequently as possible. These algorithms primarily concentrate on displaying posts that aim to extend users' time spent on the app and encourage them to return more frequently. For example, Instagram is broken up into multiple pages, main feed, stories, explore, and reels. The feed is a user's primary page where the content is driven by the user's followed accounts and recommended posts. Stories are short-term posts that can be viewed for twenty-four hours and then are removed from the headboard. Stories appear at the top of the application and are driven by who the user follows and engages with the most. The explore page is where the user can go to find new content that isn't entirely similar to their main feed. Though the explore page would imply the ability to explore new topics and posts, the algorithm for the page often keeps the user still looking at what they want to see. The last and most recently added page to Instagram is the reels page. This page is where most of the short-form content for a user is stored. This page is not solely driven by the user's following, although it can be influenced by it. Other mainstream social media sites have similar pages to these, though may have some formatting differences.

Engagement can be examined further. The application can send a user's post through a series of checks and flags. These checks are to see if the post might be engaged with more than another. The checks include the likelihood that other users like, comment, or share the post, whether a user may save the post and come back to it later, and whether the post includes a hyperlink or reference to one.

Twitter, now publicly named X, has made a few of its priority algorithms public and viewable on GitHub. After examining a few for a bit, you'll see that X includes the content of the posts in their rankings of posts. A post's ranking determines its overall reach and priority in user viewability. A post with a lower ranking will not be displayed as much as one with a high ranking. What kind of content is ranked differently? X's algorithms analyze the content of a post and have four major flags for a post (Singh). Relevance is the most important to the algorithm. If a post's content is not flagged as relevant to a user, it has a much lower chance of being seen by that user. Recency is the next flag. Though not as crucial as relevancy, recency still has a large impact on posts. Current events, trending topics, and popular ideas all factor into this (Singh). Diversity is important to X's algorithms as well, being the third flag. X tries to show posts that aren't just within the user's

thought bubble. This allows users to interact with others who may not share similar opinions or ideas (Singh). Media files are the last flag. This includes the actual form of the content. Visual media files, such as photos and videos, are flagged as of higher importance than just text posts because users are more inclined to watch instead of read.

Uber Offers an open-source library for handling tiling for geospatial data called H3. H3 divides the geospatial area into hexagons. This technology will be useful to drastically reduce server-side requests. The idea is to associate each post with a certain hexagon id ie. a specific tile on the map. When the user requests the backend they will request all posts within tiles that their radius comes in contact with (at most this will be 3 due to the nature of hexagon tiles). This means they will receive more posts than are in their immediate radius, however, this means filtering based on location and radius can be handled on the front end. This means requests will only be made to the backend when new tiles are entered rather than every time a user's location or radius changes. This unique implementation is key to a timely and cost-effective user experience.

Project Work Schedule

This project is broken into two separate work divisions with the team member that is completing the respective division outlined below. The work completed by each group member is listed in appendix entry A.3.

- Nicholas Sulzbacher: Development Environment Initialization
- Gabe Ewsuk: Full Stack Software Development & Design

Proposal Statement

1. Application Overview:

Our app allows users to connect with people who have similar interests outside their immediate network. This is accomplished through social media posts tied to a geolocation. Posts can only be viewed by people within a specific distance set by the poster promoting local meetups and easy access to local information. Like other popular and successful social media, the plan is to start on campus. This means our immediate potential market will be students. Students are generally interested in participating in group activities and campus is a common location with lots of people. This means even with a small number of users we can provide value while the app is in its infancy. The campus has the added benefit that word spreads fast. When users find value in Herd-It they will talk with their friends leading more users to get involved.

2. Mobile Platform Technology Stack:

Herd-It will initially be designed for mobile use. React-Native will be used for the application’s production for both Android and iOS devices. Below is an outline of the technology stack that will be used for initial development. The technologies used are subject to change throughout the development cycle.

- **Front-end (UI/UX): React-Native + Expo**
  Javascript will be the primary programming language for designing and implementing the user interface. Along with Javascript, React-Native will be the front-end framework used for simultaneous development for the Android and iOS platforms. The

advantages of these technologies are the reusability of code and team familiarity with the programming languages used. React-Native is also used to create a smoother and more responsive user interface.

- **Back-end Algorithms: Python**

  Python will be used as the primary back-end algorithms programming language. The development team is not only very familiar with this programming language, but Herd-It will require heavy data handling. Python is a very strong language for this purpose. Python also includes various MySQL libraries that will allow for easy and responsive communication between the algorithms and the database.

- **Back-End Framework: Django**

  In conjunction with Python, Django will be a back-end framework used in development. Django is open source and will allow the team to execute rapid development of the algorithms. There are no costs associated with Django keeping the project expense to a minimum.

- **Database: SQLITE3**

  SQLITE3 has been selected as the application's database. Team members are all familiar with SQL queries and the communications necessary. SQLITE3 is open source and is a very popular database management system for MVP products. The application will transition to Postgres as it grows to account for rapid scalability if the application needs to be scaled up.

- **Geo-Location: Google Maps**

  With the heavy location and distance-directed algorithms used in Herd-It, Google Maps and its API will be used to derive user location. The Javascript API will allow the javascript on the front end to retrieve the user's locations in the form of longitude and latitude. The Google Maps API allows for easy and fast front-end data retrieval.

- **Hosting: AWS**

  In terms of hosting, the project will leverage the capabilities of Amazon Web Services (AWS). Exploring potential free credits within AWS offers a cost-effective approach to hosting. The plan is to use EC2 (Elastic Compute Cloud) to host our backend server on the cloud which can be exposed to the public internet. We will use a docker container to ensure services can be run without dependency issues within EC2. As the app scales up developers will need to implement a load balancer and auto-scaling EC2 groups to distribute and handle the server-side load. The team will be using AWS S3 to store media files to avoid bloating the database. These can be uploaded with signed signatures provided from the backend to avoid exposing access keys. All media files will be publicly fetchable and accessible via a URI which will be used as a reference to render an image in the UI. AWS IAM roles will be used to manage and restrict access to different cloud resources ensuring the security of the application.

3. User Interface + User Experience (UI/UX) Design.

The initial minimum viable product for Herd-It will be more focused on the application algorithms and their functionality. With that noted, the UI for Herd-It will be kept relatively simple. The goal of Herd-It's UI and UX design is to enable the users and testers to showcase a fast and

simple UI. There will be three primary interfaces for the user.

The first interface will be the main feed. This will be where the majority of the user experience will be. It will house the posts associated with the primary post-viewing algorithms. The second interface will be the side panel (see Section 5: Navigation) that houses the location services of Herd-It. This panel will include the options for users to change and view their location and associated radius. The third interface will be the user's profile. This page will house the user's posts and the primary user profile functions. These functions will include the application settings and user preferences. There is one final page that will be developed if the development time permits. This page will be a secondary feed that will be an alternative discovery page.

4. User Authentication:

User authentication is an important aspect of application and user security. Django includes user authentication with JWTs. This framework will allow the team to create a user login page with verification associated with the user's email or various authentication applications. These permissions will also allow users to reset passwords if necessary.

5. Navigation:

Similar to other social media applications, upon the opening of the application, users will be met with their primary feed. This will be a feed of posts that can be scrolled through by swiping upward on the screen and moving downward on the feed. The amount of posts that are viewed within the screen will initially be set to one, but this will be adjusted if deemed necessary.

The app is designed to be used straight out of the box. This means when the user opens the app they will immediately be greeted with posts in their area regardless of if they are logged in. This was intentionally designed so that users can immediately see what is going on without going through the signup process. When the user enters the app they will see a tab navigator with 4 separate options: Feed, Map, Post Creation, and Account. The feed will render posts in a feed similar to Instagram or Twitter. Each post will include a list of comments and likes. The Map will show users what their radius is on a Google map and include clickable posts within their radius. Post Creation is where users can upload photos and post them to the feed. The account will just contain basic information about a user's account. It is important to note that a user will need to create an account if they wish to post, comment, or like other users' content.

6. Features and Functionality:

The features and functionality of Herd-It will be broken up into the following points.

- **User Profile and Profile Viewing:**

  Users will create a custom profile where they can include a profile picture and a bio. Users will sign up or log in through custom accounts tied to an email. Posts created by the user will also be visible here. Users will be able to visit other users' profiles for viewing, following, and navigating to that user's posts.

- **Geolocation-Based Posting and Viewing:**

  Herd-It users will create posts that have a location as well as a radius tied to them. The location will be tracked using latitude and longitude coordinates. These longitude and latitude values will be sent to an algorithm that uses the Haversine Formula to derive a distance value. This distance value will then be used in a distance relation algorithm as a driver for the user's feeds. Refer to the appendix entry A.7 for a flow chart diagram of these functions. The radius will be tracked as a static and unchangeable distance. Depending on the distance, users will be able to view certain posts that they are within range of. If a user is

inside of a post's radius, that specific post will be fed to the user's main page. If the post is within the user's radius, the post is sent to the discovery page of the user. Posts can include photos, text elements, and hyperlinks. When viewing a post, a user can interact with it. These interactions include liking and commenting. The original creator of the post will have the additional interaction of deleting a post.

- **Setting Changes:**
  The users of Herd-It will be able to change their radius at any time via the location services tab. This will primarily be linked to the discovery page's feed. In the user profile settings, a user will be able to update account information, personal information, and profile pictures. The application setting will be kept to a minimum for this project. They will include basic functions like a logout option and a light/dark mode.

- **User Notifications:**
  In the future, notifications will be kept to a minimum for this project. Using React-Native, notifications linked to likes, comments, and shares will be sent to the user's device.

7. Security:

Security is paramount in our project, and we have implemented a comprehensive measures to safeguard sensitive information and ensure the integrity of user data. One key aspect of our security strategy involves the use of GitHub Secrets and EC2 EC2 Secrets. These secrets play a crucial role in concealing API keys utilized for mapping components within our system, adding a layer of protection against unauthorized access.

Furthermore, all user data is stored in an encrypted SQLITE3 Database. This encryption ensures that even if unauthorized access occurs, the data remains secure and inaccessible. Additionally, we prioritize the security of user passwords by employing a one-way cryptographic hash function for storage, enhancing overall data protection.

In terms of database security, we implement stringent measures such as input validation and sanitization. This proactive approach helps prevent potential security vulnerabilities that could arise from malicious inputs. Moreover, we utilize parameterized queries, a best practice in database security, to mitigate the risk of SQL injection attacks. These measures collectively contribute to a robust security framework, ensuring the confidentiality and integrity of user data throughout the application.

Cloud services are protected by roles and IAM management by AWS. This prevents unauthorized users from gaining access to our cloud services. Additionally, rate limits are set on the backend, and authorization is required to make post requests to the database. This measure was added as a safeguard against data attacks and unauthorized use of the app.

8. App Permissions:

Ensuring a seamless and feature-rich user experience, our application incorporates specific permissions that enhance functionality while respecting user privacy.

Required Location Permissions:

To enable geolocation-based features, the app requests access to the device's location. This permission is integral to the core functionality, allowing users to create and view posts within specific distances. By leveraging location data, the application facilitates connections based on proximity, encouraging users to discover relevant content and engage with others in their local vicinity. This emphasis on location permissions aligns with our commitment to providing users with a personalized and context-aware experience.

Required Access to Camera or Photos:

The app also seeks permission to access the device's camera or photo gallery. This access is essential for users to seamlessly upload and share images related to their activities. By incorporating this permission, the application empowers users to visually express their interests and experiences, contributing to a vibrant and interactive community. The ability to share multimedia content not only enhances user engagement but also adds a dynamic dimension to the platform, fostering creativity and diverse forms of expression. Our approach to these permissions reflects our dedication to user-centric design and the creation of a socially immersive environment.

9. Monetization:

This project won't have any money-related features because it is focused on testing the technology, not making money. By keeping things simple and free of commercial technologies, we can better understand how well our technology works. This way, we can improve it without distractions, making sure it does what it's supposed to do. Our main goal is to test and refine our tech for future use, and that's where all our efforts are going.

10. Testing Devices and Environments:

In the testing strategy, a blend of local development and user group testing is employed to ensure the application's robustness across diverse devices and environments. During local development, Docker containers play a vital role in efficiently running backend servers, while Expo is utilized for front-end mobile development, streamlining the creation and testing of the app's user interface. To maintain code integrity, unit tests are conducted using the React Native Testing Library (RNTL) for the front end and UnitTest for Python on the back end. Beyond individual testing, selected users from both internal and external circles actively participate in user group testing. This collaborative approach identifies and addresses critical bugs, ensuring optimal performance and meeting user expectations across different testing scenarios.

11. Performance Requirements

The performance requirements for the application encompass vital interface elements that enhance user experience. Firstly, the integration of Google Maps into the app interface is crucial for displaying user locations and defining relevant radius parameters. Ensuring seamless navigation and geolocation features, this integration plays a pivotal role in the overall functionality. Additionally, location permissions are a key aspect, enabling the app to access device locations for personalized experiences. Push notifications serve as an essential interface requirement, allowing timely alerts and updates to users. Third-party login options will eventually be added through Google, Facebook, and Apple to streamline the onboarding process, providing users with convenient and secure authentication alternatives. The account creation process involves essential elements such as profile picture, username, and other pertinent details, contributing to a comprehensive and user-friendly interface.

Methodology

Design Pattern

The Observer design pattern will suit this project the best. Herd-It's system functions primarily on location and will be constantly updating. With this, the observer pattern will be able to watch and check when posts or users enter a different state that will either limit or send posts to the user's feed. With the events of location changes, Herd-It will be able to have real-time updates to the main feed. User interactions will also have an effect on the states of different objects in the system. Along with those considerations, with Herd-It being a mobile application, notifications will be sent to users' devices. The observer pattern will allow this to be done efficiently and quickly.

Architectural Design Documents

Database Schema: This illustrates what tables we will have, their respective fields, and how they interact. Found in appendix entry A.4.

Containers Diagram: This diagram illustrates high-level technology choices and interactions between the different containers. A "container" is essentially a virtual machine that manages dependencies and requirements for a backend server, mobile app, or database. Essentially, anything that can host code or data. The following diagram shows the logical containers that make up Herd-It. Found in appendix entry A.5.

Backend Cloud Architecture: Herd-It will be hosted on AWS Amplify. Appendix Entry A.6 shows a detailed diagram of how AWS amplify works.

Flow Chart of Backend Algorithms: The flow chart in Appendix A.7 shows the general the flow of how to view requests will work in the backend.

Trade-Offs

Since this project will be constructing a minimum viable product (MVP), many of the technologies in the system are subject to change. With this noted, all of the technologies were chosen because they are fast, cost-effective, and allow for rapid development. This project only has four months of development. For the MVP to be functional with the minimum requirements, the team will have to be able to develop quickly. The technologies outlined above enable the team to do exactly that.

Using React-Native will allow the team to develop for two platforms at once. This essentially cuts development time in half. While React-Native allows for rapid development, it has the downside of not being as performance-oriented. Even though performance may not be as high as a fully native application, the application doesn't have strict requirements to be completely performance optimized.

Django also allows for quick and easy development, but it comes with its tradeoffs. Django is a framework that doesn't allow extremely flexible components and features. With this considered, Django will still be a strong choice for development in the short project schedule.

The tech stack, featuring React Native, Django, and MySQL, brings advantages like cross-platform development and rapid backend construction. However, trade-offs include potential performance challenges with React Native, Django's opinionated nature, and considerations for real-time features and schema changes. Balancing these factors is crucial, aligning strengths and trade-offs with the project's unique requirements.

Expected Results

This project's minimum viable product should contain three key elements. The team wants to develop a fully functioning, neat, and put-together user interface that suits the functions of Herd-It nicely. The team also wants to showcase its unique approach to social media. To show this, the backend algorithms should be thoroughly fleshed out and working properly. The heavily location-based feed algorithms may have a high load on the hosting and database systems. This may be an area of concern. The final area that the team wants to focus on is the launch of the application. The team would like to test the MVP on various people within a semi-large area to nail down the proof of concept.

Accomplished Results

A Minimum Viable Product including:

- User Sign-in and Account Creation
- Password Authentication
- Profile Customization
- Main Feed Viewing
- Post Creation & Deletion
- Post Commenting & Liking
- Posts Viewable on a map
- Cloud Hosting
- Mobile Device Accessible

A Group Of Test Users including:

- Friends & Family
- Developers

References

GeeksforGeeks. (2020, August 30). Top 10 Reasons to Choose Django Framework for Your Project. https://www.geeksforgeeks.org/top-10-reasons-to-choose-django-framework-for-your-project/

Kamienski, N. (2023, November 23). React Native Pros and Cons - 2023 Updated. Pagepro. https://pagepro.co/blog/react-native-pros-and-cons/#:~:text=What%20is%20React%20Native%3F%20%20%20%20Business,Debugging%20and%20troubleshooting%20%202%20more%20rows%20.

National Council on Vocational Education. (1991). Solutions. Amazon. https://aws.amazon.com/solutions/guidance/social-media-insights-on-aws/

Singh, C. (2023, November 20). How the Twitter Algorithm Works & Tips to Beat It in 2023. SocialPilot. https://www.socialpilot.co/blog/twitter-algorithm

Slidebean. (2019, November 14). Startup Forensics: Yik Yak: A Deadly Combination. *Medium*. slidebean.medium.com/startup-forensics-yik-yak-a-deadly-combination-d84ffbe672fe.

Twitter. (2023, November 30). Twitter's Recommendation Algorithm. Retrieved from https://blog.twitter.com/engineering/en_us/topics/open-source/2023/twitter-recommendation-algorithm

Appendix

A.1

Nicholas Sulzbacher

Phone: (406)260-0464 Bozeman, MT
Email: NicholasSulz.business@gmail.com

OBJECTIVE:
A current Montana State University student studying computer science. Strong worker in a team environment while also maintaining a high standard for his work. Looking to gain professional experience in software, business, and project management.

SUMMARY OF SKILLS:

- Well-versed in C++/Java/Python and various other programming languages.
- Thorough understanding of computing hardware.
- Experienced in project management software like Smartsheet.
- Strong in both written and verbal communication.

EDUCATION:
Montana State University 2019-Present

- Current student pursuing a Bachelor's in computer science with a minor in business administration.
- Notable Courses:
    - Software Engineering
    - Multidisciplinary Engineering Design
    - Data Structures and Algorithms
    - Computer Science Theory
    - Interdisciplinary Project

Stillwater Christian School 2015-2019

- High school diploma

A.2

Gabriel Ewsuk

gabeewsuk@gmail.com
Linked-In: https://www.linkedin.com/in/gabe-ewsuk

SUMMARY OF QUALIFICATIONS

With a robust foundation in finance and technology, I bring a unique blend of expertise perfectly suited for the fintech sector. I excel in leveraging technology, including React, Python, and Redux. Furthermore, my internship experience has provided me with invaluable insights into agile development methodologies, team collaboration, and efficient project management, allowing me to deliver results quickly.

● Excellent task management. Ability to handle multiple projects simultaneously.
● Experience and proficiency in translating information from technical to executive/management terminology.

EDUCATION

Montana State University, Bozeman, MT Aug 2019 - May 2024 [Senior] ***Bachelor of Science, Computer Science [Minoring in Finance]***

- ***GPA:3.89***

CERTIFICATIONS & PROFESSIONAL TRAINING & AWARDS

● **Finalist in NVC, CU Boulder's Pitch Competition** [Spring 2020] • Won 2nd place and $13k in prize money amongst 80 competitors
● **AWS Certified Solutions Architect - Associate** [In Progress] **PROFESSIONAL EXPERIENCE**

Campaignify, Remote

Co-Founder (02/20 - 02/22)

- Grew business from startup to six figures in revenue within six months
- Grossed over $350,000 in revenue
- Managed hundreds of influencers
- Garnered over 72 million views for clients
- Worked with Skittles, Aero Postale, P&G, and Taimi
- Developed systems ran on AWS & MongoDB database system
- Hired and supervised two part-time employees
- Fulfilled dozens of marketing campaigns under tight deadlines
- Maintained communications with brands for revision & and customer satisfaction

Land Id, Remote (Bozeman)

Full Stack Software Engineer Intern (05/23 - present) - ongoing

- Merged 26 personal pull request tickets
- implemented 12 new features
- Solved 14 bug and technical support-related issues
- Implementing 10 Hubspot custom events
- Led a front-end project to increase conversions to achieve a 1% increase in weekly revenue
- Identified inaccurate market targeting taking up 10% of the total impressions
- Became proficient in React, and Redux, and familiar with Rails.

TECHNICAL SKILLS

Networks PowerShell & Bash Windows & Linux OS Exchange AD & IAM Management Office 365
DNS Virtualization Databases VPN & Firewalls
Python, Java, C, React System & Network Monitoring

A.3

Completed Work As of 4/30

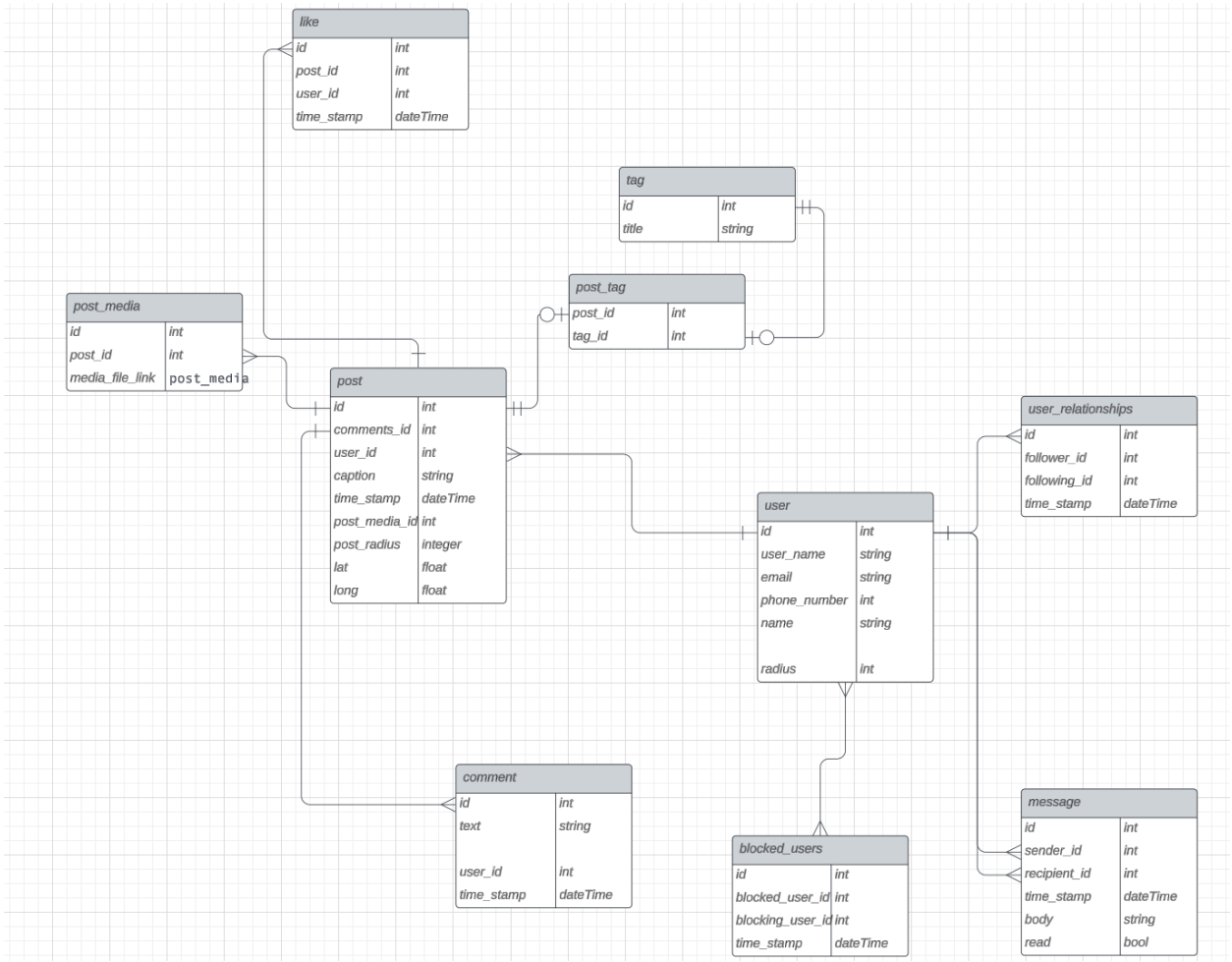
Gabriel Ewusk:

- ER Diagram Design
- Cloud Architecture Diagram Design
- Containers Diagram Design
- Similar App Research
- Technology Research
- Database Design and Modeling
- GitHub Repository & Workflow Setup
- Backend Design and Code
    - Framework Configuration
    - Models
    - View
    - Serializers
    - Business Logic for H3 Library & Entity Relationships
    - 126 unit and integration tests
    - Documentation for endpoints & setup
    - Dockerization
- Front-End Design & Code
    - Components
    - State Contexts
    - Navigation
    - Panels/Pages
    - Reusable Request Methods
    - Provider setup -> s3 and location services
    - Hooks - filtering posts on the front end
    - Framework Configuration
    - Manual Testing With Simulator
    - Hooks Unit Tests
    - Server Side Communication
- Cloud Services Setup & config
    - EC2
    - S3
    - IAM Rules

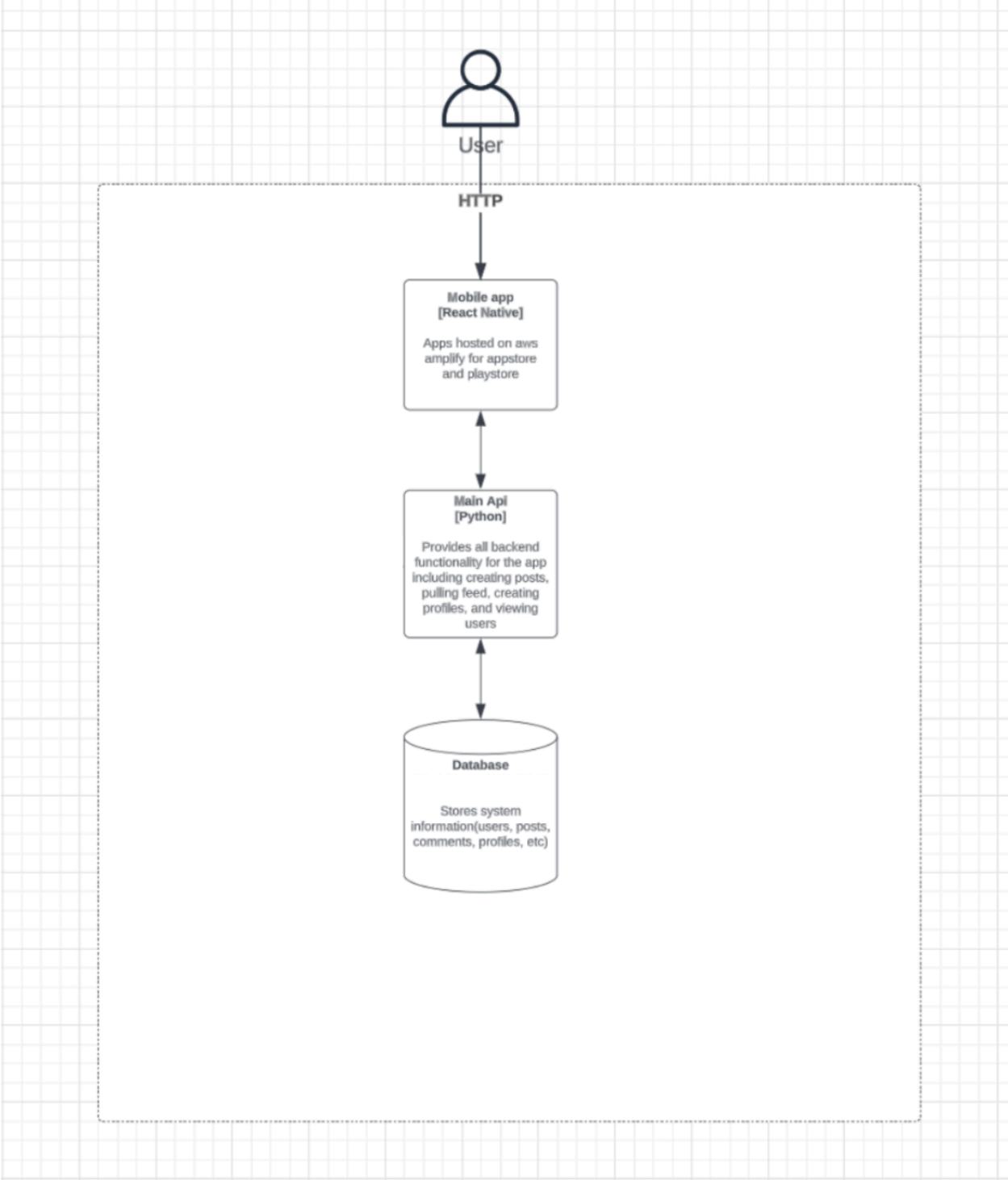
Nicholas Sulzbacher:

- UI/UX Research
- Competing Application Research
- Proposal writing and editing
- Presentation writing and editing
- Algorithms Research
- Initialized Development Environment
    - Using Expo and the Metro bundler, set up the React-Native compiler
    - Included README for installation steps
- Initial UI/UX Design
    - Asset Creation
    - This was adjusted later to be easily integrated with the backend

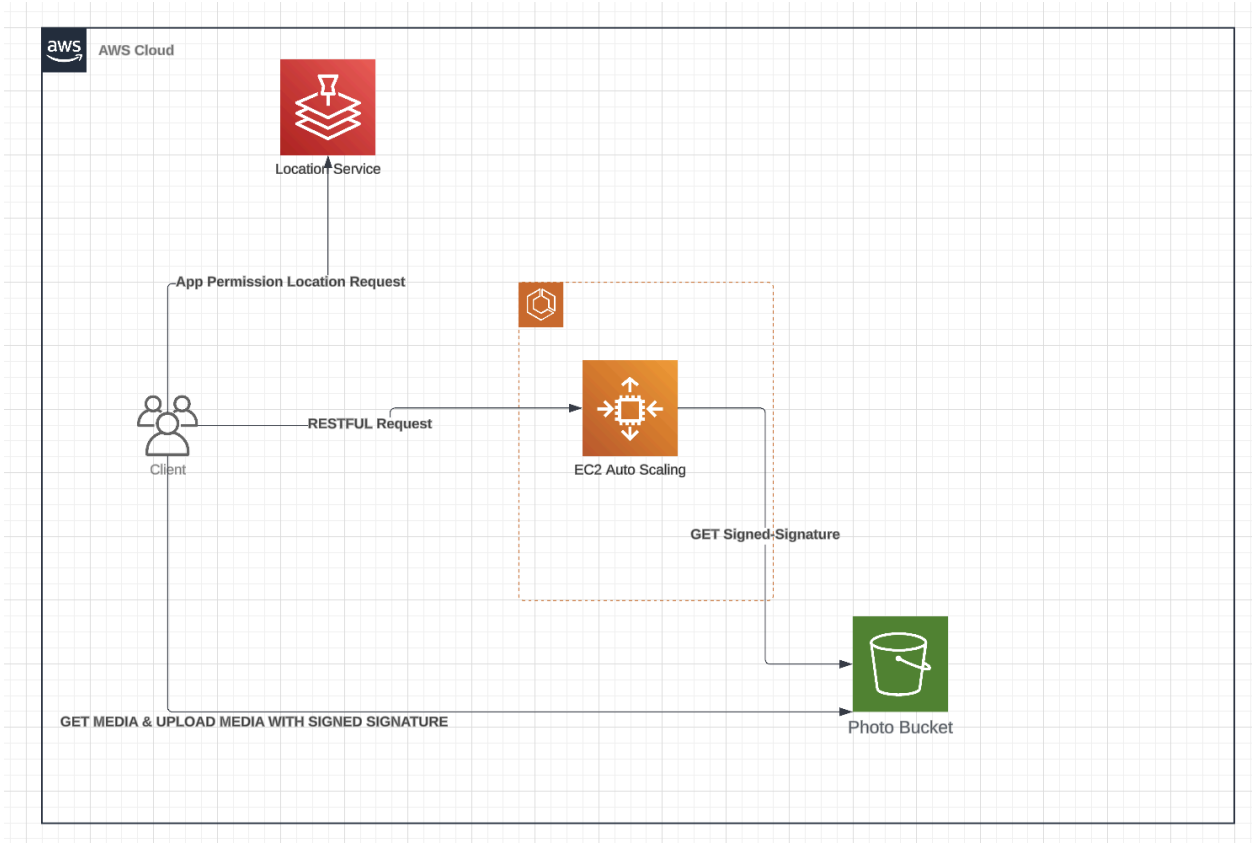
A.4


like
id int
post_id int
user_id int
time_stamp dateTime
tag
id int
title string
post_tag
post_id int
tag_id int
post_media
id int
post_id int
media_file_link post_media
post
id int
comments_id int
user_id int
caption string
time_stamp dateTime
post_media_id int
post_radius integer
lat float
long float
user_relationships
id int
follower_id int
following_id int
time_stamp dateTime
user
id int
user_name string
email string
phone_number int
name string
radius int
comment
id int
text string
user_id int
time_stamp dateTime
blocked_users
id int
blocked_user_id int
blocking_user_id int
time_stamp dateTime
message
id int
sender_id int
recipient_id int
time_stamp dateTime
body string
read bool

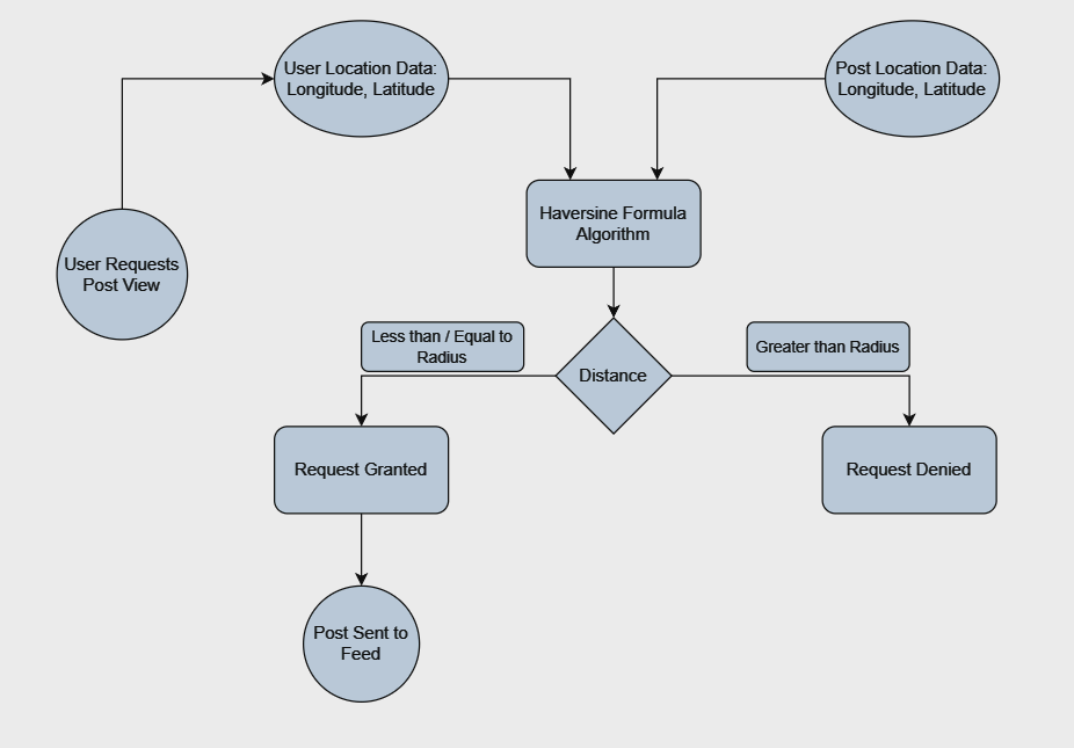
A.5


User
HTTP
Mobile app
[React Native]
Apps hosted on aws amplify for appstore and playstore
Main Api
[Python]
Provides all backend functionality for the app including creating posts, pulling feed, creating profiles, and viewing users
Database
Stores system information(users, posts, comments, profiles, etc)

A.6


AWS Cloud
Location Service
App Permission Location Request
Client
RESTFUL Request
EC2 Auto Scaling
GET Signed-Signature
Photo Bucket
GET MEDIA & UPLOAD MEDIA WITH SIGNED SIGNATURE

A.7


User Location Data: Longitude, Latitude
Post Location Data: Longitude, Latitude
User Requests Post View
Haversine Formula Algorithm
Less than / Equal to Radius
Distance
Greater than Radius
Request Granted
Request Denied
Post Sent to Feed