Team 20 - Product Backlog

Pinnect

Rafael Zhu, Tianyi wang, Benyu Jia, Yuquan Xu

Problem Statement:

In current social media like Facebook and Instagram, text and images are heavily focused while the location is optional, but Pinnect is a purely location-based social media. A Pinnect user can drop a pin with text, images, or #tags at his current location, browse dropped pins around the world, and interact with other pins by liking, sharing, or writing a comment on them. To ensure pins are up-to-date, they are normally deleted after 24 hours, but trending ones can last longer. We hope through Pinnect, people can discover what is happening around them and connect to each other in a more open way.

Background Info:

(a) General Background

Sometimes, it is hard to know what is happening around you.

- Is there a newly opened restaurant?
- There is a loud crowd on the street. What is going on?
- Was there something special that happened in your street last night that you just missed, so you have no ideas what your friends are talking about on the second day?

Pinnect tackles this problem by introducing the pins, which are blocks of text or images pinned to the map. In Pinnect, users are allowed to browse the pins around the world and leave pins at their current location. Users are also able to interact with other pins by liking and sharing them.

(b) Targeted Users

The targeted users of Pinnect are the enthusiastic, curious, and sharing people who would like to keep up with the latest social trend and share their opinions or life stories along with their locations. This location-based social media also draws people who are willing to make friends with complete strangers nearby. It opens up the possibility for more connection and communication.

(c) Similar Platforms

Social media like Facebook and Instagram, are heavily focused on messages and images while location sharing is optional, but Pinnect is a purely location-based social media. Namely, contents are tied tightly with locations.

There are already some social media that put effort in location-content association

- Zenly: Users can see friends' locations in real-time and send messages to them.

- Snapchat: An extended version of Zenly that allows users to send a video at the current location that is viewable by others.

(d) Limitations
The social media above do not resemble our ideas:

- Zenly: Does not allow users to generate contents, and no features involving interacting with others yet to be connected are implemented.
- Snapchat: Only supports sharing videos; the granularity of locations is coarse, i.e. your content spans the whole block, instead of the precise location; others are unable to immediately discover the content.

Pinnect addresses the shortcomings in the similar applications above by enabling users to share versatile contents, interact with people around the world, and put pins on more precise locations.

Functional Requirements:

1. As a user, I would like to register a Pinnect account.
2. As a user, I would like to login to my Pinnect account.
3. As a user, I would like to reset my password.
4. As a user, I would like to edit my user profile.
5. As a user, I would like to edit the preference of the account such as the language and theme of the Pinnect.
6. As a user, I would like to have a button that directs me back to my current location.
7. As a user, I would like to see the map with pins created by others.
8. As a user, I would like to see the map with pins only created by my friends or other requirements.
9. As a user, I would like to move the map freely.
10. As a user, I would like to be recommended by several pins with the nearest distance from myself.
11. As a user, I would like to open the available pins on the map and see its contents.
12. As a user, I would like to leave a like on any pins.
13. As a user, I would like to share an interesting pin to other users.
14. As a user, I would like to leave a comment on a pin.
15. As a user, I would like to see the comments on a pin.
16. As a user, I would like to have a list that stores my previous liked pins.
17. As a user, I would like to report a pin which violates the rule.
18. As a user, I would like to create a new pin at my current location of the map
19. As a user, I would like to control the visibility of my pins (public/only for friends/private)
20. As a user, I would like to add text on the new pin.
21. As a user, I would like to post new pins with an image including preview function
22. As a user, I would like to edit or delete my previous pins.
23. As a user, I would like to broadcast their pins to people within a range of 100 meters.

24. As a pin owner, I would like to see the number of people who viewed the pin.
25. As a pin owner, I would like to see the number of people who shared the pin.
26. As a user, I would like to visit a user’s profile by clicking on a pin he/she created.
27. As a user, I would like to send a friend request to others.
28. As a user, I would like to find a user by an account id.
29. As a user, I would like to see the list of my friends.
30. As a user, I would like to see my friend’s location.
31. As a user, I would like to open a chat box with a friend.
32. As a user, I would like to send or receive texts from a friend.
33. As an administrator of Pinnect, I would like to see the number of the registered account and the number of the users currently online.
34. As an administrator of Pinnect, I would like to send a warning to the users who violate the rule.
35. As an administrator of Pinnect, I would like to manage the user’s account.
36. As an administrator of Pinnect, I would like to delete any pins on the map.
37. As an administrator of Pinnect, I would like to see the number of pins.
38. As an administrator of Pinnect, I would like to see reports of pins or users that violate the rules.
39. (*if time allows*) As a user, I wish the system could recommend some users who have similar interests to me.
40. (*if time allows*) As a user, I wish to see some interesting (trending) pins around me.
41. (*if time allows*) As a user, I wish to see the popular/hot pins around the world.
42. (*if time allows*) As a user, I wish to see a user-friendly interface and easily manageable system.

Non-Functional Requirements:

Architecture

We plan to separate the frontend and backend completely. This would reduce the coupling in the project and help us divide our work more effectively.

The frontend will be written in React Native or similar scaffolding frameworks that support compiling the front-end to a mobile app.

The backend will be primarily written in Python. The server will provide several RESTful APIs with a high-performance web framework like Flask. Flask is a micro web framework written in Python that is fast and easy to maintain.

Other non-crucial components e.g. recommendation and data analysis systems are planned to be developed in Python due to its development efficiency and the numerous frameworks available for data analysis and machine learning.

Data Persistence and Security

We plan to use MongoDB for user data persistence, primarily because of its nature in storing mutable data structures with JSON and faster performance in handling data.

Security is our top priority in developing Pinnect, since users will produce privacy-sensitive information, but most importantly, a Pinnect account stands for an identity that cannot be violated illegally. We will develop rigorous permission validation and other security measures for all user operations on the server side. For example, users are only able to message other people or view restricted pins only when they are friends, and if they send an illegal request, it will be denied on the server side. All requests to the API must be authenticated and verified to prevent abuse.

Usability

We aimed to have a minimalism design for the frontend i.e. the interface should be very straightforward in order to make users spend more time for interaction. We also expect the interface to have a good response time i.e. users should not experience noticeable lag when moving maps, browsing pins, and viewing object details. Therefore, the server is expected to respond to a basic request within 500ms or less.