Team 20 Project Charter
Pinnect

Team Members:
Rafael Zhu, Tianyi wang, Benyu Jia, Yuquan Xu

Project Title:
Pinnect: Pin Your Life in a Parallel Universe

Problem Statement:
In current social media like Facebook and Instagram, text and images are heavily focused while 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.

Project Objectives
Develop an application that users could:

1. Create user accounts, set up profiles, and display them on a user-friendly web page.
2. Add friends and chat with them, and view their private *pins*.
3. Put a *pin* of variable types on the map with comments, #tags, or images to share their lives.
4. Interact with *pins* by leaving a like or sharing it with other users.
5. The existence time of the *pins* will be automatically increased if it receives likes or shares.
6. Look through the *pins* put by other users all over the world to discover something you are interested in.
7. Control the visibility of *pins* and broadcast their *pins* to users within 100 meters.

Stakeholders:
User: People who have interests to browse *pins* around the world or interact with anyone who left a *pin* on the map.
Developers: Rafael Zhu, Tianyi wang, Benyu Jia, Yuquan Xu
Project Manager: Yu Shi
Project Owners: Rafael Zhu, Tianyi wang, Benyu Jia, Yuquan Xu

Project Deliverables:

1. Develop a front end website application with an interactive map that contains *pins* around the world and functionalities to create and interact with *pins*.
2. Deliver a back-end that offers REST APIs allowing the front-end to interact with the server, developed in Python or Java.
3. Use a database to store *pins*, classify the *pin* type, tap top *pins*, and mark *pins* that last longer than 24h.
4. Apply algorithms like KNN to find broadcasting pins and recommend them to nearby users dynamically.
5. Use machine learning like decision tree/naive bayes/regression to classify *pin* types.