Process Description

Software Toolset

- Java and Android Studio IDE for front end application - simple, well documented, universally supported
- Android Bootstrap - easy to use, simple, good looking and universally supported UI elements
- Git for version control and bug tracking - already using it for development, has tools for all these built in
- Retrofit for http requests - allows APIs to be treated as Java interfaces, simplifies lots of otherwise complicated and tedious network code
- Google API for logins - allows our app profiles to be tied to Google accounts, which simplifies our authentication and makes our login as secure as Google's is
- Python for backend API and database servers, with Flask microframework - minimizes boilerplate, allows easy creation of easily configurable server instances
- Amazon Web Services for database and API server hosting - popular, reliable web host capable of handling whatever we need it to
- MySQL for database - it's free, well documented, and reliable

Group Dynamics

- Taylor Yoon will be our Project Manager.
- We will have a Frontend Team, which works on the app itself, and a Backend Team, which works on the API, algorithms, and database our app will use. Who is on what team will fluctuate as people figure out what roles they are most comfortable in. This will prevent people from getting stuck in roles they aren't interested or good at.
- In case of disagreements, people will present their arguments, and the group will vote on a decision. If in doubt, we may also chose to consult the client for clarification. This ensures the largest number of people possible are happy, encourages flexibility, and ensures quick resolution of issues (which is important in a 10 week project).

Schedule

- **Week 1:** Entire team works on Design Specification and setting up the deployment pipeline, including CI test suites.
- **Week 2:** Frontend Team works on creating a compilable, installable Zero Feature release, while Backend Team works on setting up AWS, basic API functions, and simple database table setup.
- **Week 3:** Frontend Team creates login screen and profile preferences (inventory and tag preferences) page, while Backend Team sets up API and database servers to accept and store profile info.
- **Week 4:** Frontend Team works on creating tag search window, trade window, and requests window. Backend Team works on creating search functions and API changes to support these windows.

- **Week 5:** Frontend Team works on push notifications, user matching button and screen. Backend Team works on creating matching algorithm and API calls for it.
- **Week 6:** Team works on bugfixing and catch up for things we could not accomplish in previous weeks.
- **Week 7:** Fix bugs first, then create external documentation, then add stretch features after if there is time. What stretch features we will implement first will be determined in team meetings, once we know how the whole system is structured and how complex each feature will be to implement.
- **Week 8:** Organize presentation and present as team.

Risk Summary

- Figuring out how to set up AWS in time to test everything we need to. AWS is a popular, reliable service, so finding help using it should be relatively easy and minimize risk.
- If our Frontend and Backend teams don't know exactly what the other is doing, it will lead to a non functioning app. We can minimize this through constant testing as well as abiding by the documentation we write for each part. Weekly meetings and Slack chat will help us communicate and minimize chances for miscommunications.
- Getting all major features out in time may be difficult. As our schedule is set up now, we believe it will be difficult but possible to get all major features done in time. However, we have to be ready to cut features for the sake of time if necessary. The first feature to go will likely be automatic matching, because without it, the user will still be able to accomplish their main goal in using the app, which is to trade for other users' items. However, it will be less convenient for the user.
- One person doesn't finish what they need to in time. Communication is the biggest way to avoid this. I someone runs into an issue, there needs to be a way for someone else to step in as fast as possible. Everyone needs to watch the team chat in order to ensure they know if someone else needs help.

- We have a crack team of HCDE students ready to test and give feedback on our app. This will be most useful at the Zero Feature (basic UI feedback), Beta (feature and functionality feedback), and Feature Complete (bug testing) releases.