

Bae Watch

Fake Tinder For Fake Relationships

Illya Starikov

March 1st, 2017

There is a growing problem in America: millennials. Millennials are subjectively proven to be the laziest, financially illiterate, entitled, and downright spoiled generation. The most disturbing thing about the millennial generation: their inability to get married before the age of 21. This not only poses a problem with family, but friends as well. At any social gathering, whether it be dinner with friends, thanksgiving, or even the bar with friends, relationship questions for singles are unavoidable. Questions like the following can be expected:

- “Isn’t it time to settle down and find yourself a significant other?”
- “Ayo when you gunna bring round a girl like the last one, she was 🔥🔥”
- or “When are you going to find a good, Christian girl?”

A possible solution? Bae Watch.

1 Objectives

The objective of Bae Watch is simple: find an idealized relationship to ward off unsavory family members and friends. This is accomplished by leveraging the power of social networking. Users submit required information, which is mostly scraped off Facebook, and begin filtering through potential candidates. They do so through a swipe left for “like” and right for “oh hell no”. It is a Tinder-esque (Figure 1) mechanic, but there’s is something unique that Bae Watch brings: the kinds of data it requires.

Besides the basic information that Facebook asks for (date of birth, favorite movies, IP addresses, all previous browsing history, etc.), we request information about the company you are trying impress. Things could include:

- Religious preferences.
- Minimal salary.

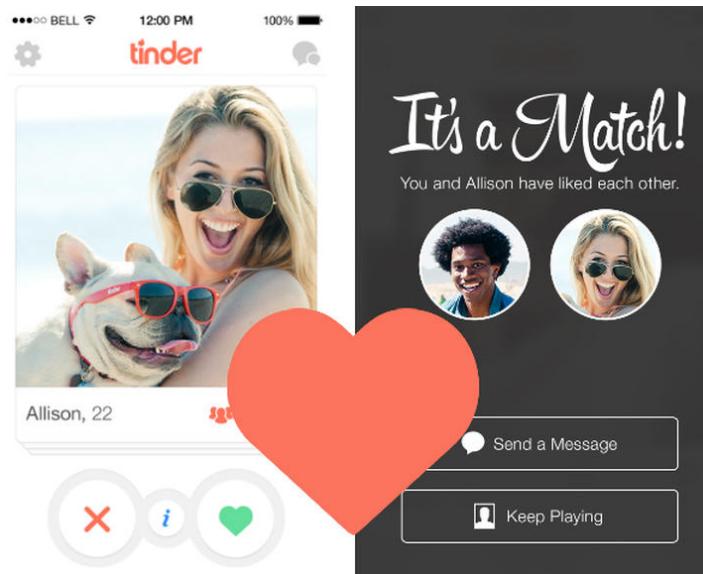


Figure 1: The real Tinder.

- Preferred level of education (i.e. Bachelors, Masters, Doctor of Philosophy) and graduating major.

And other similar data. After aggregating said data, Bae Watch's AI¹, Machine Learning², and Algorithms³ present the user with the most likely matches. The foremost will be the most likely to impress, and subsequent users will more likely be English majors.

2 Work Plan

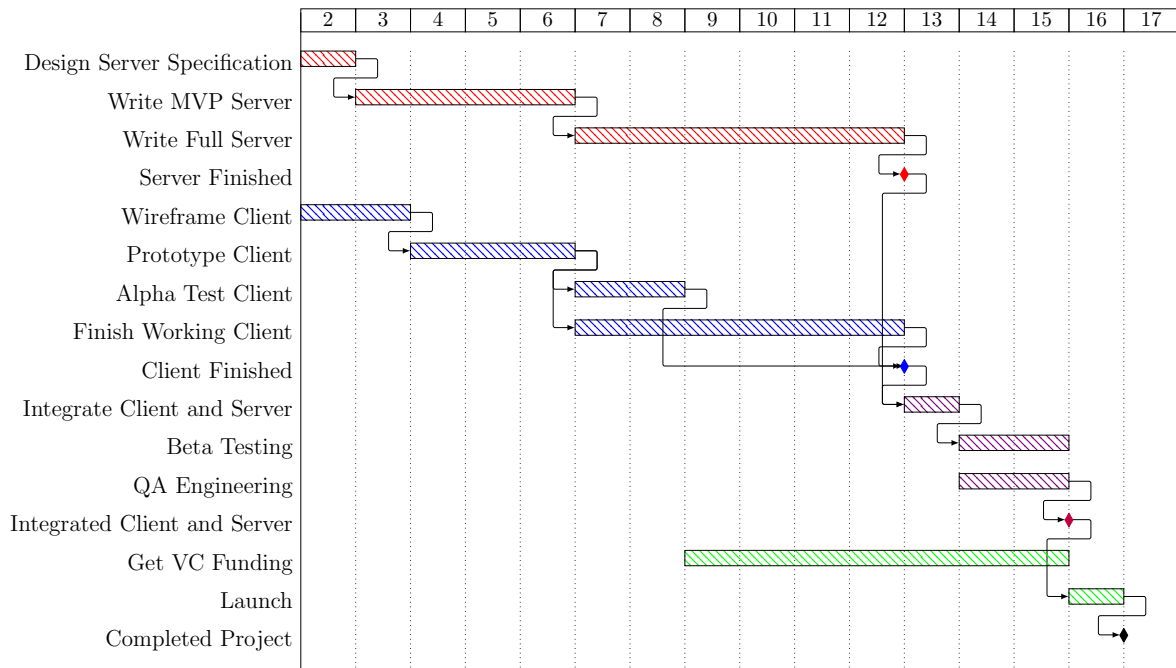
The week will begin on the second week of school, and prolong to the end of the semester. As any large scale engineering project, there will be a design phase — specifically, wireframes for the User Interface and a Server Specifications will be made. Upon completing said specifications are made, the majority of the process will be put into creating the server and client. Both the server and client will be available as private (separate) repositories on Github to monitor progress, commits, documentation, etc.

This can be eloquently described by the following Gantt chart.

¹A bunch of `if` statements.

²More `if` statements.

³You guessed it. `try` and `catch` blocks.



3 Deliverables & Milestones

The following items will be delivered upon completion.

Client and Server Code For the iOS, simply a link to the respective Github will be provided.

Server API Documentation An API reference guide (typeset in \LaTeX) be provided.

Client iOS Application An iOS application will be made available on the App Store for iOS 10 and above. This will showcase the integration of both the client and server.

In regards to milestones, they are (chronologically) as follows.

1. The full stack (server and client) will be finished at the same time, making integration easy.
2. The client and server will be integrated, and fully tested through an extensive suite of unit tests⁴ and a wide array of beta testers.
3. The team will get roughly generate a million dollars in venture capital funding by the end of the project.
4. The project will be finished at the final week of school.

⁴That I am not writing.

4 Task Breakdown

Task #1: Write Server

An asynchronous server will be written to keep track of users in a particular area. The server will also house the user data (i.e. names, IP addresses, preferences, etc.) and location data. It will be written in Python 3.

Creating the Minimal Viable Product (MVP) server consists of getting the minimal functionality finished, but enough to be usable for the client. Afterward, the server will be completed, with proper error handling, user authentication, and encryption.

Task #2: Create Client

A iOS 10 application will be created. This will involve creating the models, view controllers, programming custom views, etc. Some third party libraries will be used to make workload easier. During the entire development process, documentation will be maintained (via Doxygen) and polished at the end for an entire API design guide.

Task #3: Venture Capital Funding

Profit.