

Tindergarten (T.B.D.)

8th January 2016

Elevator Pitch

Crowdsourcing dating responses through Tinder's unofficial API.

Technology

- Elixir and Phoenix Framework (Ruby on Rails [or Node])
- Websockets
- Angular (React)
- Ionic
- Tinder Public/Private API
- Postgres
- HTML / CSS
- JavaScript

MVP Features

1. User should be able to view all matches with photos, name, age, and description.
2. User should be able to click on a match, and see the current dialogue already sent.
3. User should submit responses to a match, that other users should see in real time.
4. User should be able to upvote/downvote messages to their liking.
5. Every 10 minutes, the most upvoted message is sent to match

User Stories

- 1) **User can view all matches with match information**

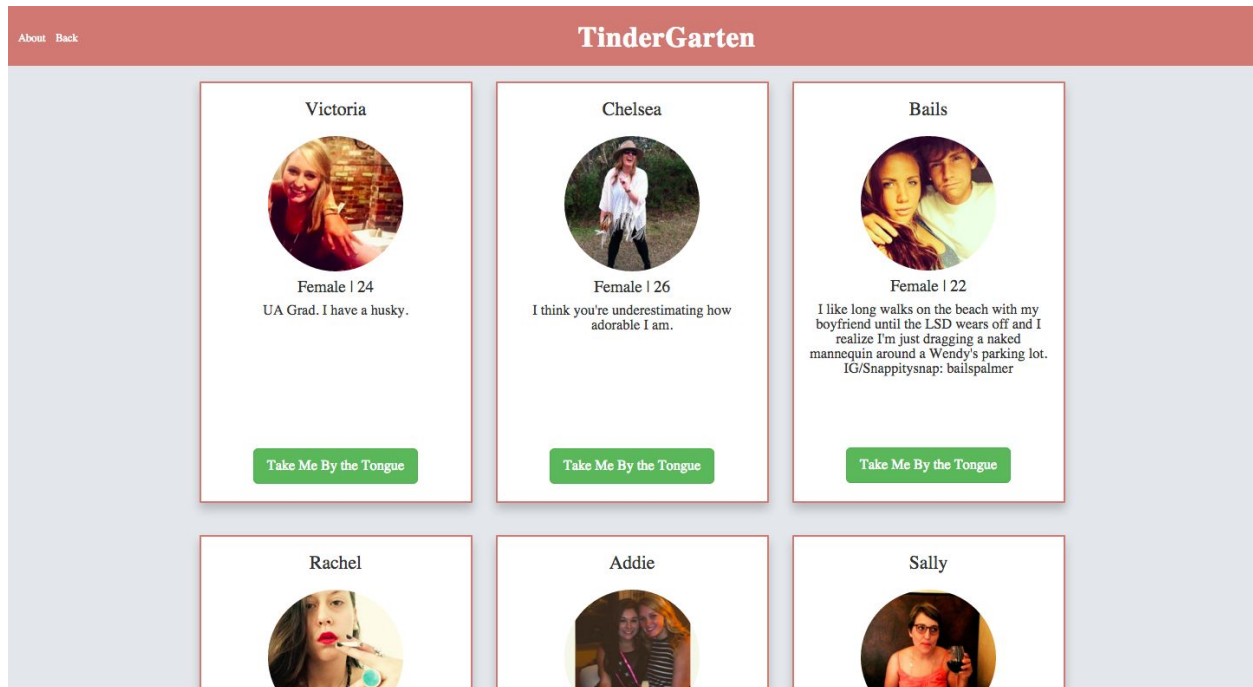
Value Statement

In order for a user to get the most out of Tindergarten, they need to be able to view all the available matches, with their name, age, description, gender, and photos.

Acceptance Criteria

1. All the available matches can be seen on the landing page by a user.
2. A user can see the basic info about all matches, and click on the them to see all the photos.

Wireframe



2) User should click on a match and see the current dialogue and chat window.

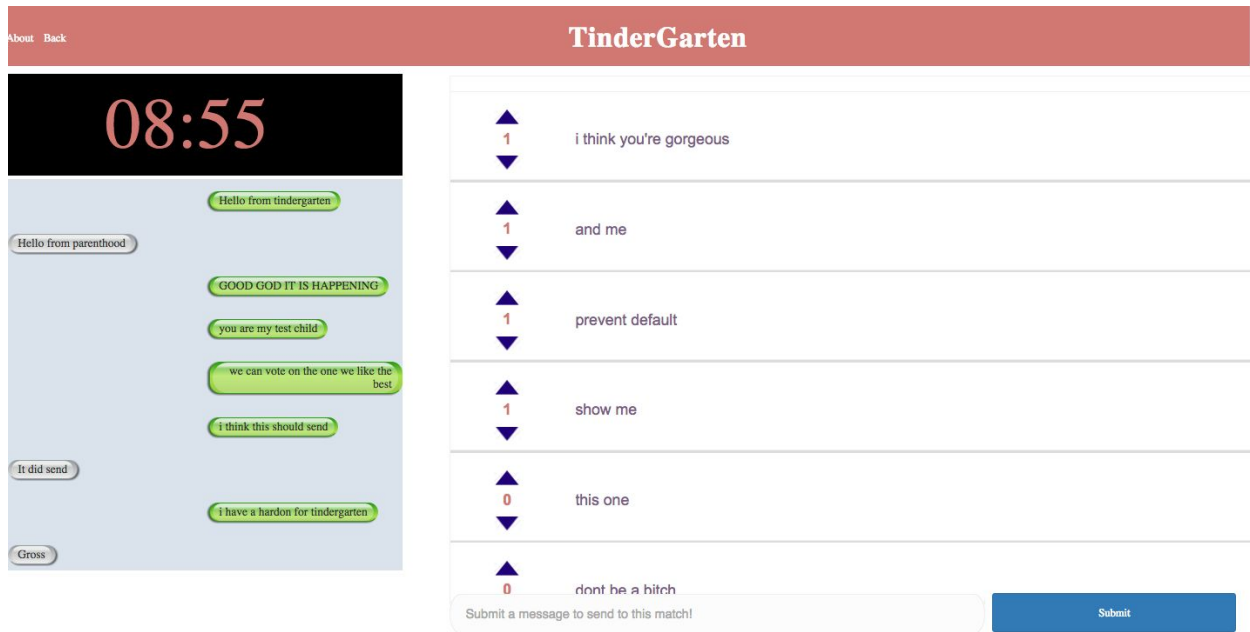
Value Statement

A user can interact with potential matches and other users of Tindergarten through the chat window.

Acceptance Criteria

1. All existing, actual messages between the match and us need to be visible to the user.
2. All current chat dialogue should be viewable to the user.

Wireframe



3) User can submit suggested chats

Value Statement

A user can submit a message to interact with other users of the site.

Acceptance Criteria

1. A user can type into the box.
2. A user can hit submit and the message goes to the database and displays on the screen
3. Other users, on submit message, will be able to see chat in real time.

4) A user can upvote/downvote chats.

Value Statement

A user can pick his/her favorite response to send a match by upvoting and downvoting current chats.

Acceptance Criteria

1. A user can see upvote/downvote arrows and current vote total.
2. A user can click on upvote/downvote arrows which will increment/decrement total votes
3. Other users can see upvotes/downvotes in real time.
4. Chats are sorted by vote total in real time.

5) Clock

To be continued...