Meetify

CS Senior Design 2021

Group Members

Dustin Seger
dustin.seger@hotmail.com

Jake Steuver
jakesteuver@gmail.com

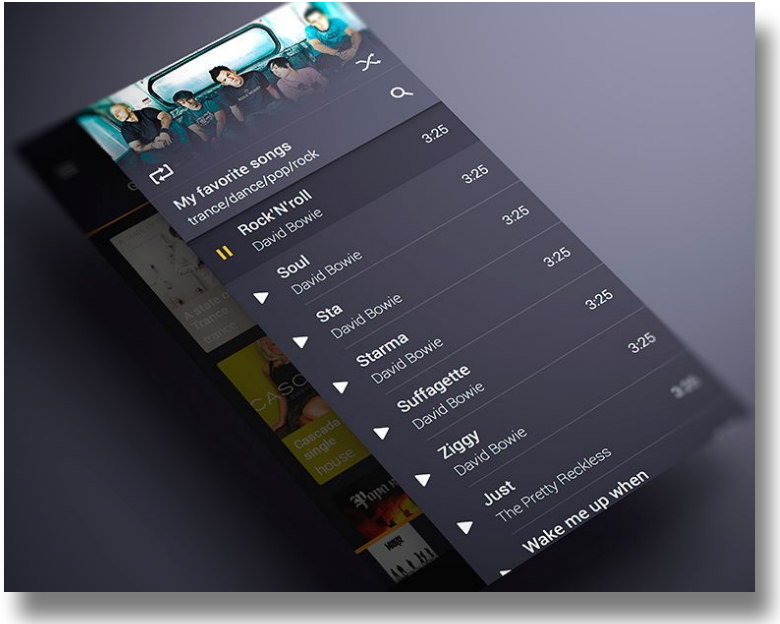
Rob Boeckermann
robboeckermann@gmail.com

Bret Patton (advisor) - bret.patton@ymail.com

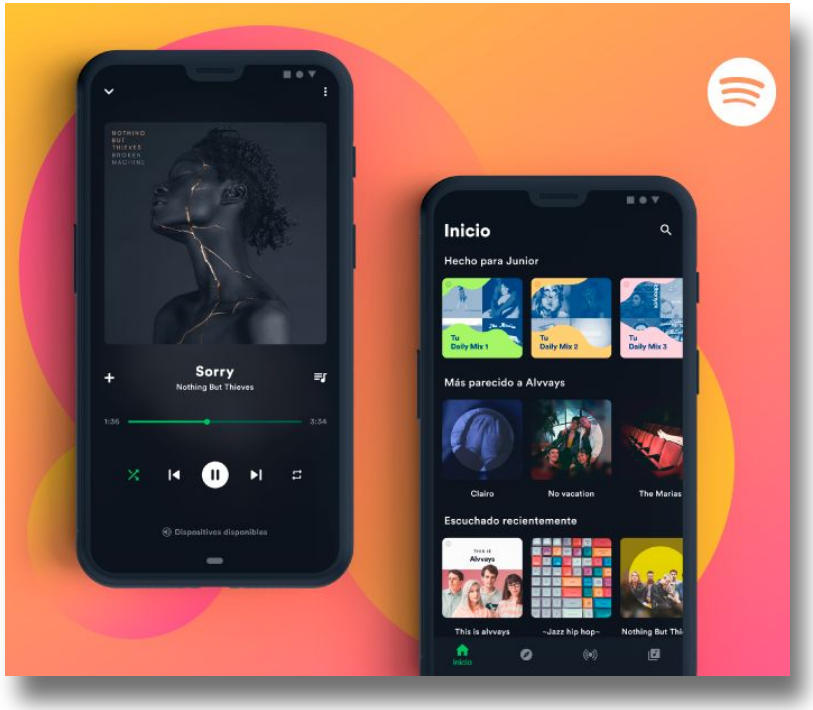
Project Purposes & Goals

In this day and age, there are countless ways to **meet people** and **listen to music** online, but **very few places to do both!**

Thus, our team wanted to create an app to combine these concepts, allowing users with **like music interests** to meet.



Intellectual Merits



Although some vaguely similar apps are out there, we have found them to be **unpopular** and/or **no longer supported**.

We think that, by tying it to the **Spotify** music library, we will be able to reach more people and create a vast network for people to meet!

Broader Impacts

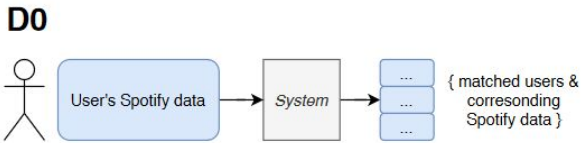
With social media being as impactful and far-reaching as it is today, it is important to have new ways of connecting with people.

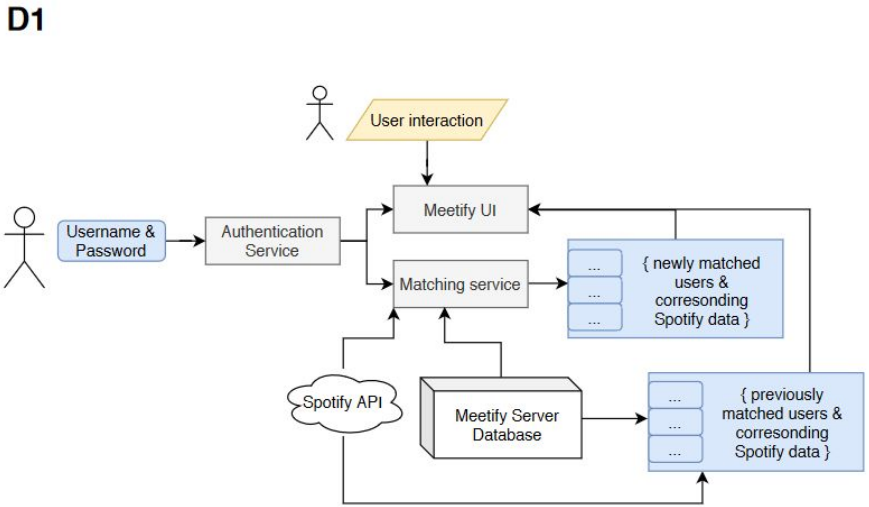
- Connecting via musical interest can lead to relationships that are different from typical social media relationships
- Meetify gives users a chance to embrace their music tastes and find others who do as well
- Building a community revolving around music

Design Overview & Technologies

- **Cloud Hosting** via **AWS**
  - Will host all other technologies
- **MySQL** database
- **Python**-based server (via **Django**)
  - Creates a deployable server
  - Acts as as a **REST API**
- **HTML/JS** web-based application (via **React**)
  - Written as website
  - Can be distributed to all major platforms
    - *Only Windows currently supported*

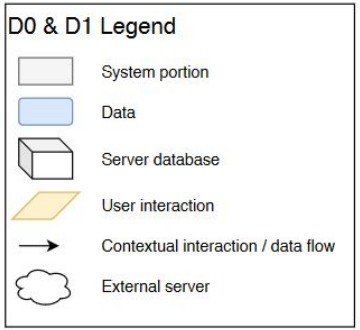
Design Diagrams



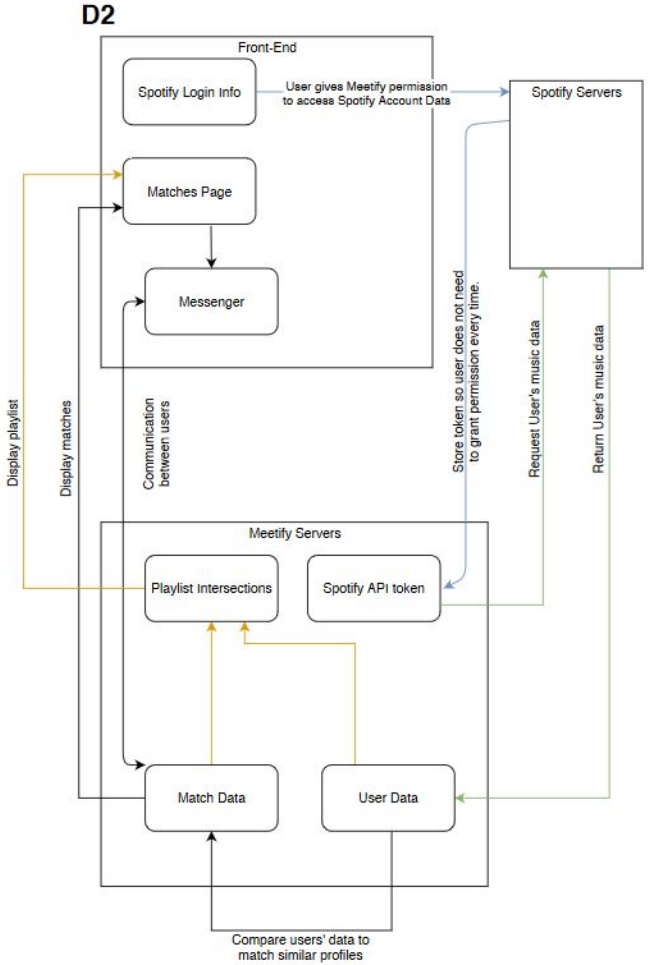


Meetify Diagrams

Goal: to create a service that allows user to match with other users based on music interest, using data from the Spotify API.



Design Diagrams (cont.)



Meetify Diagrams

Goal: to create a service that allows user to match with other users based on music interest, using data from the Spotify API.

D2 Legend

- System portion
- Data
- Contextual interaction / data flow (identical colors imply corresponding inputs/outputs

Milestones

Connect a Django server to the database	**Complete**	12/18/20
Develop Django server API capabilities	**Progress**	3/7/21
Design a prototype front-end UI	**Complete**	2/12/21
Connect the server and front-end	**Progress**	3/21/21
Host the server and UI on the cloud for public access	**Progress**	3/28/21
Develop & enhance user matching algorithm	**Progress**	3/14/21
Develop back-end messaging service	**Progress**	3/28/21

Results

- Back-end
  - MySQL database designed and hosted on AWS
  - Spotify account linking
  - Functionality to intersect users' "Liked Songs" playlists
  - Functionality to determine users' tendencies to prefer certain qualities/features of songs (acousticness, danceability, tempo, etc.)
- Front-end UI layout
- Front-end / back-end integrations
  - Account creation
  - Log-in

Challenges

- Learning Django framework
- Designing a matching algorithm
  - Using scores from elements of listening history
  - Making it scalable
- Creating messaging service
  - Utilizing Django signals and channels
- Following REST API concepts
- Learning React & related packages
- Front-end design & cleanliness
  - Implementing Material Design via a React package
  - Utilizing CSSTransition & Material UI transitions
  - Creating centralized style
- Front-end data flow management
  - Incorporating Redux
- Error handling
  - Keeping server API documentation up-to-date
  - Expecting all possible errors in front-end

Contributions

To develop and address challenges, we tended to **cross boundaries**, working together when possible.

But, in general:

- Front-end
  - Dustin Seger
- Back-end
  - Rob Boeckermann
  - Jake Steuver