Review

- Initial Goals
  - Organize your wardrobe
  - Log past outfits
  - Mobile-friendly UI
- Features
  - Upload images of individual articles of clothing + outfits
  - Organize clothing by category, color, material, style
  - Show article details
  - Log past outfits with date worn info
  - Clothing filtering capabilities
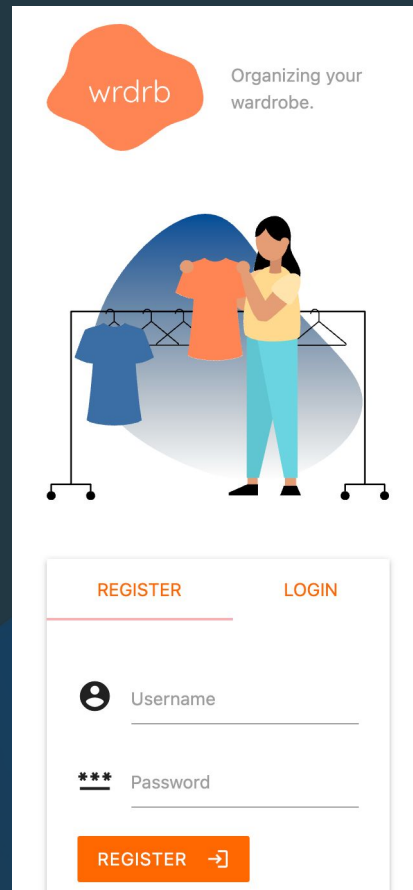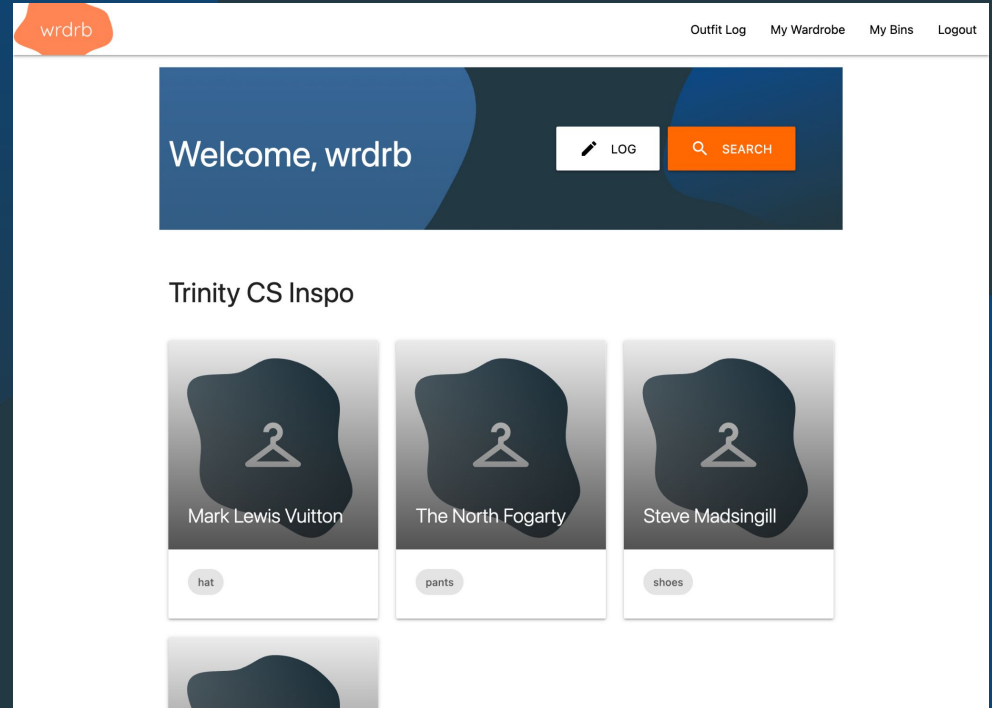
Project implementation

Tom: UI, Login flow

Ryan: Bin flow + Article flow

Alex: Article flow

Lizzie: Outfit flow

Materialize

API

Demo

Retrospective

the good the bad and the ugly

- databases are hard
- regular meetings were useful
- crunch time SUUUUUCKS
- making an API!!
- commenting
- splitting up files - nothing was super long

what would we do differently

- ScalaJS? - sharing data models, enums, TYPES, etc
- Breaking down major features better, setting *achievable* goals
- we could have combined outfits and bins because we only realized their similarity later
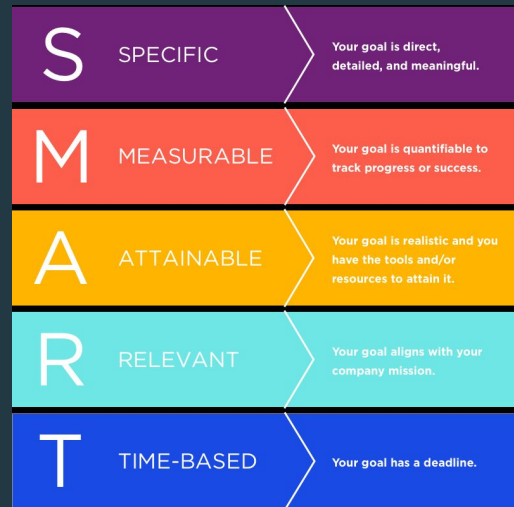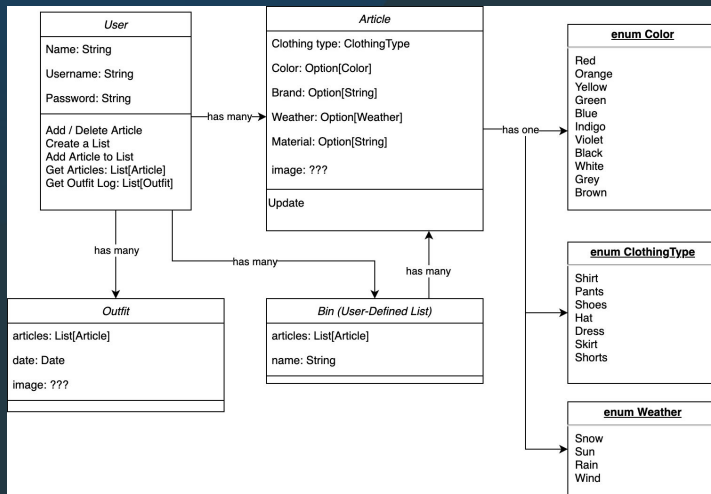
Team dynamics

- Trello
- Discord
- Google slides/docs
- Gather
- Draw.io
- Github

keep it

simple

sweetie

Did we? debateable.

User

Name: String

Username: String

Password: String

Add / Delete Article
Create a List
Add Article to List
Get Articles: List[Article]
Get Outfit Log: List[Outfit]

has many

Article

Clothing type: ClothingType

Color: Option[Color]

Brand: Option[String]

Weather: Option[Weather]

Material: Option[String]

image: ???

Update

has one

enum Color

Red
Orange
Yellow
Green
Blue
Indigo
Violet
Black
White
Grey
Brown

has many

has many

has many

Outfit

articles: List[Article]

date: Date

image: ???

Bin (User-Defined List)

articles: List[Article]

name: String

enum ClothingType

Shirt
Pants
Shoes
Hat
Dress
Skirt
Shorts

enum Weather

Snow
Sun
Rain
Wind

Scaling

- Storage server for images
  - Current solution: Base64 encoding images in the Database
  - Current solution is bad, even if it works
- Switch client-side language away from Javascript
- Load balancing, database sharding, etc...

Questions?

EBEBEB

3A6EA5

004E98

223843

FF8552

wrdb-palette

coolors