

1. Team Organization

A. Task Management

Michael

- User endpoints (first day)
- everything else in tweet
- tweet list folder
- user service
- user page
- most of styles css
- everything at bottom of app js

Mark

- Tweet endpoints (first day)
- hashtag component and template
- hashtaglist folder
- hashtagnotfound folder
- tweet like, get users who liked, context, direct replies, direct reposts
- user list folder
- user not found folder

Travis

- Hashtag endpoints (first day)
- Validate endpoints (first day)
- context folder
- hashtag service
- navbar folder

- reactivate folder
- settings folder
- signIn folder
- signUp folder
- global
- some styles.css

All

- white boarding
- states app js

B. Conflict Resolution

- Merge conflicts

Got together and fixed based on whos was what

- Design conflicts

Talked and focused on major points and didnt talk about minor details.

Were able to zero in on the feature we wanted to add and were all willing to concede minor details or design prefereances for functionality which lead to rapid conflict resolution

C. Time Management

Monday

- Endpoints
- States
- Initial whiteboard
- Layed out structure for rest of week

Tuesday

- Functionally with html templates
- Modular components and began connecting

Wednesday

- Full functionality
- Html
- Styling

Thursday

- More styling
- Debugging
- Refactoring
- Debrief

2. Project Organization

A. Version Control

- Stayed near each other
- Frequent pushes and merge to github
- Verbally communicated changes
- Almost no merge conflicts
- No surprise files changes

B. Issue Tracking

- console logging
- checking the database
- asking teammates for fresh eyes

C. Code structure

- Reused components as much as possible

- Putting all parameters inside of resolve in app.js
- confined to own folder
- separation of concerns

3. Application Design

A. User Experience

- easy navigation and access of data
- lot of error handling

B. Components and Composition

- separation of concerns
- focused on functionality first and then went for styling

C. Backend API Interpretation

- Michael: I wrote it and error handled extensive
- Others: Asked me or used swagger