Team 301

Kevin Altschuler
Emily Boyle
Sabrina Kantor
Max Rais
Michael Rascati

The Project

Mass.gov

Client

- Mass State Retirement Board (MSRB)

Topic

- Retirement pension estimator for state employees

Users

- Current employees, past employees, prospective employees

The Problem

- Current system is clunky and confusing
- No easy way to predict pension
- Too much recall demanded of users in original overwhelming workflow
    - Directions on different page
    - Two calculators
- Poorly documented calculator code

Our Solution / User Benefits

- Delighted, not frustrated
- Step-By-Step flow
- Display results graphically
- Modify fields without having to start over
- Login using SSN or fill out all input fields

Live Demo!

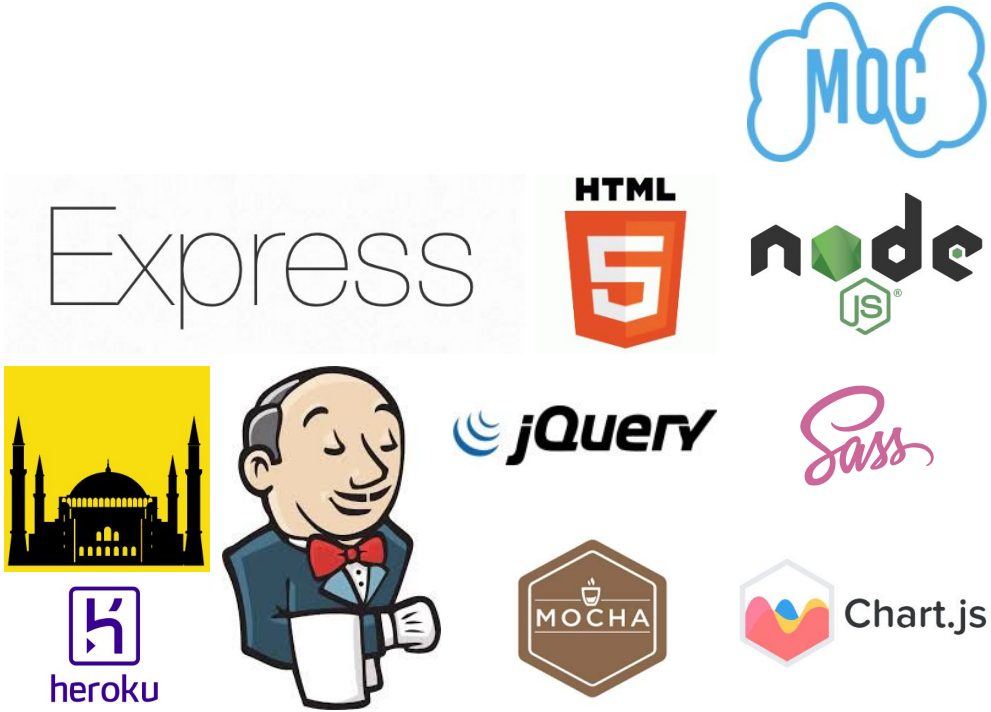
System Architecture

Frontend

Backend

Testing

Integration



Code Quality

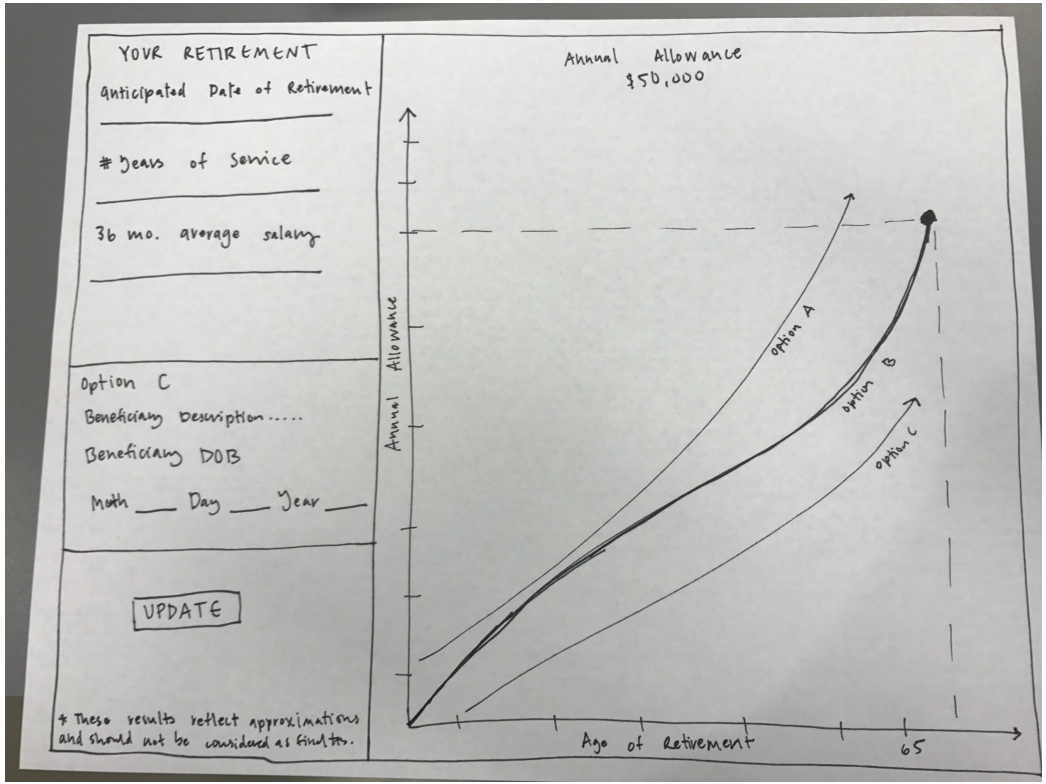
100% backend code coverage and 98% branch coverage

Continuous integration with testing

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
All files	100	98.06	100	100	
301	100	100	100	100	
app.js	100	100	100	100	
301/public/js	100	98.02	100	100	
importablecalc.js	100	100	100	100	
user-service.js	100	75	100	100	9,22

User Testing

- Paper prototype
- Input form flow
- Final calculator testing



Team Process

- Team communication on Slack, and Facebook Messenger
- Sprints in JIRA, we would assign tickets before a Sprint based on capabilities and available capacity in Sprint planning meetings
- Documentation posted on Confluence and linked in Jira Ticket

Team Process

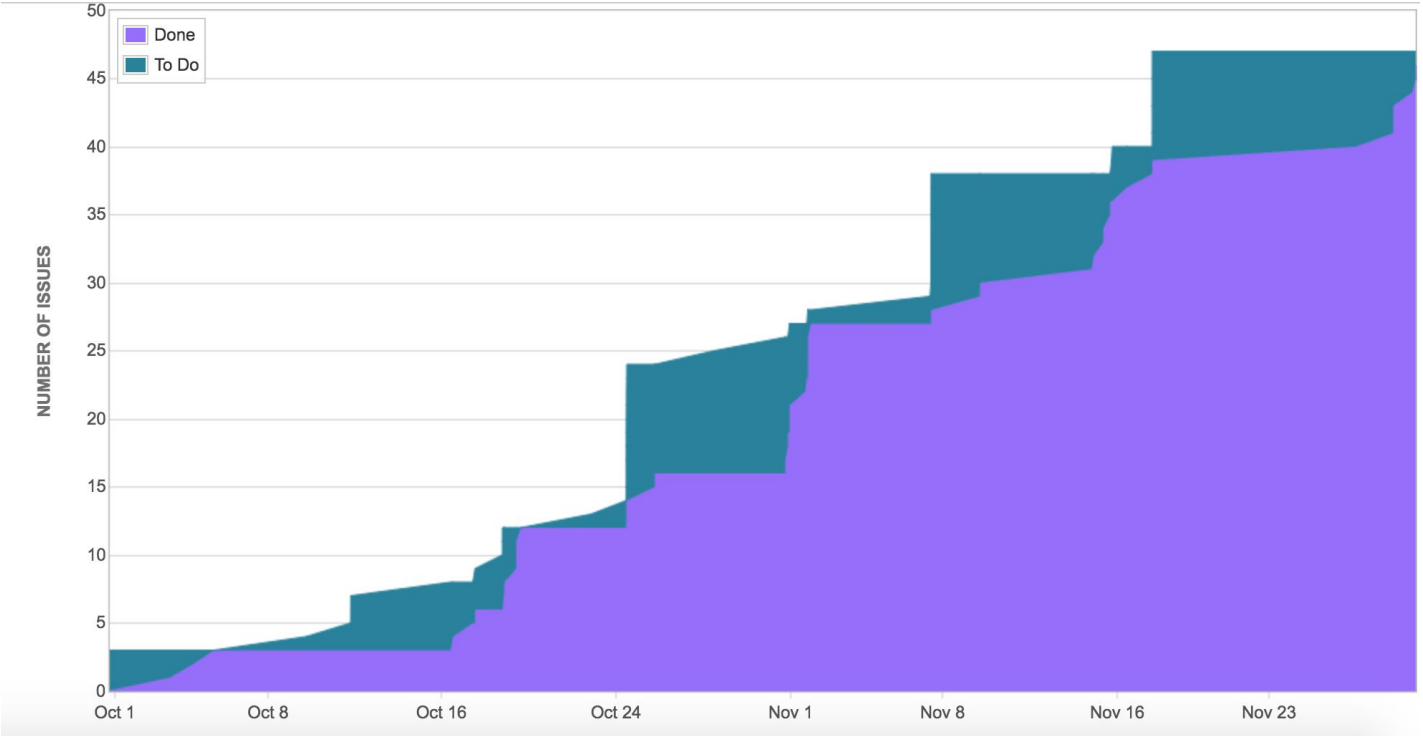
Github contributions chart (y-axis is number of commits)

Team Process

Jira Cumulative

Flow Diagram

(Not made with Chart.js)



Sprint-by-Sprint Progress

Sprint 1 - Define project requirements, description, use cases, and UI mockups

Sprint 2 - Determine tech stack based on requirements, build project template

Sprint 3 - Start with frontend, Create backend infrastructure

Sprint 4 - Calculator functionality and backend

Sprint 5 - Tightening frontend and tying all loose ends

Challenges

- System architecture
  - How to structure the calculator (wizard vs single-page design)
  - Client meeting times
- Integration/Testing
  - We have continuous integration with Jenkins for testing and Heroku for building and deploying
- Code coverage for all cases
- Tech stack
  - System is relatively simple, doesn’t require a full framework

Future/Handoff

Future:

- Funnel Analytics
- Group 3 calculations
- Could always do more cleanup

Handoff:

- Client can embed our front-end in an `iframe`
- Cleaned up and commented calculator code
- Ability for users to visualize pension

Takeaways

- Ops and CI
- Valuable, practical XD experience
- Being your own Scrum Master
- Working with a 3rd party client

Thank you

Team 301