inspector.restaurant

Final Presentation
Web Science Systems Development
ASKDJ
Jonathan Caicedo, Diana Edwards, Alex Schwartzberg, Stephanie Tan

Agenda

- Overview
  - Team Intro, Problem, Solution/Scope
- Plan
  - Technology, Roadmap
- Status
  - Midterm, Final Progress
- Process
  - Challenges, Documentation
- Demo
  - Next Steps, Questions

Team Introductions

Jonathan Caicedo
Fullstack Developer

Diana Edwards
Fullstack Developer

Alexander Schwartzberg
Lead Developer & Technical Manager

Stephanie Tan
Frontend Developer

Problem Description

- People are becoming more health conscious
- What environment is your food being prepared in?
- How much should you trust restaurants?
- Consumers do not have convenient access to health inspection data.

Solution and Scope

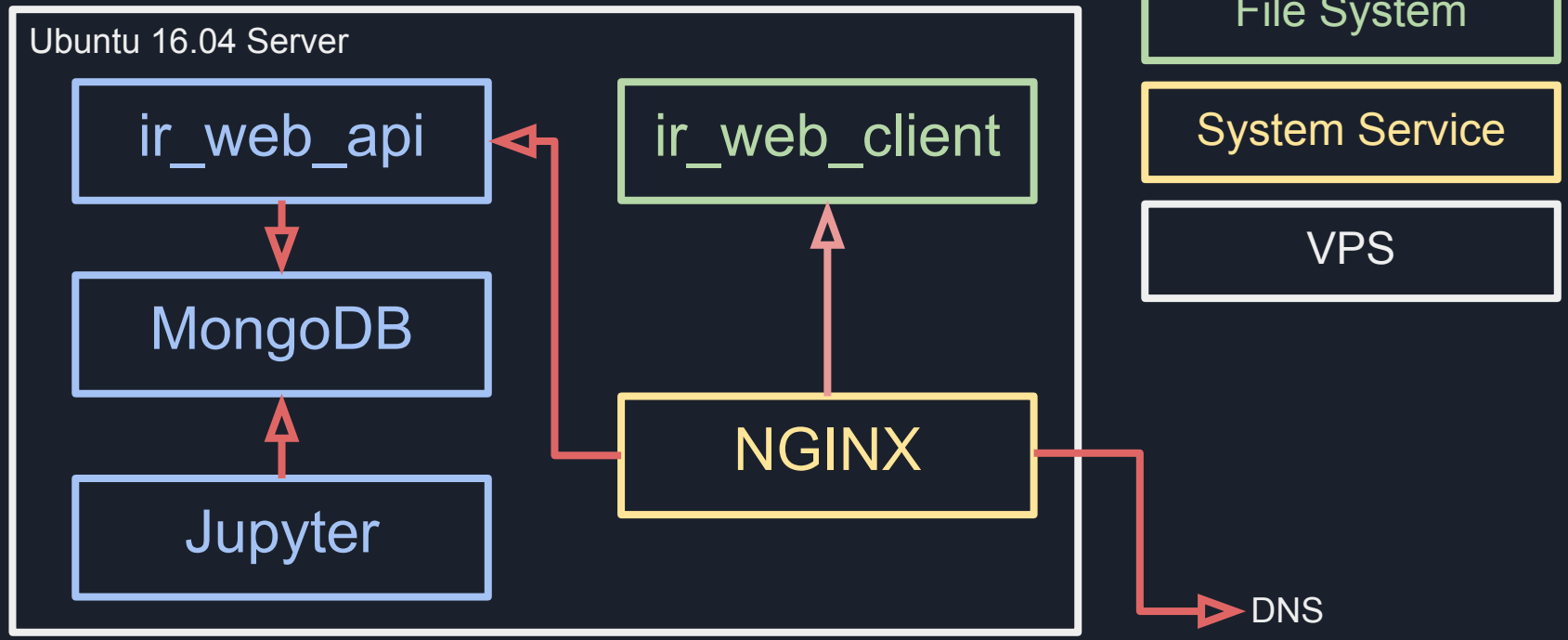
- Develop an intuitive, mobile-friendly application for viewing an establishment's health inspection records.
- Provide access to health inspection records in a web application.
- Results can be filtered by several categories to promote ease of access.

Methodology

Extreme Programming

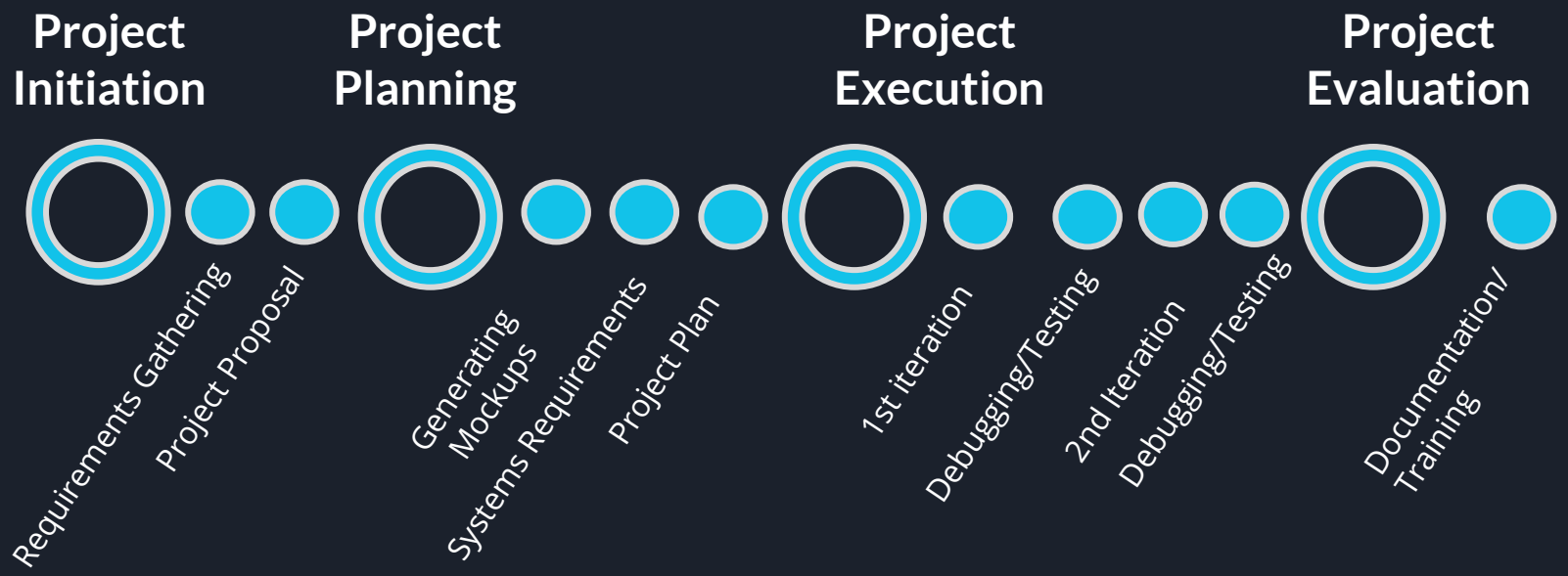
- Weekly planned meetings
- Frequently iterating and pushing code
- Automated testing - unit testing
- Quick release turnaround time

Architecture



Technology Stack

Project Roadmap



MidTerm Progress

- Database and Github accounts have been setup
- Functional mockups completed
- Server Interface completed
- Deployed to production server

End of Semester Progress

- Remains deployed on production server
- Added additional pages including “About Us” and “Terms of Use”
- Improved UX design
- Debugging of dataset Issues
- Created additional documentation
- Improved data sanitization
- Scaled server architecture to handle more requests

Challenges - Invalid data

Health Inspectors aren't winning any spelling bees.

We manually handle these edge cases in our data ingestion pipeline

We documented the issues and contacted the dataset owner to request corrections.

(awaiting reply)

Actual data from NY State

Challenges - Ransomware!

One day our database was emptied except for this document:


```
{
  "_id": "ObjectID("5adfacb03af0ca008bd3e8af"),
  "BitCoin": "1Fx9Za5bx3ejt664B3kLTsHyYhKRiNSNtd",
  "eMail": "mongodb@tfwno.gf",
  "Solution": "Your Database is downloaded and backed up on our secured servers. To recover your lost data: Send 0.2 BTC to our BitCoin Address and Contact us by eMail with your server IP Address and a Proof of Payment. Any eMail without your server IP Address and a Proof of Payment together will be ignored. You are welcome!"
}
```


Always secure your databases with authorization credentials!

Documentation

We used APIDoc to generate our API documentation

```
/**
* @api {get} /api/restaurants/:id Show
* @apiName show
* @apiGroup Restaurant
* @apiDescription Gets a Restaurant and all associated inspections
* @apiPermission public
* @apiSuccess {Model} root Restaurant with inspections records
* @apiError (500) UnknownException Could not retrieve Restaurant model
*/
module.exports.show = (req, res, next) => {
    return Restaurant.findOne({ _id: req.params.id })
    .then((response) => {
```

Documentation available at

inspector.restaurant/docs

Demo

http://inspector.restaurant

Next Steps

- Continue to fix and bugs that arise
- Tag restaurants by the type of food they offer
- Browser extension that integrates with GrubHub / Yelp

Thank you!
Questions?