

Final Progress Presentation

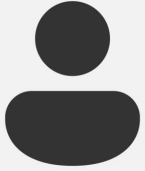
Ahmad Kassem, Alexander Keehan, Harold Velasquez, Nic, Daniel
Montoya



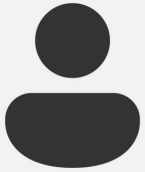
The Team



Alex Keehan
Backend Developer
Co-Team Lead



Nic
Backend Developer
Co-Team Lead



Harold Velasquez
Frontend Developer
Frontend Lead



Ahmad Kassem
Backend Developer



Daniel Montoya
Frontend Developer



The Client

WAGEL



Yoni Elmalem
Wagelbox CEO



Adam Wagman
Wagelbox CTO



Introduction

Project Motivation

- Small businesses often lack the time, expertise, and staff for professional-level email marketing.
- Wagelbox aims to automate campaign creation with Generative AI.

Capstone Goal

- Deliver production ready **mobile responsiveness** improvements and **Gen AI-powered prototypes** in 14 weeks.
- Provide small businesses with:
 - Easier & faster campaign creation
 - AI-driven personalization and creative tools
 - A platform to stand out without extra staffing

Project Plan & Scope

Front-End

- Delivered major **mobile-responsive redesign** across all primary pages and email layouts.
- Completed client requested UI bug fixes and visual polish.
- Managed delivery from a **client backlog** via weekly Agile sprints.

Back-End / AI Features

1. **Generative AI Email Builder**

- Produces email copy, HTML, and CSS templates.

2. **Image Recognition Classifier**

- Tags uploaded images to improve automated image recommendations.

3. **AI Image Generation Service**

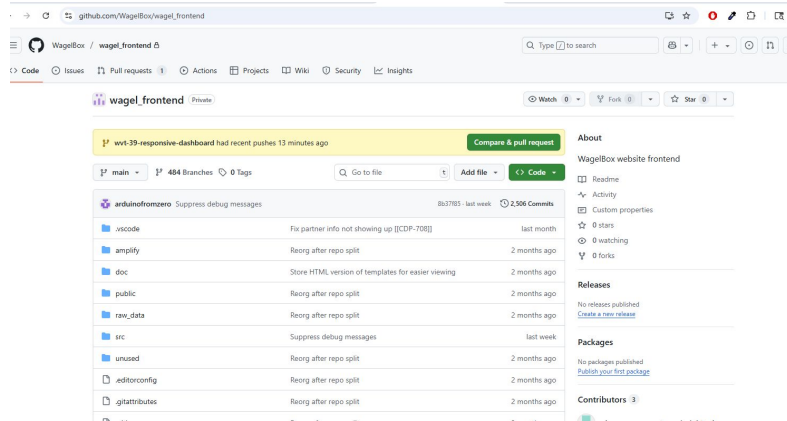
- Generates supplementary campaign visuals when client images are unavailable.

Demo



Front-end - Infrastructure onboard

- The client has had no help in their web application
- The Front-end team has helped the client in their development env
 - Bring the client on the cloud for development using EC2 instances (M6)
 - Git development workflow - Pull Requests, Branching, Git using Agile
 - Iron out packages and bugs with current setup of the app in the environment (Ubuntu)
 - Assist in dev desktop on/off to mitigate cost (python scripts)



Work Completed (Front-end)

- Documentation - assessments
- Software testing (manual execution)

WAGELBOX

Website Assessment for Mobile

Authors: Daniel Montoyas, Harold Velasquez

Last update: 7/12/2025 - 8:17PM

WVI: 14, 75

Contents

Summary	3
1. Landing	4
2. Sign-in	9
3. Sign-up	10
4. Onboarding sequence	11
5. Tasks	12
6. Dashboard	13
7. Partnerships	14
8. Campaign planner	15
9. Campaign results	16
10. Campaign results	17
11. Images	18
12. Template Gallery	19
13. Profile	20
14. Setting	21
15. Tutorials	22
16. Feedback	23
17. Take the tour	24

Wagelbox_Responsiveness_Assessment_v2.pptx
document - 14/8/25

Desktop view - Dashboard

Slide — Desktop view - Dashboard

Whitespace heavy: 65-70 % of viewport unused on ≥ 1440 px screens

Action cards (tasks, holidays) fixed-width – break to two lines on 768-1024 px

Navigation column uses absolute positioning \rightarrow potential overlap when RTL/localization

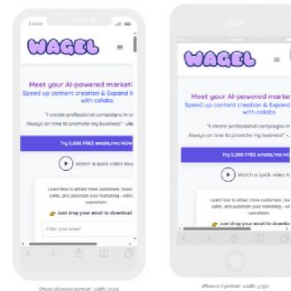
Metrics & observations

- CLS spikes when bell/clock icons load (see root-cause slide)
- LCP element sits dead-centre; shifts when viewport shrinks below 1280 px

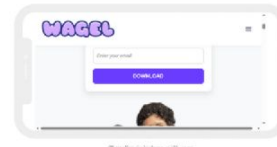
Responsive refactor ideas

- Convert main grid to display:grid with auto-fit minmax(320px, 1fr) columns
- Collapse “Create Campaigns” tabs into a horizontal scroll (a-la Gmail) for ≤ 1024 px
- Introduce **content-first ordering** for assistive tech: nav \rightarrow main \rightarrow sidebar

1. Landing



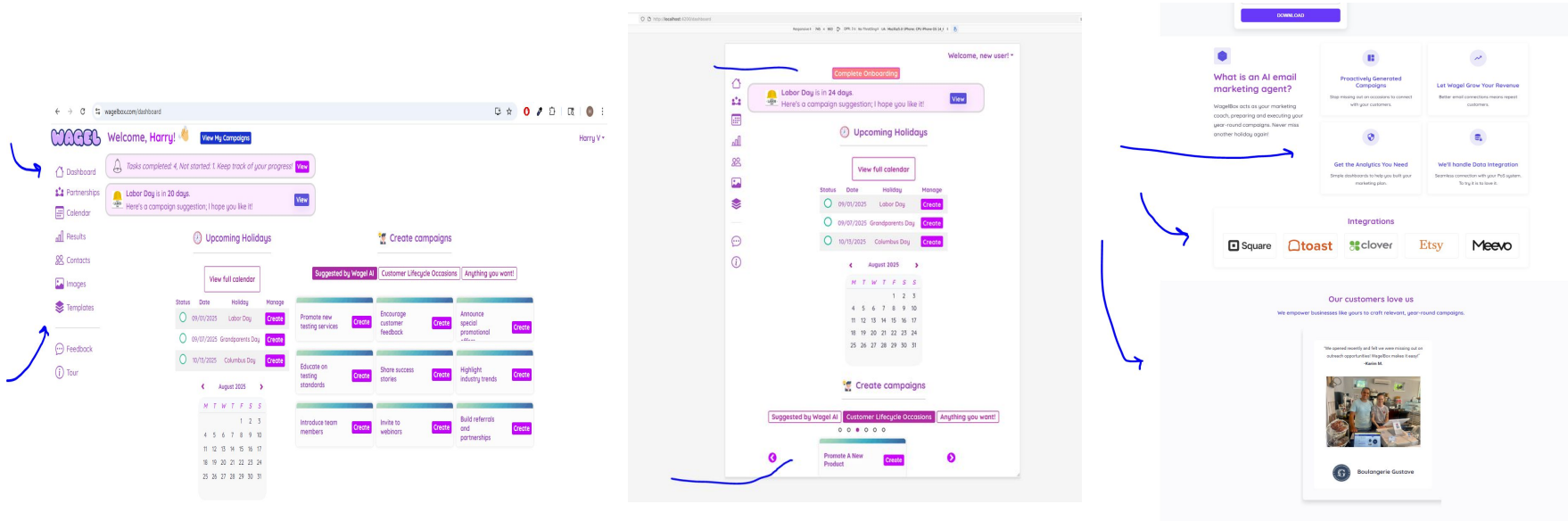
1. Make text fit without needing side scroll.
2. Email subscription should fit and not get cut off.



3. Remove side scroll make pictures smaller and form larger.

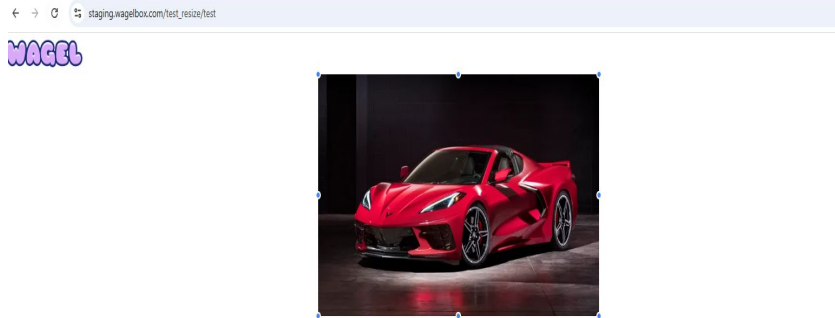
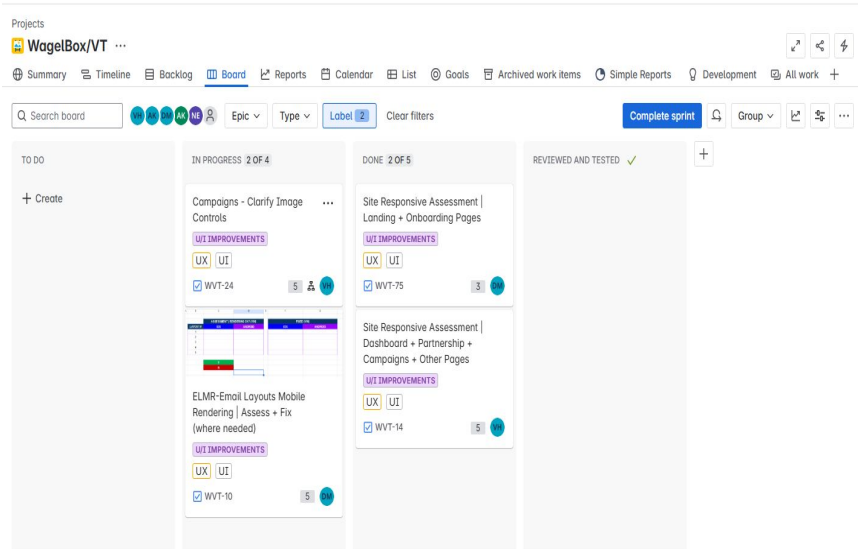
Work Completed (Front-end)

- Fixing site bugs (buttons, spacing, consistent UI)
- Made some components and pages responsive



Work Completed (Front-end)

- Prototyping features so that the client can implement ideas
- Assessing the client's current mobile issues in depth
- Deployed to a staging environment for the website

Projects
WagelBox/VT ...

Summary Timeline Backlog Board Reports Calendar List Goals Archived work items Simple Reports Development All work +

Search board [WVT] [LUX] [UI] [WVT] Epic Type Label 2 Clear filters Complete sprint Group [] [] [] []

TO DO	IN PROGRESS 2 OF 4	DONE 2 OF 5	REVIEWED AND TESTED ✓
+ Create	<p>Campaigns - Clarify Image Controls</p> <p>UIT IMPROVEMENTS</p> <p>LUX UI</p> <p>WVT-24 [5] [] []</p> <p>ELMR-Email Layouts Mobile Rendering Assess + Fix (where needed)</p> <p>UIT IMPROVEMENTS</p> <p>LUX UI</p> <p>WVT-10 [5] [] []</p>	<p>Site Responsive Assessment Landing + Onboarding Pages</p> <p>UIT IMPROVEMENTS</p> <p>LUX UI</p> <p>WVT-75 [5] [] []</p> <p>Site Responsive Assessment Dashboard + Partnership + Campaigns + Other Pages</p> <p>UIT IMPROVEMENTS</p> <p>LUX UI</p> <p>WVT-14 [5] [] []</p>	

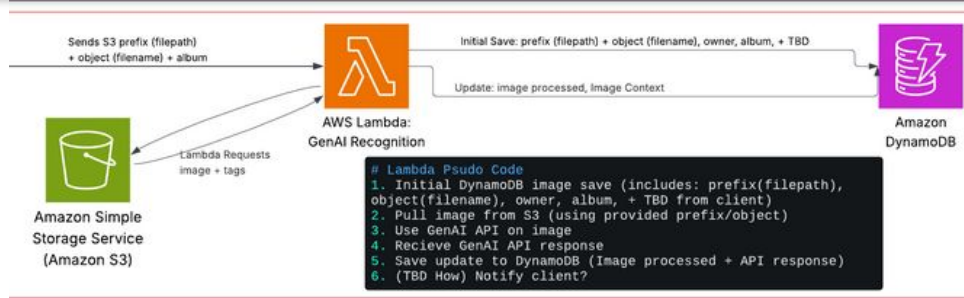
Work Completed (Image Recognition)

- Two main parts to Image Recognition
 - Tagging & Recommending
- 4 phases
 - Model research
 - Chose ChatGPT-4o
 - Tagging
 - Describe image
 - Image recommender
 - Sort images
 - Documentation

Image Recognition

Model Name/Link	Strengths	Weaknesses	API Notes	Accuracy	Cost	Notes
Amazon Rekognition	Can build custom labels for detection and detect text	Needs to be trained and can't leave running all the time	Already an AWS service, so integration should be straightforward	Not able to test without training a model and accessing AWS services	\$1 per hour for training \$4 per hour for inference. Billed on minute increments	Can set aside limited time each day to process all images to keep costs low
ChatGPT-4o	Very good output. Quick results. Already trained	Benefits from precise prompts	Integrates easily with Python backend, so should be no issue. Lots of user support	100% Very good tagging performance & descriptions	Per 1k tokens \$0.005 input \$0.015 output \$0.0075 per image	Can use VisionIdentify GPT also for more complex operations. E.g. image segmentation or chaining inputs
Clarifai	Good output. Exportable to JSON	No text recognition	Python API, so integration should be straightforward	100% no issues identified	\$0.0012 per request	Pre-trained or can have trainable models
Gemini 2.0 Flash-Lite	Good results	Slightly slower than other models. Less identified keywords than other models	Lots of user support & Python support. Should be no issues	85% performance was good, but not amazing	\$0.075 input \$0.30 output	Good performance, but results were slightly worse than other models and costs are higher
ChatGPT-4o-mini	Cheaper than GPT-4o. Still good performance	Worse than GPT-4o in performance	Python support & user support. Should be no issues	80% it's good, but not as good as GPT-4o	Per 1k tokens \$0.0003 input \$0.0004 output \$0.001 per image	Slightly worse than GPT-4o, but cheaper and still good for our purposes

Image Recognition Architecture Diagram



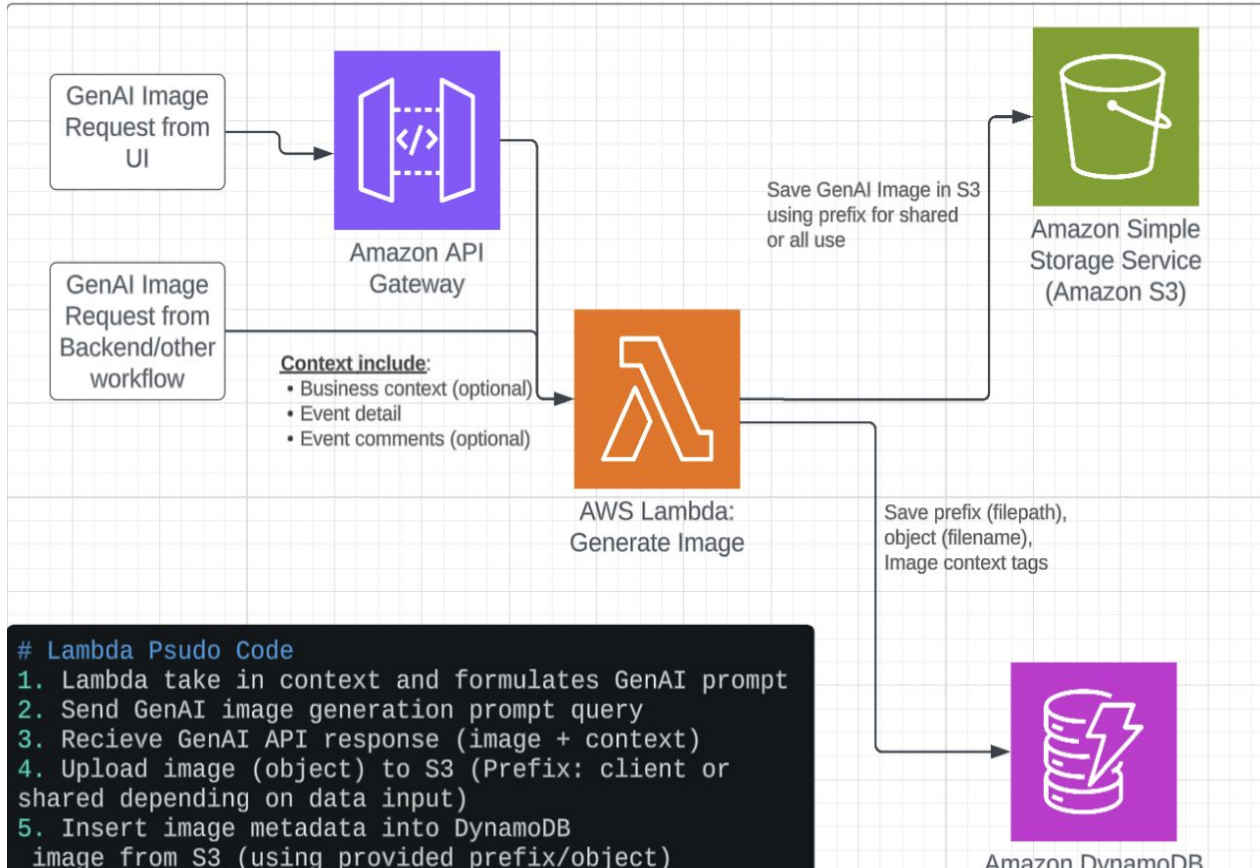
- Value Proposition of This Approach**
- 1. Completely serverless
 - 2. Only pay for what you need
 - 3. Highly scalable
 - 4. Highly available
 - 5. Interoperable (easily swap AI tool and move on)

1. User uploads image(s)
2. Image gets pulled into Lambda
3. Call the model with crafted prompt
4. Get tagging results back
5. Store tags & other relevant info into S3 & DynamoDB
6. Image recommender functionality in same Lambda
 - a. Will pull data from S3
 - b. Returns sorted list of relevant images for given campaign/holiday

All steps completed

***All integrated into client's AWS services

Image Generation Architecture



Lambda Pseudo Code

1. Lambda take in context and formulates GenAI prompt
2. Send GenAI image generation prompt query
3. Recieve GenAI API response (image + context)
4. Upload image (object) to S3 (Prefix: client or shared depending on data input)
5. Insert image metadata into DynamoDB
image from S3 (using provided prefix/object)

Image Generation progress

Component Progress:

1. Image generation research (model, what fits best, etc.)
2. Generation prototype (Able to generate images)
3. Integration into dynamodb (metadata)
4. Integration into S3 bucket (image url)
5. Aws lambda integration (lambda function)
6. Aws gateway integration (connecting to api)
7. Prompt refinement and testing

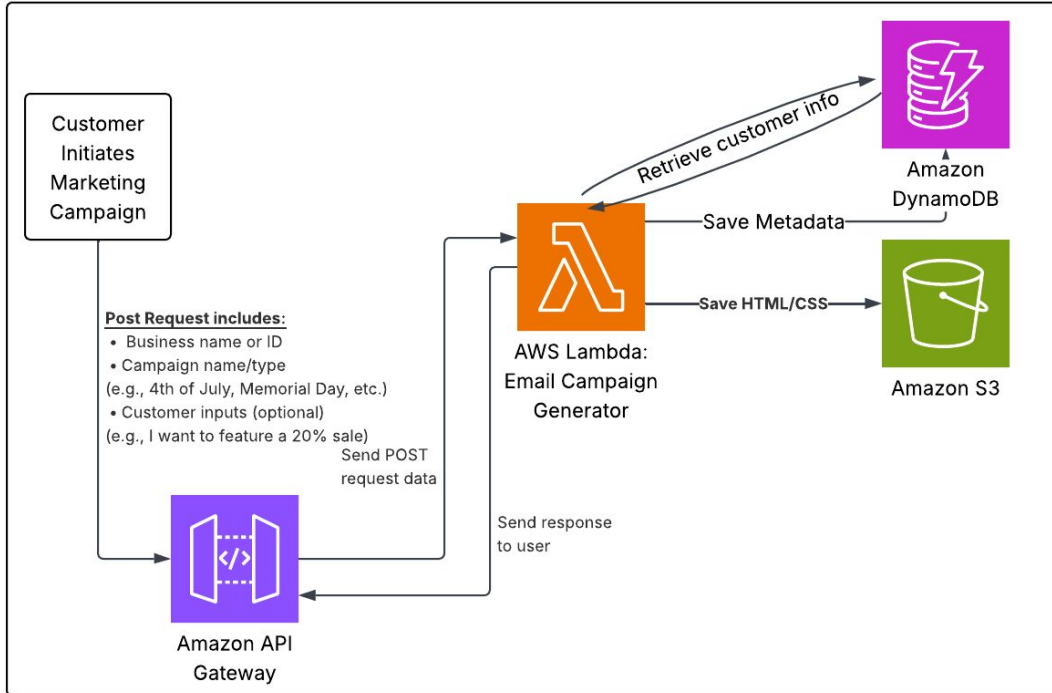
Done

In progress

Incomplete

Email Campaign Architecture

Email Campaign Generator



Component Progress:

1. Internal Component architecture
2. GenAI research, latency exploration, business cost estimates
3. Email Campaign Generator Prototyping
4. AWS DDB & S3 Integration
5. AWS Lambda Integration
6. AWS API Gateway Integration
7. External Component Definition Integration
8. Environment Transition & Integration

DONE
IN-PROGRESS
PLANNED

Challenges/Lessons Learned

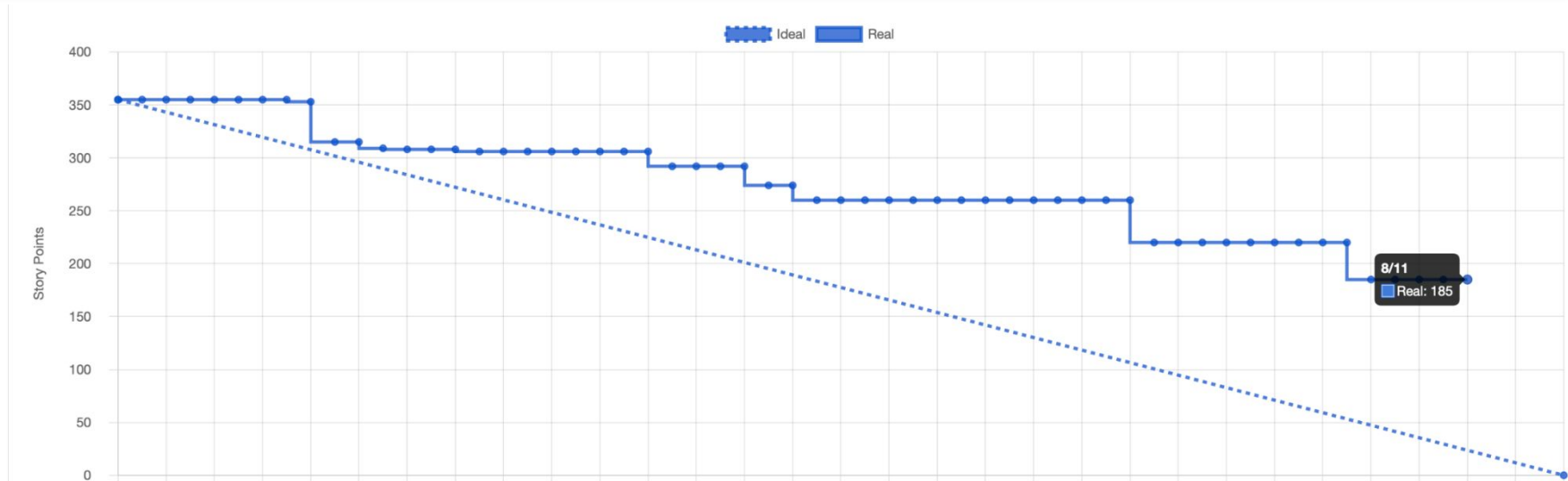
Program Management (non-technical):

- Scope creep – It's important to stick to assigned stories (no deviation in either direction)
- Non-technical work consumes time! – Documentation, client meetings, and agile backlog grooming
- Conflicting Schedules – all students work full-time, makes meeting challenging but we combat it with ample communication channels and dividing work strategically to limit inter team dependencies
- Communication is essential for continued progress in this highly agile environment

Technical

- Adapting to existing codebases can be more challenging than starting from scratch
- Needed an introduction of the codebase earlier on & explained architecture as well as the where things lived and how to work better with bootstrap styling
- AWS services: Lambda, API Gateway, DynamoDB, S3, IAM

Jira Project Backlog



- Team remained agile throughout course semester
 - Backlog exceed team capacity since Day 1
 - 31 points pending “Done” after client review
- Backlog is READY for the next Capstone team
 - Over 100 story points added to project backlog since mid-term presentation.

Questions?

