Making Recipes Readable Again

Team 12

Client: Mat Wymore	mlwymore@iastate.edu
Advisor: Mat Wymore	mlwymore@iastate.edu
Team Leader: John Paton	jpaton@iastate.edu
Meeting Scribe: Bret Knous	bjknous@iastate.edu
Design Facilitator: Vismay Gehlot	vgehlot@iastate.edu
Test Facilitator: Luke Knous	ljknous@iastate.edu
Report Facilitator: Rithwik Gokhale	rgokhale@iastate.edu

Problem Statement

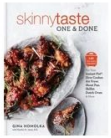
- Online recipe blogs are confusing.
- Difficult to navigate.
- Aim to reformat and modify.
- Web scraper to simplify.
- Hope this makes cooking more enjoyable.
- Hope to draw more people to cooking.

Example Website - Recipe is nowhere to be found

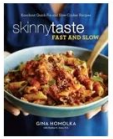
AIR FRYER CHICKEN TENDERS

I made these tasty chicken strips the other night for dinner along with some orzo with zucchini and tomatoes, and my family loved them! They are super easy and quick for weeknight meals. Air fryer recipes are one of my top requests, so I promise to share more soon!

ORDER NOW ON:
Amazon | Barnes & Noble | Indiebound | Target



ORDER NOW ON:
Amazon / Barnes & Noble / Indiebound / Target



ORDER NOW ON:
Amazon / Barnes & Noble / Indiebound / Target

ORDER NOW ON:
Amazon / Barnes & Noble / Indiebound / iBooks

ORDER NOW ON:
Barnes & Noble / Amazon

VIEW MEAL PLANS »

Functional Requirements

- Capable of loading a recipe from an arbitrary URL
- Make recipe more readable by cutting off extraneous information
- Changing instructions to include measurements / amounts
- Ability to scale recipes up or down proportionally
- Converting units between metric and imperial

Non-functional Requirements

- Must work on a typical mobile device/smartphone
- May be platform-specific (e.g. Android), though cross-platform is ideal
- Interface must be user-friendly (typical user is expected to be viewing device from roughly one -meter away and using one non-primary finger to navigate)
- Fast - "Upgraded" recipes should load in 5s or less
- Solution should not interfere with the source's revenue stream (e.g. ads should still display)
- Should not require a user account (may be optional if it would help with desired features)
- Should not require backend infrastructure

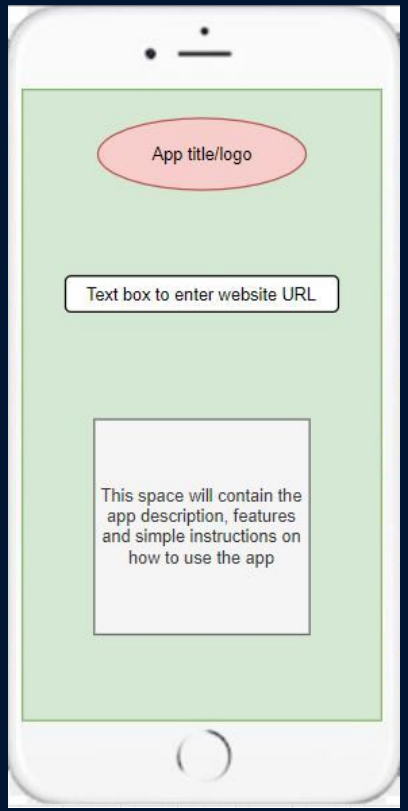
Technical/Other Constraints/Considerations

- Assumptions:
  - The end product can be used globally with no restriction
  - The make and model of a preferred device will not affect app compatibility
  - There will be no limitations with access to the required software and APIs during development
- Constraints:
  - This app will require constant internet connection since the websites are ‘enhanced in real time’
  - Specific features in the app will also require constant GPS data to provide shopping information for the specialized ingredients (optional feature)
  - A new instance of the app will be loaded at each use due to the lack of external database (one of the functional requirements put forward by the client)
  - The end product will only work on mobile devices. Thus a web version of the product will not be available.
  - The app will only work on english websites

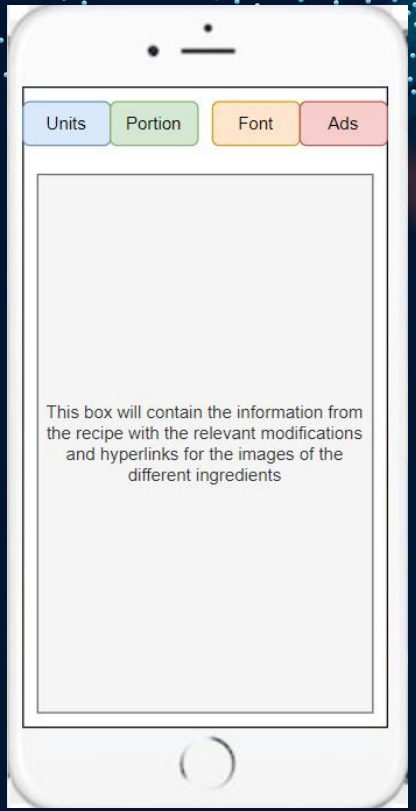
Project Deliverables

- The final product which will be delivered to the client is a cross platform mobile application which meets all the features and functional requirements which have been listed in the above sections
- Design and product mockups will be delivered to the client on a bi-weekly basis once the development has begun
- While the application will be designed for maximum ease of maneuverability by the end user, an onboarding/user guide can be created and delivered with the end product if required by the client. This project is yet to be finalised after further discussions with the client.

Conceptual Sketch



- Screen mockups developed for URL screen and recipe screen
- Employing an easy to understand/navigate design
- Designed to be used in the Kitchen while cooking
- Using neutral colors to ensure that users are distracted from the text



Market Survey - What Makes Us Unique

- Looked at existing apps in the market which provide similar services:
  - Paprika - only provides users with the option to save/bookmark recipes
  - BBC Good Food - curated list of recipes created/reviewed by professionals
  - Tasty - crowdsourced approach to recipes where individual users can suggest changes/tweaks
- Existing products only provide users with the option to save/review recipes but do not have any of the enhancements our app will be providing.
- Our proposed app is more likely to improve the overall culinary experience.

Potential Risks & Mitigation

- Flaws In Testing - It may be impossible to have test cases that work for all recipe websites
- Unusable UI - If our UI isn't done correctly issues could arise for users with imperfect eyesight
- Cyber Security - When users enter a URL, it could lead them to unsafe websites

- Targeted Testing – Focus test cases around major recipe websites to ensure usability in the majority of cases
- Design UI with the visually impaired in mind
- Run user input through URL checking tool

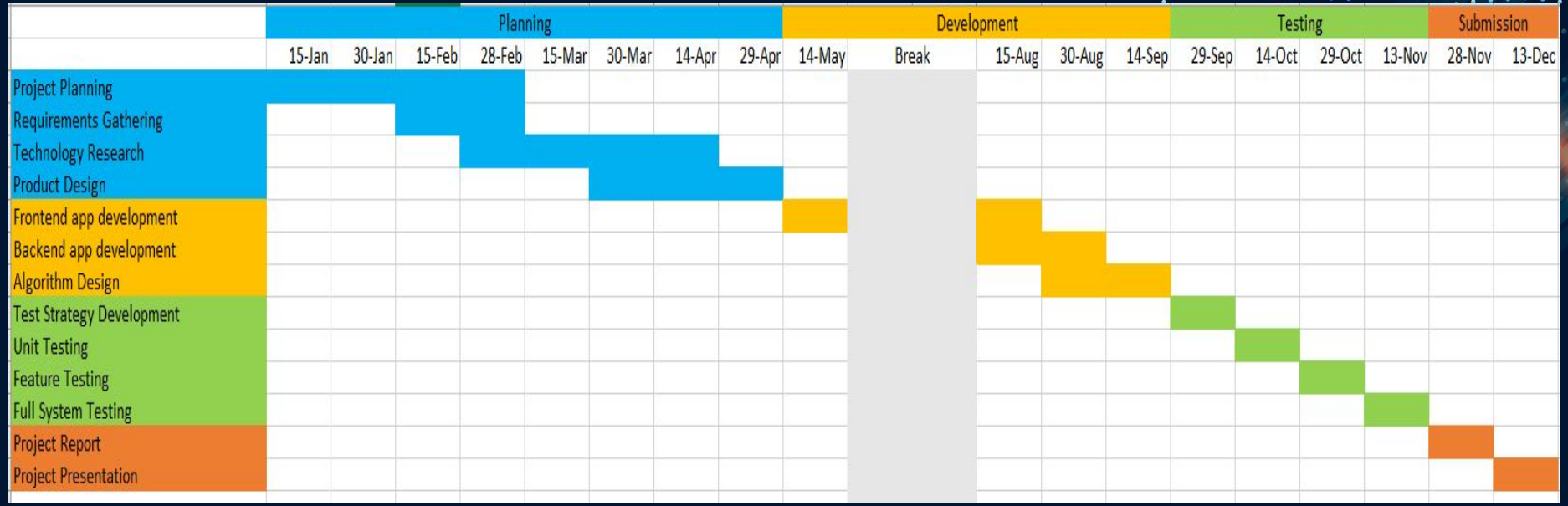
Resource/Cost Estimate

- Since the product is a mobile application, there are minimal external resource requirements and costs associated with this project.
- Hardware resources required for project completion are 1 Android & iOS smartphones
  - The team will be using their personal devices for final testing
- No known software costs are associated with the project at this moment

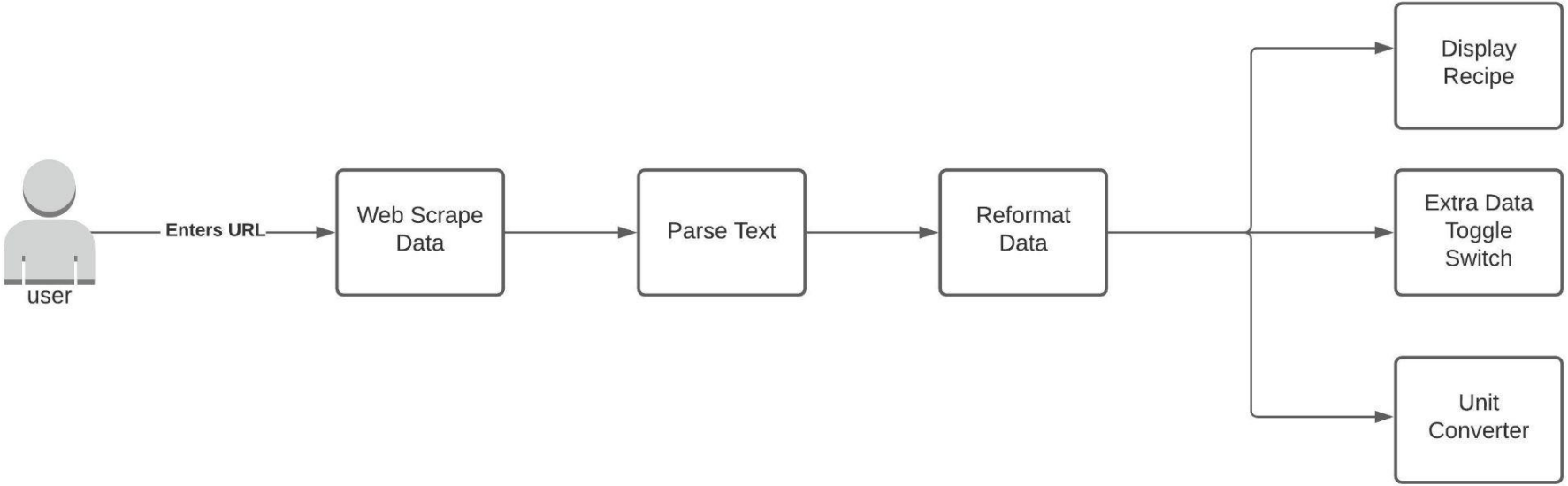
Project Milestones

- Milestone 1 - Unit Tests for Main Functionality Created
- Milestone 2 - Main Functionality Completed
- Milestone 3 - Application Functions Properly On Major Sites
- Milestone 4 - Full Functionality

Schedule

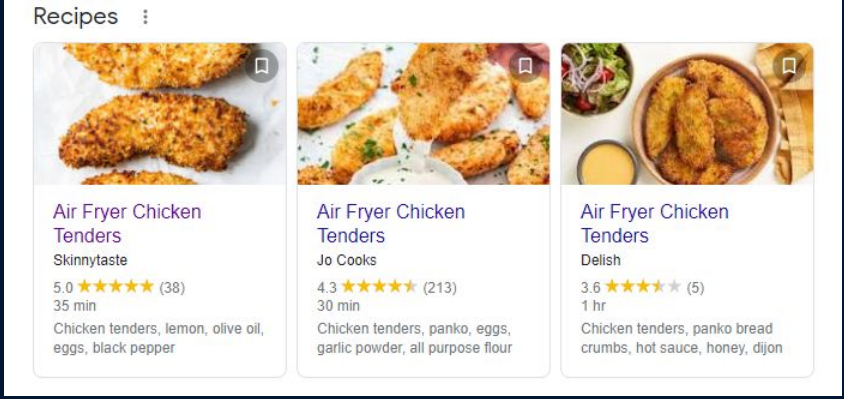

Planning
Development
Testing
Submission
15-Jan
30-Jan
15-Feb
28-Feb
15-Mar
30-Mar
14-Apr
29-Apr
14-May
Break
15-Aug
30-Aug
14-Sep
29-Sep
14-Oct
29-Oct
13-Nov
28-Nov
13-Dec
Project Planning
Requirements Gathering
Technology Research
Product Design
Frontend app development
Backend app development
Algorithm Design
Test Strategy Development
Unit Testing
Feature Testing
Full System Testing
Project Report
Project Presentation

Functional Decomposition



Design Details

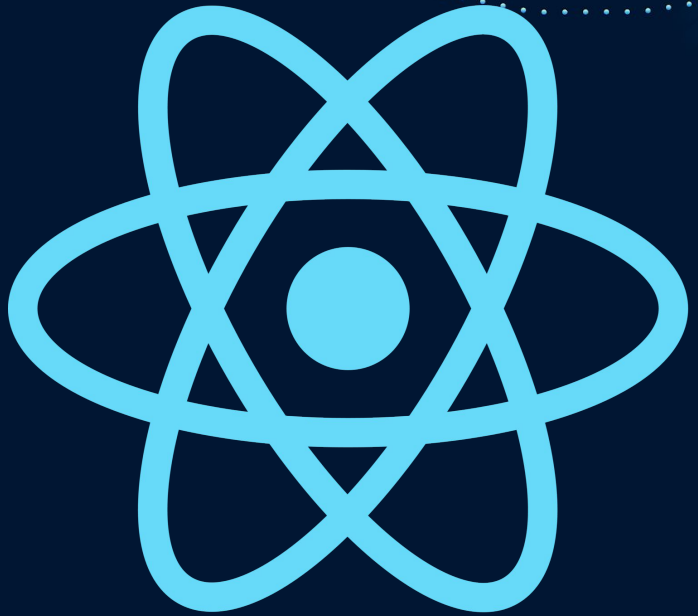
- We plan on making use of Request and Json-LD to scrape and parse recipe metadata




```
<script type="application/ld+json">
  {
    "@context": "http://schema.org",
    "@type": "Recipe",
    "name": "Classic Marinara Sauce",
    "recipeIngredient": [
      "1 28-ounce can whole tomatoes",
      "1/4 cup olive oil",
      "7 garlic peeled and slivered",
      "Small dried whole chile",
      "1 teaspoon kosher salt",
      "1 large fresh basil sprig"
    ]
    // more fields
  }
</script>
```

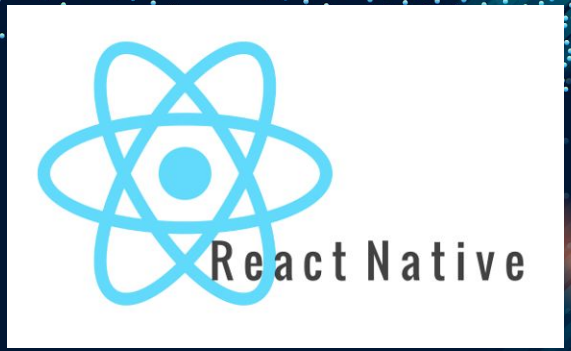
App Framework - React Native

- Developed by Facebook
- Strong Support
- Allows for streamlined dual development of IOS and Android Applications

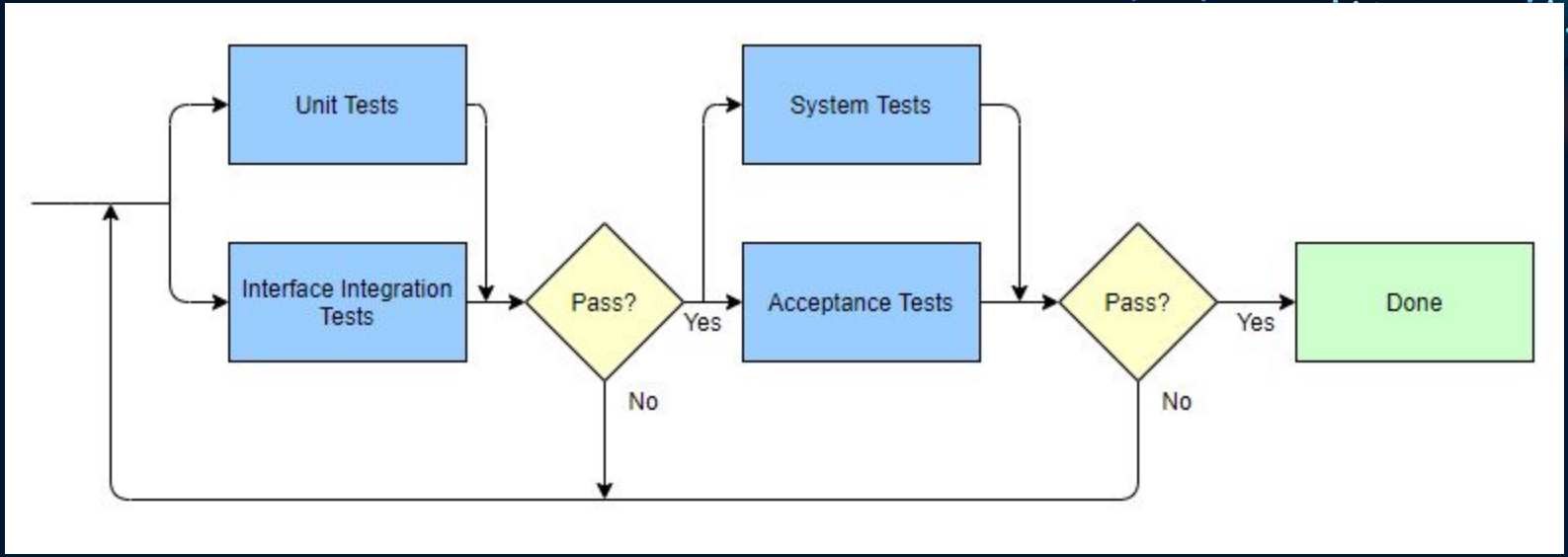


How are we testing?

- Unit Testing
  - Jest
- Interface Testing
  - React Native CLI Quickstart
  - iOS Emulator
  - Android Emulator
- Acceptance Testing
  - Presentation to client



Test Plan


Unit Tests
Interface Integration Tests
Pass?
Yes
No
System Tests
Acceptance Tests
Pass?
Yes
No
Done

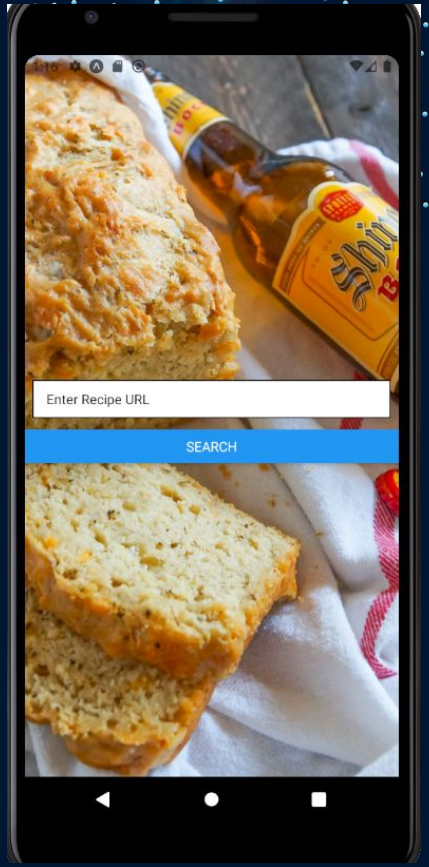
Prototyping

```
test('properly converts mL to imperial', () => {
    expect(convertToImperial(10, 'mL')).toStrictEqual([2, 'tsp'])
    expect(convertToImperial(22.5, 'mL')).toStrictEqual([1.5, 'tbsp'])
    expect(convertToImperial(135, 'mL')).toStrictEqual([4.5, 'fl oz'])
    expect(convertToImperial(240, 'mL')).toStrictEqual([1, 'cup'])
    expect(convertToImperial(357.5, 'mL')).toStrictEqual([(357.5 / 240), 'cups'])
    expect(convertToImperial(712.5, 'mL')).toStrictEqual([1.5, 'pt'])
    expect(convertToImperial(2375, 'mL')).toStrictEqual([2.5, 'qt'])
    expect(convertToImperial(3800, 'mL')).toStrictEqual([1, 'gal'])
})
```

```
Developer PowerShell
Developer PowerShell

 PASS  TestScripts/scale.test.js
 PASS  TestScripts/convertToImperial.test.js
 PASS  TestScripts/convertToMetric.test.js
----------------------|---------|----------|---------|---------|-------------------
File                  | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
----------------------|---------|----------|---------|---------|-------------------
All files             |     100 |      100 |     100 |     100 |
 convertToImperial.js |     100 |      100 |     100 |     100 |
 convertToMetric.js   |     100 |      100 |     100 |     100 |
 scale.js             |     100 |      100 |     100 |     100 |
----------------------|---------|----------|---------|---------|-------------------

Test Suites: 3 passed, 3 total
Tests:       17 passed, 17 total
Snapshots:   0 total
Time:        2.904 s
Ran all test suites.
```



Current project status

Currently approaching Milestone 1

- Planning is complete
- Testing framework is up and running
- Several tests are passing already

Plan for next semester

- Finish unit tests for main functionality first
- Split into separate roles working on the different functions of the applications
- We will split up and begin working on UI functionality, Web Scraping, Data manipulation, and additional test cases

Thank you for listening
Any Further Questions?