Tasteful Panthers Milestone 1

TEAM MEMBERS
ALEX LAUREANO
JIAHAO SHU
ANTHONY HORDESKY
LEXI FRANKLIN

FACULTY ADVISOR
PHILIP CHAN

Task Matrix for Milestone 1

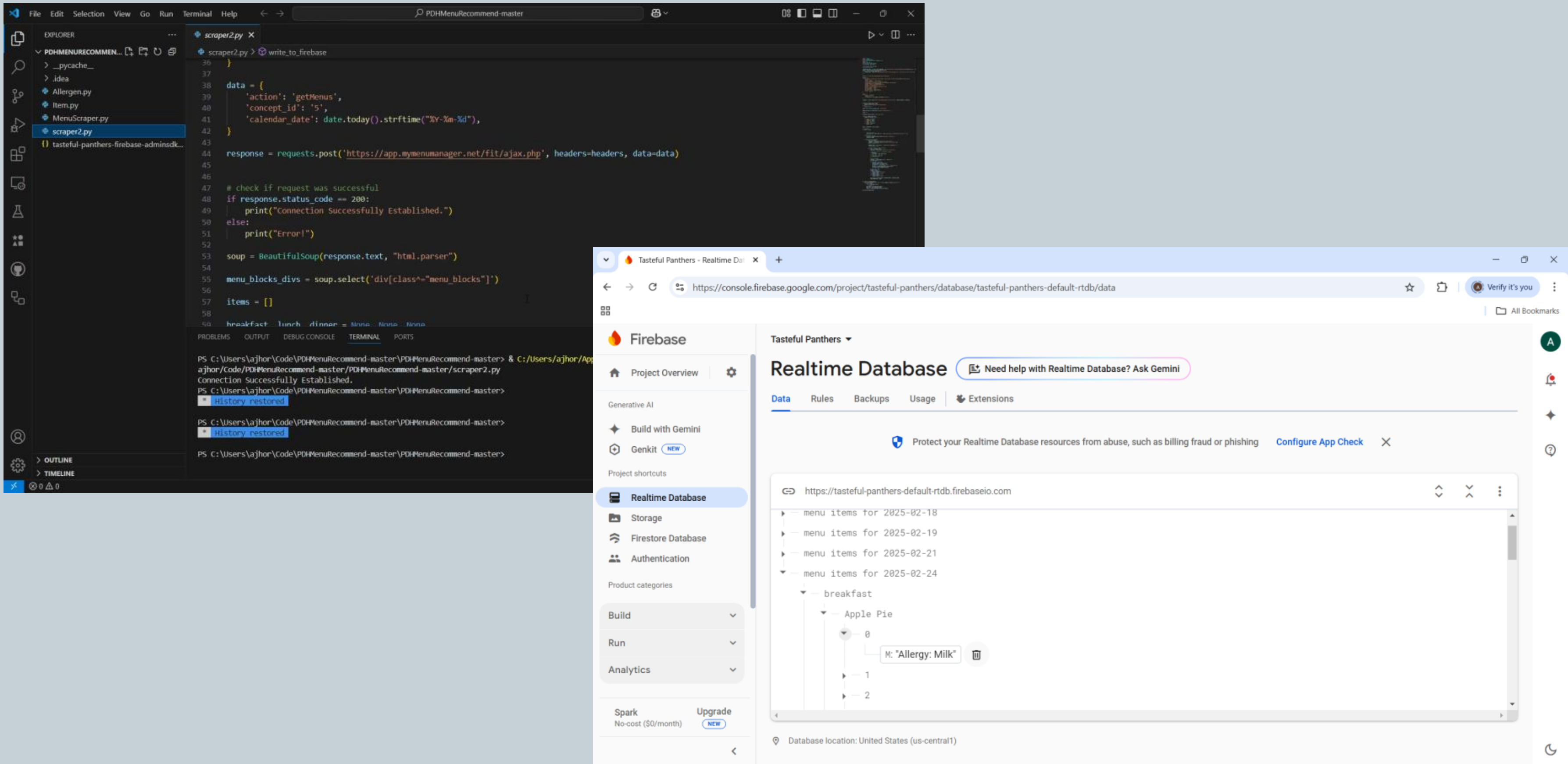
Task	**Completion**	**Alex**	**Jiahao**	**Anthony**	**Lexi**
Compare and Contrast Technical Tools	100%	Web/Information Scraping	IDE	Database	IDE
Tool Demos	100%	Web/Information Scraping	IDE,	Database	IDE
Resolve Technical Challenges	80%	Swift, sorting algorithm	Swift, sorting algorithm	Swift, algorithm, encryption/decryption	Swift, sorting algorithm
Collaboration Tools	100%	Communication	Documents/presentations	Software development	Task calendar
Requirements Document	99%	Write 40%	Write 20%	Write 20%	Write 20%
Design Document	99%	Write 20%	Write 20%	Write 20%	Write 40%
Test Plan	99%	Write 20%	Write 20%	Write 40%	Write 20%

Task 1: Compare and Contrast Technical Tools

- Flutter, programming in Dart for cross platform mobile development (iOS AND Android now)
- Chose Google Firebase as our database due to functionalities like:
  - Cloud capabilities
  - Firebase Realtime Database

Task 2: Demos

- Scraping menu items and storing them in database



```
}

data = {
    'action': 'getMenus',
    'concept_id': '5',
    'calendar_date': date.today().strftime("%Y-%m-%d"),
}

response = requests.post('https://app.mymenumanager.net/fit/ajax.php', headers=headers, data=data)


# check if request was successful
if response.status_code == 200:
    print("Connection Successfully Established.")
else:
    print("Error!")

soup = BeautifulSoup(response.text, "html.parser")

menu_blocks_divs = soup.select('div[class^="menu_blocks"]')

items = []

breakfast, lunch, dinner = None, None, None
```

```
PS C:\Users\ajhor\Code\PDHMenuRecommend-master\PDHMenuRecommend-master> & C:/Users/ajhor/Ap
ajhor/Code/PDHMenuRecommend-master/PDHMenuRecommend-master/scraper2.py
Connection Successfully Established.
PS C:\Users\ajhor\Code\PDHMenuRecommend-master\PDHMenuRecommend-master>
 * History restored

PS C:\Users\ajhor\Code\PDHMenuRecommend-master\PDHMenuRecommend-master>
 * History restored

PS C:\Users\ajhor\Code\PDHMenuRecommend-master\PDHMenuRecommend-master>
```

Realtime Database

Protect your Realtime Database resources from abuse, such as billing fraud or phishing

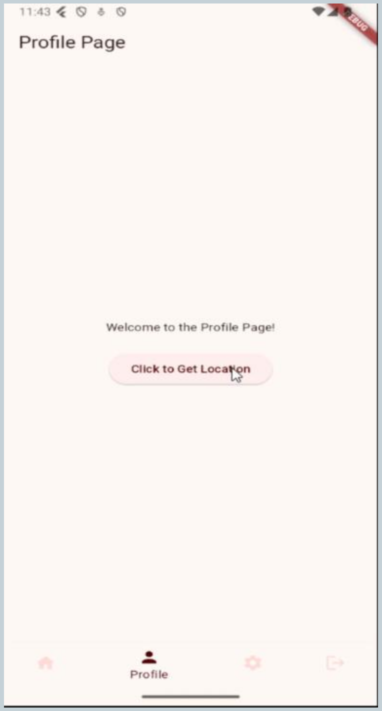
https://tasteful-panthers-default-rtdb.firebaseio.com

```
menu items for 2025-02-18
menu items for 2025-02-19
menu items for 2025-02-21
menu items for 2025-02-24
  breakfast
    Apple Pie
      0
        M: "Allergy: Milk"
      1
      2
```

Database location: United States (us-central1)

Task 2: Demos

- Simple Android app written in Dart (iOS theoretically looks the same)
- GPS and heads-up notifications in mobile app

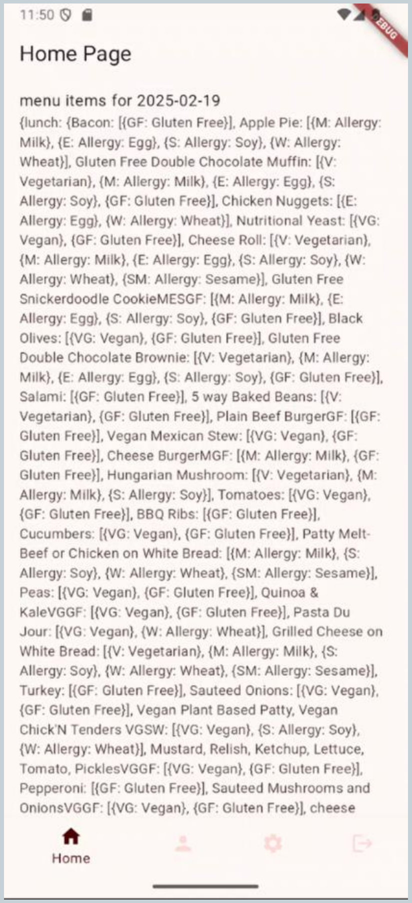


Profile Page

Welcome to the Profile Page!

Task 2: Demos

- Database integration with mobile app



Home Page

menu items for 2025-02-19

```
{lunch: {Bacon: [{GF: Gluten Free}], Apple Pie: [{M: Allergy:
Milk}, {E: Allergy: Egg}, {S: Allergy: Soy}, {W: Allergy:
Wheat}], Gluten Free Double Chocolate Muffin: [{V:
Vegetarian}, {M: Allergy: Milk}, {E: Allergy: Egg}, {S:
Allergy: Soy}, {GF: Gluten Free}], Chicken Nuggets: [{E:
Allergy: Egg}, {W: Allergy: Wheat}], Nutritional Yeast: [{VG:
Vegan}, {GF: Gluten Free}], Cheese Roll: [{V: Vegetarian},
{M: Allergy: Milk}, {E: Allergy: Egg}, {S: Allergy: Soy}, {W:
Allergy: Wheat}, {SM: Allergy: Sesame}], Gluten Free
Snickerdoodle CookieMESGF: [{M: Allergy: Milk}, {E:
Allergy: Egg}, {S: Allergy: Soy}, {GF: Gluten Free}], Black
Olives: [{VG: Vegan}, {GF: Gluten Free}], Gluten Free
Double Chocolate Brownie: [{V: Vegetarian}, {M: Allergy:
Milk}, {E: Allergy: Egg}, {S: Allergy: Soy}, {GF: Gluten Free}],
Salami: [{GF: Gluten Free}], 5 way Baked Beans: [{V:
Vegetarian}, {GF: Gluten Free}], Plain Beef BurgerGF: [{GF:
Gluten Free}], Vegan Mexican Stew: [{VG: Vegan}, {GF:
Gluten Free}], Cheese BurgerMGF: [{M: Allergy: Milk}, {GF:
Gluten Free}], Hungarian Mushroom: [{V: Vegetarian}, {M:
Allergy: Milk}, {S: Allergy: Soy}], Tomatoes: [{VG: Vegan},
{GF: Gluten Free}], BBQ Ribs: [{GF: Gluten Free}],
Cucumbers: [{VG: Vegan}, {GF: Gluten Free}], Patty Melt-
Beef or Chicken on White Bread: [{M: Allergy: Milk}, {S:
Allergy: Soy}, {W: Allergy: Wheat}, {SM: Allergy: Sesame}],
Peas: [{VG: Vegan}, {GF: Gluten Free}], Quinoa &
KaleVGGF: [{VG: Vegan}, {GF: Gluten Free}], Pasta Du
Jour: [{VG: Vegan}, {W: Allergy: Wheat}], Grilled Cheese on
White Bread: [{V: Vegetarian}, {M: Allergy: Milk}, {S:
Allergy: Soy}, {W: Allergy: Wheat}, {SM: Allergy: Sesame}],
Turkey: [{GF: Gluten Free}], Sauteed Onions: [{VG: Vegan},
{GF: Gluten Free}], Vegan Plant Based Patty, Vegan
Chick'N Tenders VGSW: [{VG: Vegan}, {S: Allergy: Soy},
{W: Allergy: Wheat}], Mustard, Relish, Ketchup, Lettuce,
Tomato, PicklesVGGF: [{VG: Vegan}, {GF: Gluten Free}],
Pepperoni: [{GF: Gluten Free}], Sauteed Mushrooms and
OnionsVGGF: [{VG: Vegan}, {GF: Gluten Free}], cheese
```

Task 3: Resolve Technical Challenges

- 2 out of the 4 of us have a Mac, so Swift dev and testing was going to be difficult for half of us
- Switched to Dart, which is the programming language for Flutter
- Flutter is a cross-platform framework that will allow us to create both an Android and iOS application
- We all successfully set up Flutter SDK and experimented programming in Dart

Task 4: Collaboration Tools

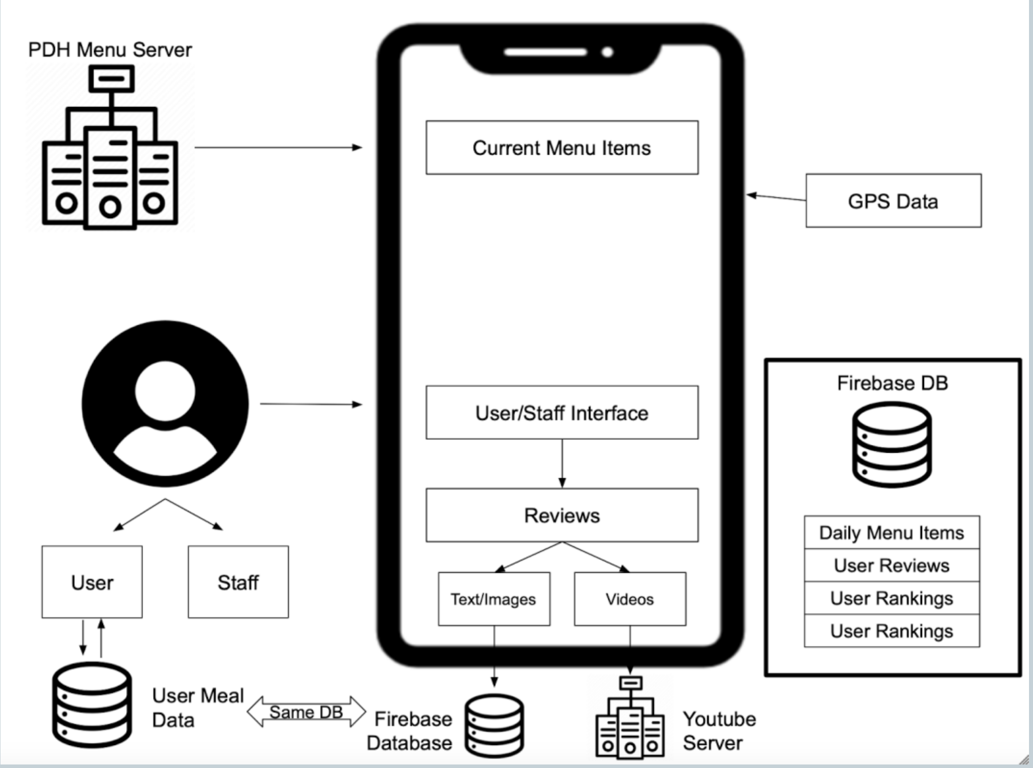
- Discord and iMessages for communication
- Google Docs for documentation
- GitHub for software dev
- No specific IDE, either VSCode or XCode depending on our systems since both support Flutter dev

Task 5: Requirements Document

- Wrote the requirements doc that covers purpose, scope, user characteristics, etc.
- In depth list of requirements for guests and staff including:
  - Functional requirements:
    - Review/suggestion creation
    - Flag raising/resolution
  - Interface requirements:
    - Login/signup interface
    - Dashboard
  - Performance requirements
    - Performance benchmarks
    - “This feature completes/returns within this many seconds”

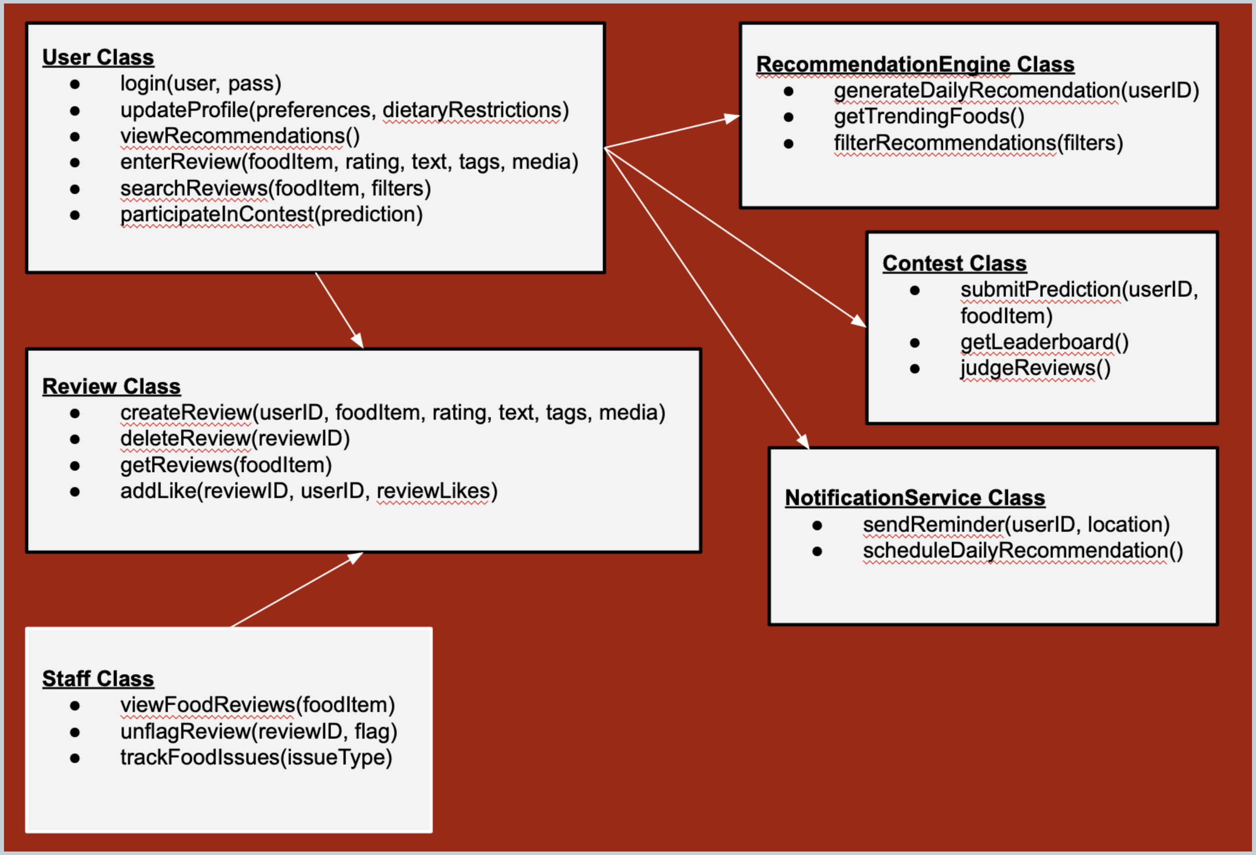
Task 6: Design Document

- System Architecture Diagram



Task 6: Design Document

- ER Database Diagram



User Class

- login(user, pass)
- updateProfile(preferences, dietaryRestrictions)
- viewRecommendations()
- enterReview(foodItem, rating, text, tags, media)
- searchReviews(foodItem, filters)
- participateInContest(prediction)

RecommendationEngine Class

- generateDailyRecomendation(userID)
- getTrendingFoods()
- filterRecommendations(filters)

Contest Class

- submitPrediction(userID, foodItem)
- getLeaderboard()
- judgeReviews()

Review Class

- createReview(userID, foodItem, rating, text, tags, media)
- deleteReview(reviewID)
- getReviews(foodItem)
- addLike(reviewID, userID, reviewLikes)

NotificationService Class

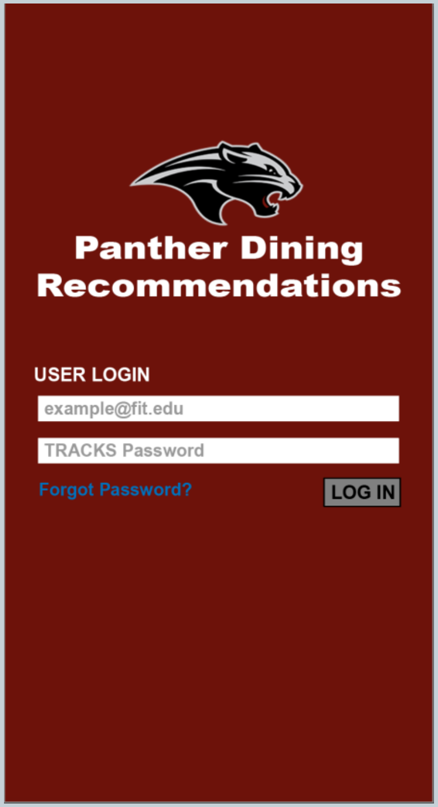
- sendReminder(userID, location)
- scheduleDailyRecommendation()

Staff Class

- viewFoodReviews(foodItem)
- unflagReview(reviewID, flag)
- trackFoodIssues(issueType)

Task 6: Design Document

- GUI Mockup
  - Login screen



Panther Dining Recommendations

USER LOGIN

example@fit.edu

TRACKS Password

Forgot Password?

LOG IN

Task 7: Test Document

- Test cases for covering the requirements document
- Following table format:

Test Case ID	Reference Number	Requirement	Input	Expected Output
TC-PC-001	3.1.1.1.1	Profile Creation (Guests)	Guest enters valid PAWS credentials and selects dietary preferences	Reviewer profile is created with a unique username and saved dietary preferences
TC-PC-002	3.1.1.1.1	Profile Creation (Guests)	Guest enters invalid PAWS credentials	Profile creation fails with an error message
TC-PMR-001	3.1.1.2.1	Meal Recommendations	Guest has previous meal reviews and dietary preferences set	Personalized meal recommendations appear in the app and via notifications
TC-GPS-001	3.1.1.2.2	GPS-Based Recommendations	Guest enters PDH with location services enabled	Meal recommendation notification is sent upon arrival

Milestone 2 Task Matrix

Task	Alex	Anthony	Jiahao	Lexi
Dev & Demo Accounts	40%	20%	20%	20%
Dev & Demo Web Scraping	20%	40%	20%	20%
Dev & Demo Mobile Reviewing	20%	20%	20%	20%
Dev & Demo Database	20%	20%	40%	20%

Thanks

- Questions?