Milestone 1 Progress Evaluation

Tasteful Panthers: Food Recommendation at Dining Halls

Team Members
Alex Laureano, elaureano2021@my.fit.edu
Jiahao Shu, jshu2023@my.fit.edu
Anthony Hordesky, ahordesky2022@my.fit.edu
Lexi Franklin, afranklin2021@my.fit.edu

Faculty Advisor
Philip Chan, pkc@cs.fit.edu

Current Milestone Task Matrix

Task	Completion	Alex	Jiahao	Anthony	Lexi	To Do
Compare and Contrast Technical Tools	100%	Web/Information Scraping	IDE	Database	IDE	N/A
Tool Demos	100%	Web/Information Scraping	IDE,	Database	IDE	Create Database-App Interaction Demo, Image & GPS Testing
Resolve Technical Challenges	80%	Swift, sorting algorithm	Swift, sorting algorithm	Swift, algorithm, encryption/decryption	Swift, sorting algorithm	N/A
Collaboration Tools	100%	Communication	Documents/presentations	Software development	Task calendar	N/A
Requirement Document	99%	Write 40%	Write 20%	Write 20%	Write 20%	N/A
Design Document	99%	Write 20%	Write 20%	Write 20%	Write 40%	Complete all interface prototypes
Test Plan	99%	Write 20%	Write 20%	Write 40%	Write 20%	N/A

Discussion of Tasks
Compare & Select Dev Tools: We deliberated for some time over the correct languages and platforms to use to develop the app, and ended up settling (albeit a bit late) on Flutter for cross-platform app development and Firebase for database operations

Provide Small Demos: Working demo's displaying app screens, web scraping, and database operations, but we need to make sure the phone can use GPS, display, capture and import images, and interact with our database to which we have saved scraped data.

Learn App Dev Language: After deciding on Flutter, we worked on tutorials to gain a functional understanding of the language we intended to use

Select tools for Collaborative Software Dev: The popularity and ease of use of the most popular options for collaborative development made our choice easy: Github for program version control and Google Docs for sharing important documents, as well as discord and normal texting for direct team communication

Requirements Doc, Test Plan: Complete

Design Doc: Many screens were left as mock ups and early prototypes which were insufficient for our purposes

Discussion of Contribution
Alex Laureano: For the first milestone, I wrote the script to scrape PDH menu information from the school website. This information is public and is available a month in advance, so we will have menu information for each day easily accessible to use in our app. I also made sure that the information scraped was noncontingent and correct. I made a web scraping demo to show that this works. I also introduced the group 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. I made a mobile app demo. Finally, I helped to write both the requirement and design documents.

Anthony Hordesky: During this current milestone, I compared and selected a database for the mobile application. We decided to go with Firebase DB due to the cloud and real time functionality that it includes. To demo our database, I integrated the web scraper with the Firebase Realtime Database i.e. wrote menu information to the database. GitHub was already chosen for our code sharing collaborative tool during our work on the project plan, along with communication via Discord and iMessages. During this current milestone, we did implicitly decide on using Google Docs for document creation and collaboration. Alex's suggestion of using Dart for app dev was agreed upon, which means that we do not need to have a specific IDE that we all work in. To help me learn Dart, I wrote the GPS demo for the app. Finally, I also wrote the requirement document, test plan document, and the milestone presentation.

Jiahao Shu: For milestone one, I tested the Flutter application on macOS and iOS devices, ensuring compatibility within the Apple ecosystem. I deployed the application to a physical iPhone device for real-world testing, verifying UI responsiveness, interaction logic, and

compatibility with hardware features. During testing, I identified potential issues such as UI rendering inconsistencies and touch interaction irregularities, providing feedback to the team for further optimization.

Lexi Franklin: For the first milestone, I designed a mock-up GUI for all screens in our app for both the guest and staff end. Creating a smooth layout was a top priority to promote usability. Using Alex's rough draft of a demo, I was able to test the functionality in macOS/iOS uploading the files from our shared GitHub. I worked a little bit with Dart in a macOS environment to familiarize myself with its functions. I wrote the majority of the design document, including creating system architecture diagrams.

Next Milestone

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%

Discussion of Planned Tasks
Dev & Demo Accounts: We have to create the user account data structure, implement it into the database and implement checking for valid login info

Dev & Demo Web Scraping: Web Scraping is functionally complete though could use some polishing

Dev & Demo Mobile Reviewing: App exists but has no functionality, we must implement reviews entirely

Dev & Demo Database: Database contains information but app has not yet been configured to use the database.

Dates Met With Client: 2/21, 2/7

Client Feedback
See Faculty Advisor Feedback below.

Dates Met
We met with our Client / Faculty Advisor on 2/7 and 2/21

Faculty Advisor Feedback
February 7th, 2025 -
 Requirements : add section numbers in different levels, focus on one user type at a time, put common requirements in different section to avoid repetition
-- Getting recommendations

   change "location services" to "getting recommendations while entering PDH"

-- Guests can suggest meals:

   a. provide links to recipes

   b. vote on suggested meals

-- Interface requirements:

  Guest Home screen / Dashboard:

* new section on Home Screen for Guests ("dashboard"):

   1. What is tasty today?

   a. recommended to try by my tasteful twin
       * explanation: we both like the same n dishes
       * when clicked, display the n dishes

   b. my favorite in case I forget
       * the dish today that I rated highest in the past
       * when clicked, go to "enter/search/view review" screen

   c. crowd's favorite
       * the dish that has the highest rating today
       * when clicked, go to leaderboard on today's dish ranking based
on ratings

   d. meal suggestion becomes reality
       * based on meal suggestions, the dish that the staff decided to
cook today
       * when clicked, go to leaderboard on meal suggestions

   2. How am I ranked?

    a. forecast rank: #n
       * when clicked, go to leaderboard for the forecasting contest

    b. top reviewer rank: #n
       * when clicked, go to leaderboard for the review contest

* new section on Home Screen for staff ("dashboard")
  a. writing reviews--guests can include images and videos

Test Plan : test case id, reference number, input, output - these are table elements
Design Document :
System architecture diagram: draw users, machines, software components for each machine (user phone and server). Refer to the senior project website for an outline.
i. Phone: ui for mobile app, GPS, camera for image/videos,
ii. Server: database split into menus, reviews, competition, etc.
GUI Mockup: layout for each screen
ER Diagram for database:
Classes/Methods Diagram: UML diagram, depends
System Arch Diagram: add PDH Server (external system where you
scrape food items)

on videos, as suggested in my slides, they can be YouTube links
in the reviews. That is, YouTube app and YouTube server are involved in
the design. This has an advantage of not needing to store videos.

on images, your app interacting with Gallery might be easier
than with Camera. That is, the images are taken outside your app and
imported from Gallery.
        4. Demos
Menu scraping: investigate wrong date issue; how to scrape multiple dates/available menu dates, at least today and next day for prediction challenge
Phone Emulator/Language: demo basic gps, simple GUI with a few screens, camera integration, sliders for rating
Database: setup user profile database for an example
If possible, phone and database communication: display simple user profile
while displaying an image is likely supported, is taking an
image supported?

are playing and taking a video supported?

if not, you might want to use existing apps such as Camera,
Gallery, YouTube
    5. Tool Comparison Tables
Phone Emulator/Language: Xcode for mac users, gps/camera integration, figure out iphone emulation on windows (vm)

Database: look at other factors, ease of integration with iOS, experience with possibilities before choosing

    6. Draft Progress Evaluation for next meeting (course website for template)

February 21st, 2025

Requirement Document -

        Meal suggestions are completely different from reviews
        Ranking suggestions on likes by users so pdh staff knows what to bring to the dining hall
        Tomorrow's Prediction needs a leaderboard for reviewers with most successful guesses and a leaderboard for most popular dishes of the day based on star ratings from reviews associated with the food item
        Leaderboards can have bars which better depict to the user who is winning, we can include the bar with their username rather than just a username and a score
        Guest Dashboard/Home Screen
            Much like a mobile app screen with sections or portals that go to new pages
            6 things for home screen
            Crowd favorite
            Your favorite
            Meal recommendation
            Tasteful Twin
            Your ranking
            Tomorrows prediction (leaderboard for forecasting correctly)

        Staff Dashboard/Home Screen
            Same mobile home screen type of layout
            Different sections than guest dashboard

        Bottom toolbar for mobile app should have home (reviews/suggestions), dashboard, profile, and settings

        Test Document -

        Design Document -
   a. Missing system architecture diagram
      i.   Add External Machines
          1. PDH Server
          2. Youtube Server
          3. Phone
          4. Database
   b. Missing home screen
   c. Writing review screen include

   i. Text box
   ii. Adding images/videos
   iii. Star ratings
   iv. Tags / Keywords
   v. Submit Button
 d. Read Reviews
   i. Search & Filter
     1. Tags
     2. Stars
 e. Leaderboards
   i. Include Scoring Rhetoric
   ii. Table Display Format
   iii. Bars or other Visuals
 f. Missing ER Diagram
 g. Missing Demos
   i. Phone to Database
   ii. Displaying Images in App
   iii. Importing Images from Gallery
   iv. GPS Demo
 h. Missing Comparison Table
   i. Phone Emulator/Language
 i. Missing Progress Evaluation Draft
   i. Provide Draft 2/22, Sat @ Noon