CSC 431

Feed Me  
  
System Architecture Specification (SAS)

|  |  |
| --- | --- |
| Clinton Jules | Developer |
| Hao Wu | Developer |
| Jacob Steirer | Developer |
| Maria Sabina Echeveste | Developer |
| Samuel Mistretta | Developer |
| Tijana Canic | Developer |
| Yuxuan Liu | Developer |

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author(s) | Change Comments |
| 1 | Mon March 30th | Feed Me Team | First Draft |
| 2 | Mon May 4th | Feed Me Team | Final Draft |
|  |  |  |  |
|  |  |  |  |

# Table of Contents

[Version History 2](#_Toc39418154)

[Table of Contents 3](#_Toc39418155)

[Table of Tables 5](#_Toc39418156)

[Table of Figures 6](#_Toc39418157)

[1. System Analysis 7](#_Toc39418158)

[1.1. System Overview 7](#_Toc39418159)

[1.2. System Diagram 7](#_Toc39418160)

[1.3. Actor Identification 7](#_Toc39418161)

[1.4. Design Rationale 7](#_Toc39418162)

[1.4.1. Architectural Style 8](#_Toc39418163)

[1.4.2. Design Pattern(s) 9](#_Toc39418164)

[1.4.3. Framework 10](#_Toc39418165)

[2. Functional Design 11](#_Toc39418166)

[2.1. Sequence Diagram 11](#_Toc39418167)

[3. Structural Design 12](#_Toc39418168)

[3.1. Class Diagram 12](#_Toc39418169)

# Table of Figures

[Figure 1: System Diagram 7](#_Toc39423806)

[Figure 2: Sequence Diagram 10](#_Toc39423807)

[Figure 3: Class Diagram 11](#_Toc39423808)

# System Analysis

## System Overview

<Describe the system in brief and your architecture choice>

The system comprises three main parts: two application views and a data manager class. The two application views (Home View & Meal View) will display options and information to the users and have classes associated with them to add functionality.

The Home View has filters for item choice and a button to generate an item based on filters and randomization. The Meal View displays choices generated from the Home View button, displays links/options to get that item and has a back button to return to the Home View.

Finally, the Database Manager class will access the database file(s) available restaurants, meals, and delivery options that can be retrieved through queries within the app structure.

The chosen architecture for this system is an event-driven architecture. This style allows the system to respond to different user choices when interacting with the system. It allows for a more natural and reactive flow of choices and displays for the system. This is best seen in the sequence diagram.

## System Diagram

Figure 1: System Diagram

![A screenshot of a cell phone

Description automatically generated]()

## Actor Identification

<Identify all actors interacting with the system>

The only actor involved with the system is the user. The user selects the use of switch filters to narrow down meal search if they choose to do so and allows the system to generate a meal choice for the user.

## Design Rationale

### Architectural Style

The style chosen was event-driven architecture as it allows the system to react to several different user choices and many of the system actions are reliant on certain events. For example, (1) the user selects the ‘Feed Me’ button so that a meal is generated this then causes (2) the system to search for restaurants that match the criteria set. Once one is randomly chosen, (3) the system opens a second view displaying the choice (4) If the user selects an external link, the app closes (5) if the user wants a different choice they select the back button which then (6) returns the user to the Home View of the application, the system then (7) counts the try as ⅓ which could lead to locking the button for an hour if three are chosen. This type of pipeline lends to an event-driven architecture naturally.

### Design Pattern(s)

* Singleton: To maintain a single consistent database reference
* Decorator: The system will be using a wrapper to utilize SQLite and implement a database within the system
* Command: To encapsulate information that will be sent to the second view upon user request, i.e. the meal information that will be sent to the Meal View from Home View.

### Framework

<Identify and briefly explain the frameworks you are using, if any. Also specify the rationale behind selecting this framework>

The mobile application will run on Xcode using UIKit[[1]](#footnote-1) framework which allows for UI elements to be easily utilized such as UI buttons and UI views which are integral to our system.

# Functional Design

<Identify all significant workflows as sequence diagrams using the following format>

## Sequence Diagram

Figure 2: Sequence Diagram

![A screenshot of a cell phone

Description automatically generated]()

# Structural Design

<Identify all components and model them using class diagrams>

## Class Diagram

Figure 3: Class Diagram

A screenshot of a cell phone

Description automatically generated

1. Links: <https://developer.apple.com/documentation/uikit> [↑](#footnote-ref-1)