# **System and Software Architecture Description (SSAD)**

**"Fooder"**

**QWERTY**

**Jake Motta - Co-Lead Programmer, Graphics Designer**

**Alex Le - Project Manager, Co-Lead Programmer**

**Brandy Kao - Co-Lead Programmer, Tester**

**Jason Springer - Co-Lead Programmer, Quality Assurance**

**3/8/2017**

# **Version History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Author | Version | Changes made | Rationale |
| 08/25/05 | PA | 2.0 | · Original template for use with Instructional ICM-Sw v1.0 | · Initial draft for use with Instructional ICM-Sw v1.0 |
| 05/22/09 | SK | 2.1 | · Embedded description in each Table | · To be consistent with ICM EPG template set standard V2.1 |
| 10/04/16 | BO | 2.2 | In Technology-Independent Model Design  · Added conceptual domain model  · Separated "Process Realization Diagram" into "Robustness diagram" and "Sequence Diagram" | · For research data collection |

# **Table of Contents**

System and Software Architecture Description (SSAD).......................................................... i

Version History............................................................................................................................. ii

I'm

Table of Contents........................................................................................................................ iii

Table of Tables............................................................................................................................. iv

Table of Figures............................................................................................................................. v

1. Introduction............................................................................................................................. 1

1.1 Purpose of the SSAD................................................................................................... 1

1.2 Status of the SSAD...................................................................................................... 1

2. System Analysis...................................................................................................................... 2

2.1 System Analysis Overview.......................................................................................... 2

2.2 System Analysis Rationale......................................................................................... 5

3. Technology-Independent Model............................................................................................ 6

3.1 Design Overview........................................................................................................... 6

3.2 Design Rationale.......................................................................................................... 8

4. Technology-Specific System Design.................................................................................... 9

4.1 Design Overview........................................................................................................... 9

4.2 Design Rationale........................................................................................................ 10

5. Architectural Styles, Patterns and Frameworks............................................................... 11

# **Table of Tables**

Table 1: Actors Summary. 2

Table 2: Artifacts and Information Summary. 3

Table 3: Process Description. 4

Table 4: Typical Course of Action. 4

Table 5: Alternate Course of Action. 4

Table 6: Exceptional Course of Action. 4

Table 7: Hardware Component Description. 7

Table 8: Software Component Description. 7

Table 9: Supporting Software Component Description. 7

Table 10: Design Class Description. 8

Table 11: Hardware Component Description. 9

Table 12: Software Component Description. 9

Table 13: Supporting Software Component Description. 10

Table 14: Design Class Description. 10

Table 15: Architectural Styles, Patterns, and Frameworks. 11

# **Table of Figures**

Figure 1: System Context Diagram.. 2

Figure 2: Artifacts and Information Diagram.. 3

Figure 3: Process Diagram.. 4

Figure 4: Conceptual Domain Model 6

Figure 5: Hardware Component Class Diagram.. 6

Figure 6: Software Component Class Diagram.. 6

Figure 7: Deployment Diagram.. 6

Figure 8: Supporting Software Component Class Diagram.. 7

Figure 9: Design Class Diagram.. 8

Figure 10: Robustness Diagram.. 8

Figure 11: Sequence Diagram.. 8

Figure 12: Hardware Component Class Diagram.. 9

Figure 13: Software Component Class Diagram.. 9

Figure 14: Deployment Diagram.. 9

Figure 15: Supporting Software Component Class Diagram.. 9

Figure 16: Design Class Diagram.. 10

Figure 17: Process Realization Diagram.. 10

### **1.** **Introduction**

#### **1.1** **Purpose of the SSAD**

This document defines the architecture for the "Fooder"” application. It breaks down the multiple aspects of the system using charts, tables, and diagrams. The goal of this document to include as many details of the entire application as well as their functions.

#### **1.2** **Status of the SSAD**

The status of the SSAD is in the original form. Updates will be made to the document as more design features are added. This is considered to be the first version of the SSAD.

### **2.** **System Analysis**

#### **2.1** **System Analysis Overview**

The primary purpose of the "Fooder” application is to help users to find a place to eat in a fun and intuitive way. Once the user is logged in to their profile "Fooder" will keep track of the user's preferences including location, types of restaurants to search for. The information on restaurants will be current and will include information such as price, ratings, description, pictures taken, and the ability to get directions to the location. All favorited restaurants will be saved to the profile for review at a later time.

##### **2.1.1** **System Context**

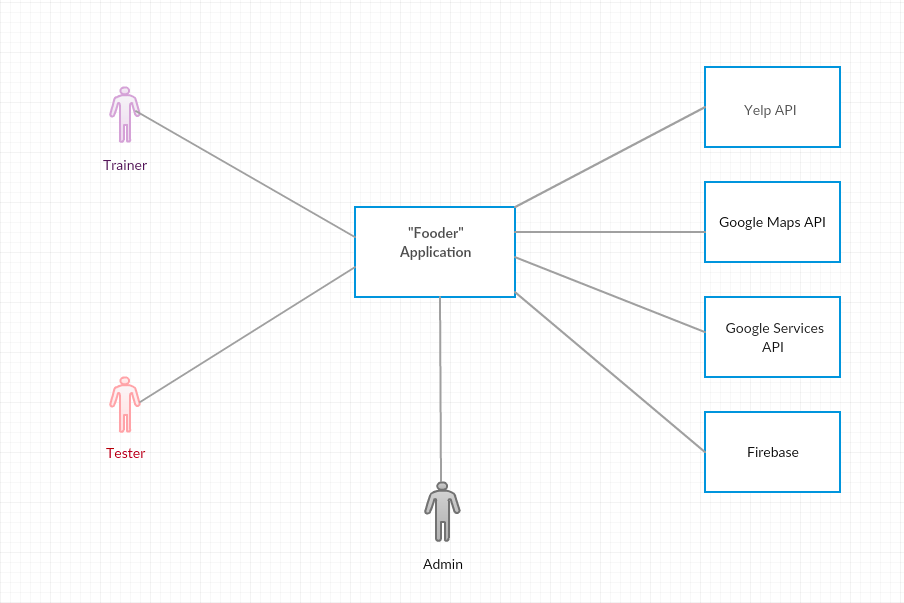


Figure 1: System Context Diagram

Table 1: Actors Summary

|  |  |  |
| --- | --- | --- |
| **Actor** | **Description** | **Responsibilities** |
| Jason Springer | Co-Lead Programmer | Coding, Quality Assurance |
| Alex Le | Project Manager | Documentation, Framework |
| Jake Motta | Co-Lead Programmer | Coding, Testing, Graphic Artist |
| Brandon Kao | Co-Lead Programmer | Coding, Testing |

##### **2.1.2** **Artifacts & Information**

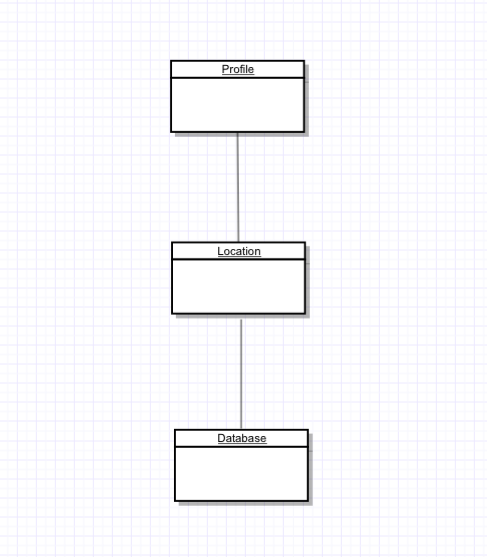
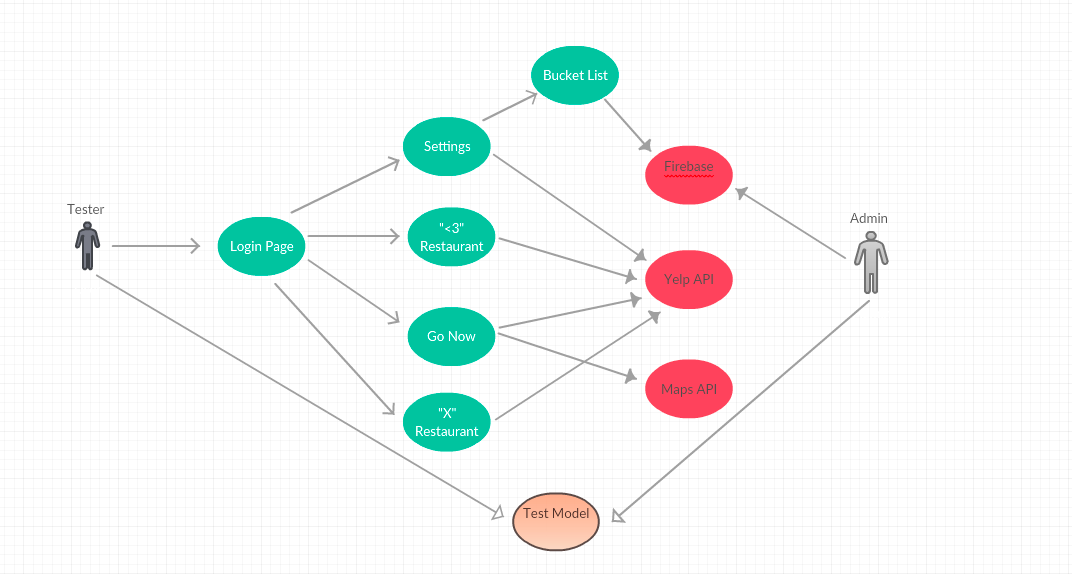


Figure 2: Artifacts and Information Diagram

Table 2: Artifacts and Information Summary

|  |  |
| --- | --- |
| **Artifact** | **Purpose** |
| Profile | Save settings, favorites list |
| Location | Find restaurants nearby |
| Database | To save favorites and registered users |

##### **2.1.3** **Behavior**



<<Use-Case Diagram>>

Figure 3: Process Diagram

###### **2.1.3.1** **Capability x**

2.1.3.1.1 Process y

Table 3: Process Description

|  |  |
| --- | --- |
| **Identifier** | Add Favorite Restaurant |
| **Purpose** | Let user save restaurants for future lookup |
| **Requirements** |  |
| **Development Risks** | Process needs to insure the user will be able to add restaurant to existing favorites database and user profile list, it will also need to inform user if restaurant cannot be added. |
| **Pre-conditions** | User must already have existing user profile, restaurant must be existing restaurant in yelp API |
| **Post-conditions** | User train model |

Table 4: Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Trainer swipes right | Restaurant image will be swiped to the right and the heart will bounce and restaurant will be added to user favorites list |

Table 5: Alternate Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Trainer clicks the heart button | Restaurant image will be swiped to the right and the heart will bounce and restaurant will be added to user favorites list |

Table 6: Exceptional Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Click on Restaurant Image | Restaurant image shifts up and restaurant details such as cost and stars will appear |
| **2** | Click on heart | Restaurant image with details gets shifted to right and restaurant is added to user favorite lists |

|  |  |
| --- | --- |
| **Identifier** | Decline Restaurants |
| **Purpose** | Decline restaurant |
| **Requirements** |  |
| **Development Risks** | The process needs to insure that users will be able to decline a restaurant and move to the next restaurant. |
| **Pre-conditions** | User must have an existing user profile and restaurant must be existing in Yelp API |
| **Post-conditions** | User train model |

Table 4: Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Trainer swipes left | Restaurant image will be swiped to the left and the “x” button will bounce and restaurant will be removed from restaurant array |

Table 5: Alternate Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Trainer clicks the “X” button | Restaurant image will be swiped to the left and the “X” button will bounce and restaurant will be removed from restaurant array |

Table 6: Exceptional Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Click on Restaurant Image | Restaurant image shifts up and restaurant details such as cost and stars will appear |
| **2** | Click on “X” button | Restaurant image with details gets shifted to left and restaurant is removed |

|  |  |
| --- | --- |
| **Identifier** | Go to restaurant NOW using Map application |
| **Purpose** | Allow user to direct themselves to restaurant immediately |
| **Requirements** |  |
| **Development Risks** | Process needs to allow users to get directions from their current location to get to their desired choice of restaurant |
| **Pre-conditions** | User must already have existing user profile, restaurant must be existing restaurant in yelp API, Google Maps Integration |
| **Post-conditions** | User train model |

Table 4: Typical Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Trainer clicks map pin icon | Device will provide a pop up giving user a message: “this will open up maps” with “okay” or “cancel” button |
| **2** | Trainer clicks “okay” button | System redirects to Google Maps application |

Table 5: Alternate Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** |  |  |

Table 6: Exceptional Course of Action

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Trainer clicks map pin icon | Device will provide a pop up giving user a message: “this will open up maps” with “okay” or “cancel” button |
| **2** | Trainer clicks “cancel” button | System will go back to regular page with all 3 available actions: decline, favorite, or go now. |

##### **2.1.4** **Modes of Operation**

Testing mode: This mode is a beta mode that can be used by testers. It will allow testers to use the application without having to create a user login.

#### **2.2** **System Analysis Rationale**

Since our aim is towards a larger audience of those with a larger age gap which includes non-technical individuals, our design will need to be extremely simple and easy to use. We aim to create an application that draws in users through noticeable application design and simple instructions.

### **3.** **Technology-Independent Model**

#### **3.1** **Design Overview**

##### **3.1.1** **System Structure**

<<Conceptual Domain Model>>

Figure 4: Conceptual Domain Model

<<Hardware Component Class Diagram>>

Figure 5: Hardware Component Class Diagram

<<Software Component Class Diagram>>

Figure 6: Software Component Class Diagram

<<Deployment Diagram>>

Figure 7: Deployment Diagram

<<Optional: Supporting Software Infrastructure Diagram>>

Figure 8: Supporting Software Component Class Diagram

Table 7: Hardware Component Description

|  |  |
| --- | --- |
| **Hardware Component** | **Description** |
| Computer | For Developing App |
| Android Device | Testing |
|  |  |

Table 8: Software Component Description

|  |  |
| --- | --- |
| **Software Component** | **Description** |
| Android Studio | Main IDE for development |
|  |  |
|  |  |

Table 9: Supporting Software Component Description

|  |  |
| --- | --- |
| **Support Software Component** | **Description** |
| Google Maps | Directions to restaurant |
| Yelp | Restaurants, Price, Distance, Pictures |
|  |  |

##### **3.1.2** **Design Classes**

###### **3.1.2.1** **<Classes n>**

<<Design Classes Class Diagram>>

Figure 9: Design Class Diagram

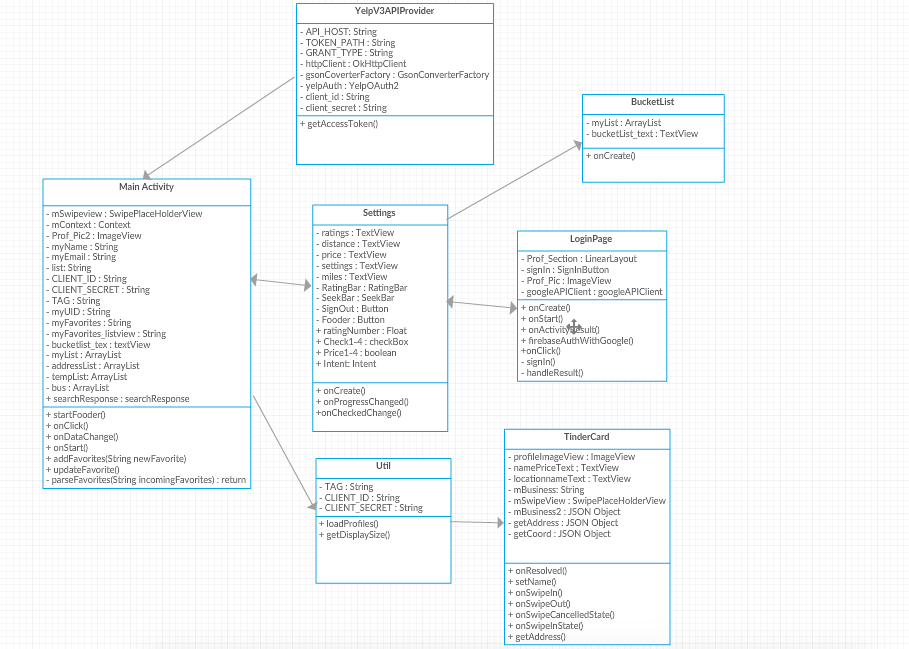
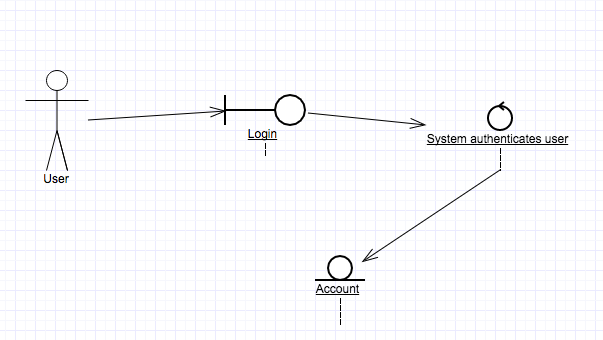


Table 10: Design Class Description

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| MainActivity | Activity | Runs main functions of app |
| SwipeCard | Functions | Swipe Functionality |
| Profile | Attributes | User profile data |
| Utils | Functions | Methods used by maid activity |

##### **3.1.3** **Process Realization**

<<Robustness Diagram>>

Figure 10: Robustness Diagram

<<Sequence Diagram>>

Figure 11: Sequence Diagram

#### **3.2** **Design Rationale**

This type of architecture/design was chosen for this application because we wanted an app that functions in a similar way to the popular Tinder application. We wanted to keep the same swiping actions and make the application make sense for restaurant use.

### **4.** **Technology-Specific System Design**

#### **4.1** **Design Overview**

##### **4.1.1** **System Structure**

<<Hardware Component Class Diagram>>

Figure 12: Hardware Component Class Diagram

<<Software Component Class Diagram>>

Figure 13: Software Component Class Diagram

<<Deployment Diagram>>

Figure 14: Deployment Diagram

Table 11: Hardware Component Description

|  |  |
| --- | --- |
| **Hardware Component** | **Description** |
| Laptops | Used for programming the app |
| Android Phones | Used to test the app |
|  |  |

Table 12: Software Component Description

|  |  |
| --- | --- |
| **Software Component** | **Description** |
| Android Studio | Used to program the app |
|  |  |
|  |  |

Table 13: Supporting Software Component Description

|  |  |
| --- | --- |
| **Support Software Component** | **Description** |
| Google Maps | Used to help locate the restaurant |
| Facebook/Google API | Used to help authenticate the user |
| Yelp API | Used to help search for restaurants |

##### **4.1.3** **Process Realization**

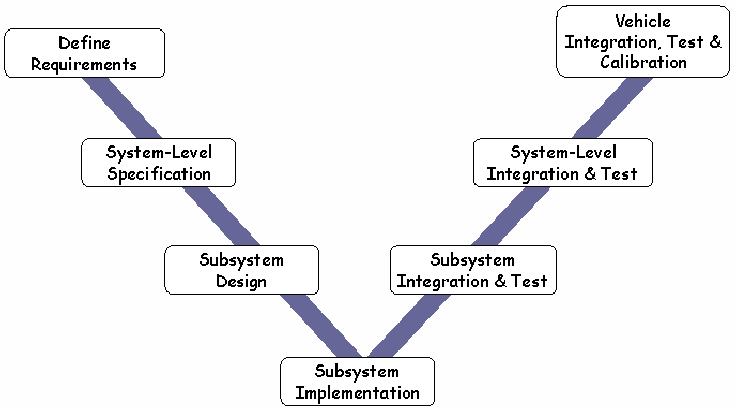


Figure 17: Process Realization Diagram

#### **4.2** **Design Rationale**

### **5.** **Architectural Styles, Patterns and Frameworks**

Table 14: Architectural Styles, Patterns, and Frameworks

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Benefits, Costs, and Limitations** |
| Java | Programming language | Easy to use, Free, Speed |
| Firebase | Database | Free from Google |
|  |  |  |