FEED ME

System Requirements Specification

Version 2.0

Document Number: SRS-002

**Project Team: **A23****

**Project Members: **Peter Holoubek** (ph1335), **Teddy Kim** (tk1931), **JinZhao Su** (js9202)**

# REVIEW AND APPROVALS

|  |  |  |  |
| --- | --- | --- | --- |
| Printed Name | Function | Date | Signature |
| Theodore Kim | Author | February 27, 2019 |  |
| JinZhao Su | Author | February 27, 2019 | https://scontent-lga3-1.xx.fbcdn.net/v/t1.15752-9/52861846_2336825663028189_6860368647928414208_n.png?_nc_cat=107&_nc_ht=scontent-lga3-1.xx&oh=8bb7d8bce3fedf38668fe3f9d05d7a6b&oe=5D258DEB |
| Petr Holoubeck | Author | February 27, 2019 | https://scontent-lga3-1.xx.fbcdn.net/v/t1.15752-9/s2048x2048/52825818_155163132053922_2270601179052376064_n.png?_nc_cat=107&_nc_ht=scontent-lga3-1.xx&oh=4b9107141ccddee8f9ebf13e5428e56c&oe=5CDB1CAF |
| Theodore Kim | Author | March 28, 2019 |  |
| JinZhao Su | Author | March 28, 2019 | https://scontent-lga3-1.xx.fbcdn.net/v/t1.15752-9/52861846_2336825663028189_6860368647928414208_n.png?_nc_cat=107&_nc_ht=scontent-lga3-1.xx&oh=8bb7d8bce3fedf38668fe3f9d05d7a6b&oe=5D258DEB |
| Petr Holoubeck | Author | March 28, 2019 | https://scontent-lga3-1.xx.fbcdn.net/v/t1.15752-9/s2048x2048/52825818_155163132053922_2270601179052376064_n.png?_nc_cat=107&_nc_ht=scontent-lga3-1.xx&oh=4b9107141ccddee8f9ebf13e5428e56c&oe=5CDB1CAF |

# REVISION HISTORY

|  |  |  |
| --- | --- | --- |
| Date | Revision Number | Purpose |
| February 27, 2019 | Version 1.0 | Initial Document Release |
| March 28, 2019 | Version 2.0 | Defect Resolution and added Systems Requirement Specification |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

TABLE OF COTENTS

[1. INTRODUCTION 1](#_Toc2260658)

[1.1 Purpose 1](#_Toc2260659)

[2. SCOPE 1](#_Toc2260660)

[2.1 Identification 1](#_Toc2260661)

[2.2 Bounds 2](#_Toc2260662)

[2.3 Objectives 2](#_Toc2260663)

[3. OVERALL SYSTEM OVERVIEW 3](#_Toc2260664)

[3.1 Context Diagram 3](#_Toc2260665)

[3.2 Additional Descriptive Items 4](#_Toc2260666)

[4. DOCUMENT OVERVIEW 5](#_Toc2260667)

[5. REFERENCE DOCUMENTS 6](#_Toc2260668)

[6. BUSINESS REQUIREMENTS 6](#_Toc2260669)

[6.1 Technology 6](#_Toc2260670)

[6.2 Economics 6](#_Toc2260671)

[6.3 Regulatory and Legal 7](#_Toc2260672)

[6.4 Market Considerations 7](#_Toc2260673)

[6.5 Risks and Alternatives 7](#_Toc2260674)

[6.6 Human Resources and Training 8](#_Toc2260675)

[7. SPECIFIC REQUIREMENTS (FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS) 8](#_Toc2260676)

[7.1 Functional Descriptive Detailed Requirements 8](#_Toc2260677)

[7.2 Requirements Use Cases 8](#_Toc2260678)

[7.3 Non-Functional Descriptive Detailed Requirements 12](#_Toc2260679)

[8. SYSTEM TEST PLAN REQUIREMENTS 18](#_Toc2260680)

[9. QUALIFICATION PROVISIONS 19](#_Toc2260681)

[10. REQUIREMENTS TRACEABILITY 19](#_Toc2260682)

[11. EVOLUTION OF THE SRS 19](#_Toc2260683)

[12. RATIONALE 20](#_Toc2260684)

[13. NOTES 20](#_Toc2260685)

[14. APPENDICES 21](#_Toc2260686)

## 1. INTRODUCTION

### Purpose

This document provides all the proposed requirements for the *Feed Me* cross-platform application. Specifically, the document addresses the general, high level overview of the application and the functional and non-functional requirements for said system. This specification would be useful for the software developers on the project. Additionally, this document specifies the considerations taken for the development of the business aspect of the application and project. The business definition is aimed at the project managers and business managers on the project as a guide for developed business practices. The components of this document which outline the assurance of the document’s quality as well as the verification and validation of the final system for its fulfillment of the requirements serve as directives for the Software Quality Assurance (SQA) team. Furthermore, this document serves as an agreement between the project client and project team on the requirements for the final system. Hence, the document serves as the foundation for communication between the two parties such that the final product delivered conforms to the client’s expectations.

## 2. SCOPE

The system being produced is the Feed Me cross platform application. The Feed Me system includes a front-end application and a back-end server. The system should be able to make informed and accurate decisions about where (and when) a user should eat with minimal user input.

One pain point which was identified among potential customers is that traditional online food recommendation services require some amount of vetting and research as services such as Yelp may have ingenuine reviews and ratings. Therefore, a service that performed quality control and screening is a need among said consumers. Hence, the customer’s expectations are that the product *make a decision* for the user rather than just offer a search functionality through a database of restaurant options. This decision should require minimal user input and should be representative of their individual preferences. Finally, the product should product a social interaction component to allow different users to coordinate food decisions collaboratively.

The product can be used not only in the daily lives of individuals looking to eat out, but also o travelers looking to become familiar with local cuisines and restaurants looking to advertise their business to potential customers.

The goal of the system as a whole is to provide a food choice as good as if an individual made the choice themselves (unassisted).

### 2.1 Identification

This document is the Systems Requirement Specification (SRS-001). It is meant to guide the development and release of the first version of the *Feed Me* application. It is being submitted by Team A23. Version 1 was submitted to the project authority on February 27, 2019 and Version 2 was submitted to the project authority on March 28, 2019.

### 2.2 Bounds

The proposed system includes a means of making decisions for users based upon their individual preferences (which they provide to the product) and restaurant data provided by 3rd party application programming interfaces (APIs). It also provides a means for businesses to advertise to the product’s users and target audiences that have demonstrated behaviors and preferences compatible to said businesses. It is important to distinguish that the system DOES NOT track information about individual restaurants (location, food type, hours of operation, etc.) but rather receives that information from the use of APIs provided by other services which do track that information. The system simply aggregates said data and matches individual’s preferences to said restaurant choices.

### 2.3 Objectives

This project is will be delivered as a single, completed product at the designated delivery date (currently, ***to be determined***, as of Version 1 of this document).

This project is not critical; however, it should be treated with high priority because competitor research has revealed several other projects currently under development that are seeking to accomplish the same goals as the proposed product, therefore, development of the project should not be delayed significantly in order to gain some advantage in the potential market. See Table 1 for the project’s initial document deliverables and their descriptions and deadlines.

TABLE 1

Project Document Deliverables

|  |  |  |
| --- | --- | --- |
| Deliverable | Description | Deadline |
| Project Proposal | An outline of the project’s motivation, description of its requirements and constraints, description of its concept, and list of deliverables | February 16, 2019 |
| Software Business Specification | An outline of the business needs of the project, as well as a development of the product requirements and proposed decomposition of the product. | February 28, 2019 |
| Software Requirements Specification | Specification of the final requirements of the product, traceability to customer expectations and constraints, and verification plans for said requirements | March 26, 2019 |
| Software Project Management Plan | Definition of the project organization and proposed management of the project including planning for workflow, risk management, and technical management. | April 4. 2019 |
| Software Analysis Specification | A breakdown of the proposed product and its requirements decomposed into smaller functional units. A technical specification for proposed behavior and operation. Also a definition of acceptable software quality and human / environment interaction standards. | April 23, 2019 |
| Software Design Document | Final document for scope of this project | March 9, 2019 |

## 3. OVERALL SYSTEM OVERVIEW

The proposed product is not a standalone system but is dependent on external products and services to achieve its full functionality. Specifically, it is dependent on 3rd party food review APIs and services (i.e. Google Reviews and Yelp) for its restaurant information and review dataset as well as user’s input to help shape the service to specifically match each person’s individual preferences. The interfaces with external parties are further explored in the context diagram (FIGURE 1).

### 3.1 Context Diagram

FIGURE 1 shows the context diagram for the system. Specifically, the system itself (Feed Me) is included in the center of the diagram, while the various external entities and the means in which they interface with the system are shown, attached to the system with arrows. The direction of arrows shows the flow of action / information within this context.

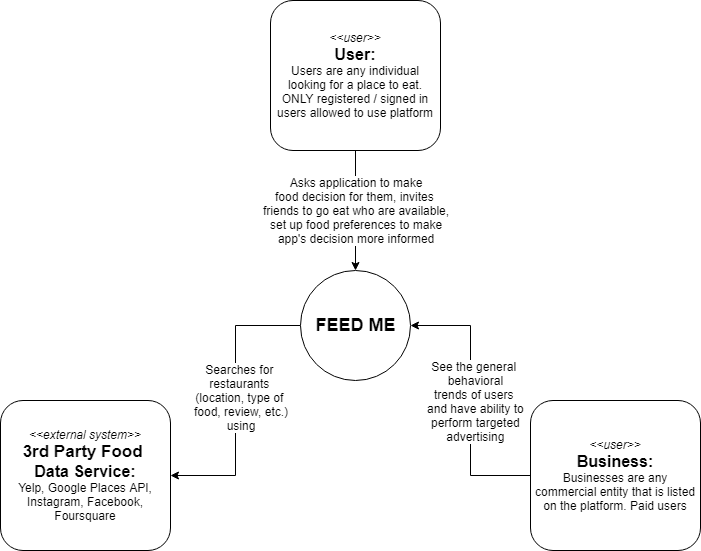


FIGURE 1:

Context Diagram

### 3.2 Additional Descriptive Items

#### 3.2.1 Product Functions

Table 2 describes the proposed functions for the product and a brief description of said functionality.

TABLE 2

Summary of Major Product Functions

|  |  |
| --- | --- |
| Function | Description |
| Account Management | The system will be allow users to create a persistent account with the application in order to save and transfer their settings / configuration to other devices. The users will also be allowed to set up their specific food / behavioral preferences in the app. |
| Food Decision Making | The system will decide for a user to decide where they should eat. It will make this decision intelligently based upon the user’s preferences, external data, location, etc. |
| Friending and inviting friends | The system will allow users to friend other users and invite friends out to eat together. |
| Business monetization and advertising | The system will allow businesses to advertise to targeted audiences within the userbase of the application. |

#### 3.2.2. User Characteristics

The primary users would be young, urban professionals and students whom, during customer research, were identified as being most likely to eat out regularly for their meals (rather than cooking). Furthermore, they are more likely to eat socially (i.e. with other individuals) and consult online services in determining where to eat. They are often technologically savvy and early adopters of new software products. A key factor in their use of technology is its incorporation into their daily routines without too much of their input. However, they are also reticent to share too much personal information on an internet based service (information privacy).

Secondary users for the system are travelers. Many tourists wish to immerse themselves in the locations to which they are travelling. Foremost to this immersion is food. However, many tourists find it hard to find a “local favorite” both representative of the local cuisine and compatible with their dietary preferences and their willingness to explore food options outside of their comfort zone. Already dependent on technology for their travel information and coordination, travelers would be open to a software which would assist in their decision making process. While technologically proficient, they would be unfamiliar physically with their destination’s environment and food, and therefore do need considerations taken to ensure that they are not led too far outside of their comfort zones.

Finally, another potential user of the product are businesses (restaurants). Restaurants often suffer from low lunch throughput, therefore being able to promote their businesses to a targeted audience which may result in new customer acquisition / sustained income would be useful to said users. Businesses are often represented by either an individual (or group) solely responsible for the marketing of said business or directly by their owners / operators. These individual are often price conscious, but open to services which provide unique advantages for growth.

#### 3.2.3 Project Constraints

The primary constraint for the project relates to the involvement of third party services to provide the data for the product. Specifically, the project developer must adhere to the specific policies and terms of use of said services such that using said data from these services does not place the project in legal responsibility for potential misuse. Further details on these specific constraints will become further available as the project matures and specific APIs are chosen for use in the product.

As the product will be collecting personal information on individual users, developers must adhere to data protection laws in the municipalities in which the product will be released. For example, the Calif. Bus. & Prof. Code §§ 22575-22578 requires commercial websites to post a privacy policy specifically outlining how the site uses its user’s information.

Finally, the project will be constrained by the deadline implemented by the primary project stakeholder, Professor Fred Strauss and the NYU Tandon School of Engineering Computer Science and Engineering Department, which provides a hardlines and budget for the project (further clarified in future releases of this document and future documents).

#### 3.2.4 Assumptions and Dependencies

The requirement specified in this document assume that datasets for 3rd party food review services are 1) public and 2) allowed to be commercialized by other businesses and services. Upon investigating specific services, if said datasets are not able to be referenced, the requirements for the project would change significantly as it affects the core functionality of the proposed product.

Another assumption made by the specification is that food preference can be predicted using a computer, mathematic model. While research has been done into the ability to make qualitative predictions based upon an input set using a computational model has been promising, it has been demonstrated that some relationships are too complex to be represented by a static model (i.e. a regression or a deep neural network). Therefore, if the decision relationship was unpredictable, it would greatly increase the feasibility of the project as well as its specified requirements.

## 4. DOCUMENT OVERVIEW

This document is the first version of the Systems Requirements Specification meant for the development and release of the first version of the Feed Me application. The purpose of the document is to give a detailed description of the requirements for the app. The remainder of this document includes nine sections and an appendix.

The next section describes the documents referenced during the creation of the SRS. Section 6 includes the requirements for the business development of the project (i.e. requirement no relating to the development of the software product). Section 7 outlines the functional and non-functional requirements for the product, and the use cases through which said requirements could be mapped onto. Section 8 outlines how the product will be tested and the requirements for that testing process. Section 9 outlines how this document will be reviewed to ensure adherence to quality standards. Section 10 traces the requirements specified in this document backwards and forwards in the project lifecycle. Section 11 describes how this document will develop as the system develops and the project is matured. Section 12 describes the rationale for the specified requirements. Finally, Section 13 includes any notes made by reviewers on the current document to take into consideration in future revisions. The last portion of the document are the appendices which include auxiliary tables related to the development of this document.

## 5. REFERENCE DOCUMENTS

Project Proposal for Feed Me, Version 1 (February 10, 2018): Theodore Kim, JinZhao Su, Petr Holoubeck.

## 6. BUSINESS REQUIREMENTS

### 6.1 Technology

The system’s business definition is driven by several key technologies. Foremost, the development of accurate and flexible computational models to represent (mathematically) complex relationships between qualitative and quantitative variables has made computer-based prediction and artificial intelligence technologically feasible (without the need for expensive hardware). Second with the advent of widespread usage of the internet as means of collecting and managing data, big data has provided a means of representing the physical world digitally. Various companies offer Application Programming Interfaces (APIs) that allow other 3rd party applications access to their datasets. These two technologies allow for a service that simply compiles and searches automatically through these various datasets intelligently.

The objective of the system is not to collect actual restaurant data, but rather provide a means of allowing users to efficiently navigate through the data provided to them through these other services to make a decision that is often better informed than a decision that they would make on their own. Hence, these drivers allow for said business model and the objectives of the system.

### 6.2 Economics

An emerging trend amongst data collecting and compilation businesses (i.e. Google, Microsoft, Facebook, etc.) is the availability of targeted advertisement. A B2B (Business-to-Business) venture providing a means of targeted advertisement is of high value to other businesses as targeted advertisement represents an effective means of reaching an audience likely to become customers. The high conversion rate of targeted advertisement is achieved by analyzing audience behaviors to match their consumption patterns to a specific business. Another trend related to targeted advertisement is the growing prevalence of “free software in exchange for data.” As in, consumers are becoming more open to the idea of having the free use of technology in exchange for the information that they divulge to potential advertisers about themselves.

The prevalence of these commodities and the advertisement business model enables the objectives and vision of the project and business.

### 6.3 Regulatory and Legal

Not applicable as of Version 2 of this document.

### 6.4 Market Considerations

The market of “intelligent technologies” is extremely popular currently, given the recent growth of the artificial intelligence and machine learning industries. The target market is younger individual who are open to using technology to meet their needs. Therefore, the willing market makes the business and project both achievable and potentially fruitful.

### 6.5 Risks and Alternatives

Table 3 represents the risks currently identified for the system, the probability of their occurrence, the mitigation strategy, the component of the system it affects, its potential impact and its current status.

TABLE 3

Identified System Risks

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Type | Risk | Probability | Discovery | Responsible Party | Mitigation Strategy | Impact | Status |
| **T** | System does not return an accurate decision (i.e. user does not like choice) | HIGH | Proposed during discussion of system concept | Software developers | The algorithm will learn from poor choices and improve itself | LOW | Identified |
| **O** | Developers do not have the knowledge to create the product | LOW | Proposed during discussion of project proposal | Software developers | The software team will divide up functional responsibilities early and research before development | HIGH | Being investi-gated |
| **Ec** | 3rd Party companies revoke use of their APIs because we are a business | LOW | Proposed during SRS Revision | Business Team | Prior to development, the project team will contact said companies and receive authorization to use their APIs (with credit) | HIGH | Identified |
| **Env** | The initial userbase does not utilize the application enough: not enough data is collected to make the app effective | HIGH | Proposed during revision of SRS | Business Team | Adequate customer research and canvasing is conducted to encourage early adoption of the product | HIGH | Identified |
| **T** | Assumptions made regarding existing technologies are incorrect (see 3.2.4) | LOW | Proposed during discussion of system concept | Software developers | Adequate research must be done to ensure that services exist to support application functionality | HIGH | Being investi-gated |
|  |  |  |  |  |  |  |  |

Types of risk are technological (T), environmental (Env), organizational (O), and economic (Ec).

### 6.6 Human Resources and Training

Not applicable as of Version 2 of this document

## 7. SPECIFIC REQUIREMENTS (FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS)

This section defines the functional and non-functional requirements for the system. It does so to a level of detail to allow developers to create a product that fulfills said requirements and that testers can use to verify that the final product did, in fact, meet those requirements. The included requirements are all assigned a unique identifier to facilitate traceability later in the project.

### 7.1 Functional Descriptive Detailed Requirements

The following are the functional requirements and specification for the system (identifiers precede the requirement with the prefix FSR, **F**unctional **S**ystem **R**equirement):

***FSR1) The system shall provide a means for users to define their individual food preferences.***

FSR1.1) The system shall give the user this option upon registration.

FSR1.2) The system shall give the user the option to change or update their preferences after initial registration.

FSR1.3) The system shall present this feature as a survey where each user will provide information as: a choice from a set of predefined answers, a quantitative value, or range of values.

FSR1.4) To ensure that the system makes accurate recommendations, users will be required to complete the survey when they register.

FSR1.5) The system shall use the user’s answers to further refine its decision-making process.

***FSR2) The system shall provide users with personalized recommendations for restaurants.***

FSR2.1) The system shall present this feature as a single action user interface element (i.e. a button).

FSR2.2) The system shall present this feature as the primary action within the product (i.e. presented to the user by default).

FSR2.3) The system shall provide five recommendations to the user.

FSR2.4) The system shall allow the user to decline or accept recommendations using a two action user interface element (i.e. two buttons, swiping, etc.).

FSR2.5) The system shall record the user’s action and use it to update its decision making algorithm.

FSR2.6) The system shall record the user’s location in order to make a more personalized decision based on nearby locations as well as track whether the user went to the restaurant.

FSR2.7) The system shall provide the same recommendations upon multiple uses of the feature (i.e. pressing the button multiple times within a small time period results in the same recommendations being generated).

***FSR3) The system shall interface with external APIs to gather data about restaurants compatible to each user.***

FSR3.1) The system shall automatically reference APIs when the users request a recommendation.

FSR3.2) The system shall utilize multiple APIs and accumulate the collected data into a single dataset.

***FSR4) The system shall allow users to maintain a public profile with the service.***

FSR4.1) The system shall allow users to create an account with the system.

FSR4.2) The system shall allow users to import account information from 3rd party platforms.

FSR4.3) The system shall enforce secure password policies when users create an account with the system.

FSR4.4) The system shall allow users to upload a profile picture.

FSR4.5) The system shall verify users with their email address or phone number.

FSR4.6) Users should be searchable by their profile’s username or email address.

FSR4.7) Users shall be allowed to log into their profile from multiple accounts.

***FSR5) The system shall allow users to add other users as friends.***

FSR5.1) The system shall allow users to search other users by their profile’s username or email address.

FSR5.2) The system shall allow users to send friend invitations to other users.

FSR5.3) The system shall allow users to accept or decline pending friend invitations.

FSR5.4) The system shall allow users to view the status of their friend invitations.

***FSR6) The system shall allow users to invite other users to eat out with them.***

FSR6.1) The system shall alert users with a notification when they are invited to eat out.

FSR6.2) The system shall check to see if other users are available to eat out by checking whether the invited user has already asked for a suggestion (implying that they had already eaten).

FSR6.3) The system shall recommend friends that are nearby to the user when the user asks for a food recommendations.

***FSR7) The system shall allow businesses to view generalized marketing data on the behavior of users using the product.***

FSR7.1) The system shall show businesses summary statistics on users that fit their business profile.

FSR7.2) The system shall allow businesses to create profiles for their business defining characteristics of the business (in the form of a survey).

FSR7.3) The system shall charge businesses to use its reporting services.

FSR7.4) The system shall not show data on individual users to businesses.

FSR7.5) The system shall only show relevant user data based on location, food preferences, budget, etc.

***FSR8) The system shall provide businesses a means of advertising on the platform to users within their target audience by offering special discounts and promotions to attract customers.***

FSR8.1) The system shall allow businesses to create custom advertisement listings to be shown to relevant users.

### 7.2 Requirements Use Cases

Nothing as of Version 1 of this document.

#### 7.2.1 Use Case Diagram

#### C:\Users\tgim9\AppData\Local\Packages\Microsoft.Office.Desktop_8wekyb3d8bbwe\AC\INetCache\Content.MSO\86292B82.tmp

FIGURE 2:

Use Case Diagram

#### 7.2.2 Use Case Descriptions

|  |  |  |
| --- | --- | --- |
| **User Registration** | | |
| **Description** | This use case provides a means for users new to the system to create a profile on the system (FSR4). | |
| **Pre-conditions** | The user performing this action does not already have an account on the service. | |
| **Flows** | Basic or Normal Flow | 1. User chooses option to create a profile on the system. 2. System redirects user to user registration screen 3. User provides email address, username, and password for new account. 4. System adds account information to database, sends verification email to provided email address. 5. User confirms email address. 6. System marks user account as verified. |
|  | Alternative Flows | 1. *Account exists at given email:*   If, in Step 3, the user provides an email address already associated with an account the system will prompt the user to use another email (repeat Step 3) or login with that email (User login Use Case)   1. *Password does not meet security policy*   If, in Step 3, the provided password does not meet the predefined security policy, the system will prompt the user to enter a new password (repeat Step 3)   1. *Missing Verification email address*   After Step 4, if the user does not receive the verification email, the system will provide the option to resend the email, or change the email address.   1. *Unverified Account*   If the user does not verify their email address, they will not be allowed to log into their account and the use case effectively ends. |
| **Post Conditions** | The user will be allowed to log into their account using the login use-case. | |
| **Special Requirements** | None | |
| **Extension Points** | None | |

|  |  |  |
| --- | --- | --- |
| **User Login** | | |
| **Description** | This use case provides a means for existing users on system to log into their saved profile (FSR4). | |
| **Pre-conditions** | The user performing this action already has an account on the service. | |
| **Flows** | Basic or Normal Flow | 1. Upon opening the app, the user is prompted to login 2. User provides email address and password for existing account. 3. System checks provided information against information in the database. 4. System informs user of successful login and redirects the user to the main system page. |
|  | Alternative Flows | 1. *User forgot their password:*   If, at Step 2, the user does not remember their password, the system will provide an option for the user to indicate such. An email will be sent to the provided email with a link to reset their password. If a account exists at that email, the user may click the link and set a new password for their profile.   1. *Account does not exist with given credentials.*   If, in Step 3, the provided email and password does not match an existing profile within the system, the system shall return an error code and require the user to repeat Step 2.   1. *First time login.*   After Step 4, if the user has never logged into the system before (i.e. it is a new account), the system will redirect the user to the Survey Use Case rather than the main landing page. |
| **Post Conditions** | After login, the user will have access to the full functionality of the system. | |
| **Special Requirements** | None | |
| **Extension Points** | At the first login prompt, the user can instead choose to sign up for a new account | |

|  |  |  |
| --- | --- | --- |
| **User Survey** | | |
| **Description** | This use case provides a means for users to complete a survey in which they can define their food preferences in order to personalize their recommendations (FSR1). | |
| **Pre-conditions** | The user performing this action must be logged into the system via the login use case. Furthermore, the user is automatically entered into this use case if they are new to the system or can manually enter into this use case later via an option within the system. | |
| **Flows** | Basic or Normal Flow | 1. The user is presented the survey page 2. The system generates questions and answer choices for the user based upon the factors which it uses to make its decisions. 3. The user answers each question (the user must provide an answer, questions cannot be skipped). 4. Upon completion of the survey, the user will be returned to the main landing page. |
|  | Alternative Flows | 1. *The user has already completed the survey:*   If, in Step 3, while the user is answering the questions. If the user had already answered the survey (i.e. the user had entered into the use case manually), the user will be presented the option to skip questions that they have already answered.   1. *The user does not complete the survey*   If, in Step 3, the user exits the system without completing the survey and there are unanswered questions left, the user will be prompted to enter said answers the next time they enter the system (i.e. they will re-enter the use case). |
| **Post Conditions** | The system will have updated the user’s preferences. | |
| **Special Requirements** | None | |
| **Extension Points** | None | |

|  |  |  |
| --- | --- | --- |
| **User Decision Making** | | |
| **Description** | This use case provides a means for users to query the system to generate restaurant recommendations for the user. | |
| **Pre-conditions** | The user performing this action must be logged into the system via the login use case. Furthermore, the user must have completed the survey use case at least one as well. | |
| **Flows** | Basic or Normal Flow | 1. The user is selected the option to initiate the decision-making process. 2. The system utilizes information gathered on the user from the survey and previous decisions, to generate a set of parameters for desirable recommendations. 3. The system queries external APIs with said parameters to generate five recommendations for the user. 4. The user is presented the recommendations and chooses whether to accept or reject each recommendation. 5. The system records the user’s actions and use it to redefine better decision making for the user for the next recommendation. |
|  | Alternative Flows | 1. *External APIs are not available:*   If, in Step 3, a portion of APIs utilized by the system are not available, the system will utilize the active APIs and inform the user that they decision may not be optimal. If no external APIs are available, the system returns an error to the user and ends the use case.   1. *No recommendation can be made*   If, in Step 3, the system cannot generate any recommendations for the user, random restaurants the search parameters will be relaxed until some restaurants are found.   1. *The user rejects all recommendations.*   After Step 4, if the user rejects all the recommendations, they are not allowed to ask for more recommendations, a sufficient amount of time (defined later) must pass between recommendation queries.   1. *The user is matched with an advertising business.*   If one of the user’s recommendations is a business that is advertising on the platform, the user will be shown that it is an advertisement and shown the text for the ad. |
| **Post Conditions** | The system will have | |
| **Special Requirements** | None | |
| **Extension Points** | When the user is shown the option to initiate the decision making feature, they will have the option to invite recommended friends to eat with them (thus altering the decision making process) | |

|  |  |  |
| --- | --- | --- |
| **User Adding Friends** | | |
| **Description** | This use case provides a means for users to add other users on the system as friends. | |
| **Pre-conditions** | The user performing this action must be logged into the system via the login use case. The user enters into the use case via a manual option within system. | |
| **Flows** | Basic or Normal Flow | 1. The user enters a user name or email address for the desired user into the search query box. 2. The system searches for users that match those parameters. 3. The user is presented a list of matching users. 4. The user chooses one of the users to request as a friend. |
|  | Alternative Flows | 1. *Location based search:*   If, in step 1, the user chooses to, they can search for users near them who have enabled location tracking for their account rather than providing a search string.   1. *No users are found that match the search query.*   If, in Step 2, the system cannot match any users within the system with the information provided, the system will return an empty list and allow the user to readjust they search query such that users are found.   1. *The invited user has already sent the current user a friend request.*   If, in Step 4, the user chooses to send a friend request with a user that has already sent them a friend request (i.e. if User A sends a request to User B, but User B already sent a request to User A) the two users are automatically made friends. |
| **Post Conditions** | The system will record the friend request and notify the other user of the pending invitation. | |
| **Special Requirements** | None | |
| **Extension Points** | None | |

|  |  |  |
| --- | --- | --- |
| **User Managing Friend Requests** | | |
| **Description** | This use case provides a means for users to view pending user friend requests and either decline or accept friend requests. | |
| **Pre-conditions** | The user performing this action must be logged into the system via the login use case. The user must have received a friend request from another user to be able to use this use case. The user can enter into this use case via an option within the system. | |
| **Flows** | Basic or Normal Flow | 1. The user managing their friend requests is resented with a list of pending invitations for the system. 2. The user can choose to either accept or reject each request. 3. The system will update the friends list of both the current user and the user that sent the request, after the current user chooses an action. 4. The user that sent the request will be notified of the current user’s decision. |
|  | Alternative Flows | 1. *No pending requests:*   If, in step 1, the current user has no pending requests, the user is presented an empty list. |
| **Post Conditions** | The system will have updated the user’s friend lists | |
| **Special Requirements** | None | |
| **Extension Points** | None | |

|  |  |  |
| --- | --- | --- |
| **User Inviting Friends to Eat** | | |
| **Description** | This use case provides a means for users to invite other users (that are accepted friends of the current user) to eat with the current user. | |
| **Pre-conditions** | The user performing this action must be logged into the system via the login use case. Furthermore, the user may only perform this action with another user that is a friend of the current user. This action can be entered manually as an extension of the decision making process. | |
| **Flows** | Basic or Normal Flow | 1. The user selects the option to invite friends 2. The system will generate a list of recommended friends based on location, if they already ate, etc. 3. The user selects a user from that list to invite to each 4. The target user is sent a notification. 5. The current user is returned to the list of users 6. The current user exits from the use case when it selects the UI option to exit the current screen. |
|  | Alternative Flows | 1. *User searches for non-recommended friends:*   If, in Step 2, the user wants to select a friend not on the recommended list, the app will allow the user to search for specific friends by username.   1. *The target user rejects the invitation*   If, in Step 4, the target user rejects the invitation, the current user is notified and the decision algorithm is altered. Otherwise, by default, the decision is made with the assumption that the invited user will attend. |
| **Post Conditions** | The system will use the invited friends to alter the decision maker algorithm. Following exit from this use case, the user is returned to the decision making use case. | |
| **Special Requirements** | None | |
| **Extension Points** | None | |

|  |  |  |
| --- | --- | --- |
| **Business Viewing User Summary Information** | | |
| **Description** | This use case provides a means for subscribed businesses to view summary statistics on relevant users. | |
| **Pre-conditions** | The business using this feature must have a normal, verified user account and be logged in. They can select this option under the business features within the system. | |
| **Flows** | Basic or Normal Flow | 1. The business is presented with the base dataset with filtering options. 2. The business can adjust the base dataset with the filtering options. 3. After applying the filtering options, the system prompts the business for payment of a microtransaction. Applying different filters require different amounts of microtransactions. 4. The system shows the dataset with the applied filters. |
|  | Alternative Flows | 1. *Business has already paid microtransactions:*   If, in Step 3, the business has already paid a microtransaction for the given set of filtering options, they will be shown the dataset with the given filters without needed to pay an additional fee.   1. *Microtransaction fails*   If, in Step 3, the microtransaction fails, the business must reenter payment details, otherwise the use case exits. |
| **Post Conditions** | The system will have updated the user’s preferences. | |
| **Special Requirements** | Businesses will need to pay a small fee to see information. Fees are scaling based upon the resolution of data (with the base dataset being free). | |
| **Extension Points** | This use case is extended by the adding advertisement use case. Upon filtering data, an option will be presented to create an advertisement towards that filtered audience, leading into the next use case | |

|  |  |  |
| --- | --- | --- |
| **Business Adding Advertisements** | | |
| **Description** | This use case provides a means for businesses to create advertisements towards specific user audience groups. | |
| **Pre-conditions** | The business must have first made a normal, verified user account and be logged in. Next, they must have paid for a filtered user summary from the previous use case and selected the option to create an advertisement for that audience. | |
| **Flows** | Basic or Normal Flow | 1. The business is presented the summary statistics for the user audience they are targeting as well as a place to enter their business details and advertisement text. 2. The business enters their information. 3. The system presents the business with a summary / preview of their advertisements. 4. The business enters billing information to be billed a small fee every time the ad is shown to a user. 5. Upon transaction completed the business is sent a receipt and the system adds the ad to the invoice. |
|  | Alternative Flows | 1. *Billing information fails:*   If, in Step 4, the provided billing information is incorrect, the business is prompted to reenter their information, or the use case exits.   1. *Cancels the transaction*   At any point, the business can cancel the transaction and return to the previous use case screen.   1. *Existing ad*   If, at Step 2, the business information is already associated with an ad given the selected audience profile, the business will be prompted to “change” the currently running ad rather than create a new ad, or cancel the current ad. |
| **Post Conditions** | The system will contain the new ad to be shown to users | |
| **Special Requirements** | None | |
| **Extension Points** | None | |

### 7.3 Non-Functional Descriptive Detailed Requirements

The following are definitions and specifications of the non-function requirements for the system. The non-function requirements are divided in several categories (as indications following the requirement identifier). Each requirement has a unique identifier (NSR#, **N**on-Functional **S**ystem **R**equirement).

**NSR1 (Usability)**: The system shall have an average learning curve of less than ten minutes (i.e. new users shall be able to use the system fully in less than ten minutes)

**NSR2 (Usability)**: The system shall provide a short tutorial to new users on how to use the system (with the option to skip said tutorial).

**NSR3 (Reliability)**: The system shall be deployed onto a service which can provide the user’s a consistent uptime (24 hours a day, 7 days a week, 365 days a year).

**NSR4 (Reliability)**: The system shall remain up during system upgrades with a notice of possible slower response to users.

**NSR5 (Performance)**: The system shall provide recommendations that are accepted by users at least 75% of the time (i.e. 75% of recommendations made to users will be accepted).’

**NSR6 (Performance)**: The system shall allow for up to twenty users to make decision requests simultaneously.

**NSR7 (Supportability)**: The system shall have the ability to scale capacity upwards or downwards in the future depending on adoption of the system.

**NSR8 (Human Interface)**: The system shall have a minimalist design.

**NSR9 (Packaging)**: The system shall be usable to all major operating systems (mobile, PC, UNIX, etc.)

## 8. SYSTEM TEST PLAN REQUIREMENTS

The test plan’s purpose is to ensure that the final system conforms to the requirements previously defined so that it meets the needs and expectations of the customers and stakeholders. The most effective way of doing so would be to test the product within a closed group of testers who belong to the target audience / user-base. Therefore, the requirement for the testing of the final system should include the following components:

* The testers must belong to the identified user groups from and exhibit the characteristics described in Section 3.2.2. This is to ensure that the testing closely resembles the wild usage of the product after release.
* The testers must be allowed full use of the platform in its production state and the developers / testers should not interfere with their usage of the platform. This requirement includes the prohibition of developers from being able to explain the platform to the testers as they would not have said interaction available after the system is deployed to actual users.

The requirements will be delivered to SQA who will create a test plan including test scenarios and expected output, then they will execute the tests and report defects.

## 9. QUALIFICATION PROVISIONS

Four types of testing will be performed: a desk check, a peer review, a walkthrough and an inspection. The first method of ensuring the quality of this artifact, a desk check, essentially involves the writers of the document proofreading their work such that they identify errors they made when drafting the artifact such as typos, obscurities and confused wording. A peer review involves team members reviewing each other’s work to identify possible defects and errors made during the drafting process. A peer review allows for more individuals to review the work and helps to improve the clarity of the artifact as individuals evaluating their own work tend not to see their own ambiguities as they are familiar with their own thought process.

Walkthroughs involve a group of reviewers having the artifact author “walkthrough” their work in person in front of the reviewers (in this case the rest of the project team and the SQA) so that the reviewers can check that the artifact adequately conveys the author’s intent. Finally, inspections involve submitting the artifact to an independent review panel for review. In the case of this document, it is submitted to the SQA and project authority (Professor Fred Strauss) for review.

## 10. REQUIREMENTS TRACEABILITY

The requirements specified in this document are derived from the expectations and preliminary requirements outlined in the Project Proposal document.

The requirements are each numbered with a unique identifier such that, later developments of the project baseline, can reference specific requirements. Using these and future identifiers, traceability can be maintained in future phases and iterations of the project. This traceability ensures that when a defect is found, it can be traced back to where it was injected so that identifying what changes need to be made is easier.

## 11. EVOLUTION OF THE SRS

Initially, this document outlines the overview of the proposed system, the context in which the system will operate and how that context will influence the creation of the system requirements, the scope of the document, and the requirements for the business (non-technical) aspect of the project. Before the submission of the document, the authors of the document will review it for thoroughness and correctness of its portrayal of the system. The initial state of the document will then be submitted for review by the project evaluators in order to identify weaknesses and deficiencies. The returned criticisms will be incorporated into the document.

The next stage of development will add the specific technical, functional and non-functional requirements for the proposed system. Furthermore, the possible use cases for the system will be specified. The review process will be repeated and defects will be improved to ensure that the document is an accurate specification of the requirements for the final system.

## 12. RATIONALE

Nothing as of Version 1 of this document.

## 13. NOTES

Nothing as of Version 1 of this document.

## 14. APPENDICES

### 14.1 schedule Tracking

TABLE 14.1-1

Individual Artifact Schedule Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or team) | Estimated | Actual | Difference |
| Initial SRS | Theodore Kim | 5 hours | 6 hours | +1 hour |
|  | JinZhao Su | 3 hours | 2 hours | -1 hour |
|  | Petr Holoubeck | 3 hours | 3.5 hours | +0.5 hours |
|  | Entire Team | 11 hours | 11.5 hours | +0.5 hours |
| Final SRS | Theodore Kim | 5 hours | 4 hours | -1 hour |
|  | JinZhao Su | 2 hours | 2 hours | 0 hours |
|  | Petr Holoubeck | 2 hours | 4 hours | +2 hours |
|  | Entire Team | 9 hours | 10 hours | +1 hour |

TABLE 14.1-2

Cumulative Schedule Tracking

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or team) | Estimated | Actual | Difference |
| Theodore Kim | 10 hours | 10 hours | 0 hours |
| JinZhao Su | 5 hours | 7.5 hours | +2.5 hours |
| Petr Holoubeck | 5 hours | 4 hours | -1 hour |
| Entire Team | 20 hours | 21.5 hours | +1.5 hours |

### 14.2 Defect Tracking

TABLE 14.1-1

Individual Artifact Defect Tracking

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Artifact or Deliverable | Who (individual or team) | Estimated | Actual | Difference |
| Initial SRS | Theodore Kim | 3 defects | 5 defects | +2 defects |
|  | JinZhao Su | 4 defects | 3 defects | -1 defects |
|  | Petr Holoubeck | 3 defects | 2 defects | -1 defects |
|  | Entire Team | 10 defects | 10 defects | 0 defects |
| Final SRS | Theodore Kim | 5 defects |  |  |
|  | JinZhao Su | 3 defects |  |  |
|  | Petr Holoubeck | 6 defects |  |  |
|  | Entire Team | 14 defects |  |  |

TABLE 14.1-2

Cumulative Schedule Defect Tracking

|  |  |  |  |
| --- | --- | --- | --- |
| Who (individual or team) | Estimated | Actual | Difference |
| Theodore Kim | 3 defects | 5 defects | +2 defects |
| JinZhao Su | 4 defects | 3 defects | -1 defect |
| Petr Holoubeck | 3 defects | 2 defects | -1 defect |
| Entire Team | 10 defects | 10 defects | 0 defects |