# SRS - Nick

## Use Case Diagram - Nick

## Use Case Descriptions – Nick

|  |  |  |
| --- | --- | --- |
| **Project** |  | |
| **Use Case** | **Tag New Meal** | |
| **System** |  | |
| **Actors** | **User** | |
| **Goal**  <a longer statement of the goal in context if needed> | To Create a meal | |
| **Trigger**  <the action upon the system that starts use case> | The user selects Tag New Meal | |
| **Preconditions**  <what we expect is already the state of the world> | 1. The user has sufficient storage 2. The user has a phone with a functional camera 3. The user is in front of a meal | |
| **Success End Condition**  <the state of the world upon successful completion> | A new meal is created | |
| **Failed End Condition**  <the state of the world if goal abandoned> | A meal is not fully created or is aborted before completion | |
| **Primary Actors;**  **Secondary Actors** | User | |
| **Description / Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean up after. Indicate sub steps using numbering> | **Step** | **Action** |
| 1.a | App is open |
| 1.b | User selects Tag new meal |
|  |  |
|  | |
|  |  |
| **Alternative Flows**  <a: condition causing branching>  <a1: action or name of sub use case> | **Step** |  |
| 1 | **Branching** |
|  | A1 | Camera application opens |
|  |  | Image is taken |
|  | A2 | **Branching** |
|  |  | Image is saved |
|  |  | Image is tagged by user |
|  | 1a | Tagged image is saved |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Project** |  | |
| **Use Case** | **View Local Meals** | |
| **System** |  | |
| **Actors** | **User** | |
| **Goal**  <a longer statement of the goal in context if needed> | User is able to view their locally saved meals | |
| **Trigger**  <the action upon the system that starts use case> | The user selects View Meals | |
| **Preconditions**  <what we expect is already the state of the world> | 1. The user has 1 or more locally saved meal(s) | |
| **Success End Condition**  <the state of the world upon successful completion> | The user views locally saved meals on their mobile | |
| **Failed End Condition**  <the state of the world if goal abandoned> | The user is unable to view locally saved meals | |
| **Primary Actors;**  **Secondary Actors** | User | |
| **Description / Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean up after. Indicate sub steps using numbering> | **Step** | **Action** |
| 1.a | The User selects View Meals |
| 1.b | The user then views the saved meal |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Alternative Flows**  <a: condition causing branching>  <a1: action or name of sub use case> | **Step** | **Branching** |
| a | The user select edit meal |
| A1 | The user then uploads and saves to web database |
| A1.1 | The application gets the geolocation for the image and ads it to the tags |
| A2 | The user generates the ORAC rating for the meal selected |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Project** |  | |
| **Use Case** | **Searches Meals** | |
| **System** |  | |
| **Actors** | **Application** | |
| **Goal**  <a longer statement of the goal in context if needed> | The user searches the web database for according to User selected tags | |
| **Trigger**  <the action upon the system that starts use case> | The user selects Search Meal | |
| **Preconditions**  <what we expect is already the state of the world> | 1. The mobile must have internet connection 2. The user must have selected to let the application use mobile data | |
| **Success End Condition**  <the state of the world upon successful completion> | A list of meals, along with individual meals is viewed by the user, according to their preferences. | |
| **Failed End Condition**  <the state of the world if goal abandoned> | No meals are viewed by the user | |
| **Primary Actors;**  **Secondary Actors** | Application  User | |
| **Description / Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean up after. Indicate sub steps using numbering> | **Step** | **Action** |
| 1.a | The photo has been taken |
| 1.b | User selects the main ingredients of their meal from a menu |
| 1.c | An ORAC rating can then be generated |
|  |  |
|  |  |
|  |  |
|  |  |
| **Alternative Flows**  <a: condition causing branching>  <a1: action or name of sub use case> | **Step** |  |
| 1.1a  1.1b | The user selects to retake the image and returns to previous screen  The user exits the application |

|  |  |  |
| --- | --- | --- |
| **Project** |  | |
| **Use Case** | **Generate Shopping list** | |
| **System** |  | |
| **Actors** | **User** | |
| **Goal**  <a longer statement of the goal in context if needed> | To generate a shopping list from USERs saved meals | |
| **Trigger**  <the action upon the system that starts use case> | A shopping list button is clicked | |
| **Preconditions**  <what we expect is already the state of the world> | 1. User has 1 or more saved meals 2. User selects generate shopping list button | |
| **Success End Condition**  <the state of the world upon successful completion> | A completed shopping list is generated for the user | |
| **Failed End Condition**  <the state of the world if goal abandoned> | An incomplete shopping list is generated | |
| **Primary Actors;**  **Secondary Actors** | User  Application | |
| **Description / Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean up after. Indicate sub steps using numbering> | **Step** | **Action** |
| 1.a | The user clicks the generate shopping list button |
| 1.b | The use then selects the meals they want to be included |
| 1.c | The application generates a list of ingredients from the tags |
| 1.d | The application then allows the user to export and save the shopping list |
| 1.e |  |
| 1.f |  |
|  |  |
| **Alternative Flows**  <a: condition causing branching>  <a1: action or name of sub use case> | **Step** |  |
| a  a1 | The user selects new or deselects meals to add to the shopping list  The application then re-generates the shopping list based on the new selection |

|  |  |  |
| --- | --- | --- |
| **Project** |  | |
| **Use Case** | **Search uploaded meals** | |
| **System** |  | |
| **Actors** | **User** | |
| **Goal**  <a longer statement of the goal in context if needed> | To allow user to look at other meals that interest them | |
| **Trigger**  <the action upon the system that starts use case> | User selects to search the uploaded meals | |
| **Preconditions**  <what we expect is already the state of the world> | 1. User has internet connection 2. User has allowed app to use mobile data | |
| **Success End Condition**  <the state of the world upon successful completion> | A completed shopping list is generated for the user | |
| **Failed End Condition**  <the state of the world if goal abandoned> | An incomplete shopping list is generated | |
| **Primary Actors;**  **Secondary Actors** | User  Application | |
| **Description / Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean up after. Indicate sub steps using numbering> | **Step** | **Action** |
| 1.a | The user clicks the generate shopping list button |
| 1.b | The use then selects the meals they want to be included |
| 1.c | The application generates a list of ingredients from the tags |
| 1.d | The application then allows the user to export and save the shopping list |
| 1.e |  |
| 1.f |  |
|  |  |
| **Alternative Flows**  <a: condition causing branching>  <a1: action or name of sub use case> | **Step** |  |
| a  a1 | The user selects new or deselects meals to add to the shopping list  The application then re-generates the shopping list based on the new selection |

## System Design

### System Architecture - Dan

### Storage/Persistent Data Strategy - Dan

### Trade-offs and Choices - Dan

### Concurrent Processes - Dan

### Package Diagram – James, Jacob

## User Interface Layouts – Nick

Appendix?

## Program Navigation Diagram - Nick

## Data Definitions - Dan

## Analysis and Design Class Diagram – James, Jacob

## Sequence Diagram – James, Jacob

## State Diagram – James, Jacob

## Requirements Traceability Matrix – Nick

## Design Assumptions - Dan