

Comprehensive Requirements Analysis Template

1. Project Overview

Provide a high-level overview of the project, its purpose, and key stakeholders.

1.1 Project Name

McDonalds Calorie Counter (McCounter)

1.2 Project Purpose

Application to search for calorie counts of each item on the McDonalds Menu.

1.3 Stakeholders

- McDonalds
- Company Employees
- Shareholders
- Marketing Team
- Developers

2. Functional Requirements

Detail the specific functions and features the system must perform.

2.1 User Stories/Use Cases

1. As a Customer, I want to filter items by Calorie Count and Allergens so that I can easily find items that fit my needs.
2. As a McDonalds Employee, I want to provide customers with information about allergens and nutritional content, so they can make informed decisions and avoid any health issues.
3. As a Shareholder, I want to be able to use data collected by the App to help identify customer preferences. This will allow me to help me know when to help future developments in the company.

4. As a member of the Marketing Team, I want to know what items on our menu are both full of 'contents', and also the least unhealthy, in order to promote a health-positive brand.
5. As a Developer, I would like to access documentation on API's used, and the project, easily so that I can continue to build features after the initial release of the software. I also want a simple way to report bugs, so that the project aligns with customer expectations.

2.2 System Features

- Search Feature to search for specific items
 - Must be detailed
 - Must originate from a CSV file
- CSV File to hold data on items
- Business-related features
 - Data Display
 - Item Search
 - May as well also let them modify data

2.3 Data Requirements

User Info

- Nickname | String

Used to personalise the users experience, making it more 'friendly;', and seemingly like a conversation, rather than a chore. Collected and used only if user signs up to program prior, and scans there McDonalds App Code (or is signed into app if on app)
- Previous Orders | Dictionary

Used to personalise the users experience, by creating a 'recommended' area, based on users previous & most brought items. This will be collected the same way as the McDonalds app does, by using the users code and logging orders. This data will be kept local to the machine for security purposes.
- Food Items | Dictionary

Used throughout the whole application. The data will collect both allergen, and calorie based data, and will also factor popularity. For example, a items 'attributes' could be:

```
menu = {  
  "Big Mac": {  
    "price": 3.99, # Price in USD  
    "calories": 550, # Calories in kcal  
    "allergens": ["Wheat", "Sesame", "Egg", "Mustard", "Milk"], # List of allergens contained  
    "popularity": 5598123 # Number of units sold this year  
  }  
}
```

-

3. Non-Functional Requirements

Specify the quality attributes and constraints of the system.

3.1 Performance

- Response Time: Less than 3 seconds per request
- Throughput: Up to 20 food to be processed per request (for main menu). Food items not required (excl. prices, and simple calorie count.)

Capacity: Up to 100 foods including 7 different variables per parent

3.2 Security

- Authentication: Token &/or Cookie authentication (Once signed in, a token will be stored in the browsers cookies which will allow the site to instantly recognise them, the token will expire after a small period of time.

Authorization: 2 Factor Authentication (Using either a 2FA Application, or the built in app. Similar to Facebook 2FA Functionality)

Data Protection: User information should be encrypted and held locally, excluding information required to be held by law.

3.3 Usability

- User Interface: Simplistic but readable design
- Accessibility: High contrast site, easily readable. Reading mode functionality.
- User Documentation: Not Applicable

3.4 Reliability

- Availability: 24/7
- Fault Tolerance: The system should be built to allow errors, and show a small error message to the user. It should also (by default) send the error message to a developer via API request

Recoverability: At least 2 24/7 running servers, using RAID 1.

4. Constraints

List any limitations or restrictions that may impact the project.

4.1 Technical Constraints

- There should also be offline compatibility to a certain extent, as it's often hard to gain wifi at McDonalds.
- We currently do not have access to a hosting server.

4.2 Regulatory Constraints

Data on food that is shown should be kept up to date using regular re-reviews, or if existing, using existing API's utilised by the current McDonalds APP Menu.

5. Assumptions and Dependencies

Document any assumptions made and dependencies on other systems or projects.

Assumption 1: The user knows how to use the device of choice to understand functionality of buttons and pictures

Dependency 1: A working device with web functionality

Dependency 2: An active internet connection

Dependency 3: A server to process requests, and store up-to-date information.

6. Acceptance Criteria

Define the conditions that must be met for the project to be considered successful.

1. Data must be accessible from a CSV File (whether via server or client)
2. Business Data Access
3. UI Moodboard & stylesheet
4. Flowchart of site logic
5. Wireframe for site
6. Graphs (Additional)

7. Sign-off

Provide space for key stakeholders to approve the requirements document.

Approved by:

