

Software Requirements

Sandwich King

Company: Sandwich King

Created By: Garron Blake

1. Introduction

1.1 Purpose

Set the expectations for the outcome of the product. The product is expected to be a well-designed, intuitive, responsive, and aesthetically-pleasing website.

1.2 Intended Audience

Anyone who likes sandwiches is the intended audience.

1.3 Intended Use

The website is intended to be an image gallery platform where a user can anonymously post pictures of their sandwiches to be seen by all users. This website can be useful for those who wish to put all their pictures of their sandwiches in one place and receive validation from strangers.

1.4 Scope

The main goal is to create a functioning website that serves images of other people's sandwiches and provides a means of uploading additional sandwich images. It would be ideal to be able to process as many image formats as possible, as long as the security integrity of the website is not threatened by possible exploits involving certain file formats. Because humans think they're a comedian and tend to do dumb things on the Internet, it should be expected for the website to have a way for trusted administrators/moderators to monitor the content that is uploaded by ordinary users and delete images of anything that is **not** a sandwich.

1.5 Definitions and Acronyms

- *Sandwich*: an image of a sandwich.
- *Hub page*: the default page (index) users are sent to when visiting the site, showing all the latest Sandwiches in a scrollable grid-like format.
- *My Sandwiches page*: the page that shows the user their entire upload history and allows the user to upload a new Sandwich, edit a previously uploaded Sandwich, and delete a Sandwich.

2. Overall Description

The product being built is not for a particular client because the primary purpose of building this product is for the creator (Garron) to have something to show to potential employers and to prove his skillset.

2.1 User Needs

- The user needs validation from strangers.
- The user needs to all of their Sandwiches in one place other than their camera roll.
- The user needs to see Sandwiches for their own personal enjoyment.

2.2 Assumptions and Dependencies

What assumptions are you making that could cause an error? Is the project reliant on any other factors that could affect the development of the software?

Assumptions:

- The frameworks, packages, and UI libraries I am using will not become obsolete/unusable in the foreseeable future.
- The UI libraries I am using will not clash with one another in a disruptive, unintended manner.

Dependencies:

- ASP.NET Core 6
- Entity Framework Core 6
 - MS SQL Server

- Razor
- Blazor (server app)
- CSharp language
- HTML language
- Mudblazor
- SyncfusionBlazor

3. System Features and Requirements

3.1 Functional Requirements

Define the functional requirements that are essential for the software.

- The website shall have a hub page to serve as the primary channel of content consumption by users.
- The website shall have an upload page ([My Sandwiches](#)) to provide the user the following three services:
 - view all of the user's previously-uploaded Sandwiches
 - edit the hashtags of previously-uploaded Sandwiches
 - attribute any applicable recipes to previously-uploaded Sandwiches
 - attribute any applicable sandwich shops to previously-uploaded Sandwiches
 - delete a Sandwich
- The My Sandwiches page shall show only the Sandwiches uploaded by the logged-in user. If the user is not logged in, the My Sandwiches page shall show no Sandwiches and indicate that the user is not logged in.
- The hub page shall have a search bar for the user to filter the list of Sandwiches by one or more hashtags. The user shall also be able to choose exclusive-and filtering if desired.
- Sandwiches shall be able to be up-voted or down-voted and shall display their score to every user that views it.
- With the exception of admins, the username of the account that uploaded the Sandwich shall be completely hidden from all other users. The admins shall be given the option to inspect the user's list of Sandwiches and perform any necessary action.

- Sandwiches shall be reportable for abuse by any user and the reports shall be easily accessed by the admins.
- To maximize performance, the images themselves shall be stored at a specific folder in the server, unreachable by any URL, and only the relevant identifiable information about the image shall be stored in a database.

3.2 External Interface Requirements

Are there any UX and UI requirements that you must keep in mind as you build?

- The website's UI shall look clean, elegant, and simple to comprehend.
- The Sandwiches shall be previewed to the user via a "card," and the hub page shall show these cards in a grid format.
- The upvote system shall be integrated in each card that generates.
- The username of the author of the Sandwich shall not be visible to any other user.

3.3 System Features

What features are required for the software to work?

- SyncfusionBlazor components (file upload and toast notifications)

3.4 Nonfunctional Requirements

Are there any non-functional requirements that you need to address (i.e. budget, team, etc.)

- Ideally, the development budget should be \$0 and there should only be one developer.