Lab 2 – TasteBuddies Product Specification

Grant Fitch

Old Dominion University

CS 411W

Professor Sarah Hosni

March 27, 2025

Version 2

Table of Contents

Table of Figures

Table of Tables

1 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functional and non-functional requirements for the TasteBuddies system. This document serves as a technical reference for developers and provides the necessary details for implementing the system. It includes a structured overview of the system's architecture, product functions, user characteristics, constraints, and dependencies to guide development.

1.2 Scope

TasteBuddies is a food recommendation and social media platform designed to help users discover dishes and restaurants based on their individual taste preferences. The system allows users to create a taste profile, collecting preferences across key categories such as flavors, cuisines, and allergies. Using this profile, the system provides personalized dish recommendations tailored to individual users.

In addition to recommendations, the platform enables users to find and connect with others who share similar taste preferences. This allows for a curated review system where users see restaurant reviews from others with similar preferences. Users will also be able to find and follow their friends and other individuals, allowing the platform to offer a group dining option as well. This option will compare the taste profile and preferences of the users in an assigned group in order to offer recommendations that cater to the group as a whole.

The system will allow users to create and update their taste profiles, which serve as the foundation for personalized dish recommendations. Users will be able to search for restaurants and specific dishes that align with their individual preferences, as well as save favorite dishes and restaurants for future reference. The platform will also support social features such as

searching for and following friends or other users with similar preferences. In addition, TasteBuddies will offer group dining functionality by comparing the taste profiles of multiple users and generating recommendations that cater to the group as a whole.

The system is not intended to function as a generic restaurant review site, nor will it store or track sensitive health data beyond basic dietary preferences. Additionally, the system does not facilitate food delivery or pickup, focusing solely on personalized dining discovery and social connection.

[This page intentionally left blank.]

1.3 Definitions, Acronyms, and Abbreviations

Taste Profile: Personalized profiles created by each user based on their taste preferences, such as preferences for spicy, sweet, salty, etc.

Taste Buddy: Users with highly similar taste profiles which lead to improved recommendations based on aligned tastes.

Personalized Dish Recommendation: A dish suggestion tailored to a user base on their taste profile and feedback from other users.

MFCD: Major Functional Component Diagram, visual illustration of the system architecture.

User Profile: A user's personal data, including name, email, preferences, and saved dishes.

Python: The programing language used to develop the TasteBuddies system.

Flask: A lightweight web framework for Python, used for the TasteBuddies backend and frontend development.

Blueprints: A modular system in the Flask framework used for organizing routes and functionality.

Digital Ocean: A cloud hosting service where the system is deployed.

SQlite: A lightweight database used for local data storage.

Jinja2: A template-based engine for developing frontend, used in Flask with Python.

SQLAlchemy: An Object-Relational Mapper for Python with Flask that provides a high-level abstraction for database queries.

1.4 References

[1] "2024 Consumer Dining Trends: How Americans Are Spending on Restaurants and Takeout," Auguste Escoffier School of Culinary Arts, https://www.escoffier.edu/blog/world-food-drink/consumer-dining-trend-statistics/ (accessed Mar. 27, 2025).

[2] "Culinary Confessions: Cooking Habits of Gen Z & Millennials," *Home Run Inn Pizza,* https://www.homeruninnpizza.com/news-blog/fun-facts/cooking-habits-gen-z-millennials/ (accessed Mar. 27, 2025)

[3] G. Fitch, "Lab 1 – TasteBuddies Product Description," *CS 411, Old Dominion University,* 2025.

[4] IEEE Computer Society, *IEEE Recommended Practice for Software Requirements Specifications (IEEE 830-1998),* IEEE, 1998.

[5] "Jinja Templating Engine," *Pallets Projects*, https://jinja.palletsprojects.com/ (accessed Mar. 27, 2025)

[6] K. Vaghasiya, "15 Fake Review Statistics You Can't Ignore (2024)," *WiserNotify*, https://wisernotify.com/blog/fake-review-stats/#combating-fake-reviews:-strategies-and-tools (accessed Mar. 27, 2025)

[7] N.-G. Wunsch, "Average annual food away from home expenditures of United States households from 2010 to 2022," *Statista*, https://www.statista.com/statistics/237215/average-away-from-home-food-expenditures-of-united-states-households/ (accessed: Mar 27, 2025).

[8] "Social Eating Connects Communities," University of Oxford, https://www.ox.ac.uk/news/2017-03-16-social-eating-connects-communities/ (accessed: Mar. 27, 2025).

[9] "SQLAlchemy ORM and Core," *SQLAlchemy Foundation*, https://www.sqlalchemy.org/ (accessed Mar. 27, 2025).

[10] T. Perkins, "Your food is more expensive – are US corporate profits to blame?," *The Gaurdian*, https://www.theguardian.com/environment/article/2024/jul/26/food-price-inflation-corporate-profit (accessed Mar. 27, 2025).

[11] "U.S. Travel and Tourism Overview (2019)," *U.S. Travel Association*, https://www.ustravel.org/system/files/media_root/document/Research_Fact-Sheet_US-Travel-and-Tourism-Overview.pdf (accessed Mar. 27, 2025).

[12] "Welcome to Flask – Flask Documentation (3.1.x)." *Pallets Projects*, https://flask.palletsprojects.com/ (accessed Mar. 27, 2025).

[This page intentionally left blank.]

1.5 Overview

The remainder of this document provides a high-level description of the TasteBuddies system, outlining its purpose, functionality, and design. Section 2 describes the system architecture and major components, including how the system processes user data and generates recommendations. It also outlines key product functions, detailing how users interact with the platform. The section then defines the characteristics of TasteBuddies users, followed by an overview of potential system constraints and assumptions or dependencies that may impact development. Section 3 outlines the external interface requirements, describing how the system interacts with the user, hardware, and software components.

2 Overall Description

2.1 Product Perspective

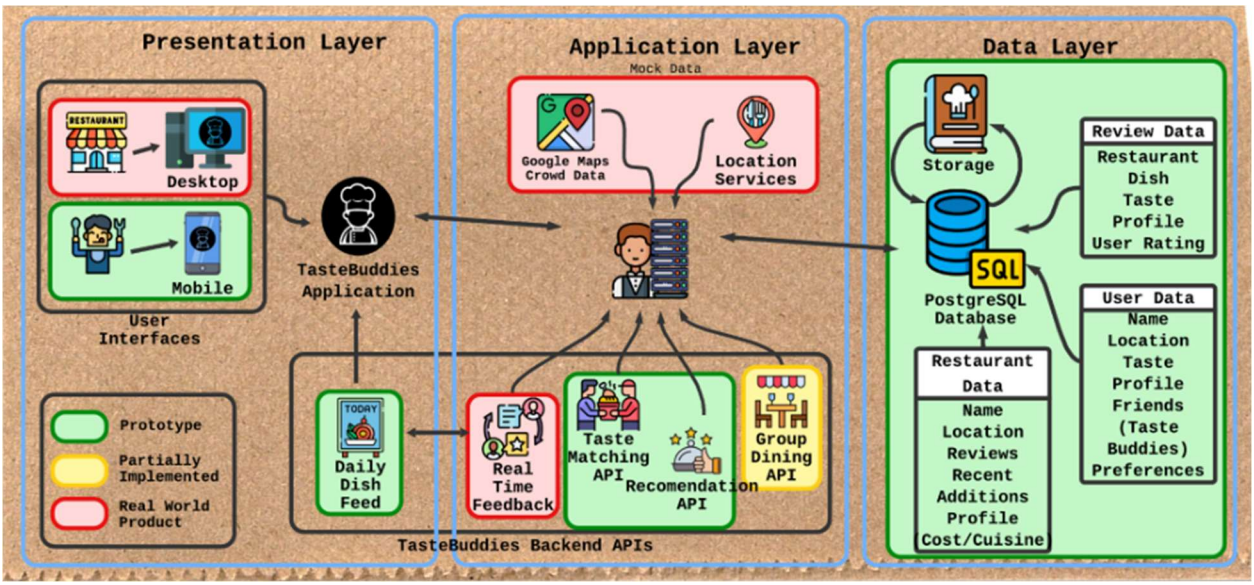
TasteBuddies is a web-based application designed to help users discover personalized food recommendations based on their taste preferences. The system is built using a three-tier architecture consisting of a presentation layer for the user interface, an application layer for managing user inputs, authentication and generating recommendations, and a data layer for storing all the user, restaurant, dish, review and recommendation information. This structure can be seen in Figure 1 that follows.

The system is structured using Flask Blueprints to help organize different functionalities. User management enables account creation and authentication as well as taste profile creation. The Taste Matching and Recommendation APIs analyze these preferences to connect users and suggest dishes based on these preferences. The Daily Dish Feed is used to display reviews from other users with similar taste preferences, as well as the dishes and restaurants that the

Recommendation API has matched to the current user. This system operates as a standalone web application, relying on internal data stored within our SQLite Database. The deployment of this app is managed through Digital Ocean, ensuring accessibility and performance.

Figure 1

Major Functional Components Diagram



2.2 Product Functions

TasteBuddies provides user authentication, taste profile management, dish recommendations, review filtering, social connections, and group dining recommendations. Users can create an account, define their taste preferences, and receive personalized dish recommendations. Based on these preferences, the system connects users with others who share similar tastes. Additionally, users can find and follow other users, save favorite dishes, and leave reviews.

A visualization of the comparison for the features that will be available in the Real World Product as compared to the current Prototype being developed can be seen in Table 1.

Table 1

Features Table

Category	Features	RWP	Prototype	Additional Notes
Account Management	Account Creation			
	Login / Authentication		Eliminated	
	Access Permissions and Preferences		Partially Implemented	Access Permissions required for database
	Taste Profile			
Mobile App Features	Social Engagement		Partially Implemented	Find friends only for group matching
	Daily Dish Feed			
	Group Restaurant Matching		Partially Implemented	Implementation is time dependent
	Dish Recommendations			
	Taste Profile Builder			
	Reviews		Partially implemented	Mock data for compatibility matching
	Community Updates		Eliminated	
	Dish Validation		Eliminated	
	Taste Matching			
	Notification Features		Eliminated	
	Engagement Features		Eliminated	
DataBase Management	Data Analytics		Eliminated	
	Data Privacy and Security			
	Trend Reports		Eliminated	
	Data Backups			
Social Engagement	TasteBuddies			
	Super TasteBuddies		Partially implemented	Hard coded
	Add/Find Buddies			
	Follow TasteBuddy		Eliminated	
	Follow Restaurant		Eliminated	
	Add Kudos		Eliminated	
Live Interactive Updates	Daily Dish feed			
	Add reviews		Partially implemented	Mock data provided
	Post restaurant update		Eliminated	
	Post dish update		Eliminated	
	Notifications		Eliminated	
Recommend-ation Engine	Taste Profile			
	Read Reviews		Partially Implemented	
	Taste Matching			
	Dish Recommendation			
	Group Restaurant Matching		Partially Implemented	Implementation time dependent
	Rewards		Eliminated	
	Adaptive Taste Profile personalization		Eliminated	Need active data over time
Search	Restaurant filtering		Eliminated	
	Dish filtering		Eliminated	
Engagement features	Rewards		Eliminated	
	Badges		Eliminated	
	Challenges		Eliminated	

In the prototype, several features are fully implemented, including user account creation, taste profile setup, and the ability to save favorite dishes. The prototype also provides basic personalized dish recommendations based on a taste-matching algorithm. Social features, including following other users and viewing their reviews, are implemented, while other social

engagement features will be partially implemented and mocked. The group dining and dish matching logic is in development during the prototype and will rely on mocked data.

Other features such as live updates, badges, and rewards have been eliminated or heavily mocked during the prototype. This also applies to the restaurant-facing user interface and upgraded users, such as the SuperTasteBuddy user type. These decisions were made to ensure that the prototype remains focused on showcasing the system's most innovative features during this phase of production.

2.3 User Characteristics

The system is designed for individual diners, travelers, and event organizers, as well as restaurants looking to reach their target customers. General users are not expected to have technical expertise and will interact with the system through an intuitive web-based interface. The initial account creation is designed to guide the users in creating a taste profile creation with with minimal effort.

Typical usage scenarios include a solo diner seeking new dishes tailored to their tastes, a group of friends searching for restaurants that satisfy everyone's preferences and dietary restrictions, or a traveler seeking the best local experiences in their limited meals.

During this prototype stage, recommendations provided by the system serve primarily as a proof of concept. While the restaurants and dishes featured in the app do exist, user profiles and reviews are currently mocked to provide a functional demonstration of the system's features. This reliance on mock data is the primary constraint of the prototype and will be eliminated in future iterations and the Real-World Product.

2.4 Constraints

The TasteBuddies prototype will operate with minimal constraints. The system is built using Flask and SQLite which will result in scalability limitations at a certain size; This is a minimal concern with the prototype and will be addressed using a PostgresSQL database for the real-world product. Privacy and security constraints will also be considered in the real-world product, as security measures and best practices must be adhered to, ensuring protect user data is protected.

2.5 Assumptions and Dependencies

TasteBuddies depends on several assumptions to operate successfully. The system depends on the availability of Flask, SQLAlchemy, and Jinja2, as these are the primary frameworks for front and backend infrastructure. The system relies on Digital Ocean for hosting and assumes permissions for dependencies to be updated within this environment. During the prototype development, TasteBuddies does not integrate with external restaurant databases or third-party APIs, but these will be considered in the real-world product.