SOEN342
Software Requirements and Specifications

Tasty Foodz Delivery App
Vision Document

Instructor:
Dr. Joumana Dargham

Team: It’s not a bug, it’s a feature

Jeremie Garzon – 40062316
Facundo Alfaro – 40177429
Barthan Thirunavukarasu – 40098158
Mark Kandaleft – 40126013
Abisan Uthamacumaran – 40078084
Faizan Ahmad – 40100581

DATE: 05/03/2023

1. Introduction

As the food delivery applications evolve and grow, it is important to have a clear vision to stay competitive. With our requirements gathering complete and documented, and a product design in place, this vision document will give an outline of our mission, values, and objectives for the development of Tasty Foodz. This document will then guide our decision making and resource allocations towards our goals. With a comparative study already complete, we will be analyzing our positioning in the marketplace. Stakeholders, user environment and needs will be identified and discussed. A product overview will compare user environments with different products. The product features and risks will be discussed, and a use case diagram will be presented.

2. Positioning

2.1 Problem Statement

The problem of	*not being able to order customized food, or ordering from small or new restaurants, whilst all of that with a long delivery time. Moreover, the usage of a food delivery application can be overwhelming and indecisive for many users.*
Affects	*Delivery person, food consumers and restaurants owners*
The impact of which is	*They are not able to order customized food or order from all restaurants. Such as new or small ones, because those restaurants don't have the privilege to sell their food online, and sometimes the food consumers receive their food very late which worsens the situation. Moreover, the usage of a food delivery application can be overwhelming, and lead to indecision or a bad experience of ordering food. Sometimes, food consumers need more help, or guidance to know what and how to order the food desired.*
A successful solution would be	*New and small restaurants would be able to deliver and sell their foods more easily, faster delivery time for users, visual flexibility of the application, Virtual Waiter which will decrease food consumer's indecisiveness, and food consumers will have the option of customizing their food*

2.2 Product Position Statement

For	*Delivery person, food consumers and restaurants owners*
Who	*want to order, deliver, and sell every kind of food*
The Tasty Foodz	*is a software product (application)*
That	*Let food consumers order food, delivered by many happy delivery drivers, from any kind of restaurant near them, so all restaurants will have equal ability to sell their food via Tasty Food.*
Unlike	*Other food delivery app such as Uber Eats, where not all restaurants can sell their food by this application, the delivery time can be very long and not beneficial for ALL stakeholders, and it is not very accessible for all food consumers. Sometimes, it can be complicated to order food.*
Our product	*Let food consumers order food from all restaurants such as small, big, new ones. The delivery time will be faster since Tasty Foodz will avoid unnecessary routes for deliveries (shortest pathway). It will also let food consumers order customized food by messaging directly the restaurants. Moreover, our product will be more flexible for all devices and users using the application and it will be a very friendly-user.* *In general, by ordering with Tasty Foodz, every user will feel as if they are ordering in person.*

3. Stakeholder Descriptions

3.1 Stakeholder Summary

The food delivery app will benefit multiple people which can be categorized into different stakeholders. The main non-user stakeholders can be split into four groups: the developers, the financial sponsors, the vendors and the delivery personnel.

Name	Description	Responsibilities
Developers	us, the team which consists of six members	We all are managers, developers, debuggers, and designers working together. We will be receiving the financial benefaction if the project is a success, if not the time invested will be our loss.
Financial sponsor	Concordia Inc.	They provided us with the financial aid to develop and run this project. They strongly believe in this application and will be secondary financial benefactors from the success.
Vendors	Main sellers on the platform. Mainly composed of restaurants and other food preparation services.	Take the user's orders and prepare it for delivery
Delivery personnel	Hired and/or independent contractors	Deliver the food to the consumers in time

3.2 User Summary

The user base can be subdivided into three groups: beta testers, tech reviewers and the general user.

- The beta testers are the first users and testers to see how the app is working. They provide us with the information regarding where the debugs are and what needs to be corrected before being launched.
- The tech reviewers are the most critical users. They can advertise the app to everyone based on their opinion and can easily make the app a success or a failure.
- The general users are the majority and main users that the app is developed for. They provide the main source of income from the app. They can be divided into the following subgroups: vendors, delivery personnel and consumers.

Name	Description	Responsibilities	Stakeholder
Beta testers	First users and testers to see how the app is working.	They provide us with the information regarding where the debugs are and what needs to be corrected before being launched.	Developers
Tech reviewers	People who review the app point out any strong and/or weak points	advertise the app to everyone based on their opinion	N/A
Vendors	Main sellers on the platform. Mainly composed of restaurants and other food preparation services.	Take the user's orders and prepare it for delivery	Vendors
Delivery personnel	Hired and/or independent contractors	Deliver the food to the consumers in time	Delivery personnel
Consumers	Recipient of the services	Choose and pay for the product and service	N/A

3.3 User Environment

- Unique environmental constraints: mobile, outdoors, in-flight, and so on?

  Due to the nature of the app, it will only be available on mobile platforms, with the possibility of developing a web app for other devices later. This is mostly because the amount of food delivery app users on mobile platforms is way higher on mobile devices. Depending on the demand, a web app may be implemented later on (like those done by Uber Eats and Doordash).

- Which system platforms are in use today:

  The vast majority of end users (vendors, delivery personnel and consumers) are using mobile devices (iOS and Android).

- Future platforms:

  Possibility of a web app. Other Food delivery apps (like Uber Eats and Doordash) have developed and are experimenting with web apps resembling their mobile apps for desktop and other platforms.

- What other applications are in use:

  The app will have to use the other services provided by the host operating system. For example, Android devices will have to use Google Play Services in order to access some features like location based services, some authentication features (like sign in with your Google account) and the ability to pay with your Google Pay among other things. It's the same thing for iOS devices (location, authentication, Apple Pay, etc).

- Does your application need to integrate with them?

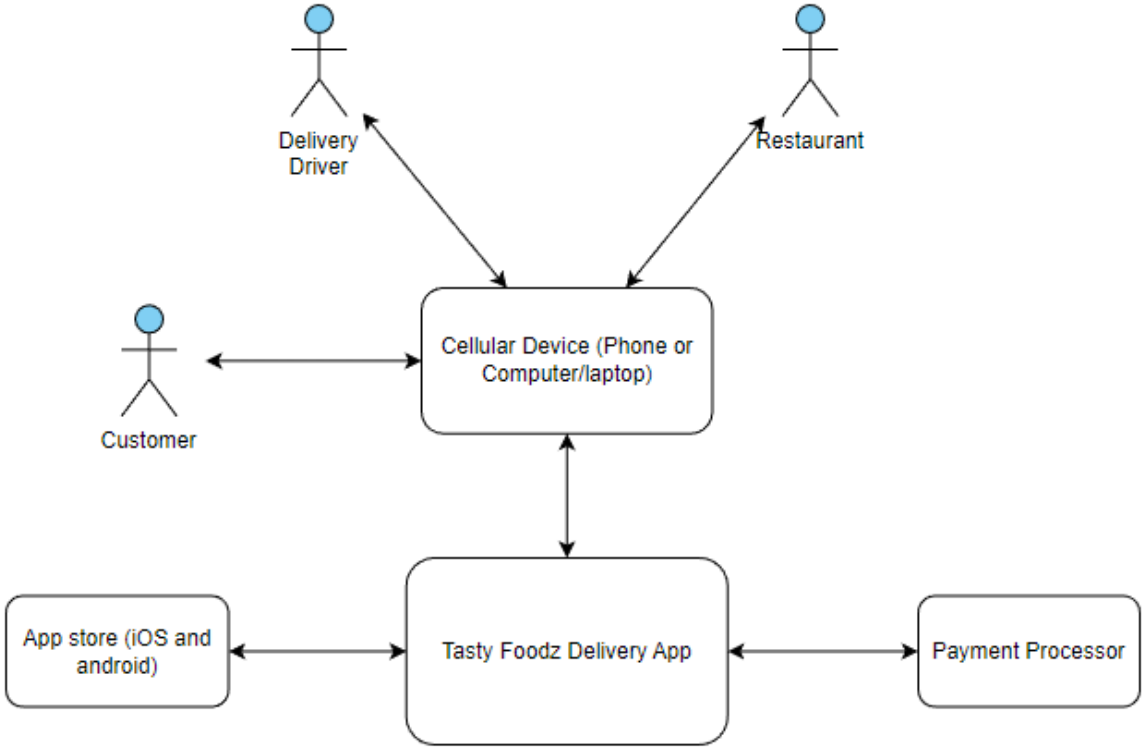
  There are some applications that we have to integrate with our app, the biggest being the location-based services, as the vast majority of users tend to order food at different locations and it makes the delivery process much easier, as the consumer can know when and where their delivery is (similar to the tracking feature on UberEats).

3.4 Key Stakeholder or User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Being able to sell custom products (food) and customize existing products if the consumer needs it	High	Vendors are forced to sell items from a pre-made menu and cannot take any custom requests	N/A	Adding new features that allow vendors to sell custom products and customize existing products
Have to be a specific type of vendor and/or have to produce a minimum amount of a product	High	Requires the vendor to be a restaurant or a fast-food chain and/or to be of a significant size and/or have a minimum output of product	N/A	Make it easier for small-time vendors to access the platform

4. Product Overview

4.1 Product Perspective



Product Perspective

Our food delivery app depends on many other systems. For one, our app must be in an app store for phone users; this means we would need to develop two versions of our app, one in Kotlin for android users and one in Swift for Apple users; furthermore, we should note that the process to enter the iOS app store is more stringent. We would also need to handle payments, and for this, we should look towards solutions that already exist, like Shopify or Stripe. Another system here is a cellular device, and it is what customers, delivery drivers, and restaurants will use to interact with our app. Lastly, restaurants are shown as actors in the diagram, however, they are a major system involved, as our app's main purpose is to allow customers to order food from restaurants.

4.2 Assumptions and Dependencies

Assumptions, if incorrect, can drastically change the vision document. It is vital to prepare for when assumptions go wrong so that we can anticipate and mitigate risks associated with cost, project delivery, and stakeholder satisfaction. Below we list assumptions and their respective dependencies.

Assumptions	Dependencies
Accepted into app stores	iOS/android app submissions (possibility of rejected submission)
Working payment processor	Successful integration of Shopify into our app and website
Successfully enlisting restaurants and delivery drivers	Financial incentives (lower commission rate for restaurants, higher pay for drivers) and good marketing

5. Product Features

5.1. Core Features

1. Start application
2. Exit application
3. Accept Touchscreen and Keyboard Input
4. Make account
5. Login to existing account
6. Make changes to account information
7. View past orders
8. Choose restaurants/items by clicking
9. Customizing item choices
10. Adding customized items to cart
11. Search for restaurants based on name, food type, etc…
12. Send and receive messages from restaurants
13. Use the digital waitress system to help make a choice
14. Removing items from cart
15. Make changes to items in the cart
16. Placing order
17. Make payment by debit, visa or paypal
18. Track delivery driver and view estimated delivery time
19. Send to, and receive messages from with your delivery driver
20. Read and/or write reviews for restaurants you have ordered from
21. Mark restaurants as favourites
22. Change language
23. Find common issues and answers on the help screen
24. Raise issues through tickets on the help screen
25. View ticket status and responses for past tickets

5.2. Other Product Requirements

1. Platform Requirements
   The application will be on IOS and android. Other platforms may get a port later on, but not for the time being.
2. Hardware Requirements
   Any device wanting to run the application smoothly needs at least 125 MB of RAM and 200 MB of storage space. A bare minimum of 50 MB of RAM is needed to run the application.
3. Performance Requirements
   In order to achieve the 5-10 ms process times we are aiming for, the device needs to be capable of multi-threading. The processor should both allow for multiple algorithms running at once, and be optimized for minimal waste to achieve this goal. If this is not possible, the program will still function but at reduced speeds.

4. Documentation Requirements
   A simple installation guide, as well as user manual are needed for less tech-savvy users to start using our application. A help centre is to be hosted online, on the application website, as well as in the application itself as a separate page. This page must both offer an FAQ, as well as accept tickets for user issues.

5. Requirement Priority
   High
   - *Platform Req* - this is a mandatory requirement to begin use of the application
   - *Hardware Req* - to achieve a stable experience, where the user can most benefit from the application features.

   Medium
   - *Performance Req* - stability and robustness should still be achieved without this requirement being completely met.

   Low
   - *Documentation Req* - as our application is simple, and uses concepts similar to other competitors on the market, both installation and use should be achievable by a very large majority of users without any help.

6. Risks and Feasibility

Risk	Risk Level L/M/H	Likelihood of Event	Mitigation Strategy
Domain Specific Risks			
Regulatory Compliance			
Hygiene regulations and food safety	**H:** If the food upon delivery does not meet safety standards, reputation and reviews will be heavily damaged.	**Likely**	Reducing likelihood of risk by working with health approved restaurants only and using insulated cooler bags.
Mode of delivery, driving regulations	**L:** Some modes of deliveries are not permitted in some countries.	**Low**	Risk avoidance can be used to make sure that no laws are broken in each country.
Internet regulations	**L:** Some countries will impose limitations	**Somewhat likely**	Risk avoidance can be used to make sure that no laws are broken in each country.
Logistics and Supply Chain Management			
Robust logistics	**M:** Sloppy code can be slow and lead to a lot of bugs	**Unlikely**	Mitigating risk consequences by doing code reviews, testing agile development and documentation.
Supply chain management	**M:** Bad management at any level will cause problems	**Likely**	Reducing risk likelihood by working in an agile environment.

Delivery delays	**L:** Food not arriving at the specified time might impact reputation.	**Very likely**	Reducing risk likelihood by implementing a solid timing feature.
Data Privacy and Security			
Hacking or unauthorized access	**H:** This could result in Ddos or flooding.	**Unlikely**	Reducing risk likelihood by hiring third party auditors who are professionals at detecting anomalys.
User personal information leak	**H:** Such a leak might lead to lawsuits.	**Unlikely**	Reduce risk likelihood by having a secure code with comprehensive penetration testing.
User payment details leak	**H:** Such a leak might be used for fraud and lead to lawsuits.	**Unlikely**	Risk is avoided by letting payements go through third parties who encrypt banking information.
Process Related Risks			
Competitive Pressure			
Inability to keep up with changing market and customer preference	**M:** If the app cannot follow trends and customer preference, ratings might be impacted.	**Somewhat likely**	Reducing consequence likelyhood by having a team of people doing market researches while collaborating with other companies to stay up to date.
Inability to improve features and user experience	**M:** Lacking behind competition in terms of performance will reduce profitability.	**Likely**	Reducing consequence likelihood by having agile development which would accelerate our response to feedback.

User Experience and Feedback			
Bad delivery reviews	**H:** Review can harshly impact profitability	**Unlikely**	Mitigate consequence by offering compensations.
Complicated features	**M:** The Virtual waiter user experience too complicated	**likely**	Reduce risk likelihood by giving users simple questions and customizable filters.
Bad visual design	**L:** A bad design would prevent user satisfaction	**Unlikely**	Reduce consequence likelihood by allowing font, color and other visual customization.
Technical Scalability and Performance			
Poor reliability as usage increases	**H:** A lack or resources, staff or management might produce a shaky architecture at the larger scale	**likely**	Reduce risk likelihood by implementing scalable architecture and performing load testing to detect instabilities.
Poor performance as usage increases	**H:** As the usage increases, the app slowing down or crashing might lead to loss of business	**likely**	Reduce risk likelihood by optimizing the code, implementing cashing to reduce load and test.

The project has its risks but it also has its rewards. From the risk assessment and mitigation plan above, we see that our biggest constraints are food hygiene quality, robust code architecture and secure personal and banking information. If any of these constraints are not satisfied, the system would collapse and the application fails.

However, our project is feasible with the help of third parties such as a specialized team of programmers that will review, optimize and test the code for security breaches and smooth operation with high usage. Third parties will also process transactions since they are known to have good security and encryption. We also work with food health and safety organizations to ensure that food quality is optimal for human consumption.

As for the competitive aspect, even though we do not have the same amount of money as the bigger corporation, we have the advantage of learning from their previous experiences. Our product will be able to compete in the market because we have a better understanding of the requirements when compared to other companies when they started. We will also have a team that is constantly researching trends and user preference while cooperating with other companies to learn what our product should be like.

7. Use Case Diagram