Cooking

h

optimizer

[redacted]

CSC 3150: Systems design
System Design Proposal Part #1
Professor Cameron
5/3/2024

Contents

Executive Summary

~~**While the *Executive Summary* is the first thing your reader will see, wait and wr**~~**ite it AFT**
~~**the rest of the doc.**~~ ~~DO NOT FORGET to add this to~~ Part 2! ~~**It's a critical part of the document.**~~

In individual and commercial restaurant environments, there is an ongoing difficulty with displa
steps, collecting data on dishes, and correcting recipe information. To solve this problem, a more adv
solution than pen-and-paper is necessary. The application specified in this document so that a chef c
information about their food preparation, consolidate data over a long period, analyze individual char
might improve the recipe, and change the recipe as needed, all within the bounds of the application.
application could also be commercially applied so that restaurant chains can leverage their large nur
and multiple locations to collect this information. An application that optimizes recipes and cooking is
to create a product usable by the multitude of chain and individual restaurant businesses.

~~**Analyze the Executive Summary sections of *previous years' project proposal example***~~
~~**equivalent information *for this project*? You may wish to put this on a separate page.**~~ **Rea**
~~**appreciate that, but it can go above the Introduction and Overview if it is short.**~~ **While the**
~~**the Introduction and Overview, this is the official BRIEF introduction of yourself**~~**, your clie**

1.0 Introduction and Overview

Problem Statement

When preparing f~~oodIn the food i~~ndustry, tracking cooking metrics is challenging. Cooki
metric~~sMetrics like~~ this are precious for most kitchens, as they allow for the analysis and imp
of recipes. However, ordinary means of tracking these statistics in a kitchen environment is
challengin~~g. A pa~~perThe paper-and-pencil method is futile, as flour dust and liquids often ren
and other writing materials invalid for long-term use. Standard note-taking applications usua
down when recording or reprinting recipes, as recipes follow a complex and unstandardized f
that is difficult to edit. Overall, cooking metrics are difficult to track and are incredibly valuab
entity that wishes to improve their cooking.

Project Vision and ~~Sco~~pe

This project aims to provide a service that optimizes cooking procedures through data co
This application’s scope is limited to serving kitchens and cooks in the United States. The pro
to optimize resource usage and product quality during the cooking process.

In response to this vision, this document sugge~~sts creating we i~~ntend to create a cross-p
cooking application that optimizes recipes in a casual and business context. Ideally, users wi
application as a recipe reference, quality tracker, and improvement tool. The application sho
users through a specific recipe step-by-step and allow them to ~~recor~~d, changemodify, and an
recipe. The data tracked should include, but is not limited to, cooking time, output quality, ou
quantity, recipe adjustments, and recipe reference photos. To ensure recipe optimization, the
optimizer should use this data to make graphs and suggestions or improve quality.

Requirements Summary

The application must be easy to use and work across platforms. This requirement
critical because the target audience probably lacks technical ability.

A responsive UI (User Interface) is required to display and interact with individual r

For larger customers, such as chain restaurants, the cooking optimizer requires a
centralized database that customers can report to. T~~his featur~~e iss necessary for b
profit motive and centralized data collection.

So that the users can iteratively improve their recipes, the optimizer is required to
the cook to note down individual adjustments to recipes. This requirement is critic
recipe improvement, as chefs could note individual changes or slight variations tha
improve the dish, creating a recipe variation.

If a commercial route is cho~~s~~en, aAn on-call support team and documentation are
required to ensure smooth usage and minimal issues.

~~The company must secure a reasonable executive user base interested~~ in this pro
~~ensure profitabi~~lity.

This application should be able to import, parse, and export recipes from individua
cookbooks, and notes. Recipes generally follow similar patterns and notations, so b
able to extract recipes from alternate sources into our application is an essential s

The application should be able to store data about yield, success rate, and average cooking time.

Stakeholders and Their Interests

One stakeholder is individual household cooks. Because individual chefs are intere
a high-quality product and rarely get the chance to iterate on their recipes, keepin
of past data is critical.

This company is a primary stakeholder in this project, as it reaps the benefits of ar
expanding market reach and can offer its technical expertise to other stakeholders
company will also have a stake in the overall food service industry and with indivi
consumers.

A stakehold~~er r we~~ wish to secure is individual restaurant chain executives. Chain
restaurants constantly produce food on a commercial scale and are often looking t
optimize their dish selections. Because our application can assist with aggregating
visualizing this information, these restaurants should want a stake in our product.

One stakehold~~er we~~ wish to avoid conflict with is the FDA (Food and Drug Adminis
As our application intends to give step-by-step instructions on food preparation, he
and safety protocols come into play, opening us to liability issues. If our applicatio
provides instructions for dishes that include risky ingredients or are dangerous to
there could be issues.

An unacknowledged stakeholder in this project is application distribution centers.
Microsoft, Google, and Apple stores have strict criteria regarding published applica
that are challenging to comply with.

Another set of stakeholders are individual recipe holders and hosting sites. Cookbo
and the recipes in them could fall under copyright protection, and the terms of ser
cooking sites often explicitly forbid mass web scrapers and reverse-engineering th
(*Terms of Service - Dotdash Meredith*, n.d.) However, the application's goal isn't to
infringe or republish individual ~~r~~ecipes, but to import and optimize them.

Expected Costs and Benefits

Business Benefits

One ~~of the~~ intangible bus~~inesses benefits~~ we receive benefit from a cooking optimize
restaurant dependency on our project. Because all relevant data is in our centralized d
restaurants continuously rely on our application for their optimization needs.

A great intangible benefit is the gifted advantage to individual cooks. For scientificall
chefs, recording every detail about their cooking and what makes one recipe attempt b
than the other is difficult to quantify and record. When an individual attempts to impro
cooking, an application to give instructions, record variations, and recognize beneficial
is a powerful tool.

Our primary monetary benefit is adoption by food service establishments. When they
a specific size, monetizing per location regularly rather than per business becomes pos
overall gain should cover more than physical server and database costs; it also has the
benefit of tying our growth to the ongoing development of the restaurant industry. Wit
product, the restaurant will attract more customers. If there are more customers, the
restaurant must expand or open another location, increasing our overall profit and crea

larger sample size of dishes to iterate on—progress then snowballs, with better food, b
restaurants, and a higher profit for all parties.

Business costs

If, beyond the minimum viable product, our company ~~se~~t out ~~to~~ mMaintaining a corp
server and database for chain restaurants, the service w~~ould be a hi~~gh is the highest on
associated with ~~t~~hea cooking optimizer. The overall cost isn’t easy to estimate, but a sho
test with a singular restaurant should be inexpensive enough to test the overall concept
the application expands, however, server and database costs increase dramatically.

Another cost to consider is the cost of an ongoing technical support team. Documentat
technical support consume a large part of development time, as code is useless without
instructions on how to use it. Documen~~tat~~ion is are necessary when creatin~~g a~~ny applica
~~matter ho~~w regardless of the ~~designwell-de~~signed it is.

An initial cost when creating the application is the development team tasked with desig
and implementing these features. Our development teams are well-equipped and well-tr
so looking toward similarly sized projects for cost analysis shouldn’t be difficult.

Constrain~~ts~~

The primary issue with our application is limited time for user input. Most
restaurants, especially high-traffic chains, use a paper or digital order system. Food is p
according to customer orders, and preparation time constraints are intense. These time
constraints dramatically limit the applications' reach, as most food services don’t have
to use an optimization applicati~~on. This~~This fact limits the use of our application to time
insensitive restaurants and high-end chains. However, despite this constraint on certai
establishments, the application still has an excellent market. This application still appli
high-end diners, preorder food locations, bakeries, and small businesses.

A significant limitation of this service is the scope reduction to only a United S
market. The prospect of international markets is exceptionally enticing for this project.
not practical to expand into these markets within the minimum viable product because
with food regulation and the possibility of localization errors. There is the possibility of
expanding into international markets later, but for the initial launch, the target is the U
States.

Another tempting possibility that we must constrain is mass data collection. T
is an ongoing trend in the technology industry to aggregate and collect as much perso
information ~~as~~ humanly p~~ossible. It is especially appealing to aggregate all r~~ecipes into
~~cooking social network, with verified cooking data tabulated as a sign~~ificant source of
~~information and recipe var~~iation. With a cooking application, this is exceptionally temp
the application can determine the ingredients used, time spent, average time cooking,
person cooked, and where a person cooked, the advertising possibilities would be limit

However, attempting this raises several issues. On the legal side, the United States anc
European Union are currently tightening their legislation on d~~ata collection~~. Regarding
~~sharing, moderation and proper data analysis would be incredibly difficult,~~ as users hav
~~qualms about falsifying or manipulating data to their~~ advantage. Coupling these with ir
security risk and high upkeep cost~~s m~~akes suchexploiti~~ng custo~~mers a project nonviab
However, securing a small market of dedicated customers without harvesting advertisi
should still provi~~d~~ebe a valid return on investment.

Recommendation

~~What do you want your reader to do upon receipt of this document? Wanderer's T~~
~~is your primary decision-maker; what actions should they take after reading your propos~~
~~"*next steps" should now be taken?* DO NOT FORGET to add~~ this to Part 2!
If this document is approved, a subset of resources should be allocated to test the conce
cooking optimization application. These resources include a small development team an
consumer testing. A development team of 3-4 people for 2-3 months is necessary to cre
minimum viable product, following this design document and associated resources. Cons
and field testing is also required to ensure the application can improve dishes in the kitc
environment. Creating a funded cooking optimization application would be the next logic
testing is successful and the minimum viable product has potential.

Document Overview

The rest of this document outlines the initiation, assessment, requirements, and recommend
creating a cooking optimization app~~licationation~~ application. This document intends to rat
and specify the design details for developing an application that assists with recipes and
optimization. The initiation section of this document provides a broad overview and the p
initiation request. After~~-~~wards, the feasibility section designates ~~potential~~ possible risks ar
assesses the overall feasibility of attempting this proj~~ect. Then, the~~re are the system
~~requirements, w~~hich system requirements outline all necessary components, including da
non-functional and functional necessities for creating this application. Finally, there is the
conclusion and recommendation, stating the following steps to pursue the creation of this
application.

~~Describe the organization and content of the rest of the Sy~~stem Proposal.
~~Be sure this needs to be **updated for the Part 2 submission** to include~~ the entire docume

2.0 System Initiation

Project Initiation Request (PIR)

PIR-00000*[PIR Number to be assigned by the Project Office]* Project Initiation Request (PIR) – Level~~2~~1 v6.0
Project Name: Cooking Optimizer Student Name: Kyler Veenstra

This Project Initiation Request (PIR) is to be completed for all requests expected t hours of effort or over 4 weeks of total duration. For larger requests requiring ove estimated project costs greater than $5,000, this template is used to assess the p and get approval to scope and plan the proposed project.
If approved, the Level 2 template (System Proposal: Part 1 and Part 2) must be co
~~**NOTE: Sections 0-4 are required.** Section 5 is optional, but any ideas on~~ estimati ~~included. **Replace the *italic* prompts with your answers/information**. [~~Expand ea ~~template as needed~~**.]**

0. General Project Information

Project Name:	Cooking optimizer
Two Sentence Request Description:	Statistical and organizational tool for prominent restauran and individual cooks. Record recipes, cooking success rat ingredients used, average yield, and other details.
Requested Launch Date(s):	May 20th 2024
Department(s) Affected By Project:	IT department, Security, Billing, UI Design.
Project's Customers:	Individual cooks or high-end restaurants that desire great business efficiency.
Date Request Submitted:	2024-04-16

1. Project Sponsor and Manager

Project Sponsor

Name:	Andy Cameron
Title:	Professor
Departme t:	Computer Science - SPU
eMail:	acameron@spu.edu

Business Project Manager & Requestor

Name:	Kyler Veenstra
Title:	Student
Departme t:	Computer Science - SPU
eMail:	veenstrak@spu.edu

2. Business Problem or Opportunity: The motivation for this request

Describe the problem or opportunity that you would like to solve. Include a simple, high-level description of this request's opportunities. Focus on the problem or opportunity, not the solution. Be sure to include any date or deadline-related dependencies or the project.

Individual cooks and high-end restaurants must work with expensive and scarce resour regularly. For these groups, high-end meals require a significant investment of time, res energy, and practice. One way to reduce the risk of failure is by carefully tracking the ti ingredients, revisions, average yield, and other details during the cooking process. How keeping track of these with a notebook or spreadsheet is difficult in a kitchen environm Because of this, an application that could keep track of these details could be beneficia

3. Justification, Impact, and Importance

What is the financial impact and justification for this request? How will the investment of time, resources, and capital be returned to o (Please note any contractual or regulatory requirements associated with the request. If you have an NPV, IRR, or ROI calculation, pleas link(s) in this section.)

Assumptions

Include at least two. Add more rows to each table as needed.
Product testing is possible with a simple, localized application. If there's a high adopt a complete project targeting multiple restaurant chains, small businesses, and chefs pursuing. The project can also be removed without significant risk if adoption decline systems prove inefficient.
Since company-owned servers could store critical information, companies that use th become indefinitely dependent on our product to store information. Recipe and kitch are currently unordered, difficult to aggregate, and hard to reference. This disorder n difficult to switch to competitors or use other systems if they emerge.

Competitive Landscape / Context

Include at least two
Household kitchens–Normal kitchens that wish to better analyze their cooking skill or yield could easily use our application. Users could also share a certain number of the recipes for free on the platform.
Small Businesses–Small businesses needing to ration their inventory, optimize their r yield, or measure their cooking time could use our application to improve their busine
Restaurant chains–Chain restaurants could, on a larger scale, track their average pro rate, ingredients used, and caloric content. To network outside their individual could pay our business to aggregate and store that data in a database, making it eas level executives to refer to when adjustments are necessary.
Luxury Restaurants–When a customer desires a high-quality product, and expensive ingredients are necessary, statistics and records can reduce the overall risk.
Existing restaurant management software–Most software is concerned with keeping t orders and delivering to specific customers. As such, we can see this kind of collaborative partner rather than a competitor.

Tangible Return, Opportunity, or Value	One Time	Ongoing
Include at least two. Estimate the best you can.	$ 0	$ 0
Restaurant dependency on our product	$ 0	$ 15/month, per location
Customers paying to display extra recipes	$10/Customer	$0

Intangible Benefits	Impact or Value
☟ *Include at least two. Estimate the best you can.*	$ 0
☟ Aggregated data from a variety of users	
☟ Connections in the food industry	
☟ Contact at every point along the food preparation process	

4. Product Requirements

The Project team will gather detailed requirements once the project is approved. Use this section to articulate the critical solution com scope the project's size and complexity. Do not describe how the solution will be implemented; instead, only list the functionality or re to receive when the product is complete/delivered.

4.1. Must Haves

4.1.1.	*Include at least two. Add more rows to each table as needed.*
4.1.2.	Import recipes
4.1.3.	This project requires an extensive database of recipes, ingredients, and food sele
4.1.4.	Desktop / mobile / web application
4.1.5.	A centralized and tested paid server needs to be present for large chains.
4.1.6.	Export recipes
4.1.7.	Step-by-step walkthrough of several recipes
4.1.8.	Responsive cooking time tracker
4.1.9.	Yield tracker.
4.1.10.	Success rate tracker

4.2. Could Haves (Nice to Haves)

4.2.1.	*Include at least two.*
4.2.2.	Common substitutes for ingredients listed
4.2.3.	Inventory control / inventory control integration
4.2.4.	Embedded cooking timer
4.2.5.	Share a few personal recipes for free.

4.3. Won't Haves (Don't Do's, aka Out of Scope)

4.3.1. *Include at least two*

4.3.2. For monetization reasons, this program will not allow users to set up self-hosted for data management.

4.3.3. This system will not integrate with employee management / payroll to remove pa compliance issues.

4.3.4. This product should not initially integrate with software that purchases ingredien

4.3.5. Because of safety issues, this application should not control autonomous applian

4.3.6. The program, at least initially, should not work internationally.

5. Project Costs (Operating and Capital: Onetime and Recurring) [Optional]

This section is typically fleshed out after the requestor has submitted a PIR and received approval for the initial scoping effort. It captu estimates, capital expenditures, and other costs associated with performing this work and creating the product/solution. If the submitt or estimates on what these costs are or suggestions on how they might be estimated, please include those here. Add brief description ***Include at least 2 comments on your thinking around these items, even if you don't have specifics yet.***

Labor Costs

Type	Team(s) Affected	Low (hrs)	High (hrs
Analysis & Design		0	0
Development		0	0
Testing and Quality Assurance		0	0
Systems Integration		0	0
Deployment		0	0
Support and Maintenance		0	0
Sales and Marketing		0	0
Total		**0**	**0**

Capital Costs

(Equipment, Software, Licenses, …)

Comments: Conceptually, this has a simple implementation. The difficulty will be in making a responsive and helpful User Interface can suit the needs of individuals and companies. The other significant cost is successful advertising, as there will be no adoption with least a small group of dedicated individuals who generate word-of-mouth advertising.

Description	Quantity	Cost (
Item 1		$ 0
Item 2		$ 0
Total		$ 0

Comments: The main cost for this product is initial server creation. Software like integrated development environments and user interface design applications we should already have hand.

Maintenance Costs (Costs after the product is live)

Type	Hours / Month Low	Hours / Mon
System / User Support	0	0
Business / Process Support	0	0
Total Support & Maintenance	**0**	**0**

Comments: The major cost after publication is server upkeep and security maintenance. Support will also be an additional cost.

3.0 Feasibility Assessment

Introduction

The following section is a feasibility analysis for the Cooking Optimizer project. This prop
measures this project's overall feasibility and attempts to give a rough estimate of how feasi
project is overall. All feasibility is rated on the following scale from greatest to least: ideal, hi
feasible, feasible, low feasibility, and not feasible. The risk will be quantified by high, mediun
risk.

Technical Feasibility

The cooking optimizer is feasible from a pure development standpoint, although there
medium risk.

Overall, development carries a low risk.

The company has experience with technical projects and database management.
prior experience should make project development more straightforward.

The solution as described requires minimal innovations or advances and can lever
standard data storage and design patterns. This fact should reduce research costs
make development more linear than your average application.

A system needs to be developed to import and store recipes. Recipes are a volatil
format with no overarching classification, and creating a standard classification wi
difficult.

The size of the application holds a medium risk.

In terms of scale, the cooking optimizer is a relatively medium-sized application co
to other projects.

Developing a proprietary application has an overall low risk.

Our company has experience developing proprietary applications on multiple occa

The company kno~~ws t We k~~now the overall~~ stru~~cturale an changes that ~~n~~eed to be
~~necessiti~~es when creating a proprietary closed-source application.

Securing the application holds a high overall risk.

Our division currently has no experience with overall access control and managem
Individuals using our application will need accounts and logins, making us respons
security standards.

Cross-platform compatibility holds a low overall risk.

This company has experience developing applications for standard operating syste
devices. Therefore, it should be reasonably feasible to create a cross-platform app

Resource Feasibility

The cooking optimizer has a low overall feasibility and high risk from a resource standp

The lack of on-hand resources represents a high overall risk.

The company, as of current, has no available servers or databases allocated for a
venture. As this is a significant part of the profit motive, it puts the application in a
position.

No on-hand staff are available for technical support, and the company has minima
experience doing as much. Hiring such a team is expensive but straightforward.

Schedule Feasibility

The feasibility of the schedule is currently ideal and carries a low overall risk.

As of now, no companies or contracts rely on this project.

If the compan~~y~~we ent~~ere~~d intoofficial contracts, the application would have a strict dea
changing the overall feasibility and increasing risk.

Organizational Feasibility

The organizational structure currently carries ideal feasibility but a medium risk.

From an organizational perspective, the project has a medium risk. This company is we
organized to handle such an experiment and has the teams and staff on hand to try su
project.

Agile frameworks are well-established and would be excellent for tackling this project.

It is organizationally challenging to cut our losses if risk or development is too difficult.

Legal and Contractual Feasibility

From a legal perspective, the project is feasible but carries medium risk.

Currently, we are not contractually obligated to develop or create this project. It is ther
possible to cut the project if the risk or cost proves too great.

Because the optimizer technically maintains instructions on preparing food, our applica
must conform to national food preparation standards, and advice from the FDA might b
required for commercial usage.

The application might have difficulty with copyright and importing, requiring legal cons
on the overall boundaries of the application.

Additional Comments

A crude prototype application is necessary to ensure that this service would be in dema
provides adequate improve~~mentsas~~sistance.

~~Additional Comm~~ents

~~A crude prototype application is necessary to ensure that this ser~~vice would be in
~~demand and provides adequate ass~~istance.

Conclusion

In its current state, this is a highly feasible project, but it carries a medium risk. To mana
overall risk of this project, rigorous testing, iteration, and review will have to be done
ensure a quality product.

**~~Be sure to update this Conclusion section, if needed, for Part 2 bas~~ed on anyth
~~learned since submittin~~g Part 1.**

4.0 Requirements Definition

Introduction

This ~~section of the docu~~mentdocument ~~section i~~s to outlineoutlines the complete list of require features and functionality. It overviews several requirements, including the data, functional, ar functional. The data requirements entail the data that must be stored to make the application The functional and non-functional requirements outline the requirements of application uses, a external requirements. Overall, this seg~~me~~nt aimIt iss to set the baseline ~~-~~requirements forof a product~~. minimum viable pr~~oduct.

~~COMPLETE DOCUMENT REQUIRED~~ REVISION: ~~All functional requiremen~~ts will eventually b ~~represented in the Requirements Model (Use Cases) unless the analyst forgets t~~hem (which ~~thing). To help the reader confirm that all functional requirements have been~~ modeled, ***inc*** ***~~specific cross-references~~*** ~~to the Section 5 Use Case associated wi~~th each requirement.

~~Any requirements not included in the MVP use cases (those for later versions) MUST~~ be describ ~~System Evolution Section, so send your reader to that section for more~~ information.

Data Requirements

1. Recipe storage~~Capture the primary known data needs for the system in this section~~. Take note ~~as you complete the first submission (Part 1), but~~ **~~you don't need to submit it until~~ the second** **~~submission (Part 2),~~** ~~as Use Case modeling will reveal additional data needs. Try to~~ group these i ~~way that makes sense. Presenting data requirements in an organized fashion~~ will help databas ~~designers and software devs down t~~he road!
    - 1.1.The ingredients list of the recipe needs to be stored for reference
    - 1.2.The complete original instructions for the recipe need to be preserved.
    - 1.3.Partitioning markers between individual recipe instructions need to be made, so the chef c on multiple parts of the recipe simultaneously.
    - 1.4.In case of data corruption and for attribution reasons, the recipe's source needs to be store
    - 1.5.A visual identifier, such as a title or image, must be available.
    - 1.6.Integrated timer locations need to be stored in the recipe data.
    - 1.7.Component and related recipes for items need to be referenced.
    - 1.8.Individual recipe notes need to be stored.
        - 1.8.1. The ability to add warning points where the recipe is easy to mess up and notes on a failures is required.
        - 1.8.2.There can be notes in positions where the recipe is viewer-dependent and needs fine adjustment.
2. Recipe variations
    - 2.1.To properly prepare for a variation, the application should note changes in recipe ingredie
    - 2.2.The application should be able to note changes in recipe preparation.
    - 2.3.Expected changes to the taste should also be recorded.
    - 2.4.Preparation time for the recipe should be noted.
3. Recipe preparation data
    - 3.1.The recipe's fail/success rate should be stored to consider opportunity cost.
    - 3.2.Yield per recipe is a necessary factor to keep track of.
    - 3.3.Storing taste ratings per preparation allows the chef to consider his and other's tastes.

3.4. Logging the differences between preparations will help chefs track when minute changes i
the recipe.

3.5. Average preparation time allows a chef to consider the time investment when preparing a

3.6. To create visible graphs, record the date the recipe was prepared.

3.7. Extra notes are needed for later consideration of the data to determine if the creation proc
a statistical outlier.

4. Business data

4.1. For businesses, we need to store an individual chef's entire local recipe database and sync
and forth with the overarching server.

4.2. We must store information relevant to company keys, such as hashes and license codes.

4.3. The application must store information about the restaurant's location and associated keys
verify its purchase and rightful use.

Non-functional Requirements

1. Input

1.1. Input errors should be removed from the application by design, making entering in
information very difficult.

1.2. Inputting information should be a simple, non-time-consuming process.

2. Installation, maintenance, and data management

2.1. The software must be portable and easy to share between devices. Special permis
rights should be unnecessary.

2.2. Hardware requirements for running the product must be minimal.

2.3. In return for payment, our business is to do system maintenance, maintain backup
handle issues.

2.4. Backups should be created at relevant intervals and be stored on company servers

2.5. The cooking optimizer must have a high overall uptime based on business hours.
Maintenance should only be scheduled during off-hours.

2.6. Installation should be relatively painless.

2.7. The application must work on mobile and desktop devices.

3. Administration

3.1. Purchasers of our business product should be able to access notes and metrics fro
employees and visualize them.

3.2. This tool should be able to report individual employee usage back to the purchase

4. Support

4.1. Documentation is necessary for upper management. Other instructional tutorials s
available to end users.

An integrated, optional tutorial would be beneficial to teach end-users how to use th
application.

4.2.

Functional Requirements

1. Basic Functionality

1.1. A primary function of the application for restaurants and individual chefs is a recip
walkthrough feature. This feature allows the user, on their device, to step through
individual instructions for preparing a dish.

1.2. Another required function is to import individual recipes from either books or webs

1.3. Individual cooks should be able to create a “variation” of the recipe based on their experience and taste. This variation could include something as simple as instructio cooking at higher altitudes or as drastic as removing key ingredients for allergen re

1.4. Chefs should be able to record when a recipe goes awry, what the problem might b been, and if it tasted better or worse.

1.5. Users should be able to record their overall success rate and relevant recipe yield.

1.6. The application should keep track of completion time from start to finish.

2. Optional but beneficial functionality.

2.1. A cooking timer built into recipes is a handy function to accompany recipe walkthr

2.2. A helpful function is an easy substitution and measurement conversion.

2.3. The user would benefit from the ability to add individual photos for reference.

3. Data information storage

3.1. There is a requirement to store recipe data~~. This d~~ata generally includes a list of ingredients, pictures, and instructions, with some variations based on taste and conditionals.

3.2. The applicat~~ion is requi~~red tomust store adjustmen~~t information both i~~n a databas ~~locally, such as taste rating and recipe ~~variations, such as taste ratings and recipe variations, both locally and in a cloud database.

4. Data visualization requirements

4.1. The user must be able to visualize their overall success rate, production rate, and work. This visualization is so that the user can optimize their overall cooking strateg

5. Access control

5.1. Accounts should be secured behind a username and password for corporate users.

5.2. Unauthorized or individual users should not have any access to databases.

5.3. Each corpora~~t~~e cCustomer should have their recipes “siloed,” unavailable for quer other companies without author~~i~~zations.

~~Non-functional Requirem~~ents

~~1. Input~~

~~1.1. Input errors should be removed from the application by design, making it very difficul invalid informat~~ion.

~~1.2. Inputting information should be a simple, non-time-consu~~ming process.

~~2. Installation, Maintenance, and Data man~~agement

~~2.1. The software must be portable and easy to share between devices. Speci~~al permissio ~~rights should be unnecessary.~~

~~2.2. Hardware requirements must be ~~minimal.

~~2.3. In return for payment, our business is to do system maintenance, maintain~~ backups, a ~~handle issues.~~

~~2.4. Backups should be created at relevant intervals and be kept on c~~ompany servers.

~~2.5. The cooking optimizer must have a high overall uptime based on business h~~ours. Mai ~~should only be scheduled during off~~-hours.

~~2.6. Installation should be relatively ~~painless.

~~2.7. The application must work on mobile and deskt~~op devices.

~~3. Administration~~

~~3.1. Purchasers of our business product should be able to access notes~~ and metrics from
~~employees and visualize~~ them.

~~3.2. This tool should be able to report individual employee usage back~~ to the purchaser.

~~4. Suppor~~t

~~4.1. Documentation is necessary for upper management. Other instructional t~~utorials shou
~~available for end u~~sers.

~~4.2. An integrated, optional tutorial would be beneficial to teach end-user~~s how to use the
~~applicatio~~n.

~~f~~

Introduction

~~Start with a brief text overview of what the drawing and supporting documentation wi~~ll show. Re
~~some of your readers may be customers or end-user types unfamiliar with Use~~ Case Diagram
~~Descriptions, so you will probably need some graphics here. Again, avoid soundin~~g like a text
~~class notes) via direct co~~pying.The following documents show a cooking optimization applica
requirements and proposed functionality. Requirements modelling in this document includes
case diagram and 20 use case descriptions. The use case diagram shows which functionality
points align with individual “actors” using our product. Each use case is numbered 1-20 in ar
order. As per the key, blue circles represent food preparation, purple circles represent manag
use cases, green circles represent recipe modification and creation, and orange circles repre
database maintenance done by the company. Each use case description corresponds to the i
in the circle. The use case descriptions specify the various aspects and edge cases that shou
considered during implementation.

Use Case Diagram

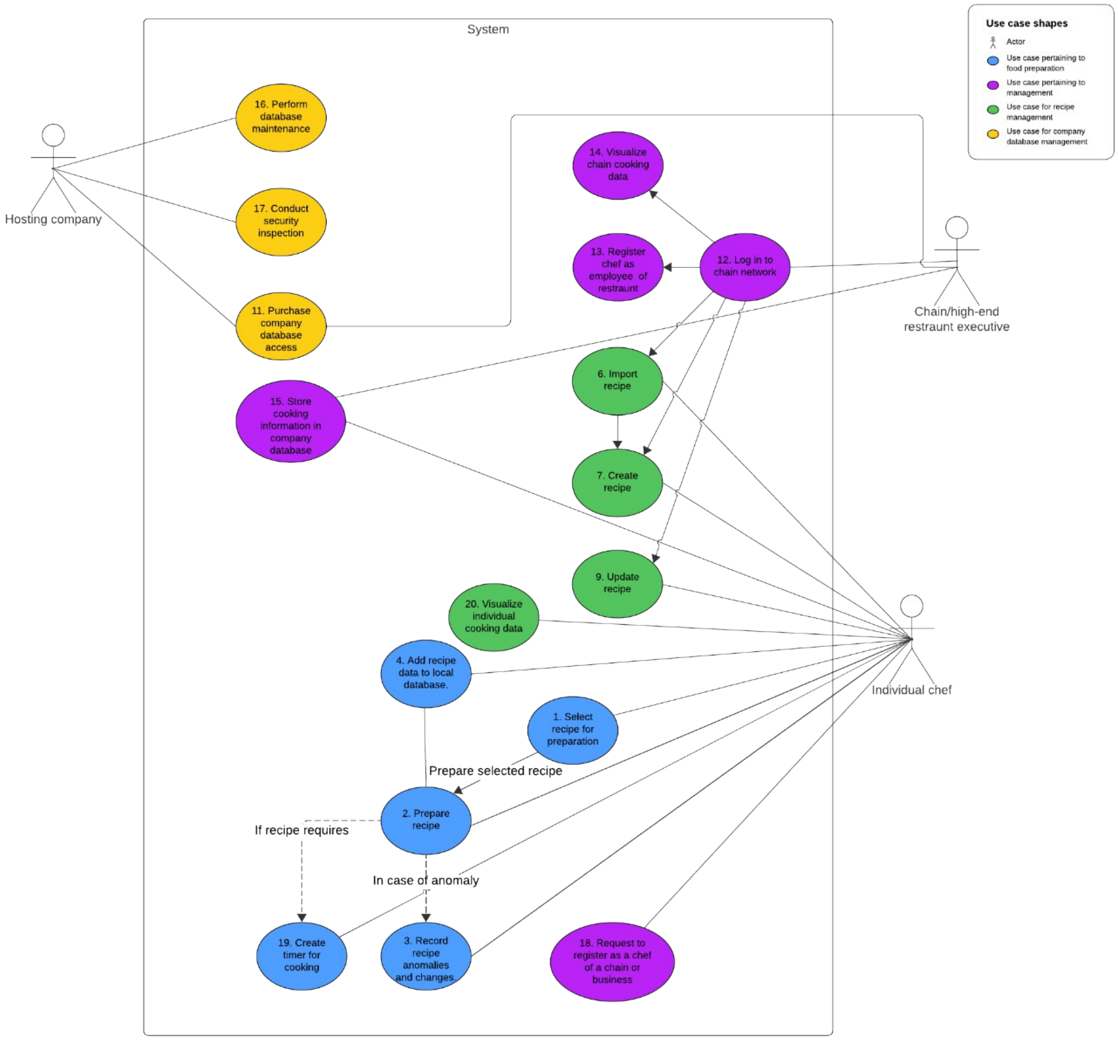
System
Use case shapes
Actor
🔵 Use case pertaining to food preparation
🟣 Use case pertaining to management
🟢 Use case for recipe management
🟡 Use case for company database management
Hosting company
Chain/high-end restraunt executive
Individual chef
16. Perform database maintenance
17. Conduct security inspection
11. Purchase company database access
15. Store cooking information in company database
14. Visualize chain cooking data
13. Register chef as employee of restraunt
12. Log in to chain network
6. Import recipe
7. Create recipe
9. Update recipe
20. Visualize individual cooking data
4. Add recipe data to local database.
1. Select recipe for preparation
Prepare selected recipe
2. Prepare recipe
If recipe requires
In case of anomaly
19. Create timer for cooking
3. Record recipe anomalies and changes
18. Request to register as a chef of a chain or business

Figure 1 - Use case diagram for cooking optimization application

~~Use Case Diag~~ram

~~Create these using a good drawing tool like LucidChart, draw.io, Google Drawing~~, Apple Notes, e

~~Use Case Descript~~ions

~~You must complete a separate Use Case Description template for each Use Case in your~~ Use Case Di

~~Use Case description MUST begin on a~~ separate page.

~~**Use-Case Descriptions**~~

Use Case Descriptions

Use Case Name: Select the recipe for preparation | **ID**: 1 | **Importance**: Must have

Primary Actor: Individual chef | **Use Case Type**:

Supporting Actors: N/A

Stakeholders and Interests:
Restaurant executive – If applicable, has a stake in ensuring dishes are created according to customer needs.

Brief Description: The chef, by order or desire, selects a recipe and variation from the list recipes to prepare.

Trigger: A chef decides to prepare a recipe.

Type (mark one): X__ External ___ Temporal

Relationships:
Association:
Individual chefs
Include:
Prepare recipe
Extend:
Record recipe anomalies and changes
Generalization:

The Normal Flow of Events:
The chef selects a recipe from the local database to prepare.

Sub-flows:
1. Get the list of recipes from the local database.
2. Select a recipe from the list to prepare.
3. Get a list of the variations of that recipe that can be prepared.
4. Select a variation to prepare.

Alternate/Exceptional Flows:
If, during step 1, the chef is registered as part of a chain,
1.1 Sync the local database with the corporate database.

Special Requirements:
Information display
1. The average cooking time for each recipe must be displayed.
2. The title of the recipe should be shown.
3. A display of recipe ingredients should be available to the chef after selection.
Security
1. Authorization should be required to sync to the corporate database.

To do/Issues:

N/A

Use Case Name: Prepare recipe	**ID**: 2	**Importance**: Must Have
Primary Actor: Individual chef	**Use Case Type**:	

Supporting Actors: N/A

Stakeholders and Interests:
Recipe executives – If applicable, the executive is interested in ensuring that the dish is cre
served, tracked, and improved.

Brief Description: Guide the chef through the steps to prepare a recipe.

Trigger: The chef begins to prepare a dish.

Type (mark one): X External ___ Temporal

Relationships:
Association:
Individual chef
Include:
Add recipe data to a database.
Extend:
Record recipe anomalies and changes.
Generalization:

The Normal Flow of Events:
A list of ingredients and the average cooking time is presented to the chef before they begi
preparing the food. Once they start, a stopwatch measures the average time needed to pre
the recipe. The application walks them through the recipe step-by-step until the dish is
complete. At that time, the stopwatch stops.

Sub-flows:
1. A list of ingredients and the average cooking time this recipe will take is retrieved from th
local database, and shown to the chef.
2. The chef begins the recipe.
3. A stopwatch measures the average time that the recipe takes.
5. The chef is then given the recipe step-by-step, using built-in timers and instructions.
6. The chef finishes the recipe and stops the stopwatch.
7. The quality and yield of the food compared to other preparation instances is noted by the
chef, as they desire.

Alternate/Exceptional Flows:
If, after step 3, the chef takes a break
>3.1 Pause the stopwatch.
If after step 2,
The chef stops making the recipe.
>2.1.1 Ask the chef if they wish to register this as a failure or drop it as a non-issue.
>2.1.2 Record the recipe as a failure in the database if required. Otherwise, drop all data.
The chef makes an error or deviates from the recipe.
>2.2.1 Have a button for the chef to press, allowing them to note the deviation from the rec

2.2.2 Continue as before.

If, at step 7, the chef decides not to enter specific information,

7.1 Ensure that the non-recorded information is not displayed in the official statistics.

Special Requirements:

Information verification

1. Ensure that blank statistics submitted to the database is not counted in official statisti
   (e.g., A blank average time doesn’t bring down the overall average completion time)

To do/Issues:

Integrations

1. Develop an integrated timer into the recipe.
2. Make sure that the chef can take breaks or cancel the recipe.
3. Develop a way for the chef to enter a deviation from the standard recipe.

Use Case Name: Record recipe anomalies and changes	**ID**: 3	**Importance**: Must Have
Primary Actor: Individual chef	**Use Case Type**:	

Supporting Actors: N/A

Stakeholders and Interests:
Restaurant executives – If applicable, they want better recipes for their customers.

Brief Description: By accident or experimentation, the chef notes a change in how the rec prepared.

Trigger: The chef makes a mistake or experiments.

Type (mark one): X?__ External ___ Temporal

Relationships:

Association:
Individual chefs

Include:
Add recipe data to the local database.

Extend:

Generalization:

The Normal Flow of Events:
The chef, either by accident or on purpose, modifies a recipe. The application allows them t note the change and how it tastes after the recipe is complete. If the chef finds it beneficial, then a variation on the recipe is added.

Sub-flows:

1. The chef changes the recipe in some way.
2. The chef then hits a button to note what the change was.
3. The recipe instructions continue as usual.
4. The chef is then asked if they wish to create a new recipe variation.
5. The chef writes new instructions for the variation and the steps in which it is changed
6. The variation is stored and sent to the local database.

Alternate/Exceptional Flows:
If at step 2, the chef doesn’t have time to note the change,
2.0 The chef makes a “Quick note,” marking where the recipe changed but not what chang
This function is invaluable when a hot pan is on, or the chef is short on time.
If, at step 4, the chef denies making the variation,
4.0 All variation comments and data are dropped.
If, at step 5, the chef does not have time to record the variation,
5.0 The comments are saved for another time temporarily so the chef can record the
information later.

Special Requirements:

Temporary storage

1.0 The chef needs to be able to store instructions and notes temporarily so that informatio is not lost.

To do/Issues: N/A

Use Case Name: Add recipe data to the local database	**ID**: 4	**Importance**: Must have
Primary Actor: Individual chef	**Use Case Type**:	

Supporting Actors: N/A

Stakeholders and Interests:
N/A

Brief Description:
The chef writes information to their local database.

Trigger:

Type (mark one): X__ External ___ Temporal

Relationships:
Association:
Chef
Include:

Extend:
Generalization:
Prepare recipe.

The Normal Flow of Events:
After finishing a recipe, the chef writes the data in the local database.

Sub-flows:
1. The chef finishes a recipe and enters extra data about success rates and variations.
2. The chef hits “complete”
3. Data is written to the local database.

Alternate/Exceptional Flows:
If, during step 3, there isn’t enough room on the device,
3.1 – Alert the user that their space has run out. Offer to either delete the oldest file or dele
the current recipe information.

Special Requirements:
Data storage
1. Ensure that there is a database that can handle the variations and recipe data.

To do/Issues:
N/A

Use Case Name: Import recipe.	**ID**: 6	**Importance**: Should have
Primary Actor: Individual chef, restaurant executive	**Use Case Type**:	

Supporting Actors:
N/A

Stakeholders and Interests:
Recipe sites and authors – Have a stake in ensuring that they receive proper credit and thei copyright is not violated.

Brief Description:
An individual chef or restaurant executive selects a recipe to add to their selection. They sc the cookbook or enter the recipe website link into the application, which converts it to our recipe format.

Trigger:

Type (mark one): X External ___ Temporal

Relationships:

Association:
Restaurant executive
Individual Chef

Include:
Create new recipe

Extend:

Generalization:

The Normal Flow of Events:
The Executive or Individual chef finds a recipe they wish to import into our application. They scan the recipe, or they submit a link for parsing. Our application finds the critical details in recipe, including ingredients, steps, and recipe tips. The application then creates a new reci based on that data.

Sub-flows:
1. A recipe is found that might be beneficial
2. The recipe is scanned or parsed from its respective media form.
3. The recipe is converted into a new recipe for storage.
4. The new recipe is added to the database.

Alternate/Exceptional Flows:
If, during step 2, the website or book is under copyright protection,
2.0 Alert the user with a warning to only copy legally available information.
2.1 Show other warnings or even deny the copy if legally prosecutable in that area.

Special Requirements:
Legal

It might be necessary to ask the user which state they reside in to determine legal jurisdiction.
To do/Issues: **1.** Determine what recipes and works fall under copyright protection in areas of the United States.

Use Case Name: Create a recipe.	**ID**: 7	**Importance**: Must have
Primary Actor: Chef, Restaurant Executive	**Use Case Type**:	
Supporting Actors: N/A		
Stakeholders and Interests: FDA – Ensures restaurant safety and food preparation regulation.		
Brief Description: A chef or executive adds a new recipe to the database.		
Trigger: A new recipe is found to add to the database. **Type** (mark one):X_ External ___Temporal		
Relationships: **Association**: Chef Restaurant executive **Include**: **Extend**: **Generalization**: Import recipe		
The Normal Flow of Events: A chef or restaurant executive notices a new recipe for their business. The executive or che notes the necessary steps, ingredients, tips, and other information. The recipe is then sent t local database, which can be synced to a company database if needed.		
Sub-flows: 1. A recipe that would benefit the executive or chef is found or made. 2. The chef or executive specifies the recipe's name, steps, ingredients, and expected yield 3. The recipe is stored in a local database.		
Alternate/Exceptional Flows: If, during step 3, an executive or employee fills out the recipe, the local database can be sy with the company database.		
Special Requirements:		

N/A
To do/Issues: Data 1. A format for recipe data needs to be created.

Use Case Name: Update recipe.	**ID**: 9	**Importance**: Must have
Primary Actor: Individual Chef, Restaurant Executive	**Use Case Type**:	

Supporting Actors: N/A

Stakeholders and Interests:
FDA – Wants to ensure food safety requirements are met
Restaurant executives – want food to be of high quality
Customers – Desire high-quality and safe food.

Brief Description: A recipe in the local or company database is updated.

Trigger: The executive or chef decides to update a recipe.

Type (mark one): X External ___ Temporal

Relationships:

Association:
Restaurant executive
Individual chef

Include:

Extend:

Generalization:

The Normal Flow of Events:
A recipe is selected for updating by the executive or chef. They decide to change ingredient preparation steps and to invalidate previous statistics. They also determine if they want to k the old version as a variant.

Sub-flows:

1. The executive or chef decides to update their local or company recipe database.
2. The user in question specifies the updates they wish to make, including ingredients and preparation steps.
3. The user decides whether to remove local or company-stored details based on preparation time and yield.
4. The user then decides whether to keep the old version and its variants.
5. The database is updated accordingly.

Alternate/Exceptional Flows: N/A

Special Requirements:
Data

1. The application must be able to refactor recipes into variants of variants.

To do/Issues: N/A

Use Case Name: Purchase company database access **ID**: 11 **Importance**: Should have

Primary Actor: Restaurant Executive **Use Case Type**:

Supporting Actors: N/A

Stakeholders and Interests:
Company – This use case is the main profit motive.
Chefs – Stake in the information provided.

Brief Description:
The restaurant location purchases a license key for their chain, allowing internet-enabled database access.

Trigger: The executives decide to purchase a recipe database service.

Type (mark one): X_ External ___ Temporal

Relationships:
Association:
Company
Restaurant executive
Include:
Extend:
Generalization:

The Normal Flow of Events:
The restaurant executive for the location decides to purchase our recipe database solution. They send an appropriate payment, and we provide them with a license key for employees a key for executives.

Sub-flows:
1. The restaurant executive for a location decides to purchase our product.
2. Our company charges an appropriate sum for the service.
3. A license key to sync to our database servers is provided for the executive.
4. A license key for employees to sync with our database servers is provided, and the executive can hand it out to his staff as he sees fit.

Alternate/Exceptional Flows:
If, after step 4, the executive requires assistance with setup,
4.0 Offer immediate support via a helpline, with the new customer as a priority.

Special Requirements:
Helpdesk requirements
1. A helpdesk is required to assist chefs and executives using our product.
2. The helpdesk must have a priority queue for new customers – remember, they might st be able to refund the product, and we don't want them to give up on it early.

Security requirements

1. A secure license key system needs to be in place.
2. Connections to company databases must be verified.

Tax requirements

1. As with any payment, there is tax legislation that needs to be followed.
2. There is a requirement to determine the location’s state of operation so taxes can be collected.

Privacy requirements

1. Private information about the company must be stored and collected with any transaction.

To do/Issues:

Legal

1. Legal consultation is required to determine data privacy compliance and tax complianc

Payment

1. Payment information must be secure.
2. The payment process and license keys have to be secure.
3. Determine account requirements for businesses (What information is needed)
4. Determine payment model (Frequency, amount, cost/benefit)

Use Case Name: Log in to the chain network | **ID**: 12 | **Importance**: Should have

Primary Actor: Restaurant Executive | **Use Case Type**:

Supporting Actors:

Stakeholders and Interests:
Our company – has a stake in ensuring our services are accessed securely.
Employee chefs – have a stake in ensuring that their data is appropriately entered and hand

Brief Description:
An executive logs in to the employee recipe network to view statistics or register a chef.

Trigger: The restaurant executive attempts to log in.

Type (mark one): _X_ External ___ Temporal

Relationships:

Association:
Restaurant executive

Include:

Extend:
Visualize chain cooking data
Register chef as an employee of restaurant
Import recipe
Update recipe
Create recipe

Generalization:

The Normal Flow of Events:
The executive decides to log into our service. They enter their credentials and view the available data.

Sub-flows:
1. The executive attempts to log in to the system by clicking a “business login.”
2. They enter their username and password.
3. They enter a 2FA Authentication.
4. Access is granted.

Alternate/Exceptional Flows:
If, during step 2, the executive hasn’t signed in on this device before,
2.0 The executive is prompted to enter their license key.
If, during any of these steps, the executive encounters problems,
>0.0 They call the available helpdesk for support.
>0.1 Support is provided, and the problem is fixed.

Special Requirements:
Helpline

1. There must be an available helpdesk for assistance with logging in.

Security

1. We must securely store emails, passwords, and 2FA secrets.

To do/Issues:

1. Figure out how to store passwords securely in the database.
2. Only allow the executive access to their section of the database. They should have no access to any other customer database.

Use Case Name: Register chef as employee of a restaurant	**ID**: 13	**Importance**: Should have
Primary Actor: Restaurant Executive	**Use Case Type**:	

Supporting Actors:
Chef – Enters their company license key

Stakeholders and Interests:
N/A

Brief Description:
The restaurant executive adds a chef as an employee of the chain.

Trigger: An executive marks the chef as part of the chain.

Type (mark one): X External ___ Temporal

Relationships:
______ **Association**:
Individual chef
______ **Include**:
______ **Extend**:
______ **Generalization**:

The Normal Flow of Events:
The executive opens the application and views chefs who wish to register with their restaur
They select one of their employees, and that employee's local database is set to be synced
the company database.

Sub-flows:
1. After the executive logs in, they can view chefs who request to register as company chef
have entered the company license key.
2. The executive then selects a chef they recognize to add.
3. The chef is added, and their database is marked to be synced with the company server.

Alternate/Exceptional Flows:
If at step 2, the executive cannot find any chefs to add,
2.0 Break the flow.

Special Requirements:
1. Mark each person trying to register with a name or email address.

To do/Issues:
N/A

Use Case Name: Visualize chain cooking data.	**ID**: 14	**Importance**: Should have
Primary Actor: Restaurant executives	**Use Case Type**:	
Supporting Actors: Individual chefs – provide the data to be visualized		
Stakeholders and Interests:		
Brief Description: The restaurant executive logs into the system, and inspects aggregated cooking data.		
Trigger: **Type** (mark one):X External ___ Temporal		
Relationships: **Association**: Individual chefs. **Include**: **Extend**: **Generalization**:		
The Normal Flow of Events: After logging in, the executive views charts of recipe data. They can see the average time a takes and how many times the dish was prepared over a period of dates. They can then issu changes to recipes or remove recipes from the menu.		
Sub-flows: 1. The executive views charts of recipe data. 2. The executive can then change items on the menu or change the recipe of a dish. 3. After entering price data for ingredients, cost, and salary, the executive can view the pro margins for dishes. 4. They can also view the average time per cooked unit and yield data.		
Alternate/Exceptional Flows: N/A		
Special Requirements: Integration 1. The employees should be able to see which items are currently on the menu.		
To do/Issues:		

Use Case Name: Store cooking information in company database	**ID**: 15	**Importance**: Should have.

Primary Actor: Individual chef, company executive	**Use Case Type**:

Supporting Actors:

Stakeholders and Interests:

Our company - Is interested in recipe data being stored reliably in our systems.

Brief Description:

The company executive or individual chef syncs their modified recipe information and statis with the company database.

Trigger: The executive or individual chef finishes creating data and is logged into a "keyed account.

Type (mark one): X External ___ Temporal

Relationships:

Association:

Restaurant executive

Individual chefs

Include:

Extend:

Generalization:

The Normal Flow of Events:

After the chef has finished preparing a recipe, or an executive finishes making edits to recip the application offers to sync with our managed database.

Sub-flows:

1. An executive or chef finishes editing or creating recipe data, such as variations, time to complete, or yield.
2. The information is set to the local database.
3. The information is synced to the company database, automatically entering information like time-to-complete and average yield.

Alternate/Exceptional Flows:

If, during step 3, the chef came up with a variation on a recipe,

3.0 The variation is synced with the database but is not marked to be sent to all locations.

Special Requirements:

To do/Issues:

Security

1.0 Via the license key, the employees should have an authentication procedure to ensure they are part of the restaurant chain.

Use Case Name: Perform database maintenance	**ID**: 16	**Importance**: Should have
Primary Actor: Hosting company	**Use Case Type**:	
Supporting Actors:		
Stakeholders and Interests: Restaurant executives – Want to ensure their recipe data is safe and secure. Individual chefs – Wants to make sure that all data is aggregated.		
Brief Description: A company employee logs in to the databases to perform maintenance or upgrades.		
Trigger: Maintenance is required. **Type** (mark one): __ External _X_ Temporal		
Relationships: **Association**: Hosting company **Include**: **Extend**: **Generalization**:		
The Normal Flow of Events:		
Sub-flows: 1. A notification will be given to customer companies of scheduled maintenance, and the planned timeframe will be communicated. 2. A backup is performed before maintenance in case of failure. 3. Maintenance begins at two hours past midnight in the UTC-8:00 time zone. This timeframe is set so that all restaurants will hopefully be empty or closed, and maintenance will not affect service.		
Alternate/Exceptional Flows: If, during step 2, customers send in complaints over service or special occasions, 2.1 Negotiate a new maintenance date that does not interfere with the occasion, if possible		
Special Requirements: N/A		
To do/Issues: Maintenance 1. Establish a standard procedure for maintenance.		

Use Case Name: Conduct security inspection. **ID**: 17 **Importance**: Should have

Primary Actor: Hosting company **Use Case Type**:

Supporting Actors:

Stakeholders and Interests:
Restaurant executives – Want company information secured.
External auditors – Are receiving compensation for services.

Brief Description:

Trigger: Regular inspection date comes.

Type (mark one): __ External __X_ Temporal

Relationships:
Association:
Hosting company
Include:
Extend:
Generalization:

The Normal Flow of Events:
A monthly or annual inspection date comes around. A company auditor or external expert te the system's security and access control and then reports the results for maintenance or recording.

Sub-flows:

1. A regularly scheduled inspection date comes.
2. Inspection is conducted either by an independent or company auditor.
3. Security controls and vulnerabilities are tested.
4. Access control is tested.
5. A report is created by the auditor and stored for later action.
6. Any critical security issues are fixed.
7. Non-critical notifications are stored for later maintenance.

Alternate/Exceptional Flows:
N/A

Special Requirements:

To do/Issues:
Security consultants

1. Figure out which consultants and testers to hire or contract.

Legal

1. Find out the best possible testing deadline and regular schedule for security inspections

Use Case Name: Register as a chef of a chain or business

ID: 18

Importance: Should have

Primary Actor: Individual chef

Use Case Type:

Supporting Actors:

Stakeholders and Interests:

Restaurant executives – Need to register their employees to collect statistics.

Hosting company – Wants chefs to be able to register with restaurant locations.

Brief Description:

The individual chef enters a license key for their location and attempts to register as a chef the restaurant location.

Trigger:

Type (mark one): X External ___ Temporal

Relationships:

______ **Association**:

______ **Include**:

______ **Extend**:

______ **Generalization**:

The Normal Flow of Events:

The individual chef receives a license key for the restaurant location from the restaurant executive. The chef then enters it into the application, and the application sends a request t the executive to add the chef to the organization.

Sub-flows:

1. The executive gives the chef of the restaurant the “employee” license key.
2. The key registers the individual chef’s application so the chef can sync with the compa server.

Alternate/Exceptional Flows:

Special Requirements:

To do/Issues:

Use Case Name: Create a timer for cooking | **ID**: 19 | **Importance**: Could have

Primary Actor: Individual chef | **Use Case Type**:

Supporting Actors: N/A

Stakeholders and Interests:
Customer – They want their food prepared safely and their dish to remain uncharred.

Brief Description:
A chef sets an embedded timer for cooking a meal.

Trigger: The chef sets a timer embedded in the recipe.

Type (mark one):_X_ External ___ Temporal

Relationships:
______**Association**:
______**Include**:
______**Extend**:
______**Generalization**:

The Normal Flow of Events:
As the chef goes through a standard recipe, the cooking time for a particular portion of a dis specified. The chef clicks on an embedded timer when the dish goes in to cook, then waits f the timer to elapse.

Sub-flows:

1. The chef reaches a point in a recipe where a timer must be set.
2. An embedded timer is generated at that point in the recipe.
3. The chef lets the timer start.
4. Once the timer elapses, alert the chef.

Alternate/Exceptional Flows:
If, during step 1, the chef does not wish for timer data to be aggregated into the overall coo time,
1.1 The chef may select an option to subtract the cooking time from the overarching stopwa
If, during step 2, the embedded timer is generated incorrectly
2.1 The chef is given the option to remove or disable the timer entirely from the recipe.
If, during step 3, there are other parts of the recipe that can be completed in the time given
3.1 Prompt the chef to move to a different subsection of the recipe simultaneously or ignore the opportunity.
If during step 4, the chef is in a noise-sensitive area, or do-not-disturb is enabled,
4.1 Consider an alternative method to alert the chef, such as text messages, vibrations, or flashing lights.

Special Requirements:
Organization

1. Recipes require breakpoints between different recipe components, so the chef can jur around while a timer runs.
2. The chef must find it easy to switch between components while preparing food.

To do/Issues:
Recipe data storage

1. The recipe requires a sort of “branching path” to determine when separate componen are being worked on to implement parts of this functionality.

Use Case Name: Visualize individual cooking data **ID**: 20 **Importance**: Must have

Primary Actor: Individual chef **Use Case Type**:

Supporting Actors: N/A

Stakeholders and Interests: N/A

Brief Description:
A chef visualizes his cooking data for individual recipes.

Trigger:

Type (mark one): X External ___ Temporal

Relationships:
______ **Association**:
______ **Include**:
______ **Extend**:
______ **Generalization**:

The Normal Flow of Events:
The individual chef decides to review his personal aggregated data on individual recipes. He
view statistics per recipe. The statistics include charts of yields, average cooking time, aver
taste, and the general resource cost.

Sub-flows:

1. The chef at the main menu presses an option to view recipe statistics and adjust.
2. The chef is shown a list of recipes and is allowed to select one.
3. After selecting the recipe, they are greeted with a list of statistics, including when the recipe was last created, the average cooking time, a graph of cooking time, and avera yield.

Alternate/Exceptional Flows:
If, during step #1, the chef decides to adjust the recipe,
1.0 Continue to use the case “Update recipe”.

Special Requirements:

To do/Issues:
Visualization

1. Consider the best method with which to visualize cooking data.

6.0 System Evolution

As the project stands now, it is worth noting that the initial minimum viable product is
required to optimize recipes for an individual user. Functionalities such as corporate databas
management and employee connection should be considered when creating the minimum vi
product, but they are not strictly required. So, the obvious next step in the evolution would b
this functionality.

The best-case scenario for the evolution of this product is to become a sort of source c
for cooking projects. If one thinks about source control in the context of code, cooking and so
iterations hold similar patterns, albeit with wildly different formats. Both require variety in ha
with code needing variations in computers and cooks having variations in ovens. Testing for
recipes is slow and complicated, and the widespread correction of instruction sets takes subs
effort. Food preparation, however, is at a disadvantage because iteration is much slower. Imp
suggestions can’t easily be shared, communication of alternative preparation methods is ofte
by proximity, and updating a recipe for a large group is very difficult. So, why not utilize an a
to share improvements? This application aims to improve cooking for an individual user: Can
data be used to improve the recipe for all chefs?

~~This section answers the question, "What about the future?" It is a brief discussion of~~ any desired sys
~~features that will not be part of the MVP initial version (see the corresponding comment~~ in the descri
~~Section 3.0). Also, include any planned or recommended upgrades to hardware/software~~ for continue
~~use or expansi~~on
~~7.0 Conclusions and Recommend~~ations.

7.0 Conclusions and Recommendations

This document has outlined a full proposal for creating a cooking optimization application. It co
complete requirements for the iteration of the cooking process, creating recipe data, and interlinking
related business, if necessary. As it is currently designed, entering recipe data into the application ar
through a specialized and modified recipe step-by-step is straightforward. When mild changes or imp
are discovered, the chef can note and correct them. This application allows a chef to branch off and i
original recipe. The cooking optimization application is meant to cover the full range of cases so a ch
improve, optimize, and iterate on their cooking process, allowing for every detail and change to be no
later reference.

In response to this document, the recommendation is to create and test a basic prototype to e
idea has potential. If the concept holds and dishes become greatly optimized, then expansion into the
sector as described is possible.

~~These may be organized as two separate sections or two *distinct* parts of section 7. In e~~ither case, th
~~place where a section introduction is neither needed nor re~~commended.

Appendices

~~Include (at least) a copy of any completed questionnaires or surveys, plus any other~~ applicable refere
~~material~~s.

~~I’m afraid I don’t understand the prompt fully and will improve upon it in my next submission~~. I don’t
~~have filled out any questions or surveys and have linked all reference materials~~ in the bibliography.

Glossary

Term	Definition
Closed source	A term meaning that the code for an application is not released to the public;, it remains the property of its owner.
Cross-Platform Application	An application that can work across fundamental application bases such as Linux, Windows, and macOS.
Proprietary	Something that is used, produced, or marketed under exclusive leg right of the inventor or maker. (*Definition of Proprietary*, n.d., sec.
~~Cross-Platform Application~~	~~An application that can work across fundamental application bases such as Linux, windows, macOS.~~
United States Food and Drug Administration (FDA)	The federal agency responsible for the regulation of food, drugs, medical devices, and other products
User Interface (UI)	The presentation layer of an application – the interface that the us interacts with.

~~make sure it covers your entire set of artifacts before sub~~mitting Part 2.

Bibliography

Adobe photoshop drawing tool. (n.d.). Adobe Photoshop. https://www.adobe.com/products/photoshop.html

Blender Open-source modelling and animation tool. (n.d.). Blender.Org. https://www.blender.

Commissioner, O. of the. (2024, April 29). *U. S. Food and drug administration*. FDA. https://www.fda.gov/

Definition of proprietary. (n.d.). Retrieved May 5, 2024, from https://www.merriam-webster.com/dictionary/proprietary

Larman, C. (n.d.). *Applying UML and patterns: An introduction to object-oriented analysis and iterative development* (3rd ed). Prentice Hall PTR, c2005.

Purpose-built databases on aws | amazon web services. (n.d.). Amazon Web Services, Inc. Re May 5, 2024, from https://aws.amazon.com/products/databases/

Terms of service—Dotdash meredith. (n.d.). Retrieved May 5, 2024, from https://www.dotdashmeredith.com/brands-termsofservice

Veenstra, K. (n.d.). *Lecture Notes*.

~~Be sure to update the Bibliograph~~y for Part 2.

~~For Part 2 subm~~ission*:*

- ~~All the above sections are to be included as a single~~ document.
- ~~All corrections and suggestions from the grader are to be addressed, as are all customer~~ question ~~in the document rev~~iews.

~~Any changes to the first submission's functional requirements or feasibility study res~~ulting from w ~~learned while creating the second half of the document should be made in each~~ section. Use "*Tra* ~~*Changes*" in MS Word so all the changes from Part 1 to Part 2 are denoted. **Your**~~ **submission ma** ~~**returned to you if you do n**~~**ot do this**~~. Remember that the reader can always turn off~~ "*Track C* ~~a clean Part 2 document artifact. It would be best to do this on a copy of your document~~ before su ~~to Canvas to ensure that the clean version formats the way you want it to look. But~~ be sure to sub ~~Canvas with "Track Changes" turned on so your *Part 1 -> Part 2* chan~~ges are apparent.

~~That's~~ it!

~~Good luck! I look forward to reading both parts of your Sys~~tem Proposal!
~~-Professor~~ Andy