Pizza with a Purpose (PWAP)

System Proposal

Prepared for:

Mr. Taiwoo Park, Disciples' Pizza

Prepared by:

Ben Edmonds, Software Developer

Nile Software Development (NSD)

Executive Summary

Nile Software was contacted by Mr.Park of Disciples' Pizza to build a system that allows them to provide a delivery service for their pizza co-op. The main objective of this system is to allow for the co-op to be able to deliver even with their food trucks changing locations throughout the day. The system will help in all areas of the co-op including ordering, making, and delivering pizza, as well as other administrative needs like ingredient availability. Development of the system will benefit Disciples' Pizza, the customers in the surrounding area, and Nile Software.

After looking critically at the project, Nile Software has decided that this system would be feasible to make and would provide great value to all stakeholders. To ensure that the system is produced effectively Nile Software would ask that Disciples' Pizza continue to communicate about requirements and deadlines for the system, so that the project is created on-time, within budget, and free of errors.

1.0 Introduction Overview

1.1 Problem Statement

Disciples' Pizza is a co-op of Christian pizza trucks that are looking to provide a quality delivery service to their customers. For this to occur they need a delivery system that can handle many different unique facets of their co-op, the main obstacle being that the pizza trucks move to different locations throughout the day. This system (PWAP) will have many different sides for customers, pizza truck works, and the couriers (delivery drivers). Disciples Pizza is seeking out Nile Software Development, an experienced company, to produce its pizza delivery system.

1.2 Project Vision and Scope

The goal of Disciples' Pizza is to provide local, high-quality pizza, while also spreading God's Love. The success of their co-op means that they are able to help many in their community locally and internationally through missions. With an advanced delivery system in place, they would be able to reach more customers and provide pizza to even more people. The plan that NSD has in place is to construct a delivery system that will allow customers ease in pizza purchasing, give owners all the information to make the pizza correctly and on time, and allow the couriers to easily get where they need to go. This will allow Disciples' Pizza to grow as co-op and better meet their area.

1.3 Requirements Summary

The PWAP system will have many different requirements under the different sections:

Section 1: Customer Side

- The system must allow customers to specify the dough, size, base sauce, cheese, and other toppings
- PWAP should create an efficient ordering process that includes presets of specialty pizzas, breadsticks, wings, and drinks
- The system must support credit card transactions, and specific food voucher cards
- The system must offer account creation if user wants to save information to make purchasing easier in the future

Section 2: Food Worker/Owner Side

- The system must notify workers within 30 seconds of new an order being placed
- PWAP should provide a clean and organized way to view ongoing and pending orders, organizing the orders based on priority and time elapsed
- PWAP must allow for orders to be put into the system by the owners when an in-person order is made
- PWAP should provide a feature for recording voice messages for a customer
- PWAP must offer the ability for owners to change ingredients based on availability and season
- The system should access storage of food voucher information in under 15 seconds, to ensure that food vouchers are used correctly and erased after usage

Section 3: Delivery Side

- The system must display list of delivery tasks, and current destination to couriers
- PWAP must update current destination of couriers based on the stage in delivery (if the pizza has been delivered the courier may need to return to a different truck than from where they left)
- The system must integrate with outside navigation service to provide direction services to couriers

1.4 Stakeholders and Interests

PWAP will have an effect on a wide range of people and groups. **NSD** is planning on continuing to build and improve the software into the future, and thus have a stake in the success of the development and eventual use of the software. **Disciples' Pizza** has the potential to improve their efficiency and profits based on how well the system accomplishes its goals, so they have a large stake in the project as well. The **users of the software** such as **food truck workers** and **delivery drivers** will be interacting with the system daily and will rely on it to do their job, so the success of the system will have a direct impact on how well they are able to do their job. The **customers of Disciples' Pizza** are also stakeholders since the PWAP system will improve their access and ease of purchase. Since Disciples' Pizza has a large impact on the **local community** and even **internationally community** through missions so the potential for growth with PWAP could have impacts that reach all over the globe.

1.5 Expected Benefits and Costs

1.5.1 Benefits

NSD will provide a system that will benefit Disciples' Pizza by:

- Improving their ability to reach their customers
- Increasing the amount of business Disciples' Pizza can handle
- Support owners by allowing them to quickly change the availability of ingredients
- Allow them to hire couriers to expand their overall co-op
- Grow Disciples' Pizza in ways that increase the amount they give back to the community (locally and internationally)
- Provide a system that is easy to use and implement into the existing food truck operations

1.5.2 Cost

The PIZZA WITH A PURPOSE system will have cost due to the following factors:

- Creation of system
    - Salaries of developers for the period of the project
    - Integration of existing software including third-party navigation apps
- Maintenance of system
    - Support NSD to solve future bugs and provide improvements based on feedback
- Required storage space on a cloud server that can be accessed by all the different food trucks, food couriers, owners

1.6 Constraints

The system will be constrained by the following factors:

- Although there is no explicit budget given to NSD for the creation of this system NSD is well informed that Disciples' Pizza is co-op that runs on a tight budget with a focus on giving back, and thus does not have an unlimited budget to spend on the creation of PIZZA WITH A PURPOSE
- There is no overtly tight time schedule, NSD will work the create the system quickly and efficiently, providing core sales functionalities first, then continuing on to the rest of the features
- The system needs to be compatible with many different platforms including Windows, iOS, and Android being the primary use cases
- Internet connection will be essential to make the various different pieces communicate with each other including all of the different food trucks, owners, couriers, and customers

- Some hardware will be required for the system to work as intended, but existing hardware should be compatible with the system
- Training for the software will be needed so that people understand how the system works effectively and to ensure that problems are minimized

1.7 Recommendation

The development of the PWAP system would be beneficial for both NSD and Disciples' Pizza. NSD recommends that Disciples' Pizza continue to communicate with NSD to create solid dates for system iterations to be created by, and ultimately a final date for a fully functional system. Disciples' Pizza should also pay careful attention to the requirements section of this document (4) and ensure that any changes or additions are discussed with NSD before development begins. NSD and Disciples' Pizza will communicate more about both recommendations to ensure the successful development and operation of the PWAP system.

1.8 Document Overview

The sections that follow in this document:

2. System Initiation

- Includes the system request and Nile Software Development response

3. Feasibility Assessment

- Outlines and analysis of the technical, resource, schedule, organizational, and legal/contractual feasibility

4. Requirements Definition

- Overview of PWAP functional and nonfunction requirements

5. Requirement Model

- Use cases diagram and descriptions for use cases

6. System Evolution

- Outline of future plans and features of PWAP

7. Conclusion and Recommendations

- Final thoughts and recommendations for the success of PWAP and the system development

2.0 System Initiation

2.1 System Request

October 11, 2019

SYSTEM REQUEST – Disciples' Pizza

Project Sponsor

Pizza with a Purpose: Mr. Taiwoo Park

Representing: Wanderer's Tools

Phone: x7258E-mail: twp@spu.edu

Opportunity Statement:

Disciples' Pizza is the new brand of the co-op of Christian wood fired pizza trucks in the greater Seattle area. Our trucks have our own unique style and taste and thus been loved by local residents for years. Our trucks move to many different places every day, so it has been challenging for us to provide our customers with a quality delivery service. However, we would like to start a delivery service by hiring food couriers (i.e., delivery persons). It is a very interesting setting – our kitchens and food couriers both move! We are going to need some technology to make *this* happen!

Proposed Product:

Background and Context:

Disciples' Pizza isn't a "company" per se, but rather a group of food trucks and a group of food couriers to bake and deliver pizza to local customers. We have 30 pizza trucks in the greater Seattle area, mostly on the west side of Lake Washington, and will have 30-40 food couriers in the near future. Most of our trucks moves every day, or even in the middle of a day depending on neighborhood and/or sports events. For its brand and to support our upcoming mission of international missionary and local church support, we would like to have our own delivery system and mobile apps for online orders.

We would like to take a full advantage of mobile devices for trucks, food couriers, and customers. Specifically, most of trucks already have tablets for credit card transactions. Food couriers have smart phones, and of course, our customers do. Our dream is that once our customer is craving for pizza, s/he opens our app, and make an order, then the nearest pizza truck receives the order. Also, one of our food couriers nearby will receive a delivery request, and the courier will deliver the pizza to the customer. We hope that all these processes would work smoothly to find the best truck and courier so that we can be always cost-effective and maximizing customer satisfaction.

Initial Vision and Scope:

➢ *Online Order and Status Tracking*

Our customers need to be able to order pizza specifying its dough, size, base sauce, cheese, and toppings. We also offer a set of specialty pizzas with presets, as well as bread sticks, wings, and drinks. We think to support credit cards as main payment method, while supporting our own food voucher cards for ones who are in need. Once a customer makes an order from our app, it automatically dispatches the work to one of our trucks --- perhaps the closest one -- - and one of our couriers. Our app needs to be able to show the current status of the order, among preparation / in oven / on its way, and the courier's location in case when the pizza is on its way. After delivery, our customers can hear a message of blessing from the truck owner who made the pizza.

➢ *Pizza Truck Order Handling*

The app in our food trucks is to notify our food truck owners of new orders, and show all ongoing and pending orders on the screen. Most of our food truck owners make multiple pizzas at the same time, while handling customers in-person, so the app needs to give the information as clear as possible. Our owners are supposed to change the current status of the orders, and it would be nice if they can record a voice message (to spread the love and blessing of God) for a customer when they hand the pizza to a courier. Also, our owners should be able to create a new order made in-person through the app.

➢ *Courier Support*

Our couriers will have a smartphone app for delivery information. It will show a list of delivery tasks, as well as the current destination. If a courier is with pizza, the destination is to be delivery address, or a pizza truck for the next delivery otherwise. Our couriers need navigation service for their pickup and delivery.

➢ *Menu change and administration support*

We may want to change ingredients and specialty pizza styles depending on availability and season, hopefully through web browsers. Also, we would like to handle the food voucher information.

➢ *Sales summary and statistics*

Stakeholders Identified:

- Pizza truck owners – who would like a more effective way to receive delivery orders and bless the customers.
- Food couriers – who would like make more efficient trips and easy 'where-to-go-next' information.
- Our customers – people who will enjoy our fresh pizza delivered quickly.
- You – as our partner.

Expected Benefits:

- Opportunity 1 – Fresh pizza, made in my neighborhood, delivered quickly.
- Opportunity 2 – Improve our sales experience and increased revenue.
- Opportunity 3 – Spread God's love and blessing.

Special Issues or Constraints:

We're not made of money. Members are willing to invest in the new equipment and programs, but we're talking a few hundred (not thousand) dollars each from maybe 30 owners. Similarly, I'm not sure everyone is going to be excited to run out and by a new phone or tablet to manage sales or use this system. (Although maybe we could handle that with a few purchases "by the co-op".)

Also, our members represent a wide range of computer skills and types of computers they know and use. Some are really into their smartphones and others aren't. Pretty typical people.

I don't think we are on a particularly tight time schedule. The core sales functionalities might be considered to be finished sooner, while others can be more down the road.

2.2 Sales Letter

October 21, 2019

Mr. Tiawoo Park

Wanderer's Tools

241 Miller St., Seattle, WA 98119

Disciples' Pizza System Request

Hello, Mr. Park. I have read over your System Request and think that there is great potential in the proposal. The idea of having the food trucks being able to deliver, even as they move from location to location using food couriers fits a great need in the market and opens even more opportunities for future development. When this system is developed it will expand the possibilities for growth for all parties involved.

Nile would love to work with you on the development of this system, especially since our company has very applicable experience. Nile has worked on the creation of systems that help with the delivery of goods for other businesses that were seeking a way to create their own online sales department. These systems had similar features to what Disciples' Pizza system needs, such as tracking the online ordering and status of the goods that were purchased. The Disciples' Pizza system will require more attention in that there are more moving pieces involved and that the delivery times will be much faster, but with this previous experience under our belt, Nile Software is more than up to the task. We at Nile, are also well versed in making systems that operate on all sorts of different platforms including iOS, Android, and PC. Your concern with not wanting to have to purchase new technology to make the system work is well deserved, but Nile should be able to overcome this hurdle and help produce a system that meets the needs of all who will use it.

Wander's Tools has a great track record of delivering systems that meet the needs of their clients. Nile would appreciate the opportunity to assist in the development of this new system and uphold this reputation. I hope to hear from you soon and begin to discuss how we can help the process.

Sincerely,

Ben Edmonds

3.0 Feasibility Assessment

3.1 Introduction

Section 3 outlines and summarizes NSD’s analysis of the feasibility of PIZZA WITH A PURPOSE. This is necessary to determine if PIZZA WITH A PURPOSE can be developed in a timely fashion, without unnecessary spending, that meets all of the requirements as described. The feasibility will be measured on the following scales:

Risk

1. Very Low: Possible issues or concerns are insignificant or unlikely to impact the project in a meaningful way
2. Low: Possible issues or concerns are mostly insignificant or will have minimal impact on the project
3. Medium: Possible issues or concerns could have an effect on the project, but the impact would not be large, or the possible effects are largely unknown
4. High: Possible issues or concerns are generally significant and could have an impact on the project
5. Very High: Possible issues or concerns are significant and are likely to impact the project in a meaningful way

Feasibility

1. Infeasible: Risks present suggest that the project is not worth attempting.

2. Feasible: Potential challenges exist but can be addressed by NSD to make the project successful

3. Ideal: No risk is present.

3.2 Feasibility Analysis

3.2.1 Technical Feasibility

PIZZA WITH A PURPOSE is technically **feasible** with a **low** risk rating.

- **User familiarity**: For the customer’s side most smartphone users will be familiar with delivery apps, but some sort of common questions section may be needed for those with less smartphone experience or expertise. Food truck works and owners may need some basic training to understand the software, but the interface should allow workers to easily get up to speed. Similarly, couriers may need some basic training, but will quickly become adapted to the system. Overall this section is **feasible** with **low** risk.
- **Analyst/Developer familiarity**: NSD has developed systems before that have some similar requirements or goals and the team is familiar with many different platforms that may or may not be needed for PIZZA WITH A PURPOSE. This section is **ideal** with **very low** risk.
- **Project size**: PIZZA WITH A PURPOSE will have many different parts (customer, food worker, owner, courier) that work together to form the whole system. Similarly, it has been expressed that having a core version with sales functionality

and then adding more features would work best for integration into Disciples' Pizza current system. By making use of development strategies that allow for iterative development the risk should be mitigated. This area is **feasible** with **medium** risk.

- **Project structure**: The requirements are subject to change as the project is in development. Disciples' Pizza has given a rough outline of requirements from the system proposal, but that could change in the future as the co-op grows and matures. That being said the overall structure of the system is well-defined even though some features or requirements could change down the road. This area is **feasible** with **medium** risk.

3.2.2 Resource Feasibility

PIZZA WITH A PURPOSE is resource **feasible** with a **low** risk rating.

- **People**: NSD is well staffed with enough members to take on a project of this size. This area is **feasible** with **low** risk.
- **Hardware**: NDS has hardware for the many different types of testing that will be done for a system like PIZZA WITH A PURPOSE. NSD will check with the owners of Disciples' Pizza to ensure that the system developed is compatible with the hardware currently owned by the co-op. Any future hardware purchases should be made known to NSD so that it can be ensured that the system will work on said hardware. This area is **feasible** with **low** risk.
- **Software:** PIZZA WITH A PURPOSE will need to be integrated with some third-party software specifically direction software for couriers. NSD will reach out to third-party direction software to ensure that PIZZA WITH A PURPOSE will work with them. This area is **feasible** with **low** risk.
- **Environment:** Due to similar experiences in the past NSD is capable of the task of developing PIZZA WITH A PURPOSE. Still, PIZZA WITH A PURPOSE will need to work for many different types of users and perform may different tasks if it is to be successful, so a small concern should be express that the features need are provided, and any future requirements are addressed quickly and efficiently. This area is **feasible** with **low** risk.

3.2.3 Schedule Feasibility

PIZZA WITH A PURPOSE is **feasible** with concern to schedule with **medium** risk present.

- Disciples' Pizza has not given a specific timetable for development but has made it clear that core sales functionality should be focused first so that the co-op can begin to integrate its existing model with PIZZA WITH A PURPOSE.
- NSD is currently working on several different projects but should be able to complete the project in a timely manner.
- NSD would like to meet with representatives for Disciples' Pizza on a semi-regular (monthly or bi-monthly) to discuss the development and create a rough timeframe that the system can follow.

3.2.4 Organizational Feasibility

PIZZA WITH A PURPOSE is organizationally **feasible** with a **low** risk rating.

- Disciples' Pizza owners and Mr. Park, the project champion, greatly wish for this system to be successful. When the system is finished it will allow for Disciples' Pizza to create more business and have a greater impact on the community as a whole.
- Employees of Disciples' Pizza (food truck workers, couriers, etc...) should see the value that PIZZA WITH A PURPOSE provides and quickly adapt to working with the system such that they can accomplish what they need to quickly and efficiently.
- NSD will make the interface easy to use so that users can quickly get up to speed with the software. This will include having fonts and sizes that are easy to read for users of any level and be optimized for use on many different potential devices that users may have. Some users may find it hard to adapt at first, but once the system is shown to be effective and provide benefits these users should have a more positive view of the system overall.

3.2.5 Legal and Contractual Feasibility

PIZZA WITH A PURPOSE is legally and contractually **feasible** with a medium **risk** rating.

Legal Feasibility:

- **Legal Concerns:** Information about the food trucks themselves, some financial records, and credit card information will need to be stored or accessed by PIZZA WITH A PURPOSE and will need to be kept secure and private. NSD will follow typical industry practices to ensure that all data is kept safe. NSD will try to keep security up to date in the future with updates to the system as deemed necessary. This area is **feasible** with **medium** risk.

Contractual Feasibility:

- The risk involved in the ownership rights of PIZZA WITH A PURPOSE is **very low,** as this will be negotiated by parties from both NSD and Disciples' Pizza in the near future as soon as the project is initiated. It is expected that NSD will own the source code of the project and that Disciples' Pizza will be allowed an unlimited number of licenses to use for their co-op.
- After the creation of the software, NSD will be continuing to provide maintenance and upgrades as deemed necessary by Disciples' Pizza. The specifics of this agreement will also be negotiated with NSD getting compensated based on what sort of maintenance is required. This area is feasible with **low** risk.

3.3 Additional Comments

- NSD is ready for the task of developing PIZZA WITH A PURPOSE and feels are experience is the past should provide us with a foundation to work from.
- NSD may deem it necessary to hire a consultant with experience of integrating third-party direction/navigation software.
- Training may be required for some individuals to ensure that there is a select group of individuals within Disciples Pizza who can teach the system to others.

3.4 Conclusion

PIZZA WITH A PURPOSE is a feasible project for NSD to take on and presents low to medium risk, with most of the risk involving narrowing down the requirements and seeing how requirements may change in the future. NSD would like to continue the process in a smooth and timely fashion so that the project may begin soon.

4.0 Requirements Definition

4.1 Introduction

Section 4 will cover the various requirements of PIZZA WITH A PURPOSE. This includes functional requirements that cover what the system will do and provide to the users of the system. This includes all of the different features that need to be present for the system to work as intended.

4.2 Functional Requirements

4.2.1 Customer Side

- Ordering (Use-Case 3)
  - The system must include an app to the customers that allow them to choose dough, size, sauce, cheese, toppings
  - Easily put in credit card information or food voucher card for payment
- Delivery Status Info (Use-Case 3)
  - The system must display the current status of the order (being prepared, cooking, in delivery, etc.)
- Communicate Message (Use-Case 6)
  - The system should allow customers to hear a message of blessing from truck owners when the pizza is at the delivery destination
- Account Creation (Use-Case 1)
  - The system must access stored customer information on a server to allow return customers to store information to make purchasing easier in the future.

4.2.2 Food truck workers/Owner side

- Current Order Display (Use-Case 4)
  - The system must notify the food truck workers of new orders, and displays all other orders that are being processed
- Status Change (Use-Case 4)
  - The system must allow for food truck owners to easily change the status of customers order so that customers can track the progress of the order through the customer side app
- Record Messages (Use-Case 4)
  - The system should allow owners to record a message once the pizza is ready that can then be shared to the customer when the pizza is delivered
- In-person Orders (Use-Case 2)
  - PWAP must give owners a way to use the owner side app for orders that are received in-person and put into the system, but flagged as in-person orders
- Update Menu and other Administrative tasks (Use-Case 7)
  - The system must allow owners will be able to change which ingredients can be ordered by the customers depending on the availability or season. This will be done through a web browser and not through the app.

- Store Food Voucher Information (Use-Case 2,3,4)
    - The system should keep track of the food voucher information so that the app can quickly (under 15 seconds) recognize when a food voucher is used and make the process as smooth and trouble-free as possible

4.2.3 Courier side

- Delivery information (Use-Case 6)
    - PWAP must display the current destination and order, and the return destination in case the trucks move when an order is being delivered
- Navigation (Use-Case 6)
    - PWAP must integrate with third-party software to provide navigation on the same screen as the delivery information so that the driver can stay on schedule

4.3 Data Requirements

4.3.1 Customer Information

- PWAP must store the customers information (first and last name, delivery address, phone number, email, password and credit card information if customer so chooses)
- PWAP should record when a customer places an order, and other analytic information (time spend ordering, total amount of orders, what food is ordered, etc..)

4.3.2 Order Information

- The system must hold all information about current orders and save information relevant to the co-op (total amount of pizzas made daily, weekly, monthly and ingredients) used so owners will know what they need to order
- The system can erase information that connects pizzas to customers and only store information essential to the co-op

4.3.3 Deliver Information

- PWAP must hold addresses that the pizza is being delivered too and the time it is to be delivered
- PWAP should also hold on the delivery app the message recorded by owners to be shared with customers

4.4 Nonfunctional requirements

4.4.1 Operation

- PWAP must be able to operate across many different devices including mobile phones and desktop computers
- PWAP will need to run on both iOS/Android for the couriers when they are out on deliveries
- PWAP must be able to run on many different devices at once and have access to the same records that hold information essential to transactions

- PWAP will allow analytics and information sensitive to the co-op to be accessed only by the food truck owners who hold the Administrator username and password

4.4.2 Security and Control

- Administrator accounts will be held by some food truck owners and will be able to access file records and be able to change information such as menu options
- Specified individuals chosen from Disciples' Pizza will be given the Administrator privileges
- All customer payment information will be encrypted and will not be able to be accessed by any food truck workers/owners or NSD

4.4.3 Cultural, Political, and Legal

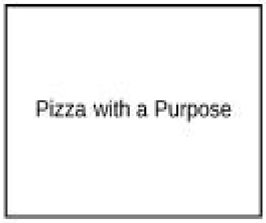
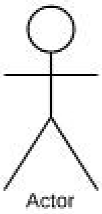
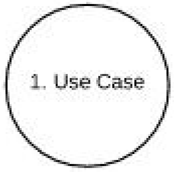
- PWAP will be developed in English and the food owner/courier side will also be in English unless Disciples' Pizza needs another language to be developed in the future
- PWAP will provided the customer side app in multiple languages to accommodate for the main languages spoken in operational areas

5.0 Requirements Model

5.1 Introduction

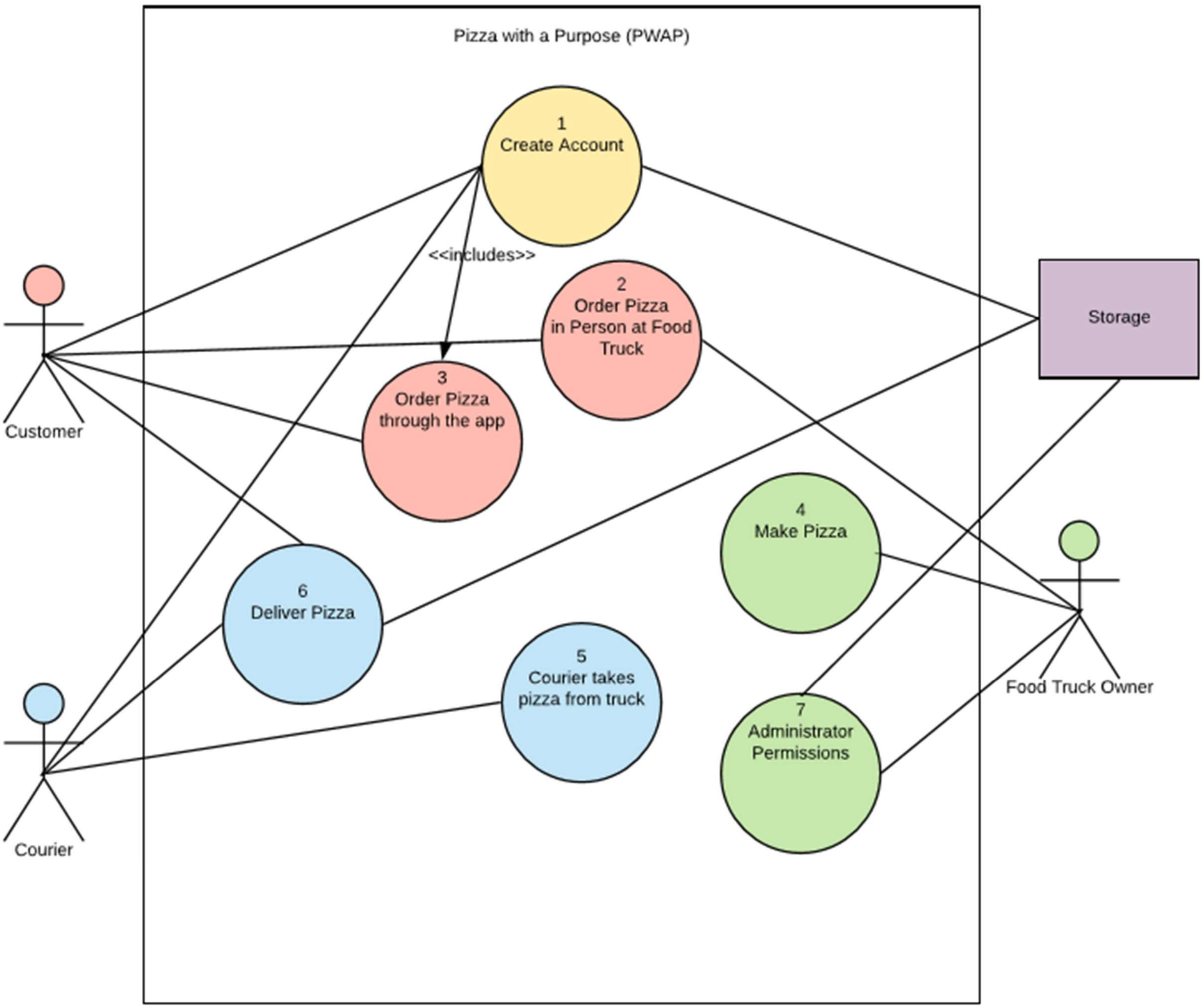
Section 5 will cover various use-cases for the PWAP system. The use-cases themselves will be in section 5.2, and then will be described in section 5.3. The use-cases are designed to show how the system will work and be centered around the function requirements that are stated in section 4.2.

The Use-Case Diagram's contain the following components.

Name	Symbol	Description
PWAP		Outlines the system and shows what is contained within the system and what is outside the system.
Actors		An Actor is any person or thing that will interact with system. Actors are outside of the box because they are not in the system but can provide inputs to the system.
Use-Case		Use-cases are the different tasks that PWAP will need to be able to do. Each use-case will have a description in 5.2.
Association	———	An association connects an actor and a use case. This represents an interaction between the two.
Include	<< includes>> →	Includes highlights the fact that one use-case needs another to work. Includes are drawn from the base case to the required case for the base.

<<

5.2 Use-Case Diagram



5.3 Use-Case Descriptions

Use-Case name: Create Account	**ID**: 1	**Importance**: High
Primary actor: Customer, Owner, Courier	**Use-Case type**: Detail, Essential	

Stakeholders and interests: Customer, Owner, Courier, Disciples' Pizza as a whole since if more people create accounts then more pizza will be sold.

Brief description: The user will create an account depending on what they need to do. Customers will create an account including information such as name, address, number, email. This information will be stored in the server that can be accessed so returning customers can log in to the system for an easier experience. Owners will create an account that they will use to take orders at the truck and use it to make changes to the menu. Couriers will create accounts that will show the destinations that they need to go to.

Trigger: Customer chooses to create an account when ordering. Owner/Courier creates an account when the system is ready to be fully used by Disciples' Pizza.

Type: [✓ External] Temporal

Relationships:

Association: Ordering Pizza through the app
Include: None
Extend: None
Generalization: None

Normal flow of events:

1. System opens a login screen which will have an option to login or create an account
2. User will choose to create an account
3. User inputs information need for account creation
4. Information is stored securely on a server to be accessed when user returns
5. User does whatever task they logged in to do

Subflows:

3a. User will input First/Last Name
3b. User will input preferred phone number and/or email
3c. User will input password for account
3d. User will input default address for delivery

Alternate / exceptional flows:

2a. Returning users will simply login to their account instead of creating an account
2b. Customer can use a guest account if they don't want to make a real account

Use-Case name: Order Pizza in Person at Food Truck

ID: 2

Importance: High

Primary actor: Customer, Food truck Owner

Use-Case type: Detailed, Essential

Stakeholders and interests: Customers, Food truck Owner, Disciples Pizza more people ordering will create more business.

Brief description: Some customers will choose to come to food trucks and order in person instead of ordering through the app. These customers will speak directly to the owner who will put the order in the system where it will be displayed along with the other orders but marked so signify it is separate from the delivery orders.

Trigger: Customer comes to food truck to order

Type: [✓ External] Temporal

Relationships:

Association: Customer, Food truck owner
Include: None
Extend: None
Generalization: None

Normal flow of events:

1. Customer comes to a food truck to order pizza
2. Owner puts an order into the system
3. The order is displayed with the delivery orders but marked as in-person

Subflows:

1a. Customer pays with credit card or cash

1b. Customer uses food voucher which is then updated with information in server to keep the list of available food vouchers accurate

Alternate / exceptional flows: None

Use-Case name: Order Pizza through the app

ID: 3

Importance: Must

Primary actor: Customer

Use-Case type: Detailed, Essential

Stakeholders and interests: Customers, Food truck Owner, Disciples Pizza more people ordering will create more business.

Brief description: Users will order through the app either by logging in their account or using a guest account. They will select the food they want to order and that will be sent to the closest food truck to be made and then delivered.

Trigger: Customer logs into the app

Type: [✓ External] Temporal

Relationships:

Association: Customer
Include: 1. Create Account
Extend: None
Generalization: None

Normal flow of events:

1. Customer signs into app
2. Customer chooses pizza and/or any sides, drinks they wish to purchase
3. Owner received order and begins to make the pizza

Subflows:

2a. Customer chooses type of dough and size of dough
2b. Customer chooses sauce and other toppings
2c. Customer inputs any special request

Alternate / exceptional flows:

1b. Customer will be asked to create an account if they are not a returning user already

Use-Case name: Make Pizza	**ID**: 4	**Importance**: High

Primary actor: Food truck worker	**Use-Case type**: Detailed, Essential

Stakeholders and interests: Customer who wants the food to be made correctly, Courier who wants the food to be made on time and correct to have customer satisfied, Owner who makes the pizza, Disciples Pizza who wants to maintain a high quality and brand reputation

Brief description: Food truck owners will receive a notification about the food being orders and will make the food in the correct order of priority. The owner will update the status of the order as it is being made. The courier will come and take the food when it is ready.

Trigger: Food truck owner receives notification of an order being placed

Type: [✓ External] Temporal

Relationships:

Association: Food truck owner
Include: None
Extend: None
Generalization: None

Normal flow of events:

1. Customer places an order through an app
2. The system sends notification to the nearest food truck
3. Owner makes the food, updating the status throughout the process
4. Owner records message to be shared with customer upon delivery

Subflows: None

Alternate / exceptional flows: None

Use-Case name: Courier takes pizza from truck	**ID**: 5	**Importance**: High
Primary actor: Courier	**Use-Case type**: Detailed, Essential	
Stakeholders and interests: Courier who will deliver the pizza, Customer will receive the pizza hopefully on time and correct, Disciples pizza who needs to uphold the reputation of the brand.		
Brief description: Courier will take the pizza from the correct food truck once the owner updates the status of the order to being completed.		
Trigger: Status of order changed to completed **Type**: [✓ External] Temporal		
Relationships: **Association**: Courier **Include**: None **Extend**: None **Generalization**: None		
Normal flow of events: 1. Status of order change to completed by food truck owner 2. System alters courier of the order 3. Courier takes the order and changes the status to being delivered		
Subflows: None		
Alternate / exceptional flows: None		

Use-Case name: Deliver Pizza | **ID**: 6 | **Importance**: High

Primary actor: Courier | **Use-Case type**: Detailed, Essential

Stakeholders and interests: Courier who is responsible for delivering the pizza onetime and to the correct destination. Customer who will receive the order and Disciples' Pizza who needs to uphold reputation.

Brief description: Courier will use directions that are a part of the system to deliver the pizza from the food truck to the correct destination.

Trigger: Courier

Type: [✓ External] Temporal

Relationships:

- **Association**: Courier, Customer, Storage
- **Include**: None
- **Extend**: None
- **Generalization**: None

Normal flow of events:

1. Delivery information is sent from a server to the courier app
2. Courier app first display directions to desired destination to pick up pizza
3. Courier takes pizza from truck, owner updates customer on status
4. Courier app display directions to customer
5. Courier gives pizza to customer
6. Message recorded by food truck owner is shared to customer

Subflows: None

Alternate / exceptional flows: None

Use-Case name: Administrator Permissions | **ID**: 7 | **Importance**: High

Primary actor: Food truck owner | **Use-Case type**: Detailed, Essential

Stakeholders and interests: Food truck owner who needs to do certain tasks to keep co-op running efficiently. Disciples' Pizza who will continue to run an effective co-op if owners are able to do whatever they need to within the system.

Brief description: Food truck owners will be able to change menu items and update available ingredients depending on the season or stock. Can also do other administrative tasks that can be decided during development.

Trigger: Food truck owner logins in with administrative credentials.

Type: [✓ External] Temporal

Relationships:

Association: Food truck owner
Include: None
Extend: None
Generalization: None

Normal flow of events:

1. Food truck owner logs into administrator account
2. Owner does whatever task they need to
3. Server Storage updates information in the system automatically

Subflows: None

Alternate / exceptional flows: None

6.0 System Evolution

6.1 System Development

PWAP Version 1

Disciples’ Pizza wished in the system request to have a version of the system with core sales functionalities to be finished soon. This first version of the system will have only what is needed to facilitate sales while the system if being fully developed. This initial version will most likely only include the food truck owner app and the web-based management system that will be used by the owners.

PWAP Version 1.2

The next step will be to add the delivery app that will be used by couriers. This will allow Disciples’ Pizza to make sure the delivery app works as intended and any essential features can be added before the customer app is added to the system.

PWAP Version 1.3

The third version will include the customer app and will be the first version of the whole system. At this point, the customer app should be able to provide all core functionalities (ordering, tracking order progress, etc.), but may still need usability improvements down the road.

PWAP Version 1.4

A final version will include all requirements that have been agreed upon as well as fixing any bugs that get discovered in the first three versions. A bug would be anything that prevents the system from working as intended and is harmful to the success of the system. At this point, the system will provide Disciples’ Pizza with a fully usable system that meets the needs of their co-op.

6.2 System Maintenance

Any further upgrades can be negotiated by Disciples’ Pizza and Nile Software Development. NSD will be able to answer any questions and will continue to provide routine updates to the system and improvements if Disciples’ Pizza identifies something they would like to be added.

Security improvements or updates may be necessary and will be recommended by NSD when we see fit. Updates to ensure compatibility may also be recommended. Disciples’ Pizza should notify NSD if they purchase new hardware to ensure compatibility.

7.0 Conclusions and Recommendations

7.1 Conclusion

The Pizza with A Purpose system will allow Disciples' Pizza to grow their co-op and meet the goal of providing quality pizza with a delivery system that meets their unique situation. The potential risks associated with the development of the system do not provide any strong threats, and thus NSD sees the project as feasible. Still, NSD would like to confirm a time and budget goal for the system before the system being developed to prevent unforeseen risks from becoming an issue. NSD is confident that it possesses the necessary developers, resources, and knowledge to complete the project to meet Disciples' Pizza's goals allowing them to reach more people and generate more business.

7.2 Recommendations

To guarantee that PWAP works as intended NSD has a few recommendations.

- Continue to communicate with NSD as the system is being developed so that any changes can be accounted for quickly and effectively without delaying the system
- Continue to communicate with NSD after the system is completed to ensure that maintenance is carried out as need and any bugs in the system can be dealt with
- Allow NSD to train a few (less than 5) people on how the system works and then have those people train anyone within Disciples' Pizza on how to use the system
- Talk to customers about their experiences with the customer side app and to couriers to see how the system can be improved in the future
- Keep computers and any devices owned by the co-op or used by couriers updated and fully functional to ensure that all parts of the system work.
- Any computer that uses the system (specifically in the food trucks) should be scheduled to turn off outside of business hours to save cost and not become unresponsive

Glossary

PWAP: The Pizza with a Purpose system. This includes the different apps that will be used by the customers and food truck owners.

NSD: Nile Software Development

Server: Storage that holds information for the co-op and can be accessed by the many different sides of the app via the internet.

Bibliography

Dennis, Alan, et al. *Systems Analysis & Design: an Object-Oriented Approach with UML*. Wiley, 2015.

Pfeiffer, William S. *Pocket Guide to Technical Communication*. Prentice-Hall, 2011.