

nonPOS Video Presentation Transcript

Introduction (Speaker: Joshua Logue)

Hello, my name is Joshua Logue and this is nonPOS, a modern point of sale system designed with small businesses in mind and ease of access as the primary goal.

When we began this project, we wanted to differentiate ourselves from a typical POS system, notoriously difficult to read or use.

Instead, we set out to design a system in which any potential user can easily find the feature or function they are looking for and immediately know how to use it once they do.

As is typical of these types of systems, nonPOS aims to create and manage a large quantity of incoming orders (whether these orders consist of food, merchandise, or services) as well as processing payments for these orders.

Finally, all of our features can be directly accessed via web browser so that virtually any device with an internet connection should be able to support and utilize nonPOS.

Target Market (Speaker: Gavin Meren)

Hello, my name is Gavin Meren, and I am going to be discussing the intended target market for nonPOS.

We designed nonPOS specifically for use in the quick-service industry, or fast-food restaurants as they are more commonly known.

The industry caters to the modern consumer lifestyle, requiring a very fast customer turnover rate and resulting in a 70-30 split between drive-thru and dine-in sales, respectively.

Alongside these factors, the industry also faces a high employee turnover rate.

We intend nonPOS to be an easy to use and simple to learn system to accommodate these factors.

Of course, in an industry where the biggest companies can easily afford to have systems designed for them, we will have to cater to smaller, non-franchise restaurants.

Quick-service restaurants run on a thin profit margin, and the price of equipment required for some POS systems might be too much to bear for some smaller restaurants.

So, to keep operational costs down and make the service cheaper, nonPOS will be run in a browser window on devices such as PCs and smart tablets.

If we only charge a \$90 monthly premium, which is competitive with other similar POS systems on the market, we will only need to capture 10% of the market, or approximately 4,000 restaurants, to generate \$360,000 in revenue a month.

I will now pass it off to Nagi to discuss our competitors.

Work Plan (Speaker: Maribel Mireles)

Hello, my name is Maribel Mireles and this is nonPOS work plan.

For stride A we set one week for the setup of nonPOS database.

For stride B two weeks set for the group to read up on languages, HTML, CSS, jQuery, and Ajax.

For stride C two weeks to read up on PHP, PDO, and database management.

For stride D we set five weeks.

In this phase HTML, CSS, and JavaScript are to create a landing page, login page, create account page, menu page, and other required pages.

Two weeks for stride E that consist of the development of the application database.

This phase allows us to create the types of data we will be storing in our database such as users, pins, menu items, sales data, prices, and orders.

Three weeks for stride F that consist of building logic and communication with the database. Since some functions require manager credentials and others only employee credentials.

Seven weeks for stride G. in this phase is where all functions are developed such as displaying menu, saving data, fetching data, and creating payments.

Three weeks for stride H.

The last phase consists of integrating the square API into our application.

Competitors (Speaker: Nagi Obeid)

Hello, my name is Nagi Obeid and I'm going to be discussing our competitors.

We wanted to take on this project because we noticed there was a sector in the quick service restaurant industry that wasn't being addressed.

Other sectors such as general retail and casual dine in restaurants had plenty of point-of-sale options.

In fact, I'd say that the market was pretty saturated with systems that catered to those types of business models.

The fast-food restaurant industry, however, requires refined systems that enables employees to navigate large menus intuitively, there are a few POS systems available and are used by small franchises and single location establishments.

Square is one of the most common systems.

However, Square is too simple as it does not have the flexibility required to process complicated orders, nor does it have the capability to handle drive thru orders.

All orders can only come in for one order stream.

It is impossible to separate orders coming in from different devices.

The next two options are Clover and Toast they are a bit more flexible and have more features than Square.

However, they can get messy to navigate when large menus are added and often times, you'll want to add custom modifiers to each item.

Modifiers are the items that can be removed or added to an item.

Clover and toast don't have the ability to accomplish the task of applying selected modifiers, only to items as needed.

Sort of like when you're at a restaurant and ordering three hamburgers, and you want each of those hamburgers to be prepared differently.

The last POS system is NCR.

NCR is probably the most popular in the industry, and honestly is the only POS system. That provides most of the essential functionality that is required.

It has drive through capability which all other systems lacked and plenty of the features we want to incorporate into our own project.

All these POS systems require expensive hardware that can easily go beyond the thousands depending on the number of units needed.

Also, monthly integration fees are a huge burden that these systems require their users to have.

In terms of web based QSR point of sale systems. There aren't any competitor competitors.

Architecture (Speaker: Kevin Mitra)

Hello, my name is Kevin Mitra and I will be covering the Architecture part of our project.

We work through the Odin server to make and to test our project to have everything working.

For the databases, we use MariaDB to have stored data such as username and password for users to gain access and also, to store items on the list.

For the server-side, we use PHP for security purposes when the user can create or login to their account, and then MySQL to work around stored data in the database as our back end.

Although, we test the application through Linux Apache and Lamp Stack web server.

For the client-side, we use JavaScript, JQuery, HTML, Bootstrap, CSS, and AJAX as our front-end.

We use the Quagga JavaScript for scanning mobile devices to detect the bar-code.

And lastly, we use the square API for making transactions through processing debit card or credit card for purchases.

Conclusion (Speaker: Kevin Mitra)

And this concludes our project overview which will provide much service to small businesses.

And now we will move on to our project demonstration.

Demonstration: nonPOS Web Page (Speaker: Kevin Mitra)

Hello, this is Kevin again, and welcome to the non Point Of Sale, where users can make small business lives easier.

Users may have the application on their computer, tablet, or mobile device to work with the orders.

So, here's the overview of our web page which includes a little bit of information, and the contact info.

And here are the links and buttons to where users may login, or to create an account to have access.

And I will turn it over to Joshua to demonstrate those.

Demonstration (Speaker: Joshua Logue)

Hello, Joshua again, and this time I'm going to look at business account creation and resetting a password.

These two features are fairly similar and interact with the back end in the same way.

First, I will go into account creation, implemented by Nagi Obeid.

From the website's landing page, we can select this create account button, which will then take us to the creation page.

Here, we can enter in the information for the account we wish to create.

Remember, this account is specifically for a new business, that is, the owner who will be controlling it.

Because we will be referencing it later, I will show the password as I make it here.

After entering the necessary information, the password is hashed, and the form submits this information via query to our backend and a new account has been created!

We are taken to this page next, which allows us to create an initial manager for our business.

We are going to ignore that for now, however, and go ahead and log out.

Now that we have taken a look at creating an account, what happens if we forget our password and lose access to our account?

Well, we can take a look at resetting a password, a feature implemented by myself.

From the login page, we simply click this second button here to begin resetting our password.

From this page, we are prompted to enter the email address we registered with, so let's go ahead and do that and click submit.

In the backend, a token has been generated and assigned to our account, creating a link personal to us.

Now, we are informed that this reset link has been emailed to us, so let's close this and check our email.

Here is the email sent from our reset attempt so when we click the link, we are taken to the actual reset page.

So, when we enter our new password here and submit, that password is once again hashed before another query is made, this time updating our backend with the new password and giving us this success message.

Now we should be able to log into our account with our new password.

And we do.

The largest issue I've had with this feature is implementing some form of time control on the reset token, so that an ignored token will eventually time out and its validity will expire.

I'm almost there and I'm sure I will get it.

Next up is Nagi and Kevin.

Demonstration (Speaker: Gavin Meren)

Hello, this is Gavin again and I am now going to demonstrate the “Employees” page, which I was responsible for implementing.

The primary purpose of this page is so that employees can easily clock-in and clock-out, allowing the system to track their shifts.

An employee can also view their personal employee info.

On top of this, the page allows for managers to view the current info about all employees working at the same business, as well as access pages for editing, deleting, and creating employee info.

When you click on the section labeled “Employees” on the business “Home” page, you will be brought to this page.

If a user is just a normal employee, the page will look like this.

The view of the page is determined by whether an employee is set up to have manager permissions in their info.

Here we can see the “Clock-In”, “Clock-Out”, and “Personal Employee Info” buttons.

We are starting out on this page clocked-in for demonstration purposes.

By clicking on the “Clock-In” and “Clock-Out” buttons, we receive feedback about the start and end times of the shift.

The times shown are in UTC.

These features are handled with attributes in the employee records of the database.

One for signifying whether an employee is clocked-in and another for holding the last clock-in time.

Now if I click on “Personal Employee Info”, you can see the info of a single employee appear on the side here.

This is a simple call to the database to retrieve the appropriate information. Let me leave this page and change over to an employee with manager permissions.

Now that we are back with manager permissions, we can see three more buttons.

“Create New Employee” takes us to the page to create more employee records, but this has already been covered in the account creation portion of the video.

“View Employees” gives us a list of all employee info, including our own.

If we click on “Manage Employee Data” we are taken to another page where we see a set of fields and two buttons.

Here we can update the data of an employee and we can also delete an employee’s data as well.

A manager PIN is needed to perform these operations as well as the ID of the employee the operation is being performed on.

In case a manager leaves themselves logged-in, nobody can mess the data.

These are also handled by simple calls to the database, but the update function requires logic to properly build the query based off what fields are filled when the button is clicked.

Allow me to quickly perform an update and delete operation.

After performing the operations, we return to the “Employees” page.

If we click on “View Employees”, we can see that “Jane” is no longer listed and “John” now has the name “Jack.”

If I had the time, I would have liked to clean up the front end of these pages.

The output could be more visible and organized better, and the containers around the fields and buttons needs to be formatted properly.

Features that I had not thought of when originally planning for this page and was not able to implement include one that translates times to the time zone of the businesses’ choosing.

I also would have liked to have an extra table to store employees that are removed rather than outright deleting them, since it would be in the interest of proper record keeping for an employer to hold onto that data.

I also planned to have a page where a manager could submit an employee ID for an employee in their business and get a list of all their shifts.

The main challenge I faced was working with the time data types in PHP.

To store the different times in the database properly and keep an accurate count of hours worked I had to figure which time objects had the functions I wanted and what kind of conversions needed to be made.

I needed to fine tune which objects I used and how I formatted the data I was taking from them so they would be compatible with the database but still contain the info I wanted.

I will now pass it off to Mary to discuss the “Settings Page”.

Demonstration: POS Register (Speaker: Nagi Obeid)

Hello, this is Nagi again.

This time I'm going to be demonstrating the POS Register section so once you click the Post Register right here.

It's going to take you to the main screen where customers can place orders and employees can take customer orders.

So, on the far left we're going to have a little section that's going to compile all the menus of a specific business.

In this business, they're going to have combos, burgers, sandwiches, dinners, and several other sections.

Then on the top right here, there's going to be a bunch of numbers that represent the multipliers that can be applied to each item, so a customer can want either one cheeseburger, can want six cheeseburgers and can want 10 cheeseburgers, and that's how they are applied.

Then on the far right, you're going to have four buttons that represent where the order is coming from: either it's a dine in order, a take-out order, a call-in order, or drive through over.

Then on the bottom you're going to have a save option, a charge order option or view open orders option.

So, let's go ahead and place an order.

Let's assume I'm a customer and I want two number ones.

OK, so I do that now.

My friend wants one of the number ones to have no onions, typically in other POS registers this is impossible.

You would have to completely separate each order so you would have to do 1 number one with no onions OK and then he would have to place his other number one with onions.

But with our system it's a bit easier since you wanted two, you could just do 2 number ones.

One, no onions, and then one regular and the drinks.

Let's get two doctor Peppers and then let's save that over and then on the far right OK.

A list of all the orders are going to be shown, and then it's going to add the total.

And this is the customer total to be charged.

Now, from here you could select where the order is going to.

So, let's say this is a dine-in order.

Now you could either save it, which is going to send it to the kitchen, or you could charge it, which is going to charge, then send to the kitchen so it'll go ahead and save.

Now you're going to be prompted to enter name.

Let's say my name is Jack.

Save the order.

If you were to open orders, let's remember that we saved it as a dine-in, and if we go all the way down there it is.

Jack. And the last five or six numbers are going to represent the ID for the order.

OK, now next I'm going to facilitate a squares API charging system that we implemented on a mobile device.

Demonstration : Process Transaction (Speaker: Nagi Obeid)

Hello, this is Nagi.

This time I'm going to be demonstrating how to process a sale using Square's payment API.

So first, let's go ahead and add an item to the queue.

Combo development.

Now this item is specifically for development purposes.

Let's go ahead and save it as takeout.

Now, save it under my name.

Charge, and now we're going to be asked to open the square application.

The price is populated into the system and all you have to do is swipe your credit or debit card.

And as you can see, the transaction looks successful.

All you have to do now is to press the blue button and you will be taken back to the main nonPos Register section.

Demonstration : Sales (Speaker: Nagi Obeid)

Hey everyone, now I'm going to be showing you how the sales functionality works.

So, as you can see the sales section is split into four divisions.

There is by the day by the week by the month by the year, and as you can see, today's sales totaled to about \$17,000.

Let's go ahead and change that.

Let's go ahead and process development sale.

Which is just a fixed number for development purposes.

It's going to be a \$5000 item.

Let's do dine-in.

And let's save that.

Now if we head back to the sales.

We can see that sales have increased to 22,000 and this is a split.

We know that it's 46% development and 53% combos.

OK, now we have the gross sales returns, net sales, taxes, and the total to be collected.

You can also view by the week.

So, it seems that yesterday and today were the only days of the week of this week that sales were collected.

A month, as you can see, there were several days where transactions were processed.

And this is a more even distribution of the split and note that the percentage is not of the sales, but of the items sold.

At gross sales, returns net sales, taxes and total are all modified, and calculated dynamically after each cell.

Demonstration : Menus (Speaker: Nagi Obeid)

Hello, now I'm going to be showing you the menu section.

Now there's two specific categories that I worked on.

Add category and Add item.

Add category refers to the menu category that's going to be added to the business' menu.

Let's say that I want to add a new category called soft drinks in the menu.

We add that category.

And to add an item you want to save that under our newly added soft drinks.

And let's grab that and give that a price of \$2.99.

Then we could add a modifier.

“EZ Ice” at 0 price.

“Extra Ice” at 0 price.

“No Ice” at 0 price.

Then Add item.

Now if we head back to the register, we will see our newly added soft drink section and in that we will see our item.

And in those we will see the modifiers.

Demonstration : Menus Remove Item (Speaker: Kevin Mitra)

Hello, this is Kevin again, and I will demonstrate in the menu page on removing and editing items from the menu list after an item was added.

So, we will go on to the Remove Item section.

And here we have our the list of items displayed inside the menu.

And right now, we have two hazelnut coffees displayed, and let's go ahead and delete one of them.

How we gonna do it?

You can either click on one of them, or you can search up hazelnut coffee in the search bar.

So, we'll go ahead and type in Hazelnut Coffee.

Once you have done it, it will be deleted from the list.

Demonstration : Menus Edit Item (Speaker: Kevin Mitra)

So, in our edit item section we can edit items on the menu you can either do one of the two things.

You can either click on the item which is on the menu list, or you can type it up on the search bar.

So, let's go ahead and type in Super Burger.

And once you have done it, you'll make changes to the item.

And that's about it for the demonstration of our menu page. The next will be the employees page which I will turn it over to Gavin to demonstrate.

Demonstration : Pin, manager Settings (Speaker: Maribel Mireles)

Hello, Maribel again. This time I will discuss the pin, and settings futures.

First, I have to log in into nonPOS system.

For pin functionality we have two different pins. One for the manager and other for the employee. When I type the pin for the employee. It takes me to this menu only for the employees.

If I go back and type the manager pin. This take to this menu with all these futures including the setting that I will discuss next.

These settings are for the manager to change his Email, Username, Password, and Sales Tax.

I'm going to work on an example from my local database that I have here in the bottom.

First, I am going to change the email, next the username, then the password.

I already change the email, username, and password. Now, I'm going to refresh my database to see the new changes.

Here is the new username, new email, and the new password.

As you could see there was no need to ask for any personal info. Variables are carried out from the log in. I still have work to do on the settings, but I am almost there.

Conclusion: Conclusion (Speaker: Nagi Obeid)

Hello, in this final slide I will be discussing what we learned and what we plan for the future.

One take away from the project is to plan everything and expect that something will go wrong.

There were often times when a feature was in the process of being developed and suddenly the data required for the future was either missing needed to be restructured or entirely unsupported.

There were even times when the approach simply wasn't the best way to go about programming The feature.

At that point we would take a few steps back and re-evaluate the method before moving forward.

For the future, one of the first things we want to do is expand item modification to allow nesting multiple sub items.

This will allow employees and managers to easily customize modifiers by adding modifiers to those modifiers.

This feature will really be great for restaurants with very complicated menus.

Where customers are often requesting to leave off or add on specific condiments and quantities to their food items.

Next, we would like to add barcode scanning functionality.

This feature will allow Nonpos to easily dig into other business models with similar needs such as gas stations with built in buffets or sandwich shops and small markets.

Next we would like to add a live kitchen view, a live kitchen view will allow cooks to see which orders came in first and to prioritize their time by gearing urgency towards orders that have been in the kitchen longer.

time management will be possible because order timers will allow the cooks and kitchen staff to see how long each order has been in the kitchen.