Pizza Empire

# Overview

Pizza Empire is a casual building game targeting mobile devices. It’s similar to Hay Day but with the setting of a chain of pizza restaurants instead of a farm. The player starts with one small pizza restaurant and expands to multiple locations and delivery services to cover as much of the country. Players can also connect with other players.

# Business Model

Ad based and freemium

Building game where everything takes time and costs in game currency. Players can spend real money in order to reduce waiting times or to get instant in game currency.

# Gameplay

The game will play like other building games. Players will perform actions and then wait for the results.

There will be three level of resources.

* Raw ingredients
* Processed ingredients
* Final products

Resources at each level will require resource(s) from the level below in order to be produced.

### Pizza Sign Design

This is the first screen the player will see when they are starting a new account.

This screen will show the player’s pizza sign. On this screen the player can change the name of their pizza empire and can customize their sign.

Names will need to be filtered for bad words!

The player can choose the color scheme and logo for their sign.

* Text color
* Text position
* Background color
* Background material / texture
  + Wood
  + Metallic
  + Rock
  + Etc.
* Border colors
  + Two colors for alternating pattern
* Border shape
  + From a choice of approximately 5 shapes
  + Solid Rectangle
  + Stars
  + Circles
  + Etc.
* Main logo image
  + From a choice of 10-20 images
* Logo size
* Logo position

The design on the pizza boxes can also use the personalized sign.

### City Screen

This is the first screen the player will see when they are returning to the game.

The city screen will show the restaurant and surrounding neighborhood. The screen will show the street signs and buildings. When you click on a building a pop up will show that building’s address.

This screen is used for delivery as well as seeing your restaurants area of service in the neighborhood.

Over time the player can upgrade his delivery services so that he can deliver to a larger area. The city screen will zoom out and show a wider area as your delivery area increases.

The player can tap on their restaurant to switch to the restaurant screen.

### Restaurant Screen

This screen will show a 2.5D Isometric (like Hay Day) view of your restaurant.

The front of the restaurant will show your pizza sign.

The screen will contain the following items:

* Phone
* Loading dock for receiving raw ingredients
* Cash register
* Pizza Oven
* Stove for cooking sauce / ground beef / chicken
* Vegetable slicer
* Cheese grater
* Meat slicer
* Dough mixer
* Pizza assembly area
* Raw ingredient storage
* Customer Seating area
* Computer for tracking customer orders
* Motorcycle / delivery station

## In game currency

There will be two types of in game currency

* Cash (bills)
  + Required for almost all in game activities
    - Buying ingredients.
    - Upgrading equipment.
    - Hiring employees
    - Etc.
  + Will be earned by selling product.
  + Can be bought using real world currency.
* Pizza coupons (like diamonds or gems in other games)
  + Can be used to speed up all activities.
  + Will be required for some upgrades / activities.
  + Can be earned by viewing certain advertising.
  + Can be bought using real world currency.

## Starting a new game

The player can login with Facebook or create an account specifically for Pizza Empire.

## Head office

This is where the player orders raw ingredients.

They can also view stats / trophies at the head office.

## Leveling / Experience

The player starts with one small restaurant. The initial menu is limited to a small amount of pizzas and soft drinks. The restaurant will have limited space and the delivery crew will be only one person.

Over time the player gains experience which increase their level. Increasing level allows players to

* Unlock new pizza recipes
* Expand their restaurant
* Open new restaurants

## Raw Ingredients

Raw ingredients are the basis for all of the products that the pizza restaurants produce. They are ordered from suppliers, cost in game currency and take time to be delivered.

As the business expands, larger quantities of ingredients can be ordered for cheaper cost per unit.

The raw ingredients are stored somewhere which can be different at different stages of game.

### Flours

#### White Flour

#### Whole Wheat Flour

### Vegetables / Fruits

#### Tomatoes

#### Pineapple

#### Hot peppers

#### Mushrooms

#### Green peppers

#### Onions

#### Corn

#### Lemons

#### Olives

#### Spinach

### Herbs and Spices

#### Basil

#### Oregano

#### Pepper

#### Olive oil

#### Salt

#### Sugar

#### Yeast

### Meat

#### Pepperoni

#### Italian Sausage

#### Ham

#### Salami

#### Chicken

#### Beef

#### Anchovies

### Dairy

#### Cream

#### Mozzarella Cheese

#### Cheddar Cheese

#### Parmesan Cheese

#### Romano Cheese

#### Feta Cheese

### Syrups / powders

#### Cola syrup

#### Citrus syrup

#### Root beer syrup

#### Iced Tea powder

#### Fancy syrup

### Miscellaneous

#### Pizza boxes

## Prepared Ingredients

Prepared ingredients are made from raw ingredients and take time to produce. Processed ingredients are used to assemble pizzas.

### Dough

#### White pizza dough

#### Whole wheat pizza dough

#### Thin crust pizza dough

#### Deep dish pizza dough

#### Calzone pizza dough

### Sauces

#### Basic pizza sauce

#### Spicy pizza sauce

#### Salsa pizza sauce

#### Alfredo pizza sauce

### Meats

#### Sliced Pepperoni

#### Sliced Italian Sausage

#### Sliced Ham

#### Sliced Salami

#### Sliced Chicken

#### Ground Beef

#### Anchovies

### Vegetables / Fruits

#### Sliced Tomatoes

#### Sliced Pineapple

#### Sliced Hot peppers

#### Sliced Mushrooms

#### Sliced Green peppers

#### Sliced Onions

#### Corn topping

#### Sliced olives

### Dairy

#### Grated Mozzarella Cheese

#### Grated Cheddar Cheese

#### Grated Parmesan Cheese

#### Grated Romano Cheese

#### Feta cheese topping

### Drinks

#### Cola

#### Citrus soda

#### Root beer

#### Iced tea

#### Lemonade

#### Fancy soda

## Pizza Recipes

Initially the player can make a few different pizza recipes. As the player gains levels new pizza recipes are unlocked and customers begin to ask for these new pizzas. This will require the player to buy new raw ingredients, produce new prepared ingredients.

### Level 1

#### Cheese pizza

#### Pepperoni pizza

#### Hawaiin pizza

### Level 2

#### Deluxe pizza

## Equipment

The equipment at the restaurant can be upgraded to produce more quantity and more rapidly.

### Phone

### Loading dock for receiving raw ingredients

### Cash register

### Pizza Oven

### Stove for cooking sauce / ground beef / chicken

### Vegetable slicer

### Cheese grater

### Meat slicer

### Dough mixer

### Pizza assembly area

### Raw ingredient storage

### Customer Seating area

### Computer for tracking customer orders

### Motorcycle / delivery station

## Employees

Employees are required in order to produce pizzas, server customers and do deliveries.

## Customers

Customers can arrive at the restaurant for take-out or dine in. They can also phone for pick up or delivery.

### Special customers

Sometimes special customers arrive as a reward for the player. When you serve a special customer you get some reward. Once you have accumulated a certain number of these rewards you can spend them on special prizes.

#### Musician

#### Movie Star

#### Mayor

#### Millionaire

#### President

## Delivery

The player starts with one motorcycle for delivery. More delivery employees can be hired in order to deliver more pizzas at once. The motorcycles can be upgraded to decrease delivery time.

To perform a delivery the player will open the map and click where the motorcycle has to go.

## Restaurant

The restaurant can be upgraded to seat more customers and look nicer. Fancier decorations are unlocked as the player gains levels. Some may also cost cash / coupons.

## Trophies

Players can accumulate trophies.

### Raw Ingredients

### Production

### Overall Sales

### Takeout

### Dine in

### Deliveries

### Special customers

### In game currency

## City Map / Neighborhood / Delivery area

Not sure how this will work but something about your area of influence / delivery radius.

## Marketing

There will be different types of marketing. This will possibly use the city map.

# Collaboration

The game will connect players to their friends via Facebook.

There will be ways for players to trade all level of resources.

There will also be a way to chat with a limited number of other players (groups).

Chat will be filtered to remove profanity / publicity / etc.

# Notifications

Players will receive notifications when products are produced.

Players will receive notifications when they haven’t done things for a long time.

# Leaderboard

There will be leaderboards for the biggest pizza empires (total pizzas served, highest revenue) and for the highest monetary spenders.

The leaderboards will be available in game and on the website.

Will most likely have to use traditional relational DB structure for this data, regardless of decision for how to store the rest of the player specific data.

The leaderboard will show the pizza empire name and sign designs for the listed empires.

# Technology

This will be an online game using a traditional client / server model.

## Goals

* Performant
* Highly scalable
* Secure
* Low possibility for cheating

## Design

* Client heavy to enable easier scalability.
* RESTful API for requests from client.
  + Given the nature of the game, the server will of course have to keep track of all of the client state, so I’m not sure if this is strictly RESTful.
* Use JSON and encryption to keep payload size to a minimum.
* Client storage to hold all of the known information in the phone.
* Data verification system built into API so that when client is making a request for data (for example when the client application is shut down and re launched), only a small amount of information is required to be transferred to verify if the data on the client side is correct.
  + Every single piece of player data would be need to be transferred from the server to the client in the case where the client side data verification fails for any reason other than network issues.
    - Client data corruption.
    - Problem with data synchronization between server and client.
    - Uninstall / reinstall of app.
  + The logic of whether the client needs to request all of the data will be left up to the client.
* Whenever a client makes a request to the API that obviously requires new data to be sent that the client couldn’t have had before, the local storage / server storage check doesn’t apply.
  + For example, when the client sends a request to order raw ingredients, new data will obviously be required from the server.
* Use some industry standard RESTful authorization.
  + Basic authentication w/SSL.
  + OAuth1.
  + OAuth2.
* The RESTful API will be versioned so that there is no problem with upgrades / older versions of the client still working.
* Multiple data caching strategies to deal with the different data access patterns for different types of data.
  + Requests from the client are likely to be extremely “write heavy” in that almost every request will change the state of the data associated with that player. This will make a traditional read data cache useless for this type of data.
    - In order to achieve acceptable performance, it’s likely that we will have to think of the storage as strictly long term storage for player specific state data.
    - It’s likely that writing to the storage every time a client makes a request will make it impossible to have acceptable performance.
    - One idea to explore is to load all of a client’s data from the storage into a cache when they log in and write it back to the storage and remove it from the cache in one chunk when they log out / become inactive for a period of time.
      * This idea will cause issues with data integrity and scalability.
        + What happens if the server crashes and all of the in memory data is lost?
        + How easy is it to implement scaling caches?
      * Investigate Azure cloud caching services to see if using this service can address both the data integrity and scalability issues.
      * This might mean that all client data is actually stored in a blob of binary data in the storage instead of in a relational mapping.
      * This idea could be modified so that client specific data is only loaded from the storage into the cache as it is requested by the client instead of all at once.
        + If most of the client data is required in order to play the game then this might be less efficient because it will require more RESTful API requests.
        + This will require performance testing.
    - Need to investigate other possible designs to handle the described scenario.
  + Data that relates to the players but can’t be strictly associated with only one player.
    - For example, items that a player has made available for trade or leaderboard stats.
    - Need to think about / investigate best data access design for this type of data.
  + Data that is not player specific and likely to be accessed very frequently.
    - Should reside in an in-memory cache and only be read from the storage whenever absolutely necessary (when the server application is launched on a server for any reason).
    - For this data we can think of the data in the database as the source of truth, which will be used when a server instance is launched.
    - This data should never change except when a new version of the game is released.
    - For example, the data about the raw ingredients in the game would fall into this category including wait times for delivery, names, prices, etc. etc.
* The data access layer should be built so that any logic layers don’t care whether they are interacting with the storage or the cache.
* Respecting the above, there also needs to be a way to specify and manage the different caching strategies mentioned above: always keep in memory, pull from storage / push to storage when player logs in / logs out, shared player information.
* Need to think a lot more about the release /upgrade process for both client and server.

## Tools

### Client

* Xamarin development tools.
* OR
* Native
  + iOS
  + Android
  + Windows mobile
  + Mac app store
  + Google app store

### Server

* .NET WEB API RESTful interface hosted on Azure.
* Azure cloud storage.
* Azure cloud caching.
* Azure MS SQL Database

# RESTFul API

The RESTful API should be based around the actions that the players will take.

## List of player actions

* Login / Start Game
* Order raw ingredient
* Prepare processed ingredient
* Assemble / cook pizza
* Take phone calls
* Perform delivery
* Seat customer
* Sell product to customer
* Upgrade equipment
* Upgrade restaurant
* Hire employee
* Check leaderboard
* View trophies
* Communicate with other players
  + Trade
  + Chat