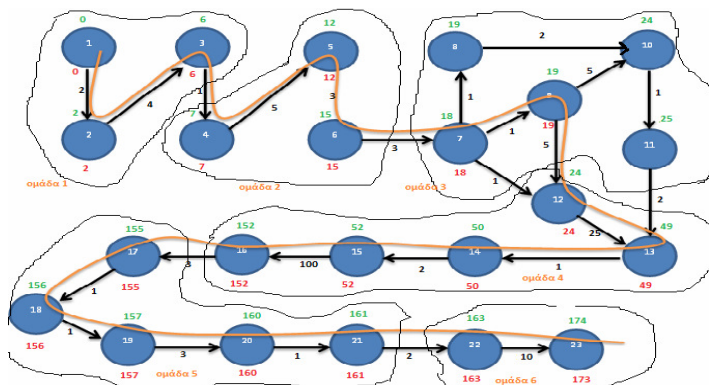




# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΨΗΦΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΘΕΜΑ: Η ΣΥΜΒΟΛΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ ΜΟΝΟΠΟΥ ΣΤΗΝ ΑΝΤΑΓΩΝΙΣΤΙΚΟΤΗΤΑ ΤΩΝ  
ΕΣΤΙΑΤΟΡΙΩΝ : «*The Food Project: Monopoly*»



ΔΙΔΑΣΚΩΝ : ΜΑΡΙΝΟΣ ΘΕΜΙΣΤΟΚΛΕΟΥΣ

ΕΠΙΜΕΛΩΝ: ΣΤΑΥΡΟΠΟΥΛΟΣ ΝΙΚΟΣ ΜΕ/11070

## Περιεχόμενα

Ευχαριστίες .....	6
Περίληψη .....	7
Abstract .....	9
ΚΕΦΑΛΑΙΟ 1ο .....	10
Πως συμβάλουν τα λογισμικά συστήματα στη διαχείριση των εστιατορίων; .....	10
Εισαγωγή .....	10
Ερευνητικό πρόβλημα .....	11
Σκοπός και αντικειμενικοί στόχοι της εργασίας .....	12
Σκοπός .....	12
Στόχοι .....	12
Δομή εργασίας .....	12
ΚΕΦΑΛΑΙΟ 2ο .....	14
Γιατί είναι πρόβλημα; .....	14
Περίληψη .....	14
Βιβλιογραφία και κριτική ανάλυση των λογισμικών συστημάτων .....	14
Περιγραφή ανοικτών θεματικών περιοχών για μελλοντική έρευνα .....	16
Συμπεράσματα .....	17
ΚΕΦΑΛΑΙΟ 3ο .....	19
Ποιος είναι ο ρόλος του λογισμικού στη διαχείριση και οργάνωση του εστιατορίου; .....	19
Περίληψη .....	19
Σκοπός και αντικειμενικοί στόχοι της επιχείρησης εστίασης .....	19
Σκοπός .....	20
Στόχοι .....	20
Περιγραφή Πρότασης .....	21
Γραφικό Περιβάλλον (Interface) Πελάτη Καφετέριας .....	24
Γραφικό Περιβάλλον (Interface) Πελάτη Εστιατορίου .....	25
Γραφικό Περιβάλλον Interface Κουζίνας – manager .....	31
Διάγραμμα πίτας για την κίνηση του μαγαζιού .....	33
Χάρτης για τα ενεργά τραπέζια ή όχι .....	33
Γραφικό Περιβάλλον Διαχειριστή ενημέρωση καταλόγου .....	34

Hardware υλοποίησης, προτάσεις φιλοξενίας της εφαρμογής .....	35
Οπτικοί ανιχνευτές (Optical Scanners) .....	35
Οθόνες αφής (Touch Screens) .....	35
Είσοδος HDMI .....	36
SmartPower για εξοικονόμηση ενέργειας.....	36
Συμβατότητα με τα πρότυπα RoHS για τη φροντίδα του περιβάλλοντος .....	36
Υπέρυθρη αφή .....	36
Ανεξαρτησία λογισμικού.....	36
Μη Λειτουργικές Απαιτήσεις.....	37
Απαιτήσεις χρήσης.....	37
Απαιτήσεις αξιοπιστίας.....	37
Απαιτήσεις επιδόσεων.....	37
Απαιτήσεις υποστήριξης.....	38
Απαιτήσεις υλοποίησης.....	38
Απαιτήσεις Βάσεων Δεδομένων .....	38
Φυσικές απαιτήσεις.....	38
Συμπεράσματα.....	39
ΚΕΦΑΛΑΙΟ 4ο.....	40
Πως θα υλοποιηθεί το λογισμικό σύστημα ;.....	40
Περίληψη .....	40
Μεθοδολογία ανάπτυξης .....	40
Ανάλυση Απαιτήσεων.....	41
Σχεδιασμός και Υλοποίηση .....	41
Επαλήθευση και Επικύρωση.....	41
Εξέλιξη.....	41
Μοντέλα ανάπτυξης λογισμικού .....	42
Μοντέλο Λειτουργικής Επαύξησης.....	42
Γλώσσα προγραμματισμού και εργαλεία υλοποίησης .....	44
Η γλώσσα C# .....	44
Η ASP.NET.....	45

Το περιβάλλον του visual studio.....	45
Η πλατφόρμα .NET Framework .....	45
Συμπεράσματα.....	47
ΚΕΦΑΛΑΙΟ 5ο .....	48
Ανάλυση απαιτήσεων .....	48
Περίληψη .....	48
Περιγραφή διαδικασίας που αυτοματοποιείτε.....	49
Εμπλεκόμενοι.....	49
Οι άνθρωποι οι οποίοι έχουν ένα ρόλο στο έργο: .....	49
Αρχιτεκτονική συστήματος .....	50
Προσδιορισμός μοντέλου του πεδίου προβλήματος.....	51
Περιορισμός Υποψηφίων κλάσεων .....	52
Παραδοχές .....	52
Προσδιορισμός μοντέλου περιπτώσεων χρήσης .....	53
Διάγραμμα περιπτώσεων χρήσης Use Case Diagram .....	54
Εικόνα 17 Διάγραμμα περιπτώσεων χρήσηςClass Diagram .....	54
Τεκμηρίωση .....	57
ΚΕΦΑΛΑΙΟ 6ο .....	58
Σχεδιασμός.....	58
Περίληψη .....	58
Βάση δεδομένων .....	58
Συμβολισμός .....	61
ΚΕΦΑΛΑΙΟ 7ο .....	64
Ποια η Συνεισφορά της Εργασίας Αυτής;.....	64
Περίληψη .....	64
Τι έμαθα που δεν ήξερα ; .....	64
Γιατί αυτό είναι σημαντικό ; .....	65
Μελλοντικές επεκτάσεις εξέλιξη .....	66
Συμπεράσματα.....	67
Παράρτημα .....	68
Περιβάλλον χρήστη Order Table .....	69
User interface.....	69
Λογικό μέρος.....	71

Περιβάλλον διαχειριστή Owner Table.....	74
User interface.....	84
Λογικό μέρος.....	86
Περιβάλλον manager.....	97
User interface.....	97
Βιβλιογραφία.....	101

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## Ευχαριστίες

Ιδιαίτερα να ευχαριστήσω τον επιβλέπον καθηγητή μου Μαρίνο Θεμιστοκλή για την συνεχή καθοδήγησή του και επίβλεψη της διπλωματικής εργασίας. Θα ήθελα επίσης να ευχαριστήσω την αδερφή μου, για την πολύτιμη βοήθειά της, το συμφοιτητή μου Χρήστο Λάλο για τις προτάσεις του και τον ρόλο του αντίλογου καθώς με βοήθησε να δω και να καλύψω κενά της εργασίας. και την Λία Τσερμίδου για την υποστήριξη της υλοποίησης της ιδέας, καθώς αποτέλεσε κίνητρο για την καταγραφή της ως διπλωματική εργασία.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

# Περίληψη

Στην παρούσα διπλωματική εργασία, επιχειρείτε η παρουσίαση ενός λογισμικού για την διαχείριση και διοίκηση καταστημάτων εστίασης. Σκοπός του λογισμικού είναι η αυτόματη παραγγελία άμεσα από τον πελάτη στην κουζίνα του καταστήματος, αλλά και η καταγραφή και αποθήκευση των παραγγελιών και η παρακολούθηση των χρόνων ετοιμασίας των παραγγελιών. Το όνομα του λογισμικού είναι monopoly και η ιδέα προέκυψε από το γνωστό στους περισσότερους παιχνίδι monopoly, καθώς ο σχεδιασμός του λογισμικού μοιάζει με το παιχνίδι. Η παρουσίαση ολοκληρώνεται μέσα σε 7 κεφάλαια. Κάθε κεφάλαιο ασχολείται με την ανάλυση κάποιου ερωτήματος. Κάθε κεφάλαιο ανοίγει με μια περιγραφή για το τι είχε αναφερθεί στο προηγούμενο και τι θα ακολουθήσει στη συνέχεια του κεφαλαίου και κλείνει με τα συμπεράσματα του κεφαλαίου, εκτός από το πρώτο το οποίο ανοίγει με μία εισαγωγή

Εκτενέστερα στο πρώτο κεφάλαιο αναλύεται το ερώτημα “πώς συμβάλουν τα λογισμικά συστήματα στην διαχείριση των εστιατορίων ;”. Προσδιορίζετε ακριβώς το ερευνητικό πρόβλημα καθώς και ο σκοπός και οι αντικειμενικοί στόχοι της διπλωματικής εργασίας.

Το δεύτερο κεφάλαιο ασχολείται με την ανάλυση του ερωτήματος “Γιατί είναι πρόβλημα ;” πιο συγκεκριμένα γιατί η διαχείριση και διοίκηση των εστιατορίων αποτελεί πρόβλημα και ως συνέπεια, η χρήση του λογισμικού monopoly έρχεται να προτείνει και να βελτιώσει το πρόβλημα αυτό. Γίνεται μία ανασκόπηση βιβλιογραφίας και το κεφάλαιο κλείνει με το συμπέρασμα ότι η διαχείριση ενός εστιατορίου δεν εξαρτάτε μόνο από το λογισμικό διαχείρισης αλλά και από τον τρόπο που θα κινηθεί η επιχείρηση εστιάζοντας στους στόχους της, αυτός είναι που θα καθορίσει την επιτυχία ή την αποτυχία.

Το τρίτο κεφάλαιο αναλύει το ερώτημα “Ποιος είναι ο ρόλος του λογισμικού στη διαχείριση και διοίκηση των εστιατορίων ;”. Εδώ προσδιορίζονται ο σκοπός και οι αντικειμενικοί στόχοι της επιχείρησης εστίασης και περιγράφεται η πρόταση του λογισμικού monopoly. Τέλος από τα συμπεράσματα κρατάμε ότι η τεχνολογία εξελίσσεται με αλματώδη βήματα και ο άνθρωπος ακολουθεί, η ζήτηση της αγοράς μεταβάλετε σε όλους τους τομείς έτσι και ο τομέας της εστίασης δίνετε να ακολουθήσει και οι επιχειρήσεις να είναι συμβατές με την ζήτηση της αγοράς και αυτή την δυνατότητα την προσφέρει το λογισμικό monopoly.

Το τέταρτο κεφάλαιο ασχολείται με το “Πως θα υλοποιηθεί το λογισμικό σύστημα;”.

Στο πέμπτο κεφάλαιο γίνεται η ανάλυση απαιτήσεων, περιγράφεται η διαδικασία που αυτοματοποιείτε, παρουσιάζετε η αρχιτεκτονική του συστήματος καθώς και το διάγραμμα περιπτώσεων χρήσης (Use Case Diagram) και το διάγραμμα κλάσεων (Class Diagram).

Στο έκτο κεφάλαιο γίνεται ο σχεδιασμός , παρουσιάζετε η βάση δεδομένων με τους πίνακες, τα πεδία και τις συσχετίσεις μεταξύ τους.

Το έβδομο και τελευταίο κεφάλαιο ασχολείται με το ερώτημα “Ποια η συνεισφορά της εργασίας αυτής;”. Γίνετε μία ανασκόπηση για το τι έμαθα που δεν ήξερα, γιατί αυτό είναι σημαντικό αλλά και για τις μελλοντικές επεκτάσεις του λογισμικού monopoly. Τέλος το κεφάλαιο κλείνει με τα συμπεράσματα από τα οποία κρατάμε ότι, το λογισμικό monopoly έρχεται να καλυτέρευση τη διαχείριση και διοίκηση ενός καταστήματος εστίασης, σίγουρα μπορεί να οργανώσει μια επιχείρηση και ιδιαίτερα επιχειρήσεις όπως τα

Goodies και Mc.Donald τα οποία έχουν συγκεκριμένο μενού και θα μπορούσε σε αυτά, να δώσει μία λύση για την συνολική διοίκηση και διαχείριση τους.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ



## Abstract

In this paper , we attempt to present a software for the management and administration of catering outlets . The purpose of the software is the automatic ordering directly from the client's to kitchen, but the recording and storage of orders and monitoring of time preparing orders. The name of the software is monopoly and the idea came from the familiar to most game monopoly, as the design of software like game . The presentation is completed within 8 chapters and at the end, in Annex A business process instance log restaurant . Each chapter deals with the analysis of a query. Each chapter opens with a description of what had been reported in the past and what will follow later in this chapter and closes with the conclusions of the chapter, except the first which opens with an introduction and chapters six, seven , eight and nine which have not and conclusions concerning the four stages for the implementation of software , these are respectively the stages of analysis , the stage of planning and implementation stage of the verification and validation stage of evolution.

In the first chapter analyzes the question "how software systems help in management of restaurants ? " . Specify exactly the research problem and the purpose and objectives of this this .

The second chapter deals with the analysis of the question " Why is it a problem? " Specifically because the management and administration of restaurants is a problem and as a consequence , the use of software monopoly comes to propose and improve this problem. Becomes a literature review and the chapter ends with the conclusion that the management of a restaurant depends not only on the management software but also on the way that will move the company focusing its objectives, this is what will determine success or failure.

The third chapter analyzes the question " What is the role of software in the management and administration of restaurants ? " . Here specify the purpose and objectives of the business focus and describe the proposed software monopoly. Finally the conclusions we hold that technology evolves strides and man follows the market demand are altered in all areas so the area of focus you give to follow and businesses be compatible with market demand and this feature the software offers monopoly.

The fourth chapter deals with " How to implement the software system? " .

In the fifth chapter become the requirements analysis describes the process that you are automating , presents the system architecture and the use case diagram (Use Case Diagram) and class diagram (Class Diagram).

In the sixth chapter, the design shows the database tables, fields and relationships between them.

The seventh and final chapter deals with the question " What is the contribution of this work ? " . Become an overview of what I learned I did not know why this is important , but also for future extensions of software monopoly. Finally, the chapter ends with the conclusions of which hold that the software monopoly comes to bettering the management and administration of a shop catering can certainly organize a business , especially businesses like Goodies and Mc.Donald which have this menu and able to them, to give a solution for the overall administration and management.

# ΚΕΦΑΛΑΙΟ 1ο

## Πως συμβάλουν τα λογισμικά συστήματα στη διαχείριση των εστιατορίων;

### Εισαγωγή

Η πολυπλοκότητα των προβλημάτων, που αντιμετωπίζουν σήμερα οι επιχειρήσεις, ο γρήγορος ρυθμός μεταβολών των συνθηκών του οικονομικού, τεχνολογικού, κοινωνικού και πολιτικού περιβάλλοντος, η ένταση του ανταγωνισμού στο διεθνή χώρο και οι ενοποιήσεις των αγορών, καθιστούν αναγκαίο τον προγραμματισμό για την ανάπτυξη και επιβίωση των επιχειρήσεων. Στα πλαίσια αυτού οι επιχειρηματίες, και τα στελέχη όλων των βαθμίδων των επιχειρήσεων και οργανισμών πρέπει:

- Να μάθουν να διευρύνουν τον ορίζοντα της δράσης τους.
- Να γίνουν πιο συνειδητοί με το περιβάλλον, να βελτιώσουν την μεθόδευση των ενεργειών τους και να αντιδρούν δυναμικά στις μεταβαλλόμενες συνθήκες του περιβάλλοντος της επιχείρησης.
- Να έχουν μακροχρόνιες στρατηγικές, χωρίς όμως να αμελούν τα βραχύχρονα προβλήματα της επιχείρησης και να λαμβάνουν αποφάσεις με σκοπό τη μεγιστοποίηση του κέρδους της επιχείρησης με το λιγότερο δυνατόν κόστος (Κιόχος 2001).

Η ανάλυση παραγόντων όπως κοινωνικοοικονομικοί, τεχνολογικοί, προμήθειας κοκ. που χαρακτηρίζουν το εξωτερικό περιβάλλον μίας επιχείρησης είναι αναγκαία καθώς η ερμηνεία του αποτελέσματος είναι αυτή που θα καθορίσει την κατάλληλη στρατηγική που πρέπει να εφαρμόσει η επιχείρηση ώστε να επωφεληθεί από μια επιχειρηματική πρόταση/επένδυση ή να αντιμετωπίσει αποτελεσματικά μια ενδεχόμενη ζημία.

Η τεχνολογική εξέλιξη επηρεάζει τις επιχειρήσεις με διάφορους τρόπους. Τεχνολογικές καινοτομίες συμβάλουν στη μεταβολή της ζήτησης ενός προϊόντος ή υπηρεσίας μιας επιχείρησης. Κατά ένα βαθμό επηρεάζουν την αξιοπιστία του προϊόντος και προσθέτουν ποιοτικά χαρακτηριστικά στην υπηρεσία ή το προϊόν. Η έλλειψη προσθήκης των τεχνολογικών εξελίξεων από μια επιχείρηση ίσως την οδηγήσουν στον περιορισμό των πωλήσεων και την μείωση των κερδών της επιχείρησης εφόσον είναι δυνατόν το σύστημα παραγωγής του τελικού προϊόντος της επιχείρησης να χαρακτηριστεί απαρχαιωμένο και το προϊόν «ξεπερασμένο». Επιπλέον, πολλές φορές οι τεχνολογικές εξελίξεις δύναται να συμβάλουν στη μείωση του κόστους παραγωγής.

Αξίζει να αναφέρουμε ότι κάποιες επιχειρήσεις όπως η κολοσσιαία επιχείρηση της Coca Cola με την κυρίαρχη θέση της στο χώρο της αγοράς εν γένει, διόλου επηρεάζεται από την τεχνολογική ανάπτυξη σε όλους τους τομείς. Επικεντρώνετε περισσότερο στην βελτίωση και ανανέωση τόσο της εικόνας του προϊόντος όσο και του μάρκετινγκ μέσω των ποικίλων διαφημίσεων και άλλων δραστηριοτήτων.

Η διαπίστωση της επίδρασης των τεχνολογικών εξελίξεων στους στόχους και τις στρατηγικές μιας επιχείρησης είναι πολύ ουσιώδες καθήκον. Η πρόβλεψη των τεχνολογικών μεταβολών είναι πιθανόν να μην είναι πολύ δύσκολη, όσο είναι η επιλογή του κατάλληλου χρόνου για αντίδραση και ο προσδιορισμός των συνεπειών των τεχνολογικών μεταβολών (Κιόχος 2001).

Η τεχνολογική και κοινωνική πρόβλεψη είναι δύο από τους βασικούς παράγοντες που συντελούν στην υγιή πορεία και εξέλιξη της επιχείρησης. Η κοινωνική πρόβλεψη είναι η προσπάθεια κατανόησης της φύσης της

μεταβαλλόμενης κοινωνίας και η εξακρίβωση της επιρροής των μεταβαλλόμενων αξιών και συμπεριφορών των μελών της κοινωνίας στην επιχείρηση. Η τεχνολογική πρόβλεψη ενδιαφέρετε για την πρόβλεψη μελλοντικών τεχνολογικών αναπτύξεων καθώς και για την επίδραση που θα έχουν τέτοιες αναπτύξεις σε μια επιχείρηση.

Στα πλαίσια της τεχνολογικής πρόβλεψης είναι και η χρήση λογισμικών σε διάφορα στάδια δράσης για την επιχείρηση. Ως λογισμικό (Software) νοείται τόσο τα προγράμματα που συντονίζουν και κατευθύνουν την λειτουργία, αλλά και επεξεργάζονται τα δεδομένα ενός υπολογιστή ή άλλης συσκευής, όσο και το βοηθητικό υλικό (έντυπα ή ηλεκτρονικά έγγραφα), που παρουσιάζουν τη χρήση και τη λειτουργία των προγραμμάτων αυτών. Το λογισμικό διακρίνεται σε δύο βασικές κατηγορίες χωρίς όμως σαφή όρια διαχωρισμού. Αυτές οι κατηγορίες είναι το λογισμικό εφαρμογών (Application software) και το λογισμικό συστήματος.

Το λογισμικό εφαρμογών αποτελείται από όλα εκείνα τα προγράμματα που καθορίζουν τον τρόπο με τον οποίο θα λειτουργήσει το υλικό, ώστε να μπορέσει ο χρήστης να ικανοποιήσει συγκεκριμένες απαιτήσεις του, βρίσκετε κοντά στην αντίληψη που έχει ο χρήστης για τις εφαρμογές, αφού του παρέχει την κατάλληλη διεπαφή, για να διεκπεραιώσει τις εργασίες του και το λογισμικό συστήματος (System software) αποτελείται από εκείνα τα προγράμματα που έχουν σκοπό να βοηθούν και να υποστηρίζουν τις εφαρμογές λογισμικού αλλά και την επικοινωνία του χρήστη, με τις εφαρμογές.

Στην παρούσα εργασία θα ασχοληθούμε με ένα λογισμικό εφαρμογής. Το λογισμικό αυτό συμβάλει στη βελτίωση της διαχείρισης και της οργάνωσης ενός εστιατορίου και δίνει τη δυνατότητα στον πελάτη να πραγματοποιεί μόνος του την παραγγελία από οθόνη αφής που είναι ενσωματωμένη στο τραπέζι του εστιατορίου.

## Ερευνητικό πρόβλημα

Η παρούσα διπλωματική εργασία επιχειρεί να αναλύσει την διαδικασία με την οποία μια λογισμική εφαρμογή διευκολύνει την διαχείριση του εστιατορίου. Το πρόβλημα της διαχείρισης ενός εστιατορίου δεν εμφανίζεται για πρώτη φορά, πληθώρα εστιατορίων ανοίγουν και σε μικρό χρονικό διάστημα κλείνουν λόγω κακής διαχείρισης. Η διαχείριση ενός εστιατορίου ξεκινά από την παρακολούθηση των εμπορευμάτων που αποτελούν τα προϊόντα που θα έχει ο κατάλογος, κάθε πιθανή έλλειψη είναι δυσφήμιση για το εστιατόριο, και προκαλεί δυσαρέσκεια στον πελάτη καθώς δεν ικανοποιεί τις προσδοκίες του.

Έχουν γραφεί πληθώρα βιβλίων για τον αντίκτυπο που έχει ένας δυσαρεστημένος πελάτης επηρεάζοντας αρνητικά άλλους 20 για παράδειγμα, ενώ ένας ευχαριστημένος προσελκύει μόνο οχτώ νέους πελάτες.

Διαχείριση όμως είναι και ο συντονισμός της κουζίνας, του προσωπικού σερβιρίσματος και η οργάνωση των χρόνων ετοιμασίας και αποστολής της παραγγελίας. Όταν τα προϊόντα και οι υπηρεσίες του εστιατορίου ικανοποιούν τους πελάτες, τότε το ίδιο αποκτά και εδραιώνει καλή φήμη. Η πολιτική ποιότητας είναι αυτή η οποία θα κάνει το εστιατόριο να εμπνέει εμπιστοσύνη, να προσελκύει και να διατηρεί νέους πελάτες και κατά αυτόν τον τρόπο να πετυχαίνει τους στόχους του, να διασφαλίζει τη θέση του στην αγορά και να έχει προοπτικές για να ανοίξει αλυσίδα καταστημάτων.

Συνοψίζοντας το ερευνητικό πρόβλημα που θα μελετήσουμε είναι πώς ένα λογισμικό μπορεί να προσφέρει την αποτελεσματικότερη διαχείριση σε ένα εστιατόριο και να συμβάλει στην επίτευξη των στόχων της επιχείρησης.

## Σκοπός και αντικειμενικοί στόχοι της εργασίας

Ο σκοπός και οι αντικειμενικοί στόχοι προσδιορίζουν το λόγο για τον οποίο γίνεται η συγγραφή αυτής της πτυχιακής εργασίας και προσανατολίζουν των αναγνώστη στο θέμα έρευνας.

### Σκοπός

Σκοπός της εργασίας είναι η σύνδεση της επιστημονικής έρευνας, με το σχεδιασμό σε πρακτικό επίπεδο μίας λογισμικής εφαρμογής, για την βελτίωση της διοίκησης και διαχείρισης μιας επιχείρησης εστίασης η οποία θα μπορεί να καλύπτει και όλους τους στόχους της επιχείρησης.

### Στόχοι

- Η ανεύρεση, διατύπωση και τεκμηρίωση της σημαντικότητας ενός προβλήματος που έχει άμεση σχέση με τη χρήση των λογισμικών σε καταστήματα εστίασης.
- Η διερεύνηση και παρουσίαση του προβλήματος με την ανασκόπηση της σχετικής με το θέμα πρόσφατης βιβλιογραφίας.
- Ο σχεδιασμός και η διεξαγωγή μιας επιστημονικής έρευνας ή μιας κριτικής ερευνητικής ανασκόπησης.
- Η συγγραφή και παρουσίαση μιας ολοκληρωμένης επιστημονικής εργασίας που να προτείνει λύσεις στο παραπάνω πρόβλημα.

## Δομή εργασίας

**Στο πρώτο κεφάλαιο** γίνεται προσπάθεια να κατανοήσουμε την συμβολή των λογισμικών συστημάτων στην διαχείριση των εστιατορίων. Ακόμα γίνεται μια εισαγωγή για την τεχνολογική εξέλιξη και την επιρροή στις επιχειρήσεις, περιγράφεται επακριβώς το ερευνητικό πρόβλημα το οποίο θα αναλυθεί στην παρούσα εργασία, οι σκοποί και οι αντικειμενικοί στόχοι της επιχείρησης και τέλος η δομή της εργασίας.

**Το δεύτερο κεφάλαιο** διευκρινίζει γιατί η διαχείριση των εστιατορίων αποτελεί πρόβλημα, αναφέρεται σχετική βιβλιογραφία και επίσης γίνεται αναφορά ανοικτών θεματικών περιοχών για μελλοντική έρευνα και τέλος καταγράφονται τα συμπεράσματα.

**Το τρίτο κεφάλαιο** διευκρινίζει ποιος είναι ο ρόλος του λογισμικού στη διαχείριση και οργάνωση του εστιατορίου. Αναφέρονται οι σκοποί και οι αντικειμενικοί στόχοι της επιχείρησης εστίασης από τη χρήση της λογισμικής εφαρμογής και περιγράφεται η πρόταση για την βελτίωση της διαχείρισης και της οργάνωσης ενός εστιατορίου. Η περιγραφή της πρότασης περιλαμβάνει το γραφικό περιβάλλον του πελάτη είτε αυτό πρόκειται για καφετέρια, είτε για εστιατόριο, το γραφικό περιβάλλον της κουζίνας, δηλαδή τον πίνακα που θα παρακολουθεί ο manager και στον οποίο θα εμφανίζονται οι παραγγελίες και γίνεται αναλυτική περιγραφή της λειτουργίας του συστήματος. Ακόμα αναφέρονται κάποια χαρακτηριστικά του προγράμματος μορφοlogy όπως το διάγραμμα για την κίνηση του μαγαζιού, ο χάρτης για τα ενεργά τραπέζια αλλά και τα μη ενεργά.

Στην πορεία παρουσιάζετε το γραφικό περιβάλλον του διαχειριστή. Τέλος αναφέρονται 2 προτάσεις ως hardware φιλοξενίας της εφαρμογής, γίνετε μία περιγραφή για τον τρόπο λειτουργίας και την τεχνολογία που έχουν οι οθόνες αφής και το κεφάλαιο 3 κλείνει με τα συμπεράσματα.

**Το τέταρτο κεφάλαιο** αναφέρεται στη μεθοδολογία ανάπτυξης την οποία θα ακολουθήσουμε για την κατασκευή του μονοπολύ. Η ανάπτυξη ενός λογισμικού συστήματος απαιτεί κάποιες δραστηριότητες αλλά και κάποιο μοντέλο ή μεθοδολογία ανάπτυξης. Αυτές οι δραστηριότητες και η μεθοδολογία ανάπτυξης αναλύονται στο 4<sup>ο</sup> κεφάλαιο. Τα τέλος γίνετε αναφορά για τα τη γλώσσα προγραμματισμού η οποία αποτελεί δομικό στοιχείο για την ανάπτυξη του λογισμικού, και είναι η C#, καθώς και τα εργαλεία υλοποίησης.

**Το πέμπτο κεφάλαιο** περιγράφει τη διαδικασία που αυτοματοποιείτε, με τη χρήση του λογισμικού συστήματος αυτού καθ αυτού. Στη συνέχεια αναφέρονται οι εμπλεκόμενοι στο πληροφοριακό σύστημα. Αναλύονται τα ερωτήματα “Τι θα κάνει το σύστημα παραγγελίας?”, “Ποιός θα χρησιμοποιεί το σύστημα?”, “Πού θα φιλοξεניתε το σύστημα”, “Πότε θα χρησιμοποιείτε το σύστημα”. Αναλύεται η αρχιτεκτονική του συστήματος με τη συνοδεία περιγραφικού σχήματος. Γίνεται προσδιορισμός του μοντέλου του πεδίου προβλήματος και περιορισμός των υποψηφίων κλάσεων για την επιλογή των τελικών κλάσεων του συστήματος. Αναπαριστάτε η λειτουργία του χρήστη από την πλευρά του ίδιου με τη χρήση διαγράμματος περιπτώσεων χρήσης (Use Case Diagram) αλλά και η μοντελοποίηση της στατικής δομής – όψης του συστήματος, με την αναπαράσταση διαγράμματος κλάσεων (Class Diagram). Με την μοντελοποίηση του συστήματος μειώνετε το σημασιολογικό χάσμα μεταξύ του πραγματικού κόσμου και του κόσμου του λογισμικού. Σαν αποτέλεσμα αυτού οι κλάσεις που εμφανίζονται είναι πέντε και όχι τέσσερις. Οι κλάσεις αυτές είναι: Η κλάση **Παραγγελία Πελάτη** (Customer Order), η κλάση **Σύστημα** (System), η κλάση **Υπεύθυνος Συντονισμού** (Manager Table), η κλάση **Ιδιοκτήτης** (Owner) και η κλάση **Στοιχεία** (Element). Γίνετε η τεκμηρίωση του παραπάνω διαγράμματος και αναφέρονται οι μη λειτουργικές απαιτήσεις.

**Το έκτο κεφάλαιο** αναφέρετε στο σχεδιασμό. Καταγράφονται τα πεδία και οι πίνακες, προκειμένου να σχεδιαστεί η βάση δεδομένων. Οι πίνακες είναι δύο βασικοί ο πίνακας Στοιχεία (Elements) και ο πίνακας Παραγγελίες (Orders). Στη συνέχεια γίνεται μια περιγραφή για τον ρόλο του κάθε πεδίου στους πίνακες, αλλά και η σχεσιακή ανάλυσή των δύο πινάκων.

**Το έβδομο κεφάλαιο** αναπτύσσεται πάνω στο ερώτημα “Ποια η συνεισφορά της εργασίας αυτής ;”. Ποιο συγκεκριμένα προσδιορίζετε τι έμαθα από την ολοκλήρωση της διπλωματικής εργασίας, αλλά και γιατί είναι αυτό σημαντικό και πώς η εργασία μπορεί να συνεισφέρει σε μία επιχείρηση εστίασης. Το κεφάλαιο κλείνει με τις μελλοντικές επεκτάσεις του συστήματος μονοπολύ και της εργασίας όπως η άμεση παραγγελία πρώτων υλών, περισσότερα στατιστικά στοιχεία, προσθήκη σχολίων για τα πιάτα κατά την παραγγελία αλλά και μετά για το πόσο καλά η όχι ήταν, σύνδεση του λογισμικού με CRM, δυνατότητα εξέλιξης της διπλωματικής με ολοκληρωμένη οικονομική και τεχνική μελέτη για το κόστος και την εγκατάσταση του λογισμικού.

## ΚΕΦΑΛΑΙΟ 2ο

### Γιατί είναι πρόβλημα;

#### Περίληψη

Στο πρώτο κεφάλαιο υπήρξε μία εισαγωγή για την πολυπλοκότητα των προβλημάτων που αντιμετωπίζουν οι επιχειρήσεις σήμερα, την ένταση του ανταγωνισμού στο διεθνή χώρο και την αναγκαιότητα του προγραμματισμού για την ανάπτυξη και επιβίωση των επιχειρήσεων. Οι αναλύσεις και η σωστή ερμηνεία σε παράγοντες εξωτερικού περιβάλλοντος δίνουν στην επιχείρηση την ικανότητα να αναγνωρίσει μια ευκαιρία ή και να αντιμετωπίσει αποτελεσματικά μια ενδεχόμενη ζημία. Αναφερθήκαμε στην πρόβλεψη κοινωνικών και τεχνολογικών παραγόντων ως δύο από τους βασικούς που συντελούν στην υγιή πορεία και εξέλιξη της επιχείρησης. Στα πλαίσια του δεύτερου παράγοντα αναφερθήκαμε στην αναγκαιότητα χρήσης λογισμικού το οποίο θα συμβάλει στην βελτίωση και τη διαχείριση του εστιατορίου. Προσδιορίστηκε επακριβώς το ερευνητικό πρόβλημα της διαχείρισης ενός εστιατορίου αλλά και ανάλυση της διαδικασίας με την οποία ένα λογισμικό μπορεί να την διευκολύνει. Επίσης προσδιορίστηκαν οι σκοποί και οι αντικειμενικοί στόχοι της επιχείρησης. Σε αυτό το κεφάλαιο καταγράφεται η σχετική βιβλιογραφία για τα λογισμικά συστήματα και την εφαρμογή τους στις επιχειρήσεις, καθώς και η κριτική ανάλυση τέλος διατυπώνετε το συμπέρασμα των παραπάνω.

#### Βιβλιογραφία και κριτική ανάλυση των λογισμικών συστημάτων

Η βιβλιογραφία για την εφαρμογή λογισμικών συστημάτων σε επιχειρήσεις στο χώρο της εστίασης είναι πολύ περιορισμένη. Ωστόσο, κατά καιρούς έχουν γραφτεί διάφορα βιβλία γενικού περιεχομένου για λογισμικά και επιχειρήσεις.

Η παράγοντες που επηρεάζουν την πορεία μια επιχείρησης είναι σύνθετοι και περίπλοκοι και δεν μπορούν να περιγραφούν με όρους σταθερούς στο χώρο και χρόνο. Υπάρχουν διαφορετικές απόψεις και προσεγγίσεις που ανάλογα με τη σκοπιά τους, φωτίζουν διαφορετικές πτυχές. Μια προσέγγιση που κυριαρχεί στο χώρο της επιχειρηματικότητας είναι ότι η επιχειρηματικότητα και όχι η τεχνολογία είναι αυτή η οποία καθορίζει την επιτυχία ή την αποτυχία σε μία επιχείρηση (Cusumano, 1999). Για τον ορισμό της επιχειρηματικότητας έχουν κατά καιρούς δοθεί πολλοί ορισμοί. Μερικοί από αυτούς είναι ότι επιχειρηματικότητα είναι μια διαδικασία δημιουργίας, ή η σύλληψη μιας ευκαιρίας και η επιδίωξη εκμετάλλευσής της, ανεξάρτητα από τον έλεγχο, ή την κατοχή των μέσων υλοποίησής της (Timmons & Spinelli 2007). Ακόμα ένας ορισμός είναι ότι επιχειρηματικότητα δεν είναι ούτε επιστήμη αλλά ούτε και τέχνη, αλλά είναι η πρακτική στην οποία η γνώση αποτελεί το μέσο για την επίτευξη του σκοπού της. Είναι τρόπος συμπεριφοράς, παρά χαρακτηριστικό της προσωπικότητας (Drucker 2003). Τέλος η επιχειρηματικότητα ορίζεται ανάλογα με την οπτική, οικονομική ή διοικητική, που την ορίζει ο καθένας (Hebert & Link). Θα μπορούσαμε να συμπεριλάβουμε σαν βήματα της επιχειρηματικότητας την καινοτομία, την ηγεσία, την διαχείριση και την ανάπτυξη της πρωτοβουλίας. Η πρωτοβουλία της εισαγωγής ενός λογισμικού για την καλύτερη διαχείριση της επιχείρησης και η καινοτομία



του λογισμικού μπορούν να αποφέρουν κέρδος αλλά και αβεβαιότητα στην επιχείρηση. Οι επιχειρήσεις θα πρέπει να ορίζουν υψηλές απαιτήσεις για την ποιότητα ενός λογισμικού η επαλήθευση και η επικύρωση των απαιτήσεων είναι από τις πιο σημαντικές διαδικασίες μέσα στην επιχείρηση, ακόμα η υψηλής ποιότητας λογισμικό απαιτεί σωστή εφαρμογή εργαλείων, μεθόδων και διαδικασιών (Gottesdiener, 2005).

Επίσης είναι σημαντικό να αναφέρουμε ότι σύμφωνα με την Ellen Gottesdiener οι απαιτήσεις ενός λογισμικού διαχωρίζονται σε 3 επίπεδα, στο πρώτο επίπεδο της πυραμίδας συναντάμε τις απαιτήσεις της επιχείρησης, σε αυτό το επίπεδο περιλαμβάνεται το όραμα για το λογισμικό του προϊόντος, που καθοδηγείτε από τους στόχους της επιχείρησης, το αντικείμενό της και την στρατηγική της επιχείρησης. Στο δεύτερο επίπεδο συναντάμε τις απαιτήσεις του χρήστη, σε αυτό το επίπεδο περιλαμβάνονται τα καθήκοντα που ο χρήστης πρέπει να ολοκληρώσει με το λογισμικό και τα απαραίτητα ποιοτικά χαρακτηριστικά του λογισμικού, ευκολία χρήσης οικειότητα στον χρήστη κ.τ.λ. Τέλος στο τρίτο επίπεδο στην βάση της πυραμίδας συναντούνται οι απαιτήσεις του λογισμικού, είναι λεπτομερείς περιγραφές από το σύνολο των λειτουργικών και μη λειτουργικών απαιτήσεων στις οποίες πρέπει το λογισμικό να ανταποκρίνεται. Οι απαιτήσεις πρέπει να καλύπτουν τις ανάγκες της επιχείρησης και τις ανάγκες των χρηστών, μένοντας πάντα εντός των ορίων του σχεδιασμού και υλοποίησης.

Η αναπτυξιακή διαδικασία ενός λογισμικού έργου μπορεί να δημιουργήσει πληθώρα προβλήματα. Ο Edward Hasted ένας από τους πρωτοπόρους στον τομέα των λογισμικών και την εφαρμογή τους στις επιχειρήσεις, στο βιβλίο του αναφέρει ότι κατά την αναπτυξιακή διαδικασία ενός λογισμικού δημιουργούνται πληθώρα προβλημάτων τα οποία χρειάζονται αντιμετώπιση, προκειμένου να εξασφαλιστεί η επιτυχία. Οι επιχειρήσεις πρέπει να δίνουν έμφαση στην αντιμετώπιση των κρίσιμων βημάτων, για την αντιμετώπιση συνολικά των αναπτυξιακών προβλημάτων λαμβάνοντας καίριες και αποτελεσματικές αποφάσεις (Hasted, 2005).

Τα προβλήματα της ανάπτυξης ενός λογισμικού σε μία επιχείρηση ή ενός πληροφοριακού συστήματος όμως δεν είναι μόνο κατά την διαδικασία της ανάπτυξης. Ο Μαρίνος Γ. Θεμιστοκλέους & η Βασιλική Γ. Μαντζάνα Στο βιβλίο τους “υπηρεσίες παγκόσμιου ιστού και Υπηρεσιοστρεφείς Αρχιτεκτονικές” web services<sup>1</sup> και soa<sup>2</sup> και στην προσπάθειά να προβάλουν την ολοένα και αυξανόμενη ανάγκη που υπάρχει για την κατανόηση και χρήση των Υπηρεσιοστρεφών Αρχιτεκτονικών και των υπηρεσιών παγκόσμιου ιστού, αναφέρουν με ακρίβεια ότι τις τελευταίες δεκαετίες, οι επιχειρήσεις-οργανισμοί έχουν εστιάσει στην Τεχνολογία της Πληροφορικής (Information Technology - IT) αναπτύσσοντας πληθώρα εφαρμογών προκειμένου να βελτιώσουν και να αυτοματοποιήσουν τις επιχειρησιακές τους διαδικασίες. Οι εφαρμογές αυτές δεν υλοποιήθηκαν σύμφωνα με ένα κοινό στρατηγικό πλάνο ανάπτυξης ολοκληρωμένης πληροφορικής αρχιτεκτονικής, αλλά με βάση τις ανάγκες του κάθε τμήματος της επιχείρησης και ανάλογα πάντα με τις τρέχουσες τεχνολογίες αιχμής. Έτσι, στους περισσότερους οργανισμούς (επιχειρήσεις) αναπτύχθηκε μία πληροφοριακή υποδομή που αποτελείται από μάζα αυτόνομων και σε αρκετές περιπτώσεις ετερογενών Πληροφοριακών Συστημάτων (ΠΣ). Με την πάροδο των χρόνων η ανάγκη για αυτοματοποιημένες και ολοκληρωμένες επιχειρησιακές διαδικασίες αυξήθηκε με αποτέλεσμα οι οργανισμοί να αδυνατούν να επιτύχουν το στόχο της ολοκλήρωσης καθώς, ετερογενείς εφαρμογές τους παρουσίαζαν αρκετά προβλήματα διασύνδεσης. Οι εφαρμογές που αποτελούσαν την πληροφοριακή υποδομή ενός οργανισμού δεν μπορούσαν να επικοινωνήσουν μεταξύ τους

<sup>1</sup> Web Services, οι υπηρεσίες παγκόσμιου ιστού (ΥΠΙ) είναι μικρές αυτόνομες, ανεξάρτητες εφαρμογές (ή τμήματα λογισμικού) που επιτρέπουν σε άλλα προγράμματα (ετερογενή ή μη) να πραγματοποιούν μια σειρά από ενέργειες.

<sup>2</sup> Η (SOA)Service Oriented Architecture είναι μια αρχιτεκτονική δομή λογισμικού που επιτρέπει στην επιχείρηση να γίνεται πιο ευέλικτη και να ανταποκρίνεται πιο γρήγορα στις απαιτήσεις της αγοράς.

με αποτέλεσμα το όραμα για μία ολοκληρωμένη πληροφοριακή υποδομή να παραμένει ανέφικτο για αρκετά χρόνια (Θεμιστοκλέους και Μαντζάνα, 2010).

Ο Michael Mc Grath εκτελεστικός πρόεδρος του ομίλου Thomas έχει συγγράψει αρκετά βιβλία που αφορούν λογισμικά και επιχειρήσεις. Ο Michael Mc Grath δίνει βάση στην στρατηγική προϊόντος και σε δύο από τα πιο σημαντικά του βιβλία αναφέρει ότι ένας από τους βασικούς καθοριστικούς παράγοντες επιτυχίας για τις εταιρείες υψηλής τεχνολογίας του σήμερα είναι η στρατηγική προϊόντος. Οι στρατηγικές ανάπτυξης και η επιχειρηματικότητα του διαδικτύου προσφέρουν στην επιχείρηση ανταγωνιστικότητα, κερδοφορία και ανάπτυξη. Αναφέρει ότι η επιχείρηση μπορεί να επιτύχει υψηλότερα επίπεδα ανάπτυξης παραγωγικότητας μέσω νέων πρακτικών και συστημάτων για την ευθυγράμμιση του προϊόντος με την στρατηγική της επιχείρησης (Mc Grath, 2000).

Η Linda Pinson αναφέρει ότι οι στόχοι του στρατηγικού marketing είναι η δημιουργία ισχυρής επωνυμίας για το προϊόν, η κράτηση αρχείου βάσης δεδομένου για τους πελάτες και τις συνήθειες τους και η αύξηση των πωλήσεων του προϊόντος ή της υπηρεσίας. Η επιχείρηση είναι απαραίτητο να ακολουθεί ένα οργανωτικό σχέδιο για την επίτευξη των στόχων της. Το οργανωτικό σχέδιο είναι το τμήμα του επιχειρηματικού σχεδίου, που καλύπτει την περιγραφή των προϊόντων και υπηρεσιών και την διοικητική ρύθμιση της επιχείρησης. Οι λεπτομέρειες του πώς η επιχείρηση τοποθετείτε προκειμένου να λειτουργήσει με αποτελεσματικό και οικονομικά αποδοτικό τρόπο (Pinson 2003).

Η έννοια της επιτυχίας για ένα εστιατόριο είναι αυτή που περιέχει ένα μίγμα των μεταβλητών που είναι συνολικά καλύτερο από αυτά που προσφέρονται από τους ανταγωνιστές. Αυτές οι μεταβλητές είναι οι ποιότητα, μενού, service, φαγητό, τιμή, τοποθεσία, ατμόσφαιρα και διαχείριση. (Kostas Katsigris, Chris Tomas, 2008)

## **Περιγραφή ανοικτών θεματικών περιοχών για μελλοντική έρευνα**

Μετά από έρευνα σε βιβλιοθήκες και διαδίκτυο, παρατηρούμε ότι η βιβλιογραφία για τα λογισμικά στις επιχειρήσεις εστίασης είναι περιορισμένη, πολλά βιβλία και συγγράμματα έχουν γραφτεί για πληροφοριακά συστήματα και για την συμβολή τους στις επιχειρήσεις, για τον τρόπο σχεδίασης, ανάπτυξης και χρήσης τους στις επιχειρήσεις, για τα σημεία που πρέπει να προσέξουν οι επιχειρήσεις ώστε το πληροφοριακό σύστημα να έχει απόδοση στην επιχείρηση, για την αξιολόγηση τους και πολλά άλλα. Επίσης βιβλία έχουν γραφτεί για τα εστιατόρια τον τρόπο λειτουργίας τους, το περιβάλλον, τη μουσική και γενικά για την διοίκηση ενός εστιατορίου.

Η Χρήση λογισμικών όμως σε επιχειρήσεις εστίασης είναι ένας τομέας όπου η βιβλιογραφία είναι περιορισμένη. Ο χορός της εστίασης είναι ένας χώρος που απασχολεί ιδιοκτήτες εστιατορίων αλλά και κάθε είδους καταστημάτων εστίασης, καθώς και τον οποιονδήποτε επισκέπτη, ας μην ξεχνάμε ότι η ανάγκη τροφής σύμφωνα με την πυραμίδα του Maslow είναι βασική ανάγκη, άρα ο τομέας της εστίασης αποτελεί θεματική περιοχή για μελλοντική έρευνα προσφέροντας λύσεις για την αποτελεσματικότερη διαχείριση των εστιατορίων.



## Συμπεράσματα

Η ανάγκη τροφής είναι σύμφωνα με την πυραμίδα του Maslow βασική ανάγκη και βρίσκετε μαζί με την ανάγκη για νερό, καταφύγιο, ζεστασιά και άλλες στις φυσιολογικές ανάγκες. Καλύπτοντας αυτές τις ανάγκες επιβίωσης που βρίσκονται στην βάση της πυραμίδας ο άνθρωπος μπορεί να αναζητά και να επιδιώκει την κάλυψη και των υπόλοιπων αναγκών. Ο χώρος της εστίασης αναφέρετε στην κάλυψη βασικών ανθρώπινων αναγκών όπως το φαγητό, όπου και αποτελεί κυρίαρχο μέρος του τουρισμού. Η ανάγκη τροφής, η Ελληνική κουζίνα αλλά και η γνωστή για τις ευεργετικές ικανότητες της μεσογειακή διατροφή καθιστά την εστίαση κυρίαρχο μέρος του τουρισμού στις χώρες τις Μεσογείου. Αλλά και σε παγκόσμιο επίπεδο η μελέτη του συγκεκριμένου τομέα ενδιαφέρει μεγάλο μέρος ιδιοκτητών γι' αυτό και επιλεκτικέ ο συγκεκριμένος τομέας της εστίασης για την μελέτη, ανάπτυξη και εφαρμογή λογισμικού διαχείρισης στην επιχείρηση. Στην Αμερική η χρήση λογισμικών για την διοίκηση και διαχείριση εστιατορίων έχει αρχίσει να εφαρμόζετε σε πολλά εστιατόρια και καταστήματα εστίασης. Επίσης η εστίαση αναφέρεται στην εξυπηρέτηση πολλών ανθρώπων ταυτόχρονα, με αποτέλεσμα να απαιτητέ από την επιχείρηση να απασχολεί εργαζόμενους για την εξυπηρέτηση αυτών. Η απασχόληση πολλών εργαζομένων για την επιχείρηση απαιτεί από αυτήν να έχει σίγουρο κέρδος για να μπορεί να ανταπεξέρχεται και στις υποχρεώσεις της απέναντι στους εργαζομένους που απασχολεί, μισθοδοσία και ασφάλιση αυτών κ.α.

Η εισαγωγή του κατάλληλου λογισμικού στην επιχείρηση προσφέρει εξοικονόμηση χρόνου, καθώς οι πελάτες θα μπορούν να ολοκληρώσουν μόνοι τους την παραγγελία αλλά και εξοικονόμηση χρήματος με την απασχόληση λιγότερων εργαζομένων για την εξυπηρέτηση των πελατών, από την άλλη χρειάζεται να εκπαιδεύσει τους συγκεκριμένους εργαζόμενους στην χρήση του λογισμικού, αλλά και να συνεργαστεί με γνώστες του συγκεκριμένου λογισμικού για την συντήρηση και συνεχής ενημέρωση αυτού. Επομένως από την μία μειώνει το προσωπικό της σε ότι αφορά το service ενώ παράλληλα απαιτείται η συνεργασία με ειδικευμένους για την υποστήριξη του λογισμικού αλλά και τη διοργάνωση σεμιναρίου για την εκπαίδευση του προσωπικού. Η σωστή οικονομική μελέτη στο σύνολό της όμως αποφέρει κέρδη στην επιχείρηση αφού μελλοντικά οι πελάτες θα φέρουν νέους πελάτες, επίσης η επιχείρηση επιχειρεί να επιτύχει τους στόχους της όπως, η ενίσχυση της ανταγωνιστικότητας στον κλάδο της, η αύξηση των κερδών της και παράλληλα η μείωση των λειτουργικών εξόδων, η αύξηση του πελατολογίου της, ο συμβατισμός με τη ζήτηση της αγοράς εν γένει. Η αύξηση κατά το βέλτιστων της ταχύτητας του χρόνου προετοιμασίας της παραγγελίας, ο μηδενισμός των παραβλέψεων που γίνονται στις παραγγελίες, η καταγραφή, η αποθήκευση και η παρακολούθηση της παραγγελίας, η παρακολούθηση του χρόνου απόκρισης και προετοιμασίας των παραγγελιών, η κάλυψη των αναγκών όλων των εθνικοτήτων και ηλικιών των πελατών της και η προσφορά ψυχαγωγίας και διασκέδασης στον τρόπο παραγγελίας. Όπως αναφέρει και ο Michael A. Cusumano (βλέπε σελ. 7) όμως ας μην ξεχνάμε ότι, η επιχειρηματικότητα και όχι η τεχνολογία είναι αυτή η οποία καθορίζει την επιτυχία ή την αποτυχία σε μία επιχείρηση.

Η εξυπηρέτηση του πελάτη είναι καίριο σημείο για την επιβίωση της επιχείρησης, και πλήθος συγκραμάτων και βιβλίων αναφέρουν πόσο σημαντικό είναι ο πελάτης να είναι ευχαριστημένος από την επιχείρηση είτε εστίασης είτε άλλου είδους επιχείρηση. Η ποιότητα του φαγητού αλλά και η ποιότητα εξυπηρέτησης θα αφήσουν καλή ή κακή εντύπωση στον πελάτη και σταδιακά θα κατατάξουν το εστιατόριο ως αντάξιο ή μη των προσδοκιών του πελάτη. Η στρατηγική προϊόντος και υπηρεσίας οι πρακτικές και τα συστήματα που θα χρησιμοποιήσει η επιχείρηση για την ευθυγράμμιση του προϊόντος με την στρατηγική της επιχείρησης, της δίνει την δυνατότητα να επιτύχει υψηλότερα επίπεδα ανάπτυξης παραγωγικότητας. Η διαχείριση του εστιατορίου απασχολεί πολλούς ιδιοκτήτες εστιατορίων. Η χρήση ενός λογισμικού διαχείρισης μπορεί να

δώσει την λύση της άμεσης και αποτελεσματικότερης εξυπηρέτησης αλλά και της αποτελεσματικότερης διαχείρισης. Η διαχείριση ενός εστιατορίου δεν εξαρτάτε μόνο από το λογισμικό διαχείρισης αλλά και από τον τρόπο που θα κινηθεί η επιχείρηση εστιάζοντας στους στόχους της, αυτός είναι που θα καθορίσει την επιτυχία ή την αποτυχία.

Σε αυτό το σημείο θα ήθελα να αναφέρω πώς το λογισμικό σύστημα που παρουσιάζετε παρακάτω, σε καμία περίπτωση δεν μπορεί να υποστηρίξει την θεωρία του Κάρλ Μάρξ<sup>3</sup>, καθώς να μην μπορεί να αποτελέσει ένα εργαλείο για την τεχνολογική εξέλιξη μίας επιχείρησης και ίσως η επιχείρηση μειώσει προσωπικό στον τομέα της εξυπηρέτησης αλλά εξισορροπεί την κατάσταση αφού θα προσλάβει και θα συνεργαστεί με ειδικευμένα άτομα πάνω στον τομέα της επιστήμης της πληροφορικής. Έτσι λοιπόν η αγοραστική δύναμη των εργαζομένων εξισορροπείτε αυτόματος. (Jostein Gaarder, 1994)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

---

<sup>3</sup> Ο Κάρλ Μάρξ ήταν Φιλόσοφος, Ιστορικός, Κοινωνιολόγος και Οικονομολόγος, έζησε τον 18 αιώνα και με την θεωρία του επηρέασε περισσότερο από όλους την πρακτική πολιτική. Έλεγε πως το προϊόν της δουλειάς όταν δεν ανήκει σε αυτόν που το παράγει, τον εργάτη ο ίδιος αποξενώνετε από την δουλειά του και από τον ίδιο του τον εαυτό και χάνει την αξιοπρέπειά του ως άνθρωπος. Αυτό το ονόμαζε αλλοτρίωση.

## ΚΕΦΑΛΑΙΟ 3ο

### Ποιος είναι ο ρόλος του λογισμικού στη διαχείριση και οργάνωση του εστιατορίου;

#### Περίληψη

Στο δεύτερο κεφάλαιο επικεντρωθήκαμε στο ερώτημα γιατί η διαχείριση ενός εστιατορίου αποτελεί εν μέρει πρόβλημα. Πιο αναλυτικά έγινε μία βιβλιογραφική επισκόπηση πάνω στην οποία βασίζετε η ανάπτυξη της εργασίας και μέσα από την ανάλυσή της γίνεται η διεξαγωγή αποτελεσμάτων. Αναφέρεται ότι η βιβλιογραφία για το λογισμικό στο χώρο της εστίασης είναι περιορισμένη, επίσης ο συγκεκριμένος χώρος ενδιαφέρει τομείς όπως ο τουρισμός, ένας τομέας σημαντικός για χώρες που βασίζονται στον τουρισμό. Ως συνέπια του παραπάνω, το λογισμικό στο χώρο της εστίασης αποτελεί και ανοικτή θεματική περιοχή για μελλοντική έρευνα. Τέλος αναφέρονται ως συμπεράσματα τα υπέρ και τα κατά, η επιρροή του τομέα της εστίασης στον τουρισμό και τονίζουμε ότι η διαχείριση ενός εστιατορίου δεν εξαρτάτε μόνο από το λογισμικό διαχείρισης αλλά και από τον τρόπο που θα κινηθεί η επιχείρηση εστιάζοντας στους στόχους της, αυτό είναι και το κλειδί της επιτυχίας ή της αποτυχίας της επιχείρησης. Σε αυτό το κεφάλαιο αναφέρουμε τους σκοπούς και τους στόχους μιας επιχείρησης εστίασης, περιγράφουμε την πρόταση για την λύση η ακόμα καλύτερα την πρόταση για την βελτιστοποίηση της διοίκησης και διαχείρισης μίας επιχείρησης εστίασης, αναφέρουμε τρεις προτάσεις για την επιλογή υλικού φιλοξενίας του λογισμικού και τέλος αναφέρουμε τα συμπεράσματα μέσα από τα οποία τονίζουμε καθαρά πως το λογισμικό καλύπτει εξ ολοκλήρου όλους τους στόχους της επιχείρησης εστίασης.

#### Σκοπός και αντικειμενικοί στόχοι της επιχείρησης εστίασης

Σκοπός της επιχείρησης από την χρήση της λογισμικής εφαρμογής είναι η επίτευξη των στόχων, δηλαδή των επιθυμητών αποτελεσμάτων τα οποία η επιχείρηση προσπαθεί να πετύχει με μια σειρά προγραμματισμένων ενεργειών και δράσεων μέσα σε ένα ορισμένο χρονικό διάστημα. Οι στόχοι αυτοί για το εστιατόριο είναι να γίνει πιο ανταγωνιστικό. Η εισαγωγή της τεχνολογίας του λογισμικού σημαίνει κατά μια έννοια μηδενισμό των αβλεψιών που μπορεί να γίνονταν στις παραγγελίες λόγω του ανθρώπινου παράγοντα.

Στόχοι της επιχείρησης επίσης είναι η επίτευξη τόσο της δυνατόν μεγαλύτερης ταχύτητας στον χρόνο προετοιμασίας της παραγγελίας, η καταγραφή, αποθήκευση και παρακολούθηση αυτής όσο και η διαχείριση των χρόνων ετοιμασίας και υλοποίησης. Η αύξηση των κερδών της επιχείρησης όμως είναι ο βασικός στόχος. Πολλαπλασιασμό των κερδών μια επιχείρηση μπορεί να αποκτήσει με την αύξηση του πελατολογίου της, την προοπτική της επεκτατικής πολιτικής δημιουργώντας αλυσίδα καταστημάτων τόσο σε τοπικό επίπεδο όσο και σε διεθνές επίπεδο. Για παράδειγμα η κολοσσιαία επιχείρηση των MacDonald's έχει κυριαρχήσει στο χώρο των fast foods.

Με την προσθήκη ενός τέτοιου καινοτόμου λογισμικού όφελος εκτός από την επιχείρηση έχουν και οι πελάτες οι οποίοι απολαμβάνουν την άμεση εξυπηρέτηση και την ψυχαγωγία καθώς ο τρόπος παραγγελίας

είναι μια διασκεδαστική και ευχάριστη διαδικασία. Βασίζετε σε οθόνη αφής με την οποία οι περισσότεροι χρήστες είναι εξοικειωμένοι, λόγω χρήσης τεχνολογικών συσκευών όπως τα έξυπνα κινητά (smartphones) με οθόνες αφής, τα αυτόματα μηχανήματα ανάληψης χρημάτων ή και αγοράς προϊόντων, τα ηλεκτρονικά βιβλία και πλήθος άλλων συσκευών. Η ψυχαγωγία και η διασκέδαση ακόμα και κατά την διαδικασία της παραγγελίας αποφέρει στην επιχείρηση αναγνωσιμότητα και φυσικά κέρδη.

Οι πελάτες ενός εστιατορίου μπορεί να είναι διαφόρων ηλικιών, οικογένειες, παρέες ενηλίκων αλλά και παρέες παιδιών, στόχος της επιχείρησης είναι να φροντίσει να καλύπτει τις ανάγκες όλων των κατηγοριών, μικρών και μεγάλων, καθώς και πελάτες που δεν είναι της ίδιας χώρας με την χώρα που εδρεύει η επιχείρηση. Για παράδειγμα μπορεί για τους νεότερους σε ηλικία η παραγγελία μέσω της εφαρμογής να είναι εύκολη και ευχάριστη το εστιατόριο θα πρέπει να φροντίσει το ίδιο εύχρηστη και οικεία να είναι και για τους μεγαλύτερους. Όσον αφορά για πελάτες άλλης χώρας το εστιατόριο θα πρέπει να έχει στόχο την εξυπηρέτηση τους με την ύπαρξη βασικών γλωσσών στο λογισμικό. Η επανάσταση της ψηφιακής εικόνας και η εξέλιξη της επικοινωνίας ανθρώπου μηχανής και οι διαδραστικές εικόνες έχουν αλλάξει τις συνήθειες διαβίωσης των ανθρώπων, οι αλλαγές αυτές αλλάζουν την ζήτηση και την προσφορά, στόχος του εστιατορίου είναι να ανταποκρίνεται πλήρως στην ζήτηση της αγοράς. Ο τρόπος που έχει σχεδιαστεί και βασιστεί η εφαρμογή είναι απλός και γνωστός στους περισσότερους καθώς βασίζετε στο γνωστό παιχνίδι «Monopoly»<sup>4</sup> και συμβαδίζει πάντα με την εξέλιξη της τεχνολογίας.

### **Σκοπός**

Η επιχείρηση σκοπεύει να εισάγει και να λειτουργήσει ένα λογισμικό για την βελτίωση των υπηρεσιών της. Μέσο του λογισμικού αυτού επιχειρεί να καλύψει όλες τις ανάγκες της, να αντιμετωπίσει τα τυχόν προβλήματα, να προβλέψει και να φροντίσει για την πρόληψη των μελλοντικών πιθανόν προβλημάτων, να συνάδουν οι ενέργειές της ώστε να καλύψουν τις μελλοντικές ανάγκες της αγοράς και τελικά να επιτύχει τους στόχους της βραχυχρόνια και μακροχρόνια διαμορφώνοντας το δικό της νέο, δυναμικό και φιλικό πρόσωπο βασισμένο στην ποιότητα και την αμεσότητα των υπηρεσιών της προς το εξωτερικό περιβάλλον και τους πελάτες της.

### **Στόχοι**

Στόχοι της επιχείρησης αποτελούν:

1. Η ενίσχυση της ανταγωνιστικότητας στον κλάδο της.
2. Η αύξηση των κερδών της και παράλληλα η μείωση των λειτουργικών εξόδων.
3. Η αύξηση του πελατολογίου της.
4. Ο συμβατισμός με τη ζήτηση της αγοράς εν γένει.
5. Η αύξηση κατά το βέλτιστων της ταχύτητας του χρόνου προετοιμασίας της παραγγελίας.
6. Ο μηδενισμός των παραβλέψεων που γίνονται στις παραγγελίες.
7. Η καταγραφή, η αποθήκευση και η παρακολούθηση της παραγγελίας.
8. Η παρακολούθηση του χρόνου απόκρισης και προετοιμασίας των παραγγελιών.
9. Η κάλυψη των αναγκών όλων των εθνικοτήτων και ηλικιών των πελατών της.
10. Η προσφορά ψυχαγωγίας και διασκέδασης στον τρόπο παραγγελίας.

---

<sup>4</sup> Monopoly: ονομάζεται το επιτραπέζιο παιχνίδι αγοραπωλησίας ιδιοκτησιών που εκδίδεται από την εταιρεία Parker σε συνεργασία με την Hasbro.

## Περιγραφή Πρότασης

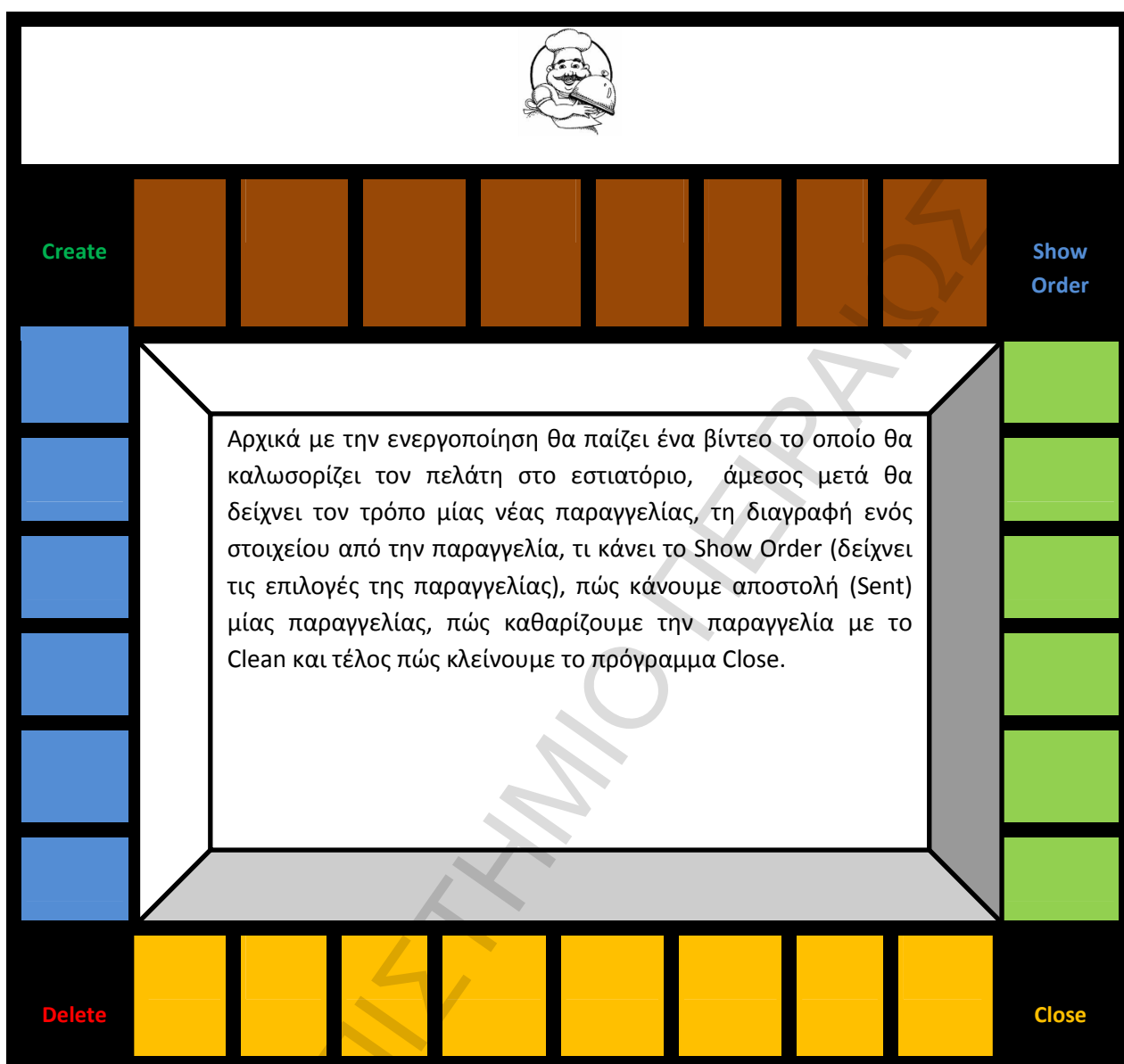
Η πρόταση ως λύση του προβλήματος η ακόμα καλύτερα για την βελτίωση της διαχείρισης και της οργάνωσης ενός εστιατορίου είναι και το σύστημα επιτραπέζια παραγγελίας. Το Σύστημα επιτραπέζια παραγγελίας, είναι ένα σύστημα το οποίο έχει σαν στόχο την άμεση παραγγελία, από τον πελάτη στην κουζίνα. Οι περισσότεροι γνωρίζουν το επιτραπέζιο παιχνίδι μονοπολι και τον τρόπο, με τον οποίο απαιτεί να κάθονται γύρο από ένα τραπέζι οι παίχτες που θα συμμετέχουν στο παιχνίδι. Η παρακάτω εικόνα ίσως είναι γνωστή σε όλους



Εικόνα 1 Παιχνίδι μονοπολι

Είναι μια εικόνα από το παιχνίδι αυτό, πάνω στο συγκεκριμένο στήσιμο βασίζεται και ο τρόπος σχεδίασης της επιτραπέζια παραγγελίας, παρακάτω ακολουθεί η εικόνα από το interface παραγγελίας στο τραπέζι όπου με καφέ χρώμα αναπαριστώνται οι θέσεις για τα πιάτα του κυρίως μενού, με πράσινο οι θέσεις για τις σαλάτες, με πορτοκαλή οι θέσεις για το επιδόρπιο και με μπλε οι θέσεις για τα αναψυκτικά και τα ποτά.





Εικόνα 2 Γραφικό περιβάλλον παραγγελίας

Ας φανταστούμε λοιπόν το τραπέζι της παραγγελίας σαν ένα μεγάλο τραπέζι αφής όπου γύρο γύρο θα υπάρχουν οι 4 μπάρες πάνω στις οποίες θα φιλοξενούνται στοιχεία (elements), ποιο συγκεκριμένα εικόνες που θα αντιπροσωπεύουν το μενού παραγγελίας ανάλογα με το είδος του καταστήματος και το αντίστοιχο μενού του. Η συγκεκριμένη λογισμική εφαρμογή έχει τη δυνατότητα να εφαρμόζετε σε πληθώρα καταστημάτων εστίασης και αυτό είναι ένα από τα βασικά πλεονεκτήματα της εφαρμογής.

#### Παράδειγμα:

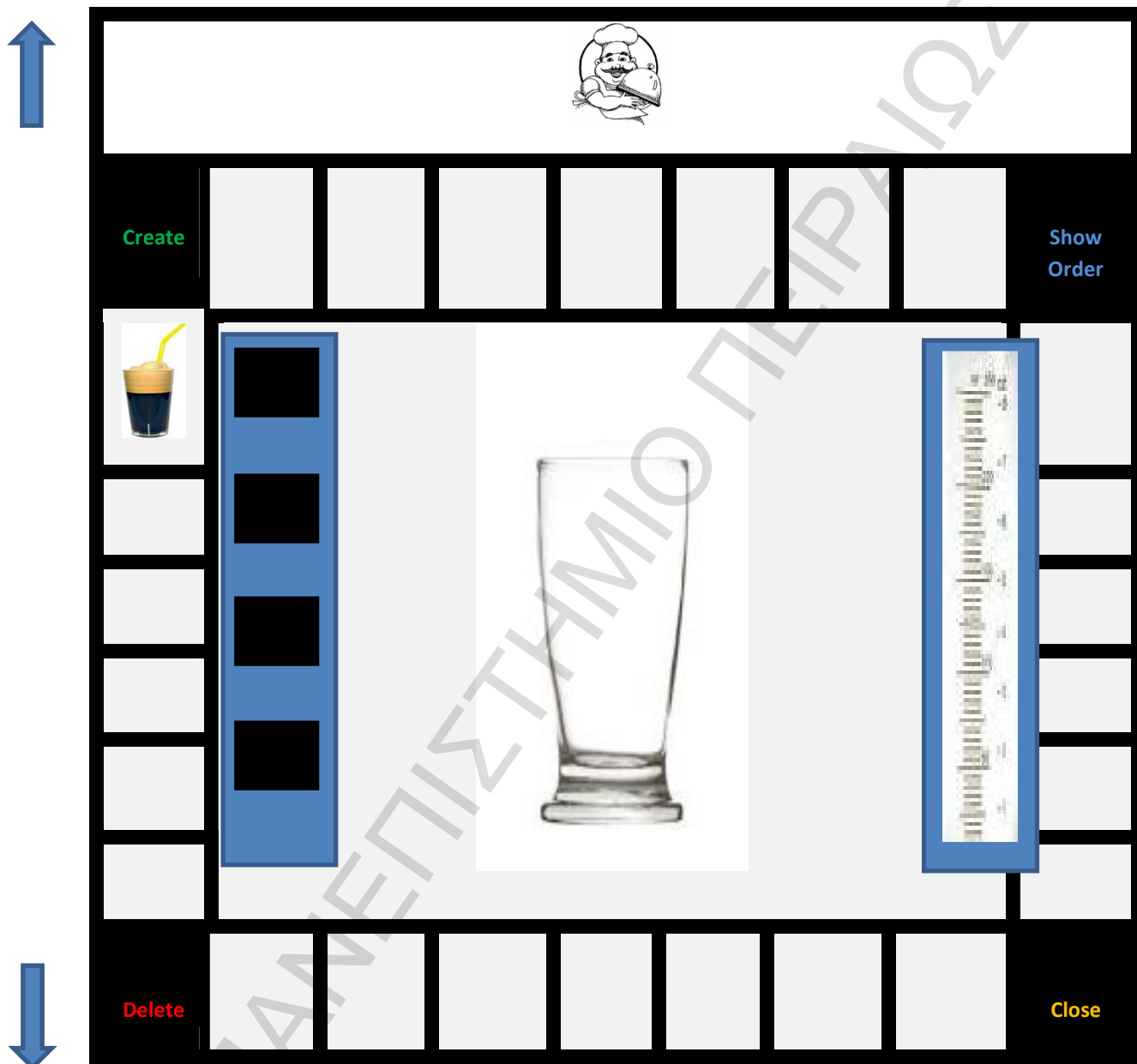
Ο παρακάτω πίνακας υποθετικά περιλαμβάνει τα είδη ενός εστιατορίου αλλά και τα καταστήματα τα οποία θα μπορούσαν να κάνουν χρήση του προγράμματος αλλάζοντας τα στοιχεία ανάλογα με το μενού του καθενός, το οποίο θα αναπαριστάται με τις αντίστοιχες εικόνες γύρο από το τραπέζι αφής ή τον κατάλογο αφής ή την εύκαμπτη οθόνη αφής. Η ευχρηστία του λογισμικού είναι στον τρόπο με τον οποίο μπορεί ο

διαχειριστής αυτού να προσθέσει και να αφαιρέσει στοιχεία και να προσαρμόσει το λογισμικό στο δικό του μενού (αναλυτικότερα παρακάτω).

Είδος Καταστήματος	Είδη
Εστιατόριο	<ul style="list-style-type: none"> <li>• Κυρίως πιάτο Α</li> <li>• Κυρίως πιάτο Β</li> <li>• Κυρίως πιάτο Γ</li> <li>• Κυρίως πιάτο Δ</li> <li>• Κυρίως πιάτο Ε</li> <li>• Κυρίως πιάτο Ζ</li> <li>• Σαλάτα Α</li> <li>• Σαλάτα Β</li> <li>• Σαλάτα Γ</li> <li>• Σαλάτα Δ</li> <li>• Σαλάτα Ε</li> <li>• Σαλάτα Ζ</li> <li>• Αναψυκτικό Α</li> <li>• Αναψυκτικό Β</li> <li>• Αναψυκτικό Γ</li> <li>• Αναψυκτικό Δ</li> <li>• Αναψυκτικό Ε</li> <li>• Αναψυκτικό Ζ</li> <li>• Επιδόρπιο – Γλυκό Α</li> <li>• Επιδόρπιο – Γλυκό Β</li> <li>• Επιδόρπιο – Γλυκό Γ</li> <li>• Επιδόρπιο – Γλυκό Δ</li> <li>• Επιδόρπιο – Γλυκό Ε</li> <li>• Επιδόρπιο – Γλυκό Ζ</li> </ul>
Μαγαζί πώλησης σάντουιτς	Θα γίνεται η επιλογή των υλικών που θα επιθυμούμε να έχει το σάντουιτς.
Μαγαζί πώλησης Burger	Επιλογή των έτοιμων προϊόντων στυλ burger, επιπλέον πατάτες, σαλάτες, αναψυκτικά και γλυκά ανάλογα με το μενού.
Μαγαζί πώλησης παγωτού	Αντίστοιχα θα επιλέγονται οι γεύσεις και τα υπόλοιπα υλικά για το παγωτό, σιρόπι, μπισκότα, τρούφα κ.τ.λ
Καφετέρια	Αντίστοιχα ανάλογα με το μενού θα γίνεται παραγγελία καφέ, αναψυκτικών, τοστ κ.τ.λ (προτεινόμενο γραφικό περιβάλλον παρακάτω).
Κινέζικο εστιατόριο	Αντίστοιχη παραγγελία όπως στο εστιατόριο ανάλογα με το μενού του εστιατορίου.
Μαγαζί με Κρέπες	Θα γίνεται η επιλογή των υλικών που θα επιθυμούμε να έχει η κρέπα.

Για τη υλοποίηση του προγράμματος η εφαρμογή θα αναπτυχθεί με βάση το μενού μη υπαρκτού gourmet εστιατορίου. Αξίζει όμως να παρουσιάσουμε και το interface της παραγγελίας καφέ.

### Γραφικό Περιβάλλον (Interface) Πελάτη Καφετέριας

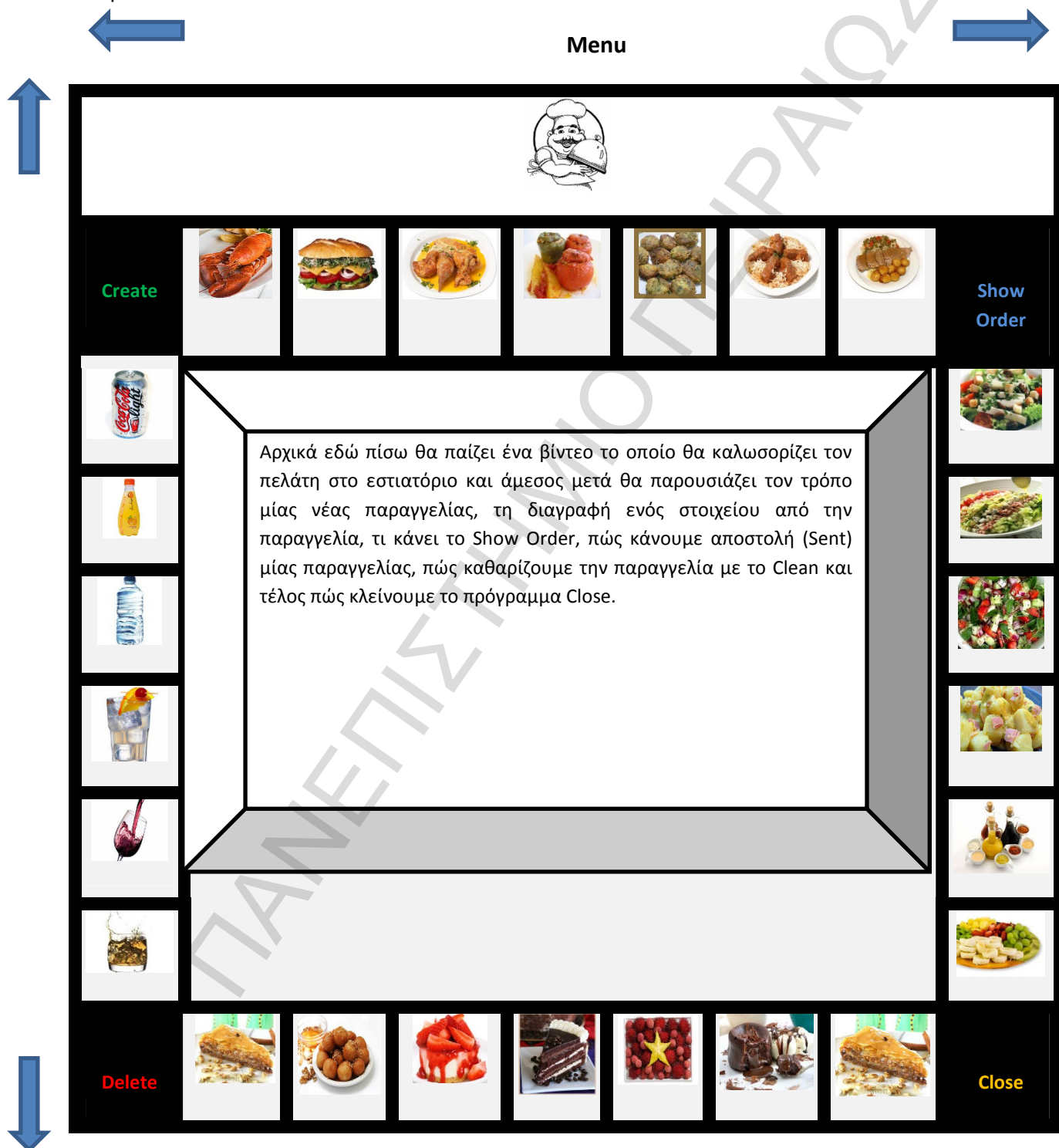


Εικόνα 3 Γραφικό περιβάλλον καφετέριας



## Γραφικό Περιβάλλον (Interface) Πελάτη Εστιατορίου

Για ένα εστιατόριο το οποίο θα περιέχει μενού για τα φαγητά, τις σαλάτες, τα ποτά – αναψυκτικά και τα γλυκά η κατηγοριοποίηση θα μπορούσε να γίνει με τον εξής τρόπο. Στην επάνω μπάρα θα μπορούσαν να τοποθετούνται τα φαγητά. Στην δεξιά μπάρα θα τοποθετούνταν τα συμπληρωματικά όπως σαλάτες, σάλτσες κ.τ.λ. Στην αριστερή θα τοποθετούνται τα αναψυκτικά και τέλος στην κάτω μπάρα θα τοποθετούνται τα επιδόρπια.

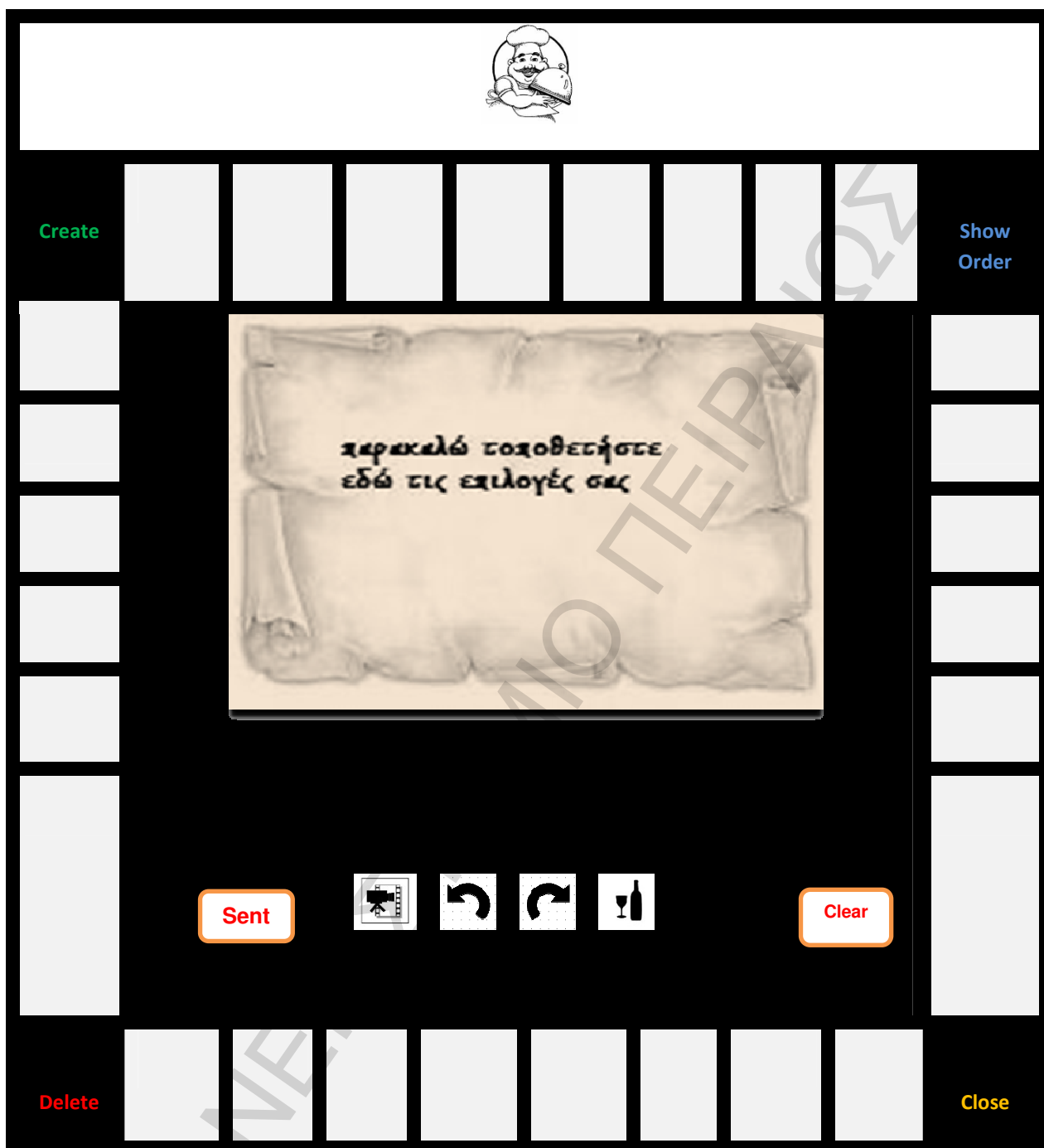


Εικόνα 4 Γραφικό περιβάλλον παραγγελίας

Όλες οι μπάρες έχουν την ιδιότητα να μετακινούνται με άγγιγμα δεξιά και αριστερά για τις οριζόντιες και πάνω κάτω για τις κάθετες, προκειμένου να βλέπει ο πελάτης το πλήρες μενού του εστιατορίου. Όταν κάποιος ανοίγει την εφαρμογή για πρώτη φορά το λογότυπο του εστιατορίου θα εμφανίζεται και θα καλωσορίζει τον πελάτη, στη συνέχεια ένα βίντεο θα παίζει καθ' όλη την διάρκεια έως ότου ο πελάτης πατήσει το κουμπί **βίντεο** και θα δείχνει στο χρήστη, τον τρόπο με τον οποίο μπορεί να κάνει παραγγελία, καθώς και την χρήση των τεσσάρων κουμπιών, τα οποία αναφέρονται παρακάτω αναλυτικότερα.

Στης τέσσερις γωνίες όπως ακριβώς και στη μοπορολί, υπάρχουν βασικά κουμπιά για την ολοκλήρωση παραγγελίας από τον πελάτη χρήστη. Στην πάνω αριστερή γωνία το κουμπί **Create**, είναι υπεύθυνο για την δημιουργία της παραγγελίας, με το πάτημά το βίντεο σταματάει να παίζει και στο φόντο εμφανίζεται ένα χαρτί πάνω στο οποίο με την μέθοδο πάρε σύρε, θα μπορεί ο πελάτης να σύρει τα αντικείμενα στο χαρτί για να διαμορφώσει την παραγγελία του. τρία βασικά κουμπιά θα εμφανίζονται μαζί με τον δίσκο στην κάτω μεριά, το ένα θα είναι το undo για την αναίρεση κάποιας ενέργειας, το άλλο θα είναι το redo για την επιστροφή της ενέργειας και το κουμπί buck το video το οποίο θα επιστρέφει την εφαρμογή πίσω στο αρχικό βίντεο.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΣ



Εικόνα 5 Γραφικό περιβάλλον παραγγελίας

Το κουμπί **Delete** θα είναι αυτό με το οποίο αφού θα έχουμε επιλέξει κάποιο στοιχείο από το δίσκο, θα το διαγράψουμε πατώντας το.

Το κουμπί **Show Order** είναι αυτό με το οποία οποιαδήποτε στιγμή θα μπορεί ο πελάτης να βλέπει τα αναλυτικά στοιχεία της παραγγελίας σε ένα πίνακα που θα εμφανίζεται αλλά και τις τιμές αναλυτικά και συνολικά με και χωρίς φπα. Επίσης με το άγγιγμα κάθε στοιχείου component θα εμφανίζεται ένα popup menu με την περιγραφή και την τιμή του πιάτο.

Παράδειγμα κουμπιού show order

Προϊόν	Τιμή
	15€
	
<b>Συνολική Αξία</b>	15€
<b>Αξία με φπα</b>	15,50€

ενεργό



Το κουμπί **Close** είναι αυτό με το οποίο μπορούμε να κλείσουμε το πρόγραμμα παραγγελίας αφού έχουμε ολοκληρώσει την παραγγελία μας.

Με το πάτημα του κουμπιού **Sent** θα εμφανίζεται ένα μήνυμα το οποίο θα περιγράφει την παραγγελία (το ίδιο μήνυμα με το κουμπί **show Order**) που έχει επιλέξει ο πελάτης, προκειμένου να είναι σίγουρος για το τι έχει επιλέξει. Και με την επιβεβαίωση αυτού η παραγγελία θα αποστέλλεται στην κουζίνα και θα εμφανίζεται το μήνυμα.

Sent

Προϊόν Τιμή



15€



ενεργό



**Συνολική  
Αξία**

15€

**Αξία με  
φπα**

15,50€

Επιβεβαιώστε για την  
ολοκλήρωση της  
αποστολής

Η παραγγελία σας έχει  
σταλεί στην κουζίνα του  
καταστήματός μας, σύντομα  
θα σερβιριστείτε.



Όταν το χέρι βρίσκεται πάνω από τον πάπυρο θα έχει την μορφή πέννας, γράφεις την παραγγελία.



πού θα σημαίνει ότι

Το κουμπί  
το χαρτί .



θα σβήνει τα στοιχεία από την παραγγελία και θα καθαρίζει τελείως



Το γνωστό undo αναιρεί μία ενεργεί.



Το γνωστό redo














Ένα από τα πιο βασικά κουμπιά, με την επιλογή του θα ενημερώνεται η κουζίνα ότι ο πελάτης επιθυμεί τα αναψυκτικά που έχει στην παραγγελία να έρθουν αμέσως, πριν από το κυρίως πιάτο.




Με την επιλογή αυτού του κουμπιού θα επιστρέφει πίσω στο αρχικό video της εφαρμογής, το οποίο περιγράφει την διαδικασία μίας παραγγελίας.

## Γραφικό Περιβάλλον Interface Κουζίνας – manager

Τα στοιχεία της παραγγελίας θα εμφανίζονται στην κουζίνα σε ένα πίνακα όπου και θα μπορεί να τα βλέπει και να ξεκινήσει την ετοιμασία της παραγγελίας.

Διάγραμμα πίτας για την κίνηση του μαγαζιού		Χάρτης για τα ενεργά τραπέζια ή όχι		
Αριθμός Τραπεζιού	Παραγγελία	κατάσταση	Αποδοχή Παραγγελίας	Διάρκεια Παραγγελίας
 T3				
 T2				
 T1				Δεν έχει ξεκινήσει χρονομέτρηση
 T1	Η παραγγελία στάλθηκε Αναμένετε απόκριση από τον διαχειριστή.	 	Πίεσε το κουμπί για την αποδοχή της παραγγελίας	Η χρονομέτρηση ξεκίνησε  Χρόνος απόκρισης (αναμονής)
	Αστακομακαρονάδα <input type="checkbox"/> Χοιρινή <input type="checkbox"/> Κρασί λευκό 1950! <input type="checkbox"/>			Χρόνος Ετοιμασίας Παραγγελίας

	Νερό μεγάλο <input type="checkbox"/>			
			<input checked="" type="checkbox"/>	Συνολικός χρόνος
	Έχουν εκτελεστεί			
	Έχουν εκτελεστεί			
	Έχουν εκτελεστεί			

Το γραφικό περιβάλλον το συντονιστή (manager) της κουζίνας περιέχει στην αρχική του κατάσταση όλα τα τραπέζια που είναι διαθέσιμα και έχουν χρώμα λάμπας λευκό. Αυτό είναι και το στάδιο 1.



Τα τραπέζια που είναι διαθέσιμα.

Στην πορεία από την στιγμή που ο πελάτης πατήσει οριστικά το sent, το χρώμα της σήμανσης γίνεται πράσινο που σημαίνει ότι η παραγγελία στάλθηκε στην κουζίνα και πιο συγκεκριμένα έχει ξεκινήσει ο χρόνος απόκρισης της παραγγελίας, ο οποίος σταματά με την απόκριση από τον διαχειριστή manager, όπου και περνάμε σε μία άλλα κατάσταση και η σήμανση θα γίνει πορτοκαλή στάδιο 2 και 3. Ο Διαχειριστής - manager καλείτε να πατήσει αποδοχή ανάλογα με την κίνηση του εστιατορίου.



Η σήμανση από τα τραπέζια που οι πελάτες έχουν στείλει την παραγγελία και έχει ξεκινήσει να μετράει ο χρόνος απόκρισης από την κουζίνα.



Στο στάδιο 3 είναι οι παραγγελίες για τις οποίες έχει ξεκινήσει η ετοιμασία, και μετρά ο χρόνος ετοιμασίας της παραγγελίας. Η ομάδα της κουζίνας έχει ξεκινήσει την κούρσα, με στόχο το άψογο service, πάντα σε επικοινωνία με τον manager για την ενημέρωση των φαγητών που ολοκληρώνετε η ετοιμασία τους. Σε αυτό το στάδιο αφού όλες οι παραγγελίες είναι τσεκαρισμένες μια μια, στο κουτάκι δίπλα από το όνομα της κάθε μίας, εμφανίζεται το μήνυμα ότι όλες οι παραγγελίες είναι ok να ολοκληρωθεί το στάδιο της ετοιμασίας. Και έτσι περνάει αυτόματα στο στάδιο 4.



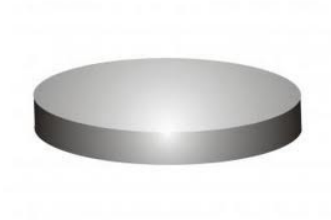


Στο στάδιο 4 είναι οι παραγγελίες οι οποίες έχουν ολοκληρωθεί, μία παραγγελία που έχει ολοκληρωθεί μπορεί να ενεργοποιηθεί ξανά. Κάθε παραγγελία έχει δικό της ID το οποίο θα είναι αυτο increment στη βάση δεδομένων που θα υποστηρίζει την εφαρμογή.

Στο στάδιο 1 θα γνωρίζει η κουζίνα το τραπέζι στο οποίο αντιστοιχεί η παραγγελία.

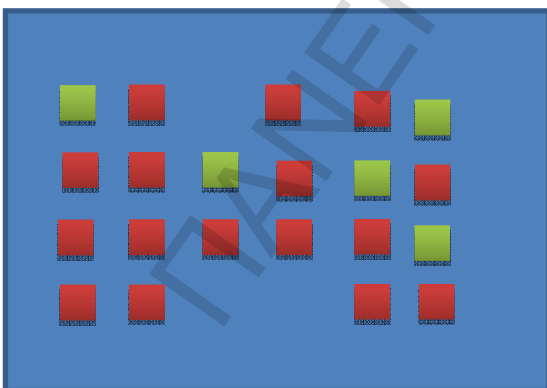
### Διάγραμμα πίτας για την κίνηση του μαγαζιού

Ακόμα ο πίνακας περιλαμβάνει στατιστικά στοιχεία, όπως μία πίτα για την κίνηση του μαγαζιού κάθε στιγμή, η οποία θα απεικονίζεται στην πάνω αριστερή γωνία.

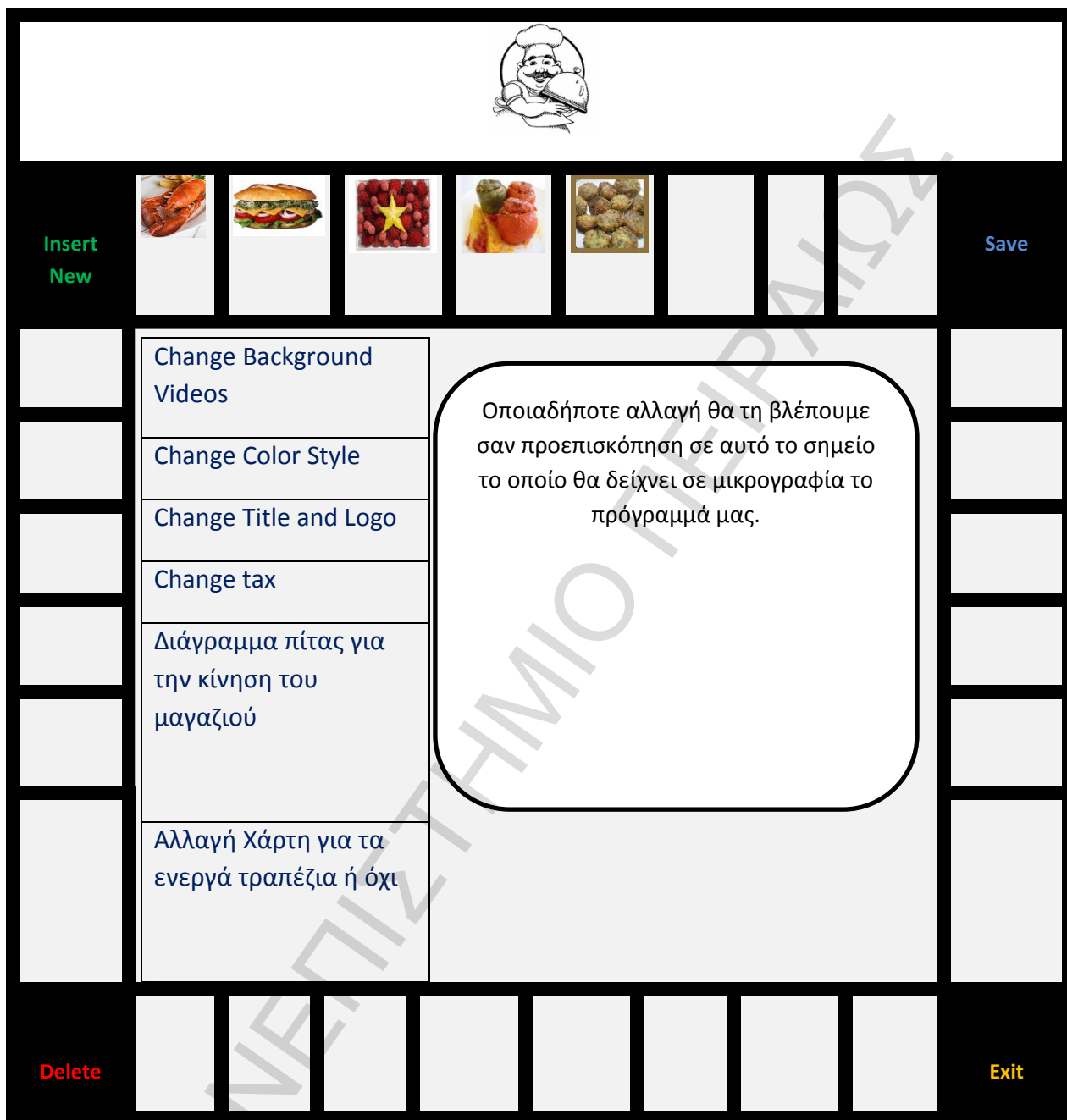


### Χάρτης για τα ενεργά τραπέζια ή όχι

Ένα ακόμα στατιστικό στοιχείο για την κατάσταση του εστιατορίου κάθε στιγμή είναι και ένας χάρτης, ο οποίος απεικονίζει τα ενεργά τραπέζια στο μαγαζί με πράσινο χρώμα και τα μη ενεργά με κόκκινο χρώμα. Ο χάρτης αυτός θα βρίσκεται στην πάνω δεξιά γωνία του πίνακα.



## Γραφικό Περιβάλλον Διαχειριστή ενημέρωση καταλόγου



Εικόνα 6 Γραφικό περιβάλλον διαχειριστή

Ο διαχειριστής έχει πρόσβαση από οπουδήποτε μέσω διαδικτύου.

Με την εντολή **Insert New** ανοίγει κάποιο παράθυρο στο οποίο επιλέγεις τα χαρακτηριστικά του κάθε πιάτου. Όπως τη φωτογραφία του, την ονομασία του, την τιμή του, την περιγραφή του και φυσικά σε ποιον επιθυμούμε να ενταχθεί. Φαγητά, Ποτά, Σαλάτες ή Γλυκά.

Η εντολή **Delete** είναι πιο απλή καθώς είναι χρήσιμη για την διαγραφή κάποιου πιάτου, απλά επιλέγοντας το πιάτο και το διαγράφει με το πάτημα του **Delete**.

**Exit** για την έξοδο από το πρόγραμμα.

**Save** για την αποθήκευση των αλλαγών.

## Hardware υλοποίησης, προτάσεις φιλοξενίας της εφαρμογής

### Οπτικοί ανιχνευτές (Optical Scanners)

Η βασική αρχή λειτουργίας τους βασίζεται στην εκπομπή κάποιας πηγής φωτός και στην ανίχνευση του μετά από την αντανάκλαση στην επιφάνεια που περιέχει τα προς καταχώρηση δεδομένα. Ανάλογα με τη μορφή των δεδομένων, οι οπτικοί ανιχνευτές διακρίνονται ως εξής:

- Οθόνες αφής (Touch Screens)
- Σαρωτές (Scanners)
- Οπτικοί αναγνώστες χαρακτήρων (Optical Character Reader - OCR)
- Οπτικοί ανιχνευτές στιγμάτων (Optical Mark Scanners - OMR)
- Αναγνώστες γραμμωτού κώδικα (Bar Code Readers)

Για την εργασία μας χρειάζεται να γνωρίζουμε για τις οθόνες αφής.

### Οθόνες αφής (Touch Screens)

Είναι οθόνες στο γυαλί των οποίων υπάρχει μία διαφανής αισθητήρια μεμβράνη, η οποία αναγνωρίζει το σημείο της οθόνης που έρχεται σε επαφή με ένα άλλο σώμα (συνήθως το δάκτυλο του χρήστη). Χρησιμοποιούνται για την καταχώρηση περιορισμένης ποσότητας δεδομένων, συνήθως με τη μέθοδο της επιλογής από ένα μενού επιλογών, με το άγγιγμα με το άγγιγμα του δαχτύλου στην συγκεκριμένη θέση. Βρίσκουν εφαρμογή στα μηχανήματα αυτόματης ανάληψης (ATM) τραπεζών, σε κινητά τηλέφωνα (smart phones), σε tablets και πληθώρα άλλες συσκευές καθώς η χρήση τους έχει γίνει αυξηθεί ευρέως. Παρακάτω προτείνονται οι λύσεις για την φιλοξενία του λογισμικού στο εστιατόριο. Αυτές οι λύσεις είναι:

1)Τραπέζι με γυάλινη επιφάνεια όπου θα φιλοξενεί το λογισμικό (πιο ακριβή λύση)

Τα «τραπέζια» αφής είναι σε θέση να αντιδρούν σε «απεριόριστα» ταυτόχρονα ερεθίσματα. Τα συγκεκριμένα τραπέζια έχουν διάφορα μεγέθη και δύο βασικά είναι των 32 και 46 ιντσών, παρέχουν ανάλυση High Definition και πλήθος άλλων ευκολιών, όπως Είσοδος HDMI, SmartPower για εξοικονόμηση ενέργειας, Ανεξαρτησία λογισμικού, Υπέρυθρη αφή, Συμβατότητα με τα πρότυπα RoHS για τη φροντίδα του περιβάλλοντος. Η ευελιξία και η απόδοση συναντούν τη μεγάλη ποικιλία χειρισμού, προσφέροντας την απόλυτη διάδραση με τους χρήστες.

## Είσοδος HDMI

Το HDMI συνδέει την πηγή με την οθόνη, καθιστώντας δυνατή τη μεταφορά ψηφιακών ασυμπίεστων σημάτων RGB. Ελαχιστοποιώντας τη μετατροπή σε αναλογικά σήματα, παρέχει άψογη εικόνα, και επειδή το σήμα δεν μειώνεται η εικόνα είναι πιο καθαρή. Το HDMI μεταφέρει έξυπνα την υψηλότερη ανάλυση εξόδου της συσκευής-πηγής. Το HDMI είναι πλήρως συμβατό με προηγούμενες εκδόσεις πηγών DVI, ωστόσο περιλαμβάνει και ψηφιακό ήχο. Το HDMI χρησιμοποιεί προστασία από αντιγραφή HDCP.

## SmartPower για εξοικονόμηση ενέργειας

Το σύστημα μπορεί να ελέγξει και να προρυθμίσει την ένταση του οπίσθιου φωτισμού ώστε να μειώσει την κατανάλωση ρεύματος έως και 50%, εξοικονομώντας σημαντικά στην κατανάλωση ενέργειας.

## Συμβατότητα με τα πρότυπα RoHS για τη φροντίδα του περιβάλλοντος

Η Philips κατασκευάζει προϊόντα απεικόνισης σε συμμόρφωση με τα αυστηρά πρότυπα για τον περιορισμό επικίνδυνων ουσιών όπως ο μόλυβδος και άλλες τοξικές ουσίες σε είδη ηλεκτρικού και ηλεκτρονικού εξοπλισμού (RoHS).

## Υπέρυθρη αφή

Με ένα αόρατο πλέγμα υπερύθρων στο πάνω μέρος της οθόνης, μπορείτε να απολαύσετε εκπληκτική ακρίβεια και γρήγορους χρόνους απόκρισης, ακόμα κι αν αγγίζετε με το δάχτυλο, με γάντι ή γραφίδα. Και τέρμα τα "σημεία φαντάσματα".

## Ανεξαρτησία λογισμικού

Οι οθόνες αφής που σχεδιάζουμε παρουσιάζουν ανεξαρτησία λογισμικού και λειτουργούν με τα περισσότερα λογισμικά για οθόνες αφής που είναι διαθέσιμα στο εμπόριο, μεταξύ των οποίων τα Windows XP/7 και Mac OS, Omnitapps – BMT1909/00. Συνδεθείτε μέσω USB σε επιτραπέζιο ή φορητό υπολογιστή.



Εικόνα 7 Τραπέζι αφής

2) Tablet συσκευή (ipad) σε κάθε τραπέζι του εστιατορίου (πιο οικονομική λύση)



Εικόνα 8 Tablet παραγγελίας

## Μη Λειτουργικές Απαιτήσεις

### Απαιτήσεις χρήσης

Το λογισμικό θα πρέπει να περιέχει φιλικό περιβάλλον προς τον χρήστη με υποστήριξη γραφικών και συνδυασμό χρήσης του χεριού ή πέννας για την χρήση του.

### Απαιτήσεις αξιοπιστίας

Η πρόσβαση από πολλούς χρήστες (ιδιοκτήτες) ταυτόχρονα προστατεύεται με δυνατότητα να χρησιμοποιούν την εφαρμογή με δικό τους κωδικό χρήστη και password.

### Απαιτήσεις επιδόσεων

Θα υπάρχει αξιοποίηση των πόρων του συστήματος, όποιων κι αν είναι αυτοί.. Επιπλέον, η απόκριση από το σύστημα βάσεων δεδομένων δεν θα ξεπερνά τα 2 δευτερόλεπτα.

## **Απαιτήσεις υποστήριξης**

Το λογισμικό θα διατίθεται σε περιβάλλον Windows και θα εγκαθίσταται ιδιαίτερα εύκολα σε οποιοδήποτε μηχάνημα με υποστήριξη και αρχείου README το οποίο θα επεξηγεί και θα διευκρινίζει τα θέματα εγκατάστασης και λειτουργίας της εφαρμογής.

## **Απαιτήσεις υλοποίησης**

Θα χρησιμοποιηθεί η γλώσσα c-sharp και asp.net λόγω της δυνατότητας ανάπτυξης δια-πλατφορμικών εφαρμογών με αντικειμενοστραφή προγραμματισμό σε συνδυασμό με χρήση SQL για υποβολή ερωτημάτων προς τη Βάση Δεδομένων.

## **Απαιτήσεις Βάσεων Δεδομένων**

Θα χρησιμοποιηθεί σύστημα Βάσεων Δεδομένων με τους ακόλουθους πίνακες : Elements, Orders, Categories, Admin αντίστοιχα ο πίνακας με τα στοιχεία έχει τα εξής πεδία: id\_ Elements, id\_ Categories, Name, Description, Image, Price. Ο πίνακας με τις παραγγελίες έχει τα πεδία Id\_ Orders, Id\_ Element, Total\_ Price, Date, Time, Table\_ Name, Checking. Ο πίνακας με τις κατηγορίες έχει τα πεδία Id\_ Categories, Drinks, Salads, Main\_ Food, Swiss. Τέλος ο πίνακας για τα στοιχεία των χρηστών έχει τα πεδία Id\_ admin, Name, Username, Password, Date. (για περισσότερα βλέπε στο κεφάλαιο 7 Σχεδιασμός).

## **Φυσικές απαιτήσεις**

Το λογισμικό αρχικά θα εγκατασταθεί σε έναν server απομακρυσμένο, με ελάχιστη απαίτηση Pentium IV, 128 MB RAM και θα έχει την δυνατότητα πρόσβασης από πολλά τερματικά PCs (Pentium II) με λειτουργικό Windows.

## Συμπεράσματα

Η περιγραφή του παραπάνω λογισμικού έρχεται να δώσει στην επιχείρηση μία λύση για την βέλτιστη διαχείριση και διοίκηση μιας επιχείρησης εστίασης. Ποιο συγκεκριμένα το λογισμικό καλύπτει όλους τους στόχους της επιχείρησης καθώς την κάνει ποιο ανταγωνιστική στον κλάδο της, μειώνοντας τα λάθη στις παραγγελίες, καθώς οι παραγγελίες γίνονται μηχανικά και όχι με παρέμβαση ανθρώπου για την καταγραφή και αποστολή τους στην κουζίνα, έτσι καμία παράβλεψη δεν γίνεται και κανένας πελάτης δεν φεύγει δυσαρεστημένος ή δεν έχει παράπονα.

Προσφέρει ψυχαγωγία και διασκέδαση στον τρόπο παραγγελίας, καθώς το πρωτότυπο περιβάλλον μέσα από το οποίο καλείτε ο χρήστης να παραγγείλει είναι εξαιρετικά απλό, εύχρηστο, οικείο και περιέχει όλες τις πληροφορίες για τα προϊόντα, ακόμα ο πελάτης μπορεί να πάρει όσο χρόνο χρειάζεται για να αποφασίσει και να ολοκληρώσει την παραγγελία του όποτε αυτός είναι έτοιμος.

Ο τρόπος παραγγελίας είναι πρωτότυπος και πρωτοποριακός για το χώρο της εστίασης και όσον αφορά την διασκέδαση αλλά και την αποτελεσματικότητα. Ακόμα το λογισμικό είναι ικανό να καλύπτει ανάγκες όλων των εθνικοτήτων και κατηγοριών προσθέτοντας στο περιβάλλον διαχείρισης εικονίδια για την επιλογή των βασικών γλωσσών αλλά και κάθε ηλικία μπορεί να χρησιμοποιήσει το λογισμικό παραγγελίας λόγω της απλότητας και ευχρηστίας αυτού. Από την πλευρά της διαχείρισης του εστιατορίου η παρακολούθηση του χρόνου απόκρισης και προετοιμασίας των παραγγελιών δίνουν την δυνατότητα στον μάνατζερ να εντοπίσει το τμήμα στο οποίο υπάρχει ελλιπής οργάνωση ή και ανάγκη υποστήριξης και βελτίωσης της διαδικασίας καθώς παρακολουθείτε ο χρόνος των βημάτων του κρίσιμου μονοπατιού<sup>5</sup>. Ακόμα η καταγραφή, παρακολούθηση και αποθήκευση των παραγγελιών δίνει την δυνατότητα στον ιδιοκτήτη και τον μάνατζερ να καταλάβει τις προτιμήσεις των καταναλωτών και μέσα από την ανάλυση των στατιστικών στοιχείων ανά χρονικά διαστήματα να συμβάλλει στην βελτίωση των επιλογών του μενού του εστιατορίου.

Η ταχύτητα του χρόνου ετοιμασίας της παραγγελίας αυξάνετε κατά το βέλτιστο δυνατό καθώς όλα τα βήματα μπορούν να ελέγχονται με σκοπό τον εντοπισμό των αδυναμιών και την βελτίωσή τους, την εξάλειψη των λαθών αλλά και την αυτοματοποίηση συγκεκριμένων διαδικασιών όπως η διαδικασία της παραγγελίας. Ως συνέπεια των παραπάνω η επιχείρηση ενισχύει το όνομά της στην αγορά, αυξάνει το πελατολόγιό της, τα κέρδη της και παράλληλα μειώνει τα λειτουργικά της έξοδα.

Η τεχνολογία εξελίσσετε με αλματώδη βήματα και ο άνθρωπος ακολουθεί, η ζήτηση της αγοράς μεταβάλετε σε όλους τους τομείς έτσι και ο τομέας της εστίασης δίνετε να ακολουθήσει και οι επιχειρήσεις να είναι συμβατές με την ζήτηση της αγοράς και αυτή την δυνατότητα την προσφέρει το λογισμικό monopoly.

---

<sup>5</sup> Κρίσιμο μονοπάτι ορίζετε ως το μονοπάτι που αποτελείτε από τις κρίσιμες δραστηριότητες και οποιαδήποτε καθυστέρηση ή μεταβολή ή διακοπή μίας από αυτές μπορεί να προκαλέσει διαταραχή στην διαδικασία και μη ομαλή λειτουργία αυτής μη διεκπεραίωση αυτής.

# ΚΕΦΑΛΑΙΟ 4ο

## Πως θα υλοποιηθεί το λογισμικό σύστημα ;

### Περίληψη

Στο τρίτο κεφάλαιο αναφέρουμε τους σκοπούς και τους στόχους μιας επιχείρησης εστίασης, περιγράφουμε την πρόταση για την λύση η ακόμα καλύτερα την πρόταση για την βελτιστοποίηση της διοίκησης και διαχείρισης μίας επιχείρησης εστίασης, αναφέρουμε τρεις προτάσεις για την επιλογή υλικού φιλοξενίας του λογισμικού και τέλος αναφέρουμε τα συμπεράσματα μέσα από τα οποία τονίζουμε καθαρά πώς το λογισμικό καλύπτει εξ ολοκλήρου όλους τους στόχους της επιχείρησης εστίασης. Στο τέταρτο κεφάλαιο αναφέρουμε την μεθοδολογία σύμφωνα με την οποία θα αναπτυχθεί το λογισμικό. Η ανάπτυξη ενός λογισμικού απαιτεί τις δραστηριότητες της ανάλυσης απαιτήσεων (requirements analysis), του σχεδιασμού και υλοποίησης (design and implementation), της επαλήθευσης και επικύρωσης (verification and validation) και της εξέλιξης (maintenance), αυτές οι διαδικασίες ακολουθούν κάποιο μοντέλο ή μεθοδολογία ανάπτυξης. Στη συνέχεια αναφέρετε η αρχιτεκτονική που ακολουθεί δηλαδή η έμφαση στις υπηρεσίες παγκόσμιου ιστού, μία ιστορική εξέλιξη αλλά και αναφορά στον τρόπο χρήσης και λειτουργίας τους. Αναφέρονται επίσης η γλώσσα προγραμματισμού και τα εργαλεία υλοποίησης, τέλος καταγράφονται τα συμπεράσματα στα οποία τονίζεται η σημασία της υπηρεσιοστρεφούς αρχιτεκτονικής και της ακολουθίας μεθοδολογίας ανάπτυξης.

### Μεθοδολογία ανάπτυξης

Οι διεργασίες ανάπτυξης λογισμικού δηλαδή το σύνολο των δραστηριοτήτων που αποβλέπουν στην ανάπτυξη ενός προϊόντος λογισμικού περιλαμβάνουν τις δραστηριότητες της **ανάλυσης απαιτήσεων (requirements analysis)**, του **σχεδιασμού και υλοποίησης (design and implementation)**, της **επαλήθευσης και επικύρωσης (verification and validation)** και της **εξέλιξης (maintenance)**. Οι διεργασίες ανάπτυξης λογισμικού ακολουθούν κάποιο μοντέλο ή κάποια μεθοδολογία ανάπτυξης. Μερικά από τα μοντέλα και τις μεθοδολογίες ανάπτυξης είναι η «**ωριμότητα της διεργασίας**», η μέθοδος «**φτιάξε και μάλωσε**», το «**μοντέλο του καταρράκτη (waterfall model)**», η «**ανάπτυξη αρχέτυπων**», η «**βηματική εκλέπτυνση**», το «**σπειροειδές μοντέλο**», η «**τυπικές μέθοδοι**», η «**επαναχρησιμοποίηση**», η «**ανάλυση απαιτήσεων**», ο «**σχεδιασμός και υλοποίηση**», η «**επαλήθευση και επικύρωση**» και η «**εξέλιξη**». Η διαδικασία ανάπτυξης λογισμικού βοηθείται από εργαλεία τα οποία χωρίζοντάς τα σε κατηγορίες μπορούμε να τα αναφέρουμε ως εξής:

**Διαχείριση έργου:** Εργαλεία PERT, εργαλεία εκτίμησης και λογιστικά φύλλα.

**Διορθωτές σχεδιαστές:** Διορθωτές κειμένου, επεξεργαστές κειμένου, σχεδιαστές διαγραμμάτων.

**Διαχείριση αλλαγών:** Εργαλεία ανίχνευσιμότητας απαιτήσεων, συστήματα ελέγχου αλλαγών.

**Διαχείριση σχηματισμών:** Εργαλεία διαχείρισης εκδόσεων, εργαλεία χτισίματος συστήματος



**Κατασκευή αρχετύπων:** Γλώσσες πολύ υψηλού επιπέδου, γεννήτριες διεπαφών

**Υποστήριξη μεθόδων:** Λεξικά δεδομένων, γεννήτριες κώδικα, σχεδιαστές

**Μετάφραση:** Μεταγλωττιστές, διερμηνευτές

**Ανάλυση προγραμμάτων:** Αναλυτές κώδικα, αναλυτές εκτέλεσης

**Έλεγχος:** Γεννήτριες ελέγχου, εργαλεία σύγκρισης αρχείων

**Αποσφαλμάτωση:** Αλληλεπιδραστικοί αποσφαλματωτές

**Τεκμηρίωση:** Συστήματα επιτραπέζιας τυπογραφίας, επεξεργαστές εικόνων

**Επανασχεδιασμός:** Συστήματα αναφορών, συστήματα αναδόμησης προγραμμάτων

## Ανάλυση Απαιτήσεων

Η ανάλυση απαιτήσεων περιλαμβάνει τις εξής φάσεις:

- Μελέτη σκοπιμότητας (*feasibility study*).
- Εκμείευση και ανάλυση απαιτήσεων (*requirements elicitation and analysis*).
- Καταγραφή απαιτήσεων (*requirements specification*).
- Επικύρωση απαιτήσεων (*requirements validation*).

## Σχεδιασμός και Υλοποίηση

Οι κύριες δραστηριότητες του σχεδιασμού και της υλοποίησης είναι:

- Αρχιτεκτονικός σχεδιασμός (*architectural design*).
- Ανάλυση υποσυστημάτων (*subsystem analysis*).
- Σχεδιασμός διεπαφών (*interface design*).
- Σχεδιασμός εξαρτημάτων (*component design*).
- Σχεδιασμός δομών δεδομένων (*data structure design*).
- Σχεδιασμός αλγορίθμων (*algorithm design*).

## Επαλήθευση και Επικύρωση

Η επαλήθευση και η επικύρωση περιλαμβάνει:

- Έλεγχο μονάδων (*unit testing*).
- Έλεγχο αρθρωμάτων (*module testing*).
- Έλεγχο υποσυστημάτων (*subsystem testing*).
- Έλεγχο συστήματος (*system testing*).
- Έλεγχο αποδοχής (*acceptance testing*). Περιλαμβάνει *alpha testing*, *beta testing*.

## Εξέλιξη

Όταν λάβουμε υπόψη μας την εξέλιξη του λογισμικού, η διαδικασία ανάπτυξης μεταβάλλεται σε βρόχο αποτελούμενο από τα παρακάτω βήματα:

1. Καθορισμός απαιτήσεων συστήματος.
2. Αποτίμηση υπάρχοντος συστήματος.
3. Προτάσεις αλλαγών.
4. Μεταβολή συστήματος.
5. Έλεγχος.

Για την υλοποίηση του λογισμικού μας θα ακολουθήσουμε την μεθοδολογία του μοντέλου του καταρράκτη γιατί θεωρούμε ότι είναι το πιο αποτελεσματικό μοντέλο καθώς χρειάζεται να ακολουθήσουμε τα βήματα του μοντέλου του καταρράκτη προκειμένου να καταφέρουμε την υλοποίηση ενός αποτελεσματικού για την επιχείρηση λογισμικού.

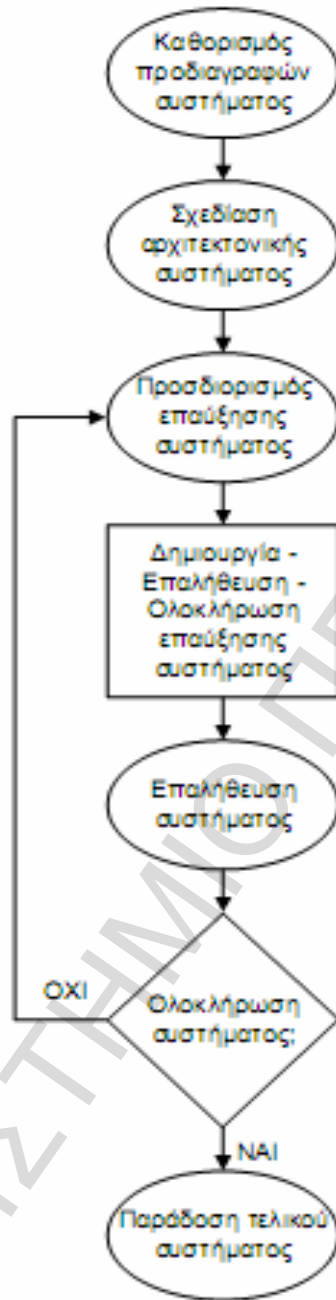
## Μοντέλα ανάπτυξης λογισμικού

Παρακάτω καταγράφονται συνοπτικά μερικά από τα μοντέλα ανάπτυξης λογισμικού, όπου για την εφαρμογή προτείνεται το μοντέλο της λειτουργικής επαύξεσης.

- Μοντέλο καταρράκτη
- Μοντέλο Προτυποποίησης
- Μοντέλο Λειτουργικής Επαύξεσης
- Λειτουργικό Μοντέλο
- Μοντέλο αυτόματου προγραμματισμού
- Μοντέλο επαναχρησιμοποίησης λογισμικού
- Αντικειμενοστραφές μμοντέλο
- Σπειροειδές μμοντέλο

## Μοντέλο Λειτουργικής Επαύξεσης

Μια εναλλακτική διαδικασία που συνδυάζει τα πλεονεκτήματα της εξελικτικής προσέγγισης με τον έλεγχο που απαιτείται για μεγάλα συστήματα είναι η **λειτουργική επαύξεση (incremental development)**. Σύμφωνα με το μοντέλο αυτό αρχικά αναπτύσσεται μια έκδοση του συστήματος που περιέχει τις περισσότερες σημαντικές και κρίσιμες λειτουργίες. Από την χρησιμοποίηση αυτής της έκδοσης κερδίζεται εμπειρία η οποία χρησιμοποιείται για την βελτίωσή της. Στην συνέχεια γίνεται μια προσαύξηση η οποία επεκτείνει την προηγούμενη έκδοση η οποία περιέχει και άλλες λειτουργίες. Η νέα έκδοση εκλεπτύνεται και προσαυξάνεται με την σειρά της με τον ίδιο τρόπο έως ότου κατασκευαστεί η τελική έκδοση. Το μοντέλο ενδείκνυται στις περιπτώσεις που υπάρχει σαφής γνώση και πολύ μικρή ή καθόλου μεταβλητότητα των απαιτήσεων του υπό ανάπτυξη λογισμικού. Άρα πρόκειται για μοντέλο που χρησιμοποιείται σε λίγες περιπτώσεις μια και το βασικό πρόβλημα της ανάπτυξης λογισμικού είναι η ασάφεια (σε μικρότερο ή μεγαλύτερο βαθμό) των απαιτήσεων του συστήματος.



Εικόνα 15 Μοντέλο λειτουργικής επαύξησης

Με αυτό το μοντέλο αποφεύγονται προβλήματα που προκύπτουν από τις συνεχείς αλλαγές, όπως στην εξελικτική προτυποποίηση. Η αρχιτεκτονική του συστήματος καθορίζεται σχετικά νωρίς, και λειτουργεί σαν πλαίσιο. Τα μέρη που αποτελούν το σύστημα αναπτύσσονται με επαυξήσεις και παραδίδονται με αυτό τον τρόπο (βλέπε κ. Βεσκούκης, 2000).

## Γλώσσα προγραμματισμού και εργαλεία υλοποίησης

### Η γλώσσα C#

Η γλώσσα προγραμματισμού που χρησιμοποιείτε για την υλοποίηση του προγράμματος είναι η c#. Η c# είναι κατά κύριο λόγο αντικειμενοστραφής γλώσσα προγραμματισμού η οποία ενσωματώνει κάποια χαρακτηριστικά διαφόρων προγραμματιστικών προτύπων.

Αναπτύχθηκε στην Microsoft, από μία ομάδα την οποία ηγούνταν ο Anders Hejlsberg, σαν μέρος του .NET Framework.

Είναι μια γλώσσα που συνεχώς εξελίσσεται και σε κάθε νέα της έκδοση προστίθενται νέα χαρακτηριστικά και συντακτικός, στόχος της είναι να κάνει τα απλά πράγματα εύκολα, τα δύσκολα εφικτά και τα άσχημα δύσκολα.

Μπορεί κάποιος να την χρησιμοποιήσει χωρίς εντατική εκμάθηση, ενώ ταυτόχρονα είναι εφαρμοσμένη με σύγχρονες προγραμματιστικές πρακτικές. Ένας προγραμματιστής μπορεί σε σύντομο χρονικό διάστημα να είναι παραγωγικός καθώς οι θεμελιώδεις αρχές της μπορούν να κατανοηθούν γρήγορα.

Η C# έχει σχεδιαστεί έτσι ώστε να μειώνεται η πιθανότητα πρόκλησης λαθών από την πολυπλοκότητα του κώδικα, αφού τη μειώνει σε μεγάλο βαθμό με το απλουστευμένο συντακτικό της και την οργάνωση κώδικα της.

Σχεδιαστικέ από τα θεμέλιά της ώστε να είναι αντικειμενοστραφής. Ο αντικειμενοστραφής προγραμματισμός επικράτησε σαν προγραμματιστικό πρότυπο την προηγούμενη δεκαετία και παραμένει στις προτιμήσεις των προγραμματιστών. Οι ανάγκες για καταναεμημένα συστήματα πελάτη-εξυπηρετητή συμπίπτουν με την ενθουλάκωση και την ανταλλαγή μηνυμάτων που είναι βασικά χαρακτηριστικά του αντικειμενοστραφούς προγραμματισμού. Κατά πολλούς ειδικούς στις γλώσσες προγραμματισμού, η επιτυχής λειτουργία των προγραμματιστικών συστημάτων σε δικτυακά περιβάλλοντα αυξανόμενης πολυπλοκότητας βασίζεται στην αντικειμενοστρέφεια. Η C# παρέχει μια ξεκάθαρη και αποδοτική αντικειμενοστρεφή πλατφόρμα παρέχοντας στους προγραμματιστές μια συλλογή βιβλιοθηκών δοκιμασμένων αντικειμένων που παρέχουν λειτουργικότητα που ποικίλει από απλούς τύπους δεδομένων, σε διεπαφές εισόδου/εξόδου ή δικτυακές και εργαλεία για τη δημιουργία παραθυρικών εφαρμογών. Αυτές οι βιβλιοθήκες μπορούν να προσαρμοστούν στις ανάγκες του προγραμματιστή.

Ακόμα η C# υποστηρίζει και τον προγραμματισμό βασισμένο σε components (component-based programming) ο οποίος επιτρέπει τον προσδιορισμό αυτόνομων μονάδων λειτουργικότητας (components) που είναι απομονωμένα και τεκμηριωμένα, παρουσιάζοντας ένα μοντέλο με ιδιότητες, μεθόδους, events και μεταδεδομένα για component. Η C# υποστηρίζει αυτό τα χαρακτηριστικά άμεσα κάνοντας έτσι τη διαδικασία δημιουργίας και χρήσης των components πολύ εύκολη.

Στο ξεκίνημα της υλοποίησης της, οι δημιουργοί της τεχνολογίας C# απέρριψαν την ολοκληρωτική χρήση της C++ σαν γλώσσα υλοποίησης. Στη νέα γλώσσα ωστόσο, κράτησαν αρκετά χαρακτηριστικά της C++ αλλά και της Java και αφαίρεσαν την άχρηστη πολυπλοκότητα και των δύο. Έτσι έχοντας κρατήσει αρκετά από τα αντικειμενοστρεφή χαρακτηριστικά και τη γενική φιλοσοφία της C++ αλλά και τη γενική ευκολία της Java

βελτιώνοντας ορισμένα σημεία της, έτσι λοιπόν είναι σχετικά εύκολη η μετακόμιση στη C# δεδομένου ότι η C και η C++ και η Java διδάσκονται.

## Η ASP.NET

Η ASP.NET δημιουργήθηκε για να καλύψει αυτή την ανάγκη. Η ASP.NET ανήκει στην κατηγορία γλωσσών ανάπτυξης web εφαρμογών που περιλαμβάνει τεχνολογίες όπως η PHP, JSP, Ruby on Rails. Τέτοιου είδους γλώσσες είναι σχεδιασμένες να τρέχουν στο server και όχι στον browser του πελάτη (χρήστη). Ο κώδικας που ο προγραμματιστής γράφει σε ASP.NET εκτελείται στο server και παράγει HTML σελίδες δυναμικά. Ανάλογα με τη λογική της εφαρμογής μια συγκεκριμένη σελίδα μπορεί να μοιάζει διαφορετική π.χ για διαφορετικούς χρήστες, ή σε διαφορετικές χρονικές στιγμές. Αυτό γίνεται αυτόματα χωρίς να χρειάζεται τροποποιήσεις η σελίδα όπως θα είχαμε στην περίπτωση των HTML σελίδων. Οι σελίδες γραμμένες σε ASP.NET ονομάζονται Web Forms και τέτοιου είδους web sites ονομάζονται εφαρμογές web (web applications). Τα αρχεία αυτής της μορφής έχουν κατάληξη «.aspx».

Μερικά από τα βασικά πλεονεκτήματα που καθιέρωσαν την ASP.NET ως μια από τις σημαντικότερες γλώσσες ανάπτυξης web εφαρμογών είναι:

- Χρησιμοποιεί το .NET framework, ένα πακέτο που προσφέρει έτοιμη λειτουργικότητα στον προγραμματιστή για την διευκόλυνσή του με βασικές διαδικασίες.
- Επιτρέπει την συγγραφή κώδικα με μια από περισσότερες από 40 γλώσσες προγραμματισμού μεταξύ των οποίων και οι πιο διαδεδομένες C# και Visual Basic.
- Βάση σχεδιασμού επιτρέπει την ανεξαρτητοποίηση της λογικής σελίδας από το παρουσιαστικό της. Αλλαγές που κάνει ενδεχομένως κάποιος στην εμφάνιση μιας web εφαρμογής δεν επηρεάζουν απαραίτητα και τη λογική της εφαρμογής.
- Υπάρχουν κορυφαία εργαλεία ανάπτυξης εφαρμογών με ASP.NET όπως είναι το Visual Studio.

## Το περιβάλλον του visual studio

Το περιβάλλον υλοποίησης για το λογισμικό μας πρόγραμμα είναι το visual studio. Το Visual Studio είναι το Integrated Development Environment της Microsoft για την ανάπτυξη εφαρμογών που στοχεύουν στο .Net Framework και άλλες πλατφόρμες της Microsoft: από εφαρμογές κονσόλας (console applications) μέχρι web services και εγγενή κώδικα (code editor) με αυτόματη συμπλήρωση (IntelliSense), που επιταχύνει το γράψιμο και ενσωματώνει ένα debugger (αποσφαλματωτή) για τον άμεσο εντοπισμό σφαλμάτων. Άλλα εργαλεία περιλαμβάνουν την σχεδίαση παραθυρικών εφαρμογών και τη σχεδίαση κλάσεων και βάσεων δεδομένων, ενώ να μπορεί να επεκταθεί με τα διάφορα επιπρόσθετα εργαλεία που κυκλοφορούν από τη Microsoft ή τρίτους παρόχους.

## Η πλατφόρμα .NET Framework

Η πλατφόρμα υλοποίησης του λογισμικού μας είναι το .NET Framework<sup>6</sup>. Το .NET Framework είναι ένα πλαίσιο λογισμικού (software framework) που προορίζεται για την πλατφόρμα των windows. Αποτελείτε από

---

<sup>6</sup> Ένα software framework είναι ένα επαναχρησιμοποιήσιμο σύστημα λογισμικού, αποτελούμενο συνήθως

μία μεγάλη βιβλιοθήκη κλάσεων και υποστηρίζει μία πλειάδα γλωσσών προγραμματισμού με τη δυνατότητα η μία να μπορεί να χρησιμοποιηθεί από την άλλη. Τα προγράμματα που γράφονται για το .NET Framework εκτελούνται σε ένα περιβάλλον εκτέλεσης γνωστό ως Common Language Runtime (CLR), ενός ειδικού λογισμικού, σχεδιασμένου να υποστηρίζει την εκτέλεση προγραμμάτων και την απρόσκοπτη συνεργασία με το λειτουργικό σύστημα. Το CLR περιέχει μια εικονική μηχανή (virtual machine) που διαχειρίζεται την εκτέλεση ενός προγράμματος και παρέχει μια σειρά σημαντικών υπηρεσιών όπως ασφάλεια, διαχείριση μνήμης και διαχείριση εξαιρέσεων. Τα προγράμματα που γράφονται για το .NET χρησιμοποιούν τη βιβλιοθήκη κλάσεων (class library) του .NET, η οποία δίνει πρόσβαση στο περιβάλλον εκτέλεσης του (runtime environment). Βασικές λειτουργίες όπως οι γραφικές διεπαφές χρηστών (Graphical User Interfaces - GUIs), η επικοινωνία με βάσεις δεδομένων, η κρυπτογραφία, η ανάπτυξη web εφαρμογών και οι δικτυακές επικοινωνίες παρέχονται μέσω του Application Programming Interface (API) του .NET και μπορούν να συνδυαστούν με κώδικα από τους προγραμματιστές για την δημιουργία ολοκληρωμένων εφαρμογών. Όσο ένα πρόγραμμα περιορίζεται στη χρήση της βιβλιοθήκης κλάσεων του .NET, μπορεί να τρέχει οπουδήποτε υπάρχει εγκατεστημένο το περιβάλλον εκτέλεσης το .NET.

Η γλώσσα C# είναι μια από τις γλώσσες που υποστηρίζει το .NET. ο μεταγλωττιστής της C# στοχεύει ειδικά στο .NET Framework. Αυτό έχει δύο σημαντικές συνέπειες για τη γλώσσα:

- Η αρχιτεκτονική της και οι μεθοδολογίες της αντικατοπτρίζουν τη δομή του .NET.
- Σε πολλές περιπτώσεις, ειδικά χαρακτηριστικά της γλώσσας εξαρτώνται στην πραγματικότητα από τα χαρακτηριστικά του .NET ή τις βασικές κλάσεις του .NET.

Εξαιτίας αυτής της εξάρτησης, είναι σημαντικό κανείς να κατανοήσει την αρχιτεκτονική και τη μεθοδολογία του .NET πριν προχωρήσει στον προγραμματισμό με τη C#.

Κάνοντας μία ιστορική αναδρομή, για να εκτιμήσουμε την αξία του .NET Framework, είναι χρήσιμο να ταξιδέψουμε στο χρόνο 18 χρόνια και να θυμηθούμε τη φύση των τεχνολογιών που υποστήριζαν τα Windows. Παρόλο που μπορεί να φαίνεται το ίδιο, όλα τα λειτουργικά συστήματα από τα Windows 3.1 (εκδόθηκαν το 1992) μέχρι τα Windows 7 και Windows Server 2008 R2 έχουν το ίδιο Windows API στον πυρήνα τους. Με την εμφάνιση όμως των νέων εκδόσεων των Windows, έχει προστεθεί νέα λειτουργικότητα στο API, σαν φυσική απόρροια της εξέλιξης του API και όχι της αντικατάστασής του από ένα νέο. Το ίδιο μπορεί να κανείς να πει για τις περισσότερες από τις τεχνολογίες frameworks που χρησιμοποιούνται για την ανάπτυξη εφαρμογών σε Windows. Για παράδειγμα το πρότυπο COM (Component Object Model) ξεκίνησε σαν OLE (Object Linking and Embedding) και αρχικά αποτελούσε το μέσο με το οποίο διαφορετικοί τύποι εγγράφων του Office μπορούσαν να συνδεθούν, ώστε για παράδειγμα να είναι εφικτό να προστεθεί ένας μικρός πίνακας Excel σε ένα αντίγραφο του Word. Από τότε το COM εξελίχθηκε στο DCOM (Distributed COM) και τέλος στο πρότυπο COM+, μια πολύπλοκη τεχνολογία που σχημάτισε τη βάση της επικοινωνίας σχεδόν όλων των components, όπως επίσης και την υλοποίηση των δοσοληψιών, των υπηρεσιών μηνυμάτων και του pooling αντικειμένων.

---

από βιβλιοθήκες και υποπρογράμματα για την υποστήριξη της εκτέλεσης άλλων προγραμμάτων. Τα frameworks μπορούν να βελτιώσουν την παραγωγικότητα κατά την ανάπτυξη του λογισμικού καθώς αφήνουν τον προγραμματιστή να επικεντρωθεί στην εφαρμογή προς ανάπτυξη και όχι στην υποδομή της. Η κύρια διαφορά με τις απλές βιβλιοθήκες λογισμικού είναι η «αντιστροφή ελέγχου», κατά την οποία ο έλεγχος αρχικοποίησης και κλήσης αντικειμένων μεταφέρεται στο framework, αντίθετα με τις βιβλιοθήκες που ήταν αποκλειστική δουλειά του προγραμματιστή.



Η Microsoft επέλεξε αυτή την εξελικτική προσέγγιση στο λογισμικό για τον προφανή λόγο της διατήρησης της συμβατότητας με τις προηγούμενες εκδόσεις (backward compatibility). Μέσα στα χρόνια, η μεγάλη του λογισμικού από άλλους παρόχους που γράφτηκε για Windows, σίγουρα δεν θα γνώριζε την ίδια επιτυχία αν κάθε φορά που η Microsoft εισήγαγε μια νέα τεχνολογία, έσπαγε την υπάρχουσα υποδομή προγραμματισμού.

Ήταν ξεκάθαρο ότι κάτι έπρεπε να αλλάξει. Η Microsoft δεν μπορούσε να επεκτείνει για πάντα τα ίδια εργαλεία λογισμικού και τις ίδιες γλώσσες, κάνοντας τα πιο πολύπλοκα για να υποστηρίξουν το νέο υλικό (hardware) και ταυτόχρονα να είναι συμβατά με τις προηγούμενες εκδόσεις. Κάποτε έρχεται ένα σημείο που πρέπει να γίνει επανεκκίνηση στη θεώρηση της ανάπτυξης λογισμικού αν είναι επιθυμητά ένα απλό αλλά ταυτόχρονα μοντέρνο σύνολο γλωσσών προγραμματισμού, περιβαλλόντων και εργαλείων ανάπτυξης λογισμικού που διευκολύνει τους προγραμματιστές στην ανάπτυξη λογισμικού.

Σε αυτό το σημείο έγινε ο ερχομός της C# και του .NET στην πρώτη τους ενσάρκωση. Με λίγα λόγια το .NET είναι ένα σύστημα λογισμικού για την υποστήριξη του προγραμματισμού στην πλατφόρμα των Windows. Μαζί με το .NET Framework, η C# είναι μία γλώσσα που σχεδιαστική από την αρχή για να υποστηρίξει και να εκμεταλλευτεί όλη την πρόοδο πάνω στα περιβάλλοντα προγραμματισμού και την θεώρηση του αντικειμενοστρεφούς προγραμματισμού που έχει συντελέσει τα τελευταία 25 χρόνια.

Είναι σημαντικό να ξεκαθαρίσουμε ότι η συμβατότητα με τα παλιότερα συστήματα δεν χάθηκε κατά τη διαδικασία αυτή. Τα υπάρχοντα προγράμματα θα συνεχίσουν να δουλεύουν και το .NET σχεδιαστική με τη δυνατότητα να δουλεύει με το υπάρχον λογισμικό. Τώρα, η επικοινωνία μεταξύ διαφόρων components σε Windows γίνεται σχεδόν εξ'ολοκλήρου σε COM components έτσι ώστε τα .NET components να μπορούν να επικοινωνούν με αυτά.

Ωστόσο, δεν είναι απαραίτητο να γνωρίζει κανείς C# για να προγραμματίσει στο .NET. Η Microsoft επέκτεινε τη C++ και άλλαξε σημαντικά τη Visual Basic για να την μετατρέψει σε μια πιο ισχυρή γλώσσα, έτσι ώστε να καταστήσει δυνατή τη χρήση τους για το .NET. Αυτές οι γλώσσες όμως φέρουν το βάρος της εξέλιξης μέσα στα τελευταία χρόνια αντί να έχουν γραφτεί από την αρχή σύμφωνα με τις σύγχρονες απαιτήσεις.

## Συμπεράσματα

Ακολουθώντας την παραπάνω μεθοδολογία της λειτουργικής επαύξησης και τις δραστηριότητες της ανάλυσης απαιτήσεων, σχεδιασμού και υλοποίησης, επαλήθευσης και επικύρωσης και τέλος της εξέλιξης αλλά και δίνοντας βάση στις υπερεσιοστρεφείς διαδικασίες σπάζοντας σε μικρότερα κομμάτια τον κώδικα και ολοκληρώνοντας την κάθε διαδικασία ξεχωριστά αλλά και επικοινωνώντας μεταξύ τους για την ολοκλήρωση της διαδικασίας είμαστε έτοιμη για την κατασκευή του λογισμικού διαχείρισης και διοίκησης της επιχείρησης εστίασης χρησιμοποιώντας στο δομικό κομμάτι την γλώσσα προγραμματισμού C# και ASP.Net.

# ΚΕΦΑΛΑΙΟ 5ο

## Ανάλυση απαιτήσεων

### Περίληψη

Στο τέταρτο κεφάλαιο αναφέραμε την μεθοδολογία σύμφωνα με την οποία θα αναπτυχθεί το λογισμικό καθώς η ανάπτυξη ενός λογισμικού απαιτεί τις δραστηριότητες της ανάλυσης απαιτήσεων (requirements analysis), του σχεδιασμού και υλοποίησης (design and implementation), της επαλήθευση και επικύρωση (verification and validation) και της εξέλιξης (maintenance), αυτές οι διαδικασίες ακολουθούν κάποιο μοντέλο ή μεθοδολογία ανάπτυξης. Στη συνέχεια αναφέραμε την αρχιτεκτονική που ακολουθείτε δηλαδή υπηρεσίες παγκόσμιου ιστού, ακόμα αναφέραμε μία ιστορική εξέλιξη αλλά και το πώς γίνεται η χρήση και λειτουργίας τους. Αναφέρθηκε επίσης η γλώσσα προγραμματισμού και τα εργαλεία υλοποίησης. Στο 5<sup>ο</sup> κεφάλαιο γίνεται μία περιγραφή της διαδικασίας που αυτοματοποιείτε, με τη χρήση του λογισμικού συστήματος αυτού καθ αυτού. Αναφέρονται οι εμπλεκόμενοι στο πληροφοριακό σύστημα, καθώς το monopoly είναι ένα πληροφοριακό κοινωνικό σύστημα, το οποίο αναπτύσσετε από ανθρώπους κατασκευαστές για ανθρώπους πελάτες. Οι εμπλεκόμενοι, δηλαδή όσοι έχουν ένα ρόλο στην ανάπτυξη του λογισμικού συστήματος είναι οι πελάτες (χρήστες, ιδιοκτήτες) του συστήματος αλλά και οι κατασκευαστές (αναλυτές, σχεδιαστές, προγραμματιστές). Αναλύονται τα ερωτήματα “Τι θα κάνει το σύστημα παραγγελίας?”, “Ποιός θα χρησιμοποιεί το σύστημα?”, “Πού θα φιλοξενηθεί το σύστημα”, “Πότε θα χρησιμοποιείτε το σύστημα”. Στη συνέχεια αναλύεται η αρχιτεκτονική του συστήματος με τη συνοδεία περιγραφικού σχήματος. Γίνεται προσδιορισμός του μοντέλου του πεδίου προβλήματος, περιορισμός των υποψηφίων κλάσεων για την επιλογή των τελικών κλάσεων του συστήματος, με τον προσδιορισμό του μοντέλου περιπτώσεων χρήσης, διακρίνονται οι εξής 4 οντότητες: του πελάτη, του συστήματος, του manager και του διαχειριστή. Στη συνέχεια αναπαριστάτε η λειτουργία του χρήστη από την πλευρά του ιδίου με τη χρήση διαγράμματος περιπτώσεων χρήσης (Use Case Diagram) αλλά και η μοντελοποίηση της στατικής δομής – όψης του συστήματος, με την αναπαράσταση διαγράμματος κλάσεων (Class Diagram). Με την μοντελοποίηση του συστήματος μειώνετε το σημασιολογικό χάσμα μεταξύ του πραγματικού κόσμου και του κόσμου του λογισμικού. Σαν αποτέλεσμα αυτού οι κλάσεις που εμφανίζονται είναι πέντε και όχι τέσσερις. Οι κλάσεις αυτές είναι: Η κλάση **Παραγγελία Πελάτη** (Customer Order), η κλάση **Σύστημα** (System), η κλάση **Υπεύθυνος Συντονισμού** (Manager Table), η κλάση **Ιδιοκτήτης** (Owner) και η κλάση **Στοιχεία** (Element). Τέλος γίνεται η τεκμηρίωση του παραπάνω διαγράμματος.



## Περιγραφή διαδικασίας που αυτοματοποιείτε

Η διαδικασία που αυτοματοποιείτε είναι η παραγγελία του πελάτη. Κατά την επίσκεψη ενός νέου πελάτη στο εστιατόριο θα έπρεπε ο σερβιτόρος να τον επισκεφτεί και να τον ενημερώσει για το μενού του εστιατορίου. Να περιμένει ώστε ο πελάτης να αποφασίσει για την επιλογή του, να καταγράψει την παραγγελία στον κατάλογο και στην συνέχεια να την παραδώσει στον manager, αυτός με την σειρά του να ενημερώσει την κουζίνα ώστε να ξεκινήσει την προετοιμασία. Στην όλη διαδικασία τα λάθη είναι συχνά και πολλές φορές οι πελάτες δεν μένουν ευχαριστημένοι. Λάθη κατά την καταγραφή της παραγγελίας, κατά την μεταφορά καθώς μπορεί να χαθεί κ.τ.λ

Εδώ το λογισμικό μπορεί να αυτοματοποιήσει την διαδικασία της παραγγελία περνώντας στον πελάτη με διασκεδαστικό τρόπο την ολοκλήρωσή της. Ο πελάτης θα μπορούσε να ολοκληρώσει μόνος του την παραγγελία από την επιλογή στον κατάλογο αφής, ή στο τραπέζακι αφής. Η ενημέρωση για το μενού θα μπορούσε να γίνει πάλι ψηφιακά από τον κατάλογο καθώς ακουμπώντας πάνω σε κάθε πιάτο θα εμφανίζετε η περιγραφή του και όλα τα περεταίρω στοιχεία του. Η πιθανότητα για λάθη σχεδόν μηδενίζετε και ο

πελάτης μπορεί με όσο χρόνο θέλει να επιλέξει το μενού του διασκεδαστικά.

### Εμπλεκόμενοι

Το παραπάνω σύστημα είναι ένα «Πληροφοριακό Κοινωνικό Σύστημα» στο οποίο μπορούμε να διακρίνουμε τους παρακάτω εμπλεκόμενους. Τα πληροφοριακά συστήματα είναι και κοινωνικά συστήματα και αναπτύσσονται από ανθρώπους κατασκευαστές για ανθρώπους πελάτες.

### Οι άνθρωποι οι οποίοι έχουν ένα ρόλο στο έργο:

Πελάτες: (χρήστες και ιδιοκτήτες του συστήματος) οι χρήστες είναι οι πελάτες οι οποίοι επισκέπτονται το εστιατόριο και κάνουν χρήση του συστήματος επιτραπέζιας παραγγελίας (MONOPOLIS) για να παραγγείλουν το μενού τους. Ακόμα ως χρήστες του συστήματος είναι και οι εργαζόμενοι στην κουζίνα του κάθε εστιατορίου οι οποίοι ενημερώνονται για την κάθε παραγγελία και ξεκινά η διαδικασία κατασκευής της παραγγελίας. Επίσης σε περίπτωση που το εστιατόριο έχει στη διάθεσή του manager για να ενημερώνει την κουζίνα και να διαχειρίζεται τον πίνακα, καθώς και να κατευθύνει και οργανώνει τους σερβιτόρους και αυτός προφανώς είναι εμπλεκόμενος του συστήματος. Διαχειριστής του συστήματος είναι ο ιδιοκτήτης του εστιατορίου αλλά και χρήστης αυτού αφού θα το χρησιμοποιεί για να ενημερωθεί για την κατάσταση του εστιατορίου κάθε στιγμή.

Κατασκευαστές: (αναλυτές, σχεδιαστές, προγραμματιστές) Κατασκευαστές είναι οι εμπλεκόμενοι που θα φροντίσουν για την ανάλυση, σχεδίαση και υλοποίηση του πληροφοριακού συστήματος.

Συνοψίζοντας και απαντώντας στα ερωτήματα για το τι θα κάνει το σύστημα, ποιος θα χρησιμοποιεί το σύστημα και πού και πότε θα χρησιμοποιείται το σύστημα, έχουμε τα εξής:

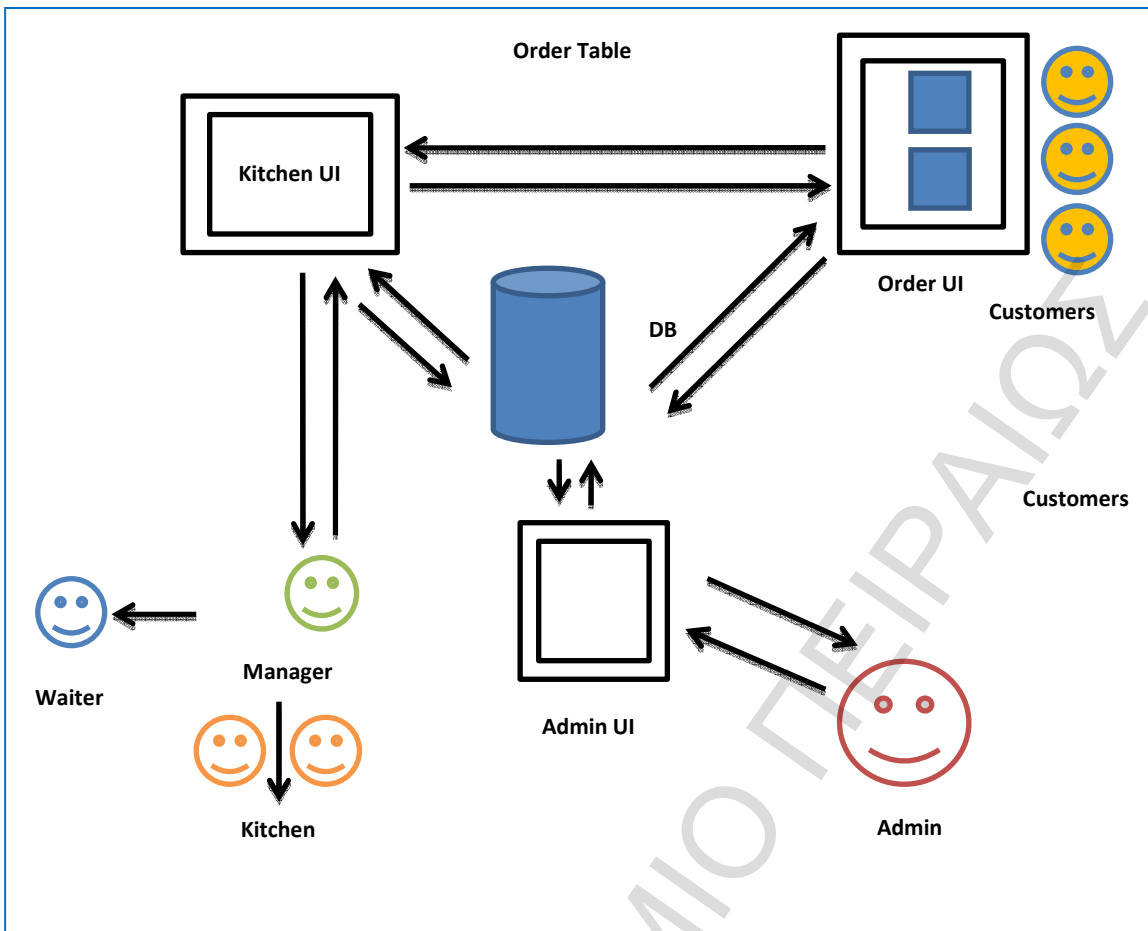
(ΤΙ) το σύστημα είναι ένα σύστημα παραγγελίας επομένως θα καλύπτει την διαδικασία παραγγελίας από το τραπέζι στην κουζίνα.

(ΠΟΙΟΣ) το σύστημα θα το χρησιμοποιούν οι πελάτες του καταστήματος που κάνει χρήση αυτού του συστήματος. Στην συγκεκριμένη περίπτωση οι πελάτες του εστιατορίου. Επίσης θα το χρησιμοποιεί και ο ιδιοκτήτης για την ενημέρωσή του για την κίνηση του μαγαζιού και άλλες διαδικασίες προσθαφαίρεσης στοιχείων κ.τ.λ. ακόμα χρήση του συστήματος θα κάνουν οι εργαζόμενοι στην κουζίνα ή ο manager για την ενημέρωση της αποδοχής έναρξης και ολοκλήρωσης της παραγγελίας.

(ΠΟΥ & ΠΟΤΕ) το σύστημα θα λειτουργεί μόνο τις ώρες λειτουργίας του καταστήματος καθημερινά και Σαββατοκύριακα 4μμ – 2πμ, έτσι θα απασχολεί μέσω πόρους για εκείνες μόνο τις ώρες. Θα χρησιμοποιείται από τους πελάτες και τους εργαζόμενους μέσα στο κατάστημα και από τον ιδιοκτήτη του καταστήματος από οπουδήποτε έχει πρόσβαση μέσω τερματικού υπολογιστή. Ο ιδιοκτήτης θα κάνει αλλαγές στο πρόγραμμα μόνο ώρες εκτός λειτουργίας.

## **Αρχιτεκτονική συστήματος**

Το σύστημα αποτελείται από μία βάση Data Base στην οποία αποθηκεύονται οι παραγγελίες των πελατών. Οι πελάτες επιλέγουν την παραγγελία τους από τα τραπεζάκια αφής ή από τους καταλόγους αφής και αμέσως μετά την επιβεβαίωση η παραγγελία καταχωρείται στην βάση δεδομένων. Οι πελάτες έχουν μία επικοινωνία με την βάση δεδομένων για την καταχώρηση της παραγγελίας. Στην συνέχεια αφού γίνει η καταχώρηση της παραγγελίας αμέσως ενημερώνετε η κουζίνα για την παραγγελία προκειμένου να ξεκινήσει την ετοιμασία της. Η ενημέρωση της κουζίνας γίνεται με την εμφάνιση της παραγγελίας σε ένα πίνακα τον οποίο διαχειρίζεται ο manager, αυτός τσεκάρει μία μία τις παραγγελίες που ετοιμάζονται και ενημερώνει τον σερβιτόρο. Τέλος επικοινωνία με την βάση έχει και ο Διαχειριστής Admin και ο ιδιοκτήτης για την παρακολούθηση της κίνησης του μαγαζιού κάθε στιγμή.



## Προσδιορισμός μοντέλου του πεδίου προβλήματος

Για την υλοποίηση του διαγράμματος συμπεριφοράς θα πρέπει να γίνει πρώτα μία πλήρης περιγραφή της λειτουργίας του συστήματος.

Ο **πελάτης** αφού εισέλθει στο εστιατόριο και κάτσει στο τραπέζι **επιλέγει νέα παραγγελία**. Ολοκληρώνει την επιλογή των στοιχείων και την αποστέλλει στον **manager**. Το **σύστημα αποστέλλει** την παραγγελία στον πίνακα που τον ελέγχει ο **manager** και ξεκινά τον χρόνο αναμονής από την κουζίνα. Ο **manager βλέπει** την παραγγελία και **ενημερώνει** την έναρξης της παραγγελίας. Το **σύστημα** σταματάει και αποθηκεύει τον χρόνο απόκρισης από την κουζίνα και αμέσως ξεκινά την μέτρηση του χρόνου ετοιμασίας. Με την ολοκλήρωση της παραγγελίας ο **manager** ενημερώνει το **σύστημα** και αυτό σταματά το χρόνο ετοιμασίας το **σύστημα** αποθηκεύει την παραγγελία στην βάση δεδομένων και στον πίνακα που κρατάει τις παραγγελίες. Η παραγγελία αποστέλλεται στον **πελάτη**. Ο **διαχειριστής** του συστήματος μπορεί να ενημερώνεται οποιαδήποτε στιγμή για την κατάσταση του εστιατορίου, όμως δεν μπορεί κατά την διάρκεια λειτουργίας του καταστήματος να επέμβει και να αλλάξει στοιχεία σε αυτό. Το **σύστημα** αποθηκεύει τις αλλαγές που έχει κάνει ο διαχειριστής στο ίδιο.

Εξαγωγή αρχικής λίστας ουσιαστικών και πιθανών κλάσεων από την περιγραφή που δόθηκε παραπάνω.

### Λίστα ουσιαστικών

<b>Πελάτης</b>	<b>Είναι βασική οντότητα</b>
Εστιατόριο	Δεν είναι βασική οντότητα, είναι εκτός των ορίων του συστήματος
Τραπέζι	Δεν είναι βασική οντότητα, είναι εκτός των ορίων του συστήματος
Παραγγελία	Δεν είναι βασική οντότητα
<b>Σύστημα</b>	<b>Είναι βασική οντότητα</b>
Παραγγελία	Διπλοεγγραφή
Πίνακα	Δεν είναι βασική οντότητα
<b>Manager</b>	<b>Είναι βασική οντότητα</b>
Κουζίνα	Δεν είναι βασική οντότητα
Manager	Διπλοεγγραφή
Σύστημα	Διπλοεγγραφή
Χρόνο απόκρισης	Είναι ιδιότητα της κλάσης Σύστημα
χρόνο ετοιμασίας	Είναι ιδιότητα της κλάσης Σύστημα
<b>Διαχειριστής</b>	<b>Είναι βασική οντότητα</b>
Στοιχεία	Είναι ιδιότητα της κλάσης Σύστημα

### Περιορισμός Υποψηφίων κλάσεων

Απορρίπτουμε τα εξής από τον παραπάνω πίνακα:

- Διπλοεγγραφές
- Αναφορές στο ίδιο το σύστημα
- Αναφορές σε χειριστές του συστήματος οι οποίοι βρίσκονται έξω από το πεδίο και αναφορές σε οντότητες εκτός του πεδίου του προβλήματος
- Ουσιαστικά που πιθανόν να αποτελέσουν ιδιότητες άλλων κλάσεων

### Παραδοχές

Ο Manager είναι αυτός που ενημερώνει την κουζίνα και τον σερβιτόρο για την αποστολή της παραγγελίας αλλά αυτό δεν είναι μέσα στα όρια του συστήματος.

## Προσδιορισμός μοντέλου περιπτώσεων χρήσης

Το μοντέλο περιπτώσεων χρήσης έχει ως εξής: Διακρίνουμε τις εξής 4 οντότητες

*πελάτης* : Ο πελάτης σαν οντότητα είναι αυτός ο οποίος:

Δημιουργεί νέα παραγγελία.

Προσθέτει σε αυτή στοιχεία

Αφαιρεί από αυτή στοιχεία

Βλέπει τα στοιχεία της παραγγελίας του οποιοδήποτε στιγμή πριν την αποστολή αλλά και μετά.

Επεξεργάζεται την παραγγελία

Αποστέλλει την παραγγελία

*Σύστημα* : Το σύστημα είναι και αυτό οντότητα και ο ρόλος του είναι:

Να ενημερώνει τον manager για την παραγγελία

Να αποθηκεύει την παραγγελία με την ολοκλήρωση της στην βάση δεδομένων.

Αποθηκεύει στη βάση δεδομένων τις αλλαγές στα στοιχεία που έχει κάνει ο διαχειριστής.

*Manager* : ο Manager και αυτός σαν οντότητα έχει τον δικό του ρόλο και αυτός είναι:

Να ενημερώνει για την έναρξη της παραγγελίας

Να ενημερώνει για την ολοκλήρωσή της

*Διαχειριστής* : Ο διαχειριστής έχει και αυτός τον δικό του ρόλο και μπορεί :

Να ενημερώνεται από το σύστημα για την κατάσταση του εστιατορίου

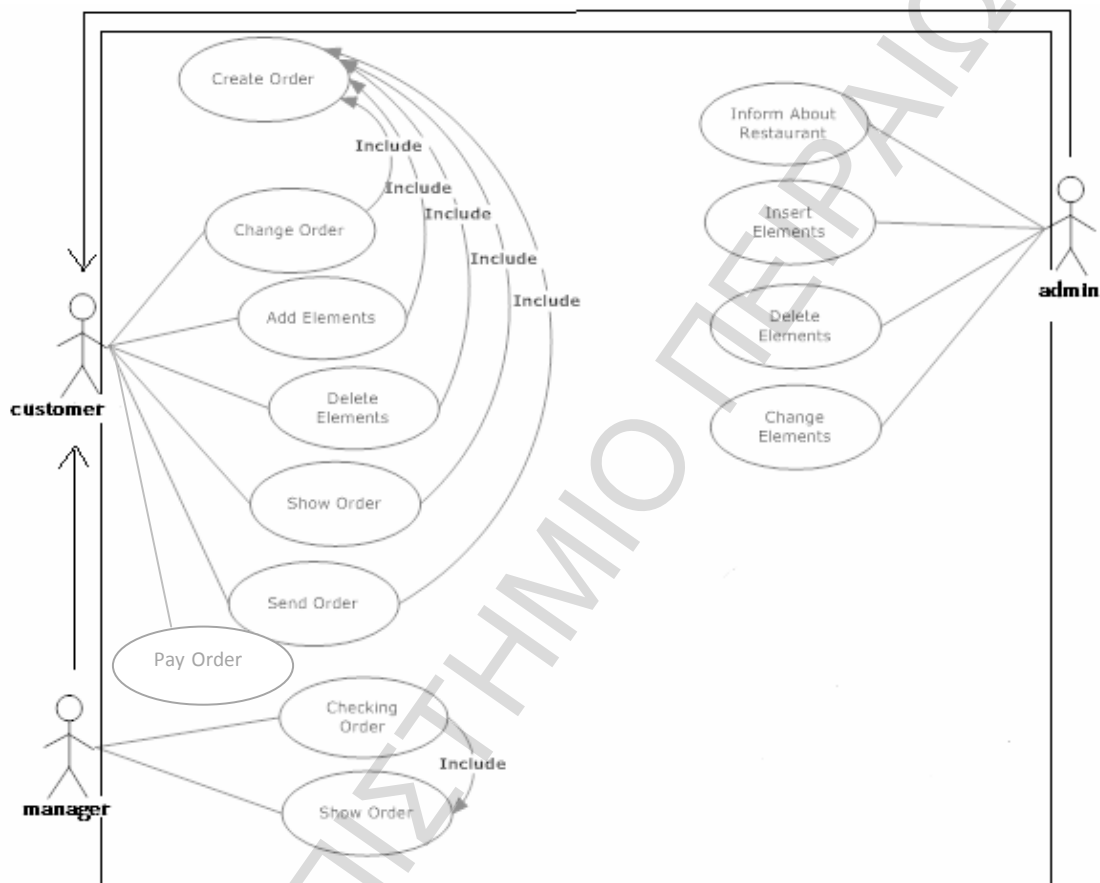
Να διαγράφει στοιχεία από το σύστημα

Να προσθέτει στοιχεία σε αυτό

Να επεξεργάζεται στοιχεία του συστήματος

## Διάγραμμα περιπτώσεων χρήσης Use Case Diagram

Με τα διαγράμματα περιπτώσεων χρήσης, περιγράφουμε την λειτουργία του χρήστη από την πλευρά του χρήστη.



Εικόνα 17 Διάγραμμα περιπτώσεων χρήσης

## Class Diagram

Με τα διαγράμματα κλάσεων γίνεται η μοντελοποίηση της στατικής δομής – όψης του συστήματος. Σκοπός είναι η περιγραφή της δομής δεδομένων που υποστηρίζουν τις επιχειρηματικές διαδικασίες. Επίσης εστιάζουμε στην αντικειμενοστρεφή προσέγγιση, δεδομένα μαζί με λειτουργίες. Είναι σημαντικό να μοντελοποιούμε την δομή του συστήματος για να μειώσουμε το «σημασιολογικό χάσμα» μεταξύ του πραγματικού κόσμου και του κόσμου του λογισμικού. Για να παραστήσουμε πράγματα ιδέες και έννοιες που είναι σημαντικές για το πεδίο εφαρμογής και για να ορίσουμε ένα κοινό λεξιλόγιο για τους αναλυτές και τους χρήστες. Συνοψίζοντας ένα διάγραμμα κλάσεων αναπαριστά τη στατική δομή του συστήματος σε επίπεδο κλάσεων (class structure) και τα περιεχόμενά τους (contents). Αναπαριστά τις σχέσεις μεταξύ των κλάσεων με τη χρήση σχέσεων όπως είναι η συσχέτιση (association), η σχέση εξειδίκευσης (specialization), η σχέση γενίκευσης (generalization) και άλλες. Ένα σύστημα τυπικά έχει πολλά διαγράμματα κλάσεων. Μια κλάση μπορεί να συμμετέχει σε πολλά διαγράμματα κλάσεων.

Έχουμε τις εξής 5 κλάσεις: CustomerOrder, System, ManagerTable, Owner, Element.

Για τις παραπάνω κλάσεις διακρίνουμε ανεργίες που θα μας βοηθήσουν σχεδιάσουμε αντίστοιχα μεθόδους και υποκλάσεις για την καθεμία.

### CustomerOrder:

- Παίζει το video για την ενημέρωση του τρόπου παραγγελίας.
- Δημιουργεί μία παραγγελία με το πάτημα του *Create* , αλλάζοντας το video με μία επιφάνεια όπου θα δεχτεί τα στοιχεία.
- Επιλέγει τα στοιχεία για την παραγγελία και με drop down τα βάζει στην παραγγελία.
- Μπορεί να σβήσει κάποιο στοιχείο αφού το έχει πρώτα επιλέξει και μετά πατήσει *Delete*.
- Μπορεί να καθαρίσει την επιφάνεια από όλα τα στοιχεία πατώντας το *Clear*.
- Και με το *Close* κλίνει το πρόγραμμα.

### System:

- Με το πάτημα του *Sent*, ελέγχει για το αν υπάρχουν στοιχεία στην επιφάνεια, τα οποία στην ουσία με το που τα αφήνει εκεί απλά προστίθενται σε μία λίστα. Αν υπάρχουν στη λίστα τα καταχωρεί μέσα στη βάση, αν όχι εμφανίζει μήνυμα ότι δεν έχουν επιλεγεί στοιχεία προς παραγγελία (μέσα στον πίνακα αποθηκεύονται ακόμα στοιχεία όπως το id του τραπεζιού και το id της παραγγελίας, τα id και τα ονόματα των τραπεζιών βρίσκονται σε μία λίστα <ListTable>).
- Επίσης τραβάει αυτά τα στοιχεία και τα εμφανίζει στον πίνακα της κλάσης *KitchenTable* ο οποίος είναι για την ενημέρωση της παραγγελίας στην κουζίνα.
- Αποθήκευση του χρόνου απόκρισης.
- Αποθήκευση του χρόνου ετοιμασίας.
- Αποθήκευση συνολικού χρόνου παραγγελίας.

### ManagerTable:

- Όταν έρθουν στοιχεία στον πίνακα εμφανίζονται στο αντίστοιχο id row από το τραπέζι που ήρθαν ως “νέα παραγγελία από το τραπέζι “id και όνομα”” και το χρώμα της λάμπας γίνεται πράσινο ακόμα ξεκινά η μέτρηση του χρόνου απόκρισης.
- Με το check από τον manager η παραγγελία ανοίγει και εμφανίζονται αναλυτικά τα στοιχεία της παραγγελίας, η λάμπα γίνεται πορτοκαλή, ο χρόνος απόκρισης σταματά και καλείται μία μέθοδο αποθήκευσης αυτού στην βάση από τον πίνακα System.
- Όταν όλα τα στοιχεία της παραγγελίας είναι τσεκαρισμένα τότε εμφανίζεται μήνυμα το οποίο λέει “είναι σίγουρα όλα ok”. Αν επιλεγεί ναι τότε η λάμπα γίνεται κόκκινη, ο χρόνος ετοιμασίας σταματά και μαζί με τον χρόνο απόκρισης προστίθενται στο συνολικό χρόνο, ο οποίος εμφανίζεται μαζί με τους άλλους δυο, επίσης το κουτάκι για check γίνεται κουτάκι με «x».
- Η παρέα φύγει από το μαγαζί τότε ο manager πρέπει να πατήσει πάνω στο «x» και το χρώμα της λάμπας θα γίνεται πάλι άσπρο.

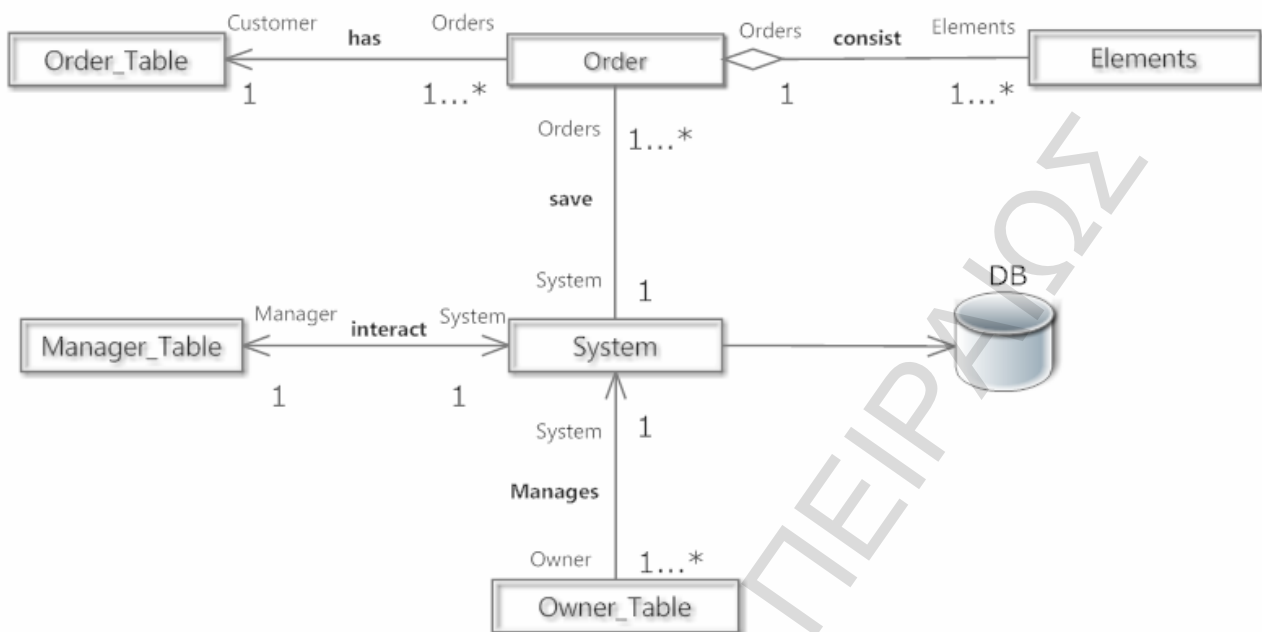
### Owner

- Μπορεί να προσθέτει ή και να διαγράφει τραπέζια από την λίστα.
- Μπορεί να βλέπει τα ενεργά τραπέζια όσα έχουν κάνει σεντ μη διαθ με χ διαθ
- Μπορεί να βλέπει την πίττα.
- Αλλάζει video.
- Αλλάζει στοιχεία.

### Element

- Περιέχει





Εικόνα 18 διάγραμμα κλάσεων

## Τεκμηρίωση

Ένας πελάτης μπορεί να κάνει τουλάχιστον μία ή και περισσότερες παραγγελίες, μία παραγγελία όμως έχει έναν και μόνο εντολέα – πελάτη.

Μία παραγγελία αποτελείται από τουλάχιστον ένα ή και περισσότερα στοιχεία, ένα στοιχείο ανήκει και χαρακτηρίζει συγκεκριμένη παραγγελία. (για την καλύτερη κατανόηση του παραπάνω αναφέρετε το εξής παράδειγμα του μαθητή και της βαθμολογίας που μπορεί να πάρει. Ένας μαθητής λοιπόν μπορεί να πάρει από 1 έως 20, όμως αν πάρει για παράδειγμα 15 τότε αυτή η βαθμολογία χαρακτηρίζει αυτόν και μόνο το μαθητή για το συγκεκριμένο μάθημα, ακόμα και αν τον ίδιο βαθμό πάρουν και άλλη μαθητές το 15 όμως χαρακτηρίζει τον συγκεκριμένο μαθητή έτσι όπως ένα στοιχείο μία παραγγελία).

Το σύστημα μπορεί να αποθηκεύσει μία η και περισσότερες παραγγελίες, μία παραγγελία κάνει χρήση ενός και μόνο συστήματος για την αποθήκευσή της.

Το σύστημα αλληλεπιδρά με ένα manager, ένας manager ενημερώνετε και ενημερώνει το σύστημα.

Ένας η και περισσότεροι μπορεί να είναι οι ιδιοκτήτες και διαχειριστές του συστήματος, ένα σύστημα έχει έναν οι και περισσότερους διαχειριστές.

# ΚΕΦΑΛΑΙΟ 6ο

## Σχεδιασμός

### Περίληψη

Το πέμπτο κεφάλαιο περιγράφει τη διαδικασία που αυτοματοποιείτε, με τη χρήση του λογισμικού συστήματος αυτού καθ' αυτού. Αναλύεται η αρχιτεκτονική του συστήματος με τη συνοδεία περιγραφικού σχήματος. Γίνεται προσδιορισμός του μοντέλου του πεδίου προβλήματος και περιορισμός των υποψηφίων κλάσεων για την επιλογή των τελικών κλάσεων του συστήματος. Αναπαριστάτε η λειτουργία του χρήστη από την πλευρά του ιδίου με τη χρήση διαγράμματος περιπτώσεων χρήσης (Use Case Diagram) αλλά και η μοντελοποίηση της στατικής δομής – όψης του συστήματος, με την αναπαράσταση διαγράμματος κλάσεων (Class Diagram). Με την μοντελοποίηση του συστήματος μειώνετε το σημασιολογικό χάσμα μεταξύ του πραγματικού κόσμου και του κόσμου του λογισμικού. Το 6<sup>ο</sup> κεφάλαιο αναφέρετε στο σχεδιασμό και την υλοποίηση. Καταγράφονται τα πεδία και οι πίνακες, προκειμένου να σχεδιαστεί η βάση δεδομένων. Οι πίνακες είναι δύο βασικοί ο πίνακας Στοιχεία (Elements) και ο πίνακας Παραγγελίες (Orders). Ο πίνακας Elements έχει τα πεδία Id\_ Element, Name, Description, Image, Categories και Price. Ο πίνακας Orders έχει τα πεδία Id\_ Order, Id\_ Element, Total Price, Date, Time, Table\_ Name και Checking. Στη συνέχεια γίνεται μια περιγραφή για τον ρόλο του κάθε πεδίου στους πίνακες, αλλά και η σχεσιακή ανάλυσή των δύο πινάκων. Ακολουθεί η ανάλυση της διαδικασίας και η σχηματική αναπαράσταση της λειτουργίας των υπηρεσιών καθώς και η σειρά που ακολουθεί η μία την άλλη. αυτό είναι και το ποιο κρίσιμο κομμάτι του λογισμικού monopoly. Ακόμα γίνεται αναλυτική περιγραφή των υπηρεσιών (Web Service).

### Βάση δεδομένων

Η βάση δεδομένων περιέχει πίνακα Elements ο οποίος έχει τις εξής στήλες:

#### Elements

<b>Id_ Elements</b>	(χαρακτηρίζει μοναδικά το element)
<b>Id_ Categories</b>	(ξένο κλειδί από τον πίνακα για την κατηγορία που ανήκει το στοιχείο )
<b>Name</b>	(το όνομα του element)
<b>Description</b>	(η περιγραφή για το element)
<b>Image</b>	(η εικόνα που απεικονίζει το elements)
<b>Price</b>	(η τιμή του element)

Ακόμα περιέχει έναν πίνακα Orders στον οποίο θα καταχωρείται η παραγγελία.

Ποιο συγκεκριμένα ο πίνακας έχει τις εξής στήλες:

### Orders

- Id\_Orders** (χαρακτηρίζει μοναδικά την παραγγελία)
- Id\_Element** (είναι το ξένο κλειδί από τον πίνακα element)
- Total\_Price** (είναι η συνολική τιμή της παραγγελίας άθροισμα των τιμών των στοιχείων)
- Date** (είναι η ημερομηνία της παραγγελίας)
- Time** (είναι η χρονική στιγμή που ολοκληρώθηκε η παραγγελία )
- Table\_Name** (είναι το τραπέζι από το οποίο προέρχεται η παραγγελία)
- Checking** (είναι το τσεκάρισμα της παραγγελία για να έρθουν ή όχι τα ποτά άμεσα)

Ακόμα έναν πίνακα με όνομα Categories που περιέχει τις κατηγορίες για τα στοιχεία

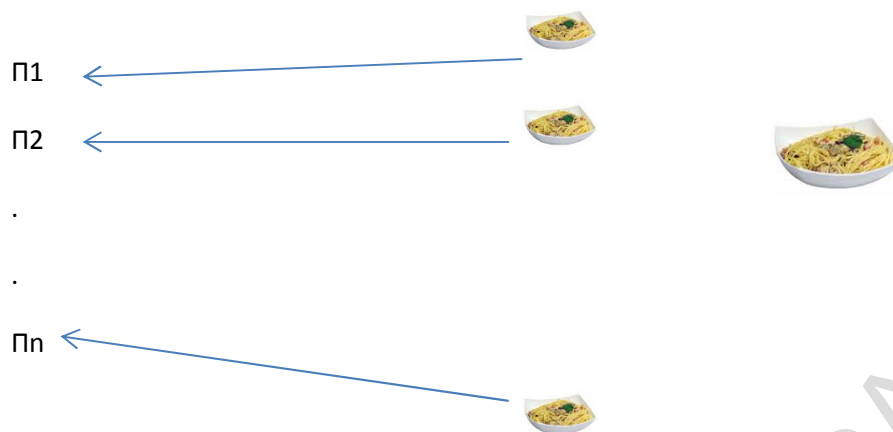
Ο πίνακας έχει τις εξής στήλες:

### Categories

- Id\_Categories** (το Id για τις 4 κατηγορίες )
- Drinks** (Κατηγορία ποτά)
- Salads** (Κατηγορία πρώτο πιάτο, σαλάτες)
- Main\_food** (Κατηγορία κυρίως φαγητό )
- Swiss** (Κατηγορία επιδόρπια γλυκά)

Ένα στοιχείο μπορεί να υπάρχει σε **πολλές** και διαφορετικές παραγγελίες, (κάθε παραγγελία μπορεί να έχει το ίδιο προϊόν π.χ 10 παραγγελίες έχουν καρμπονάρα και coca cola).

**Μία παραγγελία** μπορεί να αποτελείται από πολλά στοιχεία .



Τέλος υπάρχουν στη βάση ένας πίνακας για την εγγραφή και χρήση της βάσης του διαχειριστή admin. Έτσι λοιπόν έχουμε έναν πίνακα admin με τα εξής πεδία:

#### **Admin**

**Id\_admin** (Το κλειδί του κάθε διαχειριστή)

**Name** (Το όνομα του κάθε διαχειριστή)

**Username** (Το συνθηματικό του κάθε διαχειριστή)

**Password** (Το password για την είσοδο του κάθε διαχειριστή)

**Date** (Η ημερομηνία που επισκέφτηκε ο διαχειριστής την εφαρμογή)

## Συμβολισμός



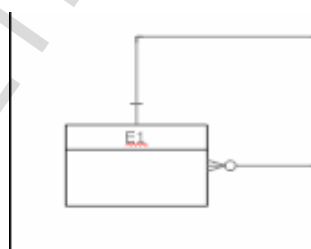
- a) Συσχέτιση 1:1 με υποχρεωτική συμμετοχή της οντότητας E1 και προαιρετική συμμετοχή της οντότητας E2 στη συσχέτιση.



- b) Συσχέτιση 1:N με υποχρεωτική συμμετοχή της οντότητας E1 και προαιρετική συμμετοχή της οντότητας E2 στη συσχέτιση.



- c) Συσχέτιση N:M με προαιρετική συμμετοχή της οντότητας E1 και υποχρεωτική συμμετοχή της οντότητας E2.

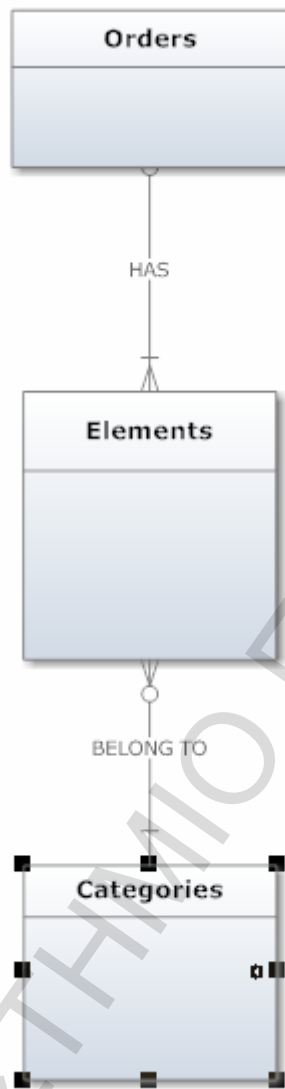


- d) Αυτοπαθής συσχέτιση 1:N με υποχρεωτική συμμετοχή του γονικού ρόλου της οντότητας E και προαιρετική συμμετοχή του απογονικού ρόλου της οντότητας E στη συσχέτιση.



- e) Οντότητα συσχέτισης.

(Γεώργιος Βασιλακόπουλος, 2009)



Εικόνα 19 Μοντέλο οντοτήτων συσχετίσεων

1. Το μοντέλο αυτό περιγράφει την συσχέτιση παραγγελιών, στοιχείων και κατηγοριών.
2. Μία παραγγελία πρέπει να περιέχει τουλάχιστον ένα ή και περισσότερα στοιχεία.
3. Κάθε στοιχείο ανήκει σε μία ή και σε καμία παραγγελία.
4. Ένα στοιχείο πρέπει να ανήκει σε κάποια κατηγορία.
5. Μία κατηγορία μπορεί να έχει πολλά στοιχεία ένα ή και κανένα.

Orders	
Id_Orders	int
Id_Element	int
Total_Price	float
Date	date
Time	time
Table_Name	varchar (25)
Checking	varchar (25)

Id\_Elements = Id\_Elements

Elements	
Id_Elements	int
Id_Categories	int
Name	varchar (25)
Description	text
Image	image
Price	float

Id\_Categories = Id\_Categories

Categories	
Id_Categories	int
Drinks	varchar (25)
First_Food	varchar (25)
Main_Food	varchar (25)
Swiss	varchar (25)

Εικόνα 20 Σχεσιακό Μοντέλο

# ΚΕΦΑΛΑΙΟ 7ο

## Ποια η Συνεισφορά της Εργασίας Αυτής;

### Περίληψη

Το 6<sup>ο</sup> κεφάλαιο αναφέρθηκε στο σχεδιασμό και την υλοποίηση. Καταγράφηκαν τα πεδία και οι πίνακες, προκειμένου να σχεδιαστεί η βάση δεδομένων. Έγινε μια περιγραφή για τον ρόλο του κάθε πεδίου στους πίνακες, αλλά και η σχεσιακή ανάλυσή των δύο πινάκων. Ακλούθησε η ανάλυση της διαδικασίας και η σχηματική αναπαράσταση της λειτουργίας των υπηρεσιών καθώς και η σειρά που ακολουθεί η μία την άλλη. αυτό είναι και το πιο κρίσιμο κομμάτι του λογισμικού monopoly. Ακόμα έγινε αναλυτική περιγραφή των υπηρεσιών (Web Service). Το κεφάλαιο 7<sup>ο</sup> αναπτύσσεται πάνω στο ερώτημα “Ποια η συνεισφορά της εργασίας αυτής ;”. Ποιο συγκεκριμένα προσδιορίζετε τι έμαθα από την ολοκλήρωση της διπλωματικής εργασίας, αλλά και γιατί είναι αυτό σημαντικό και πώς η εργασία μπορεί να συνεισφέρει σε μία επιχείρηση εστίασης. Μέσα από την εργασία γίνεται μία πρόταση εφαρμογής λογισμικού η οποία είναι χρήσιμη γιατί μπορεί να δώσει δυνατότητες εξέλιξης σε μια επιχείρηση και να την βοηθήσει να αναπροσαρμοστεί στο περιβάλλον που συνεχώς εξελίσσεται. Το κεφάλαιο κλείνει με τις μελλοντικές επεκτάσεις του συστήματος monopoly και της εργασίας όπως η άμεση παραγγελία πρώτων υλών, περισσότερα στατιστικά στοιχεία, προσθήκη σχολείων για τα πιάτα κατά την παραγγελία αλλά και μετά για το πόσο καλά η όχι ήταν, σύνδεση του λογισμικού με CRM, δυνατότητα εξέλιξης της διπλωματικής με ολοκληρωμένη οικονομική και τεχνική μελέτη για το κόστος και την εγκατάσταση του λογισμικού.

### Τι έμαθα που δεν ήξερα ;

Με την υλοποίηση της συγκεκριμένης εργασίας, έγινε μια προσπάθεια κατασκευής και παρουσίασης του λογισμικού monopoly για την αποτελεσματικότερη διοίκηση και διαχείριση ενός καταστήματος εστίασης.

Η αρχιτεκτονική που ακολουθήθηκε έδωσε έναν ευκολότερο τρόπο ανάπτυξης, επίβλεψης και λειτουργίας του λογισμικού. Οι υπηρεσίες παγκόσμιου ιστού, Web Service, σαν μικρά αυτόνομα τμήματα κώδικα δίνουν τη δυνατότητα χρήσης τους, οποιαδήποτε στιγμή αποφεύγοντας τις διπλοεγγραφές κώδικα σε ένα λειτουργικό σύστημα. Ο έλεγχος και η επιδιόρθωση της λειτουργικότητας είναι σαφώς πιο εύκολος, άμεσος και αποτελεσματικότερος, γιατί εντοπίζετε άμεσος η υπηρεσία η οποία παρουσιάζει το πρόβλημα. Ακόμα και η εξέλιξη του λογισμικού και η προσθήκη νέων λειτουργιών, γίνεται ευκολότερα.

Η φιλοξενία του λογισμικού σε απομακρυσμένα μηχανήματα, κάποιας εταιρείας που προσφέρει υπηρεσίες τέτοιου είδους Cloud Computing - Hosting, μπορεί να δώσει μόνιμη λύση στην επιχείρηση που θέλει να αποφύγει τυχόν προβλήματα με τα δικά της μηχανήματα και να απολάυσει την ασφάλεια των υπηρεσιών κάποιας εταιρείας.

Επομένως έμαθα ότι ένα λογισμικό, μπορεί να φανεί χρήσιμο σε μία επιχείρηση, να δώσει σε αυτή αποτελεσματικότερη διοίκηση και διαχείριση στη λειτουργία της.



Έμαθα ότι με τη χρήση του κατάλληλου λογισμικού μια επιχείρηση μπορεί να επιτύχει τους στόχους της όπως:

1. Η ενίσχυση της ανταγωνιστικότητας στον κλάδο της.
2. Η αύξηση των κερδών της και παράλληλα η μείωση των λειτουργικών εξόδων.
3. Η αύξηση του πελατολογίου της.
4. Ο συμβατισμός με τη ζήτηση της αγοράς εν γένει.
5. Η αύξηση κατά το βέλτιστων της ταχύτητας του χρόνου προετοιμασίας της παραγγελίας.
6. Ο μηδενισμός των παραβλέψεων που γίνονται στις παραγγελίες.
7. Η καταγραφή, η αποθήκευση και η παρακολούθηση της παραγγελίας.
8. Η παρακολούθηση του χρόνου απόκρισης και προετοιμασίας των παραγγελιών.
9. Η κάλυψη των αναγκών όλων των εθνικοτήτων και ηλικιών των πελατών της.
10. Η προσφορά ψυχαγωγίας και διασκέδασης στον τρόπο παραγγελίας.

Επίσης έμαθα ότι η πρωτοβουλία της εισαγωγής ενός λογισμικού, για την καλύτερη διαχείριση της επιχείρησης και η καινοτομία του λογισμικού μπορούν να αποφέρουν κέρδος αλλά και αβεβαιότητα στην επιχείρηση.

Έμαθα ότι η επιχειρηματικότητα και όχι η τεχνολογία, είναι αυτή που καθορίζει την επιτυχία ή την αποτυχία σε μία επιχείρηση.

Ακόμα έμαθα ότι οι Υπηρεσίες Παγκόσμιου Ιστού είναι μικρές αυτόνομες, ανεξάρτητες εφαρμογές (ή τμήματα λογισμικού) που επιτρέπουν σε άλλα προγράμματα (ετερογενή ή μη) να πραγματοποιούν μία σειρά από ενέργειες. Μπορούν να βρίσκονται σε διαφορετικά ανά τον κόσμο σημεία αλλά να συνθέτουν και να συνεργάζονται για την υλοποίηση και λειτουργία του ίδιου λογισμικού.

Αλλά και ότι το Cloud Computing είναι μία δομή, με την οποία μας δίνεται η δυνατότητα να έχουμε πρόσβαση και να χρησιμοποιούμε web εφαρμογές χωρίς να τις διαθέτουμε στον υπολογιστή μας ή σε κάποια άλλη συσκευή που είναι διασυνδεδεμένη με το ιντερνέτ. Σε αυτή τη δομή η εφαρμογή βρίσκεται σε ένα server και εμείς τη χρησιμοποιούμε χωρίς να χρειάζεται να την εγκαταστήσουμε στον υπολογιστή μας.

### **Γιατί αυτό είναι σημαντικό ;**

Έγινε μία σύνταξη ολοκληρωμένης πρότασης για την χρήση ενός λογισμικού, αναλύοντας την αρχιτεκτονική κατασκευής αλλά και προτείνοντας λύσεις για την εφαρμογή του. Η πρόταση αυτού του λογισμικού αλλά και οι λύσεις, είναι σημαντικά, γιατί μπορούν να δώσουν τη δυνατότητα εξέλιξης σε μία επιχείρηση, και να την βοηθήσουν να αναπροσαρμοστεί στο περιβάλλον που συνεχώς εξελίσσεται, αλλά και να προσδιορίσει νέους στόχους με τη χρήση του λογισμικού monopoly. Για μια επιχείρηση εστίασης η μελέτη της συγκεκριμένης εργασίας θα την βοηθήσει στον προσανατολισμό της επίτευξης των υπάρχοντων στόχων μέσα από τη χρήση της τεχνολογίας ή στην δημιουργία νέων στόχων με μεγαλύτερες προσδοκίες για την επιχείρηση.

## Μελλοντικές επεκτάσεις εξέλιξη

Το λογισμικό είναι ευέλικτο από τη φύση του και μπορεί να τροποποιείται. Καθώς αλλάζουν οι απαιτήσεις λόγω των μεταβαλλόμενων επιχειρηματικών συνθηκών, το λογισμικό που υποστηρίζει την επιχείρηση πρέπει επίσης να εξελίσσεται και να μεταβάλλεται. Αν και υπήρχε μια διαχωριστική γραμμή μεταξύ της διαδικασίας ανάπτυξης λογισμικού και της διαδικασίας εξέλιξης (συντήρησης), η διάκριση αυτή ολοένα και χάνει το νόημά της αφού όλο και λιγότερα συστήματα λογισμικού είναι σήμερα εντελώς καινούργια .

Το λογισμικό μπορεί να είναι κατασκευασμένο να καλύπτει βραχυπρόθεσμους στόχους σε μια επιχείρηση αλλά και μακροπρόθεσμους στόχους υπάρχοντες ή μη, με προοπτική να μπορεί να επεκτείνεται και να εξελίσσεται. Τέτοιου είδους μελλοντικές επεκτάσεις μπορεί να είναι πολλές και διάφορες, επεκτάσεις που να καλύπτουν νέες ανάγκες για την επιχείρηση, και να δημιουργούν έφορο έδαφος για της εξέλιξη της σε μια επιχείρηση απόλυτος βιώσιμη και ανταγωνιστική.

Οι μελλοντικές επεκτάσεις της λογισμικής εφαρμογής είναι πολλές και σίγουρα ανοίγουν για την επιχείρηση ευκαιρίες για κέρδος και εξέλιξη. Μερικές από αυτές είναι:

- Άμεση παραγγελία πρώτων υλών όπως π.χ ζυμαρικά, φρούτα, λαχανικά, κρέατα, καφές, αναψυκτικά, έτοιμα γλυκά κ.τ.λ με ενημέρωση του συστήματος και αυτό με τη σειρά του να κάνει αυτόματα άμεση παραγγελία στον παραγωγό. Π.χ αν τα ζυμαρικά έχουν φτάσει στο όριο το σύστημα να πετάει μήνυμα για την κατανάλωσή τους και να ζητά παραγγελία ζυμαρικών, με την έγκριση από τον διαχειριστή το σύστημα να πραγματοποιεί την παραγγελία στον προμηθευτή.
- Περισσότερα στατιστικά στοιχεία για τις παραγγελίες, όπως π.χ ποιο ήταν το πιάτο με την περισσότερη κατανάλωση για τον μήνα Ιανουάριο, ή ποια εποχιακά πιάτα προτιμώνται ή αντίστοιχα δεν προτείνονται καθόλου.
- Η χρήση του internet στο tablet ή το τραπέζι παραγγελίας κατά την διάρκεια παραμονής του πελάτη στο εστιατόριο, είναι μια επέκταση που θα δώσει την δυνατότητα στο χρήστη να συνδυάσει την δουλειά ή και την διασκέδασή του με την εστίαση.
- Στοιχεία λογιστικά για το κόστος αγοράς πρώτων υλών, τα έσοδα και τα έξοδα κάθε μέρα, εβδομάδα, μήνα ή και χρόνο. Δεδομένα ευαίσθητα που αφορούν των ιδιοκτήτη.
- Συνεργασία του λογισμικού με εφαρμογές σε κινητά τηλέφωνα για την ολοκλήρωση μιας παραγγελίας, θα γίνεται η κράτηση του αριθμού και των στοιχείων όπως όνομα, επώνυμο και τοποθεσία μέσω συντεταγμένων και ανάλογα με την απόσταση το μαγαζί θα μπορεί ή όχι να εξυπηρετεί τον πελάτη.
- Σύνδεση του λογισμικού με CRM (Customer Relationship Manager), ώστε να επωφεληθεί από τα πλεονεκτήματα αυτού.

Τα βασικότερα πλεονεκτήματα της χρήσης του CRM είναι και τα βασικά χαρακτηριστικά λειτουργίας της σύγχρονης επιχείρησης που διαθέτει όραμα και στρατηγική:

- Γρήγορη διακίνηση της πληροφορίας
- Ταχύτεροι κύκλοι πωλήσεων

- Ιεράρχηση διαδικασιών
  - Ολοκληρωμένη πληροφόρηση εικόνας επιχείρησης
  - Ανάλυση εργασιών – δραστηριοτήτων
  - Εντοπισμός σημαντικότερων πελατών
  - Στοχευμένη εμπορική επικοινωνία
  - Περιορισμός απωλειών
  - Δημιουργία άψογης σχέσης με τους πελάτες
  - Ποιοτική παροχή υπηρεσίας
- Δυνατότητα εξέλιξης της διπλωματικής όπως ολοκληρωμένης οικονομικής και τεχνικής μελέτης, για την εγκατάσταση και χρήση του λογισμικού σε μια επιχείρηση, από φοιτητή τμήματος πληροφορικής.
  - Δυνατότητα προσθήκης σχολείων για κάθε πιάτο, όπως για παράδειγμα η επιθυμία ενός πελάτη να είναι η μπριζόλα του καλοψημένη ή όχι.

## Συμπεράσματα

Η πολυπλοκότητα των προβλημάτων, που αντιμετωπίζουν σήμερα οι επιχειρήσεις, ο γρήγορος ρυθμός μεταβολών των συνθηκών του οικονομικού, τεχνολογικού, κοινωνικού και πολιτικού περιβάλλοντος, η ένταση του ανταγωνισμού στο διεθνή χώρο και οι ενοποιήσεις των αγορών, καθιστούν αναγκαίο τον προγραμματισμό για την ανάπτυξη και επιβίωση των επιχειρήσεων.

Η τεχνολογική εξέλιξη επηρεάζει τις επιχειρήσεις με διαφόρους τρόπους. Τεχνολογικές καινοτομίες συμβάλουν στη μεταβολή της ζήτησης ενός προϊόντος ή υπηρεσίας μιας επιχείρησης. Κατά ένα βαθμό επηρεάζουν την αξιοπιστία του προϊόντος και προσθέτουν ποιοτικά χαρακτηριστικά στην υπηρεσία ή το προϊόν. Η έλλειψη προσθήκης των τεχνολογικών εξελίξεων από μια επιχείρηση ίσως την οδηγήσουν στον περιορισμό των πωλήσεων και την μείωση των κερδών της επιχείρησης εφόσον είναι δυνατόν το σύστημα παραγωγής του τελικού προϊόντος της επιχείρησης να χαρακτηριστεί απαρχαιωμένο και το προϊόν «ξεπερασμένο». Επιπλέον, πολλές φορές οι τεχνολογικές εξελίξεις δύναται να συμβάλουν στη μείωση του κόστους παραγωγής.

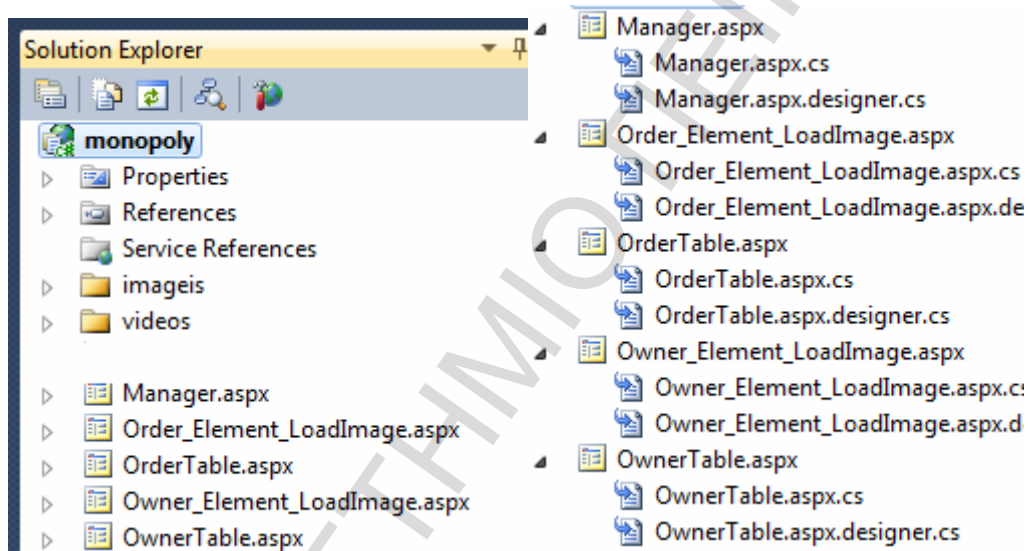
Το λογισμικό μπορεί να βοηθήσει τη διαχείριση και διοίκηση ενός καταστήματος εστίασης, σίγουρα μπορεί να οργανώσει μια επιχείρηση και ιδιαίτερα επιχειρήσεις όπως τα Goodies και Mc.Donald τα οποία έχουν συγκεκριμένο μενού και θα μπορούσε σε αυτά να δώσει μία λύση για την συνολική διοίκηση και διαχείριση τους.

Λόγο κόστους τα τραπεζάκια αφής ενδείκνυνται κυρίως για εστιατόρια hi-tech , είδη Στην Αμερική έχουν τοποθετηθεί και σε hi-tech εστιατόρια όπου οι πελάτες τρώνε πάνω σε αυτά παραγγέλνοντας ότι θέλουν άμεσα και διαβάζοντας αναλυτικά το τι περιέχουν τα πιάτα τους, πως είναι μαγειρεμένα τα φαγητά και πόσες θερμίδες και άλλα θρεπτικά στοιχεία έχουν αυτά. Σε αντίθετη περίπτωση τα tablet είναι μια ποιο οικονομική λύση.

Είναι ένα λογισμικό που μπορεί να καλύψει τους στόχους της επιχείρησης και έχει δυνατότητες εξέλιξης και επέκτασης. Η εργασία προσδοκεί να δημιουργήσει ερεθίσματα στους αναγνώστες, νέες ιδέες και προβληματισμούς.

## Παράρτημα

Παρακάτω παρουσιάζεται ο κώδικας για την δοκιμαστική έκδοση της εφαρμογής στα περιβάλλον του χρήστη, του διαχειριστή και του manager αντίστοιχα και ο σαφής διαχωρισμός του σε κώδικα εμφάνισης (user interface) και λογικού μέρους. Το λογισμικό διαχωρίζετε σύμφωνα με την φιλοσοφία της γλώσσας c sharp με την οποία έχει υλοποιηθεί και ο κώδικας εμφάνισης είναι υπεύθυνος για την εμφάνιση της σελίδας, ενώ ο κώδικας της λογικής αφορά την λογική λειτουργίας της σελίδας.



## Περιβάλλον χρήστη Order Table

### User interface





Στιγμιότυπο από την διαδικασία επιλογής μενού.



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="OrderTable.aspx.cs"
Inherits="monopoly.WebFormOrderTable" %>
```

```
<%@ Register assembly="ASPNetFlashVideo.NET3" namespace="ASPNetFlashVideo"
tagprefix="ASPNetFlashVideo" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title></title>
```

```
<style type="text/css">
```

```
.style2
```

```
{
```

```
width: 93px;
```

```
}
```

```
.style3
```

```
{
```

```
width: 391px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body bgcolor="#000000">
```

```
<form id="WebForm1" runat="server">
```

```
<table align="center" style="width: 600px; height: 750px" border="2">
```

```
<tr>
```

```
<td align="center" class="style2">
```

```
<asp:Panel ID="OrderName" runat="server" Height="106px"
```

```
BackImageUrl="~/imageis/papiros.png">
```

```
<asp:Label ID="Name" runat="server" Text="Name"
```

```
ForeColor="Red" Width="80px"></asp:Label>
```

```
</asp:Panel>
```

```
</td>
```

```
<td align="center" bgcolor="Yellow" class="style3">
```

```
<asp:Image ID="Image2" runat="server"
```

```
ImageUrl="~/imageis/sef.png"
```

```
Height="107px" />
```

```
</td>
```

```
<td align="center">
```

```
Height="106px"
```

```
<asp:Panel ID="OrderDescription" runat="server"
```

```
BackImageUrl="~/imageis/papiros.png" BackColor="Black">
```

```
<asp:Label ID="Description" runat="server"
```

```
Text="Description"
Width="80px" ForeColor="Red"></asp:Label>
```

```
</asp:Panel>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td align="left" class="style2">
```

```
<asp:Button ID="CreateButton" CssClass="button" runat="server"
```

```
Text="Create" BackColor="Black" ForeColor="#009900" Font-
```

```
Bold="True"
```

```
Height="90px" Width="90px" onclick="CreateButton_Click"
```

```
/>
```

```
</td>
```

```
<td class="style3">
```





```


        ImageUrl="~/imageis/undo.png" Width="23px"
onclick="Btnundu_Click" />
Enabled="False" Height="26px"
        ImageUrl="~/imageis/redo.png" Width="25px"
onclick="Btnredu_Click" />
Enabled="False" Height="27px"
        ImageUrl="~/imageis/drink.jpg" Width="27px"
onclick="Btndrink_Click" />
Text="Clear" Enabled="False"
        &nbsp;<asp:ImageButton ID="Btnclear" runat="server"
        Width="46px" onclick="Btnclear_Click" />
    </asp:Panel>
</td>
<td>
    <asp:Panel ID="SaladsPanel" runat="server" Height="440px"
        Width="90px" Enabled="False" Direction="RightToLeft">
    </asp:Panel>
</td>
</tr>
<tr>
<td align="left" class="style2">
    <asp:Button ID="DeleteButton" CssClass="button" runat="server"
        BackColor="Black" Font-Bold="True" ForeColor="Red"
        Width="90px" onclick="DeleteButton_Click" />
</td>
<td class="style3">
    <asp:Panel ID="SweetsPanel" runat="server" Height="90px"
        ScrollBars="Horizontal" Width="391px"
        HorizontalAlign="Right" Wrap="False" Enabled="False">
    </asp:Panel>
</td>
<td align="right">
    <asp:Button ID="CloseButton" CssClass="button" runat="server"
        BackColor="Black" Font-Bold="True" ForeColor="Yellow"
        Width="90px" onclick="CloseButton_Click" />
</td>
</tr>
</table>

</form>

</body>
</html>

```

## Λογικό μέρος

▸  OrderTable.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Services;
using System.Media;
using System.Drawing;
using System.Web.Hosting;
using System.IO;
using System.Data.SqlClient;
using System.Data;
using System.Text;

namespace monopoly
{
    public partial class WebFormOrderTable : System.Web.UI.Page
    {
        /*
        είναι ένα public dictionary. Σε αυτό το dictionary με όνομα Orders κρατά τα
        προϊόντα απο το πιάτο αλλά και τον αριθμό για το πάτημα του καθενός κάθε φορά */
        public Dictionary<string, int> Orders
        {
            get//πέρνει την κατάσταση του Dictionary αν είναι κενό δημιουργεί ένα νέο, αλλιώς
            επιστρέφει το ίδιο
            {
                if (ViewState["Orders"] == null)
                {
                    ViewState["Orders"] = new Dictionary<string, int>();
                }
                return ViewState["Orders"] as Dictionary<string, int>;
            }
            set//είτε το δημιουργεί είτε το επιστρέφει, θέτει σε αυτό μία μεταβλητή value για
            να το αρχικοποιήσει..!
            {
                ViewState["Orders"] = value;
            }
        }

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!Page.IsPostBack)
            {
                this.Orders = new Dictionary<string, int>();
            }
            // Σύνδεση με την βάση
            SqlConnection sc = new
            SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
            //Μεταβλητές για ερώτηση
            SqlCommand cmdrink;
            SqlCommand cmsalad;
            SqlCommand cmmainfood;
            SqlCommand cmswiss;
            //δημιουργία dataset
            DataSet dsdrink = new DataSet();
```

```

DataSet dssalad = new DataSet();
DataSet dsmainfood = new DataSet();
DataSet dsswiss = new DataSet();
try
{
    //η σύνδεση άνοιξε
    sc.Open();
    //4 ερωτήματα πέρνουν απο την βάση τα στοιχεία για τα elements
    cmdrink = new SqlCommand("Select
Id_Elements,Name,Description,Image,Price,categories FROM Element WHERE categories = 'drink'",
sc);
    cmsalad = new SqlCommand("Select
Id_Elements,Name,Description,Image,Price,categories FROM Element WHERE categories = 'salad'",
sc);
    cmmainfood = new SqlCommand("Select
Id_Elements,Name,Description,Image,Price,categories FROM Element WHERE categories =
'main_food'", sc);
    cmswiss = new SqlCommand("Select
Id_Elements,Name,Description,Image,Price,categories FROM Element WHERE categories = 'swiss'",
sc);

    //Εισαγωγή δεδομένων σε adapters
    SqlDataAdapter adapterdrink = new SqlDataAdapter(cmdrink);
    SqlDataAdapter adaptersalad = new SqlDataAdapter(cmsalad);
    SqlDataAdapter adaptermainfood = new SqlDataAdapter(cmmainfood);
    SqlDataAdapter adapterswiss = new SqlDataAdapter(cmswiss);

    adapterdrink.Fill(dsdrink, "Element");
    adaptersalad.Fill(dssalad, "Element");
    adaptermainfood.Fill(dsmainfood, "Element");
    adapterswiss.Fill(dsswiss, "Element");

    //Τα δεδομένα μπαίνουν σε πίνακες
    DataTable dtdrink = dsdrink.Tables["Element"];
    DataTable dtsalad = dssalad.Tables["Element"];
    DataTable dtmainfood = dsmainfood.Tables["Element"];
    DataTable dtswiss = dsswiss.Tables["Element"];

    //Μεταβλητές που κρατούν αριθμό για τις γραμμές των πινάκων
    int countdrink = dtdrink.Rows.Count;
    int countsalad = dtsalad.Rows.Count;
    int countmainfood = dtmainfood.Rows.Count;
    int countswiss = dtswiss.Rows.Count;

    //Τρέχουν 4 for για να διαβάζονται τα στοιχεία για κάθε πίνακα ανα γραμμή
    for (int j = 0; j < countdrink; j++)
    {
        DataRow row = dtdrink.Rows[j]; //στη row ρίχνω την γραμμή j του πίνακα
        //στον πίνακα data της γραμμής j ρίχνω την στήλη Image όπου είναι η εικόνα
        byte[] data = (byte[])dtdrink.Rows[j]["Image"];
        ImageButton imgdrink = new ImageButton();
        imgdrink.Width = 65; //πλάτος εικόνας
        imgdrink.Height = 65; //ύψος εικόνας
        imgdrink.BorderColor = System.Drawing.Color.Black; //περίγραμμα εικόνας
        imgdrink.BorderStyle = BorderStyle.Dashed; //στήλ περιγράμματος
        imgdrink.BorderWidth = Unit.Pixel(1); //πάχος περιγράμματος
        //κάθε φορά τραβάει το url για να γεμίζει με την εικόνα το imagebutton, η
        εικόνα είναι αυτή που τραβιέτε κάθε φορά
        imgdrink.ImageUrl = "Order_Element_LoadImage.aspx?ImageId=" +
row["Id_Elements"];
        //για id του κουμπιού κρατάω το id της γραμμής απο τον πίνακα που το έχω
        τραβήξει από την βάση
        imgdrink.ID = dtdrink.Rows[j]["Id_Elements"].ToString();
        //ο ακροατής για το event
        imgdrink.Click += new ImageClickEventHandler(imgdrink_Click);
    }
}

```

```

        //προσθέτω στον panel το κουμπί
        DrinkPanel.Controls.Add(imgdrink);
    }

    for (int j = 0; j < countsalad; j++)
    {
        DataRow row = dtsalad.Rows[j];
        byte[] data = (byte[])dtsalad.Rows[j]["Image"];
        ImageButton imgsalad = new ImageButton();
        imgsalad.Width = 65;
        imgsalad.Height = 65;
        imgsalad.BorderColor = System.Drawing.Color.Black;
        imgsalad.BorderStyle = BorderStyle.Dashed;
        imgsalad.BorderWidth = Unit.Pixel(1);
        imgsalad.ImageUrl = "Order_Element_LoadImage.aspx?ImageId=" +
row["Id_Elements"];
        imgsalad.ID = dtsalad.Rows[j]["Id_Elements"].ToString();
        imgsalad.Click += new ImageClickEventHandler(imgsalad_Click);
        SaladsPanel.Controls.Add(imgsalad);
    }

    for (int j = 0; j < countmainfood; j++)
    {
        DataRow row = dtmainfood.Rows[j];
        byte[] data = (byte[])dtmainfood.Rows[j]["Image"];
        ImageButton imgmainfood = new ImageButton();
        imgmainfood.Width = 65;
        imgmainfood.Height = 65;
        imgmainfood.BorderColor = System.Drawing.Color.Black;
        imgmainfood.BorderStyle = BorderStyle.Dashed;
        imgmainfood.BorderWidth = Unit.Pixel(1);
        imgmainfood.ImageUrl = "Order_Element_LoadImage.aspx?ImageId=" +
row["Id_Elements"];
        imgmainfood.ID = dtmainfood.Rows[j]["Id_Elements"].ToString();
        imgmainfood.Click += new ImageClickEventHandler(imgmainfood_Click);
        FoodsPanel.Controls.Add(imgmainfood);
    }

    for (int j = 0; j < countswiss; j++)
    {
        DataRow row = dtswiss.Rows[j];
        byte[] data = (byte[])dtswiss.Rows[j]["Image"];
        ImageButton imgswiss = new ImageButton();
        imgswiss.Width = 65;
        imgswiss.Height = 65;
        imgswiss.BorderColor = System.Drawing.Color.Black;
        imgswiss.BorderStyle = BorderStyle.Dashed;
        imgswiss.BorderWidth = Unit.Pixel(1);
        imgswiss.ImageUrl = "Order_Element_LoadImage.aspx?ImageId=" +
row["Id_Elements"];
        imgswiss.ID = dtswiss.Rows[j]["Id_Elements"].ToString();
        imgswiss.Click += new ImageClickEventHandler(imgswiss_Click);
        SweetsPanel.Controls.Add(imgswiss);
    }
}

catch (Exception)
{
}

finally
{
    //κλείνει η βάση αφού έχει φορτώσει τις εικόνες
    sc.Close();
}

```

```

    }
}

int countdrink = 0;
//Τι συμβαίνει με το πάτημα μίας εικόνας απο τα ποτά.!
void imgdrink_Click(object sender, ImageClickEventArgs e)
{//απαραίτητο για να ξεχωρίζει απο πού έχει έρθει, σε ποιο κουμπί αντιστοιχεί
    ImageButton imgdrink = sender as ImageButton;
    imgdrink.BorderColor = System.Drawing.Color.Black;
    imgdrink.BorderStyle = BorderStyle.Double;
    var imageId = imgdrink.ID; // στη μεταβλητή imageId κρατά το ID του κουμπιού που
πατήθηκε
    SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
    SqlCommand cmddrink;
    sc.Open();
    cmddrink = new SqlCommand("Select Id_Elements,Name,Description,Image FROM Element
WHERE categories = 'drink' AND Id_Elements =" + imageId, sc);

    countdrink++;
    Label countdr;

    SqlDataAdapter adaptersalad = new SqlDataAdapter(cmddrink);
    DataSet dsdrink = new DataSet();
    adaptersalad.Fill(dsdrink, "Element");
    DataTable dtdrink = dsdrink.Tables["Element"];
    DataRow row = dtdrink.Rows[0];//στη row ρίχνω την γραμμή j του πίνακα
//στον πίνακα data της γραμμής 0 ρίχνω την στήλη Image όπου είναι η εικόνα
byte[] data = (byte[])dtdrink.Rows[0]["Image"];
    ImageButton imgdrinkcopy;

    SqlDataReader reader = cmddrink.ExecuteReader();
    reader.Read();
    Label lbn = new Label();
    Label lbd = new Label();
    lbn.Text = reader["Name"].ToString() + "<br/>";
    lbd.Text = reader["Description"].ToString() + "<br/>";
    OrderName.Controls.Add(lbn);
    OrderDescription.Controls.Add(lbd);

    if (Orders.ContainsKey(imageId))
    {
        Orders[imageId]++;
    }
    else
    {
        Orders.Add(imageId, 1);
    }
    foreach (var order in Orders)
    {
        imgdrinkcopy = new ImageButton();
        imgdrinkcopy.Width = 40;//πλάτος εικόνας
        imgdrinkcopy.Height = 40;//ύψος εικόνας
        //κάθε φορά τραβάει το url για να γεμίσει με την εικόνα το imagebutton, η
εικόνα είναι αυτή που τραβιέτε κάθε φορά
        imgdrinkcopy.ImageUrl = "Order_Element_LoadImage.aspx?ImageId=" + order.Key;
        OrderPanelFood.Controls.Add(imgdrinkcopy);
        countdr = new Label();
        countdr.Text = order.Value.ToString();
        OrderPanelFood.Controls.Add(countdr);
    }
}
}

```

```

int countsalad = 0;
//Τι συμβαίνει με το πάτημα μίας εικόνας απο τις σαλάτες.!
void imgsalad_Click(object sender, ImageClickEventArgs e)
{
    ImageButton imgsalad = sender as ImageButton;
    imgsalad.BorderColor = System.Drawing.Color.Black;
    imgsalad.BorderStyle = BorderStyle.Double;
    var imageId = imgsalad.ID; // στη μεταβλητή imageId κρατά το ID του κουμπιού που
πατήθηκε
    SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
    SqlCommand cmdsalad;
    sc.Open();
    cmdsalad = new SqlCommand("Select Id_Elements,Name,Description,Image FROM Element
WHERE categories = 'salad' AND Id_Elements =" + imageId, sc);

    countsalad++;
    Label countsd;

    SqlDataAdapter adaptersalad = new SqlDataAdapter(cmdsalad);
    DataSet dssalad = new DataSet();
    adaptersalad.Fill(dssalad, "Element");
    DataTable dtsalad = dssalad.Tables["Element"];
    DataRow row = dtsalad.Rows[0];//στη row ρίχνω την γραμμή j του πίνακα
//στον πίνακα data της γραμμής 0 ρίχνω την στήλη Image όπου είναι η εικόνα
byte[] data = (byte[])dtsalad.Rows[0]["Image"];
ImageButton imgsaladcopy;

    SqlDataReader reader = cmdsalad.ExecuteReader();
    reader.Read();
    Label lbn = new Label();
    Label lbd = new Label();
    lbn.Text = reader["Name"].ToString() + "<br/>";
    lbd.Text = reader["Description"].ToString() + "<br/>";
    OrderName.Controls.Add(lbn);
    OrderDescription.Controls.Add(lbd);

    if (Orders.ContainsKey(imageId))
    {
        Orders[imageId]++;
    }
    else
    {
        Orders.Add(imageId, 1);
    }
    foreach (var order in Orders)
    {
        imgsaladcopy = new ImageButton();
        imgsaladcopy.Width = 40;//πλάτος εικόνας
        imgsaladcopy.Height = 40;//ύψος εικόνας
        //κάθε φορά τραβάει το url για να γεμίσει με την εικόνα το imagebutton, η
εικόνα είναι αυτή που τραβιέτε κάθε φορά
        imgsaladcopy.ImageUrl = "Order_Element_LoadImage.aspx?ImageId=" + order.Key;
        OrderPanelFood.Controls.Add(imgsaladcopy);
        countsd = new Label();
        countsd.Text = order.Value.ToString();
        OrderPanelFood.Controls.Add(countsd);
    }
}

int countmainfood = 0;
//Τι συμβαίνει με το πάτημα μίας εικόνας απο τα κυρίως φαγητά!
void imgmainfood_Click(object sender, ImageClickEventArgs e)
{

```

```

        ImageButton imgmainfood = sender as ImageButton;
        imgmainfood.BorderColor = System.Drawing.Color.Black;
        imgmainfood.BorderStyle = BorderStyle.Double;
        var imageId = imgmainfood.ID; // στη μεταβλητή imageId κρατά το ID του κουμπιού
        που πατήθηκε
        SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
        SqlCommand cmdmainfood;
        sc.Open();
        cmdmainfood = new SqlCommand("Select Id_Elements,Name,Description,Image FROM
Element WHERE categories = 'main_food' AND Id_Elements =" + imageId, sc);

        countmainfood++;
        Label countmf;

        SqlDataAdapter adaptermainfood = new SqlDataAdapter(cmdmainfood);
        DataSet dsmainfood = new DataSet();
        adaptermainfood.Fill(dsmainfood, "Element");
        DataTable dtmainfood = dsmainfood.Tables["Element"];
        DataRow row = dtmainfood.Rows[0]; //στη row ρίχνω την γραμμή j του πίνακα
        //στον πίνακα data της γραμμής 0 ρίχνω την στήλη Image όπου είναι η εικόνα
        byte[] data = (byte[])dtmainfood.Rows[0]["Image"];
        ImageButton imgmainfoodcopy;

        SqlDataReader reader = cmdmainfood.ExecuteReader();
        reader.Read();
        Label lbn = new Label();
        Label lbd = new Label();
        lbn.Text = reader["Name"].ToString() + "<br/>";
        lbd.Text = reader["Description"].ToString() + "<br/>";
        OrderName.Controls.Add(lbn);
        OrderDescription.Controls.Add(lbd);

        if (Orders.ContainsKey(imageId))
        {
            Orders[imageId]++;
        }
        else
        {
            Orders.Add(imageId, 1);
        }
        foreach (var order in Orders)
        {
            imgmainfoodcopy = new ImageButton();
            imgmainfoodcopy.Width = 40; //πλάτος εικόνας
            imgmainfoodcopy.Height = 40; //ύψος εικόνας
            //κάθε φορά τραβάει το url για να γεμίσει με την εικόνα το imagebutton, η
            εικόνα είναι αυτή που τραβιέτε κάθε φορά
            imgmainfoodcopy.ImageUrl = "Order_Element_LoadImage.aspx?ImageId=" +
order.Key;
            OrderPanelFood.Controls.Add(imgmainfoodcopy);
            countmf = new Label();
            countmf.Text = order.Value.ToString();
            OrderPanelFood.Controls.Add(countmf);
        }
    }

    int countswiss = 0;
    //Τι συμβαίνει με το πάτημα μίας εικόνας απο τα γλυκά!
    void imgswiss_Click(object sender, ImageClickEventArgs e)
    {
        ImageButton imgswiss = sender as ImageButton;
        imgswiss.BorderColor = System.Drawing.Color.Black;
        imgswiss.BorderStyle = BorderStyle.Double;
    }

```



```

        var imageId = imgswiss.ID; // στη μεταβλητή imageId κρατά το ID του κουμπιού που
πατήθηκε
        SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
        SqlCommand cmdswiss;
        sc.Open();
        cmdswiss = new SqlCommand("Select Id_Elements,Name,Description,Image FROM Element
WHERE categories = 'swiss' AND Id_Elements =" + imageId, sc);

        countswiss++;
        Label countsw;

        SqlDataAdapter adapterswiss = new SqlDataAdapter(cmdswiss);
        DataSet dsswiss = new DataSet();
        adapterswiss.Fill(dsswiss, "Element");
        DataTable dtswiss = dsswiss.Tables["Element"];
        DataRow row = dtswiss.Rows[0]; //στη row ρίχνω την γραμμή j του πίνακα
//στον πίνακα data της γραμμής 0 ρίχνω την στήλη Image όπου είναι η εικόνα
        byte[] data = (byte[])dtswiss.Rows[0]["Image"];
        ImageButton imgswisscopy;

        SqlDataReader reader = cmdswiss.ExecuteReader();
        reader.Read();
        Label lbn = new Label();
        Label lbd = new Label();
        lbn.Text = reader["Name"].ToString() + "<br/>";
        lbd.Text = reader["Description"].ToString() + "<br/>";
        OrderName.Controls.Add(lbn);
        OrderDescription.Controls.Add(lbd);

        if (Orders.ContainsKey(imageId))
        {
            Orders[imageId]++;
        }
        else
        {
            Orders.Add(imageId, 1);
        }
        foreach (var order in Orders)
        {
            imgswisscopy = new ImageButton();
            imgswisscopy.Width = 40; //πλάτος εικόνας
            imgswisscopy.Height = 40; //ύψος εικόνας
            //κάθε φορά τραβάει το url για να γεμίσει με την εικόνα το imagebutton, η
            εικόνα είναι αυτή που τραβιέτε κάθε φορά
            imgswisscopy.ImageUrl = "Order_Element_LoadImage.aspx?ImageId=" + order.Key;
            OrderPanelFood.Controls.Add(imgswisscopy);
            countsw = new Label();
            countsw.Text = order.Value.ToString();
            OrderPanelFood.Controls.Add(countsw);
        }
    }
    //τι συμβαίνει με το πάτημα του create
    protected void CreateButton_Click(object sender, EventArgs e)
    { //ενεργοποιεί τα κουμπιά και τον panel
        Btntent.Enabled = true;
        Btncam.Enabled = true;
        Btnundu.Enabled = true;
        Btnredu.Enabled = true;
        Btndrink.Enabled = true;
        Btnclear.Enabled = true;
        OrderPanelFood.Enabled = true;
        //δημιουργεί ένα label
        Label lbl = new Label();
    }

```



```

        lbl.Text = "now you can chose your products".ToString();
        OrderPanelFood.Controls.Add(lbl);
        SweetsPanel.Enabled = true;
        SaladsPanel.Enabled = true;
        FoodsPanel.Enabled = true;
        DrinkPanel.Enabled = true;
        OrderPanelFood.BorderColor = System.Drawing.Color.Green;
        OrderPanelFood.BorderStyle = BorderStyle.Dotted;
        OrderPanelFood.BorderWidth = Unit.Pixel(1);
        Name.ForeColor = System.Drawing.Color.Azure;
        Description.ForeColor = System.Drawing.Color.Azure;
    }

    //με το πάτημα το sent ανοίγει το url της κουζίνας
    protected void Btnsent_Click(object sender, EventArgs e)
    {
        //string Fullurl = "Manager.aspx/";
        //OpenNewBrowserWindow(Fullurl, this);
        SqlConnection con = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
        //SqlCommand cmd;
        foreach(var pair in Orders)
        using (SqlCommand cmd = new SqlCommand("INSERT INTO Order (Id_Elements,
Num_Elements) VALUES (@word, @freq)", con))
        {
            con.Open();
            // for (int i = 0; i < Orders.Count; i++)
            // {

            cmd.Parameters.Add(new SqlParameter("@word", pair.Keys);
            cmd.Parameters.Add(new SqlParameter("@freq", pair.Value);
            cmd.ExecuteNonQuery();
            // }
            //con.Close();
        }
    }

    //Αυτή είναι η μέθοδος η οποία χρησιμοποιήτε για να ανοίξει μια νέα σελίδα στον broser
    με το πάτημα του Sent
    public static void OpenNewBrowserWindow(string Url, Control control)
    {
        ScriptManager.RegisterStartupScript(control, control.GetType(), "Open",
"window.open('" + Url + "');", true);
    }
    //παίζει το flash video
    protected void Btncam_Click(object sender, ImageClickEventArgs e)
    {
        OrderPanelFood.BackColor = Color.Chocolate;
        var onoff = 1;
        if (onoff == 0)
        {
            FlashVideo.Enabled = true;
            FlashVideo.EnableViewState = true;
            onoff = 1;
        }
        else if (onoff == 1)
        {
            FlashVideo.Enabled = false;
            FlashVideo.EnableViewState = false;
            onoff = 0;
        }
    }
}

```

```

protected void Btnundu_Click(object sender, ImageClickEventArgs e)
{
    OrderPanelFood.BackColor = Color.Beige;
}

protected void Btnredu_Click(object sender, ImageClickEventArgs e)
{
    OrderPanelFood.BackColor = Color.Bisque;
}

//το κουμπί με τα ποτά κάνει τις δικές του ενέργειες
Boolean bln = true;
protected void Btndrink_Click(object sender, ImageClickEventArgs e)
{
    //OrderPanelFood.BackColor = Color.BlueViolet;

    if (bln)
        TrueDrinks();
    else
        FalseDrinks();
    bln = !bln;
}

private void TrueDrinks()
{
}

private void FalseDrinks()
{
}

//καθερίζει την παραγγελία
protected void Btnclear_Click(object sender, EventArgs e)
{
    OrderPanelFood.Controls.Clear();
    Orders = new Dictionary<string, int>();
    Orders = null;
}

//σβίνει πιάτα
protected void DeleteButton_Click(object sender, EventArgs e)
{
    OrderPanelFood.Controls.Clear();
}

//κλείνει το πρόγραμμα
protected void CloseButton_Click(object sender, EventArgs e)
{
    ClientScript.RegisterClientScriptBlock(Page.GetType(), "script",
"window.close();", true);
}

protected void ShowOrderButton_Click(object sender, EventArgs e)
{
    //OrderPanelFood.BackColor = Color.Bisque;
    Label Lbl = new Label();
    Lbl.Text = "You can pay here";
    OrderPanelFood.Controls.Add(Lbl);
}
}
}

```

## Order\_Element\_LoadImage.aspx

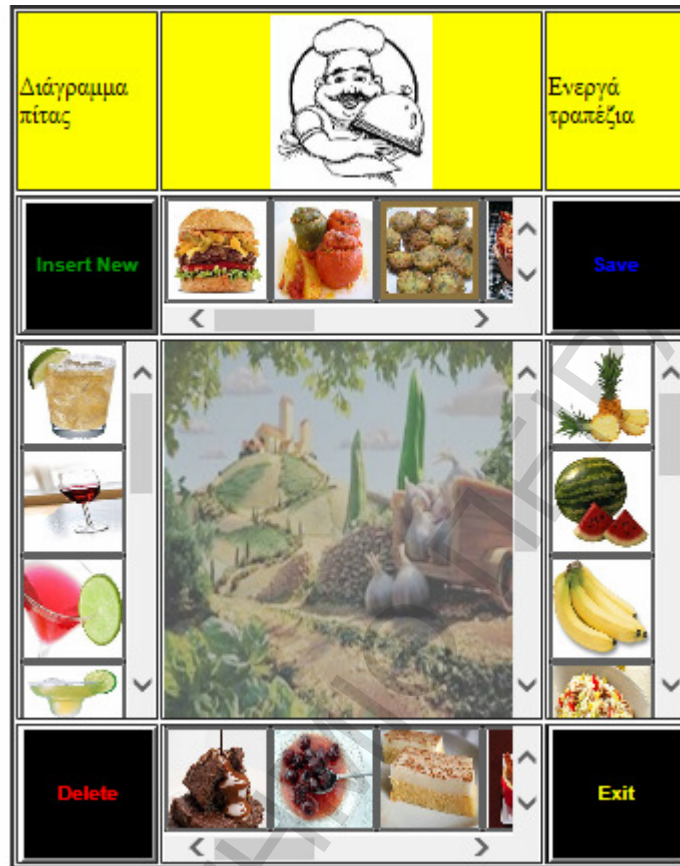
Αυτή η κλάση χρησιμεύει για να γεμίζουν τα κουμπιά με τις εικόνες από την βάση.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace monopoly
{
    public partial class Element_Drink : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Request.QueryString["ImageID"] != null)
            {
                SqlConnection sc = new
                SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
                SqlCommand cmd;
                DataSet ds = new DataSet();
                try
                {
                    //η σύνδεση άνοιξε
                    sc.Open();
                    //επιλογή φωτογραφίας
                    cmd = new SqlCommand("Select Id_Elements,Image FROM Element WHERE
                    Id_Elements=" + Request.QueryString["ImageID"], sc);
                    //εκτέλεση ερωτήματος
                    SqlDataAdapter adapter = new SqlDataAdapter(cmd);
                    adapter.Fill(ds, "Element");
                    DataTable dt = ds.Tables["Element"];
                    int i = dt.Rows.Count;
                    //όσο ο πίνακας έχει γραμμές i
                    for (int j = 0; j < i; j++)
                    {
                        DataRow row = dt.Rows[j];
                        byte[] data = (byte[])dt.Rows[j]["Image"];
                        Response.BinaryWrite(data);
                        Response.Flush();
                        Response.End();
                    }
                }
                catch (Exception)
                {
                }
                finally
                {
                    sc.Close();
                }
            }
        }
    }
}
```

## Περιβάλλον διαχειριστή Owner Table

Owner interface



Μετά το πάτημα του delete ενεργοποιούνται τα τέσσερα, panel τώρα για το κάθε ένα με το πάτημα κάθε εικόνας διαγράφετε από την βάση. Επίσης έχει εμφανιστεί ένα κουμπί lock το οποίο το πατάμε για να ξανακλειδώσουν τα panel. Σημειώνεται ότι όταν φορτώνει η εφαρμογή οι τα panel είναι κλειδωμένα, δεν μπορεί ο διαχειριστής να διαγράψει κάτι παρα μόνο με το πάτημα του delete, ακόμα με την επιλογή Dell all μπορούν να διαγραφούν όλες οι εγγραφές από τη βάση.



```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="OwnerTable.aspx.cs"
Inherits="monopoly.WebForm2" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head id="Head1" runat="server">
```

```
<title></title>
```

```
<style type="text/css">
```

```
.style1
```

```
{
```

```
width: 236px;
```

```
}
```

```
.style2
```

```
{
```

```
color: #0000FF;
```

```
}
```

```
</style>
```

```
<script type="text/javascript">
```

```
function RefreshPage() {
```

```
window.location.reload()
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<form id="form2" runat="server">
```

```
<table align="center" style="width: 445px; height: 403px" border="2">
```

```
<tr>
```

```
<td bgcolor="#CCFF66">Διάγραμμα πίτας</td>
```

```
<td align="center" bgcolor="#CCFF66">
```

```
<asp:Image ID="Image1" runat="server"
```

```
ImageUrl="~/imageis/sef.png" />
```

```
</td>
```

```
<td bgcolor="#CCFF66"> Ενεργά τραπέζια</td>
```

```
</tr>
```

```
<tr>
```

```
<td align="left">
```

```
<asp:Button ID="InserNewButton" CssClass="button"
```

```
runat="server"
```

```
Text="Insert New" BackColor="Black" ForeColor="#009900"
```

```
Font-Bold="True"
```

```
Height="90px" Width="90px" onclick="InserNewButton_Click"
```

```
/>
```

```
</td>
```

```
<td class="style1">
```

```
<asp:Panel ID="FoodsPanel" runat="server"
```

```
ScrollBars="Horizontal" Width="250px"
```

```
Height="90px" Wrap="False" Enabled="False">
```

```
</asp:Panel>
```

```
</td>
```

```
<td align="right">
```

```
<asp:Button ID="SaveButton" CssClass="button" runat="server"
```

```
Text="Save"
```

```
BackColor="Black" Font-Bold="True" ForeColor="Blue"
```

```
Height="90px"
```

```
Width="90px" onclick="SaveButton_Click" />
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td>
```

```

        <asp:Panel ID="DrinkPanel" runat="server" Height="250px"
ScrollBars="Vertical"
        Width="90px" Enabled="False" >
        </asp:Panel>
    </td>
    <td class="style1">
        <asp:Panel ID="Tablle" runat="server" Height="250px"
HorizontalAlign="Center" Width="250px" BackColor="#CCFF66"
ScrollBars="Vertical">
            <asp:Button ID="btnlock" runat="server"
onclick="lock_Click" Text="lock"
                Visible="False" />
            <br />
            <asp:Label ID="LblNameOfProduct" runat="server"
                style="text-align: left" Text="Name Of Product"
Visible="False"></asp:Label>
            <br />
            <asp:TextBox ID="TxbNameOfProduct" runat="server"
Visible="False" Width="194px">give the name of product</asp:TextBox>
            <br />
            <br />
            <asp:Label ID="LblDiscription" runat="server"
Text="Discription Of Product"
                Visible="False"></asp:Label>
            <br />
            <asp:TextBox id="TxbDiscription" rows="5"
TextMode="multiline" runat="server"
                Visible="False" Width="206px" >give the discription of
product</asp:TextBox>
            <br />
            <br />
            <asp:Label ID="LblImage" runat="server" Text="Upload an
Image" Visible="False"></asp:Label>
            <br />
            <asp:FileUpload ID="FiUpImage" runat="server"
Visible="False" Width="208px" />
            <br />
            <br />
            <asp:Label ID="LblCategory" runat="server" Text="Chose the
Category "
                Visible="False"></asp:Label>
            <br />
            <asp:DropDownList ID="DDLCategories" runat="server"
style="margin-left: 0px"
                Visible="False">
                <asp:ListItem>main_food</asp:ListItem>
                <asp:ListItem Value="salad">salad</asp:ListItem>
                <asp:ListItem>drink</asp:ListItem>
                <asp:ListItem>swiss</asp:ListItem>
            </asp:DropDownList>
            <br />
            <br />
            <asp:Label ID="LblPrice" runat="server" Text="Price Of
Product" Visible="False"></asp:Label>
            <br />
            <asp:TextBox ID="TxbPrice" runat="server" Visible="False"
Width="79px">give the price of product </asp:TextBox>
            <br />
            <br />
            <span class="style2"><strong>change color style<br />
change background video<br />
            change title and logo<br /> change tax<br /> change map
for tables<br />
            <br />

```


```

                </strong></span>
                <br />
            </asp:Panel>
        </td>
        <td>
            <asp:Panel ID="SaladsPanel" runat="server" Height="250px"
ScrollBars="Vertical"
                Width="90px" Enabled="False">
            </asp:Panel>
        </td>
    </tr>
    <tr>
        <td align="left">
            <asp:Button ID="DeleteOwnerButton" CssClass="button"
runat="server" Text="Delete"
                BackColor="Black" Font-Bold="True" ForeColor="Red"
Height="90px"
                Width="90px" onclick="DeleteOwnerButton_Click" />
        </td>
        <td class="style1">
            <asp:Panel ID="SweetsPanel" runat="server" Height="90px"
Enabled="False">
                ScrollBars="Horizontal" Width="250px" Wrap="False"
            </asp:Panel>
        </td>
        <td align="right" bgcolor="Black">
            <asp:Button ID="ExitButton" CssClass="button" runat="server"
Text="Exit"
                BackColor="Black" Font-Bold="True" ForeColor="Yellow"
Height="90px"
                Width="90px" onclick="ExitButton_Click"/>
        </td>
    </tr>
</table>
</form>
</body>
</html>

```



## Λογικό μέρος

►  OwnerTable.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;
using System.IO;
using System.Data;
using System.Drawing;

namespace monopoly
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            // Σύνδεση με την βάση
            SqlConnection sc = new
            SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
            //Μεταβλητές για ερώτηση
            SqlCommand cmdrink;
            SqlCommand cmsalad;
            SqlCommand cmmainfood;
            SqlCommand cmswiss;
            DataSet dsdrink = new DataSet();
            DataSet dssalad = new DataSet();
            DataSet dsmainfood = new DataSet();
            DataSet dsswiss = new DataSet();
            try
            {
                //η σύνδεση άνοιξε
                sc.Open();
                //4 ερωτήματα πέρνουν απο την βάση τα στοιχεία για τα elements
                cmdrink = new SqlCommand("Select
                Id_Elements,Name,Description,Image,Price,categories FROM Element WHERE categories = 'drink'",
                sc);
                cmsalad = new SqlCommand("Select
                Id_Elements,Name,Description,Image,Price,categories FROM Element WHERE categories = 'salad'",
                sc);
                cmmainfood = new SqlCommand("Select
                Id_Elements,Name,Description,Image,Price,categories FROM Element WHERE categories =
                'main_food'", sc);
                cmswiss = new SqlCommand("Select
                Id_Elements,Name,Description,Image,Price,categories FROM Element WHERE categories = 'swiss'",
                sc);
                //Εισαγωγή δεδομένων σε adapters
                SqlDataAdapter adapterdrink = new SqlDataAdapter(cmdrink);
                SqlDataAdapter adaptersalad = new SqlDataAdapter(cmsalad);
                SqlDataAdapter adaptermainfood = new SqlDataAdapter(cmmainfood);
                SqlDataAdapter adapterswiss = new SqlDataAdapter(cmswiss);

                adapterdrink.Fill(dsdrink, "Element");
                adaptersalad.Fill(dssalad, "Element");
                adaptermainfood.Fill(dsmainfood, "Element");
                adapterswiss.Fill(dsswiss, "Element");
            }
            catch { }
        }
    }
}
```

```

//Τα δεδομένα μπαίνουν σε πίνακες
DataTable dtdrink = dsdrink.Tables["Element"];
DataTable dtsalad = dssalad.Tables["Element"];
DataTable dtmainfood = dsmainfood.Tables["Element"];
DataTable dtswiss = dsswiss.Tables["Element"];

//Μεταβλητές που κρατούν τις γραμμές των πινάκων
int countdrink = dtdrink.Rows.Count;
int countsalad = dtsalad.Rows.Count;
int countmainfood = dtmainfood.Rows.Count;
int countswiss = dtswiss.Rows.Count;

//Τρέχουν 4 for για να διαβάζονται τα στοιχεία για κάθε πίνακα ανα γραμμή
for (int j = 0; j < countdrink; j++)
{
    DataRow row = dtdrink.Rows[j];
    byte[] data = (byte[])dtdrink.Rows[j]["Image"];
    ImageButton imgdrink = new ImageButton();
    imgdrink.Width = 65;
    imgdrink.Height = 65;
    //περνά ένα ImageId το οποίο είναι το Id του κάθε Element για κάθε γραμμή
    imgdrink.ImageUrl = "Owner_Element_Drink.aspx?ImageId=" +
row["Id_Elements"];
    //
    imgdrink.ID = dtdrink.Rows[j]["Id_Elements"].ToString();
    imgdrink.Click += new ImageClickEventHandler(imgdrink_Click);
    imgdrink.BorderColor = System.Drawing.Color.Black;
    imgdrink.BorderStyle = BorderStyle.Double;
    DrinkPanel.Controls.Add(imgdrink);
}
CheckBox chdrink = new CheckBox();
chdrink.Text = "Dell all";
DrinkPanel.Controls.Add(chdrink);

for (int j = 0; j < countsalad; j++)
{
    DataRow row = dtsalad.Rows[j];
    byte[] data = (byte[])dtsalad.Rows[j]["Image"];
    ImageButton imgsalad = new ImageButton();
    imgsalad.Width = 65;
    imgsalad.Height = 65;
    //περνά ένα ImageId το οποίο είναι το Id του κάθε Element για κάθε γραμμή
    imgsalad.ImageUrl = "Owner_Element_Salad.aspx?ImageId=" +
row["Id_Elements"];
    imgsalad.ID = dtsalad.Rows[j]["Id_Elements"].ToString();
    imgsalad.Click += new ImageClickEventHandler(imgsalad_Click);
    imgsalad.BorderColor = System.Drawing.Color.Black;
    imgsalad.BorderStyle = BorderStyle.Double;
    SaladsPanel.Controls.Add(imgsalad);
}
CheckBox chsalad = new CheckBox();
chsalad.Text = "Dell all";
SaladsPanel.Controls.Add(chsalad);

for (int j = 0; j < countmainfood; j++)
{
    DataRow row = dtmainfood.Rows[j];
    byte[] data = (byte[])dtmainfood.Rows[j]["Image"];
    ImageButton imgmainfood = new ImageButton();
    imgmainfood.Width = 65;
    imgmainfood.Height = 65;

```

```

        //περνά ένα ImageId το οποίο είναι το Id του κάθε Element για κάθε γραμμή
        imgmainfood.ImageUrl = "Owner_Element_MainFood.aspx?ImageId=" +
row["Id_Elements"];
        imgmainfood.ID = dtmainfood.Rows[j]["Id_Elements"].ToString();
        imgmainfood.Click += new ImageClickEventHandler(imgmainfood_Click);
        imgmainfood.BorderColor = System.Drawing.Color.Black;
        imgmainfood.BorderStyle = BorderStyle.Double;
        FoodsPanel.Controls.Add(imgmainfood);

    }
    CheckBox chmainfood = new CheckBox();
    chmainfood.Text = "Dell all";
    FoodsPanel.Controls.Add(chmainfood);

    for (int j = 0; j < countswiss; j++)
    {
        DataRow row = dtswiss.Rows[j];
        byte[] data = (byte[])dtswiss.Rows[j]["Image"];
        ImageButton imgswiss = new ImageButton();
        imgswiss.Width = 65;
        imgswiss.Height = 65;
        //περνά ένα ImageId το οποίο είναι το Id του κάθε Element για κάθε γραμμή
row["Id_Elements"];
        imgswiss.ID = dtswiss.Rows[j]["Id_Elements"].ToString();
        imgswiss.Click += new ImageClickEventHandler(imgswiss_Click);
        imgswiss.BorderColor = System.Drawing.Color.Black;
        imgswiss.BorderStyle = BorderStyle.Double;
        SweetsPanel.Controls.Add(imgswiss);

    }
    CheckBox chswiss = new CheckBox();
    chswiss.Text = "Dell all";
    SweetsPanel.Controls.Add(chswiss);
}

catch (Exception)
{
}

finally
{
    sc.Close();
}
}

//ΤΑ EVENT ΜΕ ΤΟ ΠΑΤΗΜΑ ΤΩΝ ΕΙΚΟΝΩΝ
//Τι συμβαίνει με το πάτημα μίας εικόνας απο τα ποτά.!
void imgdrink_Click(object sender, ImageClickEventArgs e)
{
    ImageButton imgdrink = sender as ImageButton;
    var imageId = imgdrink.ID;
    SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
    SqlCommand cmdrink;
    sc.Open();
    cmdrink = new SqlCommand("Delete FROM Element WHERE categories = 'drink' AND
Id_Elements =" + imageId, sc);
    cmdrink.ExecuteNonQuery();
    sc.Close();
}

//Τι συμβαίνει με το πάτημα μίας εικόνας απο τις σαλάτες.!
void imgsalad_Click(object sender, ImageClickEventArgs e)

```

```

    {
        ImageButton imgsalad = sender as ImageButton;
        var imageId = imgsalad.ID;
        SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
        SqlCommand cmdsalad;
        sc.Open();
        cmdsalad = new SqlCommand("Delete FROM Element WHERE categories = 'salad' AND
Id_Elements =" + imageId, sc);
        cmdsalad.ExecuteNonQuery();
        sc.Close();
    }
    //Τι συμβαίνει με το πάτημα μίας εικόνας απο τα φαγητά.!
    void imgmainfood_Click(object sender, ImageClickEventArgs e)
    {
        ImageButton imgmainfood = sender as ImageButton;
        var imageId = imgmainfood.ID;
        SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
        SqlCommand cmdmainfood;
        sc.Open();
        cmdmainfood = new SqlCommand("Delete FROM Element WHERE categories = 'main_food'
AND Id_Elements =" + imageId, sc);
        cmdmainfood.ExecuteNonQuery();
        sc.Close();
    }
    //Τι συμβαίνει με το πάτημα μίας εικόνας απο τα γλυκά.!
    void imgswiss_Click(object sender, ImageClickEventArgs e)
    {
        ImageButton imgswiss = sender as ImageButton;
        var imageId = imgswiss.ID;
        SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
        SqlCommand cmdswiss;
        sc.Open();
        cmdswiss = new SqlCommand("Delete FROM Element WHERE categories = 'swiss' AND
Id_Elements =" + imageId, sc);
        cmdswiss.ExecuteNonQuery();
        sc.Close();
    }
}

//με το πάτημα του Delete γίνεται έλεγχος και σβήνει τα επιλεγμένα
protected void DeleteOwnerButton_Click(object sender, EventArgs e)
{
    btnlock.Visible = true;
    DrinkPanel.Enabled = true;
    DrinkPanel.BorderWidth = 3;
    DrinkPanel.BorderColor = Color.Red;
    SaladsPanel.Enabled = true;
    SaladsPanel.BorderWidth = 3;
    SaladsPanel.BorderColor = Color.Red;
    SweetsPanel.Enabled = true;
    SweetsPanel.BorderWidth = 3;
    SweetsPanel.BorderColor = Color.Red;
    FoodsPanel.Enabled = true;
    FoodsPanel.BorderWidth = 3;
    FoodsPanel.BorderColor = Color.Red;

    foreach (Control chdrink in DrinkPanel.Controls)
    {
        if ((chdrink is CheckBox) && ((CheckBox)chdrink).Checked)
            deletedrink();
    }
    foreach (Control chmainfood in FoodsPanel.Controls)

```

```

    {
        if ((chmainfood is CheckBox) && ((CheckBox)chmainfood).Checked)
            deletemainfood();
    }
    foreach (Control chsalad in SaladsPanel.Controls)
    {
        if ((chsalad is CheckBox) && ((CheckBox)chsalad).Checked)
            deletesalad();
    }
    foreach (Control chswiss in SweetsPanel.Controls)
    {
        if ((chswiss is CheckBox) && ((CheckBox)chswiss).Checked)
            deleteswiss();
    }
}

// η μέθοδος που καλείται με το πάτημα του delete και σβείνει τα ποτά που έχουν
επιλεγεί
protected void deletedrink()
{
    SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
    SqlCommand cmdrink;
    sc.Open();
    cmdrink = new SqlCommand("Delete FROM Element WHERE categories = 'drink'", sc);
    cmdrink.ExecuteNonQuery();
    sc.Close();
}

// η μέθοδος που καλείται με το πάτημα του delete και σβείνει τα κυρίως φαγητά που
έχουν επιλεγεί
protected void deletemainfood()
{
    SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
    SqlCommand cmmainfood;
    sc.Open();
    cmmainfood = new SqlCommand("Delete FROM Element WHERE categories = 'main_food'",
sc);
    cmmainfood.ExecuteNonQuery();
    sc.Close();
}

// η μέθοδος που καλείται με το πάτημα του delete και σβείνει σαλάτες που έχουν
επιλεγεί
protected void deletesalad()
{
    SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
    SqlCommand cmsalad;
    sc.Open();
    cmsalad = new SqlCommand("Delete FROM Element WHERE categories = 'salad'", sc);
    cmsalad.ExecuteNonQuery();
    sc.Close();
}

// η μέθοδος που καλείται με το πάτημα του delete και σβείνει τα γλυκά που έχουν
επιλεγεί
protected void deleteswiss()
{
    SqlConnection sc = new
SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
    SqlCommand cmswiss;
    sc.Open();
}

```

```

        cmswiss = new SqlCommand("Delete FROM Element WHERE categories = 'swiss'", sc);
        cmswiss.ExecuteNonQuery();
        sc.Close();
    }
    // με το πάτημα του lock
    protected void lock_Click(object sender, EventArgs e)
    {
        DrinkPanel.BorderWidth = 0;
        DrinkPanel.Enabled = false;
        SaladsPanel.BorderWidth = 0;
        SaladsPanel.Enabled = false;
        FoodsPanel.BorderWidth = 0;
        FoodsPanel.Enabled = false;
        SweetsPanel.BorderWidth = 0;
        SweetsPanel.Enabled = false;
        btnlock.Visible = false;
    }

    protected void InserNewButton_Click(object sender, EventArgs e)
    {
        LblCategory.Visible = true;
        TxbNameOfProduct.Visible = true;
        LblDiscription.Visible = true;
        TxbDiscription.Visible = true;
        LblImage.Visible = true;
        FiUpImage.Visible = true;
        LblNameOfProduct.Visible = true;
        DDLCategories.Visible = true;
        LblPrice.Visible = true;
        TxbPrice.Visible = true;
    }

    protected void ExitButton_Click(object sender, EventArgs e)
    {
        ClientScript.RegisterClientScriptBlock(Page.GetType(), "script", "window.close();",
true);
    }

    protected void SaveButton_Click(object sender, EventArgs e)
    {
        SqlConnection sc = new SqlConnection
("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
        SqlCommand cmd;

        if (FiUpImage.HasFile)
        {
            int length = FiUpImage.PostedFile.ContentLength;
            byte[] pic = new byte[length];
            FiUpImage.PostedFile.InputStream.Read(pic, 0, length);
            try
            {
                sc.Open();
                cmd = new SqlCommand("Insert into
Element(Name,Description,Image,Price,categories)
values(@Name,@Discription,@Image,@Price,@Category)", sc);

                cmd.Parameters.AddWithValue("@Name", TxbNameOfProduct.Text);
                cmd.Parameters.AddWithValue("@Discription", TxbDiscription.Text);
                cmd.Parameters.AddWithValue("@Image", pic);
                cmd.Parameters.AddWithValue("@Price", TxbPrice.Text);
            }
        }
    }

```

```
        cmd.Parameters.AddWithValue("@Category", DDLCategories.Text);
        cmd.ExecuteNonQuery();
        sc.Close();
        TxbNameOfProduct.Text = "";
        TxbDiscription.Text = "";
        TxbPrice.Text = "";
    }
    catch (Exception)
    {
    }
    finally
    {
        sc.Close();
    }
}
}
}
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

#### Owner\_Element\_LoadImage.aspx



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace monopoly
{
    public partial class Owner_Element_Drink : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Request.QueryString["ImageID"] != null)
            {
                SqlConnection sc = new
                SqlConnection("Server=cleverpc;database=monopoly;Trusted_Connection=True;");
                SqlCommand cmd;
                DataSet ds = new DataSet();
                try
                {
                    //η σύνδεση άνοιξε
                    sc.Open();
                    //επιλογή φωτογραφίας
                    cmd = new SqlCommand("Select Id_Elements,Image FROM Element WHERE
                    Id_Elements=" + Request.QueryString["ImageID"], sc);
                    //εκτέλεση ερωτήματος
                    SqlDataAdapter adapter = new SqlDataAdapter(cmd);
                    adapter.Fill(ds, "Element");
                    DataTable dt = ds.Tables["Element"];
                    int i = dt.Rows.Count;
                    //όσο ο πίνακας έχει γραμμές i
                    for (int j = 0; j < i; j++)
                    {
                        DataRow row = dt.Rows[j];
                        byte[] data = (byte[])dt.Rows[j]["Image"];
                        Response.BinaryWrite(data);
                        Response.Flush();
                        Response.End();
                    }
                }
                catch (Exception)
                {
                }
                finally
                {
                    sc.Close();
                }
            }
        }
    }
}
```





## Περιβάλλον manager



### Manager interface

Χάρτης για ενεργά τραπέζια				Διάγραμμα πίτας
Αριθμός Τραπεζιού	Παραγγελία	Κατάσταση	Αποδοχή Παραγγελίας	Διάρκεια Παραγγελείας
2:49:17 πμ 05/06/2014				
<input type="button" value="Sent"/> <input type="button" value="showOrder"/> <input type="button" value="okOrder"/>				
<input type="checkbox"/> Chiken				
<input type="checkbox"/> Potatos				
<input type="checkbox"/> Cheesecake				
Η παραγγελία στάλθηκε στις : sent				
Έγινε απόκριση από την κουζίνα στις : show				
Η παραγγελία είναι έτοιμη στις : okOrder				
				



Στον παραπάνω πίνακα δεν έχει πατηθεί ακόμα η εντολή sent για την αποστολή της παραγγελίας στην κουζίνα, και η λάμπα είναι άσπρη καθώς περιμένει την παραγγελία, υποθετικά το τραπέζι είναι κενό.

Χάρτης για ενεργά τραπέζια				Διάγραμμα πίτας
Αριθμός Τραπεζιού	Παραγγελία	Κατάσταση	Αποδοχή Παραγγελίας	Διάρκεια Παραγγελείας
2:50:45 πμ    05/06/2014				
<input type="button" value="Sent"/> <input type="button" value="showOrder"/> <input type="button" value="okOrder"/>				
<input type="checkbox"/> Chicken <input type="checkbox"/> Potatos <input type="checkbox"/> Cheesecake				
Η παραγγελία στάλθηκε στις : 2:50:43 πμ Έγινε απόκριση από την κουζίνα στις : show Η παραγγελία είναι έτοιμη στις : okOrder				
				

Μόλις πατηθεί το sent κρατείτε ο χρόνος που η παραγγελία στάλθηκε στην κουζίνα, καθώς και η λάμπα γίνεται πορτοκαλί, πλέον αναμένεται η απόκριση από την κουζίνα ότι είδε την παραγγελία και ξεκινά την ετοιμασία.


Χάρτης για ενεργά τραπέζια				Διάγραμμα πίτας
Αριθμός Τραπεζιού	Παραγγελία	Κατάσταση	Αποδοχή Παραγγελίας	Διάρκεια Παραγγελείας
2:51:25 πμ      05/06/2014				
<input type="button" value="Sent"/> <input type="button" value="showOrder"/> <input type="button" value="okOrder"/>				
<input type="checkbox"/> Chicken <input type="checkbox"/> Potatos <input type="checkbox"/> Cheesecake				
Η παραγγελία στάλθηκε στις : 2:50:43 πμ Έγινε απόκριση από την κουζίνα στις : 2:51:23 πμ Η παραγγελία είναι έτοιμη στις : okOrder				
				

Εδώ έχει γίνει η απόκριση από την κουζίνα και καταγράφεται ο χρόνος που έγινε η απόκριση καθώς και η λάμπα γίνεται πράσινη, δηλαδή έχει ξεκινήσει η ετοιμασία της.

Χάρτης για ενεργά τραπέζια				Διάγραμμα πίτας
Αριθμός Τραπεζιού	Παραγγελία	Κατάσταση	Αποδοχή Παραγγελίας	Διάρκεια Παραγγελείας
2:52:03 πμ    05/06/2014				
<input type="button" value="Sent"/> <input type="button" value="showOrder"/> <input type="button" value="okOrder"/>				
<input checked="" type="checkbox"/> Chicken <input checked="" type="checkbox"/> Potatos <input type="checkbox"/> Cheesecake				
Η παραγγελία στάλθηκε στις : 2:50:43 πμ Έγινε απόκριση από την κουζίνα στις : 2:51:23 πμ Η παραγγελία είναι έτοιμη στις : 2:51:51 πμ				
				

Η παραγγελία είναι έτοιμη και ο manager επιλέγει το κουμπί okOrder τότε αμέσως καταγράφεται ο χρόνος και η λάμπα γίνεται κόκκινη, που σημαίνει ότι όλλα έχουν τελειώσει.

## Λογικό μέρος

▶  Manager.aspx

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace monopoly
{
    public partial class Manager : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            date.Text = DateTime.Now.ToString("dd/MM/yyyy");
            time.Text = DateTime.Now.ToString("h:mm:ss tt");
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Timer1.Enabled = true;
            sent.Text = time.Text.ToString();
            showOrder.Enabled = true;
            okOrder.Enabled = true;
            Image2.ImageUrl = "~/imageis/orangelight.jpg";
        }

        protected void showOrder_Click(object sender, EventArgs e)
        {
            show.Text = time.Text.ToString();
            Image2.ImageUrl = "~/imageis/greenlight.jpg";
            OrderManager.Enabled = true;
        }

        protected void OrderManager_Click(object sender, EventArgs e)
        {
            okOrder.Text = time.Text.ToString();
            Image2.ImageUrl = "~/imageis/redlight.jpg";
        }
    }
}
```

## Βιβλιογραφία

### Ελληνόγλωσση

Κιόχος, Π., Παπανικολάου, Γ. (2001). *Προγραμματισμός Δράσεως Επιχειρήσεων*. Αθήνα: Εκδόσεις Σταμούλη.

Θεμιστοκλέους, Μ. και Μαντζάνα, Β. (2010). *Υπηρεσίες Παγκόσμιου Ιστού και Υπηρεσιοστρεφείς Αρχιτεκτονικές*. Αθήνα: Αυτοέκδοση.

Υψηλάντης, Π. (2001). *Πληροφοριακά Συστήματα Διοίκησης: Από τη Θεωρία στην Πράξη*. Αθήνα: Εκδόσεις Πατάκη.

Gaarder, J. (1994). *Ο κόσμος της Σοφίας Μυθιστόρημα για την ιστορία της φιλοσοφίας*. Αθήνα: Λιβάνη (458-479).

Δημητριάδης, Α. (2001). *Διοίκηση Διαχείριση Πληροφοριακών Συστημάτων*.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

### *Δικτυακοί χώροι*

[http://www.pi-schools.gr/programs/ktp/previous\\_version/book2/01\\_6.pdf](http://www.pi-schools.gr/programs/ktp/previous_version/book2/01_6.pdf)

[http://el.wikipedia.org/wiki/%CE%9C%CE%B5%CF%83%CE%BF%CE%B3%CE%B5%CE%B9%CE%B1%CE%BA%CE%AE\\_%CE%B4%CE%B9%CE%B1%CF%84%CF%81%CE%BF%CF%86%CE%AE](http://el.wikipedia.org/wiki/%CE%9C%CE%B5%CF%83%CE%BF%CE%B3%CE%B5%CE%B9%CE%B1%CE%BA%CE%AE_%CE%B4%CE%B9%CE%B1%CF%84%CF%81%CE%BF%CF%86%CE%AE)

<http://www.ert-archives.gr/V3/public/main/page-assetview.aspx?tid=0000069955&tsz=0&autostart=0>

<http://www0.dmst.aueb.gr/louridas/lectures/dais/process/process.html>

[http://ptgmedia.pearsoncmg.com/images/9780789738035/samplepages/0789738031\\_Sample.pdf](http://ptgmedia.pearsoncmg.com/images/9780789738035/samplepages/0789738031_Sample.pdf)

<http://alison.com/courses/Introduction-to-Cloud-Computing>

<http://www.youtube.com/watch?v=ys9cGe-EFOQ>

[http://x-integrate.de/x-in-cms.nsf/id/Error?open&errmsg=HTTP%20Web%20Server%3A%20IBM%20Notes%20Exception%20-%20Entry%20not%20found%20in%20index&hc=yF1DD\\_41EF5F6E9\\_1931E65E45E2FFDE&errseq=1](http://x-integrate.de/x-in-cms.nsf/id/Error?open&errmsg=HTTP%20Web%20Server%3A%20IBM%20Notes%20Exception%20-%20Entry%20not%20found%20in%20index&hc=yF1DD_41EF5F6E9_1931E65E45E2FFDE&errseq=1)

<http://www.cmswire.com/news/topic/cloud+computing>

[http://www.icsd.aegean.gr/website\\_files/proptyxiako/16828056.pdf](http://www.icsd.aegean.gr/website_files/proptyxiako/16828056.pdf)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## Ξενόγλωσση

Aaen, I. (2003). "Software Process Improvement: Blueprints versus Recipes" *IEEE Software*, 20(3):86–93.

Barry, W. Boehm (1988). "A spiral model of software development and enhancement" *IEEE Computer*, 21(5):61–72.

Watts, S. Humphrey (1989). *Managing the Software Process*. Addison-Wesley, Reading, MA, (247–285).

Mark C. Paulk, Curtis, B., Beth Chrissis, M. and Charles V. Weber (1993). "Capability maturity model, version 1.1." *IEEE Software*, 10(4):18–27.

Roger, S. Pressman (2000). *Software Engineering: A Practitioner's Approach*, McGraw-Hill Publishing Company, London, 5th edition. European Adaptation by Darrel Ince, (17–49).

Sommerville, I. (2001). *Software Engineering*. Addison-Wesley, Harlow, England, 6th edition, (42–70)

Katsigris, C., Thomas C. (2008). *Design and Equipment for Restaurants and Foodservice*. Canada: John Wiley & Sons.

Computer Information Systems and Industrial Management 11th IFIP TC 8 International Conference, CISIM 2012, Venice, Italy, September 26-28, 2012.

Birchfield, J. C., Birchfield, J. Jr. (2007). *Design and Layout of Foodservice Facilities*. 3rd ed., Hoboken NJ: John Wiley & Sons.

Payne-Palacio, J., Theis, M. (2011). *Foodservice Management Principles and Practices*. New York: Pearson Education (US)

Laudon, K. C., Laudon, J. P. (2011). *Management Information Systems - Managing the Digital Firm*, 12th ed., New Jersey: Pearson Prentice Hall.

Kenneth, C., Laudon, J. P. (2005). *Management Information Systems Managing the Digital Firm*. Pearson Prentice Hall.

The Software Requirements Memory Jogger A Pocket Guide to Help Software And Business Teams Develop And Manage Requirements Memory Jogger 2005 Michael A. Cusumano.

The Business of Software: What Every Manager, Programmer, and Entrepreneur Must Know to Thrive and Survive in Good Times and Bad Michael A. Cusumano.

Hasted, E. (2005). *Software That Sells: A Practical Guide to Developing and Marketing Your Software Project*. Indianapolis: Wiley.

Product Strategy for High Technology Companies Michael McGrath.

Next Generation Product Development : How to Increase Productivity, Cut Costs, and Reduce Cycle Times Michael McGrath.

Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust Ken Schwaber.



Software Requirements Karl E Wieggers Joy Beatty

Anatomy of a Business Plan: The Step-by-Step Guide to Building a Business and Securing Your Company's Future Linda Pinson

Systems Analysis and Design with UML An Object-Oriented Approach 3<sup>rd</sup> Edition Alan Dennis, Barbara Haley Wixom, David Tegarden

Cloud Computing For Dummies Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper (Apr 15, 2013)

The Cloud at Your Service Jothy Rosenberg, Arthur Mateos (Nov 22, 2010)

Securing the Cloud: Cloud Computer Security Techniques and Tactics Vic (J.R.) Winkler (Apr 29, 2011)

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ