



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τίτλος Εργασίας

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΑΝΘΡΩΠΙΝΟΥ ΔΥΝΑΜΙΚΟΥ

Όνομα φοιτητή

ΜΑΝΤΖΑΡΗΣ ΒΑΣΙΛΕΙΟΣ

Αθήνα, 2020



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Τριμελής Εξεταστική Επιτροπή

ΤΣΑΔΗΜΑΣ ΑΝΑΡΓΥΡΟΣ

**Μέλος Ε.ΔΙ.Π, Τμήμα Πληροφορικής και Τηλεματικής, Χαροκόπειο
Πανεπιστήμιο**

ΚΑΜΑΛΑΚΗΣ ΘΩΜΑΣ

**Αναπληρωτής Καθηγητής, Τμήμα Πληροφορικής και Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο**

ΜΠΑΡΔΑΚΗ ΚΛΕΟΠΑΤΡΑ

**Επίκουρη Καθηγήτρια, Τμήμα Πληροφορικής και Τηλεματικής,
Χαροκόπειο Πανεπιστήμιο**

Ο ΜΑΝΤΖΑΡΗΣ ΒΑΣΙΛΕΙΟΣ

δηλώνω υπεύθυνα ότι:

- 1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλλει τα πνευματικά δικαιώματα τρίτων.
- 2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.

Ευχαριστίες

Για τη διεκπεραίωση της Πτυχιακής Εργασίας, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου, κ. Τσαδήμα Ανάργυρο για την πολύτιμη βοήθεια και καθοδήγηση που προσέφερε αλλά και την εμπιστοσύνη που μου έδειξε.

Επίσης τον καθηγητή μου κ. Ηρακλή Βαρλάμη για την βοήθεια αλλά και το κίνητρο που μου έδωσε μέσα από την διδασκαλία του, όλα τα χρόνια της φοίτησής μου.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για την σημαντική τους στήριξη κατά την διάρκεια των σπουδών μου.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη στα Ελληνικά	6
Abstract	7
Κατάλογος Εικόνων	8
Κατάλογος Πινάκων	9
Συντομογραφίες	11
Εισαγωγή	12
ΚΕΦ.1: Θεωρητικό Υπόβαθρο	13
1.1 Ορισμός Συστήματος Διαχείρισης Ανθρώπινου Δυναμικού	13
ΚΕΦ.2: Ανάλυση Σεναρίου	17
2.1. Εισαγωγή	17
2.2. Ανάλυση και Σχεδίαση	17
2.2.1 Use Case Diagram	17
2.2.2 Class Diagram	21
2.2.3 Activity Diagrams	25
2.2.4 Deployment Diagram	30
2.2.5 Entity Relationship Diagram	32
2.2.6 API	33
2.2.7 Λειτουργικές & Μη Λειτουργικές Απαιτήσεις	34
ΚΕΦ.3: Υλοποίηση Εφαρμογής	36
3.1 Node.js	36
3.2 React.js	39
3.3 Express.js	40
3.4 MySQL	40
3.5 Sequelize	41
3.6 Bootstrap	41
3.7 React Material Table	42
3.8 JSON Web Token	42
3.9 Επιπλέον Modules	44
ΚΕΦ.4: Εγκατάσταση Εφαρμογής	45
ΚΕΦ.5: Σενάρια Χρήσης	47
5.1 Γενικά	47
5.2 Διαχειριστής (Admin)	50
5.3 Διευθυντής (Manager)	63
5.4 Υπάλληλος (Employee)	64
ΚΕΦ.6: ΕΠΙΛΟΓΟΣ	65
ΒΙΒΛΙΟΓΡΑΦΙΑ	66

Περίληψη στα Ελληνικά

Τα τελευταία χρόνια λόγω της αποδοχής των νέων τεχνολογιών από όλο και περισσότερες εταιρείες, είναι σαφής και η ραγδαία αύξηση της ανάγκης για συστήματα ηλεκτρονικής διαχείρισης των υπαλλήλων τους αλλά και την ψηφιοποίηση και αυτοματοποίηση λειτουργιών μέσα από αυτά.

Σκοπός αυτής της εργασίας είναι η ανάλυση, σχεδίαση και υλοποίηση ενός web based συστήματος διαχείρισης Ανθρώπινου Δυναμικού.

Πρόκειται για ένα σύστημα με στόχο την διαχείριση των υπαλλήλων και την αυτοματοποίηση των διαδικασιών ενός οργανισμού. Αρχικά γίνεται μια ανάλυση γύρω από την έννοια του Συστήματος Διαχείρισης Ανθρώπινου Δυναμικού (Human Resource Management System, HRMS) και τις ανάγκες που πρέπει να υλοποιεί. Επίσης γίνεται αναφορά και επεξήγηση στις γλώσσες προγραμματισμού, βιβλιοθήκες, και τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση του. Στην συνέχεια αναλύεται η δομή και σχεδίαση του συστήματος με την χρήση UML διαγραμμάτων. Επιπρόσθετα αναλύονται όλα τα χαρακτηριστικά και οι λειτουργίες που προσφέρονται από το σύστημα στους χρήστες του. Τέλος, δείχνεται η υλοποίηση της εφαρμογής, μέσα από την παραγωγή διαφόρων σεναρίων και παραδειγμάτων.

Λέξεις κλειδιά: Σύστημα Διαχείρισης Ανθρώπινου Δυναμικού, HRMS, Οργανισμοί, Υπάλληλοι, Διαδικασίες

Abstract

In recent years, due to the increase of companies in the field of technology, the rapid increase in the need for electronic management systems of employees of a company but also the digitization and automation of operations through them is clear.

The purpose of this work is to analyze, design and implement a web based Human Resource Management system. It is a system aimed at managing employees and automating the processes of an organization. Initially, an analysis is made of the concept of the Human Resource Management System (HRMS) and the needs it oughts to implement. There is also a reference and explanation in the programming languages, libraries, and technologies used to implement it. The structure and design of the system is then analyzed using UML diagrams. In addition, all the features and functions offered by the system to its users are analyzed. Finally, the implementation of the application is shown, through the production of various scenarios and examples.

Keywords: HRMS, management systems, web based, UML

Κατάλογος Εικόνων

Εικόνα 1 - Use Case Diagram	21
Εικόνα 2 - Class Diagram	25
Εικόνα 3 - Add Employee Activity Diagram	29
Εικόνα 4 - Manage Salary Details Activity Diagram	30
Εικόνα 5 - Make Payment Activity Diagram	32
Εικόνα 6 - Make Application Activity Diagram	34
Εικόνα 7 - Deployment Diagram	35
Εικόνα 8 - ER Diagram	37
Εικόνα 9 - Αρχιτεκτονική NodeJS [14]	42
Εικόνα 10 - JSON Web Token	48
Εικόνα 11 - Login Screen	52
Εικόνα 12 - Register Screen	53
Εικόνα 13 - Incorrect Login	54
Εικόνα 14 - Inactive Account	54
Εικόνα 15 - Dashboard Admin	55
Εικόνα 16 - Infoboxes	56
Εικόνα 17 - Calendar	56
Εικόνα 18 - Add Event	57
Εικόνα 19 - Bar Charts	58
Εικόνα 20 - Recent Applications	59
Εικόνα 21 - Departments	59
Εικόνα 22 - Add Employee	60
Εικόνα 23 - Employee List	61
Εικόνα 24 - Job List	62
Εικόνα 25 - Add Application	62
Εικόνα 26 - Application List	63
Εικόνα 27 - Salary Details	64
Εικόνα 28 - Salary List	64
Εικόνα 29 - Make Payment	65
Εικόνα 31 - Expense List	66
Εικόνα 32 - Dashboard Manager	67
Εικόνα 33 - Dashboard Employee	68
Εικόνα 34 - Employee Salary Card	68

Κατάλογος Πινάκων

Πίνακας 1 - Πλεονεκτήματα από την χρήση των HRMS [10]

18

Κατάλογος Σχημάτων

Συντομογραφίες

HRMS	Human Resource Management System
HRIS	Human Resource Information System
JS	JavaScript
UML	Unified Modelling Language
ERD	Entity Relationship Diagram
HTTP	HyperText Transfer Protocol
API	Application Programmable Interface
JSON	JavaScript Object Notation
ORM	Object Relational Mapping

Εισαγωγή

Αντικείμενο της εργασίας αποτελεί η δημιουργία μια διαδικτυακής (Web based) εφαρμογής, η οποία κάνει χρήση σύγχρονων τεχνολογιών προγραμματισμού, όπως της γλώσσας Javascript. Η ιδέα της εφαρμογής είναι να αναπτυχθεί ένα Web site, το οποίο θα έχει το ρόλο της διαχείρισης των υπαλλήλων ενός οργανισμού ή μιας εταιρείας από τους υπαλλήλους του τμήματος Ανθρώπινου Δυναμικού ή Γραμματείας. Επίσης, θα δίνει την ικανότητα και στα υπόλοιπα μέλη της εταιρείας/οργανισμού να εισέρχονται στο σύστημα εκτελώντας διαφορετικές λειτουργίες και έχοντας πρόσβαση σε διαφορετικές πληροφορίες, αναλόγως με το επίπεδο της πρόσβασης τους. Οι κατηγορίες αυτές των επιπέδων πρόσβασης διαχωρίζονται σε Διαχειριστής (Administrator), Διευθυντής (Manager) και απλός Υπάλληλος (Employee).

Με την αναφορά σε “διαχείριση” των υπαλλήλων μιας εταιρείας/οργανισμού εννοείται η διαχείριση των πληροφοριών που κατέχει η εταιρεία για αυτούς, παρέχοντας έναν εύκολο τρόπο προβολής, προσθήκης, αλλαγής και αποθήκευσης τους.

Επίσης το σύστημα λειτουργεί και ως ένα εργαλείο Payroll/Expense management καθώς υπάρχει η δυνατότητα αποθήκευσης δεδομένων εξόδων και πληρωμών της εταιρείας και προβολής των μηνιαίων/ετήσιων εξόδων/πληρωμών με την χρήση διαγραμμάτων.

ΚΕΦ.1: Θεωρητικό Υπόβαθρο

1.1 Ορισμός Συστήματος Διαχείρισης Ανθρώπινου Δυναμικού

Τα **HRMS** (Human Resource Management Systems) ή **HRIS** (Human Resource Information Systems) είναι συστήματα λογισμικού που διαχειρίζονται μέρος ή όλη τη λειτουργία ανθρώπινου δυναμικού ενός οργανισμού/εταιρείας. Συνηθισμένα στοιχεία του συστήματος είναι δημογραφικά στοιχεία για το προσωπικό, διαχείριση αμοιβών, πριμ, αδειών και συνεντεύξεων, εκπαίδευση, μισθοδοσία και αναφορές αποδόσεων. [4]

Σύμφωνα με τον Gara και τον Walker τα Πληροφοριακά Συστήματα Ανθρώπινου Δυναμικού θα βοηθήσουν τις επιχειρήσεις «...να τηρήσουν μία ορθή, ολοκληρωμένη και ενημερωμένη βάση δεδομένων, όπου θα μπορούν να ανακτηθούν όποτε κρίνεται χρήσιμο μέσα από εκτυπώσεις...». Σε κάθε περίπτωση σκοπός είναι να παρέχουν πληροφόρηση στο τμήμα ανθρώπινου δυναμικού και να υποστηρίξει αποφάσεις που έχουν να κάνουν με το ανθρώπινο δυναμικό. [1], [4], [5]

Τα Πληροφοριακά Συστήματα Διοίκησης Ανθρώπινου Δυναμικού ή HRIS (Human Resource Information Systems) είναι συστήματα βάσει των οποίων μπορούμε να συλλέξουμε, να αποθηκεύσουμε, να χρησιμοποιήσουμε, να αναλύσουμε, να ανακτήσουμε και να διανείμουμε πληροφορίες σχετικές με το ανθρώπινο δυναμικό ενός οργανισμού. [6]

Τα Πληροφοριακά Συστήματα Ανθρώπινου Δυναμικού είναι η σύνθεση βάσεων δεδομένων, εφαρμογών πληροφορικής, hardware και software απαραίτητα για την συλλογή, καταγραφή, αποθήκευση, διαχείριση, διανομή, παρουσίαση και εκμετάλλευση δεδομένων που αφορούν στο ανθρώπινο δυναμικό. [3]

Σύμφωνα με τους Kettley και Reilly (2003) ένα Πληροφοριακό Σύστημα Ανθρώπινου Δυναμικού αποτελείται από ένα πλήρως ολοκληρωμένο δίκτυο στο εύρος ενός οργανισμού από δεδομένα, πληροφορίες, υπηρεσίες, βάσεις δεδομένων, εργαλεία και συναλλαγές για το ανθρώπινο δυναμικό. Εάν αυτές οι συναλλαγές γίνονται με τη χρήση του internet μιλούμε για “electronic-HR” ή “e- HR” που πολλές φορές συμπεριλαμβάνουν την εφαρμογή συμβατικών, δικτυακών τεχνολογιών και τεχνολογιών φωνής, portals και web εφαρμογών (Panayotopoulou, L., Vakola, M. and Galanaki E., 2007). [7], [8]

Οι ραγδαίες εξελίξεις στα Πληροφοριακά Συστήματα τις τελευταίες πέντε δεκαετίες επηρέασαν αναπόφευκτα και τις επιχειρήσεις, οι οποίες χρησιμοποιούν την τεχνολογία ως ένα εργαλείο που θα αυξήσει την αποτελεσματικότητα και αποδοτικότητα τους. Έτσι, όλο και περισσότερες εταιρείες εφαρμόζουν Πληροφοριακά Συστήματα και στη Διοίκηση του Ανθρώπινου Δυναμικού.

Παρακάτω αναφέρονται ορισμένα οφέλη που θα καρπωθεί η επιχείρηση από τη χρήση των Πληροφοριακών Συστημάτων Ανθρώπινου Δυναμικού:

- **Εύκολη πρόσβαση** σε δεδομένα και πληροφορίες που αφορούν στο προσωπικό.
- **Αυτοματοποίηση των διαδικασιών** και μείωση της γραφειοκρατίας. Έτσι, οι λειτουργίες διενεργούνται σε οριζόντιο επίπεδο και γίνονται περισσότερο ευέλικτες καταργώντας την κάθετη δομή που προκαλεί γραφειοκρατία.^[9]
- Περισσότερο **ακριβή και συγκεκριμένα δεδομένα** με μικρότερη πιθανότητα λάθους.
- **Γρηγορότερη πρόσβαση και επεξεργασία** με τη χρήση τεχνολογιών που οδηγούν τόσο στην ικανοποίηση του εργαζομένου που κάνει για παράδειγμα αίτηση για μια πληροφορία όσο και στη γρηγορότερη λήψη διοικητικών αποφάσεων.
- **Δημιουργία «οργανωσιακής μνήμης»** στην επιχείρηση που θα αποθηκεύει πληροφορίες για όλους τους εργαζόμενους (ακόμα και αυτούς που αποχώρησαν) και θα μπορεί να προβλέπει μελλοντικές αλλαγές καριέρας των υπαλλήλων, να

αναδεικνύει τους εξαιρετικούς από τους «μέτριους» εργαζομένους, τις δυνάμεις και αδυναμίες των στελεχών και τις εκπαιδευτικές ανάγκες ή τις ανάγκες της επιχείρησης για στελέχωση.

- Μπορεί να **εκμηδενίσει αποστάσεις** και έτσι κάνει δυνατή την επεξεργασία δεδομένων όλων των υπαλλήλων ακόμα και αν η επιχείρηση έχει υποκαταστήματα σε πολλές περιοχές. Οι κατανεμημένες βάσεις δεδομένων και οι τεχνολογίες επικοινωνίας επιτρέπουν την παροχή της πληροφορίας σε πολλά μέρη ταυτόχρονα. [9]
- Τα σύγχρονα HRMS είναι μέρος ενός **ολοκληρωμένου πληροφοριακού συστήματος** που έχει μια επιχείρηση.
- Παροχή σύγχρονων υπηρεσιών και δημιουργία μιας **σύγχρονης εικόνας** για την επιχείρηση που την καθιστά **ελκυστική** στην αγορά εργασίας και στο υπάρχον και δυνητικό προσωπικό.
- **Μείωση του κόστους εργασίας** μέσω της μείωσης χρόνου εκτέλεσης ή κατάργησης ορισμένων διαδικασιών.
- **Αλλαγή στις σχέσεις** των εργαζομένων και των στελεχών γραμμής, όπως και αλλαγή των σχέσεων των στελεχών γραμμής με το τμήμα Ανθρώπινου Δυναμικού. Πλέον, οι εργαζόμενοι μπορούν να αποτελέσουν μέρος της διοίκησης και το τμήμα ανθρώπινου δυναμικού να συνεργαστεί παραγωγικά με τα υπόλοιπα στελέχη.
- **Αναβάθμιση του ρόλου του HR- manager** και μεγαλύτερη αναγνώριση της σημαντικότητάς του.
- **Βελτίωση των παρεχόμενων υπηρεσιών** του τμήματος Ανθρώπινου Δυναμικού που προέρχεται από τη γνώση των δεδομένων και όχι από την κτήση τους.
- Δημιουργία **ανταγωνιστικού πλεονεκτήματος** για την επιχείρηση, δεδομένου ότι το ανθρώπινο κεφάλαιο είναι ο πιο σημαντικός πόρος της επιχείρησης και ο μόνος που «αντιγράφεται» δύσκολα από τις ανταγωνίστριες εταιρείες.
- Μεταφορά του κέντρου βάρους από παραδοσιακές λειτουργίες σε **στρατηγική διοίκηση** του ανθρώπινου δυναμικού. Το τμήμα ανθρώπινου δυναμικού αποδεσμεύεται από πιο τετριμμένες διαδικασίες και εξασφαλίζει χρόνο να ασχοληθεί με στρατηγικά σχέδια.

	Αριθμητικός μέσος (mean)	Απόκλιση (standard deviation)
Γρήγορη ανταπόκριση και πρόσβαση στις πληροφορίες	4,377	0,813
Βελτίωση ελέγχου των δεδομένων	4,186	0,892
Μείωση καταχώρησης δεδομένων πολλές φορές	3,991	1,026
Οργάνωση διαδικασιών στους ανθρώπινους πόρους	3,835	0,927
Λιγότερα λάθη	3,779	1,1
Προτυποποίηση διαδικασιών	3,522	0,983
Μείωση γραφειοκρατίας	3,412	1,257
Μείωση ανθρώπινου δυναμικού	3,404	1,002
Καλύτερος έλεγχος των λειτουργιών ανθρώπινου δυναμικού	3,287	1,227
Λήψη αποφάσεων βασισμένη σε καλή πληροφόρηση	3,257	1,14
Βελτίωση υπηρεσιών στους πελάτες	3,254	1,225
Δημιουργία ανταγωνιστικού πλεονεκτήματος	3,063	1,149

Πίνακας 1 - Πλεονεκτήματα από την χρήση των HRMS ^[10]

ΚΕΦ.2: Ανάλυση Σεναρίου

2.1. Εισαγωγή

Στο παρόν κεφάλαιο θα πραγματοποιηθεί μια αναλυτική περιγραφή του σεναρίου που ακολουθήθηκε, προκειμένου να πραγματοποιηθεί η υλοποίηση της εφαρμογής. Παραγράφω καταγράφονται τα UML διαγράμματα όπως Use Case Diagram, Activity Diagram καθώς και Class Diagram που σχεδιάστηκαν κατά τη φάση της ανάλυσης και σχεδίασης της εφαρμογής.

Η εφαρμογή που θα πραγματοποιηθεί στην συνέχεια έχει ως αρχική ιδέα να αναπτυχθεί ένα σύστημα διαχείρισης ανθρώπινου δυναμικού για έναν οργανισμό/εταιρεία.

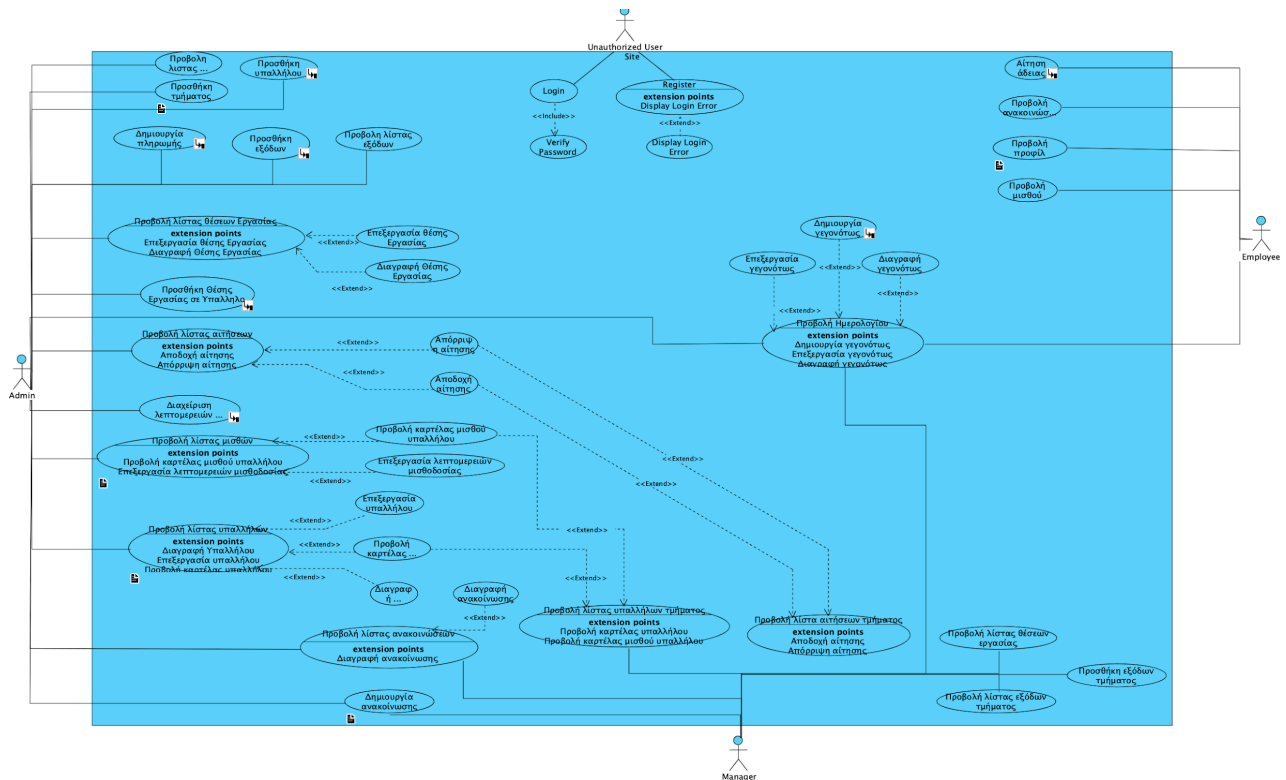
2.2. Ανάλυση και Σχεδίαση

Για την ανάλυση και σχεδίαση της εφαρμογής χρησιμοποιήθηκε η γλώσσα UML.

Σύμφωνα με το επίσημο εγχειρίδιο αναφοράς, η Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language ή UML) είναι μια γραφική γλώσσα γενικού σκοπού, η οποία χρησιμοποιείται για τον προσδιορισμό, οπτικοποίηση, ανάπτυξη και τεκμηρίωση των κατασκευασμάτων (artifacts) ενός συστήματος λογισμικού]. Η UML αποτελεί industry standard για τη μοντελοποίηση συστημάτων λογισμικού και χρησιμοποιείται στη μοντελοποίηση συστημάτων βασισμένων σε αντικείμενα (αντικειμενοστρεφή συστήματα).^[11]

2.2.1 Use Case Diagram

Κάθε χρήστης που εισέρχεται στο περιβάλλον της εφαρμογής μπορεί να πραγματοποιήσει διαφορετικές λειτουργίες και να έχει πρόσβαση σε διαφορετικές πληροφορίες ανάλογα με το ρόλο του.



Εικόνα 1 - Use Case Diagram

Το συγκεκριμένο διάγραμμα δείχνει όλες τις περιπτώσεις χρήση του συστήματος από τους χρήστες. Το σύστημα μπορεί να χρησιμοποιηθεί μόνο από εξουσιοδοτημένους χρήστες τους οποίους χωρίζουμε σε 3 κατηγορίες

- Διαχειριστής/Admin (Ο υπάλληλος του τμήματος Ανθρώπινου Δυναμικού)
- Διευθυντής/Manager
- Απλός Υπάλληλος/Employee

Αρχικά, ένας μη εξουσιοδοτημένος χρήστης έχει τη δυνατότητα να αποκτήσει πρόσβαση στο σύστημα κατόπιν εγγραφής. Πιο συγκεκριμένα, υπάρχει η λειτουργία Register. Η λειτουργία Register έχει σχεδιαστεί για δημιουργία λογαριασμών Employee. Προτείνεται η λειτουργία προσθήκη υπαλλήλου από τον Admin του συστήματος για την δημιουργία λογαριασμών για Admin ή/και Manager.

Ο χρήστης αφού δημιουργήσει λογαριασμό θα πρέπει να περιμένει να ενεργοποιηθεί ο λογαριασμός του από τον Admin. Σε περίπτωση δημιουργίας του λογαριασμού από τον Admin, ο λογαριασμός είναι ήδη ενεργοποιημένος.

Το βήμα αυτό συμπεριλήφθηκε καθώς είναι σημαντικό να ελέγχονται τα στοιχεία για την επαλήθευση της ταυτότητας του χρήστη και της βεβαίωσης ότι πρόκειται για υπάλληλο του οργανισμού/εταιρείας.

Η λειτουργία Login σχετίζεται με τις λειτουργίες επαλήθευσης κωδικού πρόσβασης και εμφάνιση σφάλματος σύνδεσης. Η σύνδεση της λειτουργίας Login και Verify Password πραγματοποιείται με Include, διότι προκειμένου να συνδεθεί κανείς στο σύστημα θα πρέπει να επαληθευτεί με επιτυχία ο κωδικός πρόσβασης.

Αντίθετα, η σύνδεση της λειτουργίας Login και Display Login Error πραγματοποιείται με Extend. Η συσχέτιση αυτή αποτυπώνεται με αυτόν τον τρόπο καθώς η λειτουργία Display Login Error δεν πραγματοποιείται πάντα καθώς προϋποθέτει σφάλμα εισαγωγής του κωδικού πρόσβασης.

Ο **Admin** ως ο χρήστης για τον οποίο έχει σχεδιαστεί το σύστημα αυτό έχει πρόσβαση στο μεγαλύτερο αριθμό των λειτουργιών του συστήματος.

Οι λειτουργίες του συστήματος εκτελούνται πάνω στα δεδομένα των πληροφοριών της εταιρείας/οργανισμού και είναι χωρισμένα στις εξής κατηγορίες:

- Τμήματα της εταιρείας (Departments)
- Υπάλληλοι της Εταιρείας (Employees)
 - Προσωπικά Στοιχεία
 - Οικονομικά Στοιχεία
- Θέσεις Εργασίας των Υπαλλήλων (Jobs)
- Αιτήσεις Άδειας των Υπαλλήλων
- Έξοδα των Τμημάτων (Expenses)
- Ανακοινώσεις Τμημάτων (Announcements)

Πάνω σε αυτές τις κατηγορίες μπορούν να εκτελέσουν διαφορετικές λειτουργίες οι χρήστες του συστήματος ανάλογα με τον ρόλο που έχουν.

Admin

Ο διαχειριστής του συστήματος έχει δικαίωμα στις εξής λειτουργίες:

- Προβολή λίστας τμημάτων
- Προσθήκη τμήματος
- Προβολή λίστας υπαλλήλων
- Προσθήκη Υπαλλήλου
- Προβολή λίστας Θέσεων Εργασίας
- Προσθήκη Θέσης Εργασίας σε Υπάλληλο
- Προβολή Λίστας Αιτήσεων Αδειών
- Κατάθεση Αίτησης Άδειας
- Αποδοχή/Απόρριψη Αιτήσεων Υπαλλήλων
- Προβολή Λίστας Μισθών Υπαλλήλων
- Διαχείριση Λεπτομερειών Μισθοδοσίας
- Δημιουργία Πληρωμής
- Προσθήκη εξόδων
- Προβολή λίστας Ανακοινώσεων
- Προσθήκη Ανακοίνωσης
- Δημιουργία προσωπικών γεγονότων σε Ημερολόγιο

Manager

Ο διευθυντής της εταιρείας έχει δικαίωμα στις εξής λειτουργίες που αφορούν μόνο το τμήμα στο οποίο έχει τοποθετηθεί

- Προβολή λίστας Υπαλλήλων Τμήματος
- Προβολή λίστας Θέσεων Εργασίας Τμήματος
- Προσθήκη Θέσης Εργασίας σε Υπάλληλο Τμήματος
- Προβολή Λίστας Αιτήσεων Τμήματος
- Κατάθεση Αίτησης Άδειας

- Αποδοχή/Απόρριψη Αιτήσεων Υπαλλήλων Τμήματος
- Προσθήκη εξόδων Τμήματος
- Προβολή λίστας Ανακοινώσεων Τμήματος
- Προσθήκη Ανακοίνωσης Τμήματος
- Δημιουργία προσωπικών γεγονότων σε Ημερολόγιο

Employee

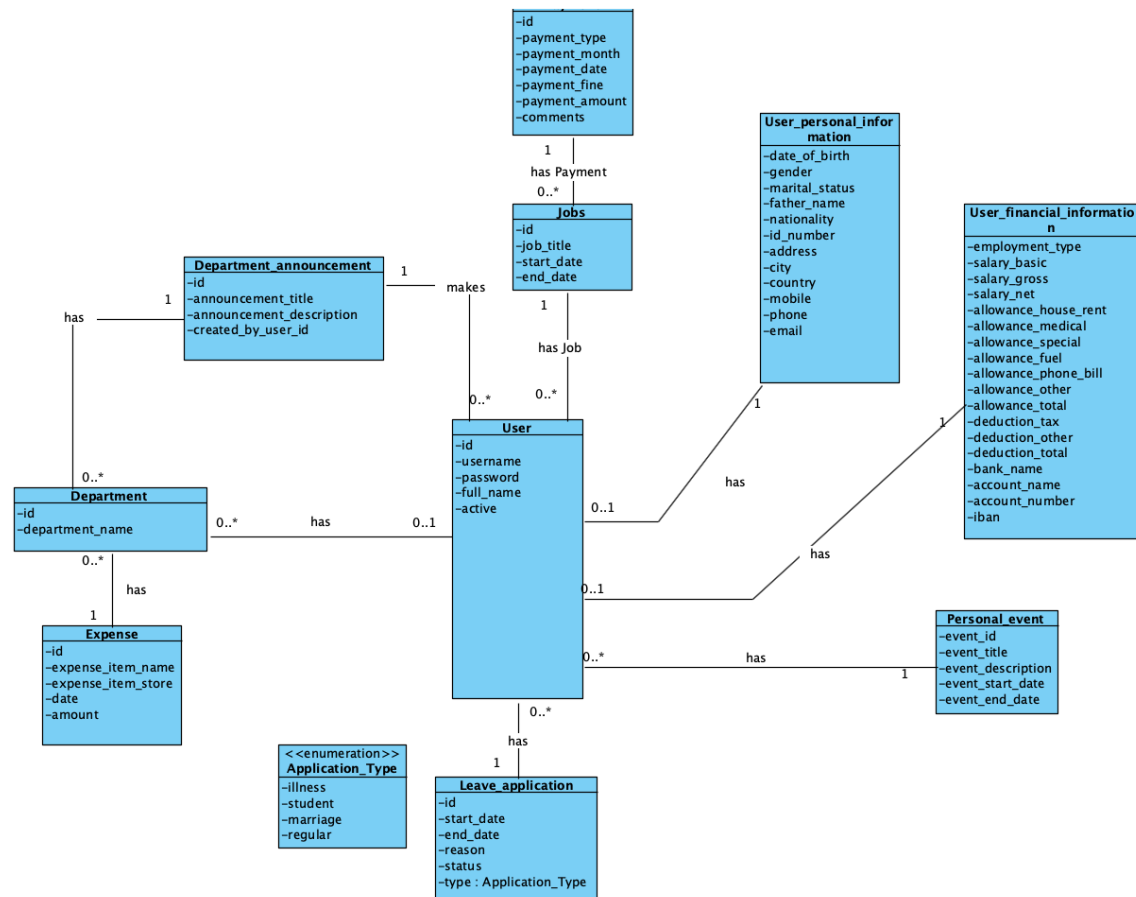
Ο απλός Υπάλληλος της εταιρείας έχει δικαίωμα στις εξής λειτουργίες:

- Κατάθεση αίτησης Άδειας
- Προβολή των αιτήσεων των αδειών του
- Προβολή των στοιχείων του
- Προβολή λίστας ανακοινώσεων τμήματος
- Δημιουργία προσωπικών γεγονότων σε Ημερολόγιο

Καθώς πρέπει να κατανοήσουμε καλύτερα τις έννοιες μερικών βασικών λειτουργιών του συστήματος, είναι σημαντική η απεικόνιση των οντοτήτων του συστήματος μέσα από ένα Class Diagram.

2.2.2 Class Diagram

Σε ένα Class Diagram αναπαρίστανται τα χαρακτηριστικά για τις οντότητες του συστήματος. Στη συγκεκριμένη περίπτωση, αυτές είναι: ο χρήστης (User), οι προσωπικές πληροφορίες του χρήστη (User_Personal_Information), οι οικονομικές πληροφορίες του χρήστη (User_Financial_Information), τα προσωπικά γεγονότα του χρήστη (Personal_event), οι θέσεις εργασίας του χρήστη (Jobs), οι πληρωμές των θέσεων εργασίας, οι αιτήσεις αδειών των χρηστών (Leave_application), τα τμήματα της εταιρείας (Department), και τα έξοδα των τμημάτων (Expense).



Εικόνα 2 - Class Diagram

User

Για τον χρήστη του συστήματος αποθηκεύεται το username του το οποίο χρησιμοποιείται κατα το Login. Επίσης αποθηκεύεται ο προσωπικός κωδικός του χρήστη, κρυπτογραφημένος με την χρήση της βιβλιοθήκης bcrypt. Επίσης υπάρχει το Ονοματεπώνυμο του χρήστη αλλά και η κατάσταση του λογαριασμού του η οποία διακρίνεται σε ενεργός και μη ενεργός χρήστης.

UserPersonalInformation

Σε ξεχωριστό πίνακα και με συνάρτηση 1 προς 1 με τον πίνακα User αποθηκεύονται οι προσωπικές πληροφορίες του χρήστη όπως η ημερομηνία γέννησης του, το φύλο του, η οικογενειακή κατάσταση, όνομα Πατρός, εθνικότητα, αριθμός ταυτότητας, διεύθυνση, πόλη και χώρα κατοικίας, κινητό και σταθερό τηλέφωνο και email

UserFinancialInformation

Σε ξεχωριστό πίνακα και με συνάρτηση 1 προς 1 με την οντότητα User αποθηκεύονται οι οικονομικές πληροφορίες του χρήστη όπως ο τύπος απασχόλησης του (Full Time, Part Time), ο μισθός, ο συνολικός του μισθός, ο καθαρός μισθός, διάφοροι τύποι επιδομάτων, διάφοροι τύποι φόρων, όνομα τράπεζας, όνομα λογαριασμού, αριθμός λογαριασμού και αριθμός IBAN.

Ο συνολικός μισθός προκύπτει από την πρόσθεση του μισθού με το σύνολο των επιδομάτων.

Ο καθαρός μισθός προκύπτει από τον μισθό μείον το σύνολο των φόρων.

Personal Event

Για τα προσωπικά γεγονότα του κάθε χρήστη, τα οποία απεικονίζονται ως ημερολόγιο στη διεπαφή χρήστη αποθηκεύεται ο τίτλος, η περιγραφή, η ημερομηνία και ώρα έναρξης και η ημερομηνία και ώρα λήξης.

Jobs

Στην οντότητα αυτή αποθηκεύονται όλες οι θέσεις εργασίας των χρηστών του συστήματος. Όπως βλέπουμε και στην συσχέτιση του User με την οντότητα Jobs ο κάθε χρήστης μπορεί να έχει παραπάνω από μια θέση εργασίας, καθώς μπορεί κατά το ιστορικό του στην συγκεκριμένη εταιρεία να είχε πάνω από μια θέσεις. Έτσι συλλέγεται ιστορικό για όλους τους χρήστες, ακόμα και για όσους έχουν αποχωρήσει.

Οι πληροφορίες που αποθηκεύονται είναι ο τίτλος, η ημερομηνία έναρξης και η ημερομηνία λήξης.

Payment

Στην οντότητα αυτή αποθηκεύεται το ιστορικό των πληρωμών της εταιρείας προς τις θέσεις εργασίας, και επομένως προς του χρήστες του συστήματος. Κάθε φορά που γίνονται πληρωμές ο Admin μπορεί να καταθέσει τα στοιχεία των πληρωμών και στο σύστημα, κρατώντας έτσι ένα ηλεκτρονικό ιστορικό, και έχοντας μια καλύτερη εικόνα των πληρωμών της εταιρείας με την χρήση διαγραμμάτων.

Αποθηκεύονται ο τρόπος πληρωμής, ο μήνας της πληρωμής, η ημερομηνία της πληρωμής (πότε κατατέθηκαν τα χρήματα), πιθανά πρόστιμα που επιβλήθηκαν, το σύνολο της πληρωμής και πιθανά σχόλια.

Leave Application

Στην οντότητα αυτή αποθηκεύονται οι αιτήσεις αδειών που κάνει ένας χρήστης. Οι πληροφορίες που αποθηκεύονται είναι η ημερομηνία έναρξη της άδειας, η ημερομηνία λήξης, ο τύπος άδειας (Normal, Student, Illness, Marriage), πιθανά επιπρόσθετα σχόλια και η κατάσταση της άδειας (Pending, Approved, Rejected).

Department

Στην οντότητα αυτή αποθηκεύονται τα τμήματα της εταιρείας / οργανισμού. Αποθηκεύεται το όνομα του τμήματος.

Expense

Στην οντότητα αυτή αποθηκεύονται τα έξοδα των τμημάτων της εταιρείας/οργανισμού. Αποθηκεύονται πληροφορίες όπως το όνομα του προϊόντος, το όνομα του καταστήματος στο οποίο αγοράστηκε, η ημερομηνία αγοράς, το ποσό της αγοράς

Department Announcement

Στην οντότητα αυτή αποθηκεύονται οι ανακοινώσεις των τμημάτων της οποίες μπορούν να δουν οι χρήστες των τμημάτων αυτών. Αποθηκεύονται στοιχεία όπως ο τίτλος της ανακοίνωσης, η περιγραφή, και το id του χρήστη που δημιούργησε την ανακοίνωση.

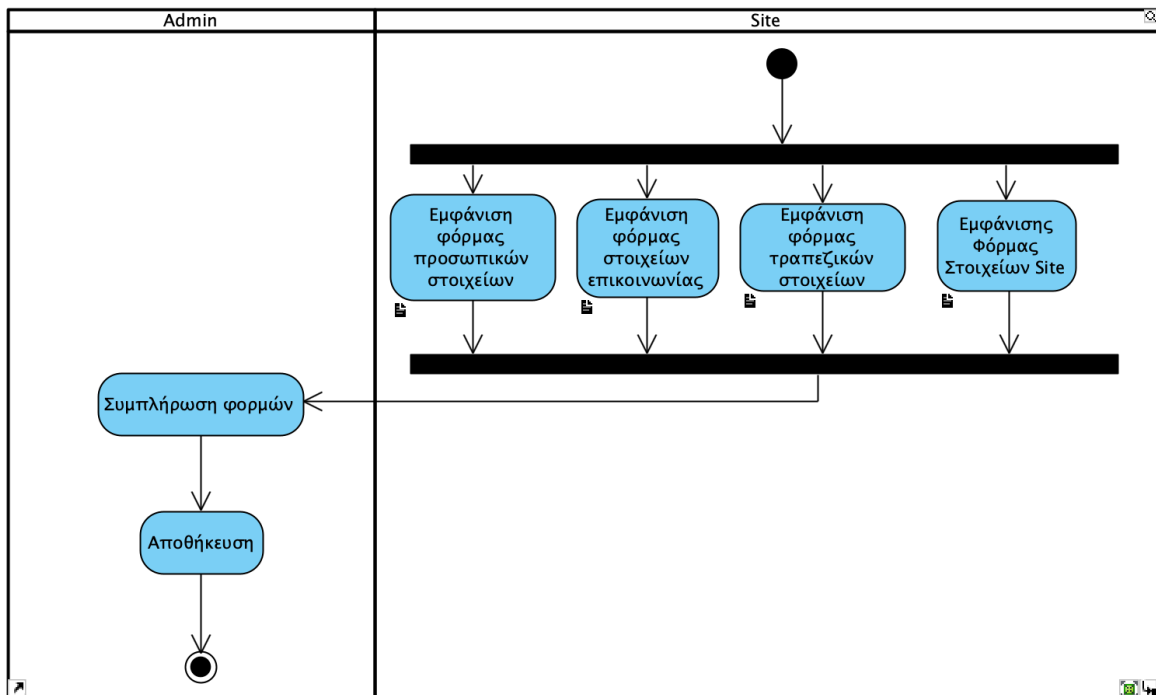
2.2.3 Activity Diagrams

Για την καλύτερη κατανόηση μερικών βασικών λειτουργιών του συστήματος που συζητήθηκαν στο κεφάλαιο 2.2.1 έχουν δημιουργηθεί Activity Diagrams

Τα Activity Diagrams μας επιτρέπουν να δούμε μία συγκεκριμένη λειτουργία του συστήματος με μεγαλύτερη λεπτομέρεια.

Προσθήκη Υπαλλήλου

Η λειτουργία της προσθήκης υπαλλήλου προσφέρεται μόνο στους Admin του συστήματος.



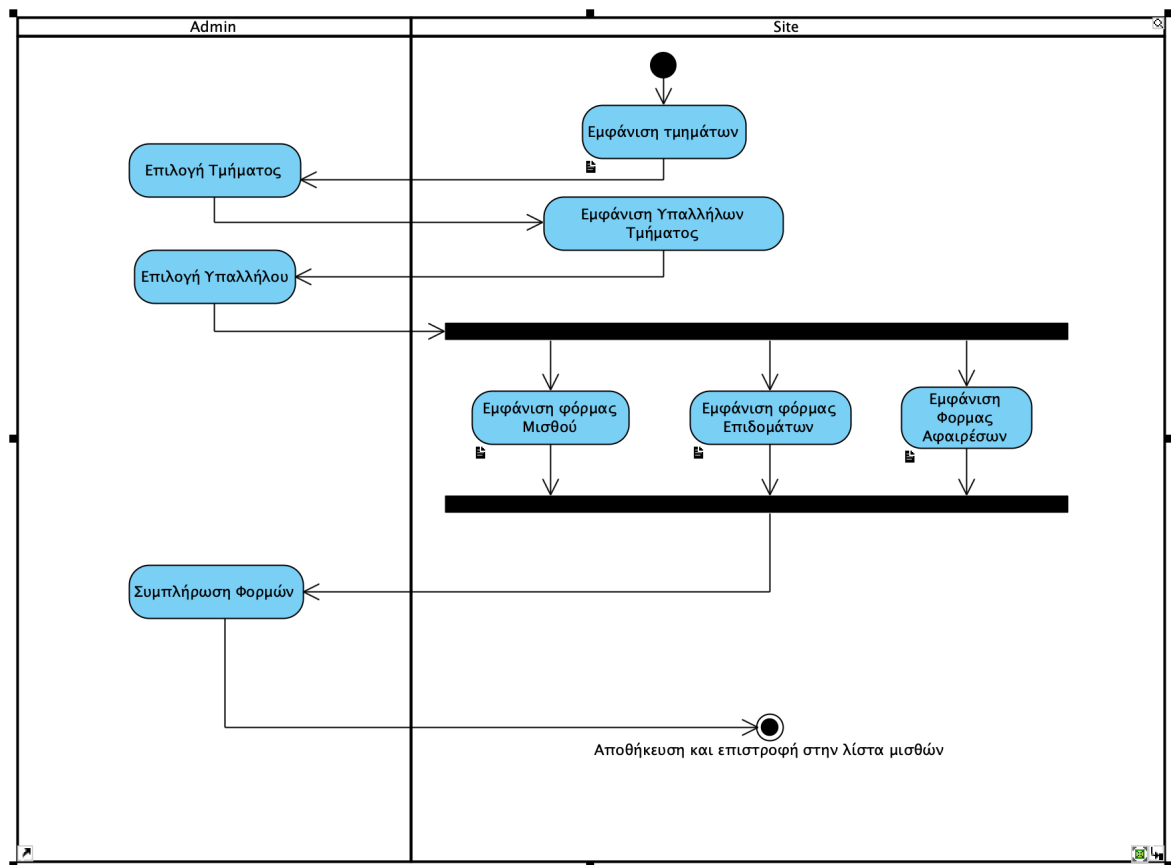
Εικόνα 3 - Add Employee Activity Diagram

Στο βήμα αυτο εμφανίζονται φόρμες για την συμπλήρωση στοιχείων όπως προσωπικών (π.χ. Όνομα, Τηλέφωνο, Αριθμός Λογαριασμού, Username, Password κλπ)

Διαχείριση Λεπτομερειών Μισθοδοσίας

Η λειτουργία διαχείρισης λεπτομερειών μισθοδοσίας εκτελείται από τον Admin του συστήματος.

Κατα την διαδικασία αυτή ο διαχειριστής θα έχει την δυνατότητα να προσθέσει τα στοιχεία μισθοδοσίας ενός υπαλλήλου. Στα στοιχεία αυτά περιλαμβάνεται ο μισθός, συνολικός (μικτός) μισθός, καθαρός μισθός, διάφορα επιδόματα όπως Στεγαστικό, για Βενζίνη, για τηλεφωνικούς λογαριασμούς κτλ, και διάφοροι φόροι όπως εισοδηματικές εισφορές κτλ.



Εικόνα 4 - Manage Salary Details Activity Diagram

Αρχικά το σύστημα εμφανίζει το σύνολο των τμημάτων. Στην συνέχεια ο χρήστης επιλέγει τον χρήστη του τμήματος του οποίου τα στοιχεία θέλει να διαχειριστεί. Επίσης σε περίπτωση που ο διαχειριστής δεν θυμάται σε ποιο τμήμα είναι ο υπάλληλος, υπάρχει η δυνατότητα για επιλογή “All Departments” για να αποφευχθεί το φιλτράρισμα ανα τμήμα.

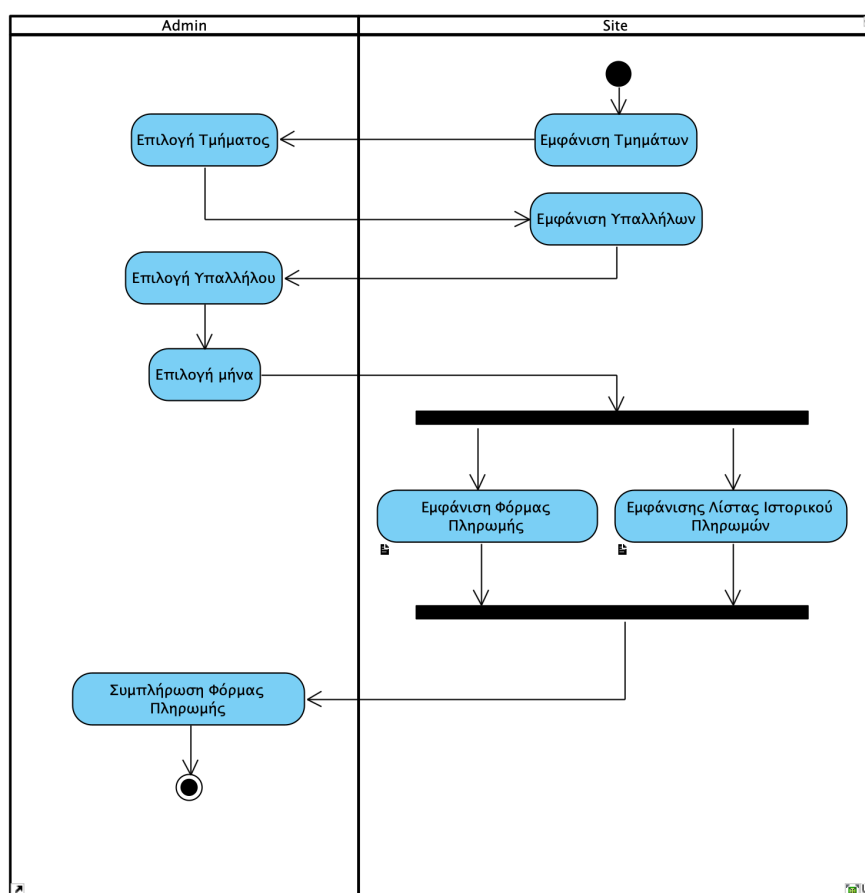
Έπειτα εμφανίζονται οι φόρμες για συμπλήρωση των στοιχείων μισθοδοσίας. Μετα την επιτυχημένη συμπλήρωση τους ο διαχειριστής επιλέγει “Αποθήκευση” και τα στοιχεία αποθηκεύονται στην Βάση Δεδομένων.

Τέλος η σελίδα ανακατευθίνει τον χρήστη στην λίστα μισθών.

Προσθήκη Πληρωμής

Κατα τη λειτουργία αυτή ο Διαχειριστής έχει την δυνατότητα προσθήκης μιας πληρωμής σε μια θέση εργασίας, δηλαδή κατ' επέκταση σε έναν χρήστη.

Η αποθήκευση πληρωμών δίνει την δυνατότητα στο σύστημα να κρατάει ένα ιστορικό εξόδων και να λειτουργεί ως εργαλείο για την λήψη μελλοντικών στρατηγικών αποφάσεων της εταιρείας (π.χ Πρόσληψη νέων υπαλλήλων).



Εικόνα 5 - Make Payment Activity Diagram

Κατά την λειτουργία αυτή ο Διαχειριστής επιλέγει το τμήμα και στην συνέχεια έναν υπάλληλο που τον ενδιαφέρει. Έπειτα επιλέγει τον μήνα για τον οποίο θα ήθελε να καταθέσει μια πληρωμή.

Στην συνέχεια εμφανίζονται η φόρμα με τα στοιχεία πληρωμής. Το ποσό της πληρωμής είναι “κλειδωμένο” και παράγεται από το ποσό του καθαρού μισθού του συγκεκριμένου χρήστη. Ο μόνος τρόπος αλλαγής του ποσού πληρωμής είναι μέσω της προσθήκης κάποιου προστίμου.

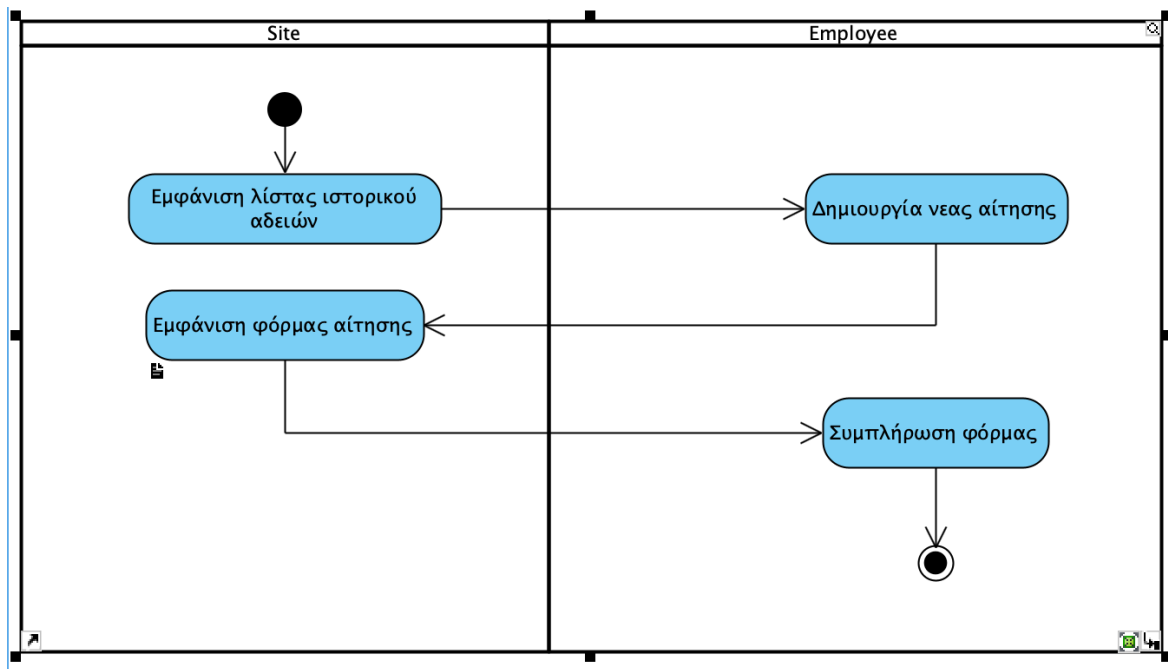
Παράλληλα εμφανίζεται μια λίστα με το ιστορικό πληρωμών του συγκεκριμένου χρήστη στην εταιρεία. Η ύπαρξη της λίστας αυτής βοηθάει τον Διαχειριστή να έχει μια εικόνα των προηγούμενων πληρωμών για την αποφυγή πιθανού λάθους κατα την πληρωμή.

Αίτηση Άδειας Απουσίας

Ο κάθε χρήστης του συστήματος, ανεξαρτήτου ρόλου, έχει την δυνατότητα αίτησης άδειας απουσίας. Ο χρήστης συμπληρώνει τα στοιχεία της αίτησης, όπως τύπο αίτησης, ημερομηνία έναρξης, ημερομηνία λήξης και πιθανά σχόλια.

Κατα την αποστολή της αίτησης, η αίτηση αποθηκεύεται στην Βάση Δεδομένων με κατάσταση ως “εκκρεμής”. Για την αποδοχή/απόρριψη της αίτησης θα πρέπει να επέμβει ο διαχειριστής του συστήματος ή ο διευθυντής του τμήματος στο οποίο ανήκει ο χρήστης.

Είναι σημαντικό να σημειωθεί πως θα πρέπει να μην δίνεται η δυνατότητα σε έναν διαχειριστή ή διευθυντή να αποδέχεται τις δικές του αιτήσεις. Έτσι στην λίστα αιτήσεων γίνεται έλεγχος αν η αίτηση ανήκει στον χρήστη που είναι συνδεδεμένος και αν είναι, δεν του δίνεται η ικανότητα επεξεργασίας.



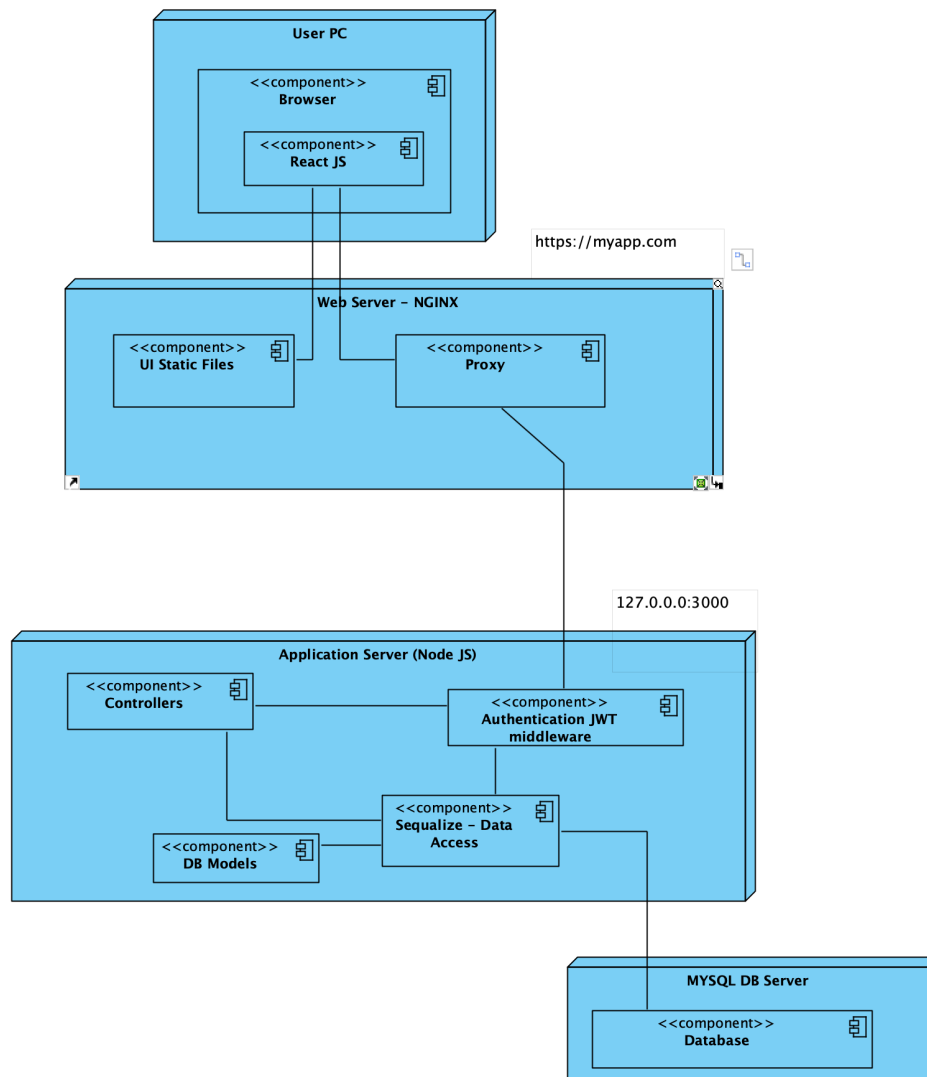
Εικόνα 6 - Make Application Activity Diagram

2.2.4 Deployment Diagram

Τα διαγράμματα ανάπτυξης (*deployment diagrams*) αναπαριστούν την αντιστοίχιση του λογισμικού σε επεξεργαστικές μονάδες-κόμβους. Μπορούν να χρησιμοποιηθούν για να δείξουν ποια συστατικά τρέχουν σε ποιους κόμβους. Ένας **κόμβος** (*node*) είναι ένα φυσικό αντικείμενο που στη γενική περίπτωση έχει τουλάχιστον μνήμη και δυνατότητα επεξεργασίας.

Στην περίπτωση της εφαρμογής που έχει υλοποιηθεί στην παρούσα πτυχιακή εργασία, το σύστημα, εκτός από την Βάση Δεδομένων που έχει στηθεί σε server, έχει υλοποιηθεί τοπικά (localhost).

Παρόλα αυτά, έχει δημιουργηθεί το Deployment Diagram για να αναδειχθεί το πως μπορεί να είναι αντιστοιχισμένο το λογισμικό με μερικές παραμετροποιήσεις και παράταξη σε server.



Εικόνα 7 - Deployment Diagram

Στο ανωτέρω Deployment Diagram υπάρχουν 4 κόμβοι, ένας κόμβος είναι ο υπολογιστής που χρησιμοποιεί ο χρήστης του συστήματος (User PC), ένας είναι ο Web Server που είναι υπεύθυνος για την εξυπηρέτηση των στατικών αρχείων αλλά και την ανακατεύθυνση στον τοπικό Application Server, ένας Application Server που είναι υπεύθυνος για την παραγωγή των html αρχείων από Javascript κώδικα (λόγω της χρήσης Node.js και React.js) και τέλος ένας MySQL server ο οποίος κατέχει την βάση δεδομένων που χρησιμοποιεί το σύστημα

Στον Application Server διακρίνουμε τους διακριτούς κόμβους όπως το Authentication Middleware το οποίο είναι υπεύθυνο για την γέννηση και επαλήθευση των JSON Web Tokens, μετά από έλεγχο του username και password του χρήστη.

Έπειτα υπάρχει η επικοινωνία με τους Controllers οι οποίοι διαχειρίζονται τις διάφορες HTTP κλήσεις που γίνονται από τους χρήστες προς το σύστημα.

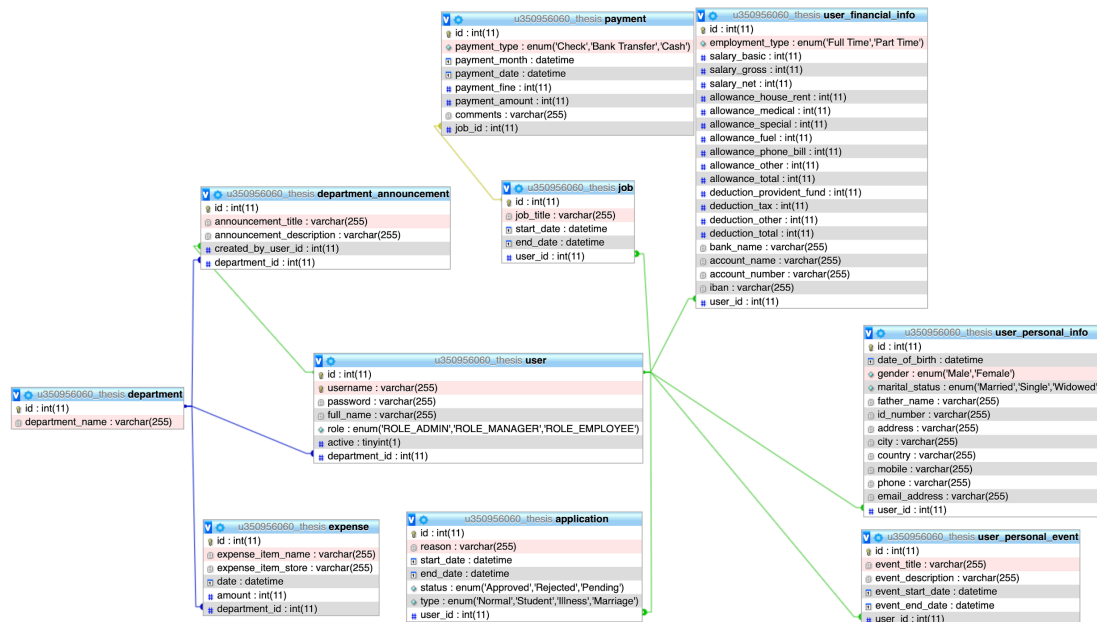
Οι controllers οι οποίοι στο σύνολο τους αποτελούν το API του συστήματος έχουν πρόσβαση μόνο στο ORM (Object Relational Mapping) και τα μοντέλα του. Η ORM υλοποίηση στην συνέχεια έχει άμεση πρόσβαση στο Σχεσιακή Βάση Δεδομένων εκτελώντας τα queries για την δημιουργία, επεξεργασία και διαγραφή εγγραφών.

Έτσι κρατάμε το σύστημα ασφαλές καθώς η επικοινωνία του client με τον server γίνεται μέσω ενός API το οποίο απαιτεί αυθεντικοποίηση.

2.2.5 Entity Relationship Diagram

Το μοντέλο Οντοτήτων Συσχετίσεων ή ERM είναι μια αφηρημένη και εννοιολογική αναπαράσταση των δεδομένων που χρησιμοποιείται από τους μηχανικούς λογισμικού. Η μοντελοποίηση ER είναι μια μέθοδος μοντελοποίησης βάσεων δεδομένων που χρησιμοποιείται για να παραχθεί το εννοιολογικό σχήμα (conceptual schema) μιας σχεσιακής βάση και των απαιτήσεών της. Η διαδικασία αυτή καταλήγει στο να παραχθεί μια διαγραμματική αναπαράσταση του σχήματος της βάσης δεδομένων που ονομάζεται διάγραμμα Οντοτήτων - Συσχετίσεων (Entity-Relationship diagram – ERD). Από αυτό το διάγραμμα σε συνδυασμό με το Class Diagram, που προαναφέρθηκε, εξαρτάται όλη η δομή και η δημιουργία της βάσης δεδομένων και του κώδικα. Γι' αυτό το λόγο η σχεδίαση του έπρεπε να είναι λεπτομερής και προσεχτική καθώς όποιο λάθος θα μπορούσε να προκαλέσει σφάλμα στη λειτουργία και φυσικά μεγάλη καθυστέρηση στην υλοποίηση του. Σε μια σχεσιακή βάση δεδομένων η αποθήκευση των δεδομένων γίνεται με τη μορφή πινάκων. Κάποια από τα δεδομένα των πινάκων υποδεικνύουν δεδομένα σε

άλλους πίνακες και με αυτόν τον τρόπο οι πίνακες συσχετίζονται. Η λογική αναπαράσταση αυτής της δομής αποδίδεται με το διάγραμμα Οντοτήτων – Συσχετίσεων.



Εικόνα 8 - ER Diagram

Με το ER Diagram μπορούμε να κατανοήσουμε καλύτερα την κατανομή και συσχέτιση μεταξύ των πινάκων της Βάσης Δεδομένων της εφαρμογής.

2.2.6 API

Η επικοινωνία του client με τον server και η παραλαβή των δεδομένων από την Βάση Δεδομένων γίνεται μέσω API.

Το μεγαλύτερο μέρος του API που έχει κατασκευαστεί είναι προστατευμένο και απαιτεί την αυθεντικοποίηση των χρηστών για την κλήση του.

Μερικά links όπως εκείνο για το login και register των χρηστών δεν απαιτούν καποια αυθεντικοποίηση.

Επίσης, έχει προστεθεί ένα επιπλέον μέτρο ασφαλείας στο API με στόχο την αποφυγή απο Authentication Hijacking, ή/και Data Exposure attacks. Με την χρήση middleware functions μπορεί να επιτραπεί η χρήση ενός endpoint του API μόνο από συγκεκριμένους ρόλους χρηστών. Έτσι ακόμα και αν ένας αυθεντικοποιημένος χρήστης προσπαθήσει να χρησιμοποιήσει άλλα API endpoints, δεν θα έχει πρόσβαση σε αυτά.

Συνεπώς οι middleware συναρτήσεις αυτές έχουν χρησιμοποιηθεί κατάλληλα σε όλα τα API Endpoints που χρησιμοποιεί ο client, για την αποφυγή διαρροής ευαίσθητης πληροφορίας.

2.2.7 Λειτουργικές & Μη Λειτουργικές Απαιτήσεις

Στο παρόν υποκεφάλαιο θα αναφερθούμε στις λειτουργικές και μη λειτουργικές απαιτήσεις του συστήματος.

Μία λειτουργική απαίτηση που θα πρέπει να αναλυθεί είναι εκείνη της διαχείρισης της πληροφορίας κατά την πιθανή αποχώρηση/διαγραφή των χρηστών από το σύστημα.

Σε διάφορα σημεία της υλοποίησης της εφαρμογής, κατά την διαγραφή των χρηστών, εμφανίζεται κατάλληλο μήνυμα στον χρήστη ως προς την ενημέρωσή και τελική συγκατάθεση του για την διαγραφή της πληροφορίας.

Κατά την οριστική διαγραφή ενός χρήστη διαγράφονται επίσης στοιχεία του χρήστη όπως το προσωπικά τους και οικονομικά του στοιχεία, οι πιθανές ανακοινώσεις του, τα προσωπικά γεγονότα του, οι αιτήσεις του και το ιστορικό θέσεων εργασίας του.

Σε αντίθεση με αυτά **δεν** διαγράφεται πληροφορία που θα επηρεάσει την συνολική εικόνα εξόδων και πληρωμών της εταιρείας/οργανισμού.

Συνεπώς **δεν** διαγράφεται το ιστορικό πληρωμών του (Payment).

Κατα την διαγραφή ενός τμήματος, γίνεται πρώτα έλεγχος για ύπαρξη χρηστών που βρίσκονται στο σύστημα. Αν το τμήμα έχει χρήστες τότε ο διαχειριστής πρέπει να μεταφέρει τους χρήστες σε διαφορετικό τμήμα προτού του επιτραπεί η διαγραφή του τμήματος.

Όσον αφορά τις μη λειτουργικές απαιτήσεις θα πρέπει το σύστημα που θα υλοποιηθεί να είναι **αποδοτικό** δηλαδή να έχει χαμηλό (response item). Επίσης θα πρέπει να είναι ασφαλές προσφέροντας τρόπους αυθεντικοποίησης (π.χ. api tokens). Τέλος θα πρέπει να είναι εύχρηστο προσφέροντας ένα απλό και κατανοητό user interface.

ΚΕΦ.3: Υλοποίηση Εφαρμογής

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκαν ένα σύνολο από γλώσσες προγραμματισμού, τεχνολογίες, libraries και frameworks.

Για τον server του συστήματος χρησιμοποιήθηκε το framework Node.js και η βιβλιοθήκη Express.js. Για τον client του συστήματος χρησιμοποιήθηκε η βιβλιοθήκη React.js. Στην βάση δεδομένων χρησιμοποιήθηκε MySQL.

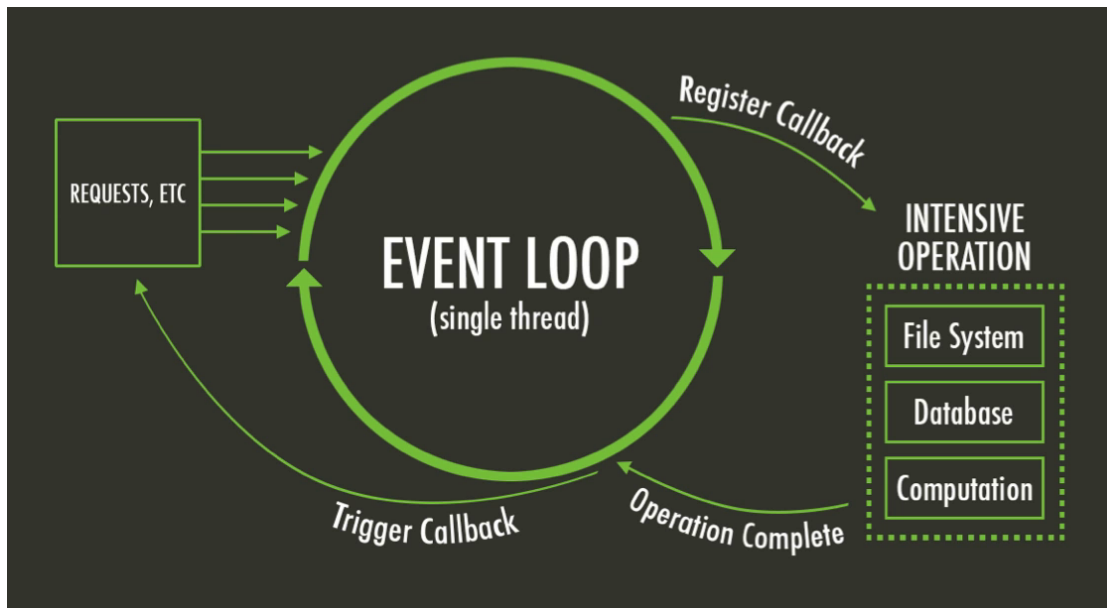
3.1 Node.js

Το Node.js είναι μια πλατφόρμα ανάπτυξης λογισμικού (κυρίως διακομιστών) χτισμένη σε περιβάλλον JavaScript. Στόχος του είναι να παρέχει ένα εύκολο τρόπο δημιουργίας κλιμακωτών διαδικτυακών εφαρμογών. Σε αντίθεση από τα περισσότερα σύγχρονα περιβάλλοντα ανάπτυξης εφαρμογών δικτύων μία διεργασία node δεν στηρίζεται στην πολυνηματικότητα (multithreading) αλλά σε ένα μοντέλο ασύγχρονης επικοινωνίας εισόδου/εξόδου. ^[12] Αυτού του είδους το μοντέλο λειτουργίας στοχεύει στη βελτίωση της διεκπεραιωτικής ικανότητας των διαδικτυακών εφαρμογών με πολλές λειτουργίες εισόδου/εξόδου, όπως και εφαρμογών ιστού πραγματικού χρόνου (προγράμματα επικοινωνίας πραγματικού χρόνου, παιχνίδια φυλλομετρητών). ^[13]

Η αρχιτεκτονική της πλατφόρμας φέρνει τον προγραμματισμό οδηγούμενο από γεγονότα (event – driven programming) στους εξυπηρετητές, επιτρέποντας την ανάπτυξη γρήγορων διακομιστών σε Javascript. ^[14] Ο οδηγούμενος απο γεγονότα προγραμματισμός είναι ένα μοντέλο στο οποίο η ροή του προγράμματος καθορίζεται από γεγονότα όπως δράσεις του χρήστη (κλικ ποντικιού, πάτημα κουμπιού), εξόδους αισθητήρων ή μηνύματα από άλλα προγράμματα/νήματα. Είναι το κυρίαρχο μοντέλο προγραμματισμού που χρησιμοποιείται στις γραφικές διεπαφές χρηστών και σε εφαρμογές επικεντρωμένες

στην εκτέλεση συγκεκριμένων ενεργειών ως απόκριση σε είσοδο του χρήστη. Σε οδηγούμενες από γεγονότα εφαρμογές υπάρχει συνήθως ένας βασικός βρόχος που αναμένει την εμφάνιση γεγονότων και στη συνέχεια πυροδοτεί μια συνάρτηση επανάκλησης (callback function), όταν εμφανιστεί κάποιο γεγονός. ^[15]

Με τη βοήθεια του μοντέλου που περιγράφεται παραπάνω, ο προγραμματιστής μπορεί να δημιουργήσει μεγάλης έκτασης εξυπηρετητές χωρίς τη χρήση της πολυνηματικότητας, αλλά με την εκμετάλλευση ενός απλοποιημένου μοντέλου. Το απλοποιημένο μοντέλο του οδηγούμενου από γεγονότα προγραμματισμού χρησιμοποιεί επανακλήσεις για να σηματοδοτήσει την ολοκλήρωση μιας διεργασίας. ^[14] Η πλατφόρμα της Node δημιουργήθηκε γιατί η παραλληλία είναι δύσκολα υλοποιήσιμη σε πολλές γλώσσες πραγματισμού εξυπηρετητών και συχνά οδηγεί σε μειωμένη απόδοση. Η ανάπτυξη της Node είναι βασισμένη στην ανοιχτού κώδικα μηχανή V8 Javascript της Google, διαθέτει εξαιρετική ταχύτητα και παρουσιάζει ευχέρεια στα βασικά διαδικτυακά πρωτόκολλα HTTP, DNS, TCP. ^[16] Τέλος η βάση της πλατφόρμας, η γλώσσα Javascript, είναι τόσο διαδεδομένη ώστε την κάνει άμεσα προσβάσιμη στο σύνολο της κοινότητας των προγραμματιστών διαδικτύου.



Εικόνα 9 - Αρχιτεκτονική NodeJS [14]

Μερικά πλεονεκτήματα της NodeJS είναι:

- Αυξημένη ταχύτητα. Όπως προαναφέρθηκε, η Node είναι μία πλατφόρμα ανάπτυξης λογισμικού που χρησιμοποιεί την μηχανή V8, η οποία κατασκευάστηκε από την Google για την ενσωμάτωση στον περιηγητή της, τον Chrome. Η συγκεκριμένη μηχανή μεταγλωττίζει και εκτελεί κώδικα JavaScript σε εξαιρετικές ταχύτητες κυρίως λόγω του γεγονότος ότι ο μεταγλωττιστής της μετατρέπει απευθείας την Javascript σε κώδικα μηχανής.
- Προσφέρει το βρόχο γεγονότων. Ο βρόχος γεγονότων είναι ένα μονό νήμα που εκτελεί ασύγχρονα αξιοποιώντας παράλληλα νήματα. Αυτή η προσέγγιση τείνει να ξεπεραστεί λόγω της αυξημένης μνήμης που απαιτεί και της φήμης της στη δυσκολία προγραμματισμού. Αντίθετα, όταν μια Node εφαρμογή απαιτεί την εκτέλεση μιας λειτουργίας εισόδου/εξόδου αποστέλλει μια ασύγχρονη εργασία στο βρόχο γεγονότων, μαζί με μια συνάρτηση επανάκλησης και συνεχίζει την κανονική ροή του προγράμματος. Τέλος, όταν ολοκληρωθεί η ασύγχρονη διεργασία, ο βρόχος γεγονότων επιστρέφει στη διαδικασία και εκτελεί την ανάκλησή της.

- Παρέχει την εργαλειοθήκη npm. Η συγκεκριμένη βιβλιοθήκη σαφώς δεν είναι μοναδική στη Node και θυμίζει εργαλειοθήκες από άλλα οικοσυστήματα, αλλά κάνει εξαιρετική δουλειά στην αναγνώριση και εγκατάσταση εξαρτήσεων των διαφόρων εργασιών. Επίσης τα πακέτα εγκατάστασης εξαρτήσεων κρατούνται απομονωμένα μεταξύ εργασιών και έτσι αποφεύγονται συγκρούσεις εκδόσεων.
- Βασίστηκε σε μία ήδη διαδεδομένη γλώσσα, τη Javascript. Τα πιο διαδεδομένα πλαίσια ανάπτυξης εφαρμογών στην πλευρά του πελάτη (client-side) στηρίζονται κατα κόρον, αν όχι αποκλειστικά στη λογική της Javascript. Με τη Node δε χρειάζεται πλέον η μετάφραση της λογικής της πλευράς πελάτη σε αυτή της πλευράς του εξυπηρετητή, αφού είναι κοινή. Επίσης δεν απαιτείται η μετάφραση των HTTP δεδομένων που στέλνονται σε διαφορετικά αντικείμενα στην πλευρά του εξυπηρετητή.

3.2 React Js

Η React παρουσιάστηκε ως έργο ανοιχτού κώδικα τον Μάιο του 2013.

Πρόκειται για μια Javascript βιβλιοθήκη για την δημιουργία διεπαφών χρήστη. Σύμφωνα με το Facebook, πρόκειται για ένα πλαίσιο λογισμικού προσανατολισμένο στο View (απο το μοντέλο MVC). Δεν απαιτεί καμία εξάρτηση (dependency) για να λειτουργήσει σωστά.

Ως πιο νέο πλαίσιο λογισμικού, είναι πιο γρήγορο από άλλες βιβλιοθήκες ή libraries και πιο εύκολο στην εκμάθηση. Ακόμη ένα ισχυρό χαρακτηριστικό της React είναι ότι κάθε δομικό στοιχείο (component) της αντιπροσωπεύει ένα στοιχείο της διεπαφής χρήστη (user interface) - ένα στοιχείο φόρμας/πίνακα, έναν τίτλο σελίδας κλπ.

Η React δημιουργεί το δικό της DOM (εικονικό) όπου λειτουργούν τα components. Αυτό δίνει στους προγραμματιστές μεγάλη ευελιξία και εκπληκτικά κέρδη απόδοσης, επειδή υπολογίζει την μεταβολή που απαιτείται για να γίνει στο DOM εκ των προτέρων. Έτσι αποφεύγει τις δαπανηρές λειτουργίες του DOM και ενημερώνει αποτελεσματικά.

Η React χρησιμοποιεί σύνταξη JSX, ένα συνονθύλευμα Javascript και HTML. Το JSX απλοποιεί ολόκληρη τη διαδικασία σύνταξης components για τις ιστοσελίδες.

Μια από τις μεγαλύτερες προκλήσεις με τα frameworks/libraries της Javascript είναι ότι δεν είναι καθόλου φιλικά προς τις μηχανές αναζήτησης (Search Engine Optimization / SEO). Παρόλο που πρόσφατα έγιναν ορισμένες βελτιώσεις στον τομέα, δεν βοήθησε και τόσο. Παραδόξως, η React ξεχωρίζει, καθώς μπορεί και να εκτελεστεί στο server (server side rendering) και το εικονικό DOM θα επιστραφεί και θα αποτυπωθεί στο πρόγραμμα περιήγησης ως κανονική σελίδα, βελτιωμένη αυτή τη φορά ως προς το Search Engine Optimization.

Αυτό που είναι αξιοσημείωτο στη React είναι ότι προσφέρει τη δυνατότητα επαναχρησιμοποίησης των components οποιαδήποτε στιγμή. Αυτό είναι πολύ σημαντικό αποτέλεσμα εξοικονόμησης χρόνου. Τα components στη React είναι απομονωμένα και η αλλαγή σε ένα δεν επηρεάζει τα υπόλοιπα. Αυτό επιτρέπει στους προγραμματιστές να επαναχρησιμοποιήσουν τα components που δεν παράγουν αλλαγές και καθιστούν τον προγραμματισμό πιο ακριβή, εργονομικό και άνετο για αυτούς. ^[18]

3.3 Express.js

Το Express.js ή απλά Express, είναι ένα framework που χρησιμοποιήθηκε στο περιβάλλον Node.js για την κατασκευή του Restful API. ^[19]

3.4 MySQL

Η δομημένη γλώσσα ερωταποκρίσεων SQL (Structured Query Language, SQL) χρησιμοποιείται για τη διαχείριση των δεδομένων μιας βάσης δεδομένων. Η διαχείριση περιλαμβάνει τόσο τη δημιουργία και μεταβολή των πινάκων της εφαρμογής, όσο και την καταχώρηση και ανάκτηση δεδομένων με βάση συγκεκριμένα κριτήρια επιλογής. Μια τυπική γλώσσα SQL περιλαμβάνει τις επόμενες δομικές μονάδες:

Γλώσσα ορισμού δεδομένων (**Data Definition Language, DDL**): Η γλώσσα αυτή περιλαμβάνει εντολές που μας επιτρέπουν να υλοποιήσουμε πίνακες, σχέσεις ανάμεσα σε πίνακες και γενικά όλη τη δομή μιας βάσης δεδομένων.

Γλώσσα χειρισμού δεδομένων (**Data Manipulation Language, DML**): Η γλώσσα αυτή επιτρέπει τη διαχείριση των δεδομένων της εφαρμογής, όπως την εισαγωγή, διαγραφή, ανάκτηση και τροποποίηση δεδομένων.

Ορισμός όψεων της βάσης (**View Definition**): Επιτρέπει τη δημιουργία όψεων της βάσης δεδομένων οι οποίες ορίζονται ως εικονικοί πίνακες (**virtual tables**) οι οποίοι περιέχουν δεδομένα από ένα ή περισσότερους πίνακες της βάσης.

Ορισμός εξουσιοδοτήσεων (**Authorization**): Επιτρέπει τη δημιουργία ομάδων χρηστών και την απόδοση διαφορετικών δικαιωμάτων πρόσβασης σε κάθε έναν από αυτούς, προκειμένου η κάθε ομάδα χρηστών, να διαχειρίζεται μόνο τα δικά της δεδομένα.

Διαχείρισης ακεραιότητας (**Integrity**): Επιτρέπει το λεπτομερή έλεγχο των δεδομένων που καταχωρούνται στη βάση, έτσι ώστε να μην παραβιάζονται οι κανόνες ακεραιότητας (**integrity constraints**) που έχουμε ορίσει και οι οποίοι όταν τηρούνται απομακρύνουν τον κίνδυνο καταχώρησης ασυνεπών δεδομένων (**inconsistent data**).

Η γλώσσα διαχείρισης δεδομένων (**Data Control Language, DCL**): χειρίζεται τις εξουσιοδοτήσεις των δεδομένων

Η γλώσσα χειρισμού δεδομένων (Data Manipulation Language DML), επιτρέπει την διαχείριση των δεδομένων των πινάκων της βάσης, και πιο συγκεκριμένα, την εισαγωγή, διαγραφή και τροποποίηση των εγγραφών των πινάκων. Επιπλέον, έχουμε την δυνατότητα να ανακτήσουμε από τους πίνακες, δεδομένα τα οποία πληρούν κάποια κριτήρια. Η πραγματοποίηση αυτών των διαδικασιών, γίνεται χρησιμοποιώντας τις εντολές INSERT, DELETE, UPDATE και SELECT.

3.5 Sequelize

Το Sequelize είναι ένα promise-based ORM (Object-Relational Mapping) που υποστηρίζει PostgreSQL, MySQL (που χρησιμοποιείται στην συγκεκριμένη περίπτωση), SQLite και MSSQL. [20]

3.6 Bootstrap

Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα (Ελεύθερο λογισμικό) για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript. Είναι το πιο δημοφιλές πρόγραμμα στο GitHub και έχει χρησιμοποιηθεί από τη NASA και το MSNBC, μεταξύ άλλων.

Το Bootstrap αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter ως ένα πλαίσιο για την ενθάρρυνση της συνέπειας στα εσωτερικά εργαλεία. [21]

3.7 React Material Table

Κατά την υλοποίηση του συστήματος, δημιουργήθηκε η ανάγκη για χρήση πινάκων για την προβολή λίστας πληροφοριών (προβολή λίστας υπαλλήλων, θέσεων εργασίας, κτλ).

Καθώς μπορεί το σύστημα να κατέχει μεγάλο μέγεθος πληροφορίας για τους υπαλλήλους, είναι αναγκαία η ύπαρξη λειτουργιών pagination, φιλτραρίσματος και ταξινόμησης, . Δηλαδή θα πρέπει ο χρήστης να έχει την δυνατότητα να ελέγχει το όριο των εγγραφών ανά μία σελίδα πίνακα, να ταξινομεί τον πίνακα ως προς όποια στήλη του επιθυμεί, αλλά και να μπορεί να κάνει αναζήτηση.

Έπειτα από εκτενή μελέτη όλων των υπάρχοντων βιβλιοθηκών, αλλά και την επιτυχή προσπάθεια της δικής μου υλοποίησης πινάκων, αποφασίστηκε να χρησιμοποιηθεί η βιβλιοθήκη React Material Table, η οποία συνδυάζει άριστα την ύπαρξη όλων των λειτουργιών που αναφέρθηκαν. [22]

3.8 JSON Web Token

Τα JSON Web Tokens αποτελούν ένα standard μιας και η πλατφόρμα που φέρουν μεταδίδεται μέσω JSON.

Τα JWTs περιέχουν μέσα τους όλη την απαραίτητη πληροφορία. Αυτό σημαίνει ότι ένα JWT είναι σε θέση να μεταφέρει βασική πληροφορία για αυτό, το φορτίο του (που συνήθως είναι πληροφορίες που σχετίζονται με έναν χρήστη) και μια υπογραφή.

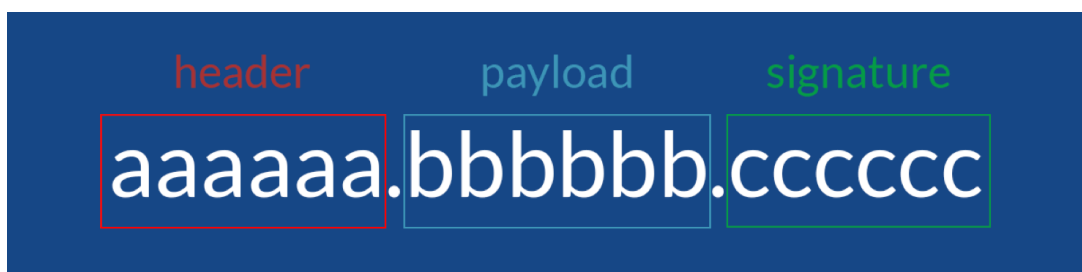
Τα JWTs μπορούν να μεταφερθούν εύκολα. Από τη στιγμή που τα JWTs είναι αυτόνομα, μπορούν να χρησιμοποιηθούν μέσα σε HTTP headers όταν πιστοποιούν ένα API. Μπορούν επίσης να σταλούν μέσω του URL.

Είναι εύκολο να αναγνωρισθεί ένα JWT. Είναι 3 αλφαριθμητικά χωρισμένα με τελεία. Για παράδειγμα: aaaaaaaaaa.bbbbbbbbbbbb.ccccccccccccccc.

Ας δούμε την σημασία του κάθε πεδίου.^[23]

Από τη στιγμή που έχουμε 3 πεδία, καθένα από αυτά δημιουργείται διαφορετικά. Τα πεδία αυτά είναι:

- Header
- Payload
- Signature



Εικόνα 10 - JSON Web Token

Ο header περιλαμβάνει 2 μέρη:

- αναφορά του τύπου, που είναι JWT
- ο hashing αλγόριθμος που χρησιμοποιείται

Ένα παράδειγμα φαίνεται παρακάτω: {

```
"typ": "JWT",
"alg": "HS256"
}
```

Επειδή αυτό το κομμάτι κωδικοποιείται (base64encode), έχουμε το πρώτο μέρος του JWT που μοιάζει κάπως έτσι: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 .

Το payload μεταφέρει το κυρίως φορτίο του JWT. Εδώ είναι που μπαίνει η πληροφορία που θέλουμε να μεταδώσουμε καθώς και άλλες πληροφορίες σχετικές με το token.

Ένα παράδειγμα του payload φαίνεται παρακάτω: {

```
"iss": "scotch.io",
"exp": 1300819380,
"name": "Chris Sevilleja", "admin": true
}
```

Το οποίο κωδικοποιείται σε

```
eyJpc3MiOiJzY290Y2guaW8iLCJleHAiOiJzMDA4MTkzODAsIm5hbWUiOiJDaHJpcyBTZXZpbGxlamEiLCJhZG1pbil6dHJ1ZX0 .
```

Αυτό αποτελεί το δεύτερο μέρος του JWT.

Το τρίτο και τελευταίο μέρος αποτελεί η υπογραφή. Αυτό το μέρος αποτελείται από τη hash τιμή των header, payload και ενός μυστικού κλειδιού. Το κλειδί αυτό βρίσκεται στο server. Έτσι ο server είναι σε θέση να πιστοποιήσει υπάρχοντα tokens και να δημιουργήσει νέα. Έτσι προκύπτει και το τελευταίο μέρος του token:

```
03f329983b86f7d9a9f5fef85305880101d5e302afafa20154d094b229f75773 .
```

3.9 Επιπλέον Modules

- Chart.js - Για την δημιουργία διαγραμμάτων στην ιστοσελίδα
- Moment - Χρησιμοποιείται για το format των ημερομηνιών.
- Axios - Promise based client για HTTP Requests
- ReactDatePicker - Για τα ημερολόγια του συστήματος

ΚΕΦ.4: Εγκατάσταση Εφαρμογής

Θα πρέπει να έχετε εγκατεστημένο το framework Node.js για την επιτυχή εκτέλεση των εντολών και την εκκίνηση του συστήματος. Το Node.js μπορείτε να το εγκαταστήσετε στο σύστημα σας από αυτόν τον [ιστότοπο](#).

Για την εγκατάσταση της εφαρμογής θα πρέπει να προβείτε στο αποθετήριο κώδικα μέσω αυτού του [συνδέσμου](#), και θα κάνετε clone (κατέβασμα) τον κώδικα.

Εναλλακτικά, μπορείτε να κάνετε clone το project μέσω command line και της εντολής git.

```
git clone https://github.com/vasilismantz/thesis-fullstack.git
```

Στην συνέχεια μεταβείτε στον φάκελο που κατεβάσατε τον κώδικα με την χρήση των εντολών cd. Αρχικά, θα πρέπει να εγκαταστήσουμε όλα τα node_modules.

Εκτελούμε την εντολή

```
npm install
```

Στην συνέχεια μεταβαίνουμε στον φάκελο server και εκτελούμε την ίδια εντολή

```
cd server && npm install
```

Έπειτα μεταβαίνουμε και στον φάκελο client και εκτελούμε την ίδια εντολή

Αν είμαστε στον φάκελο server μεταβαίνουμε πρώτα πίσω στον αρχικό φάκελο με την εντολή `cd ..`

```
cd client && npm install
```

Για την εκκίνηση της εφαρμογής μεταβαίνουμε στον αρχικό φάκελο και εκτελούμε την εντολή

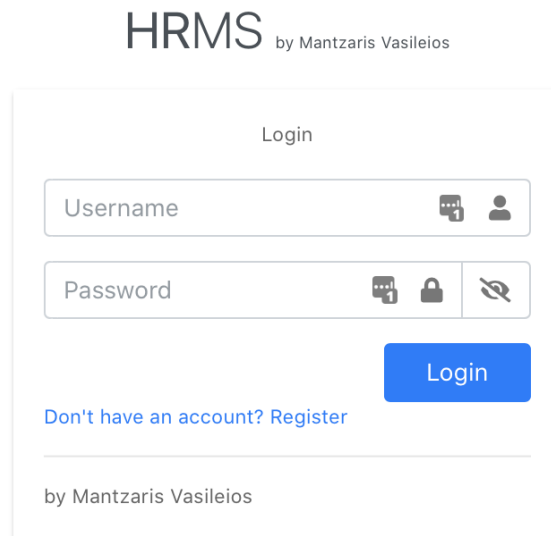
```
npm start
```

Η είσοδος στο σύστημα γίνεται από το [127.0.0.1:3000](#)

ΚΕΦ.5: Σενάρια Χρήσης

5.1 Γενικά

Κατά την πλοήγηση του χρήστη στο 127.0.0.1:3000 συναντάει το login screen του συστήματος.



The screenshot shows the login interface for 'HRMS by Mantzaris Vasileios'. It features a central white box with a light gray border. At the top of the box, the word 'Login' is centered. Below it are two input fields: 'Username' and 'Password'. The 'Username' field has a small icon of a person and a speech bubble with the number '1'. The 'Password' field has icons for a speech bubble with '1', a lock, and an eye with a slash. To the right of the password field is a blue 'Login' button. Below the button is a link that says 'Don't have an account? Register'. At the bottom of the box, it says 'by Mantzaris Vasileios'.

Εικόνα 11 - Login Screen

Στην σελίδα αυτή ο χρήστης του συστήματος έχει την δυνατότητα να βάλει τα στοιχεία του και να εισέλθει στην εφαρμογή.

Σε περίπτωση που δεν έχει λογαριασμό ο χρήστης έχει 2 επιλογές:

- Με την επιλογή “Don’t have an account? Register” μπορεί να κάνει εγγραφή με τα στοιχεία του, και να περιμένει από τον διαχειριστή να ενεργοποιήσει το λογαριασμό του, δίνοντας του έτσι πρόσβαση στο σύστημα. Ο λογαριασμός που δημιουργείται από αυτή την διαδικασία είναι απλού Υπαλλήλου.

Register

Username

Fullname

Password

Retype Password

Register

[Already have an account? Login](#)

by Mantzaris Vasileios

Εικόνα 12 - Register Screen

- Μπορεί να ενημερώσει τον διαχειριστή του συστήματος (δια ζώσης ή μέσω email, τηλεφώνου) ώστε να δημιουργήσει εκείνος το λογαριασμό μέσα από την εφαρμογή. Αυτός ο τρόπος προτείνεται για δημιουργία λογαριασμών Διευθυντών ή Διαχειριστών.

Στην συνέχεια κατα τη σύνδεση του υπαλλήλου στο σύστημα υπάρχουν 3 σενάρια.

- Κατά την είσοδο λανθασμένων στοιχείων εμφανίζεται το ακόλουθο μήνυμα

ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΑΝΘΡΩΠΙΝΟΥ ΔΥΝΑΜΙΚΟΥ / ΜΑΝΤΖΑΡΗΣ ΒΑΣΙΛΕΙΟΣ

49

HRMS

by Mantzaris Vasileios

Incorrect Credentials!

Login

vasmantz

...

Login

[Don't have an account? Register](#)

by Mantzaris Vasileios

Εικόνα 13 - Incorrect Login

- Κατα την είσοδο λογαριασμού ο οποίος δεν έχει ενεργοποιηθεί εμφανίζεται το ακόλουθο μήνυμα

HRMS

by Mantzaris Vasileios

Account is not active!

Login

vasmantz

....

Login

[Don't have an account? Register](#)

by Mantzaris Vasileios

Εικόνα 14 - Inactive Account

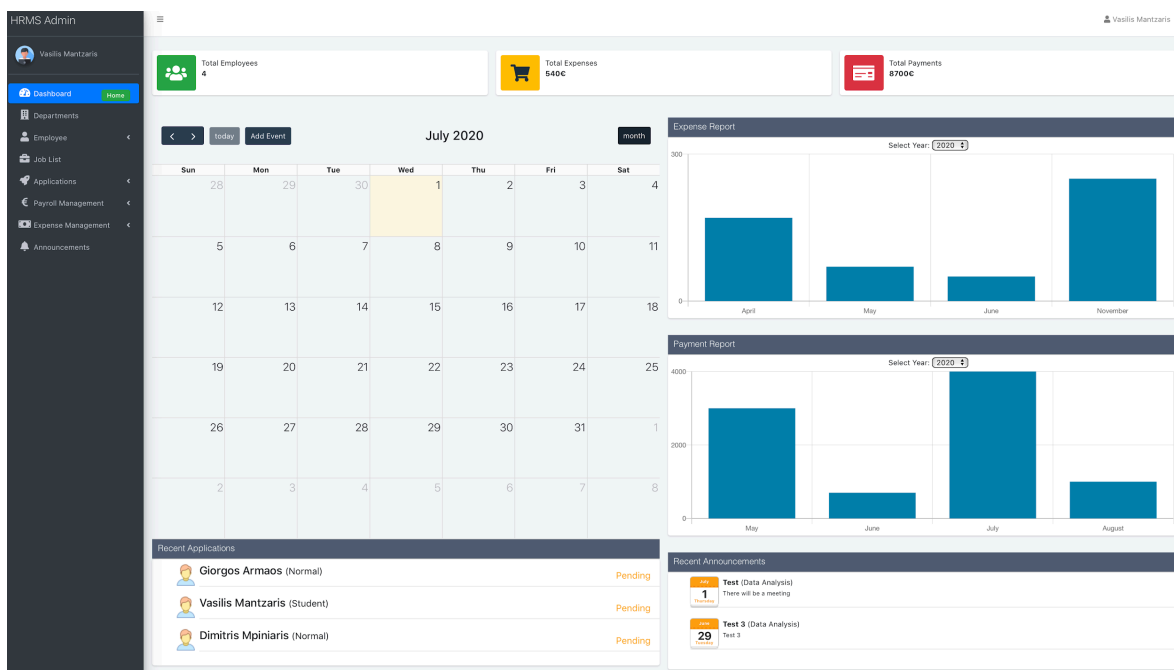
- Σε αντίθετη περίπτωση γίνεται είσοδος του χρήστη στο σύστημα

Κατα την είσοδο του Διαχειριστή στο σύστημα, βλέπει την σελίδα **Dashboard**. Στην σελίδα αυτή υπάρχει μια συλλογή πληροφοριών με στόχο την γρήγορη ενημέρωση του χρήστη για την κατάσταση του συστήματος.

Στο αριστερό μέρος υπάρχει πάντα το Sidebar (κάθετη μπάρα πλοήγησης) το οποίο βοηθάει για την περιήγηση του χρήστη στην εφαρμογή.

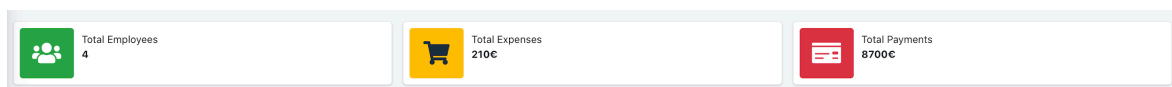
Η ανάλυση των σεναρίων θα γίνει σε 3 κατηγορίες λόγω των τριών διαφορετικών οθονών που έχουν δημιουργηθεί, όσοι και οι διακριτοί ρόλοι του συστήματος.

5.2 Διαχειριστής (Admin)



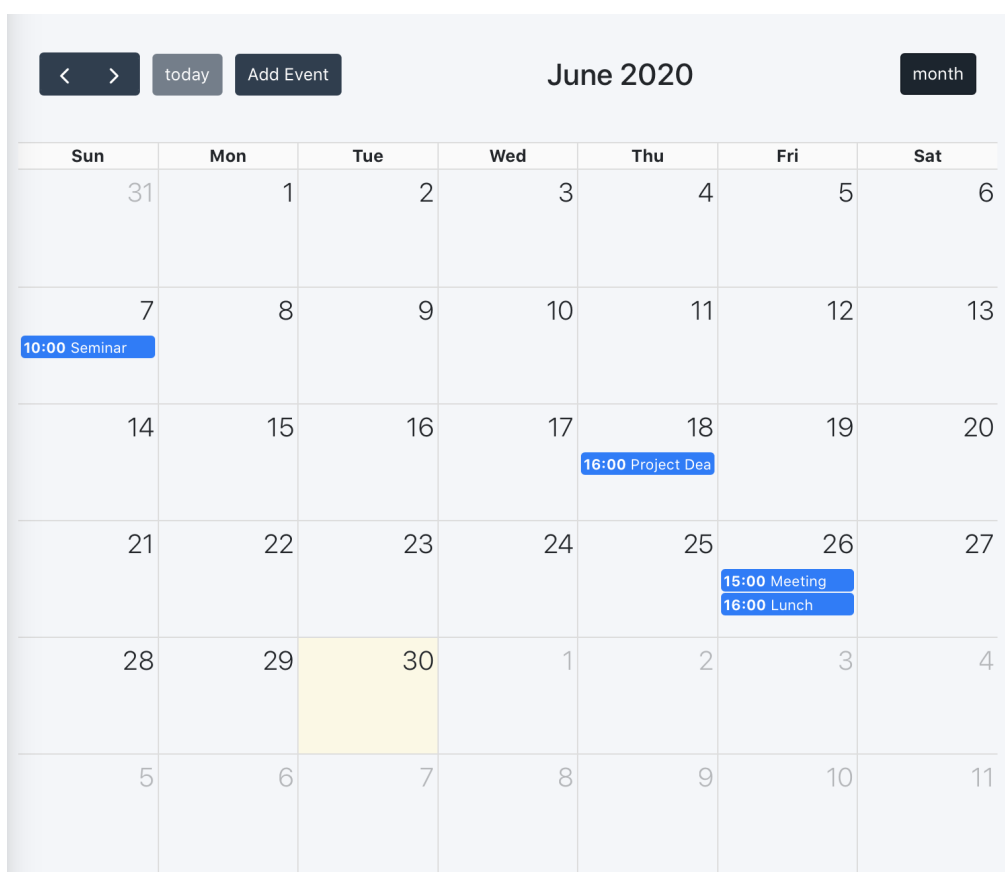
Εικόνα 15 - Dashboard Admin

Στο πάνω μέρος υπάρχουν 3 Infoboxes τα οποία απεικονίζουν πληροφορίες όπως το σύνολο των χρηστών του συστήματος, το σύνολο των εξόδων της εταιρείας και το σύνολο των πληρωμών.



Εικόνα 16 - Infoboxes

Στην συνέχεια υπάρχει ένα ημερολόγιο, μέσω του οποίου ο χρήστης μπορεί να προσθέσει, να επεξεργαστεί και να διαγράψει τα προσωπικά του γεγονότα. Κατά την επιλογή μια ημερομηνίας ή την επιλογή Add Event που βρίσκεται στο πάνω μέρος του Ημερολογίου, ο χρήστης μπορεί να προσθέσει ένα προσωπικό γεγονός.



Εικόνα 17 - Calendar

The image shows a modal window titled "Add Event" with a close button (X) in the top right corner. The form contains the following fields:

- Title ***: A text input field with the placeholder "Enter a Title" and a calendar icon on the right.
- Description**: A text input field with the placeholder "Enter a Description".
- Start Date ***: A date selection field with the placeholder "Select Start Date".
- End Date ***: A date selection field with the placeholder "Select End Date".

Below these fields is a label "Required Fields *" and a green "Submit" button. In the bottom right corner of the modal is a blue "Close" button. The background shows a calendar interface for June 2020, with a "month" button and an "Expense Report" tab visible.

Εικόνα 18 - Add Event



Θα πρέπει να συμπληρωθεί ο Τίτλος, ημέρα και ώρα έναρξης και λήξης. Η ημερομηνία λήξης θα πρέπει να είναι μετά την ημερομηνία έναρξης. Σε αντίθετη περίπτωση εμφανίζεται κατάλληλο μήνυμα.

Στην δεξιά πλευρά βρίσκονται δύο διαγράμματα τύπου Bar Chart τα οποία απεικονίζουν τα έξοδα και τις πληρωμές της εταιρείας ανα μήνα. Ο χρήστης έχει επίσης την επιλογή, αλλάζοντας την χρονιά, να βλέπει δεδομένα προηγούμενων χρόνων για γρήγορη σύγκριση.





Εικόνα 19 - Bar Charts

Στην κάτω αριστερά πλευρά υπάρχει ο χώρος για τις πρόσφατες αιτήσεις. Εμφανίζονται μόνο ο τύπος της αίτησης, ποιος χρήστης την κατέθεσε και η κατάσταση της. Οι αιτήσεις που εμφανίζονται στον χώρο αυτό είναι οι αιτήσεις μέσα σε διάστημα των 2 εβδομάδων.

Recent Applications		
	Giorgos Armaos (Normal)	Pending
	Vasilis Mantzaris (Student)	Pending

Εικόνα 20 - Recent Applications

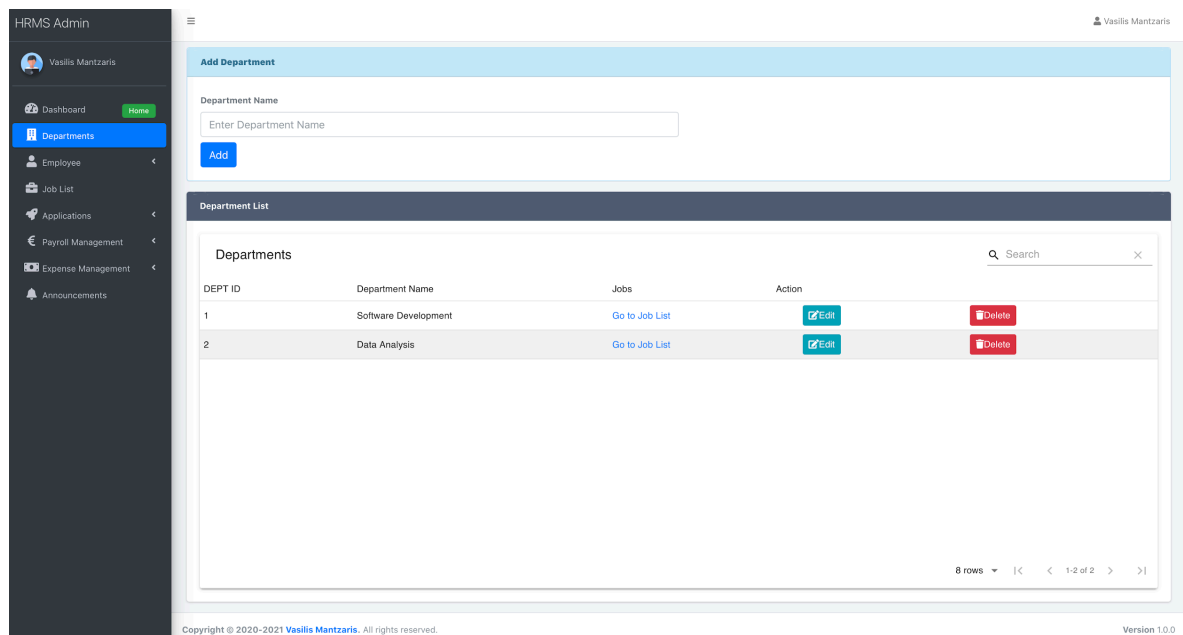
Στην κάτω δεξιά πλευρά υπάρχει ο χώρος για τις πρόσφατες ανακοινώσεις. Έτσι ο χρήστης μπορεί να δει απο την αρχική του οθόνη τις 2 πιο πρόσφατες ανακοινώσεις της εταιρείας.

Recent Announcements	
	Test (Data Analysis) There will be a meeting
	Test 3 (Data Analysis) Test 3

Εικόνα 21 - Recent Announcements

Τμήματα

Στην σελίδα αυτή ο διαχειριστής μπορεί να προσθέσει ένα τμήμα, να δει την λίστα με το σύνολο των τμημάτων της εταιρείας αλλά και να επεξεργαστεί και να διαγράψει τμήματα.



Εικόνα 22 - Departments

Επίσης υπάρχει ο σύνδεσμος “Go to job List” ο οποίος ανακατευθύνει τον χρήστη στην λίστα θέσεων εργασίας του εκάστοτε τμήματος.

Υπάλληλοι

Η διαχείριση των υπαλλήλων έχει χωριστεί σε 2 σελίδες.

Η πρώτη σελίδα αφορά την προσθήκη ενός νέου υπαλλήλου στο σύστημα.

HRMS Admin

Vasilis Mantzaris

Dashboard **Home**

Departments

Employee

Add Employee

Employee List

Job List

Applications

Payroll Management

Expense Management

Announcements

Personal Details

First Name *
Enter first Name

Last Name *
Enter last Name

Date of Birth *
Select Date Of Birth

Gender *
Choose...

Marital Status *
Choose...

Father's name *
Enter Father's Name

ID Number *
Enter ID Number

Bank Information

Bank Name
Enter Bank name

Account Name
Enter Account name

Account Number
Enter Account number

IBAN
Enter Iban

Job

Job Title *
Enter Job Title

Start Date *
Select Date Of Birth

End Date *
Select Date Of Birth

Contact Details

Physical Address *
Enter Address

Country *
Enter Country

City *
Enter City

Mobile *
Enter Mobile

Phone
Enter Phone

Email *
Enter Email

Official Status

Employee ID *
Enter Username

Password *
Enter Password

Department *
Choose...

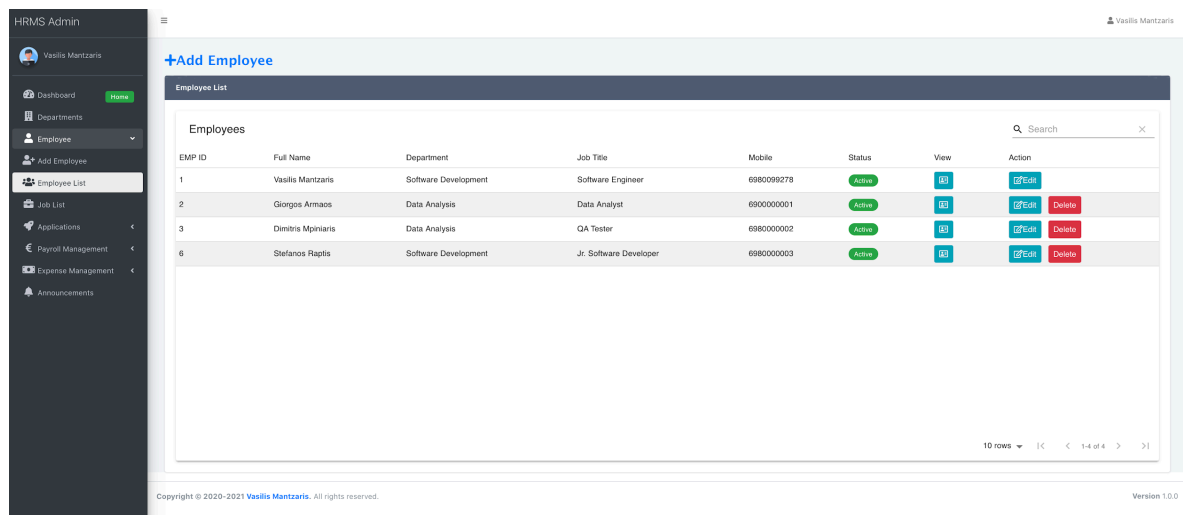
Role *
Choose...

Submit

Εικόνα 23 - Add Employee

Στην σελίδα αυτή ο διαχειριστής μπορεί να εισάγει όλα τα στοιχεία του υπαλλήλου, καθώς και την θέση εργασίας που θα κατέχει στην εταιρεία. Τα υποχρεωτικά πεδία διαχωρίζονται από τους κόκκινους αστερίσκους. Σε αυτό το βήμα δεν προστίθενται τα οικονομικά στοιχεία του χρήστη όπως ο μισθός του. Αυτό γίνεται ανεξάρτητα, σε επόμενο βήμα.

Η δεύτερη σελίδα αφορά την προβολή της λίστας όλων των υπαλλήλων.



Εικόνα 24 - Employee List

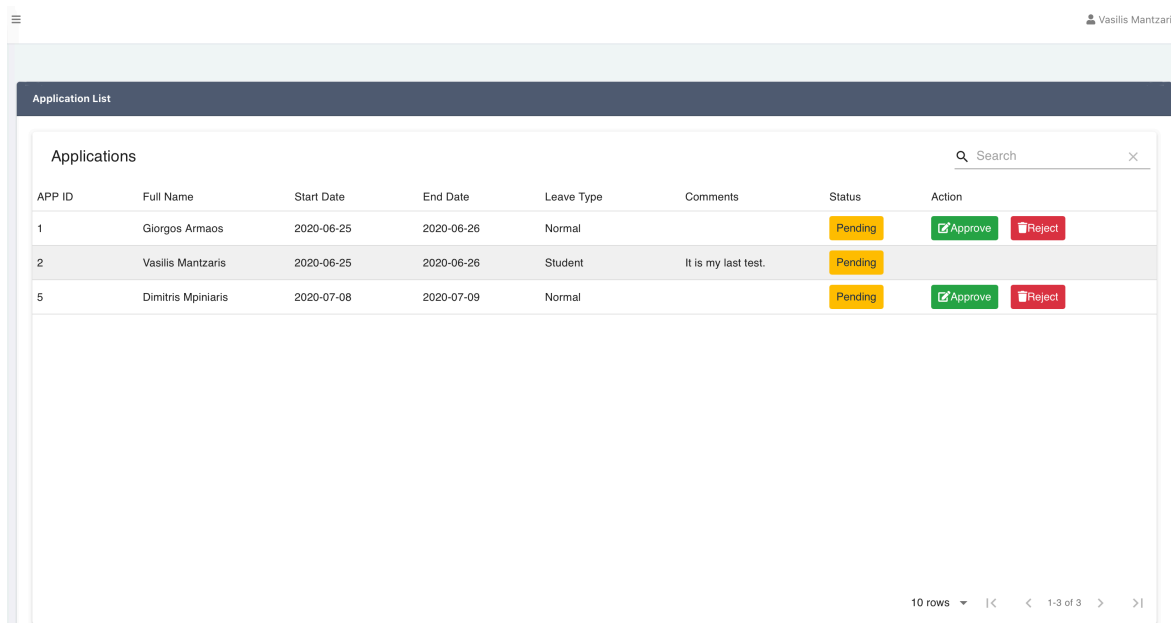
Στην σελίδα αυτή ο διαχειριστής μπορεί να έχει εικόνα για το σύνολο των υπαλλήλων, την κατάσταση των λογαριασμών τους (ενεργός/μη ενεργός), την επεξεργασία και την διαγραφή τους. Επίσης ο διαχειριστής μπορεί να επιλέξει το εικονίδιο με την καρτέλα, για να γίνει ανακατεύθυνση στην καρτέλα του υπαλλήλου, η οποία προβάλλει το σύνολο των δεδομένων του.

Θέσεις Εργασίας

Στην σελίδα αυτή ο διαχειριστής βλέπει την λίστα των θέσεων εργασίας της εταιρείας. Αρχικά θα πρέπει να επιλέξει το τμήμα για το οποίο ενδιαφέρεται, ή να παραλείψει την κατηγοριοποίηση ανα τμήμα επιλέγοντας “All Departments”. Επίσης έχει την επιλογή για προσθήκη μιας θέσης εργασίας μέσω της ένδειξης “Add Job”. Η θέση εργασίας θα πρέπει να προστίθεται σε κάποιο υπάρχων υπάλληλο.

Εικόνα 26 - Add Application

Στην δεύτερη σελίδα ο διαχειριστής μπορεί να δει την λίστα με τις αιτήσεις των υπαλλήλων της εταιρείας και να τις εγκρίνει/απορρίψει. Είναι σημαντικό να σημειώσουμε πάλι πως δεν δίνεται η δυνατότητα στους διαχειριστές και διευθυντές να εγκρίνουν τις δικές τους αιτήσεις.



APP ID	Full Name	Start Date	End Date	Leave Type	Comments	Status	Action
1	Giorgos Armaos	2020-06-25	2020-06-26	Normal		Pending	<button>Approve</button> <button>Reject</button>
2	Vasilis Mantzaris	2020-06-25	2020-06-26	Student	It is my last test.	Pending	
5	Dimitris Mpinariis	2020-07-08	2020-07-09	Normal		Pending	<button>Approve</button> <button>Reject</button>

Εικόνα 27 - Application List

Διαχείριση Λεπτομερειών Μισθού

Η διαχείριση λεπτομερειών μισθού έχει χωριστεί σε 3 σελίδες.

Στην πρώτη σελίδα ο διαχειριστής, επιλέγοντας το τμήμα και τον υπάλληλο που τον ενδιαφέρει μπορεί να διαχειριστεί της λεπτομέρειες των οικονομικών του στοιχείων όπως μισθός, φόροι και επιδόματα.

Manage Salary Details

Select Department:

All departments

Select User:

Vasilis Mantzaris

Salary Details

Employment Type *

Full Time

Basic Salary *

3000

Allowances

House Rent Allowance

400

Medical Allowance

0

Special Allowance

0

Fuel Allowance

0

Phone Bill Allowance

0

Other Allowance

0

Deductions

Tax Deduction

1000

Other Deduction

0

Total Salary Details

Gross Salary

3400

Total Deductions

1000

Net Salary

2400

Εικόνα 28 - Salary Details

Στην δεύτερη σελίδα μπορεί να δει την λίστα με τα οικονομικά στοιχεία όλων των θέσεων εργασίας, άρα κατ' επέκταση και των υπαλλήλων, της εταιρείας. Επίσης έχει την δυνατότητα, επεξεργασίας αλλά και προβολής της οικονομικής καρτέλας του υπαλλήλου, η οποία δείχνει με περισσότερη λεπτομέρεια όλα τα οικονομικά του στοιχεία.

List of Employees and Their Salaries

Employees

Search

X

EMP ID	Full Name	Gross Salary	Deductions	Net Salary	Emp Type	View	Action
1	Vasilis Mantzaris	3400	1000	2400	Full Time		
2	Giorgos Armaos	1200	300	900	Full Time		
3	Dimitris Mpiniaris	1000	300	700	Full Time		
6	Stefanos Raptis	2000	500	1500	Part Time		

10 rows

<

<

1-4 of 4

>

>

Εικόνα 29 - Salary List

Στην τρίτη σελίδα ο διαχειριστής μπορεί να καταχωρήσει μια πληρωμή μισθού στο σύστημα. Πιο συγκεκριμένα επιλέγει το τμήμα και τον υπάλληλο που τον ενδιαφέρει, καθώς και το μήνα της πληρωμής. Στην συνέχεια εμφανίζεται το ιστορικό του χρήστη, με συμπληρωμένα τα οικονομικά στοιχεία του χρήστη. Η μόνη αλλαγή που μπορεί να γίνει είναι η προσθήκη κάποιου πιθανού προστίμου και σχολίου στην πληρωμή.

Payment Month	Payment Date	Gross Salary	Department	Net Salary	Fine Deduction	Payment Amount
July-2020	27-Jun-20	3400	1000	2400	0	3000
May-2020	27-Jun-20	3400	1000	2400	0	3000

Εικόνα 30 - Make Payment

Έξοδα Τμημάτων

Η διαχείριση των εξόδων των τμημάτων χωρίζεται σε 3 σελίδες.

Στην πρώτη σελίδα ο διαχειριστής ή ο διευθυντής προσθέτει έξοδα στα τμήματα (γραφεία, καρέκλες ...).

Add Expense

Item Name:

Purchased From:

Purchase Date: Pick Date

Amount Spent:

Select Department: Choose one...

[Save](#)

Εικόνα 31 - Add Expense

Μέσω της δεύτερης σελίδας έχουν πρόσβαση στη λίστα με τα έξοδα των τμημάτων, φιλτραρισμένα ανά μήνα.

+Add Expense

Expenses

Select Date: Jun 2020 [Search](#)

Expense Report

ID	Item Name	Purchased From	Purchase Date	Department	Amount
1	Test	Test	25 Jun,2020	Software Development	€ 20
2	Test 2	Test 2	25 Jun,2020	Data Analysis	€ 20
3	Test	test	26 Jun,2020	Data Analysis	€ 10

10 rows |< < 1-3 of 3 > >|

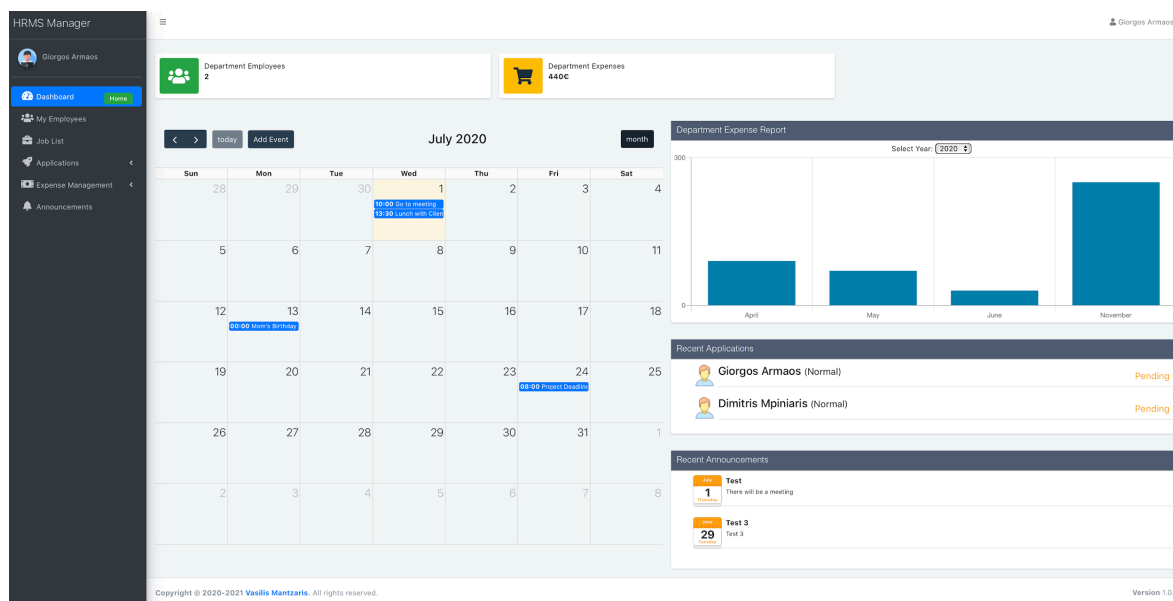
Εικόνα 32 - Expense List

Ανακοινώσεις Τμημάτων

Στην σελίδα αυτή, ο διαχειριστής και ο διευθυντής μπορούν να προσθέσουν ανακοινώσεις στα τμήματα. Ο διαχειριστής έχει την επιλογή να δημοσιεύσει

ανακοινώσεις σε όλα τα τμήματα της εταιρείας. Επίσης ο διαχειριστής μπορεί να δει τις ανακοινώσεις όλων των τμημάτων, ενώ ο διευθυντής, μόνο του τμήματος του.

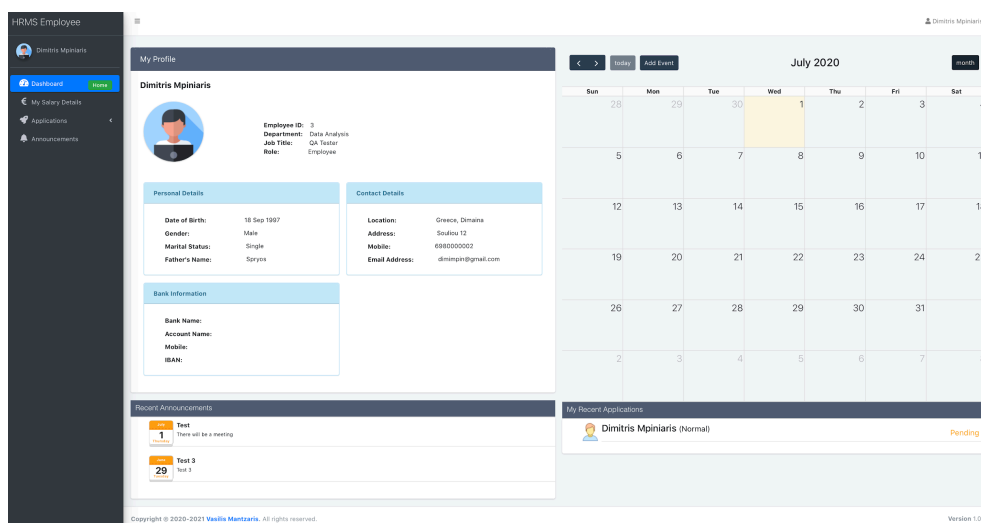
5.3 Διευθυντής (Manager)



Εικόνα 33 - Dashboard Manager

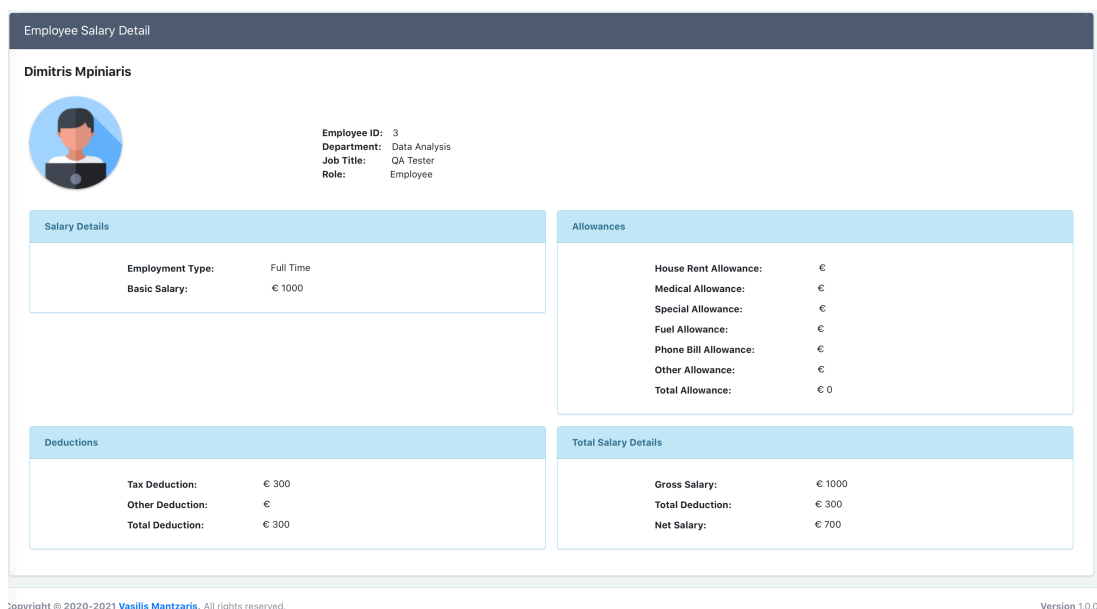
Σε αντίθεση με τον διαχειριστή η οθόνη του Διευθυντή δεν περιέχει πληροφορία ως προς το σύνολο των πληρωμών του συστήματος. Εκτός από το σύστημα πληρωμών, ο διευθυντής μπορεί να συλλέξει πληροφορίες για το τμήμα του όπως, το σύστημα εξόδων, τις ανακοινώσεις, τους υπαλλήλους του τμήματος, τις θέσεις εργασίας τους και τις αιτήσεις τους. Επίσης βλέπει τις πρόσφατες ανακοινώσεις του τμήματος

5.4 Υπάλληλος (Employee)



Εικόνα 34 - Dashboard Employee

Στην οθόνη του απλού χρήστη φαίνεται η καρτέλα του για τον έλεγχο των πληροφοριών του. Επίσης έχει δικαίωμα στην υποβολή προσωπικών γεγονότων αλλά και στην προβολή των πρόσφατων αιτήσεων του. Έπειτα μπορεί να προβάλλει την καρτέλα των οικονομικών του στοιχείων, τις ανακοινώσεις του τμήματος αλλά και να καταθέσει αίτηση.



Εικόνα 35 - Employee Salary Card

ΚΕΦ.6: ΕΠΙΛΟΓΟΣ

Η πτυχιακή μου εργασία αναλύει, σχεδιάζει και υλοποιεί ένα ηλεκτρονικό σύστημα διαχείρισης ανθρώπινου δυναμικού. Η υλοποίηση έχει γίνει με την χρήση της client-server αρχιτεκτονικής. Για την υλοποίηση της χρησιμοποιήθηκε κατά κόρον η γλώσσα Javascript τόσο στον server, με την χρήση Node.js, Express.js, αλλά και στον client, με την χρήση React.js. Επίσης χρησιμοποιήθηκαν μια πληθώρα βιβλιοθηκών για την υλοποίηση των λειτουργιών του συστήματος.

Κατα την διάρκεια της πτυχιακής μου εργασίας ανέπτυξα δεξιότητες ως προς την σχεδίαση, ανάλυση και υλοποίηση σύνθετων συστημάτων. Επίσης βελτιώθηκα στην διαχείριση χρόνου και κατανομής του φόρτου εργασίας. Γενικά, η ανάπτυξη του συγκεκριμένου συστήματος ήταν μια αναμφισβήτητη ευκαιρία βελτίωσης για μένα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Kanthawongs, P. (2004). *Does HRIS matter for HRM today?* <https://www.sdbor.edu/cffile/Agendas/HRFIS/HOW%20AN%20HRIS%20CAN%20IMPACT%20HR.do>
- [2] Hollinger, K., "How to Select an HRIS and Save Your Saturdays" HRTX Forum Library - HR Information Systems Management <https://cdn.ymaws.com/alachicago.org/resource/resmgr/newsletter/novdec2012.pdf>
- [3] Broderick, R. and Boudreau, J.W. (1992). "Human Resource Information Technology and the competitive edge", Academy of Management Executive https://www.researchgate.net/publication/37149629_Human_Resource_Management_Information_Technology_and_the_Competitive_Edge
- [4] Kovach, K. & Cathcart, C. (1999). "Human Resource Information Systems (HRIS): Providing Business with Rapid Data Access, Information Exchange and Strategic Advantage", *Public Personnel Management*, Vol.28, No.2
- [5] Walker, A.J. (2001), How The Web And Other Key Trends Are Changing Human Resources, In Walker, A. (Ed.), *Web-Based Human Resources*
- [6] Tannenbaum, S.I. (1990). Human Resource Information System Information: User Group Implications" *Journal of Systems Management*, Vol. 41 No.1 Pp.27 <https://search.proquest.com/openview/65a6d83e1080fdc2f1ca947be8ead040/1?pq-origsite=gscholar&cbl=40682>
- [7] Kettle, P. and Reilly, P. (2003) eHR: An Introduction. IES Report 398, Brighton, UK: Institute of Employment Studies <https://www.employment-studies.co.uk/system/files/resources/files/398.pdf>
- [8] Panayotopoulou, L., Vakola, M. and Galanaki E., (2007), "E-HR adoption and the role of HRM: evidence from Greece", *Personnel Review*, Vol. 36 No. 2, pp. 277-294. https://www.academia.edu/7206134/E-HR_adoption_and_the_role_of_HRM_evidence_from_Greece
- [9] Schnitt, D.L. (1993). "Reengineering the organization using information technology", *Journal of Systems Management*, January, pp.14-20, 41-42.
- [10] Ngai, E., & Wat, F. (2006). "Human resource information systems: A review and empirical analysis", *Personnel Review*, Vol.35 No.3, pp. 306
- [11] Booch G., Rumbaugh J. and Jacobson I., *The Unified Modeling Language User Guide*, Addison Wesley, Boston, MA, 1999

- [12] Tilkov S, Vinoski S. (2010). "Node.js: Using JavaScript to Build High-Performance Network Programs". IEEE Internet Computing. vol. 14. Issue. 6. <https://ieeexplore.ieee.org/document/5617064>
- [13] Laurent Orsini (2013). "What You Need To Know About Node.js". <https://readwrite.com/2013/11/07/what-you-need-to-know-about-nodejs/>
- [14] P. Teixeira (2012-10-01). "Professional Node.js: Building JavaScript Based Scalable Software", John Wiley & Sons, <https://books.google.gr/books?id=ZH6bpbcrIvYC>
- [15] Patrick Schaumont (2013). A Practical Introduction to Hardware/Software Codesign. Springer. https://www.academia.edu/32647425/A_Practical_Introduction_to_Hardware_Software_Codesign
- [16] George Ornbo (2012). Sams Teach Yourself Node.js in 24 Hours. Sams Publishing. ISBN 9780672335952. <https://ptgmedia.pearsoncmg.com/images/9780672335952/samplepages/0672335956.pdf>
- [17] Node.js <https://nodejs.org/en/about/>
- [18] React.js <https://reactjs.org>
- [19] Express.js <https://expressjs.com>
- [20] Sequelize <https://sequelize.org>
- [21] Bootstrap <https://getbootstrap.com>
- [22] Material Table <https://material-ui.com/components/tables/>
- [23] Jwt <https://jwt.io/introduction/>