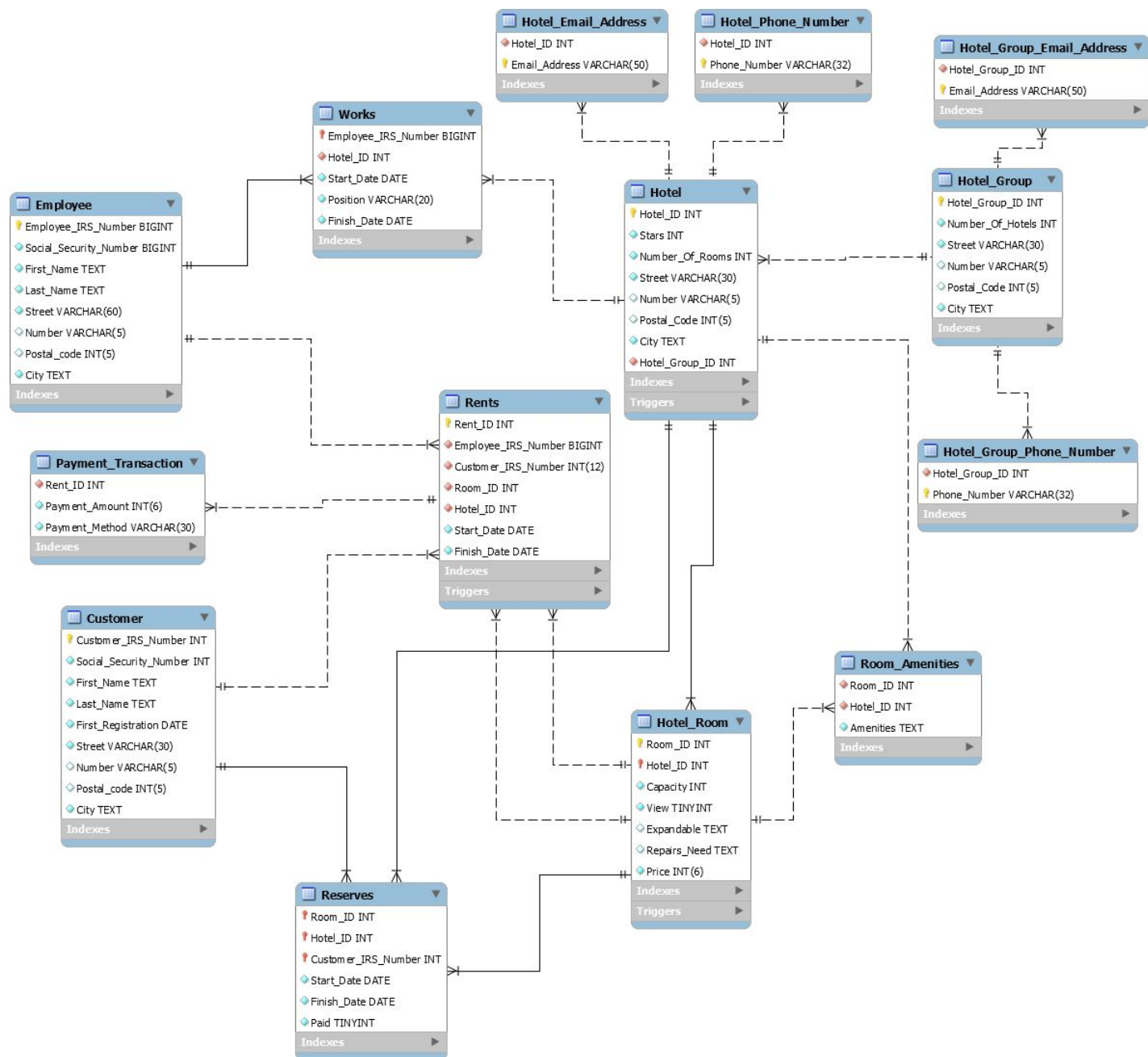


**ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**  
**ΘΕΜΑ**  
**2017-2018 Εαρινό Εξάμηνο**

Γεωργίου Δημήτριος	el15106	6ο Εξάμηνο
Χαρδούβελης Γεώργιος-Ορέστης	el15100	6ο Εξάμηνο
Ελευθερίου Σοφία	el14795	8ο Εξάμηνο



Στην παρούσα εργασία καλούμαστε να αναπτύξουμε ένα “e-Hotels” site, το οποίο να έχει μία βάση από 5 αλυσίδες ξενοδοχείων. Σε αυτό, ο πελάτης μπορεί να δει όλες τις πληροφορίες που χρειάζεται για τα δωμάτια των επιμέρους ξενοδοχείων και να κάνει εύκολα και αποδοτικά κρατήσεις σε πραγματικό χρόνο.

**α.** Με βάση το E-R διάγραμμα που δόθηκε, σχεδιάστηκε το παρακάτω σχεσιακό διάγραμμα.





Ακόμη, εδώ αναλυτικότερα φαίνονται όλα τα tables με κλειδιά τους και τα γνωρίσματα του καθενός, όπως φαίνεται στον rhrmyadmin.



## Customer

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Customer_IRS_Number 	int(10)
<input type="checkbox"/>	2	Social_Security_Number 	int(10)
<input type="checkbox"/>	3	First_Name	text
<input type="checkbox"/>	4	Last_Name	text
<input type="checkbox"/>	5	First_Registration	date
<input type="checkbox"/>	6	Street	text
<input type="checkbox"/>	7	Number	varchar(5)
<input type="checkbox"/>	8	Postal_Code	int(5)
<input type="checkbox"/>	9	City	text



## Employee

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Employee_IRS_Number 	bigint(20)
<input type="checkbox"/>	2	Social_Security_Number 	bigint(20)
<input type="checkbox"/>	3	First_Name	text
<input type="checkbox"/>	4	Last_Name	text
<input type="checkbox"/>	5	Street	text
<input type="checkbox"/>	6	Number	varchar(5)
<input type="checkbox"/>	7	Postal_code	int(5)
<input type="checkbox"/>	8	City	text


## Hotel

	#	Όνομα	Τύπος	
<input type="checkbox"/>	1	Hotel_ID 	int(10)	
<input type="checkbox"/>	2	Stars	int(4)	
<input type="checkbox"/>	3	Number_Of_Rooms	int(10)	
<input type="checkbox"/>	4	Street	text	
<input type="checkbox"/>	5	Number	varchar(5)	
<input type="checkbox"/>	6	Postal_Code	int(5)	
<input type="checkbox"/>	7	City	text	
<input type="checkbox"/>	8	Hotel_Group_ID 	int(10)	



## Hotel\_Email\_Address

	#	Όνομα	Τύπος	Σ
<input type="checkbox"/>	1	Hotel_ID 	int(10)	
<input type="checkbox"/>	2	Email_Address 	varchar(50)	u



## Hotel\_Group

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Hotel_Group_ID 	int(10)
<input type="checkbox"/>	2	Number_Of_Hotels	int(10)
<input type="checkbox"/>	3	Street	text
<input type="checkbox"/>	4	Number	varchar(5)
<input type="checkbox"/>	5	Postal_Code	int(5)
<input type="checkbox"/>	6	City	text

### Hotel\_Group\_Email\_Address

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Hotel_Group_ID 	int(10)
<input type="checkbox"/>	2	Email_Address 	varchar(50)




### Hotel\_Group\_Phone\_Number

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Hotel_Group_ID 	int(10)
<input type="checkbox"/>	2	Phone_Number 	varchar(32)


### Hotel\_Phone\_Number

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Hotel_ID 	int(10)
<input type="checkbox"/>	2	Phone_Number 	varchar(32)



## Hotel\_Room

	#	Όνομα	Τύπος	Σ
<input type="checkbox"/>	1	Room_ID 	int(10)	
<input type="checkbox"/>	2	Hotel_ID  	int(10)	
<input type="checkbox"/>	3	Capacity	int(1)	
<input type="checkbox"/>	4	View	tinyint(1)	
<input type="checkbox"/>	5	Expandable	text	u
<input type="checkbox"/>	6	Repairs_Need	text	u
<input type="checkbox"/>	7	Price	int(4)	

## Payment\_Transaction

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Rent_ID 	int(10)
<input type="checkbox"/>	2	Payment_Amount	int(6)
<input type="checkbox"/>	3	Payment_Method	varchar(30)

## Rents

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Rent_ID 	int(10)
<input type="checkbox"/>	2	Employee_IRS_Number 	bigint(20)
<input type="checkbox"/>	3	Customer_IRS_Number 	int(12)
<input type="checkbox"/>	4	Room_ID 	int(10)
<input type="checkbox"/>	5	Hotel_ID 	int(10)
<input type="checkbox"/>	6	Start_Date	date
<input type="checkbox"/>	7	Finish_Date	date

## Reserves

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Room_ID 	int(10)
<input type="checkbox"/>	2	Hotel_ID 	int(10)
<input type="checkbox"/>	3	Customer_IRS_Number 	int(10)
<input type="checkbox"/>	4	Start_Date	date
<input type="checkbox"/>	5	Finish_Date	date
<input type="checkbox"/>	6	Paid	tinyint(1)



### Rooms\_By\_Capacity

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	<b>Room_ID</b>	int(10)
<input type="checkbox"/>	2	<b>Capacity</b>	int(1)
<input type="checkbox"/>	3	<b>Hotel_ID</b>	int(10)

### Rooms\_In\_Cities

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	<b>Room_ID</b>	int(10)
<input type="checkbox"/>	2	<b>City</b>	text
<input type="checkbox"/>	3	<b>Hotel_ID</b>	int(10)

### Room\_Amenities

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	<b>Room_ID</b> 	int(10)
<input type="checkbox"/>	2	<b>Hotel_ID</b> 	int(10)
<input type="checkbox"/>	3	<b>Amenities</b>	text



## Works

	#	Όνομα	Τύπος
<input type="checkbox"/>	1	Employee_IRS_Number 	bigint(20)
<input type="checkbox"/>	2	Hotel_ID 	int(10)
<input type="checkbox"/>	3	Start_Date	date
<input type="checkbox"/>	4	Position	varchar(20)
<input type="checkbox"/>	5	Finish_Date	date

**b.** Για κάθε πίνακα έχουν οριστεί συγκεκριμένοι περιορισμοί με στόχο την διασφάλιση της ακεραιότητας της βάσης δεδομένων.

### Primary Keys

Σε κάθε table (εκτός από αυτά που ορίζουν σχέσεις μεταξύ οντοτήτων) έχουν οριστεί ένα (ή παραπάνω σε περίπτωση weak entities) primary entities, τα οποία είναι μοναδικά για κάθε στοιχείο του εκάστοτε πίνακα (πχ IRS Number στους Employees & Customers, Hotel\_ID & Room\_ID σε συνδυασμό στα Hotel Rooms, κοκ), ενώ οφείλει να μην είναι κενό.

Σημαντικό είναι να αναφέρουμε πως στον πίνακα Rents προσθέσαμε ένα επιπλέον πεδίο που το ορίσαμε σαν Primary Key, για ευκολία και για αποτελεσματική σύνδεση του με τον πίνακα Payment Transaction.

### Unique Keys

Πέρα από τα Primary Keys, υπάρχουν και ορισμένες ακόμη στήλες τα οποία είναι μοναδικά στον κάθε πίνακα. Αυτά είναι το Social Security Number στους πίνακες Employees & Customers.

Στα tables που χαρακτηρίζουν άτομα (Employees, Customers), τα primary keys είναι τα IRS Numbers τους και άρα είναι μοναδικά, το οποίο είναι λογικό δεδομένου ότι κάθε άτομο έχει διαφορετικό. Με την ίδια λογική, στους ίδιους πίνακες έχουν οριστεί ως unique και τα Social Security Numbers.

Ακόμη, έχουμε τα hotel και hotel group ID's στους πίνακες που ορίζουν τα τηλέφωνα και τις ηλεκτρονικές διευθύνσεις των ξενοδοχείων και ξενοδοχειακών μονάδων αντίστοιχα.

Στο πεδίο με τις παροχές των δωματίων, unique τιμές αποτελούν τα Hotel\_ID και Room\_ID σε συνδυασμό.

Τέλος στο Payment Transaction, το Rent ID είναι unique πεδίο.

Στα τελευταίες περιπτώσεις, τα πεδία αυτά είναι Foreign Keys συνδεόμενα με τους αντίστοιχους πίνακες στους οποίους είναι Primary Keys. Έτσι είναι λογικό να επιθυμούμε να είναι μοναδικά.

### **Foreign Keys**

Σε ορισμένους πίνακες υπάρχουν εξωτερικά κλειδιά που προσδιορίζουν μοναδικές εγγραφές σε άλλους πίνακες. Έτσι, κάθε εξωτερικό κλειδί πρέπει να έχει έγκυρη τιμή δηλαδή να προσδιορίζει μοναδικά μία υπάρχουσα εγγραφή σε κάποιο άλλο πίνακα, το οποίο επιτυγχάνεται με το keyword FOREIGN KEY.

Ακόμη, αφού οριστούν, μπορούμε να προσθέσουμε και το keyword ON DELETE CASCADE ώστε άμα διαγραφεί κάτι από το parent table να διαγραφεί και από αυτόν τον πίνακα. Όπου χρειάζεται, προσθέτουμε και το ON UPDATE CASCADE ώστε να γίνεται το ίδιο και στην ανανέωση στοιχείων. Δεν προστέθηκε σε στοιχεία τα οποία δεν είναι λογικό να αλλάξουν (βλ. IRS Numbers) αλλά και σε περιπτώσεις που δεν ήταν επιθυμητό. Συγκεκριμένα, στους πίνακες rents και reserves, δεν έχει προστεθεί on cascade delete στα foreign keys αφού σε περίπτωση που διαγραφεί ένα από τα στοιχεία από άλλον πίνακα που αφορά αυτή την κράτηση / ενοικίαση, θέλουμε να παραμείνει στον πίνακα σαν ιστορικό. Επίσης υπάρχουν περιπτώσεις που δεν ενδείκνυνται για αλλαγή κλειδιού (πχ Hotel\_Group\_ID) και δεν δίνεται η δυνατότητα αλλαγής του από τον χρήστη, οπότε δεν είναι απαραίτητο να υπάρχουν στα αντίστοιχα foreign keys.

### **NOT NULL**

Με αυτή την εντολή ορίζουμε τα πεδία τα οποία δεν επιτρέπεται να είναι κενά. Αυτά είναι τα περισσότερα πεδία καθώς ως επί το πλείστον έχουμε τιμές που είναι απαραίτητο να ορίζονται στην βάση μας.

Μια τιμή που δεν είναι ορισμένες ως NOT NULL είναι ο αριθμός του δρόμου στη διεύθυνση. Αυτό γιατί υπάρχει περίπτωση ο πελάτης ή υπάλληλος να μένει σε κάποια απομακρυσμένη περιοχή και να μην ορίζεται κάποιος αριθμός.

Επιπλέον, δεν είναι ορισμένες ως NOT NULL και τα πεδία Expandable -αν και πως μπορεί το δωμάτιο να επεκταθεί για να φιλοξενήσει μεγαλύτερο αριθμό δωματίων- και Repairs -αν και τι προβλήματα που χρειάζονται επισκευή θέλει το δωμάτιο -. Αυτό καθώς, αρχικά, μπορεί να μην γίνεται επέκταση ή να μην χρειάζονται επισκευές οπότε υπάρχει η δυνατότητα αντί να γραφτεί το παραπάνω στο αντίστοιχο πεδίο να έχει μείνει κενό στην βάση. Ακόμη, πρόκειται για παραμέτρους που δεν είναι απαραίτητες καθώς δεν δίνεται στον χρήστη δυνατότητα φιλτραρίσματος με βάση αυτές.

Οι παραπάνω λοιπόν τιμές ορίζονται ως DEFAULT NULL, δηλαδή αν δεν προστεθεί τιμή είναι εξ'ορισμού NULL.

## **Data Types και Φυσικοί Περιορισμοί**

Καταρχάς, για κάθε πεδίο έχει οριστεί το αντίστοιχο Data Type για κάθε πεδίο, είτε είναι αριθμός είτε κείμενο. Ιδιαίτερη προσοχή αξίζει να δοθεί στο street number που δεν έχει οριστεί αριθμός καθώς μπορεί να υπάρχουν και χαρακτήρες σε αυτό (πχ 73B).

Επίσης σε όσα πεδία είναι εφικτό έχει μπει και περιορισμός χαρακτήρων, όπως στα IRS Numbers, Social Security Numbers, ο ταχυδρομικός κώδικας που έχουν συγκεκριμένο πλήθος νούμερων. Επιπλέον, σε άλλα πεδία έχει μπει επίσης περιορισμός (πχ το street number αποκλείεται να περνάει τα 5 ψηφία κοκ) για να εξοικονομηθεί χώρος.

Οι παραπάνω περιορισμοί έχουν τεθεί σε επίπεδο MySQL. Υπάρχουν και άλλοι περιορισμοί σε επίπεδο php ή html, όπως τα αστέρια, η χωρητικότητα, τα hotel rooms είναι θετικοί ακέραιοι αριθμοί, ορισμένα ακόμη datatypes) κοκ.

Στην αμέσως επόμενη παράγραφο θα φανούν μερικοί ακόμη τέτοιοι περιορισμοί.

## **ΣΥΣΧΕΤΙΣΕΙΣ ΜΕΤΑΞΥ ΠΙΝΑΚΩΝ**

Σε πίνακες με κλειδιά που αλληλοεξαρτώνται, όταν γίνεται insert, το αντίστοιχο πεδίο επιλέγεται από συγκεκριμένη λίστα που εμφανίζεται. Για παράδειγμα, αν γίνει insert στον πίνακα works, το στοιχείο Employee IRS Number επιλέγεται από λίστα, δεν εισάγεται όπως τα υπόλοιπα πριν γίνει υποβολή.

Ομοίως, όταν γίνεται εισαγωγή σε ξενοδοχείο (ή δωμάτιο) επιλέγεται από λίστα το η αντίστοιχη αλυσίδα (και το ξενοδοχείο αντίστοιχα).

Το ίδιο ισχύει και για όλα τα πεδία με unique foreign keys (Phone Number και Email Address για ξενοδοχεία και αλυσίδες, ενοικιάσεις, κρατήσεις, παροχές κοκ.

Παρόμοιες ενέργειες γίνονται και για τα updates. Τα στοιχεία κάθε οντότητας που δεν μπορεί (ή δεν θα έπρεπε να αλλάξει), δεν εισάγεται εκ νέου από τον χρήστη αλλά επιλέγεται από λίστα και του δίνεται η δυνατότητα να επεξεργαστεί τα υπόλοιπα στοιχεία.

Τέλος και στα deletes προφανώς γίνεται επιλογή από λίστα το στοιχείου που θα διαγραφεί.

## TRIGGERS

Για να εξασφαλισθεί η ακεραιότητα της βάσης, κάποιες ενέργειες οφείλουν να εκτελούνται αυτόματα. Αυτό γίνεται αρχικά με τα keywords CASCADE που είδαμε στα Foreign Keys, όπου αν διαγραφεί / ενημερωθεί η αντίστοιχη στήλη από τον πίνακα-πατέρα διαγράφεται και από τον πίνακα-παιδί.

Παράλληλα, δημιουργήσαμε και τα Triggers: delete\_Hotel, delete\_Hotel\_Room, update\_Hotel, update\_Hotel\_Room, Check\_In. Τα triggers έχουν ακριβώς την προαναφερόμενη χρησιμότητα.

Χάρη στα πρώτα 4, όταν διαγράφουμε ή προσθέτουμε ένα δωμάτιο σε ένα ξενοδοχείο ή ένα ξενοδοχείο σε μια αλυσίδα ξενοδοχείων, αλλάζει (μεγαλώνει ή μικραίνει κατά ένα) ο αριθμός των δωματίων/ξενοδοχείων αντίστοιχα.

Το τελευταίο Trigger εξυπηρετεί ώστε, όταν γίνεται το Check In, δηλαδή μια κράτηση (αν είχε γίνει και κράτηση πριν) γίνεται ενοικίαση, τότε το πεδίο paid στην αντίστοιχη γραμμή στον πίνακα Reserves γίνεται True. Αυτό ορίζεται έτσι δεδομένου ότι η πληρωμή θα γίνεται πάντα στο κάθε ξενοδοχείο όταν πραγματοποιείται η ενοικίαση.

**c.** Με βάση την λίστα με τα DDL's (τα tables με κλειδιά τους και τα γνωρίσματα του καθενός) τα κίτρινα κλειδιά δηλώνουν τα Primary Keys ενώ τα γκρι τα Ευρετήρια (Indexes). Τα ευρετήρια είναι βοηθητικές δομές αρχείου και κάνουν την αναζήτηση μιας εγγραφής σε ένα αρχείο πιο αποδοτική. Η δομή ενός ευρετηρίου καταλαμβάνει σημαντικά λιγότερο χρόνο από το αρχείο.

Τα primary keys, δηλαδή τα κλειδιά εκείνα που είναι μοναδικά για κάθε στοιχείο του αντίστοιχου πεδίου και πρακτικά προσιορίζουν τα διαφορετικά στοιχεία, είναι Indexes, καθώς και τα Unique Keys. Επιπλέον, δεδομένου πως χρησιμοποιούμε InnoDB, και τα foreign keys είναι εξ'ορισμού indexes.

Έτσι παραπάνω βλέπουμε όλα τα indexes που έχουν δημιουργηθεί. Δεν δημιουργήθηκαν άλλα καθώς δεν κρίθηκε απαραίτητο, εφόσον τα ήδη υπάρχοντα είναι ικανά και αναγκαία για οποιαδήποτε αναζήτηση χρειάζεται να κάνουμε.

**d.** Για την ανάπτυξη της εφαρμογής χρησιμοποιήσαμε: την τελευταία έκδοση του MySQL Server για την υλοποίηση της Βάσης Δεδομένων, Apache (2.4 έκδοση) για την επικοινωνία client-server, και PHP 7 για το server side της εφαρμογής και HTML & CSS για το client side της εφαρμογής.

Το περιβάλλον στο οποίο αναπτύχθηκε η εφαρμογή είναι macOS.

e. Για να εγκαταστήσει κάποιος την εφαρμογή μας από την αρχή, πρώτο βήμα είναι η εγκατάσταση και το configuration των προγραμμάτων που αναφέρθηκαν στο παραπάνω ερώτημα.

Έτσι πρέπει να γίνει εγκατάσταση την τελευταίας έκδοσης της MySQL, ακολουθώντας τα απλά βήματα του [MySQL installer](#).

Για να εγκατασταθεί σωστά ο Apache πρέπει, να κατεβεί το λογισμικό από το site και μετά ο χρήστης να τρέξει το αρχείο .msi και να ακολουθησει τις οδηγίες. Στο τέλος κάνει επανεκκίνηση και ο server πρέπει να τρέχει. Αυτό μπορεί να το ελέγξει εφόσον μπαίνοντας στο URL: <http://localhost/>, πρέπει να εμφανίζει μήνυμα "It works!". Γενικά εδώ μπορεί να προκύψουν προβλήματα με το configuration, κυρίως με το port με το οποίο επικοινωνεί, εφόσον εξ'ορισμού είναι το 80, το οποίο συχνά χρησιμοποιείται και από άλλες εφαρμογές (πχ Skype). Έτσι μπορεί να κάνει αλλαγές στο configuration file (httpd.conf). Στη συγκεκριμένη περίπτωση, μπορεί στα σημεία Listen 80 να μεταβάλλει το port σε 8080 και να δουλέψει, τσεκάροντας πλέον το URL <http://localhost:8080>.

Τέλος, πρέπει να γίνει εγκατάσταση και το configuration της PHP Engine. Αρχικά πρέπει ο χρήστης να κατεβάσει το zip file από την αντίστοιχη [ιστοσελίδα](#) και να τρέξει το .msi αρχείο. Στη συνέχεια, πρέπει στο Configuration Directory Panel του Apache να προσδιοριστεί που βρίσκεται το http.conf αρχείο (by default είναι στον φάκελο C:\Program Files\Apache Software Foundation\Apache<version>\conf\).

Προσοχή να επιλεγθούν τα στοιχεία MySQL / MySQLi στην εγκατάσταση.

Τέλος πρέπει να γίνει ξανά η εγκατάσταση του Apache. Για να ελεγχθεί, μπορεί να γραφτεί σε php γλώσσα να εκτυπώνεται ένα μήνυμα (πχ <?php echo "PHP has been installed successfully!"; ?>) και να αποθηκευτεί στον φάκελο htdocs σαν test.php.

Τώρα, στο URL <http://localhost:<port>/test.php> βάζοντας στο port αυτό που έχει προσδιοριστεί στον Apache, θα πρέπει στην σελίδα να φαίνεται το μήνυμα που γράφτηκε.

Είναι σημαντικό να γίνει configure το document root, προσδιορίζοντας localhost, που είναι μια διεύθυνση IT τοπικά ορισμένη στον υπολογιστή και στο οποίο πρέπει να βρίσκονται τα αρχεία php για να τρέξουν.

Βλέποντας λίγο πιο συγκεκριμένα τα βήματα, δεδομένου πως η εργασία αυτή είναι μία web browser εφαρμογή, πρέπει να γίνεται σωστή επικοινωνία με τον server. Τη βάση μας την διαχειρίζεται άμεσα ο MySQL server. Τελικά βέβαια, η εφαρμογή τρέχει μέσω του phpmyadmin, από όπου μπορούμε και να επεξεργαστούμε άμεσα την βάση. Το

Apache είναι ο διαμεσολαβητής μας, που στέλνει τα requests από την PHP στην βάση δεδομένων και είναι μείζονος σημασίας για να τρέξει η εφαρμογή.

Τέλος, για να ανοίξει ο χρήστης την εφαρμογή, εφόσον καλύπτονται οι παραπάνω προϋποθέσεις, είναι κατεβάσει το zip αρχείο που παραδόθηκε και να κάνει extract το περιεχόμενο του στο document root. Ύστερα πρέπει να ανοίξει το αρχείο welcome.php στο οποίο βρίσκεται η αρχική μας σελίδα. Από εκεί η ίδια η εφαρμογή σε καθοδηγεί, αλλά περισσότερες πληροφορίες βρίσκονται και παρακάτω.

## Δομή Εφαρμογής - User Interface

Η εφαρμογή μας (HO-HOteleies services) είναι έτοιμη για χρήση όταν ανοίξει ο χρήστης το [welcome page](#).

Εκεί του δίνεται η δυνατότητα να επιλέξει την ταυτότητα του, δηλαδή αν είναι [admin](#) της εφαρμογής, [υπάλληλος](#) σε κάποιο ξενοδοχείο ή [πελάτης](#).

Σε κάθε περίπτωση γίνεται ανακατεύθυνση σε μια νέα σελίδα.

Για τον admin ή τους υπαλλήλους ζητείται και username και αντίστοιχο password (admin, 12345678 για τον admin, 1234 για τον employee).

- Admin

Ο admin μπορεί να εισάγει νέο στοιχείο σε κάθε πίνακα που έχει υλοποιηθεί πέρα από τον Payment\_Transaction. Γενικά στον admin δίνεται μεγάλη ελευθερία όσον αφορά στην εισαγωγή στοιχείων. Στον Payment Transaction, που σαν στοιχεία έχει μόνο Rent\_ID, Payment\_ID και Payment\_Method δεν μπορεί να εισάγει νέα στοιχεία εφόσον ενημερώνεται αυτόματα με την κράτηση.

Σχετικά με την ενημέρωση στοιχείων, ο admin μπορεί να κάνει update σε όλα τα παραπάνω, πέρα από το Rents. Αυτό είναι λογικό δεδομένου ότι αφού γίνει η ενοικίαση μετά το check in, δεν θα έπρεπε να επιτρέπεται νέα αλλαγή στην βάση.

Το ίδιο ισχύει και για την διαγραφή στοιχείων, δηλαδή γίνεται σε όλους πέρα του Payment Transaction και του Rents. Σχετικά με την τελευταία περίπτωση, δεν θέλουμε να μπορεί να γίνει διαγραφή αφού επιθυμούμε να κρατιέται το ιστορικό.

- Employee

Ο Employee, εφόσον βάλει το αντίστοιχο username και password, έχει την δυνατότητα να κάνει εισαγωγή στοιχείων στους πίνακες Customers, Reserves και

Rents. Γενικά, ο employee πρέπει να μπορεί να κάνει αρκετές αλλαγές ώστε να μπορεί να επιτελέσει την δουλειά του αλλά όχι παραπάνω ελευθερίες. Έτσι, μπορεί να προσθέσει πελάτες, κρατήσεις (σε περίπτωση που κάποιος πελάτης θέλει να κάνει τηλεφωνικά μέσω του υπαλλήλου) και προπάντων να προσθέσει ενοικιάσεις όταν γίνεται το check in.

Σχετικά με την ενημέρωση και διαγραφή πινάκων, ο υπάλληλος μπορεί να κάνει στους customers, Reserves και όχι στον Rents. Στις πρώτες περιπτώσεις είναι εύλογο εφόσον μπορεί να υπάρξει αλλαγή στην κράτηση, να ακυρωθεί ή ο πελάτης να άλλαξε ορισμένα από τα στοιχεία του ή να διώχθηκε από την εταιρεία.

Στον Rents δεν μπορεί να πραγματοποιήσει τις παραπάνω ενέργειες δεδομένου ότι αφού γίνει η ενοικίαση μετά το check in, δεν θα έπρεπε να επιτρέπεται νέα αλλαγή στην βάση.

- Customer

Ο πελάτης ανακατευθύνεται σε νέα σελίδα που μπορεί να κάνει αναζήτηση δωματίων και κρατήσεις. Στην αρχική σελίδα μπορεί να δει τα διαθέσιμα δωμάτια με ή χωρίς φίλτρα. Συγκεκριμένα, μπορεί να αναζητήσει δωμάτια με βάση την ημερομηνία που τον ενδιαφέρει, την χωρητικότητα του δωματίου, την περιοχή, την αλυσίδα ξενοδοχείων που ανήκει το ξενοδοχείο, την κατηγορία του ξενοδοχείου, τον συνολικό αριθμός των δωματίων που έχει το ξενοδοχείο, το εύρος τιμής του κάθε δωματίου, καθώς και οι παροχές του δωματίου. Τα κριτήρια αυτά είναι ανεξάρτητα και δεν απαιτούνται όλα.

Σε κάθε επιλογή που εμφανίζεται, ο πελάτης μπορεί να επιλέξει να δημιουργήσει κράτηση. Έτσι εμφανίζεται pop-up παράθυρο στο οποίο ζητούνται τα απαραίτητα στοιχεία για να πραγματοποιηθεί η κράτηση. Αν δεν είναι εγγεγραμμένος στην λίστα των πελατών, καλείται να κάνει εγγραφή πρώτα.

Παράλληλα, έχουν δημιουργηθεί και [2 views](#) στην βάση δεδομένων μας, τα [Rooms\\_In\\_Cities](#) και [Rooms\\_By\\_Capacity](#), τα οποία όλα τα δωμάτια στην βάση μας ανά περιοχή και ανά χωρητικότητα αντίστοιχα. Πατώντας τα αντίστοιχα κουμπιά στην σελίδα, ο χρήστης μπορεί να δει τις λίστες αυτές.