



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Εφαρμογή Κινητής Συσκευής και Πληροφοριακό
Σύστημα για το Έξυπνο Ξενοδοχείο**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

των

ΣΜΥΡΛΗ ΓΕΩΡΓΙΟΥ (ΑΕΜ: 2464)

ΚΟΣΚΙΝΑ ΧΑΡΑΛΑΜΠΟΥ (ΑΕΜ: 2550)

Επιβλέπων: Δημόκας Νικόλαος
Επίκουρος Καθηγητής

Καστοριά Δεκέμβριος - 2023



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Εφαρμογή Κινητής Συσκευής και Πληροφοριακό Σύστημα για το Έξυπνο Ξενοδοχείο

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

των

ΣΜΥΡΛΗ ΓΕΩΡΓΙΟΥ (ΑΕΜ: 2464)

ΚΟΣΚΙΝΑ ΧΑΡΑΛΑΜΠΟΥ (ΑΕΜ: 2550)

Επιβλέπων: Δημόκας Νικόλαος
Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την Πέμπτη 21 Δεκεμβρίου 2023

Νικόλαος Δημόκας
Επίκουρος Καθηγητής

Ιωάννης Βαρδάκας
Αναπληρωτής Καθηγητής

Σπύρος Νικολάου
Λέκτορας

Καστοριά, Δεκέμβριος - 2023

Copyright © 2023 – ΣΜΥΡΛΗΣ ΓΕΩΡΓΙΟΣ, ΚΟΣΚΙΝΑΣ ΧΑΡΑΛΑΜΠΟΣ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τους συγγραφείς και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Δυτικής Μακεδονίας.

Ως συγγραφείς της παρούσας εργασίας δηλώνουμε πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Ευχαριστίες

Ευχαριστίες οφείλουμε καταρχάς στον επιβλέποντα καθηγητή της σχολής μας, κ. Δημόκα Νικόλαο, που με τη βοήθεια και την καθοδήγησή του έφτασε στο πέρας μία δημιουργική διαδικασία που συμβάλλει στην εξέλιξη μας. Επίσης, ευχαριστούμε τις οικογένειές μας, που μας στήριξαν και μας στηρίζουν σε κάθε μας προσπάθεια.

Περίληψη

Στο πλαίσιο αυτής της πτυχιακής εργασίας δημιουργήθηκε ένα περιβάλλον λογισμικού, το οποίο διευκολύνει και κάνει πιο αποδοτική τη λειτουργία ενός ξενοδοχείου. Το συγκεκριμένο περιβάλλον αποτελείται από μία εφαρμογή κινητής συσκευής για τους πελάτες του ξενοδοχείου, μία web εφαρμογή για τη διαχείριση του ξενοδοχείου και μία υπηρεσία ιστού (REST API) για τη διεπαφή με μία βάση δεδομένων.

Για την ανάπτυξη των εκάστοτε εφαρμογών χρησιμοποιήθηκαν σύγχρονες τεχνολογίες ανάπτυξης λογισμικού. Πιο συγκεκριμένα, η εφαρμογή κινητού αναπτύχθηκε μέσω του framework Xamarin, η βάση δεδομένων αναπτύχθηκε στην SQLite, ενώ για την δημιουργία του API χρησιμοποιήθηκε η τεχνολογία Flask. Τέλος, για την ανάπτυξη της web εφαρμογής, χρησιμοποιήθηκαν επίσης τα πρότυπα των γλωσσών HTML, CSS και Javascript.

Ειδικότερα, αναπτύξαμε μία εφαρμογή κινητού με το όνομα Smart Hotel για την διευκόλυνση των αναγκών του πελάτη στο ξενοδοχείο. Μέσω της εφαρμογής, ο πελάτης μπορεί να καλέσει την ρεσεψιόν. Επίσης μπορεί να στείλει αίτημα καθαριότητας του δωματίου του, όπως και αίτημα κράτησης σε κάποιο εστιατόριο του ξενοδοχείου. Παράλληλα, δημιουργήθηκε μια web εφαρμογή για την διαχείριση των αιτημάτων του ξενοδοχείου, που επιτρέπει στο προσωπικό του ξενοδοχείου να ενημερωθεί για τα αιτήματα των πελατών και στη συνέχεια να τα αποδεχτεί ή να τα απορρίψει, ενημερώνοντας ταυτόχρονα και τους πελάτες για την εξέλιξη των αιτημάτων τους. Επιπροσθέτως, στην ιστοσελίδα του ξενοδοχείου ενημερώνονται οι γενικές πληροφορίες του ξενοδοχείου, οι οποίες εμφανίζονται επίσης και στο κινητό του πελάτη.

Οι πληροφορίες που ενημερώνονται, μέσα από την web εφαρμογή και την εφαρμογή κινητού, είναι αποθηκευμένες σε μια βάση δεδομένων τύπου SQLite. Η βάση αυτή έχει τους αναγκαίους πίνακες (tables) ώστε να προσδιορίζεται ο τύπος χρήστη, οι κατάλληλες πληροφορίες του κάθε δωματίου, του κάθε αιτήματος των πελατών και των γενικών πληροφοριών του ξενοδοχείου. Μέσω του Flask API, οι εφαρμογές επικοινωνούν με την βάση, ανταλλάσσοντας πληροφορίες.

Το συγκεκριμένο περιβάλλον λογισμικού έχει ως στόχο να βελτιώσει την συνολική εμπειρία των πελατών ενός ξενοδοχείου, αυτοματοποιώντας την επικοινωνία μεταξύ πελατών – ξενοδοχείου.

Λέξεις Κλειδιά: *Xamarin, Flask, Έξυπνο Ξενοδοχείο, Smart Hotel, REST API, SQLite*

Abstract

In the context of this paper, a software environment related to the facilitation of the operation of a hotel was created. This environment consists of a mobile application for the customers of the hotel, a web application for the hotel's management and a REST API software with a Database.

Modern software development technologies were used for the development of each application. More specifically, the mobile application was developed through the Xamarin framework, the database is SQLite type, while Flask technology was used to create the API. For the development of the web application, the standard tools of the HTML, CSS and JavaScript languages were used.

More specifically, a mobile application called Smart Hotel has been developed to facilitate the needs of the customer at the hotel. Through the app, the customer can call the front desk, upload general hotel information, such as the hotel's address or website. He can also send a request for when his room will be cleaned, as well as for a table reservation at the hotel restaurant. At the same time, a web application was created for the hotel management, which allows the hotel staff to be informed about the requests of the customers and then to accept or reject them, at the same time informing the customers about the progress of their requests. Additionally, general information of the hotel is updated through the web application in order to be showed to the customer's mobile.

The data that has been updated through the web application and the mobile application, is stored in an SQLite database. This database has the necessary tables to determine the type of user, the appropriate information of each room, each customer request, and the general information of the hotel. Through the Flask API, applications communicate with the database, exchanging information.

The specific software environment aims to improve the overall experience of a hotel's customers by automating the communication between customers and the hotel. In essence, it saves time for both the staff and the customers, as it improves their communication in a smart and modern way.

Key Words: *Xamarin, Flask, Smart Hotel, REST API, SQLite*

Πίνακας Περιεχομένων

Εισαγωγή.....	1
1. Βιβλιογραφική Ανασκόπηση.....	2
1.1 Λογισμικό.....	2
1.1.1 Ιστορική Αναδρομή.....	2
1.2 Έξυπνο Ξενοδοχείο.....	3
1.3 Ανασκόπηση εφαρμογών Έξυπνων Ξενοδοχείων.....	9
1.3.1 Microsoft Smart Hotel 360.....	9
1.3.2 Hilton's Connected Room.....	10
1.3.3 Marriott's IoT Guestroom Lab.....	11
1.3.4 CitizenM's mobile app.....	12
2. Τεχνολογίες.....	14
2.1 Xamarin.....	14
2.2 Flask.....	16
2.3 SQLite.....	17
2.4 Βασικά Στοιχεία Ανάπτυξης Ιστοσελίδων.....	19
2.4.1 HTML.....	20
2.4.2 Javascript.....	20
2.4.3 CSS.....	21
3. Μοντέλο Δεδομένων.....	23
3.1 Ενδιάμεσο Λογισμικό.....	26
3.2 Διεπαφή Κινητής Συσκευής.....	28
3.3 Διεπαφή Ιστού.....	31
4. Εγχειρίδια Χρήστη.....	34
4.1 Εγχειρίδιο Κινητής Συσκευής.....	34
4.1.1 Login.....	35
4.1.2 Περιήγηση στην εφαρμογή.....	36
4.1.3 Η οθόνη Reception.....	37
4.1.4 Restaurant Reservation.....	38
4.1.5 Οθόνη Cleaning.....	44
4.1.6 Οθόνη About.....	49
4.1.7 Οθόνη Rating.....	50
4.2 Εγχειρίδιο Εφαρμογής Ιστού.....	51

4.2.1 Login	51
4.2.2 Reception	52
4.2.3 Restaurant και Cleaning	54
4.2.4 Rating, About και Logout	57
Συμπεράσματα.....	59
Βιβλιογραφία	61

Λίστα Εικόνων

Εικόνα 2.1 Xamarin Logo	14
Εικόνα 2.2 Flask Logo	16
Εικόνα 2.3 SQLite Logo.....	17
Εικόνα 2.4 HTML, JS, CSS	20
Εικόνα 3.1 Διάγραμμα Σχέσεων - Οντοτήτων.....	23
Εικόνα 4.1 Η εγκαταστημένη εφαρμογή SmartHotel σε Android	34
Εικόνα 4.1.1 Η οθόνη Login	35
Εικόνα 4.1.2.1 Η κεντρική οθόνη	36
Εικόνα 4.1.2.2 Η κυλιόμενη λίστα περιήγησης	37
Εικόνα 4.1.3 Η οθόνη Reception.....	38
Εικόνα 4.1.4.1 Η οθόνη Restaurant	39
Εικόνα 4.1.4.2 Κύλιση αιτήματος	40
Εικόνα 4.1.4.3 Η οθόνη δημιουργίας νέου αιτήματος Restaurant	41
Εικόνα 4.1.4.4 Επιλογή ημερομηνίας για κράτηση τραπεζιού.....	42
Εικόνα 4.1.4.5 Επιλογή ώρας για κλείσιμο τραπεζιού	43
Εικόνα 4.1.4.6 Καταχωρημένη οθόνη κράτησης τραπεζιού.....	44
Εικόνα 4.1.5.1 Η οθόνη Cleaning	45
Εικόνα 4.1.5.2 Οθόνη Cleaning απενεργοποιημένου Housekeeping.....	46
Εικόνα 4.1.5.3 Επιλογή ημερομηνίας για Αίτημα Καθαρισμού Δωματίου	47
Εικόνα 4.1.5.4 Επιλογή ώρας και λεπτού για Αίτημα Καθαρισμού Δωματίου	48
Εικόνα 4.1.5.5 Οθόνη Cleaning με αίτημα που δεν έχει «ανέβει»	49
Εικόνα 4.1.6 Οθόνη About.....	50
Εικόνα 4.1.7 Η οθόνη Rating.....	51
Εικόνα 4.2.1 Η σελίδα Login	52
Εικόνα 4.2.2 Η σελίδα Reception.....	53
Εικόνα 4.2.3.1 Η σελίδα Restaurant	54

Εικόνα 4.2.3.2 Η σελίδα Cleaning	56
Εικόνα 4.2.4.1 Η σελίδα Rating.....	57
Εικόνα 4.2.4.2 Η σελίδα About Us	58

Εισαγωγή

Η παρούσα πτυχιακή χωρίζεται σε τέσσερα κεφάλαια, ώστε να γίνει κατανοητή τόσο η θεωρητική επεξήγηση όσο και η ανάλυση του πρακτικού μέρους της.

Στο πρώτο κεφάλαιο γίνεται η εισαγωγή στην αναγκαιότητα του προγραμματισμού λογισμικού στη σημερινή εποχή, στις ανάγκες για ανάπτυξη λογισμικών και στο λογισμικό που αναπτύχθηκε στη συγκεκριμένη εργασία. Αναφέρονται παραδείγματα σχετικών εργασιών με την παρούσα, που χρησιμοποιούνται από ξενοδοχεία στη σύγχρονη εποχή. Επιπρόσθετα, παρουσιάζεται η ανάγκη για την ανάπτυξη τέτοιου είδους εφαρμογών, καθώς όσο αυξάνεται η τεχνολογία και οι δυνατότητές της τόσο αυξάνονται και οι ανάγκες όσον αφορά τον κλάδο των ξενοδοχείων. Παράλληλα, περιγράφεται η ανάλυση των αντίστοιχων εφαρμογών που έχουν αναπτυχθεί.

Στο δεύτερο κεφάλαιο εξετάζονται οι τεχνολογίες που χρησιμοποιήθηκαν στην παρούσα εργασία. Πιο συγκεκριμένα, παρουσιάζεται το κάθε framework που χρησιμοποιήθηκε. Στο πλαίσιο αυτού του κεφαλαίου, περιγράφονται τα πλεονεκτήματα της κάθε μίας τεχνολογίας και γιατί επιλέχθηκε, αλλά επιδεικνύονται και αντίστοιχες τεχνολογίες που θα μπορούσαν να αξιοποιηθούν εναλλακτικά.

Στο τρίτο κεφάλαιο αναλύεται το σύστημα που αναπτύχθηκε. Αρχικά, παρουσιάζεται το διάγραμμα οντοτήτων – σχέσεων και το σχεσιακό μοντέλο του συστήματος. Στη συνέχεια, περιγράφεται η λειτουργικότητα των συστημάτων, το πώς πραγματοποιείται η ενημέρωση των δεδομένων από και προς τους πελάτες και αντίστοιχα από και προς την διοίκηση του ξενοδοχείου.

Στο τέταρτο κεφάλαιο περιγράφεται το Εγχειρίδιο Χρήστη των εφαρμογών του συστήματος που έχει δημιουργηθεί για την παρούσα πτυχιακή.

1. Βιβλιογραφική Ανασκόπηση

1.1 Λογισμικό

Ο προγραμματισμός λογισμικού είναι η διαδικασία δημιουργίας, σχεδίασης, ανάπτυξης, και διαχείρισης λογισμικού ή εφαρμογών υπολογιστών. Οι μηχανικοί λογισμικού χρησιμοποιούν γλώσσες προγραμματισμού και εργαλεία για να δημιουργήσουν λογισμικό που εκτελεί συγκεκριμένες λειτουργίες ή επιλύει προβλήματα [1].

Ο προγραμματισμός λογισμικού είναι ουσιώδης στον σύγχρονο κόσμο. Η ανάγκη για λογισμικό εκτείνεται σε πολλούς τομείς, από την επιχειρηματική και επιστημονική κοινότητα έως τον τομέα της ψυχαγωγίας. Οι εφαρμογές λογισμικού καλύπτουν μια ευρεία γκάμα λειτουργιών, όπως διαχείριση δεδομένων, αυτοματοποίηση εργασιών, ανάλυση δεδομένων, επικοινωνία, ψυχαγωγία και πολλά άλλα.

Στον σύγχρονο κόσμο, ο προγραμματισμός λογισμικού είναι αναπόσπαστο κομμάτι της τεχνολογικής εξέλιξης. Οι εφαρμογές και τα λογισμικά χρησιμοποιούνται για τη βελτίωση της απόδοσης, την αυτοματοποίηση διαδικασιών, τη διευκόλυνση της επικοινωνίας, την ανάλυση δεδομένων και την καινοτομία. Ο προγραμματισμός λογισμικού απαιτεί εξειδικευμένες γνώσεις και δεξιότητες, και οι επαγγελματίες προγραμματιστές αποτελούν ουσιαστικό κομμάτι της σύγχρονης κοινωνίας που συμβάλλει στη συνεχή τεχνολογική εξέλιξη, στην καινοτομία.

1.1.1 Ιστορική Αναδρομή

Ο προγραμματισμός υπάρχει αιώνες τώρα. Έχει εντοπιστεί τον 9ο αιώνα μ.Χ. μία προγραμματιζόμενη ακολουθία μουσικής, που δημιουργήθηκε από τους Πέρσες αδερφούς Banu Musa [2]. Στην πιο κοντινή στη σύγχρονη μορφή του, εντοπίζεται το 1843 όπου και αναγνωρίζεται επίσημα ως η χρονιά δημιουργίας του πρώτου λογισμικού για υπολογιστή. Αυτός θεωρείται ο αλγόριθμος που ανέπτυξε η μαθηματικός Άντα Λάβλεϊς σχετικά με τον υπολογισμό μιας ακολουθίας αριθμών Bernoulli, που προορίζονταν να εκτελεστεί από την Αναλυτική Μηχανή του Τσαρλς Μπάμπατζ. Συχνά η Άντα Λάβλεϊς αναγνωρίζεται ως η πρώτη μηχανικός λογισμικού στην ιστορία [3].

Τη δεκαετία του 1880 ο Χέρμαν Χόλεριθ επινόησε την έννοια της αποθήκευσης δεδομένων σε μορφή αναγνωρίσιμη από μηχανές. Η Μηχανή Τούρινγκ εφευρέθηκε το 1936 από τον «πατέρα της επιστήμης υπολογιστών», μαθηματικό Άλαν Τούρινγκ. Μια μηχανή Τούρινγκ μπορεί να προσαρμοστεί ώστε να προσομοιώνει την λογική οποιουδήποτε αλγορίθμου, και είναι ιδιαίτερα χρήσιμη στο να εξηγεί τις λειτουργίες μιας κεντρικής μονάδας επεξεργασίας (επεξεργαστή) στο εσωτερικό του υπολογιστή. [4] Το 1945 ο Γερμανός Πολιτικός Μηχανικός Κόνραντ Ζούσε αρχίζει να αναπτύσσει το Plan Calculus, την πρώτη αλγοριθμική γλώσσα προγραμματισμού, ενώ το 1948 ο Tom

Kilburn αναπτύσσει έναν από τους πρώτους υπολογιστές, τον «The Manchester Small», όπου εκτελεί μαθηματικούς υπολογισμούς [5].

Το 1949 εισήχθη η έννοια του υπολογιστή αποθηκευμένου προγράμματος στο πανεπιστήμιο του Cambridge, που τα προγράμματα και τα δεδομένα αποθηκευόταν και χειρίζονταν με τον ίδιο τρόπο στη μνήμη του υπολογιστή. Το 1952 η μαθηματικός Grace Hopper ολοκληρώνει το «A-0», ένα λογισμικό που επιτρέπει τον χρήστη ενός υπολογιστή να χρησιμοποιεί Αγγλικές λέξεις αντί για αριθμούς ώστε να δίνει οδηγίες σε έναν υπολογιστή [6].

Με την αρχή του «αγώνα του διαστήματος» μεταξύ Σοβιετικής Ένωσης και ΗΠΑ, το 1955, με κορυφαία στιγμή το ταξίδι του Σοβιετικού Κοσμοναύτη Γιούρι Γκαγκάριν στο διάστημα, παρατηρείται πληθώρα ανάπτυξης λογισμικού και υλικού [7]. Στα τέλη της δεκαετίας του 1960, ο όρος «Software Factory», ήταν δημοφιλής και σχετιζόταν με πρόσθετα εργαλεία στους υπολογιστές και συστήματα διαχείρισης υπολογιστή και σχετικών ηλεκτρονικών εργαλείων [8]. Τη δεκαετία του 1970, οι προγραμματιστές δημιούργησαν γλώσσες προγραμματισμού, όπως η C, που επέτρεπαν την πιο αποδοτική ανάπτυξη λογισμικού, ενθαρρύνοντας την επαναχρησιμοποίηση κώδικα και την διεπαφή με υλικό υπολογιστή.

Την δεκαετία του 80, η ανάπτυξη λογισμικού επεκτάθηκε σε διάφορους τομείς όπως τα βιντεοπαιχνίδια και οι εφαρμογές γραφείου. Η δημιουργία λογισμικού γίνεται ολοένα και πιο προσιτή για περισσότερους ανθρώπους, ενώ από το 1990 με την επανάσταση του Διαδικτύου δημιουργούνται συνεχώς λογισμικά διαφόρων ειδών [5].

1.2 Έξυπνο Ξενοδοχείο

Η καινοτομία ήταν πάντα σύμμαχος του τουριστικού τομέα και, τώρα περισσότερο από ποτέ, η τεχνολογία έχει γίνει βασικό στοιχείο για την τόνωση και εξέλιξη του τουρισμού. Όσες εταιρείες συνεχίσουν να προσδεύουν στη διαδικασία ψηφιοποίησής τους, θα βρουν ένα ουσιαστικό πλεονέκτημα που θα τους επιτρέψει να ηγηθούν μίας νέας εποχής, η οποία θα χαρακτηρίζεται από υψηλό ανταγωνισμό, αποτελεσματικότητα και βελτιωμένες υπηρεσίες/παροχές. Στη σύγχρονη εποχή, η ταχύτητα και η αμεσότητα είναι βασικά στοιχεία της καθημερινότητας. Με την έλευση των νέων τεχνολογιών, η συμπεριφορά των χρηστών έχει αλλάξει. Στη δεκαετία του 1980 ήταν αδιανόητο να μπορείς να πληρώνεις τη διαμονή από μια κινητή συσκευή με δύο μόνο κλικ. Με τον ίδιο τρόπο, η άφιξη τεχνολογιών τελευταίας γενιάς, όπως το Internet of Things (IoT), τα Big Data ή η Τεχνητή Νοημοσύνη, μας σε νέα πρότυπα διαχείρισης των ξενοδοχείων. Παράλληλα, λοιπόν, με τα «έξυπνα κινητά» (Smartphone) ή τις «έξυπνες τηλεοράσεις» (Smart TV), μπορούμε πλέον να αναφερόμαστε και σε «έξυπνα ξενοδοχεία» (Smart Hotel).

Το Smart Hotel είναι ένα ξενοδοχείο εξοπλισμένο με ριζοσπαστικές τεχνολογίες καινοτομίας, όπως η Τεχνητή Νοημοσύνη και το IoT (Internet of Things), με στόχο τη

βελτίωση της διαχείρισης, της αποτελεσματικότητας και του ελέγχου των υπηρεσιών από τη διοίκηση του ξενοδοχείου, καθώς και την προσφορά καλύτερης εξυπηρέτησης στους επισκέπτες. Αυτό επιτυγχάνεται με την εφαρμογή αισθητήρων στα διάφορα στοιχεία του κτιρίου, είτε των επίπλων. Τα αντικείμενα αυτά είναι σε θέση να στέλνουν δεδομένα συνεχώς σε μια κεντρική πλατφόρμα, η οποία τα λαμβάνει σε πραγματικό χρόνο, τα επεξεργάζεται και τα μετατρέπει σε πολύτιμες πληροφορίες για τους υπεύθυνους και το προσωπικό διαχείρισης ξενοδοχείων [9].

Τα έξυπνα ξενοδοχεία έχουν σχεδιαστεί για να προσφέρουν διαμονή βελτιωμένη ως προς τις υπηρεσίες και την άνεση των επισκεπτών, επιτρέποντας τη βέλτιστη διαχείριση τους προς όφελος της ασφάλειας, της αποτελεσματικότητας και της ικανότητας ελέγχου. Βασικά τους χαρακτηριστικά και υπηρεσίες συνήθως διακρίνονται σε: χαρακτηριστικά και υπηρεσίες συνήθως διακρίνονται σε:

Βελτίωση πρακτικών/διαδικαστικών ζητημάτων, όπως η είσοδος του επισκέπτη και η καταχώρισή του στο σύστημα πελατών του ξενοδοχείου. Ειδικότερα, προβλέπεται συχνά αυτόματο check-in/check-out, βάσει του οποίου οι επισκέπτες μπορούν να κάνουν check-in και check-out χρησιμοποιώντας εφαρμογές για κινητά ή περίπτερα αυτοεξυπηρέτησης, μειώνοντας την ανάγκη για παραδοσιακές αλληλεπιδράσεις στη ρεσεψιόν. Επίσης, μπορεί να προβλέπεται δυνατότητα εισόδου χωρίς κλειδί, ώστε οι επισκέπτες να χρησιμοποιούν τα smartphone ή τις έξυπνες κάρτες τους για πρόσβαση στα δωμάτιά τους, εξαλείφοντας την ανάγκη για φυσικά κλειδιά ή κάρτες-κλειδιά.

Ως προς την ασφάλεια, μπορούν να εφαρμοστούν ενισχυμένα μέτρα ασφαλείας, όπως ψηφιακές κλειδαριές θυρών και συστήματα επιτήρησης, τα οποία συχνά ενσωματώνονται για την αύξηση της ασφάλειας και την προστασία της ιδιωτικής ζωής των επισκεπτών. Επιπλέον, λόγω της ανάπτυξης των ανέπαφων πληρωμών και της αυξημένης συχνότητάς τους, σε πολλά ξενοδοχεία οι επισκέπτες πλέον μπορούν να πληρώσουν για υπηρεσίες χρησιμοποιώντας ψηφιακά πορτοφόλια μέσω κινητού ή άλλες μεθόδους ανέπαφης πληρωμής για την ελαχιστοποίηση της φυσικής επαφής.

Η ανάγκη για αξιολόγηση των υπηρεσιών από τους πελάτες οδηγεί στη συγκέντρωση σχολίων των επισκεπτών. Δηλαδή, τα ξενοδοχεία χρησιμοποιούν τεχνολογικά μέσα για να συλλέγουν τα σχόλια των πελατών τους και να ενεργούν σύμφωνα με αυτά, αντιμετωπίζοντας τις ανησυχίες τους και βελτιώνοντας τη συνολική τους εμπειρία.

Αξίζει να σημειωθεί η βελτίωση εμπειρίας εντός του δωματίου, για παράδειγμα με τον χειρισμό των συσκευών με φωνητική εντολή ή την παροχή προηγμένων συσκευών τεχνολογίας. Συγκεκριμένα, όλο και περισσότερα δωμάτια έξυπνων ξενοδοχείων διαθέτουν συστήματα που ενεργοποιούνται με τη φωνή (π.χ. Amazon Alexa ή Google Assistant), τα οποία επιτρέπουν στους επισκέπτες να ελέγχουν τον φωτισμό, τη

Θερμοκρασία, τις κουρτίνες και τα συστήματα ψυχαγωγίας με φωνητικές εντολές. Άλλη μία παροχή είναι ο έξυπνος φωτισμός και ο έλεγχος του κλίματος. Τα αυτοματοποιημένα συστήματα φωτισμού και HVAC προσαρμόζουν τις ρυθμίσεις με βάση τις προτιμήσεις των επισκεπτών ή την πληρότητα, εξοικονομώντας ενέργεια, μειώνοντας τα έξοδα του ξενοδοχείου και αυξάνοντας την άνεση των επισκεπτών.

Συχνά, τα έξυπνα δωμάτια διαθέτουν tablet ή οθόνες αφής, που παρέχουν πληροφορίες σχετικά με τις ξενοδοχειακές υπηρεσίες, αλλά και τα τοπικά αξιοθέατα και τουριστικές δραστηριότητες, και δίνουν τη δυνατότητα στους επισκέπτες να ζητήσουν επιπλέον υπηρεσίες, όπως προγραμματισμένη καθαριότητα ή υπηρεσία δωματίου. Στα σύγχρονα ξενοδοχεία, είναι απαραίτητη η ύπαρξη Wi-Fi υψηλής ταχύτητας για την προσωπική σύνδεση των επισκεπτών, συμπεριλαμβανομένης της πλοήγησης στο διαδίκτυο και των διαδικτυακών συσκέψεων.

Πολύ σημαντική είναι και η παροχή εξατομικευμένων εμπειριών, καθώς αρκετά ξενοδοχεία χρησιμοποιούν αναλυτικά στοιχεία δεδομένων και τεχνητή νοημοσύνη για να προσαρμόσουν σε ατομικό επίπεδο τις εμπειρίες των επισκεπτών, όπως να προτείνουν προτιμήσεις θερμοκρασίας δωματίου, να παρέχουν υπηρεσίες στοχευμένες στις επιθυμίες του επισκέπτη ή να προσφέρουν εξατομικευμένες προτάσεις. Πολλά έξυπνα ξενοδοχεία παρέχουν, επίσης, πρόσβαση σε πλατφόρμες ψυχαγωγίας, όπως το Netflix και το Hulu, μέσω έξυπνων τηλεοράσεων, ώστε οι επισκέπτες να έχουν πρόσβαση στο περιεχόμενο που προτιμούν. Τέλος, ορισμένα ξενοδοχεία διαθέτουν έξυπνους καθρέφτες, οι οποίοι εμφανίζουν πληροφορίες, ημερήσιες μετεωρολογικές προβλέψεις ή ακόμη και εικονικές εμπειρίες δοκιμής για είδη μόδας.

Παροχή βελτιωμένων υπηρεσιών, όπως η ύπαρξη ψηφιακού θυρωρού, η αυτοματοποιημένη υπηρεσία δωματίου (room service) και η διαχείριση της ενέργειας. Ειδικότερα, οι εικονικές υπηρεσίες θυρωρού είναι συχνά διαθέσιμες μέσω εφαρμογών ή συσκευών στο δωμάτιο, και προσφέρουν συστάσεις, πραγματοποιούν κρατήσεις ή απαντούν σε ερωτήματα επισκεπτών. Επίσης, οι επισκέπτες μπορούν να ζητήσουν υπηρεσία δωματίου ή ειδικές παροχές μέσω εφαρμογών, μειώνοντας την ανάγκη για διαπροσωπικές αλληλεπιδράσεις. Τέλος, η εγκατάσταση έξυπνων αισθητήρων και συστημάτων αυτοματισμού βοηθούν τα ξενοδοχεία να μειώσουν την κατανάλωση ενέργειας σβήνοντας τα και απενεργοποιώντας τα συστήματα HVAC, όταν τα δωμάτια δεν είναι κατειλημμένα.

Τα έξυπνα ξενοδοχεία στοχεύουν στην παροχή μιας απρόσκοπτης και εξατομικευμένης εμπειρίας στους επισκέπτες, βελτιώνοντας παράλληλα τη λειτουργική αποτελεσματικότητα για το προσωπικό του ξενοδοχείου. Ωστόσο, το επίπεδο τεχνολογίας και υπηρεσιών μπορεί να διαφέρει από το ένα ξενοδοχείο στο άλλο, ανάλογα με τον προϋπολογισμό, την αγορά-στόχο και την τεχνολογική υποδομή τους [10].

Εκτός από τα βασικά χαρακτηριστικά και τις υπηρεσίες που αναφέρθηκαν προηγουμένως, ακολουθούν ορισμένες πρόσθετες πτυχές και εξειδικευμένες τεχνολογίες που σχετίζονται με τα έξυπνα ξενοδοχεία:

Χρήση εξελιγμένης τεχνολογίας, όπως εικονικής πραγματικότητας ή τεχνητής νοημοσύνης. Ειδικότερα, τα έξυπνα ξενοδοχεία συχνά ενσωματώνουν συσκευές και αισθητήρες Internet of Things (IoT) για την άμεση επικοινωνία μεταξύ διαφόρων συστημάτων. Για παράδειγμα, οι αισθητήρες μπορούν να ανιχνεύσουν τότε ένα δωμάτιο δεν είναι κατειλημμένο και να προσαρμόσουν αυτόματα τις ρυθμίσεις φωτισμού και θερμοκρασίας (η λειτουργία αυτή μπορεί να συνδυαστεί και με τις υπηρεσίες διαχείρισης της ενέργειας, βλ. ανωτέρω). Επιπλέον, ορισμένα έξυπνα ξενοδοχεία ενδέχεται να προσφέρουν εμπειρίες Επαυξημένης Πραγματικότητας (AR) ή Εικονικής Πραγματικότητας (VR) για να ενισχύσουν την εμπειρία και την ικανοποίηση των επισκεπτών. Αυτό θα μπορούσε να περιλαμβάνει εικονικές περιηγήσεις σε τοπικά αξιοθέατα ή καθλητικές εμπειρίες εντός του ίδιου του δωματίου.

Η χρήση τεχνολογιών όπως η Τεχνητή Νοημοσύνη (AI) και η Μηχανική Μάθηση επιτρέπει στα ξενοδοχεία να αναλύουν τη συμπεριφορά και τις προτιμήσεις των επισκεπτών, για να προσφέρουν πιο εξατομικευμένες εμπειρίες. Ένα τέτοιο παράδειγμα είναι τα chatbot AI, τα οποία μπορούν να παρέχουν άμεσες απαντήσεις σε οποιοδήποτε ερώτημα του επισκέπτη. Ορισμένα ξενοδοχεία ενδέχεται να παρέχουν υπηρεσίες μέσω ρομπότ, χρησιμοποιώντας ρομπότ για εργασίες, όπως παροχή υπηρεσίας δωματίου, καθαρισμό και υπηρεσίες θυρωρείου. Αυτά τα ρομπότ μπορούν να λειτουργούν αυτόνομα ή να ελέγχονται από απόσταση.

Βιώσιμη διαχείριση του ξενοδοχείου και ασφάλεια, όπως η ενσωμάτωση οικολογικών πρακτικών και η προστασία προσωπικών δεδομένων των επισκεπτών. Συγκεκριμένα, στον τομέα της ενεργειακής απόδοσης και της βιωσιμότητας, τα έξυπνα ξενοδοχεία συχνά ενσωματώνουν πράσινες τεχνολογίες, όπως ενεργειακά αποδοτικό φωτισμό, ηλιακούς συλλέκτες και συστήματα εξοικονόμησης νερού, για να ελαχιστοποιήσουν τις περιβαλλοντικές επιπτώσεις τους. Πολλά έξυπνα ξενοδοχεία εγκαθιστούν Συστήματα Διαχείρισης Έξυπνων Κτιρίων, τα οποία ελέγχουν διάφορες πτυχές της υποδομής του ξενοδοχείου, συμπεριλαμβανομένης της θέρμανσης, του εξαερισμού, του κλιματισμού (HVAC), του φωτισμού και της ασφάλειας, για βέλτιστη απόδοση και άνεση (η λειτουργία αυτή μπορεί να συνδυαστεί και με τις υπηρεσίες διαχείρισης ενέργειας και την αξιοποίηση του IoT, βλ. ανωτέρω). Επιπλέον, είναι σημαντικές οι έξυπνες λύσεις στάθμευσης, με τεχνολογίες όπως η αναγνώριση πινακίδων κυκλοφορίας για αυτοματοποιημένη πρόσβαση στάθμευσης και πληρωμή.

Ως προς την ασφάλεια των δεδομένων και του απορρήτου, πρέπει απαραίτητα να λαμβάνονται ισχυρά μέτρα κυβερνοασφάλειας για την προστασία των πληροφοριών των επισκεπτών από μη εξουσιοδοτημένη πρόσβαση ή παραβιάσεις, δεδομένου

μάλιστα και του αυξημένου όγκου των συλλεγόμενων δεδομένων. Τέλος, είναι απολύτως απαραίτητα τα συστήματα απόκρισης έκτακτης ανάγκης, στο πλαίσιο των οποίων τα έξυπνα ξενοδοχεία μπορεί να ενσωματώνουν προηγμένα χαρακτηριστικά ασφαλείας, όπως αυτοματοποιημένες ειδοποιήσεις έκτακτης ανάγκης, οδηγίες εκκένωσης και επικοινωνία σε πραγματικό χρόνο με τους επισκέπτες σε κρίσιμες καταστάσεις.

Επιπλέον υπηρεσίες για την προστασία των επισκεπτών, όπως τεχνολογίες υγείας και ευεξίας. Αρκετά έξυπνα ξενοδοχεία προσφέρουν τεχνολογίες σχετικές με την υγεία των πελατών, όπως ανιχνευτές γυμναστικής, οθόνες ύπνου και ανέσεις ευεξίας στο δωμάτιο. Άλλη σχετική τεχνολογία είναι η τεχνολογία Blockchain, το οποίο μπορεί να χρησιμοποιηθεί σε έξυπνα ξενοδοχεία για την επαλήθευση της ταυτότητας των επισκεπτών, διασφαλίζοντας το απόρρητο και διευκολύνοντας τις ασφαλείς συναλλαγές. Τέλος, στο πλαίσιο της επέκτασης του διεθνούς πελατολογίου τους, τα έξυπνα ξενοδοχεία ενδέχεται να προσφέρουν πολύγλωσση υποστήριξη, όπως πολύγλωσσους ψηφιακούς βοηθούς ή μεταφραστικές υπηρεσίες.

Αξίζει να σημειωθεί ότι οι συγκεκριμένες τεχνολογίες και υπηρεσίες που προσφέρονται από τα έξυπνα ξενοδοχεία μπορεί να διαφέρουν σε μεγάλο βαθμό ανάλογα με την τοποθεσία, την επωνυμία, τον προϋπολογισμό και το πελατειακό κοινό του ξενοδοχείου. Επιπλέον, καθόσον η τεχνολογία συνεχίζει να εξελίσσεται, τα έξυπνα ξενοδοχεία πιθανότατα θα ενσωματώσουν ακόμη περισσότερα χαρακτηριστικά και πρωτοποριακές υπηρεσίες στο μέλλον.

Στο πλαίσιο της τεχνολογικής εξέλιξης και της αξιοποίησής της προς την βελτίωση της εμπειρίας των επισκεπτών, το Διαδίκτυο των Πραγμάτων (IoT) παίζει σημαντικό ρόλο στην επίτευξη αυτού του στόχου. Το IoT αναφέρεται στη διασύνδεση διαφόρων συσκευών και συστημάτων, η οποία τους επιτρέπει να συλλέγουν και να ανταλλάσσουν δεδομένα. Ειδικά στην περίπτωση των έξυπνων ξενοδοχείων, το IoT χρησιμοποιείται για τη δημιουργία ενός δικτύου συσκευών και αισθητήρων που μπορούν να επικοινωνούν και να συνεργάζονται για τη βελτίωση των υπηρεσιών επισκεπτών και της λειτουργικής αποτελεσματικότητάς τους.

Οι αισθητήρες και οι συσκευές IoT στα δωμάτια επισκεπτών μπορούν να παρακολουθούν και να ελέγχουν παροχές όπως ο φωτισμός, το κλίμα και τα στόρια. Για παράδειγμα, οι αισθητήρες μπορούν να ανιχνεύσουν πότε ένα δωμάτιο δεν είναι κατειλημμένο και να προσαρμόσουν τις ρυθμίσεις για εξοικονόμηση ενέργειας. Οι έξυπνοι θερμοστάτες, ο φωτισμός και τα συστήματα HVAC στα δωμάτια του ξενοδοχείου μπορούν να συνδεθούν με συστήματα κεντρικής διαχείρισης. Αυτό επιτρέπει αυτοματοποιημένες ρυθμίσεις με βάση την πληρότητα και τις συνθήκες περιβάλλοντος, με αποτέλεσμα την εξοικονόμηση ενέργειας.

Οι συσκευές IoT μπορούν να θυμούνται και να προβλέπουν τις προτιμήσεις των επισκεπτών. Για παράδειγμα, εάν ένας επισκέπτης προτιμά μια συγκεκριμένη θερμοκρασία δωματίου, το σύστημα μπορεί να προσαρμοστεί αυτόματα σε αυτήν τη ρύθμιση όταν αυτός επιστρέψει. Το IoT μπορεί να ενισχύσει και την ασφάλεια του ξενοδοχείου, ενσωματώνοντας κάμερες παρακολούθησης, αισθητήρες κίνησης και συστήματα ελέγχου πρόσβασης. Οι έξυπνες κλειδαριές με δυνατότητες IoT μπορούν επίσης να παρέχουν προηγμένες λειτουργίες ασφαλείας, όπως απομακρυσμένο έλεγχο πρόσβασης και παρακολούθηση σε πραγματικό χρόνο. Ως προς τις δυνατότητες συντήρησης και επισκευών, οι αισθητήρες IoT μπορούν να παρακολουθούν την κατάσταση διαφόρων συστημάτων και εξοπλισμού του ξενοδοχείου. Αυτό επιτρέπει την προγνωστική συντήρηση, όπου τα ζητήματα εντοπίζονται και αντιμετωπίζονται πριν οδηγήσουν σε μεγαλύτερα προβλήματα.

Οι συσκευές IoT μπορούν να συλλέγουν και να αναλύουν τη συμπεριφορά και τις προτιμήσεις των επισκεπτών. Αυτές οι πληροφορίες μπορούν να χρησιμοποιηθούν για την προσφορά εξατομικευμένων υπηρεσιών, όπως η πρόταση για κοντινά αξιοθέατα ή η σύσταση συγκεκριμένων επιλογών για φαγητό. Το IoT μπορεί να χρησιμοποιηθεί για την οργάνωση των λειτουργιών του προσωπικού. Για παράδειγμα, το προσωπικό καθαριότητας μπορεί να ειδοποιείται άμεσα όταν ένα δωμάτιο εκκενώνεται, ώστε τα δωμάτια να καθαρίζονται έγκαιρα, όπως ήδη αναφέρθηκε.

Συνοπτικά, το IoT διαδραματίζει κρίσιμο ρόλο στη λειτουργία των έξυπνων ξενοδοχείων, επιτρέποντας την απρόσκοπτη ενοποίηση διαφόρων συστημάτων και συσκευών για τη βελτίωση της εμπειρίας των επισκεπτών και τη βελτιστοποίηση των λειτουργιών του ξενοδοχείου [11].

Η τεχνολογία έξυπνων ξενοδοχείων δίνει τη δυνατότητα στους ξενοδόχους να βελτιστοποιήσουν το λειτουργικό τους κόστος. Μέσω της τεχνολογίας Internet of Things (IoT), διάφορες συσκευές και συστήματα εντός του ξενοδοχείου μπορούν να διασυνδεθούν, επιτρέποντας την παρακολούθηση σε πραγματικό χρόνο και την αποτελεσματική διαχείριση. Για παράδειγμα, οι αισθητήρες κίνησης μπορούν να ανιχνεύσουν πληρότητα στα δωμάτια επισκεπτών, επιτρέποντας στα ξενοδοχεία να βελτιστοποιήσουν την κατανάλωση ενέργειας ρυθμίζοντας αυτόματα τη θερμοκρασία δωματίου και απενεργοποιώντας τον κλιματισμό όταν τα δωμάτια δεν είναι κατειλημμένα. Με τον τρόπο αυτό, επιτυγχάνεται και η πράσινη ανάπτυξη των ξενοδοχείων, αλλά και ολόκληρου του τουριστικού κλάδου, με αποτέλεσμα τη συνολική εξέλιξη του.

Η υιοθέτηση της τεχνολογίας έξυπνων ξενοδοχείων δεν περιορίζεται σε μία τοποθεσία ή αλυσίδα ξενοδοχείων. Είναι μια παγκόσμια τάση που αναδιαμορφώνει τον κλάδο. Για παράδειγμα, στη Σιγκαπούρη, αρκετά ξενοδοχεία έχουν αγκαλιάσει τις έξυπνες τεχνολογικές εξελίξεις, όπως το Yotel, το οποίο διαθέτει λύσεις υψηλής

τεχνολογίας, ανέπαφες. Στο Λας Βέγκας και τη Νέα Υόρκη διάσημες αλυσίδες ξενοδοχείων, όπως οι Marriott, Hilton, Wynn και άλλες έχουν εφαρμόσει τεχνολογία έξυπνων ξενοδοχείων για να βελτιώσουν τις εμπειρίες των επισκεπτών και τις λειτουργίες τους.

1.3 Ανασκόπηση εφαρμογών Έξυπνων Ξενοδοχείων

1.3.1 Microsoft Smart Hotel 360

Το Microsoft SmartHotel360 είναι ένα ολοκληρωμένο παράδειγμα εφαρμογής που χρησιμοποιείται από τη Microsoft για να παρουσιάσει διάφορες τεχνολογίες Microsoft Azure cloud, τεχνητής νοημοσύνης (AI) και Internet of Things (IoT). Έχει σχεδιαστεί ως εφαρμογή αναφοράς για να βοηθήσει τους μηχανικούς λογισμικού και τους οργανισμούς να μάθουν και να κατανοήσουν πώς να δημιουργούν σύγχρονες, εγγενείς εφαρμογές στο cloud χρησιμοποιώντας το Microsoft Azure και άλλες τεχνολογίες της Microsoft. Το SmartHotel360 δεν είναι μια πραγματική αλυσίδα ξενοδοχείων. Αντίθετα, χρησιμεύει ως δείγμα έργου για την επίδειξη των δυνατοτήτων των υπηρεσιών και εργαλείων cloud της Microsoft.

Ακολουθούν ορισμένα βασικά χαρακτηριστικά και στοιχεία που σχετίζονται συνήθως με την εφαρμογή αναφοράς SmartHotel360:

- Το SmartHotel360 περιλαμβάνει εφαρμογές ιστού και κινητών που παρέχουν στους επισκέπτες του ξενοδοχείου λειτουργίες όπως κράτηση δωματίου, check-in, check-out και αιτήματα για υπηρεσία δωματίου.
- Η εφαρμογή έχει σχεδιαστεί για να αξιοποιεί διάφορες υπηρεσίες Azure, όπως η υπηρεσία Azure App Service, οι λειτουργίες Azure, το Azure Cosmos DB, το Azure IoT Hub, το Azure Machine Learning και άλλα.
- Το SmartHotel360 συχνά περιλαμβάνει συσκευές και αισθητήρες IoT που τοποθετούνται σε δωμάτια ξενοδοχείου και κοινόχρηστους χώρους για τη συλλογή δεδομένων σχετικά με την πληρότητα, τη θερμοκρασία και άλλους περιβαλλοντικούς παράγοντες.
- Η εφαρμογή ενσωματώνει AI και αναλυτικά στοιχεία για να επεξεργάζεται δεδομένα από συσκευές IoT, να αναλύει τις προτιμήσεις των πελατών και να κάνει εξατομικευμένες προτάσεις στους επισκέπτες.
- Το SmartHotel360 περιλαμβάνει συνήθως ένα σύστημα κρατήσεων που επιτρέπει στους επισκέπτες να κάνουν κράτηση δωματίων και υπηρεσιών online και ενσωματώνεται με άλλα συστήματα διαχείρισης ξενοδοχείων.
- Η εφαρμογή μπορεί να διαθέτει chatbots ή εικονικούς βοηθούς που μπορούν να βοηθήσουν τους επισκέπτες με κοινές ερωτήσεις και αιτήματα.

- Λειτουργίες ασφαλείας, όπως ο έλεγχος ταυτότητας και η εξουσιοδότηση, επιδεικνύονται συχνά στην εφαρμογή, παρουσιάζοντας την ταυτότητα και τις υπηρεσίες ασφαλείας της Microsoft.
- Το SmartHotel360 μπορεί να ενσωματωθεί με πύλες πληρωμής για την ασφαλή διαχείριση των πληρωμών των επισκεπτών.
- Η εφαρμογή επιδεικνύει βέλτιστες πρακτικές για την ανάπτυξη και τη διαχείριση σύγχρονων εφαρμογών εγγενών στο cloud χρησιμοποιώντας αγωγούς Azure DevOps και CI/CD.

Το SmartHotel360 χρησιμοποιείται κυρίως για εκπαιδευτικούς σκοπούς και ως σημείο αναφοράς για μηχανικούς λογισμικού που θέλουν να μάθουν για τις υπηρεσίες Azure και τις σύγχρονες πρακτικές ανάπτυξης εφαρμογών. Παρέχει ένα πρακτικό παράδειγμα για το πώς να δημιουργήσετε μια επεκτάσιμη, εγγενή εφαρμογή στο cloud χρησιμοποιώντας τεχνολογίες της Microsoft [12].

1.3.2 Hilton's Connected Room

Το Hilton Connected Room είναι μια τεχνολογική πρωτοβουλία της Hilton Hotels & Resorts που στοχεύει να προσφέρει στους επισκέπτες μια πιο εξατομικευμένη και βολική εμπειρία ξενοδοχείου μέσω της χρήσης καινοτόμου τεχνολογίας.

Μερικά από τα πιο γενικά χαρακτηριστικά γνωρίσματα του Hilton Connected Room είναι τα εξής:

- Οι επισκέπτες μπορούν να χρησιμοποιήσουν την εφαρμογή Hilton Honors για κινητά για να κάνουν check-in, να επιλέξουν το δωμάτιό τους και να χρησιμοποιήσουν το smartphone τους ως ψηφιακό κλειδί για πρόσβαση στο δωμάτιό τους.
- Οι επισκέπτες μπορούν να χρησιμοποιήσουν την εφαρμογή για να ελέγξουν διάφορες λειτουργίες του δωματίου, όπως φωτισμό, θερμοκρασία και ρυθμίσεις τηλεόρασης. Αυτό επιτρέπει μια πιο εξατομικευμένη και άνετη διαμονή.
- Ροή περιεχομένου: Οι έξυπνες τηλεοράσεις στα συνδεδεμένα δωμάτια συχνά υποστηρίζουν υπηρεσίες ροής περιεχομένου όπως το Netflix, το Hulu και άλλα, επιτρέποντας στους επισκέπτες να παρακολουθούν τις αγαπημένες τους εκπομπές και ταινίες.
- Η εφαρμογή μπορεί να προσφέρει εξατομικευμένες προτάσεις για φαγητό, διασκέδαση και τοπικά αξιοθέατα με βάση τις προτιμήσεις των επισκεπτών και τις προηγούμενες διαμονές.
- Ορισμένα Connected Rooms ενδέχεται να ενσωματώνουν βοηθούς που ενεργοποιούνται με φωνή, όπως το Alexa του Amazon ή το Google Assistant,

επιτρέποντας στους επισκέπτες να ελέγχουν τις λειτουργίες του δωματίου και να ζητούν πληροφορίες χρησιμοποιώντας φωνητικές εντολές.

- Οι επισκέπτες μπορούν να χρησιμοποιήσουν την εφαρμογή για να ζητήσουν πρόσθετες υπηρεσίες, όπως υπηρεσία δωματίου, καθαριότητα ή συντήρηση, καθιστώντας πιο βολική την επικοινωνία με το προσωπικό του ξενοδοχείου.
- Το Hilton Connected Room μπορεί να παρέχει έναν μηχανισμό σχολίων που επιτρέπει στους επισκέπτες να παρέχουν σχόλια σε πραγματικό χρόνο για τη διαμονή τους, βοηθώντας το ξενοδοχείο να βελτιώσει την ποιότητα των υπηρεσιών.

1.3.3 Marriott's IoT Guestroom Lab

Το Marriott's IoT (Internet of Things) Guestroom Lab είναι μια πρωτοβουλία της Marriott International, μιας από τις μεγαλύτερες αλυσίδες ξενοδοχείων στον κόσμο που συνεργάζεται με την Samsung και την Legrand, για να εξερευνήσει και να δοκιμάσει καινοτόμες τεχνολογίες και λύσεις IoT για να βελτιώσει την εμπειρία των επισκεπτών στα δωμάτια του ξενοδοχείου. [13] Παρακάτω ακολουθούν ορισμένες βασικές λεπτομέρειες σχετικά με το IoT Guestroom Lab της Marriott:

- Ενσωμάτωση IoT: Το εργαστήριο IoT Guestroom της Marriott εστιάζει στην ενσωμάτωση διαφόρων συσκευών και τεχνολογιών IoT σε δωμάτια επισκεπτών για να τα κάνει πιο έξυπνα και βολικά.
- Εξατομίκευση: Η πρωτοβουλία στοχεύει να προσφέρει στους επισκέπτες μια εξαιρετικά εξατομικευμένη εμπειρία χρησιμοποιώντας αισθητήρες και συσκευές IoT για την κατανόηση και την κάλυψη των ατομικών προτιμήσεων.
- Βοηθεί με φωνητικό έλεγχο: Ορισμένες ιδιοκτησίες της Marriott έχουν πειραματιστεί με βοηθούς που ελέγχονται με φωνή, όπως η Alexa της Amazon, για να επιτρέπουν στους επισκέπτες να ελέγχουν τις λειτουργίες του δωματίου και να έχουν πρόσβαση σε πληροφορίες χρησιμοποιώντας φωνητικές εντολές.
- Συνδεδεμένος φωτισμός: Μπορεί να εγκατασταθούν συστήματα φωτισμού IoT που επιτρέπουν στους επισκέπτες να προσαρμόσουν τα επίπεδα φωτισμού και τη θερμοκρασία χρώματος ανάλογα με τις προτιμήσεις τους.
- Έξυπνοι καθρέφτες: Ορισμένα δωμάτια μπορεί να διαθέτουν έξυπνους καθρέφτες που μπορούν να εμφανίζουν πληροφορίες όπως προγνώσεις καιρού, ειδήσεις και εξατομικευμένα μηνύματα.
- Καινοτόμος ψυχαγωγία: Η Marriott έχει εξερευνήσει την ενσωμάτωση υπηρεσιών ροής και διαδραστικών επιλογών ψυχαγωγίας στα δωμάτια επισκεπτών, επιτρέποντας στους επισκέπτες να έχουν εύκολη πρόσβαση στο αγαπημένο τους περιεχόμενο.

- Έλεγχος κλίματος: Οι αισθητήρες και οι θερμοστάτες IoT μπορούν να χρησιμοποιηθούν για να παρέχουν ακριβή έλεγχο της θερμοκρασίας του δωματίου και της ποιότητας του αέρα.
- Ενεργειακή απόδοση: Οι τεχνολογίες IoT χρησιμοποιούνται συχνά για τη βελτιστοποίηση της χρήσης ενέργειας στα δωμάτια φιλοξενίας, μειώνοντας τα απόβλητα και τις περιβαλλοντικές επιπτώσεις.
- Ασφάλεια και απόρρητο δεδομένων: Η Marriott λαμβάνει σοβαρά υπόψη την ασφάλεια των δεδομένων και το απόρρητο των επισκεπτών, εφαρμόζοντας μέτρα για την προστασία των δεδομένων των επισκεπτών και τη διασφάλιση ενός ασφαλούς περιβάλλοντος IoT.
- Σχόλια και δοκιμές: Το εργαστήριο IoT Guestroom της Marriott χρησιμεύει ως πεδίο δοκιμών για νέες τεχνολογίες και έννοιες. Τα σχόλια των επισκεπτών και τα δεδομένα από αυτές τις αίθουσες δοκιμών βοηθούν τη Marriott να βελτιώσει και να βελτιώσει την εμπειρία των επισκεπτών [14].

1.3.4 CitizenM's mobile app

Το CitizenM είναι μια αλυσίδα ξενοδοχείων γνωστή για την καινοτόμο προσέγγισή της στη φιλοξενία και την εμπειρία των επισκεπτών. Προσφέρουν μια εφαρμογή για κινητά που συμπληρώνει το σύγχρονο ήθος των ξενοδοχείων τους με γνώμονα την τεχνολογία. Παρακάτω παρουσιάζονται ορισμένες λειτουργίες, που σχετίζονται κυρίως με την εφαρμογή για κινητά του CitizenM:

- Easy Booking: Η εφαρμογή επιτρέπει στους χρήστες να περιηγούνται και να κάνουν κράτηση δωματίων σε ξενοδοχεία CitizenM σε όλο τον κόσμο. Οι χρήστες μπορούν να επιλέξουν την επιθυμητή τοποθεσία, να ελέγξουν τη διαθεσιμότητα και να κάνουν κρατήσεις.
- Check-in και check-out μέσω κινητού: Οι επισκέπτες μπορούν να χρησιμοποιήσουν την εφαρμογή για μια απρόσκοπτη διαδικασία check-in και check-out. Αυτή η λειτουργία βοηθά στη μείωση του χρόνου αναμονής στη ρεσεψιόν και προσφέρει μια πιο βολική εμπειρία άφιξης και αναχώρησης.
- Είσοδος χωρίς κλειδί: Ορισμένα ξενοδοχεία CitizenM προσφέρουν είσοδο χωρίς κλειδί μέσω της εφαρμογής. Οι επισκέπτες μπορούν να χρησιμοποιήσουν τα smartphone τους ως ψηφιακά κλειδιά για πρόσβαση στα δωμάτιά τους.
- Χειριστήρια δωματίου: Η εφαρμογή μπορεί να παρέχει έλεγχο σε λειτουργίες του δωματίου, όπως φωτισμό, θερμοκρασία, περσίδες και ρυθμίσεις τηλεόρασης. Αυτό το επίπεδο προσαρμογής επιτρέπει στους επισκέπτες να δημιουργήσουν ένα άνετο περιβάλλον σύμφωνα με τις προτιμήσεις τους.

- Τροποποιήσεις κράτησης: Οι χρήστες μπορούν να τροποποιήσουν τις κρατήσεις τους, να παρατείνουν τη διαμονή τους ή να ζητήσουν πρόσθετες υπηρεσίες μέσω της εφαρμογής, μειώνοντας την ανάγκη για άμεση αλληλεπίδραση με το προσωπικό του ξενοδοχείου.
- Ανεπαφικές πληρωμές: Η εφαρμογή συχνά υποστηρίζει ασφαλείς πληρωμές μέσω κινητού τηλεφώνου, επιτρέποντας στους επισκέπτες να εξοφλούν τους λογαριασμούς τους χωρίς να χρειάζεται να χειρίζονται φυσικά μετρητά ή κάρτες.
- Τοπικές προτάσεις: Η εφαρμογή του CitizenM μπορεί να παρέχει στους επισκέπτες συστάσεις για κοντινά εστιατόρια, αξιοθέατα και δραστηριότητες με βάση την τοποθεσία και τις προτιμήσεις τους.
- Ιστορικό διαμονής και αφοσίωση: Η εφαρμογή μπορεί να διατηρεί αρχείο του ιστορικού διαμονής των επισκεπτών, διευκολύνοντας τους συχνούς επισκέπτες να έχουν πρόσβαση στις προηγούμενες κρατήσεις τους και να κερδίζουν ανταμοιβές αφοσίωσης.
- Υπηρεσίες επισκεπτών: Οι επισκέπτες μπορούν να ζητήσουν πρόσθετες υπηρεσίες, όπως υπηρεσία δωματίου ή καθαριότητας, μέσω της εφαρμογής, βελτιώνοντας τη συνολική εμπειρία τους.
- Σχόλια και Κριτικές: Η εφαρμογή συχνά περιλαμβάνει λειτουργίες για τους επισκέπτες που μπορούν να υποβάλλουν σχόλια, κριτικές ή αξιολογήσεις με βάση τη διαμονή τους, βοηθώντας το CitizenM να βελτιώνει συνεχώς τις υπηρεσίες του [15].

2. Τεχνολογίες

Στο παρόν κεφάλαιο παρουσιάζονται οι τεχνολογίες οι οποίες χρησιμοποιήθηκαν για την εκπόνηση της εργασίας.

2.1 Xamarin



Εικόνα 2.1 Xamarin Logo

Πηγή: <https://dotnet.microsoft.com/en-us/apps/xamarin>

Για την ανάπτυξη της εφαρμογής των πελατών του ξενοδοχείου, χρησιμοποιήθηκε η τεχνολογία Xamarin. Η τεχνολογία αυτή, πρόκειται στην ουσία για μια δημοφιλής πλατφόρμα ανάπτυξης εφαρμογών για κινητές συσκευές. Το Xamarin επιτρέπει στους μηχανικούς λογισμικού να δημιουργούν εφαρμογές με έναν, κοινό κώδικα γραμμένο στη γλώσσα C#, αλλά να διανέμονται σε διαφορετικά λειτουργικά συστήματα, όπως Android και iOS. Το Xamarin έτσι δίνει την δυνατότητα επαναχρησιμοποίησης του κώδικα μεταξύ διαφορετικών λειτουργικών συστημάτων, ώστε ο προγραμματιστής να μπορεί να αναπτύξει μια εφαρμογή που τρέχει σε διαφορετικά λειτουργικά συστήματα χωρίς να χρειάζεται να αναπτύξει διαφορετικό κώδικα κάθε φορά ανάλογα το λειτουργικό σύστημα. Έτσι αυτό το framework μειώνει τον χρόνο ανάπτυξης μιας εργασίας σε συσκευές με διαφορετικό λειτουργικό σύστημα. Ουσιαστικά το Xamarin λειτουργεί ως γέφυρα μεταξύ της C# και των γηγενών εργαλείων ανάπτυξης των εκάστοτε λειτουργικών συστημάτων [16].

Το Xamarin οπότε έχει τρόπο ενοποιημένης ανάπτυξης, όπου είναι εφικτή η ανάπτυξη κοινού κώδικα για έκδοση εφαρμογών σε διαφορετικά λειτουργικά συστήματα. Οι εφαρμογές αυτές είναι γηγενείς στα λειτουργικά τους συστήματα, καθώς με κοινό κώδικα, το Xamarin μπορεί να παράγει αρχεία είτε .apk (εκτελέσιμα αρχεία στο Android) είτε .ipa (εκτελέσιμα αρχεία για iPhone), είτε ακόμα και .exe (για Windows). Οι εφαρμογές τύπου Xamarin.Android μεταγλωττίζονται από C# σε Java, αντίστοιχα οι Xamarin.iOS σε Objective-C ή Swift. Γίνεται έτσι η αλληλεπίδραση με τις εργαλειοθήκες ανάπτυξης των εκάστοτε λειτουργικών συστημάτων, δηλαδή Android

SDK και Xcode αντίστοιχα. Για όλη αυτή την λειτουργικότητα το Xamarin χρησιμοποιεί την πλατφόρμα λογισμικού Mono.

Το Xamarin αναπτύσσεται και παράγεται από την ομώνυμη εταιρεία, η οποία ιδρύθηκε τον Μάιο του 2011 στο San Francisco. Ιδρύθηκε από μηχανικούς οι οποίοι συμμετείχαν στην δημιουργία του Mono, του Xamarin.Android (που ονομαζόταν Mono for Android) και του Xamarin.iOS (που ονομαζόταν MonoTouch), τα οποία είναι cross-platform υλοποιήσεις των CLI και Common Language Specifications (δηλαδή της Microsoft .NET) [17]. Αγοράστηκε από την Microsoft τον Φεβρουάριο του 2016, επιτρέποντας στην Microsoft να ενσωματώσει την τεχνολογία Xamarin στο οικοσύστημα της και να ενισχύσει την υποστήριξη για την ανάπτυξη εφαρμογών cross-platform[18]. Μεταξύ γνωστών εφαρμογών που έχουν αναπτυχθεί με αυτή την τεχνολογία, είναι η εφαρμογή για την Alaska Airlines, η εφαρμογή Captio, η Storygo[19].

Σύμφωνα με το Xamarin πάνω από 1.4 εκατομμύρια μηχανικοί λογισμικού το χρησιμοποιούσαν σε 120 χώρες του πλανήτη τον Απρίλιο του 2017. Το γεγονός αυτό, σε συνδυασμό με το ότι σε εφαρμογές τύπου Xamarin ο κώδικας είναι C# .NET, συμβάλει στο να γίνεται εύκολη η ανάπτυξη κώδικα, καθώς υπάρχει αρκετή κοινότητα να απαντήσει σε ερωτήσεις και να συμβάλει στην τεκμηρίωση του κάθε προβλήματος που συμβαίνει σε έναν προγραμματιστή. Υπάρχει δηλαδή ζωντανό κοινό σε γνωστά δίκτυα προγραμματιστών, όπως είναι το StackOverflow.

Παρόμοια λειτουργικότητα με το Xamarin έχουν κι άλλα frameworks[20]:

- Το **Flutter** της Google είναι ένα αρκετά γνωστό open-source παράδειγμα, με γλώσσα προγραμματισμού την Dart, όπου έχει κοινό κώδικα για διάφορα λειτουργικά συστήματα.
- Το Framework **React Native** επίσης, με γλώσσα προγραμματισμού την Javascript, χρησιμοποιεί κλασικές μεθόδους σχεδίασης web, όπως HTML και CSS και είναι κατάλληλο για ανάπτυξη εφαρμογής σε iOS και Android.
- Το **Apache Cordova** (γνωστό και ως PhoneGap), είναι ένα framework ανοικτού τύπου κώδικα, που επιτρέπει στους μηχανικούς λογισμικού να δημιουργήσουν κινητές εφαρμογές χρησιμοποιώντας HTML, CSS και JavaScript. Ο κώδικας αυτός μπορεί να εκτελεστεί σε πολλά λειτουργικά συστήματα, συμπεριλαμβανομένων των Android και iOS.
- Το **Vue Native** είναι μια επέκταση του Framework Vue.js, όπου χρησιμοποιώντας τη σύνταξη και βιβλιοθήκες του Vue.js γίνεται ανάπτυξη εφαρμογών για Android και iOS με κοινό κώδικα.
- Το **NativeScript** είναι ένα εξίσου open-source framework που επιτρέπει στους μηχανικούς λογισμικού να δημιουργήσουν κινητές εφαρμογές χρησιμοποιώντας JavaScript, TypeScript ή Angular. Ο κώδικας μπορεί να τρέξει επίσης σε Android και σε iOS.

2.2 Flask



Εικόνα 2.2 Flask Logo

Πηγή: <https://flask.palletsprojects.com/en/2.3.x/>

Για την ανάπτυξη του API και την διαχείριση της βάσης δεδομένων χρησιμοποιήθηκε η τεχνολογία Flask. Το Flask είναι ένα microframework που χρησιμοποιεί γλώσσα Python, είναι ανοικτού κώδικα (open source) και χρησιμεύει για την ανάπτυξη web εφαρμογών. Είναι ελαφρύ, ευέλικτο και διαθέτει έναν εύκολο στη χρήση δρομολογητή (router) για την διαχείριση των αιτημάτων HTTP. Το Flask δεν περιλαμβάνει πολλές προεπιλεγμένες δυνατότητες, αλλά επιτρέπει στους προγραμματιστές να επιλέξουν και να ενσωματώσουν δικές τους βιβλιοθήκες και λειτουργίες, καθιστώντας το ιδανικό για εξατομικευμένες εφαρμογές.

Ορίζεται ως microframework, καθώς δεν χρειάζεται συγκεκριμένα εργαλεία ή βιβλιοθήκες. Δεν χρειάζεται κάτι που να προϋπάρχει σε βιβλιοθήκες τρίτων που παρέχουν κοινές λειτουργίες. Ωστόσο το Flask υποστηρίζει επεκτάσεις που μπορούν να προσθέσουν δυνατότητες εφαρμογής σαν να είχαν υλοποιηθεί στο ίδιο το Flask [21] [22].

Το Flask δημιουργήθηκε από τον Armin Ronacher και κυκλοφόρησε για πρώτη φορά το 2010. Ο ίδιος ήταν μέλος του Pocco, μια παγκόσμια ομάδα από φίλους της Python. Σύμφωνα με τον Armin, η ιδέα του Flask ήταν αρχικά ένα πρωταπριλιάτικο αστείο. Η Pocco, ίδρυσε διάφορα Projects τα οποία χρησιμοποιούνται στο Flask, όπως το Werkzeug, που αποτελεί μια βιβλιοθήκη για την Python για Web Server Gateway Interface, το πρότυπο Jinja για την Python, τις βιβλιοθήκες MarkupSafe και ItsDangerous, όπου η τελευταία χρησιμοποιείται για να αποθηκεύει μια session μιας εφαρμογής flask σε cookie.

Πρόκειται για μια από τις πιο δημοφιλείς τεχνολογίες για τον σχεδιασμό API αλλά και συνολικά backend. Σε έρευνα του statista.com τον Ιούνιο του 2023 είναι από τα πιο δημοφιλή web frameworks. Ενδεικτικό είναι το γεγονός πως μεγάλες πλατφόρμες, όπως το LinkedIn, το Netflix και το Reddit χρησιμοποιούν Flask. Όλα αυτά καθιστούν το Flask να έχει μια από τις μεγαλύτερες κοινότητες διαδικτυακά, η οποία και αυτή μπορεί να στηρίξει το έργο ενός προγραμματιστή. Χαρακτηριστικό το γεγονός πως έχουν ανέβει μέχρι σήμερα, τον Οκτώβριο του 2023, πάνω από 50.000 διαφορετικές ερωτήσεις στο γνωστό site Stack Overflow, με tag "Flask"[23].

Λόγω της ανοικτής φύσης του, το Flask έχει αναπτυχθεί και συνεχίζει να εξελίσσεται από προγραμματιστές διεθνώς. Δηλαδή, πολλοί μηχανικοί λογισμικού σε όλο τον κόσμο συνεισφέρουν στην ανάπτυξη του Flask, προσθέτοντας νέα χαρακτηριστικά, διορθώνοντας σφάλματα και βελτιώνοντας το πλαίσιο για την ανάπτυξη web εφαρμογών. Είναι ανοικτό στο github στην διεύθυνση <https://github.com/pallets/flask>.

Υπάρχουν διαφορετικές τεχνολογίες οι οποίες θα μπορούσαν να είχαν αναπτυχθεί εναλλακτικά του Flask. Αξίζει να σημειωθούν μερικές [24]:

- **Laravel (PHP):** Το Laravel είναι ένα framework για την ανάπτυξη web εφαρμογών και API. Διαθέτει μια ευέλικτη αρχιτεκτονική και πολλά χρήσιμα εργαλεία για τη δημιουργία αξιόπιστων API.
- **Ruby on Rails (Ruby):** Το Ruby on Rails είναι ένα framework για την ανάπτυξη web εφαρμογών και API. Είναι γνωστό για την ευκολία χρήσης και την ταχύτητα ανάπτυξης που προσφέρει. Αξίζει να σημειωθεί πως έχει μεγάλη ζήτηση στην αγορά.
- **ExpressJS (JavaScript):** Το ExpressJS είναι ένα ελαφρύ και ευέλικτο framework για την ανάπτυξη API σε JavaScript.
- **ASP.NET (C#):** Το ASP.NET είναι το framework ανάπτυξης της Microsoft για τον χώρο του Web και των εφαρμογών. Προσφέρει ισχυρές λειτουργίες για την ανάπτυξη ασφαλών και αποδοτικών API.
- **Django (Python):** Το Django είναι ένα ανοιχτού κώδικα Python framework για την ανάπτυξη web εφαρμογών και API. Έχει μια ισχυρή κοινότητα και προσφέρει πολλά εργαλεία για τη διαχείριση της βάσης δεδομένων, την αυθεντικοποίηση, και την ανάπτυξη γρήγορων API.

Κάθε ένα από αυτά τα frameworks έχει τα πλεονεκτήματά του και τις ειδικότητές του. Η επιλογή του κατάλληλου framework εξαρτάται από τα απαιτούμενα χαρακτηριστικά της εφαρμογής, τις γνώσεις της ομάδας και τις προσωπικές προτιμήσεις των εκάστοτε προγραμματιστών.

2.3 SQLite



Εικόνα 2.3 SQLite Logo

Πηγή: https://www.sqlite.org/images/sqlite370_banner.gif

Για την δημιουργία της βάσης δεδομένων, χρησιμοποιήθηκε η τεχνολογία SQLite. Η SQLite είναι μια βιβλιοθήκη στη γλώσσα C, που υλοποιεί έναν μηχανισμό SQL βάσης δεδομένων, μικρό, γρήγορο, αυτόνομο και υψηλής αξιοπιστίας. Η SQLite είναι ο πιο δημοφιλής μηχανισμός βάσης δεδομένων στον κόσμο [25].

Γενικά, η SQLite είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, που περιέχεται σε μια C προγραμματιστική βιβλιοθήκη. Ως Σύστημα Διαχείρισης Βάσεων Δεδομένων (ΣΔΒΔ), ορίζεται ένα λογισμικό μέσω του οποίου γίνεται η δημιουργία, η διαχείριση, η συντήρηση και η χρήση μιας ηλεκτρονικής βάσης δεδομένων, ανάλογα με τον τύπο βάσης δεδομένων που επιλέγεται. Οι σημαντικές προσεγγίσεις ΣΔΒΔ είναι τέσσερις:

- Η ιεραρχική
- Η δικτυακή
- Η σχεσιακή
- Η αντικειμενοστραφής

Η προσέγγιση των σχεσιακών βάσεων δεδομένων εμφανίστηκε αρχικά περίπου το 1980. Μετά από αρκετά χρόνια πειραματισμού των χρηστών, έγινε η πιο διαδεδομένη προσέγγιση των ΣΔΒΔ και παρέμεινε έτσι από τότε. Το σχεσιακό μοντέλο δεδομένων παριστάνει δεδομένα και τις σχέσεις τους ως ένα σύνολο πινάκων. Κάθε πίνακας αποτελείται από στήλες με μοναδικά ονόματα. Μια γραμμή του πίνακα παριστάνει μια σχέση ανάμεσα σε ένα σύνολο από τιμές [26].

Η SQL αποτελεί την πιο διαδεδομένη γλώσσα διαχείρισης σχεσιακών βάσεων δεδομένων και είναι ορισμένη παγκόσμια ως πρότυπο. Η SQL παρέχει δυνατότητες για:

- τον ορισμό, τη διαγραφή και τη μεταβολή πινάκων και κλειδιών
- τη σύνταξη ερωτήσεων (queries),
- την εισαγωγή, διαγραφή και μεταβολή στοιχείων,
- τον ορισμό όψεων (views) πάνω στα δεδομένα,
- τον ορισμό δικαιωμάτων πρόσβασης,
- τον έλεγχο της ακεραιότητας των στοιχείων,
- τον έλεγχο συναλλαγών (transaction)

Η SQLite πρόκειται οπότε για ένα κλασικό παράδειγμα σχεσιακού ΣΔΒΔ. Σε αντίθεση με τα περισσότερα, το SQLite δεν διαθέτει ξεχωριστή διαδικασία διακομιστή. Το SQLite διαβάζει και καταγράφει απευθείας τις πληροφορίες σε συνηθισμένα αρχεία δίσκου, όπου εμπεριέχεται μια πλήρης βάση δεδομένων SQL. Η μορφή αρχείου της βάσης δεδομένων είναι cross-platform. Αυτά τα χαρακτηριστικά καθιστούν την SQLite μια δημοφιλή επιλογή. Η SQLite επίσης πρόκειται για μία συμπαγής βιβλιοθήκη. Με όλες τις δυνατότητες ενεργοποιημένες, το μέγεθος της βιβλιοθήκης μπορεί να είναι

μικρότερο από 750KB, ανάλογα με το λειτουργικό σύστημα και τις ρυθμίσεις βελτιστοποίησης του μεταγλωττιστή (compiler) [27].

Εξίσου για την παρούσα εργασία θα μπορούσε να χρησιμοποιηθεί μια παρόμοια τεχνολογία όπως το SQLite. Αναφέρονται παραδειγματικά[28]:

- Η MySQL είναι ένα Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων ανοιχτού κώδικα που είναι γνωστό για την απόδοση και την αξιοπιστία του. Είναι μια καλή επιλογή για εφαρμογές web και πιο σύνθετα έργα που βασίζονται σε βάσεις δεδομένων.
- Το PostgreSQL είναι ένα ακόμα σχεσιακό ΣΔΒΔ ανοιχτού κώδικα που είναι γνωστό για τις προηγμένες δυνατότητες και την επεκτασιμότητα του. Είναι μια ισχυρή επιλογή για εφαρμογές που απαιτούν πολύπλοκα ερωτήματα και ακεραιότητα δεδομένων.
- Το MariaDB είναι μια παραλλαγή της MySQL και προσφέρει συμβατότητα αντικατάστασης drop-in. Είναι γνωστό ότι είναι πιο ανοιχτό και έχει κάποιες βελτιώσεις απόδοσης σε σχέση με τη MySQL.
- Το MongoDB είναι μια βάση δεδομένων NoSQL που έχει σχεδιαστεί για τη διαχείριση μη δομημένων ή ημιδομημένων δεδομένων. Είναι μια εξαιρετική επιλογή για εφαρμογές όπου η ευελιξία και η επεκτασιμότητα είναι ζωτικής σημασίας.
- Για την ανάπτυξη μιας εφαρμογής σε Windows, ο Microsoft SQL Server είναι μια ισχυρή επιλογή. Προσφέρει χαρακτηριστικά όπως υψηλή διαθεσιμότητα και προτείνετε από την Microsoft.

2.4 Βασικά Στοιχεία Ανάπτυξης Ιστοσελίδων



Εικόνα 2.4 HTML, JS, CSS

Πηγή: <https://www.medium.com/levelupweb/%20kUcnzFif1UJBKHE8oi5c6g.jpeg>

Για την δημιουργία της ιστοσελίδας για την διοίκηση του ξενοδοχείου, χρησιμοποιήθηκαν τα Βασικά Στοιχεία Ανάπτυξης Ιστοσελίδων. Δηλαδή, ο κώδικας είναι γραμμένος σε HTML, Javascript για την προγραμματιστική λογική του κώδικα και CSS για την εμφάνιση της σελίδας. Παρακάτω γίνεται βασική ανασκόπηση του τι είναι αυτά τα τρία βασικά εργαλεία.

2.4.1 HTML

Η HTML είναι το ακρωνύμιο των λέξεων Hypertext Markup Language (Γλώσσα Σήμανσης Υπερκειμένου). Η HTML αρχικά βασιζόταν στην Standard Generalized Markup Language (SGML – Πρότυπη Γλώσσα Γενικευμένης Σήμανσης), ένα πολύ μεγαλύτερο, πιο περίπλοκο σύστημα επεξεργασίας εγγράφων. Για να γραφτεί HTML δεν χρειάζεται γνώση της SGML.

Η HTML ορίζει ένα σύνολο κοινών στοιχείων για ιστοσελίδες: επικεφαλίδες, παραγράφους, λίστες και πίνακες. Επίσης ορίζει μορφοποιήσεις χαρακτήρων, όπως έντονα και παραδείγματα κώδικα. Αυτά τα στοιχεία και οι μορφοποιήσεις δηλώνονται μέσα σε έγγραφα HTML χρησιμοποιώντας προσαρτήματα (tags). Κάθε προσάρτημα έχει ένα συγκεκριμένο όνομα και διακρίνεται απ' το περιεχόμενο του εγγράφου, χρησιμοποιώντας έναν συμβολισμό, που συζητούμε λίγο αργότερα.

Η HTML δεν περιγράφει την διάταξη της ιστοσελίδας. Τα προσαρτήματα HTML δηλώνουν απλώς ότι ένα στοιχείο είναι μια επικεφαλίδα ή μια λίστα. Δεν λένε τίποτε για το πώς μορφοποιείται η επικεφαλίδα ή η λίστα. Τα προγράμματα περιήγησης (browsers), λειτουργούν και ως συστήματα μορφοποίησης HTML. Επίσης, η HTML έχει ένα καθορισμένο σύνολο προσαρτημάτων, που μπορούν να αξιοποιηθούν από προγραμματιστές. Δεν είναι εφικτή η δημιουργία νέων τύπων προσαρτημάτων, ώστε να δημιουργηθούν νέα στυλ ή χαρακτηριστικά.

Τα αρχεία HTML είναι έγγραφα απλού κειμένου (ASCII), γεγονός που σημαίνει πως δεν περιέχουν πληροφορίες οι οποίες αφορούν μια συγκεκριμένη πλατφόρμα ή ένα συγκεκριμένο πρόγραμμα. Κάθε συντάκτης κειμένου μπορεί να χρησιμοποιηθεί για να τα δημιουργήσει [29].

2.4.2 Javascript

Η γλώσσα προγραμματισμού Javascript χρησιμοποιείται για να μετατρέπει ιστοσελίδες σε εφαρμογές. Η Javascript χρησιμοποιείται για να χειρίζεται τα περιεχόμενα μιας ιστοσελίδας και να επιτρέπει σε χρήστες να αλληλεπιδρούν με

ιστοσελίδες χωρίς να ξαναφορτώνουν την σελίδα. Επιτρέπει στον προγραμματιστή να χειρίζεται τα περιεχόμενα της ιστοσελίδας, αφότου αυτή φορτωθεί, κάνοντας δυνατή την παροχή άμεσης ανάδρασης στους χρήστες. Έτσι μπορούν οι χρήστες να κάνουν αλλαγή σε μία φόρμα και οι πόροι να φορτώσουν όπως είναι προγραμματισμένοι.

Η Javascript ολοκληρώνεται μέσα στον φυλλομετρητή (Browser). Με αυτόν τον τρόπο είναι δυνατή η διαχείριση διάφορων χαρακτηριστικών της συμπεριφοράς του browser, καθώς και αντικειμένων που περιλαμβάνονται στην σελίδα. Η Javascript χρησιμοποιεί καθοδηγούμενο από συμβάντα μοντέλο (event-driven model) εκτέλεσης. Όταν ενσωματώνεται κώδικας Javascript σε μία ιστοσελίδα δηλαδή, αυτός δεν εκτελείται μέχρι να πυροδοτηθεί το συμβάν που σχετίζεται με αυτόν.

Το προσάρτημα <script> στην HTML χρησιμοποιείται για να εμπεριέχει κώδικα Javascript, με τρόπο παρόμοιο με αυτόν που χρησιμοποιείται το προσάρτημα <style> για την CSS που θα δούμε στη συνέχεια. Υπάρχουν φυσικά και άλλοι τρόποι χρήσης Javascript σε μία ιστοσελίδα, αλλά για την παρούσα εργασία επιλέχτηκε αυτό το προσάρτημα[29].

Τέλος, η Javascript είναι γλώσσα υψηλού επιπέδου, γλώσσα σεναρίων που βασίζεται σε πρωτότυπα (prototype-based) και είναι δυναμική. Η σύνταξή της είναι επηρεασμένη από τη C. Η Javascript, αντιγράφει πολλές ονομασίες από την Java, αλλά γενικά οι δύο αυτές γλώσσες δεν σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm) και υποστηρίζει τον αντικειμενοστραφές, προστακτικό αλλά και συναρτησιακό στυλ προγραμματισμού [30], [31].

2.4.3 CSS

Η CSS (Cascading Style Sheets) είναι μια γλώσσα υπολογιστή, που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους, η οποία χρησιμοποιείται για τον σχεδιασμό και την βελτίωση της εμφάνισης των ιστοσελίδων. Η CSS προσφέρει τη δυνατότητα προσδιορισμού του πώς θα εμφανίζονται τα στοιχεία της HTML, όπως κείμενο, εικόνες, σύνδεσμοι και άλλα, στην οθόνη του χρήστη. Χρησιμοποιώντας την CSS, είναι εφικτός ο καθορισμός χρωμάτων, γραμματοσειρών, περιθωρίων και άλλων χαρακτηριστικών σχεδιασμού, επιτρέποντας την δημιουργία όμορφων ιστοσελίδων. Η CSS αποτελεί βασικό εργαλείο για κάθε σελίδα που επιδιώκει να προσφέρει μια εμφανίσιμη και ελκυστική εμπειρία στους χρήστες[32].

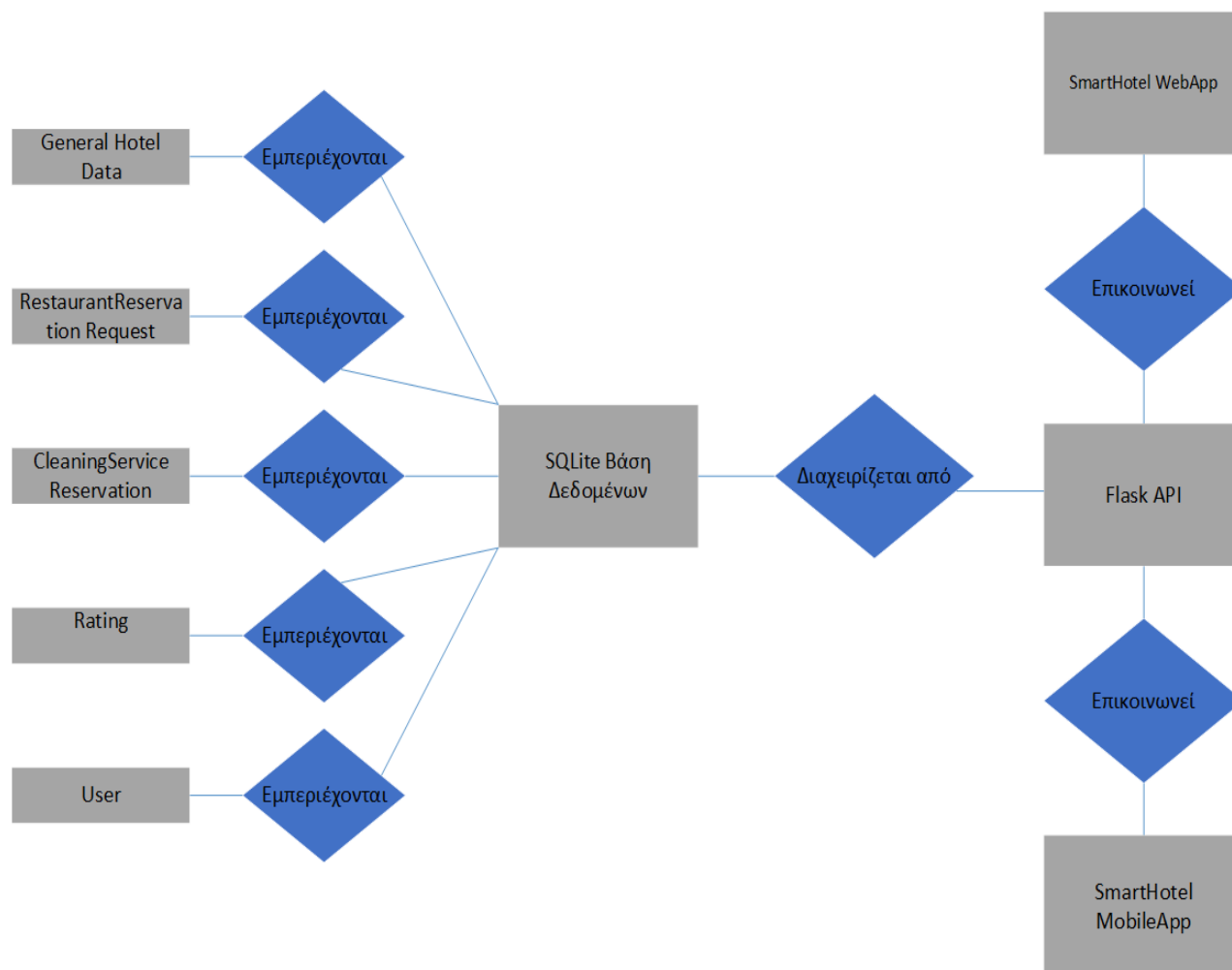
Η CSS δηλαδή είναι η γλώσσα που ορίζει πώς θα εμφανίζεται ένα αρχείο HTML στους φυλλομετρητές (browsers). Με την CSS ορίζεται το πώς θα προβάλλονται τα

στοιχεία της HTML, μέσω των προσαρτημάτων `<style>`, ή των αρχείων τύπου «.css». Δημιουργήθηκε από την κοινοπραξία του παγκόσμιου ιστού W3C (World Wide Web Consortium).

Κύρια χαρακτηριστικά της CSS είναι οι κανόνες. Ένας κανόνας CSS αποτελείται από τον επιλογέα (selector) που μας λέει σε ποιο στοιχείο HTML θα εφαρμοστεί ο κανόνας και από την δήλωση ή τις δηλώσεις (declarations) που μας λένε τι είδους στυλ θα εφαρμοστεί στον επιλογέα. Η κάθε δήλωση τελειώνει πάντα με το ελληνικό ερωτηματικό «;», ενώ το σύνολο των δηλώσεων για κάθε επιλογέα περιέχεται ανάμεσα σε αγκύλες «{» και «}».

Αντίστοιχες γλώσσες υπήρχαν ήδη, όπως η DSSSL και η FOSI, αλλά δεν ήταν κατάλληλες για τους σημερινούς τύπους των ιστοσελίδων σε περιβάλλοντα web [33].

3. Μοντέλο Δεδομένων



Εικόνα 3.1 Διάγραμμα Σχέσεων - Οντοτήτων

Στο παραπάνω διάγραμμα φαίνονται οι σχέσεις των Οντοτήτων του Συστήματος που αναπτύχθηκε. Παρατηρούνται τα tables που ανήκουν στη Βάση Δεδομένων, το ενδιάμεσο λογισμικό – API, η Διεπαφή της Κινητής Συσκευής για τους πελάτες του ξενοδοχείου και η Διεπαφή Ιστού για το προσωπικό και την διαχείριση. Το παρόν υποκεφάλαιο ασχολείται με τον σχηματισμό της βάσης δεδομένων, καθώς αφορά το μοντέλο δεδομένων που υλοποιεί όλο το σύστημα.

Η βάση δεδομένων έχει πέντε tables. Το User για να μπορεί κάποιος να κάνει login, είναι ένα table το οποίο αποτελείται από το Πρωτεύον Κλειδί id, που για κάθε καταχώρηση στη βάση έχει έναν διαφορετικό αριθμό. Αποτελείται από το αλφαριθμητικό πεδίο username, όπου αναγράφεται το όνομα κάθε χρήστη/διαχειριστή. Από το κωδικοποιημένο αλφαριθμητικό πεδίο hashed_password, όπου αναγράφεται ο κωδικός αφότου έχει κρυπτογραφηθεί με αλγόριθμο κατακερματισμού. Το επίσης αλφαριθμητικό πεδίο role, που είτε θα είναι “user” είτε “admin”, ώστε να γίνεται ο διαχωρισμός του ρόλου του χρήστη είτε σε απλό χρήστη-

πελάτη είτε σε διαχειριστή για το προσωπικό και την διοίκηση του ξενοδοχείου. Επίσης ένα πεδίο αριθμού με την ονομασία RoomNumber, για να είναι αντιληπτό το ποιο δωμάτιο σχετίζεται με τον κάθε πελάτη.

Το table General Hotel Data υπάρχει για να αναγράφονται οι γενικές πληροφορίες του ξενοδοχείου. Τα αλφαριθμητικά πεδία HotelName, HotelAddress, HotelInfo, AboutHotelContent και ReceptionTelephone, όπου εμπεριέχουν γενικές πληροφορίες του Ξενοδοχείου, σχετικές με τις ονομασίες τους, όπως και το εξίσου αλφαριθμητικό HotelEmail για την διεύθυνση του email. Επίσης εμπεριέχονται το αλφαριθμητικό RestaurantMenuLink, για την διαδικτυακή διεύθυνση του Menu του Restaurant και το Πρωτεύον Κλειδί id, αριθμός διαφορετικός για κάθε καταχώρηση στο table.

Το table RestaurantReservationRequest προσδιορίζει τον σκοπό του μέσω της ονομασίας του, σχετίζεται, δηλαδή, με την κράτηση τραπέζιού στο εστιατόριο του ξενοδοχείου. Αντιστοίχως και το CleaningServiceReservation Request αναφέρεται στην υποβολή αιτημάτων για την καθαριότητα του δωματίου. Και τα δύο εμπεριέχουν τα αναγκαία πεδία για αυτή την διεργασία, όπως το πρωτεύον κλειδί του αιτήματος, ένα αριθμητικό id και το αριθμητικό RoomNumber με το οποίο σχετίζονται. Για το εστιατόριο, RestaurantReservPeopleNumber για τον αριθμό των συνδαιτημόνων στο τραπέζι και τα αλφαριθμητικά RestaurantReservComment για τα σχόλια του αιτήματος και το RestaurantReservDateTime για το πότε να γίνει η κράτηση. Στο table της υπηρεσίας καθαρισμού, όπου χρειάζεται μόνο το αλφαριθμητικό CleaningServiceReservDateTime.

Υφίστανται πεδία τα οποία γίνονται επεξεργάσιμα μόνο από την διαχείριση του ξενοδοχείου στα παραπάνω δύο tables που αφορούν τα αιτήματα των πελατών. Η δυνατότητα αυτή συμβαίνει μέσα από την λειτουργία του ενδιάμεσου λογισμικού – API. Είναι το αριθμητικό RequestState, όπου ανάλογα με τον αριθμό γίνεται κατανοητό αν το αίτημα είναι σε αναμονή(1), εγκρίθηκε(2), απορρίφθηκε(3) ή παρήλθε(4). Η περίπτωση του να είναι 0 είναι μόνο για τοπική χρήση από τις εφαρμογές διεπαφής και σημαίνει πως το αίτημα δεν έχει σταλθεί. Επίσης υπάρχει το αλφαριθμητικό πεδίο AdminMessage ώστε να μπορεί να αφήνει ένα μήνυμα η διοίκηση σχετικό με το αίτημα του πελάτη. Αυτά τα δύο πεδία βρίσκονται και στα δύο tables.

Τέλος το πεδίο Ratings. Πρόκειται για το πεδίο που σχετίζεται με την κριτική βαθμολόγηση των πελατών στις υπηρεσίες του ξενοδοχείου. Εμπεριέχει τα αριθμητικά πεδία RateHospitality, RateComfort, RateLocation, RateFacilities, RateOverall και το πεδίο RoomNumber, ώστε να αναγράφεται το δωμάτιο με το οποίο σχετίζονταν η κριτική. Η βαθμολόγηση είναι από 0 έως 5.

Η βάση δεδομένων δημιουργείται αυτόματα μέσα από το ενδιάμεσο λογισμικό. Όπως αναγράφεται και στο 2^ο κεφάλαιο, το Σύστημα Διαχείρισης της είναι SQLite.

Εφαρμογή Κινητής Συσκευής και Πληροφοριακό Σύστημα για το Έξυπνο Ξενοδοχείο
Σμυρλής Γεώργιος, Κοσκινάς Χαράλαμπος

3.1 Ενδιάμεσο Λογισμικό

Για την δημιουργία του Ενδιάμεσου Λογισμικού χρησιμοποιήθηκε η τεχνολογία Flask. Μέσα από το ενδιάμεσο λογισμικό δημιουργείται και ενημερώνεται η Βάση Δεδομένων. Οι εφαρμογές κινητής συσκευής και ιστού μπορούν εύκολα και ασφαλώς να λάβουν και να ενημερώσουν τα κατάλληλα δεδομένα της Βάσης, σχετικά με την εκάστοτε ανάγκη. Το ενδιάμεσο λογισμικό φιλοξενείται σε μια διεύθυνση IP τοπικά ή διαδικτυακά, όπου κατορθώνει να έρχεται σε επικοινωνία και να διαχειρίζεται εκάστοτε την βάση δεδομένων, ορθά με βάση την Διεπαφή των πελατών και της διαχείρισης του ξενοδοχείου. Το ενδιάμεσο λογισμικό είναι τύπου REST.

Το ενδιάμεσο λογισμικό που υλοποιήθηκε στο παρόν σύστημα πρόκειται για αρχείο που ονομάστηκε `hotelapi` και έχει κατάληξη `py` λόγω του ότι είναι στη γλώσσα προγραμματισμού Python. Η λειτουργικότητες για εγγραφή νέου χρήστη, σύνδεση, ενημέρωση δεδομένων και λήψη δεδομένων γίνονται όλες μέσω του Ενδιάμεσου Λογισμικού. Το παρόν λογισμικό περιέχει τις αναγκαίες συναρτήσεις για την λειτουργικότητα ολόκληρου του συστήματος.

Για να κληθεί μια συνάρτηση χρειάζεται ένα HTTP Request στην διεύθυνση που κάνει host το API και ανάλογα την συνάρτηση σε συγκεκριμένη διαδρομή. Ένα HTTP Request μπορεί να είναι μόνο τύπου «GET», «POST» και «DELETE» στο συγκεκριμένο API. Το περιεχόμενο που επιστρέφει το ενδιάμεσο λογισμικό είναι τύπου JSON, όπως είναι και το περιεχόμενο που περιμένει σε ένα HTTP Request.

Έχει υλοποιηθεί συνάρτηση για την λειτουργία Login, για την είσοδο είτε του πελάτη είτε του χρήστη με σκοπό αλληλεπίδραση με την Βάση Δεδομένων μέσω του API. Οι διεπαφές, είτε η διεπαφή κινητής συσκευής είτε του Ιστού, στέλνουν σε μορφή JSON τα αλφαριθμητικά πεδία `username` και `password`, όπου αφότου το πεδίο `password` κρυπτογραφηθεί, αν τα δύο πεδία ταιριάζουν σε μια εγγραφή χρήστη στη Βάση Δεδομένων, το Ενδιάμεσο Λογισμικό επιστρέφει ένα Token στην εφαρμογή διεπαφής. Το token εμπεριέχει κωδικοποιημένα τον ρόλο του χρήστη, δηλαδή αν είναι `user` ή `admin`, τον αριθμό δωματίου του και το `username` του. Το HTTP Request χρειάζεται να είναι τύπου "POST" στη διαδρομή όπου η IP κάνει host το API, όπου ακολουθείται από τον αριθμό πύλης και τέλος το: «/login». Για παράδειγμα: «`http://192.168.1.12:5000/login`»

Για την ενημέρωση των General Hotel Data υφίσταται συνάρτηση `set_hotel_data`, η οποία περιμένει ένα JSON όπου εμπεριέχει μέσα τα πεδία του table. Επίσης και η παρούσα συνάρτηση για να υλοποιηθεί σωστά, χρειάζεται token να υπάγεται σε `admin`. Αν όλα είναι σωστά, το ενδιάμεσο λογισμικό ενημερώνει το table με τα νέα στοιχεία. Για την λήψη των General Hotel Data έχει υλοποιηθεί η συνάρτηση `get_hotel_data`, όπου επιστρέφει όλα τα πεδία του General Hotel Data, χωρίς να ελέγχει ποιος την

κάλεσε. Για να κληθούν οι παραπάνω συναρτήσεις, χρειάζεται αίτημα POST ή GET αντίστοιχα στη διαδρομή η οποία τελειώνει με «/api/hotel».

Για τα αιτήματα κράτησης είτε εστιατορίου είτε καθαριότητας, έχουν υλοποιηθεί ξεχωριστές συναρτήσεις. Για τις περιπτώσεις που ο πελάτης ζητάει να λάβει το τι αιτήματα έχει κάνει, το ενδιάμεσο λογισμικό του επιστρέφει μόνο τα δικά του αιτήματα. Αντίστοιχα, όταν ένας admin στέλνει HTTP Request τύπου “GET”, του επιστρέφονται όλα τα σχετικά αιτήματα που έχει κάνει οποιοσδήποτε. Για τους πελάτες, έχουν δημιουργηθεί οι κατάλληλες συναρτήσεις για την λήψη των αιτημάτων τους.

Στην περίπτωση που ο πελάτης θέλει να κάνει ένα νέο αίτημα, αυτό συμβαίνει μέσω της Διεπαφής Κινητής Συσκευής, όπου στέλνεται ένα HTTP Request τύπου “POST” στο ενδιάμεσο λογισμικό. Το αίτημα αυτό περιέχει σε μορφή JSON τα πεδία του αιτήματος που αφορούν το αίτημα του πελάτη. Για παράδειγμα, για ένα αίτημα καθαριότητας, το HTTP Request θα περιλαμβάνει JSON το οποίο θα περιέχει μόνο το πεδίο `CleaningServiceReservDateTime`, όπου αναγράφεται η συγκεκριμένη ώρα που θέλει ο πελάτης καθαρισμό του δωματίου του. Το αίτημα επίσης θα περιλαμβάνει στις κεφαλίδες του και το token του πελάτη, το οποίο θα αξιοποιήσει για να ενημερώσει την βάση για το ποιος πελάτης κάνει το παρών αίτημα. Η διεύθυνση τελειώνει σε «/api/hotel/cleaning». Έτσι το ενδιάμεσο λογισμικό, ελέγχει την εγκυρότητα του token και το αν αφορά έναν πελάτη, δηλαδή έναν user και τέλος κάνει νέα σειρά στο table `CleaningServiceReservations`, όπου θα γεμίσει τα πεδία “RoomNumber”, “CleaningServiceReservDateTime”. Τέλος το πεδίο “RequestState” θα το μετατρέψει σε 1, το οποίο σημαίνει πως το αίτημα έχει ανέβει στη βάση και είναι σε κατάσταση αναμονής.

Για τους admin έχει δημιουργηθεί συνάρτηση «get_all_restaurant_reservations» στο path που τελειώνει σε «/api/hotel/restaurants» για τα αιτήματα κράτησης τραπεζιού στο εστιατόριο και «get_all_cleaning_service_reservations» στο path που τελειώνει σε «/api/hotel/cleanings» για τα αιτήματα καθαρισμού δωματίου. Και στα δύο τελευταία παραδείγματα, αν το token είναι έγκυρο και ανήκει σε admin το ενδιάμεσο λογισμικό θα επιστρέψει JSON όπου θα εμπεριέχει τις εκάστοτε λίστες με όλα τα αιτήματα που έχουν δημιουργηθεί, ασχέτως της κατάστασης τους ή τον δημιουργό τους.

Στο ενδιάμεσο λογισμικό έχει εξίσου υλοποιηθεί και η συνάρτηση που διαχειρίζεται τη Βάση Δεδομένων, σχετικά με την απάντηση της διαχείρισης του ξενοδοχείου στα παραπάνω αιτήματα. Όπως έχει προαναφερθεί, ο admin μπορεί να μεταβάλλει στο αίτημα μόνο τα πεδία `RequestState` και `AdminMessage`. Γι’ αυτό το HTTP Request πρέπει να είναι τύπου POST και να περιλαμβάνει μήνυμα JSON με αυτά τα δύο πεδία, όπως επίσης και `authorization` με `admin token`. Όπως και στην συνάρτηση που απευθύνεται στα αιτήματα για κλείσιμο τραπεζιού στο εστιατόριο, έτσι και στην συνάρτηση που

απευθύνεται στα αιτήματα για καθαρισμό δωματίου, χρειάζεται να χρησιμοποιηθεί από τον διαχειριστή το πρωτεύον κλειδί (id) του κάθε αιτήματος, καθώς η διαδρομή τελειώνει σε: /api/hotel/restaurant ή cleaning και /αριθμός id. Έχουν δημιουργηθεί και οι συναρτήσεις για τις διαγραφές των αιτημάτων καθαρισμού και κλείσιμο τραπεζιού στο εστιατόριο, όπως και ενημέρωσης των αιτημάτων από την διαχείριση του ξενοδοχείου. Οι διαγραφές γίνονται μόνο από χρήστη με ρόλο admin, όπου ο έλεγχος γίνεται στις συναρτήσεις αυτές και τα paths είναι ίδια με τον τρόπο που γίνεται η μεμονωμένη ενημέρωση των πεδίων από τους διαχειριστές.

Για το table Ratings, δηλαδή για τις αξιολογήσεις των πελατών προς το ξενοδοχείο, έχουν υλοποιηθεί οι κατάλληλες συναρτήσεις. Για να κάνει στείλει ένας πελάτης μια κριτική, έχει υλοποιηθεί συνάρτηση που ελέγχει την εγκυρότητα του token του πελάτη, όπου με ένα HTTP Request στη διαδρομή που τελειώνει σε «/api/hotel/rating» και με περιεχόμενο τα πεδία του table σε μορφή JSON. Τότε και το ενδιαμέσο λογισμικό θα αποθηκεύσει στη βάση μια γραμμή στο table Ratings όπου θα εμπεριέχει την κριτική και τον πελάτη. Αντίστοιχα, ένας πελάτης μπορεί να κάνει GET σε αυτή την διεύθυνση, ώστε να του έρθουν οι πληροφορίες του τι κριτική έχει κάνει. Αντίστοιχη συνάρτηση έχει δημιουργηθεί και για την διαχείριση του ξενοδοχείου, όπου σε αυτή την περίπτωση το ενδιαμέσο λογισμικό απαντάει με όλες τις κριτικές που έχουν γίνει.

Τέλος, για την ολοκληρωμένη λειτουργικότητα του API, περιλαμβάνονται βιβλιοθήκες. Για την δημιουργία και διαχείριση της βάσης δεδομένων, η βιβλιοθήκη sqlite3. Για την δημιουργία, κωδικοποίηση και ανάλυση των tokens η βιβλιοθήκη «jwt» και το εργαλείο «wraps» που υπάγεται στην βιβλιοθήκη “functools”. Για την κρυπτογράφηση του πεδίου password κάθε χρήστη το εργαλείο Brycrypt του Flask. Εξίσου για την λειτουργικότητα ανάγνωσης και συγγραφής JSON αλλά και συνολικότερα των HTTP Requests, χρησιμοποιούνται τα εργαλεία του framework Flask. Για την συγγραφή του κώδικα χρησιμοποιήθηκε το IDE Visual Studio Code και για το version control του το Git.

3.2 Διεπαφή Κινητής Συσκευής

Η διεπαφή κινητής συσκευής αναπτύχθηκε όπως προαναφέρεται στο δεύτερο κεφάλαιο, με την τεχνολογία Xamarin. Το IDE το οποίο χρησιμοποιήθηκε είναι το Visual Studio 2022 Community. Έχει δοκιμαστεί σε διάφορες συσκευές Android. Λειτουργεί με κατώτερη δυνατή έκδοση AndroidSDK την 22 και στοχεύει στην έκδοση 31 που δημοσιεύτηκε τον Αύγουστο του 2021. Η εφαρμογή όμως, λόγω του framework της, έχει φτιαχτεί ώστε να έχει κοινό κώδικα και να λειτουργεί και σε iPhone. Δυστυχώς η εφαρμογή δεν έχει δοκιμαστεί ποτέ σε iPhone, εξαιτίας της πολιτικής της ίδιας εταιρείας της Apple, που χρειάζεται πληρωμή για να είναι εφικτή η σχεδίαση και η αποσφαλμάτωση σε περιβάλλον iOS και iPhone. Το framework του κώδικα της C# είναι

το .NET Standard 2.0 και τα controls που εμφανίζονται στους χρήστες είναι όλα του Xamarin. Η εφαρμογή αυτή είναι το εργαλείο των πελατών του ξενοδοχείου στο σύστημα της παρούσας πτυχιακής.

Για την επικοινωνία με το API, έχει δημιουργηθεί αρχείο HotelAPIService.cs. Το αρχείο αυτό χρησιμοποιεί την βιβλιοθήκη Newtonsoft.Json για την διαχείριση των json περιεχομένων στα HTTP Requests. Σε αυτό το αρχείο είναι η διεύθυνση του API και όλες οι συναρτήσεις που αφορούν τα HTTP Requests τα οποία χρειάζεται να υπάρχουν στην εφαρμογή. Οι συναρτήσεις τρέχουν ασύγχρονα με το κύριο thread της εφαρμογής, για την καλύτερη εμπειρία του χρήστη. Για όλα τα JSON τα οποία αποστέλλονται είτε λαμβάνονται έχουν δημιουργηθεί και οι κατάλληλες κλάσεις, που περιέχουν σαν πεδία τους τα πεδία των JSON.

Έχει σχεδιαστεί η διεπαφή με τον χρήστη, με διάφορες φόρμες οθόνης, για την πλήρη λειτουργικότητα της εφαρμογής. Το Xamarin, για την σχεδίαση των φορμών, χρησιμοποιεί τη γλώσσα XAML. Η πρώτη φόρμα οθόνης που εμφανίζεται σε χρήστη που μόλις εγκατέστησε την εφαρμογή, αφορά την σύνδεση του πελάτη. Με τα κατάλληλα στοιχεία (username, password), καλείται η συνάρτηση login που έχει γραφτεί στο αρχείο HotelAPIService, η οποία πραγματοποιεί HTTP Request τύπου POST στο API, όπου αν τα στοιχεία είναι σωστά επιστρέφεται στην εφαρμογή ένα Token. Αυτό το Token αποθηκεύεται στις ιδιότητες της εφαρμογής, έτσι ώστε να μην χρειαστεί ο πελάτης να εισάγει ξανά τα στοιχεία την επόμενη φορά που θα χρησιμοποιήσει την εφαρμογή από το κινητό του. Χρησιμοποιώντας την βιβλιοθήκη της C# System.Net.Http, δημιουργείται ένα αντικείμενο HttpClient, το οποίο είναι δυνατό να πραγματοποιήσει HTTP Requests, αιτήματα δηλαδή στο ενδιάμεσο λογισμικό, για την κατάλληλη ανταλλαγή πληροφοριών, τα οποία μπορούν να παραμετροποιηθούν όπως χρειάζεται από τους προγραμματιστές.

Η φόρμα που εμφανίζεται αφότου έχει γίνει η ορθή σύνδεση του πελάτη, είναι η κεντρική φόρμα της εφαρμογής. Παρουσιάζει τις δυνατότητες της εφαρμογής σε ένα μενού, όπου ο χρήστης/πελάτης διαλέγει μεταξύ επιλογών. Από την κεντρική φόρμα υπάρχουν οι επιλογές της Ρεσεψιόν (Reception), της Κράτησης Τραπεζιού στο Εστιατόριο (Restaurant Reservation), της Υπηρεσίας Καθαρισμού (Cleaning Service), της Σχετικά με το Ξενοδοχείο (About our Hotel) και η επιλογή της αξιολόγησης (Rate us).

Στην φόρμα οθόνης Ρεσεψιόν, ο πελάτης μπορεί να διακρίνει κάποιες γενικές πληροφορίες του Ξενοδοχείου. Από τον τίτλο και την διεύθυνσή του, μέχρι το email του και ένα κείμενο σχετικά με το ξενοδοχείο. Επίσης υπάρχει η δυνατότητα κλήσης της Ρεσεψιόν. Η λειτουργικότητα αυτή έχει πραγματοποιηθεί μέσα από το HTTP Request τύπου GET «get_hotel_data», το οποίο πραγματοποιείται από το αρχείο HotelAPIService, επιτόπου με την είσοδο του χρήστη σε αυτή τη φόρμα.

Εξίσου έχει αναπτυχθεί η φόρμα οθόνης που αφορά την κράτηση τραπεζιού στο εστιατόριο. Με την είσοδο του χρήστη πραγματοποιούνται οι κατάλληλες συναρτήσεις για την λήψη των αιτημάτων κρατήσεως τραπεζιού του συνδεδεμένου πελάτη, όπως και για την λήψη του URL του μενού του ξενοδοχείου. Έτσι, στο κινητό, παρουσιάζεται πάνω πάνω η δυνατότητα μετάβασης στο μενού και παρακάτω η δημιουργία νέου αιτήματος. Έπειτα φαίνονται τα αιτήματα που έχουν ήδη γίνει, μαζί με όλα τους τα πεδία, δηλαδή την κατάσταση του αιτήματος, το σχόλιο (αν υπάρχει) της διαχείρισης του ξενοδοχείου, την επιθυμούμενη ημερομηνία, τον αριθμό ατόμων και το σχόλιο του πελάτη.

Στη περίπτωση όπου ο χρήστης/πελάτης επιλέξει να δημιουργήσει νέο αίτημα κράτησης τραπεζιού στο εστιατόριο, έχει δημιουργηθεί φόρμα οθόνης κατάλληλη για την δημιουργία ενός τέτοιου αιτήματος. Με τα controls του Xamarin, έχει ένα πεδίο κειμένου στο πάνω μέρος της οθόνης όπου εξηγεί στον χρήστη/πελάτη πώς να κάνει ένα αίτημα. Έπειτα ακολουθεί η επιλογή αριθμού πελατών, η επιλογή ημερομηνίας και ώρας. Η επιλογή ώρας δεν αφορά λεπτά, ώστε να γίνεται ευκολότερη η ανάλυση από την διοίκηση του πότε θέλει ένας πελάτης να κλείσει τραπέζι. Έπειτα ακολουθεί ένα control όπου ο χρήστης μπορεί να εισαγάγει το κείμενο του για το σχόλιο αυτού του αιτήματος. Τέλος, στο κατώτερο σημείο της φόρμας, υπάρχει η επιλογή Submit, όπου αν ο χρήστης την πατήσει θα κληθεί η συνάρτηση για την αποστολή του αιτήματος στο ενδιαμέσο λογισμικό, όπως και παραπάνω, με τα στοιχεία να έχουν γίνει JSON.

Αντίστοιχα με παραπάνω λειτουργεί και η υλοποίηση της οθόνης της Υπηρεσίας Καθαρισμού. Με την είσοδο του χρήστη, καλούνται οι συναρτήσεις για την λήψη των αναγκαίων δεδομένων, όπως σε αυτή τη περίπτωση του «DailyStandardHousekeeping» του χρήστη, όπου αφορά το αν έχει ενεργοποιημένη την καθημερινή καθαριότητα του δωματίου του και τα αιτήματα τα οποία έχει κάνει για εξειδικευμένη ημερομηνία καθαριότητας. Αντίστοιχα παρουσιάζονται και αυτά, ενώ υπάρχουν οι επιλογές για την δημιουργία νέου αιτήματος, όπου ένα “DatePicker” και ένα “TimePicker” του Xamarin εμφανίζονται αν ο χρήστης το επιλέξει και έτσι δημιουργείται ένα αίτημα. Για την αποστολή των αιτημάτων στο API, η επιλογή submit που βρίσκεται στο τέλος της φόρμας.

Ακολουθεί η φόρμα «Σχετικά με το Ξενοδοχείο». Η παρούσα φόρμα μπορεί να χρησιμοποιηθεί ποικιλοτρόπως, στην περίπτωσή μας επιλέχτηκε η εμφάνιση ενός html περιεχομένου, που εμπεριέχει πρόσθετες πληροφορίες για το ξενοδοχείο, οι οποίες λαμβάνονται από την εφαρμογή με την είσοδο του χρήστη στη φόρμα. Απλή επίσης είναι και η φόρμα αξιολόγησης (Rate Us), στην οποία με το τράβηγμα του δακτύλου του ο πελάτης μπορεί να αξιολογήσει από 0 έως πέντε αστέρια και τις πέντε επιλογές που έχει και για να τις δημοσιεύσει πατάει την επιλογή submit, όπου η φόρμα αρχικά δημιουργεί ένα αντικείμενο της κλάσης Rates, της οποίας τα πεδία της είναι αριθμητικά και πρόκειται για τις πέντε κατηγορίες αξιολόγησης. Καλεί την συνάρτηση του αρχείου

HotelAPIService, PostRatingAsync, η οποία σαν input έχει ένα αντικείμενο της κλάσης Rates, που στη συγκεκριμένη περίπτωση θα είναι το αντικείμενο με τα πεδία της αξιολόγησης που δημιούργησε ο χρήστης/πελάτης.

Για την ευκολότερη περιήγηση του χρήστη στην εφαρμογή, στην πάνω αριστερά μεριά της εφαρμογής υπάρχει η επιλογή του κάθετου μενού. Το παρόν μενού δημιουργήθηκε με τα εργαλεία του Xamarin και έχει τις επιλογές της Κεντρικής φόρμας και όλων όσων προαναφέρθηκαν οι οποίες υπάρχουν στη κεντρική φόρμα. Διαθέτει επίσης, την επιλογή «Logout», για την αποσύνδεση του χρήστη. Με την επιλογή αυτή, αυτόματα διαγράφεται από τις ιδιότητες της εφαρμογής το πεδίο «Token». Με την έναρξη της εφαρμογής γίνεται ο έλεγχος του αν υπάρχει στις ιδιότητες συμπληρωμένο πεδίο με τον τίτλο «Token». Εάν δεν υπάρχει, η αρχική οθόνη θα είναι Login.

Τέλος, στην παρούσα διεπαφή έχουν χρησιμοποιηθεί εικόνες που σχεδιάστηκαν ή βρέθηκαν στο διαδίκτυο. Για το version control της εφαρμογής χρησιμοποιήθηκε το λογισμικό Git, όπου είναι και ανεβασμένο το ιστορικό ανάπτυξης της εφαρμογής στο GitHub. Έγινε εφικτή η διανομή εκτελέσιμων αρχείων για Android μέσω του Visual Studio.

3.3 Διεπαφή Ιστού

Για την διεπαφή της διαχείρισης του ξενοδοχείου, αναπτύχθηκαν σελίδες html. Χρησιμοποιήθηκαν, όπως προαναφέρεται στο δεύτερο κεφάλαιο, οι γλώσσες HTML, CSS και Javascript. Πρόκειται στη πραγματικότητα, για αρχεία τα οποία διαβάζονται στους φυλλομετρητές (browsers). Είναι εφικτό να φιλοξενοούνται σε ένα τοπικό δίκτυο ή σε μια παγκόσμια διεύθυνση, όπως επίσης και να ανοίξουν απλώς από έναν μία συσκευή. Έχουν δημιουργηθεί ώστε να είναι εύχρηστη η διαχείριση του πληροφοριακού μας συστήματος από την διαχείριση του ξενοδοχείου, σε μια εύκολη και ελκυστική διεπαφή.

Όλες οι σελίδες εκτός της login, έχουν αναπτυχθεί ώστε να εμπεριέχουν ένα menu στην αριστερή πλευρά τους για την περιήγηση του προσωπικού στις σελίδες. Η κάθε σελίδα έχει ονομασία που να γίνεται αντιληπτό το με τι έχει να κάνει. Σε όλες τις σελίδες έχουν χρησιμοποιηθεί τα βασικά εργαλεία των γλωσσών για ανάπτυξη ιστοσελίδων. Για την επικοινωνία με το ενδιαμέσο λογισμικό, είναι ενσωματωμένη στις σελίδες η διεύθυνση IP όπου φιλοξενείται αυτό.

Για την ορθή χρήση τους, η πρώτη σελίδα που πρέπει να ανοίξει είναι η σελίδα login.html. Η σελίδα αυτή, αφότου ο χρήστης εισαγάγει τις πληροφορίες του, δηλαδή ποιο το username του και ποιο το password του, στέλνει HTTP Request τύπου POST μέσω javascript στο api, στην διεύθυνση /login, με περιεχόμενο τύπου JSON το username και το password. Αφότου το API απαντήσει, το τόκεν αποθηκεύεται στον

φυλλομετρητή με την μορφή cookie, όπου και θα αξιοποιηθεί για την δημιουργία HTTP Requests.

Η σελίδα υπό τον τίτλο Reception είναι η πρώτη που φορτώνει αυτόματα μετά την είσοδο του διαχειριστή. Με το που φορτώνει, κάνει κλήση στο api τύπου GET στη σελίδα /api/hotel που καλείται η συνάρτηση του API get_hotel_data, όπου μέσα από αυτήν το API στέλνει στη σελίδα σε μορφή JSON τα πεδία GeneralHotelData. Η σελίδα αυτόματα τα εμφανίζει σε πεδία που είναι εφικτή η επεξεργασία τους. Ο διαχειριστής από εκεί μπορεί να ενημερώσει τα πεδία και να τα «ανεβάσει» στο API, όπου σε εκείνη τη περίπτωση η σελίδα θα αποστείλει HTTP Request τύπου post στη διεύθυνση /api/hotel/ με περιεχόμενο json όλα τα πεδία του generalhoteldata και με το token το οποίο είναι αποθηκευμένο σε μορφή cookie.

Σε κάθε σελίδα εκτός της σελίδας Login, στην αριστερή πλευρά, υπάρχει το μενού περιήγησης στην εφαρμογή Ιστού. Περιλαμβάνει τις επιλογές της σελίδας Reception, Restaurant, Cleaning, Rating, About και Logout. Με αυτό το μενού εξασφαλίζεται η περιήγηση του χρήστη-διαχειριστή στις σελίδες της εφαρμογής Ιστού.

Η σελίδα Restaurant, με το που ανοίξει, κάνει HTTP Request τύπου GET. Το αίτημα αυτό πραγματοποιείται στη διεύθυνση που τελειώνει σε /api/hotel/restaurants, όπου έχει υλοποιηθεί η συνάρτηση στο ενδιάμεσο λογισμικό που επιστρέφει όλα τα αιτήματα κλεισίματος τραπέζιού που έχουν δημιουργηθεί. Έτσι, παίρνει το JSON, που έχει δημιουργηθεί η συνάρτηση στη γλώσσα Javascript με τίτλο displayRequests και input τα data που έλαβε από το JSON. Κάνοντάς τα ταξινόμηση από όποιο έχει το αρχαιότερο ReservDateTime, εμφανίζει όλα τα πεδία σε μορφή κάρτας μέσω της βοήθειας της CSS.

Στα πεδία τα οποία χρειάζεται να αξιοποιηθούν από τον χρήστη-διαχειριστή, έχει επιπρόσθετα controls για την επεξεργασία τους. Υπάρχουν επίσης τα κουμπιά σε κάθε κάρτα, για Submit και Delete. Στη περίπτωση του Submit, δημιουργείται JSON μέσω της Javascript με τα αναγκαία πεδία, όπου και στέλνεται με POST στη διεύθυνση που τελειώνει σε /api/hotel/restaurant/request.id, όπου request id η αριθμητική τιμή του πεδίου RequestId του κάθε αιτήματος. Έτσι γίνεται εφικτή η μεμονωμένη ενημέρωση του πεδίου που πατήθηκε το κουμπί Submit. Στην περίπτωση του κουμπιού Delete, στέλνεται HTTP Request τύπου Delete στην ίδια διεύθυνση, όπου το API έχει φροντίσει να διαγράψει το row με το συγκεκριμένο πρωτεύον κλειδί.

Αντίστοιχα λειτουργεί και η σελίδα Cleaning. Έχει την ίδια λειτουργικότητα, με την διαφορά στις διευθύνσεις σε /api/hotel/cleanings αντί για restaurants και στα πεδία των JSON αρχείων που λαμβάνονται και αποστέλλονται, όπως και στα πεδία που εμφανίζονται στις κάρτες, αντίστοιχα με το τι περιλαμβάνει κάθε table.

Η σελίδα About έχει δημιουργηθεί χωρίς κάποια ιδιαίτερη λειτουργικότητα. Μπορεί να αξιοποιηθεί στο μέλλον ποικιλότροπος. Τώρα απλά περιέχει ένα κείμενο στο πεδίο της HTML, σχετικά με το σύστημα του έξυπνου ξενοδοχείου και την χρήση της εφαρμογής Ιστού.

Η σελίδα Rating φορτώνει αυτόματα όλες τις καταχωρήσεις των Rating ανά κάθε δωμάτιο. Αυτή η ενέργεια γίνεται μέσω HTTP Request τύπου «GET» στο API, όπου επιστρέφονται οι αξιολογήσεις που έχουν γίνει από όλους τους χρήστες στα δωμάτια. Τέλος, μέσω της Javascript κάνει υπολογισμούς για τους μέσους όρους της κάθε αξιολόγησης όλων των δωματίων. Έχει αναπτυχθεί για να δείχνει στο προσωπικό του ξενοδοχείου τις καταχωρήσεις κριτικών των χρηστών-πελατών.

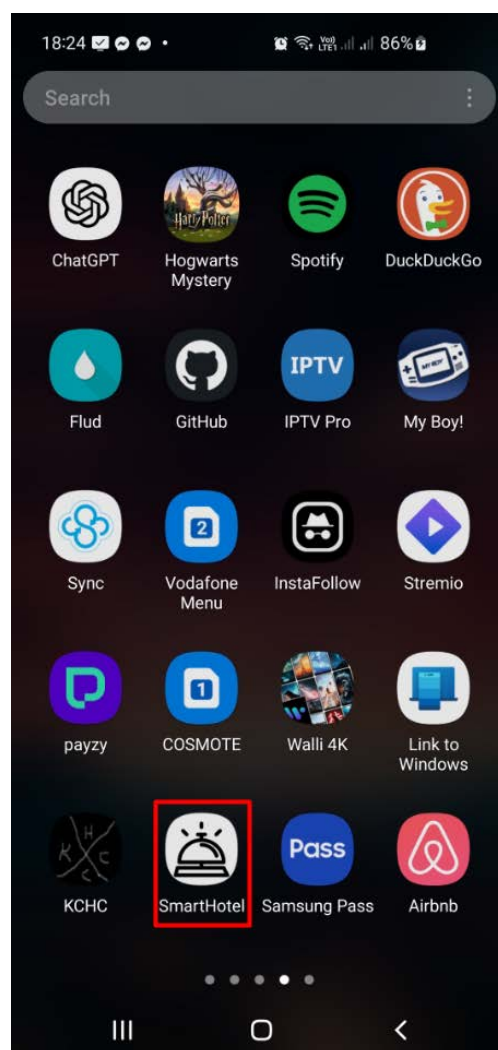
Υφίσταται επίσης η επιλογή Logout στο μενού αριστερά. Με την επιλογή αυτή, από το πεδίο <script> που εμπεριέχει Javascript, διαγράφεται το cookie στον φυλλομετρητή το οποίο εμπεριέχει το token, αφότου γίνει σε μορφή alert επιβεβαίωση από τον χρήστη-διαχειριστή. Με την εκτέλεση της παραπάνω λειτουργίας, μεταφέρει αυτόματα τον φυλλομετρητή στη σελίδα login.

Ολοκληρώνοντας, στη διεπαφή ιστού έχουν χρησιμοποιηθεί εικόνες και φωτογραφίες που είναι δημόσιες στο διαδίκτυο. Για το version control των σελίδων χρησιμοποιήθηκε το λογισμικό Git, όπου είναι και ανεβασμένο το ιστορικό ανάπτυξης της εφαρμογής στο GitHub. Δοκιμάστηκαν αρκετοί φυλλομετρητές (browsers) για την ορθή λειτουργία τους, οι Brave, Google Chrome, Mozilla Firefox, Opera, Safari και Microsoft Edge.

4. Εγχειρίδια Χρήστη

4.1 Εγχειρίδιο Κινητής Συσκευής

Η εφαρμογή κινητής συσκευής έχει αναπτυχθεί για τους πελάτες του ξενοδοχείου. Είναι η κατάλληλη εφαρμογή για εύκολη και γρήγορη πρόσβαση στα δεδομένα που χρειάζεται ένας πελάτης σχετικά με το ξενοδοχείο. Επίσης, μέσω της εφαρμογής, μπορεί ένας πελάτης να διαχειριστεί την υπηρεσία καθαριότητας, να κλείσει τραπέζι στο εστιατόριο και να κάνει αξιολόγηση του ξενοδοχείου. Έπειτα από την εγκατάσταση σε μια συσκευή που φέρει το λειτουργικό σύστημα Android, είναι εφικτή η λειτουργία της. Χρειάζεται σύνδεση στο διαδίκτυο ή στο τοπικό δίκτυο που έχει ανέβει η βάση δεδομένων. Με καταχωρημένα στη βάση τα στοιχεία χρήστη-πελάτη, μπορεί η εφαρμογή να τρέξει όπως χρειάζεται.

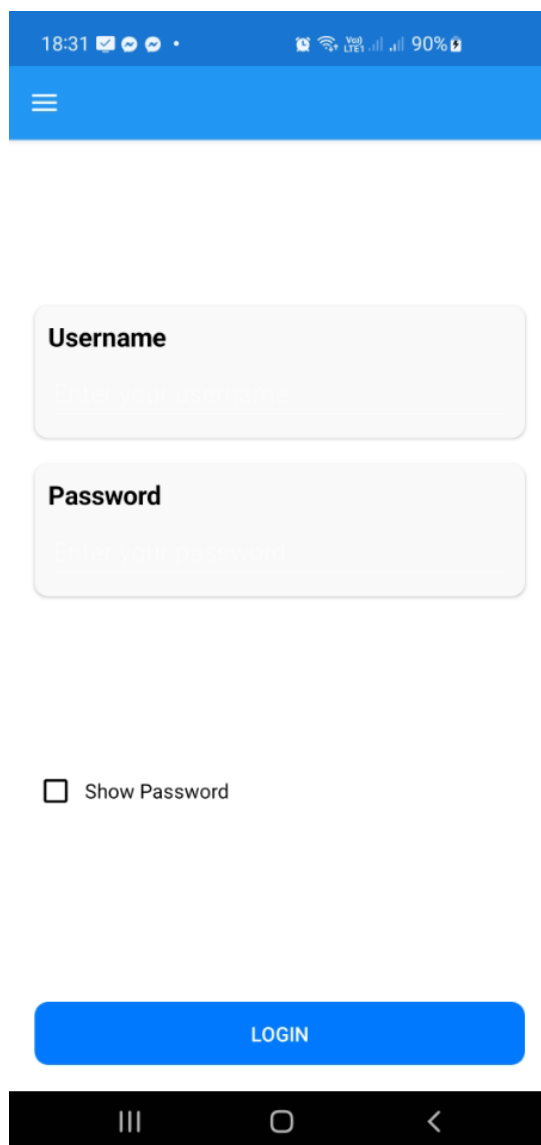


Εικόνα 4.1 Η εγκαταστημένη εφαρμογή SmartHotel σε Android

Στην παραπάνω εικόνα φαίνεται η εφαρμογή αφότου έχει εγκατασταθεί σε μια κινητή συσκευή Android. Στο κόκκινο πλαίσιο φαίνεται το εικονίδιο της εφαρμογής SmartHotel και ο τίτλος της.

4.1.1 Login

Όταν η εφαρμογή ανοίξει για πρώτη φορά και η συσκευή βρίσκεται στο δίκτυο όπου κάνει host τη βάση δεδομένων, παρουσιάζεται η οθόνη Login στον χρήστη-πελάτη.

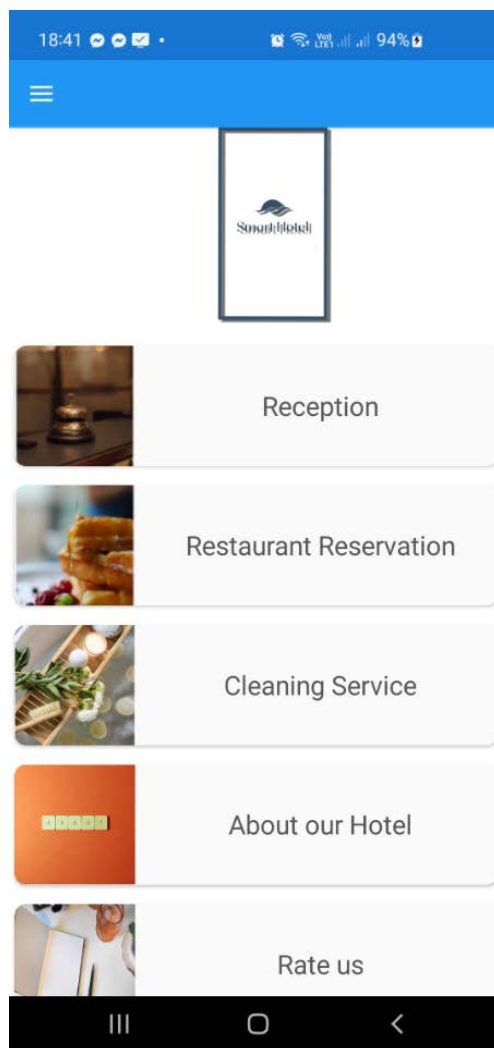


Εικόνα 4.1.1 Η οθόνη Login

Στην παραπάνω οθόνη χρειάζεται να συμπληρωθεί το Username του χρήστη-πελάτη καθώς και ο κωδικός του. Για λόγους ασφαλείας, ο κωδικός που εισάγεται δεν φαίνεται. Για να γίνει φανερός, χρειάζεται να γίνει άγγιγμα στο κουτάκι κάτω αριστερά του «Show Password».

Αφότου ο χρήστης-πελάτης εισαγάγει ορθώς τα στοιχεία σύνδεσής του, μπορεί να πατήσει στο κουμπί «Login» στην κάτω πλευρά της οθόνης. Αν τα στοιχεία είναι σωστά, τότε θα εμφανιστεί η κεντρική οθόνη της εφαρμογής και θα ληφθεί Token (τεκμήριο) σύνδεσης στην εφαρμογή, το οποίο και θα αποθηκευτεί στη συσκευή. Έτσι, δεν θα χρειαστεί ξανά εισαγωγή στοιχείων για την είσοδο στις λειτουργίες της εφαρμογής. Αν τα στοιχεία είναι λάθος ή η σύνδεση στη βάση είναι αδύνατη, τότε εμφανίζεται μήνυμα λάθους.

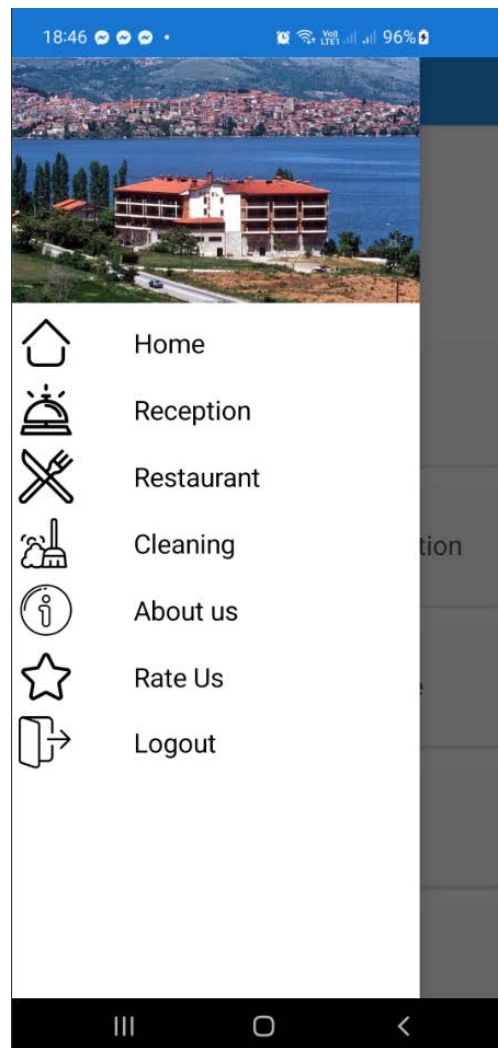
4.1.2 Περιήγηση στην εφαρμογή



Εικόνα 4.1.2.1 Η κεντρική οθόνη

Η κεντρική οθόνη της εφαρμογής εμφανίζει ένα κατακόρυφο μενού περιήγησης. Στο κατακόρυφο αυτό μενού, παρατηρούνται κάρτες οι οποίες κάνουν εφικτή την είσοδο σε διάφορες οθόνες της εφαρμογής. Οι κάρτες που παρατηρούνται, όπως φαίνεται και από την παραπάνω εικόνα, είναι οι: «Reception», «Restaurant Reservation», «Cleaning Service», «About our Hotel» και «Rate us».

Όπως και κάθε άλλη οθόνη εκτός της Login, στο πάνω μέρος υπάρχει μια μπλε γραμμή, η οποία αριστερά φέρει ένα εικονίδιο τύπου «σάντουιτς». Αν πατηθεί αυτό το εικονίδιο, εμφανίζεται μια κυλιόμενη λίστα, η οποία χρησιμοποιείται για την εύκολη και γρήγορη περιήγηση στην εφαρμογή.



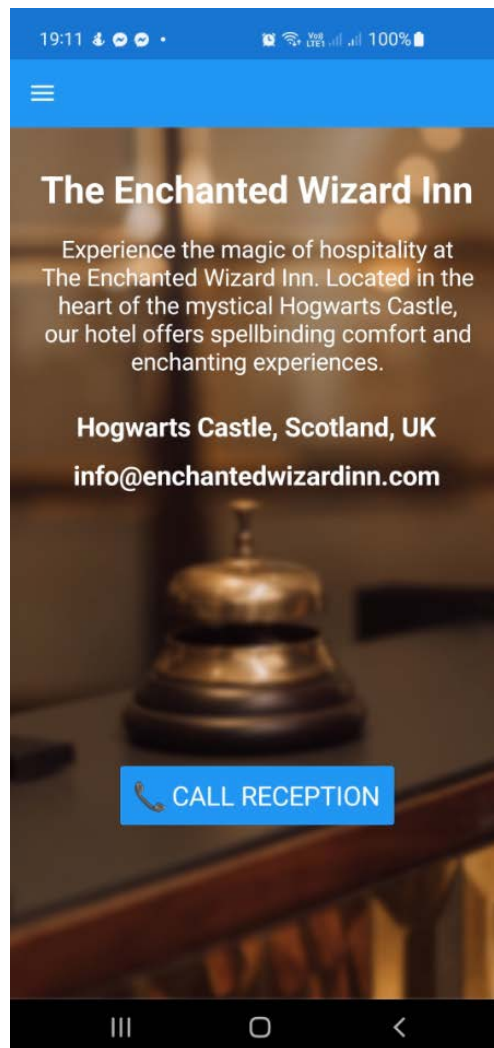
Εικόνα 4.1.2.2 Η κυλιόμενη λίστα περιήγησης

Στην κυλιόμενη λίστα παρατηρούνται διάφορες οθόνες που μπορούν να χρησιμοποιηθούν για λόγους περιήγησης. Με κλικ στην επιλογή κάθε επιλογής, η εφαρμογή μεταφέρεται σε διαφορετική οθόνη. Υπάρχει επίσης η επιλογή Logout, με την οποία γίνεται η διαγραφή του αποθηκευμένου Token σύνδεσης και εμφανίζεται η οθόνη Login στην οποία ο χρήστης-πελάτης καλείται να εισαγάγει εκ νέου τα στοιχεία του για να προχωρήσει στις λειτουργίες της εφαρμογής.

Σε κάθε οθόνη μπορούν να αξιοποιηθούν τα πλήκτρα του λογισμικού Android, στην κάτω πλευρά του κινητού. Με το πλήκτρο «<» κάτω δεξιά όπως φαίνεται και στις εικόνες, μπορεί ο χρήστης-πελάτης να γυρίσει στην προηγούμενη οθόνη που βρισκόταν μέσα στην εφαρμογή.

4.1.3 Η οθόνη Reception

Με την είσοδο στην οθόνη Reception η εφαρμογή «κατεβάζει» από τη Βάση Δεδομένων, μέσω του Ενδιάμεσου Λογισμικού, στοιχεία, τα οποία παρουσιάζονται στην οθόνη.



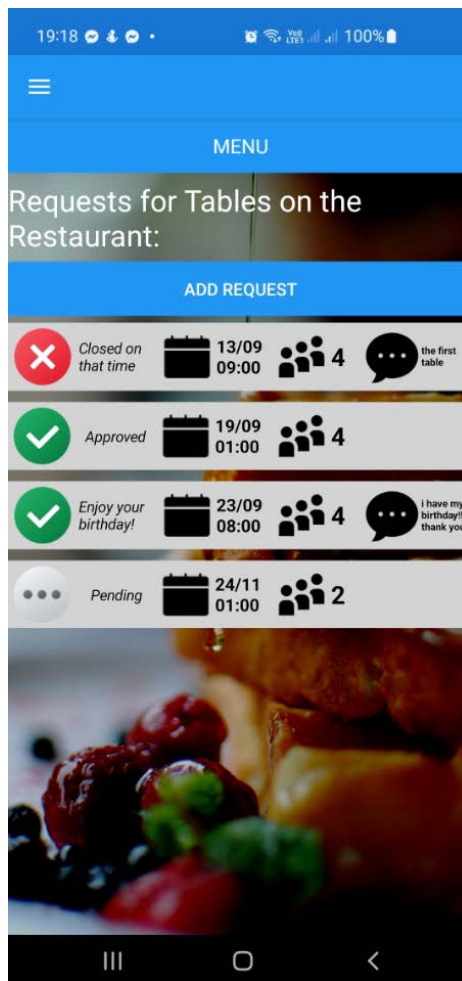
Εικόνα 4.1.3 Η οθόνη Reception

Από την παρούσα οθόνη, ο χρήστης-πελάτης παρατηρεί το όνομα του ξενοδοχείου και γενικές πληροφορίες του. Μπορεί να δει τη διεύθυνσή του και το Email του. Τέλος, είναι εφικτή η κλήση στην ρεσεψιόν του ξενοδοχείου από το κουμπί «Call Reception».

Η οθόνη Reception ενημερώνεται αυτόματα κάθε φορά που αλλάζει κάποια πληροφορία σχετικά με το ξενοδοχείο στη βάση δεδομένων.

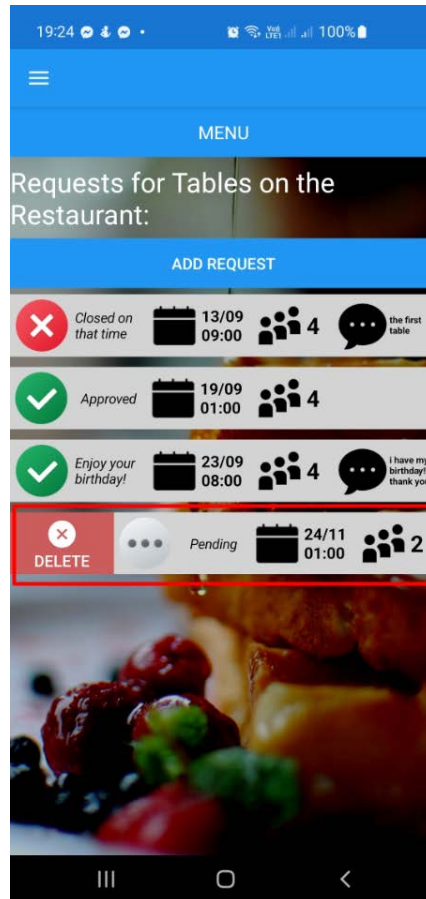
4.1.4 Restaurant Reservation

Στην οθόνη Restaurant παρατηρείται στο άνω μέρος της κουμπί «Menu». Το κουμπί αυτό οδηγεί τον χρήστη-πελάτη στη διεύθυνση όπου είναι ανεβασμένο το Menu του Εστιατορίου. Παρακάτω παρατηρείται το κουμπί «Add Request», όπου οδηγεί τον χρήστη-πελάτη στη οθόνη δημιουργίας νέου αιτήματος για κράτηση τραπεζιού στο ξενοδοχείο.



Εικόνα 4.1.4.1 Η οθόνη Restaurant

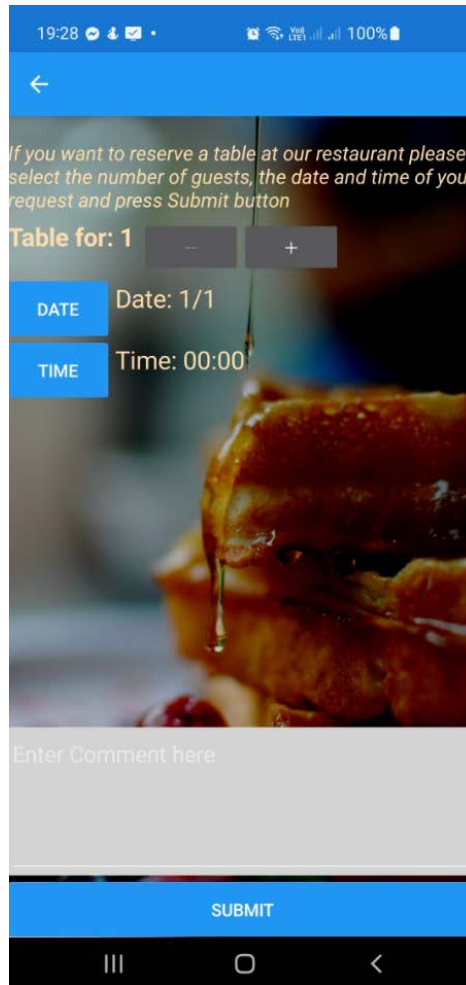
Τέλος, υφίσταται μια λίστα με τα αιτήματα τα οποία έχουν ήδη δημιουργηθεί. Σε κάθε γραμμή που παρουσιάζει αίτημα, από τα αριστερά προς τα δεξιά, γίνεται εμφανής αρχικά η κατάσταση του, δηλαδή αν έχει εγκριθεί ή απορριφθεί ή αν βρίσκεται σε αναμονή, με τα εικονίδια τύπου «τικ», «X» και «...». Έπειτα, το μήνυμα του διαχειριστή σχετικά με το αίτημα. Στη συνέχεια παρατηρείται η ημερομηνία και η ώρα για την κράτηση τραπεζιού, ο αριθμός των συνδαιτημόνων και το σχόλιο του πελάτη σχετικά με το αίτημα. Με την κύλιση της «κάρτας» αιτήματος προς τα δεξιά γίνεται εφικτή η διαγραφή του.



Εικόνα 4.1.4.2 Κύλιση αιτήματος

Στην παραπάνω εικόνα εμφανίζεται ένα αίτημα να έχει κυλίσει προς τα δεξιά. Εμφανίζεται το κουμπί «Delete», το οποίο επιτρέπει στον χρήστη πελάτη να διαγράψει το σχετικό αίτημα.

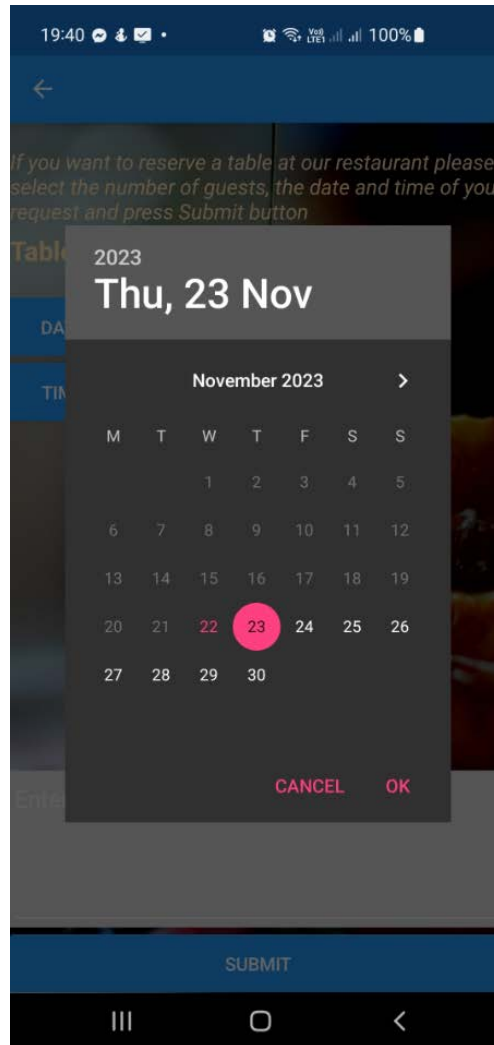
Για την δημιουργία νέου αιτήματος γίνεται η χρήση του κουμπιού «Add Request», όπου και ανοίγει η οθόνη δημιουργίας αιτήματος κράτησης τραπεζιού.



Εικόνα 4.1.4.3 Η οθόνη δημιουργίας νέου αιτήματος Restaurant

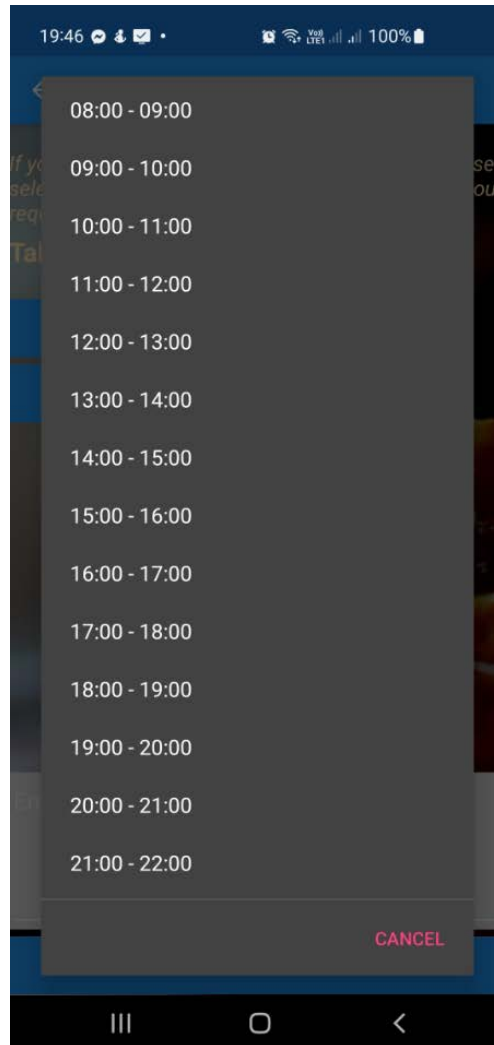
Στην παρούσα οθόνη ο χρήστης-πελάτης μπορεί να ορίσει τις παραμέτρους του αιτήματος. Για την επιλογή του αριθμού των ατόμων, γίνεται χρήση των πλήκτρων «+» και «-». Ανώτατο επιτρεπτό όριο είναι τα 8 άτομα.

Για την επιλογή ημερομηνίας, ο χρήστης-πελάτης αγγίζει το κουμπί «Date».



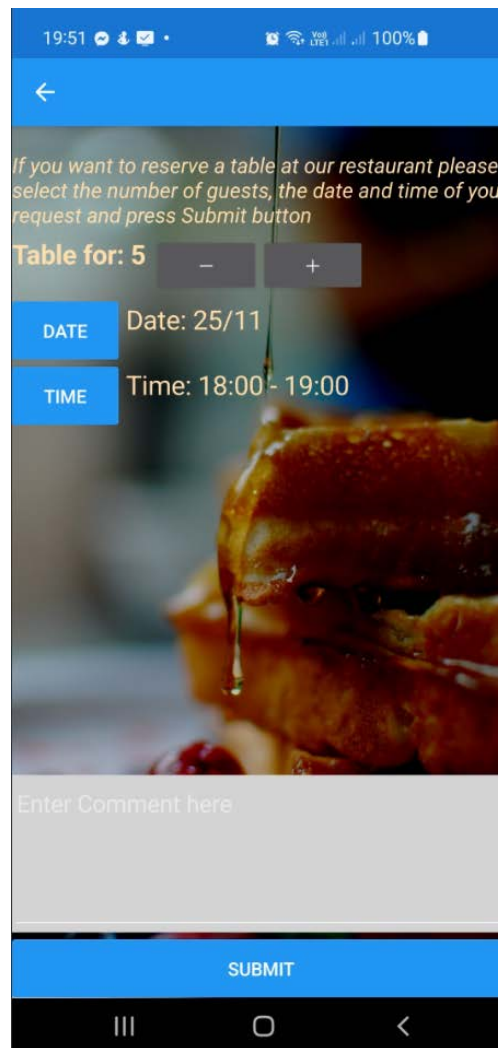
Εικόνα 4.1.4.4 Επιλογή ημερομηνίας για κράτηση τραπεζιού

Στο αναδυόμενο πλαίσιο ημερολογίου, είναι εφικτή η επιλογή μήνα και ημέρας μήνα για κράτηση τραπεζιού στο ξενοδοχείο. Για την ακύρωση της διαδικασίας και το κλείσιμο του αναδυόμενου πλαισίου, μπορεί ο χρήστης-πελάτης να πατήσει «Cancel». Για την επιλογή μήνα και ημέρας, γίνεται απλά πάτημα είτε στον μήνα και τέλος στην ημέρα μήνα, όπου εμφανίζεται κυκλωμένη στο πλαίσιο ημερολογίου και τέλος γίνεται πάτημα στο «OK» στο κάτω δεξιά μέρος του πλαισίου.



Εικόνα 4.1.4.5 Επιλογή ώρας για κλείσιμο τραπεζιού

Για την επιλογή της ώρας γίνεται άγγιγμα στο κουμπί «Time», όπου εμφανίζεται το παραπάνω αναδυόμενο πλαίσιο στην οθόνη. Το πλαίσιο αυτό περιέχει επιλογές ανά ώρα για κράτηση τραπεζιού και είναι κυλιόμενο από τα πάνω προς τα κάτω. Με το άγγιγμα σε ένα πεδίο ώρας, το αναδυόμενο πλαίσιο κλείνει και στην οθόνη επιλογής εμφανίζεται η ώρα που επιλέχτηκε.



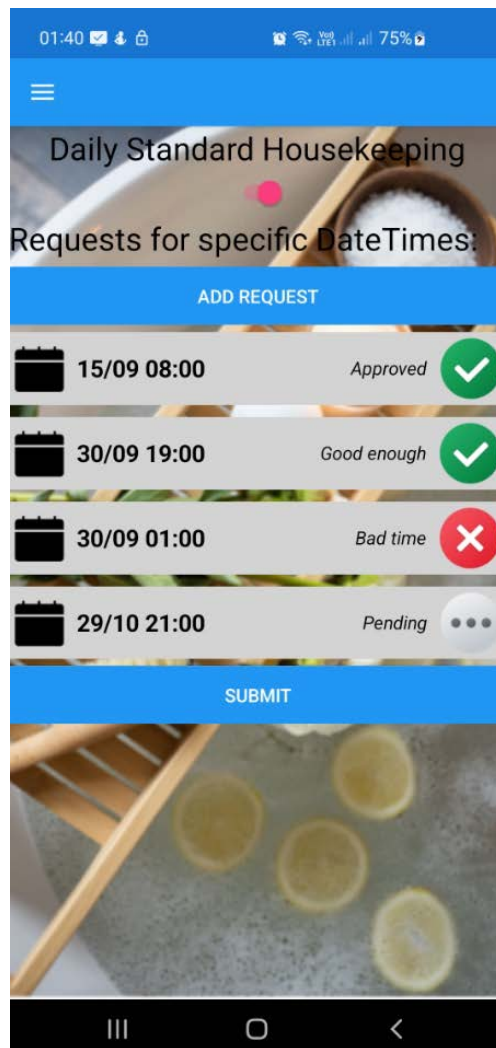
Εικόνα 4.1.4.6 Καταχωρημένη οθόνη κράτησης τραπέζιού

Στην παραπάνω εικόνα (4.1.4.6) εμφανίζονται οι επιλογές αριθμού ατόμων, ημέρας και ώρας για την κράτηση του τραπέζιού. Τέλος, μπορεί να εισαχθεί σχόλιο στο πλαίσιο πάνω από το κουμπί «Submit». Για να δημιουργηθεί το αίτημα χρειάζεται απλά η χρήση του κουμπιού «Submit» και το αίτημα θα εμφανιστεί στην λίστα αιτημάτων στην οθόνη Restaurant.

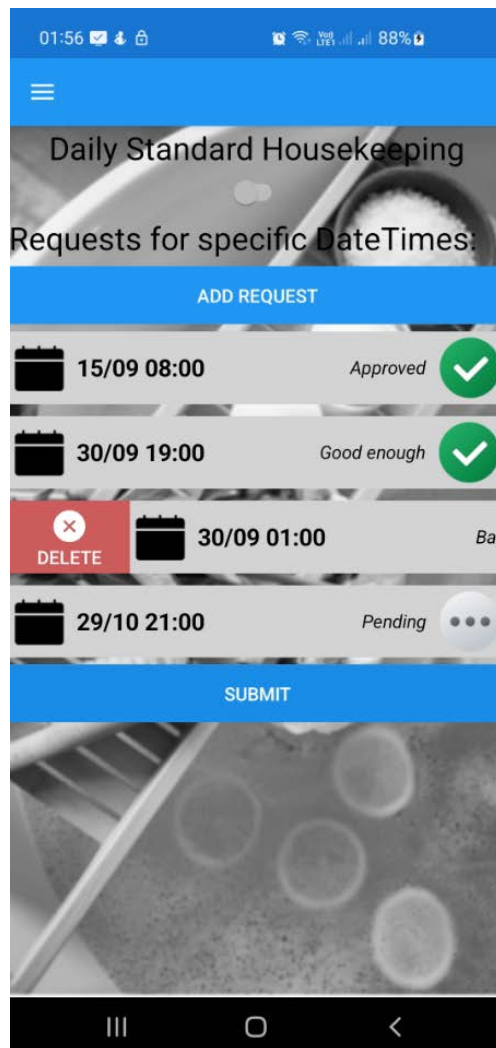
4.1.5 Οθόνη Cleaning

Με την είσοδο του χρήστη πελάτη στην οθόνη Cleaning γίνονται εμφανή τα αιτήματα που έχει ήδη κάνει και η κατάσταση «Daily State Housekeeping». Η κατάσταση αυτή ενεργοποιείται με την μπάρα που υπάρχει στο άνω μέρος της οθόνης, με το σύρσιμο στα δεξιά. Με σύρσιμο στα αριστερά, απενεργοποιείται. Γίνεται αντιληπτό πως όταν απενεργοποιηθεί, το φόντο της οθόνης θα γίνει ασπρόμαυρο.

Εφαρμογή Κινητής Συσκευής και Πληροφοριακό Σύστημα για το Έξυπνο Ξενοδοχείο
Σμυρλής Γεώργιος, Κοσκινάς Χαράλαμπος

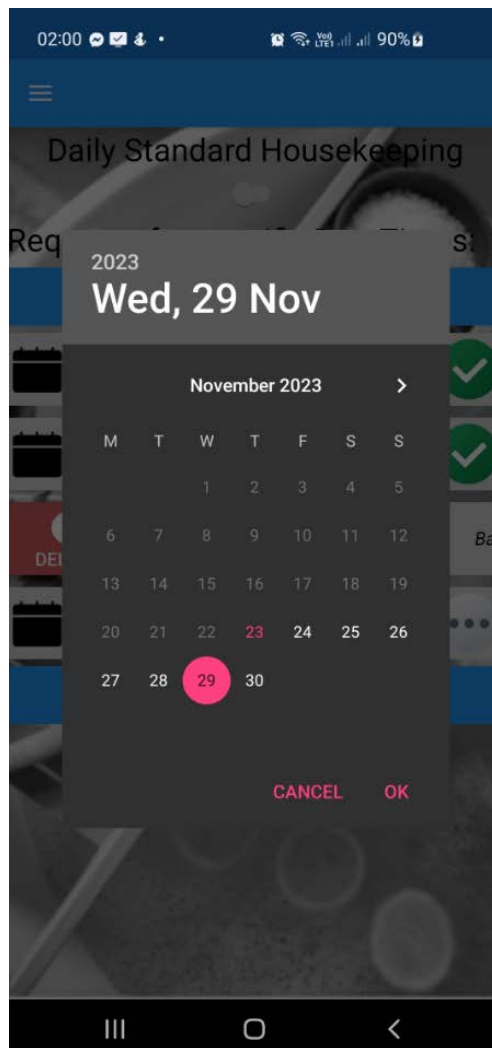


Εικόνα 4.1.5.1 Η οθόνη Cleaning



Εικόνα 4.1.5.2 Οθόνη Cleaning απενεργοποιημένου Housekeeping

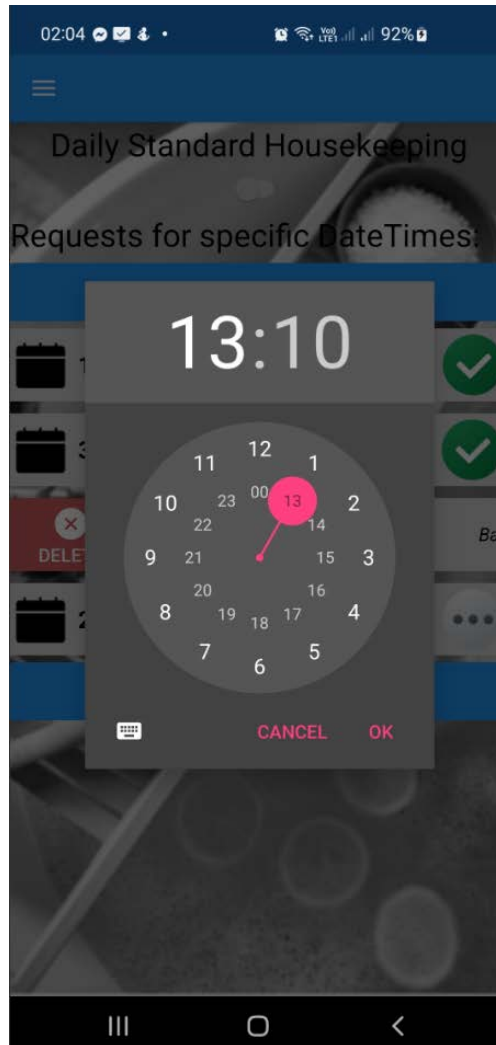
Παρατηρείται επίσης το κουμπί «Add Request» και από κάτω η λίστα των αιτημάτων. Το πεδίο του κάθε αιτήματος και εδώ, αντίστοιχα με το κεφάλαιο 4.1.4, αν κυλίσει προς τα δεξιά θα εμφανιστεί η επιλογή «Delete». Στην κάθε γραμμή αιτήματος, από αριστερά προς τα δεξιά, εμφανίζεται η ημερομηνία και η ώρα του αιτήματος. Στην δεξιά πλευρά της κάθε γραμμής, εμφανίζεται το εικονίδιο κατάστασής της, όπως και το μήνυμα του διαχειριστή. Για να δημιουργηθεί ένα αίτημα συγκεκριμένης ημερομηνίας καθαρισμού, γίνεται χρήση του κουμπιού «Add Request», όπου εμφανίζεται επιλογή ημερομηνίας και επιλογή ώρας, όπως φαίνεται στις παρακάτω εικόνες (4.1.5.3, 4.1.5.4)



Εικόνα 4.1.5.3 Επιλογή ημερομηνίας για Αίτημα Καθαρισμού Δωματίου

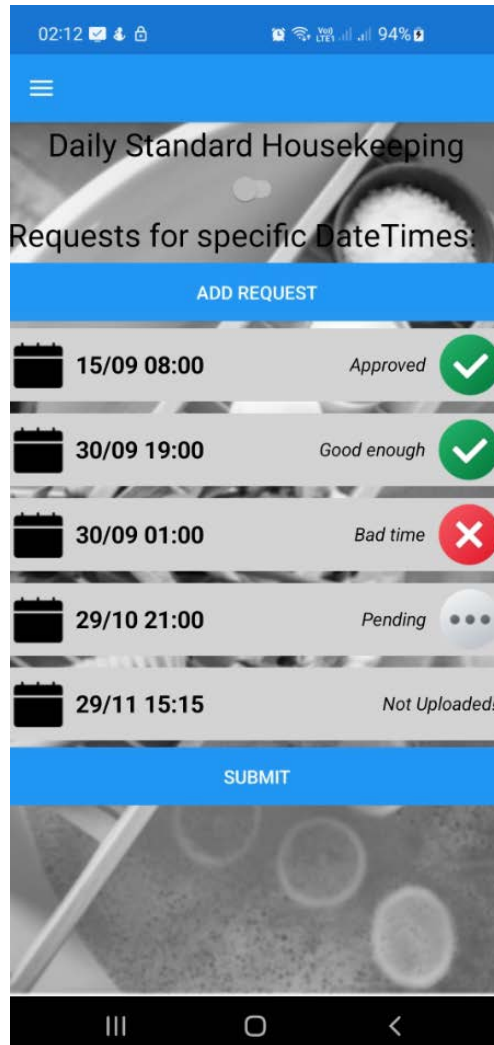
Η επιλογή ημερομηνίας γίνεται όπως και στην περίπτωση που αναφέρεται στο κεφάλαιο 4.1.3, σχετικά με το αίτημα για τραπέζι στο εστιατόριο. Απλά, με το που γίνει η χρήση του πεδίου «OK», εμφανίζεται αναδυόμενο πλαίσιο επιλογής συγκεκριμένης ώρας και λεπτού.

Στο συγκεκριμένο πλαίσιο, γίνεται πρώτα η επιλογή ώρας. Αυτό συμβαίνει με το άγγιγμα του χρήστη σε έναν αριθμό, όπου υπάρχουν όλες οι ώρες του 24ωρου. Στον εξωτερικό κύκλο, με πιο άσπρα γράμματα, εμφανίζονται οι ώρες 1 έως 12, ενώ στον εσωτερικό οι ώρες 13 έως 23 αλλά και 0 (12 η ώρα το βράδυ). Με το άγγιγμα στην επιθυμούμενη ώρα, ο χρήστης έπειτα επιλέγει και το λεπτό της ώρας, όπου εκεί υφίσταται ένας μόνο κύκλος, ο εξωτερικός, σε στυλ ρολογιού.



Εικόνα 4.5.1.5.4 Επιλογή ώρας και λεπτού για Αίτημα Καθαρισμού Δωματίου

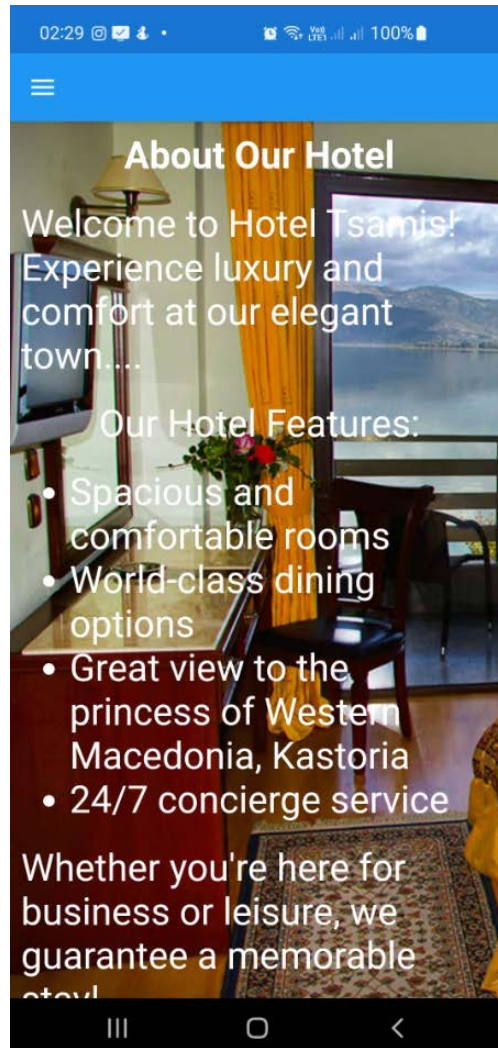
Τέλος, αφότου γίνουν οι παραπάνω δύο επιλογές, δηλαδή ημερομηνίας και ώρας, το αίτημα εμφανίζεται στη λίστα αιτημάτων, με το δεξιό μέρος του να αναγράφει «Not Uploaded!». Για να γίνει το «ανέβασμά» του στη βάση, ώστε να ενημερωθεί το προσωπικό του ξενοδοχείου, χρειάζεται να γίνει η χρήση του κουμπιού «Submit». Το κουμπί αυτό ανεβάζει όσα αιτήματα έχουν δημιουργηθεί αλλά δεν έχουν σταλεί στη βάση δεδομένων. Έπειτα, αν υπάρχει σύνδεση δικτύου και επικοινωνεί σωστά με το ενδιαμέσο λογισμικό, τότε θα αλλάξει η κατάσταση του αιτήματος σε «Pending», δηλαδή το αίτημα θα βρίσκεται σε αναμονή.



Εικόνα 4.1.5.5 Οθόνη Cleaning με αίτημα που δεν έχει «ανέβει»

4.1.6 Οθόνη About

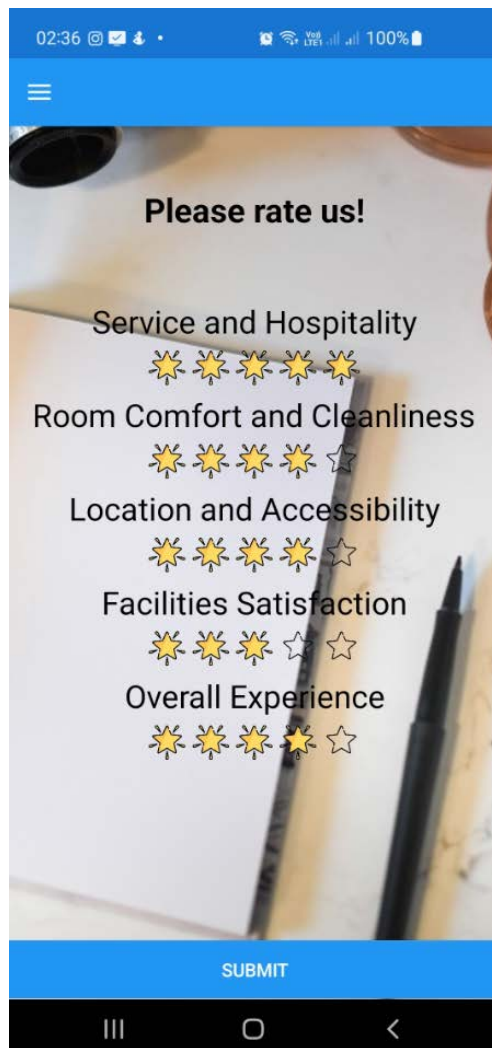
Η οθόνη About εμφανίζει επιπρόσθετες πληροφορίες για το ξενοδοχείο. Μπορεί να παραμετροποιηθεί από την διοίκηση του ξενοδοχείου για να δείχνει κάτι άλλο, όπως τις οδηγίες χρήσης της Εφαρμογής.



Εικόνα 4.1.6 Οθόνη About

4.1.7 Οθόνη Rating

Η οθόνη Rating προάγει την δυνατότητα κριτικής προς το ξενοδοχείο. Αναδεικνύονται τα πεδία της κριτικής, δηλαδή «Service and Hospitality», «Room Comfort and Cleanliness», «Location and Accessibility», «Facilities Satisfaction», «Overall Experience». Κάτω από τον κάθε τίτλο πεδίου, εμφανίζονται πέντε αστέρια. Στην αρχική τους μορφή κανένα από αυτά δεν λάμπει. Ο χρήστης-πελάτης, μπορεί να κρίνει από το 0 έως το 5 το κάθε πεδίο, κάνοντας τα αστέρια να λάμπουν με το άγγιγμά του. Τέλος για το ανέβασμα της κριτικής χρησιμοποιείται πάλι το κουμπί Submit, οπότε και εμφανίζεται αναδυόμενο πεδίο με μήνυμα πως η κριτική αποστάληκε ή μήνυμα λάθους.



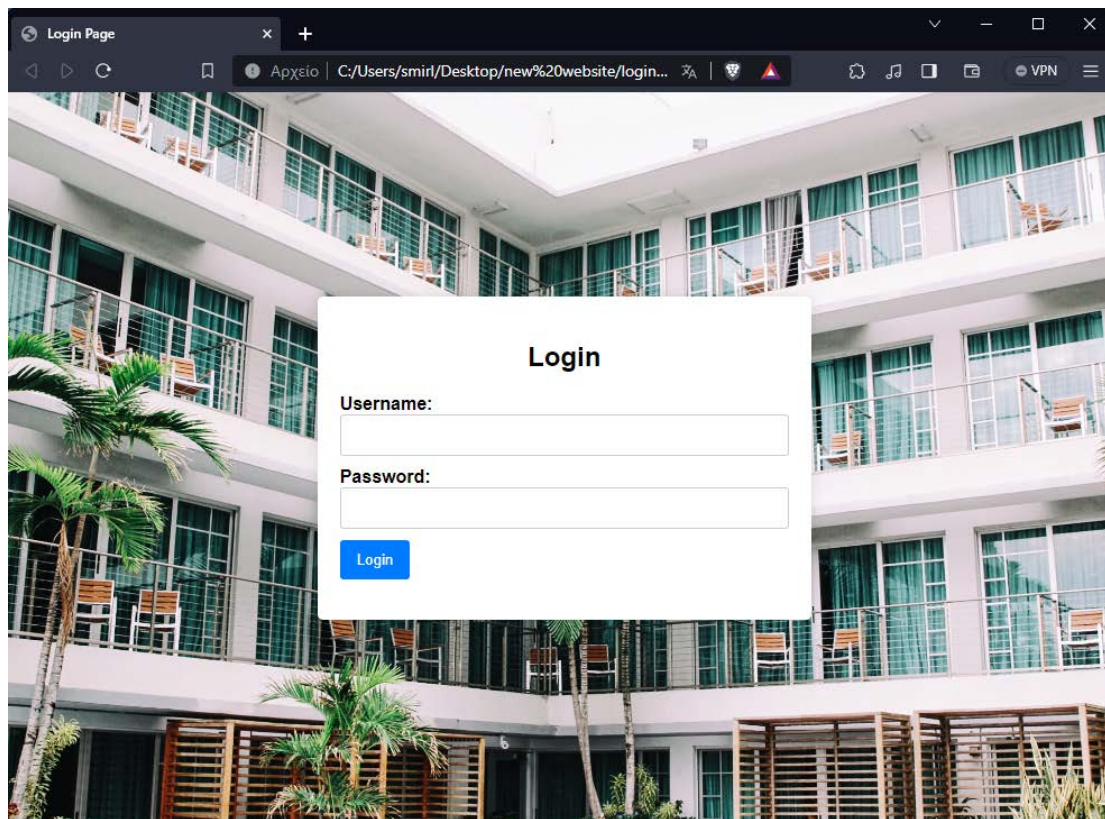
Εικόνα 4.1.7 Η οθόνη Rating

4.2 Εγχειρίδιο Εφαρμογής Ιστού

Η εφαρμογή ιστού αναπτύχθηκε για την πλευρά της διαχείρισης του ξενοδοχείου και του προσωπικού, των εργαζόμενων δηλαδή στις υπηρεσίες του ξενοδοχείου. Για την ορθή λειτουργία της Εφαρμογής Ιστού, ο χρήστης χρειάζεται να είναι τύπου Admin και να πραγματοποιήσει αρχικά την σύνδεσή του στη σελίδα login.html. Για τον αρχικό χρήστη τύπου διαχειριστή, δημιουργείται ο λογαριασμός του μέσα στη βάση δεδομένων από τους προγραμματιστές.

4.2.1 Login

Η σελίδα Login είναι η πρώτη σελίδα που πρέπει να ανοιχτεί, αν είναι η πρώτη φορά που γίνεται χρήση της Εφαρμογής Ιστού. Με το άνοιγμα αυτής της σελίδας, όπως παρατηρούμε, υπάρχουν δύο αλφαριθμητικά πεδία για την εισαγωγή του ονόματος και του κωδικού.



Εικόνα 4.2.1 Η σελίδα Login

Με την καταχώρηση των στοιχείων και το πάτημα του κουμπιού login, γίνεται ο έλεγχος από το ενδιαμέσο λογισμικό των στοιχείων αν συμβαδίζουν με υπάρχων καταχωρημένο χρήστη στη Βάση Δεδομένων. Εάν τα στοιχεία είναι ορθά, επιστρέφεται στον φυλλομετρητή (Browser) το κατάλληλο Token, που θα φέρει τα στοιχεία του ποιος είναι και αυτομάτως αποθηκεύεται σε μορφή Cookie. Επίσης, ανοίγει η επόμενη σελίδα, η σελίδα της Ρεσεψιόν που φέρει τις γενικές πληροφορίες του ξενοδοχείου.

4.2.2 Reception

Στην παρούσα σελίδα είναι δυνατή η επεξεργασία και μετατροπή των πληροφοριών που εμφανίζονται στα επιμέρους πεδία στον πίνακα, κάτω από τον τίτλο «Reception». Για να αλλαχτούν δηλαδή δεδομένα που υπάρχουν στη βάση, γίνεται η όποια μετατροπή και εν τέλει ο χρήστης πατάει το κουμπί Submit που βρίσκεται κάτω δεξιά. Με το κουμπί Refresh, γίνεται η ανανέωση των υπαρχόντων στοιχείων. Κάθε φορά που ανοίγει αυτή η σελίδα κάνει λήψη των πεδίων που εμφανίζονται από την Βάση Δεδομένων.

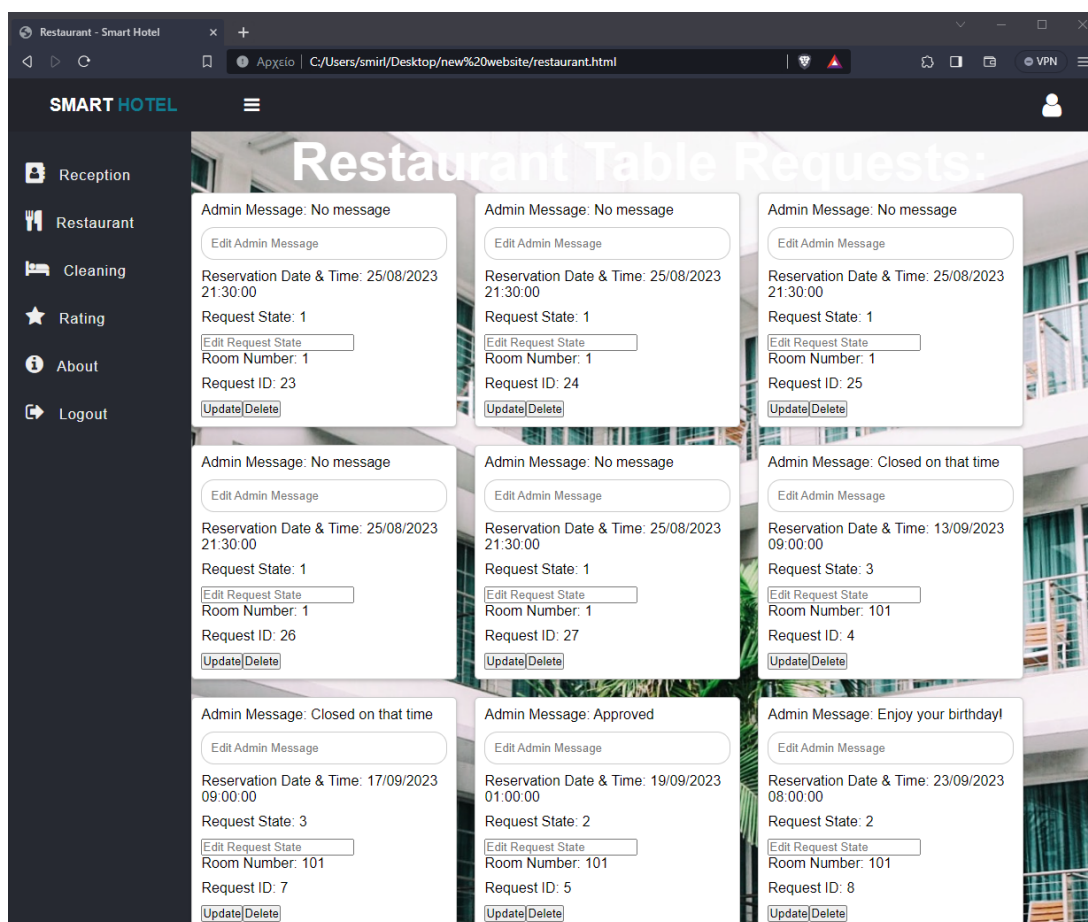
Hotel Name:	The Enchanted Wizard Inn
Hotel Address:	Hogwarts Castle, Scotland, UK
Hotel Info:	Experience the magic of hospitality at The Enchanted Wizard Inn. Located in the heart of the mystical Hogwarts Castle, our hotel offers spellbinding comfort and enchanting experiences.
Hotel Email:	info@enchantedwizardinn.com
Hotel Website:	www.enchantedwizardinn.com
Restaurant Menu Link:	https://external-content.duckduckgo.com/iu/?u=https%3A%2F%2Fim1.din
Reception Telephone:	00302351047552

Εικόνα 4.2.2 Η σελίδα Reception

Το πεδίο Hotel Name φέρει το όνομα του Ξενοδοχείου. Το πεδίο Hotel Address την διεύθυνσή του. Το πεδίο Info αξιοποιείται ώστε να εμπεριέχονται δύο με τρεις προτάσεις σχετικές με το ξενοδοχείο. Το πεδίο Email αξιοποιείται για να φέρει το email που μπορούν οι πελάτες να επικοινωνήσουν με το ξενοδοχείο. Στο πεδίο Hotel Website φιλοξενείται η ηλεκτρονική διεύθυνση του ξενοδοχείου, που φέρει το url της. Το πεδίο Restaurant Menu Link πρέπει να φέρει την διεύθυνση του μενού του ξενοδοχείου, είτε την ανεβασμένη ηλεκτρονικά διεύθυνση φωτογραφίας του μενού. Τέλος, στο πεδίο Reception Telephone όπου αναγράφεται το τηλέφωνο της Ρεσεψιόν του ξενοδοχείου. Για την ομαλή χρήση του πεδίου, η admin θα πρέπει να εισαγάγει τον πλήρη αριθμό τηλεφώνου, μαζί και με τον κωδικό χώρας σε αριθμητική μορφή, δηλαδή για την Ελλάδα 0030.

Στο πάνω αριστερά μέρος της εμφανίζεται επιγραφή που φέρει την ονομασία του Συστήματος. Με το κλικ στο εικονίδιο που θυμίζει «σάντουιτς», ακριβώς δεξιά από την επιγραφή «SmartHotel», αποκρύπτονται οι τίτλοι των σελίδων που βρίσκονται στο μενού στην αριστερή πλευρά της σελίδας. Οι τίτλοι μόνο, όχι τα εικονίδια. Από το μενού αριστερά είναι εφικτή η περιήγηση στις σελίδες που έχουν αναπτυχθεί, με απλό κλικ σε αυτές.

4.2.3 Restaurant και Cleaning



Εικόνα 4.2.3.1 Η σελίδα Restaurant

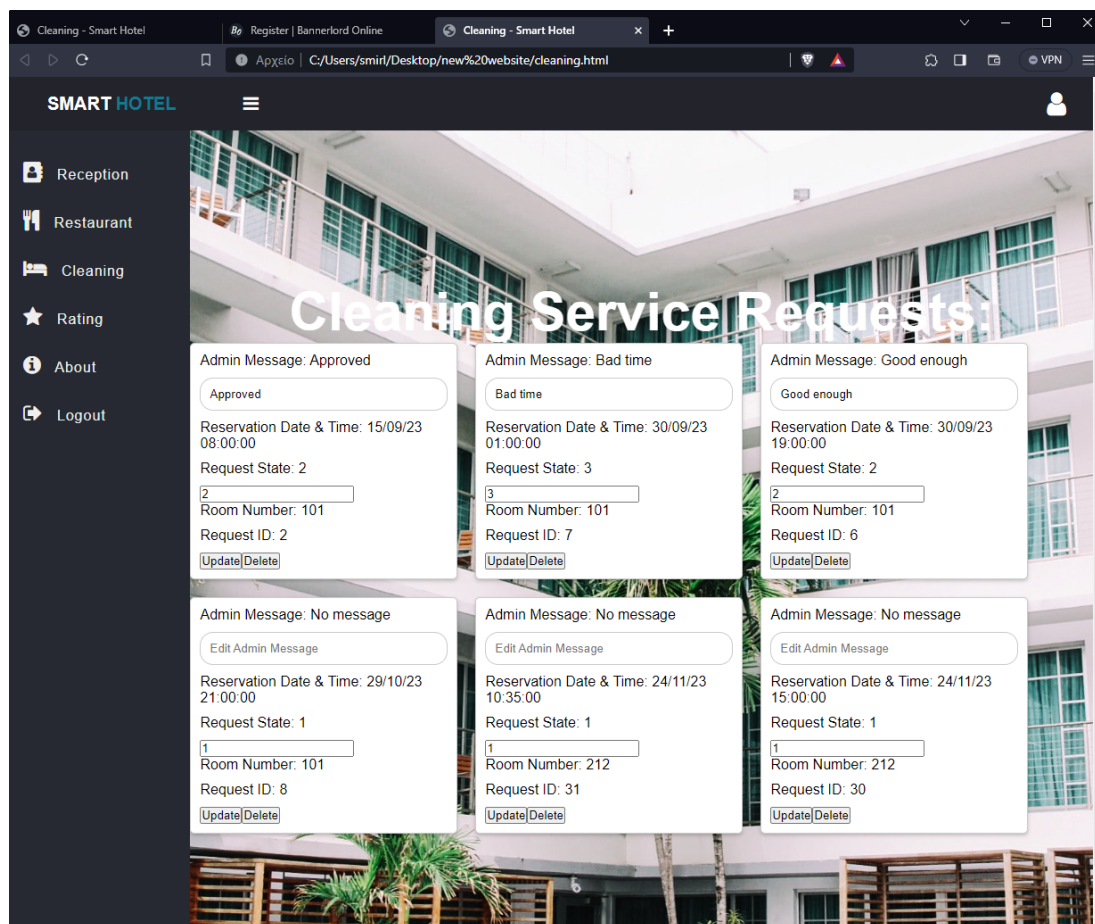
Η παρούσα σελίδα έχει δημιουργηθεί για την διαχείριση των αιτημάτων των πελατών, σχετικά με την κράτηση τραπεζιού στο εστιατόριο του ξενοδοχείου. Τα αιτήματα παρουσιάζονται σε χρονολογική σειρά, από το παλαιότερο στο νεότερο. Κάτω από τον τίτλο «Restaurant Table Requests» παρατηρούνται κάρτες, οι οποίες πρόκειται για κάθε αίτημα το οποίο έχει δημιουργηθεί από τους πελάτες. Επεξεργάσιμα εδώ είναι το αλφαριθμητικό πεδίο Admin Message και το αριθμητικό πεδίο Request State.

Το πεδίο Admin Message έχει δημιουργηθεί για να περιλαμβάνει ένα μήνυμα που αφήνει η διαχείριση του ξενοδοχείου σχετικά με συγκεκριμένο αίτημα του πελάτη. Το μήνυμα αυτό είναι αλφαριθμητικό και μπορεί να αξιοποιηθεί είτε για να φαίνεται ο λόγος ακύρωσης ενός αιτήματος, είτε για να φαίνεται οποιαδήποτε άλλη χρήσιμη πληροφορία χρειάζεται να μάθει ο πελάτης σχετικά με το αίτημά του. Δηλαδή μπορεί να αναγράφει «Approved» σε περίπτωση που ο χρήστης-διαχειριστής κρίνει πως το αίτημά του θα υλοποιηθεί, ή «Declined» αν δεν είναι εφικτό. Αντί για «Declined» βέβαια, μπορεί να εμπεριέχει το γιατί δεν γίνεται αποδεκτό, ώστε ο πελάτης να δημιουργήσει ένα νέο αίτημα που θα γίνει αποδεκτό.

Το πεδίο Request State πρόκειται για την κατάσταση του αιτήματος. Όταν ένα αίτημα δημιουργείται και δεν έχει καταχωρηθεί στη βάση έχει την τιμή 0, ενώ όταν ένα αίτημα «ανεβαίνει» στη βάση δεδομένων, λαμβάνει την τιμή 1, όπου σημαίνει πως είναι σε αναμονή. Στη παρούσα σελίδα δηλαδή, επειδή με το που φορτώνεται λαμβάνει τα αιτήματα από τη βάση, δεν μπορεί να υπάρξει αίτημα με τιμή 0. Η τιμή 2, συμβολίζει πως το αίτημα έχει γίνει αποδεκτό και θα υλοποιηθεί. Η τιμή 3 από την άλλη, χρησιμοποιείται για την απόρριψη ενός αιτήματος, ενώ μπορεί να αξιοποιηθεί και η τιμή 4 ώστε να συμβολίζει πως το αίτημα έχει ολοκληρωθεί.

Σε κάθε κάρτα γίνονται παρατηρήσιμα τα πεδία του αιτήματος. Από την ημερομηνία που θέλει να συμβεί η κράτηση, στο ποιο δωμάτιο την δημιουργεί. Επίσης, στην κάτω πλευρά του, φαίνονται και δύο κουμπιά που φέρουν τους τίτλους «Update» και «Delete». Τα δύο αυτά κουμπιά ενημερώνουν την βάση δεδομένων. Με το κουμπί Update, καταχωρείται η απάντηση του χρήστη-διαχειριστή, για το πεδίο Admin Message και για το κουμπί Request State, δηλαδή για το αν εγκρίθηκε ή απορρίφθηκε το αίτημα του. Με το κουμπί «Delete» διαγράφεται ένα αίτημα από τη βάση δεδομένων. Τέλος το πεδίο «Request ID» δεν είναι για κάποια αξιοποίηση, χρησιμοποιείται για λόγους λογισμικού και δείχνει σε αύξουσα σειρά όλα τα αιτήματα που έχουν γίνει, δηλαδή κάποιος μπορεί μέσα από αυτό να αντιληφθεί ποιο αίτημα συνέβη πριν από κάποιο άλλο.

Αντίστοιχα με την σελίδα Restaurant, η σελίδα Cleaning έχει δημιουργηθεί για την εμφάνιση και απάντηση στα αιτήματα των πελατών. Στη συγκεκριμένη σελίδα, οι κάρτες πρόκειται για τα αιτήματα των πελατών σχετικά με συγκεκριμένες ημερομηνίες που επιθυμούν να γίνει καθαριότητα του δωματίου τους.



Εικόνα 4.2.3.2 Η σελίδα Cleaning

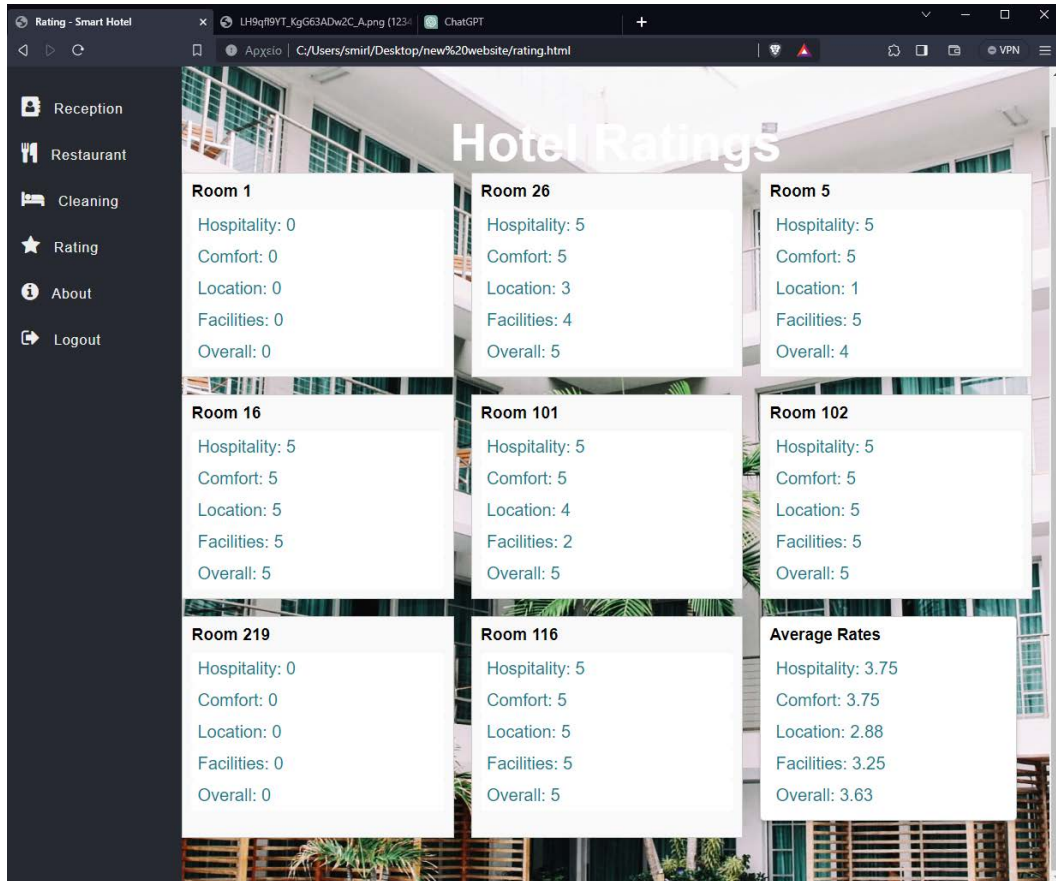
Κάτω από τον τίτλο Cleaning Service Requests, εντοπίζονται με χρονολογική σειρά, από την παλαιότερη στη νεότερη αίτηση. Στην κάθε κάρτα είναι εμφανής η ημερομηνία που αιτείται για καθαρισμό, όπως και το δωμάτιο που κάνει την αίτηση. Επίσης το δωμάτιο το οποίο κάνει το αίτημα φαίνεται έπειτα από το RoomNumber. Επιπλέον, φαίνεται το Request ID, το οποίο πρόκειται για το πρωτεύον κλειδί του κάθε αιτήματος που είναι αύξοντας αριθμός για κάθε αίτημα. Εμφανίζεται επίσης στην κάθε κάρτα, το Admin Message, το οποίο είναι το μήνυμα των χρηστών-διαχειριστών προς το αίτημα, όπως και το Request State, δηλαδή η κατάσταση του αιτήματος.

Σε κάθε κάρτα γίνεται εφικτή η απάντηση του χρήστη-διαχειριστή στο κάθε αίτημα. Εξίσου και εδώ, το πεδίο Admin Message κάνει εφικτό τον χρήστη-διαχειριστή να αφήσει ένα μήνυμα σχετικά με το αίτημα του πελάτη. Μέσα από το Request State, όπως προαναφέρθηκε, ο αριθμός 1 πρόκειται για το ότι το αίτημα βρίσκεται σε αναμονή για απάντηση, ο αριθμός 2 πως το αίτημα εγκρίνεται από την διαχείριση του ξενοδοχείου και ο αριθμός 3 πως το αίτημα απορρίπτεται. Επίσης μπορεί ο αριθμός 4 να αξιοποιηθεί για να δείχνει πως το αίτημα ολοκληρώθηκε.

Με την επεξεργασία λοιπόν των παραπάνω πεδίων, γίνεται εφικτή η απάντηση στα αιτήματα των χρηστών. Στο κάτω μέρος των καρτών εμφανίζονται τα κουμπιά Submit και Delete. Για την απάντηση στο αίτημα οπότε, αφότου γίνει η επεξεργασία των

πεδίων Admin Message και Request State, ο χρήστης-διαχειριστής με το πάτημα του κουμπιού ενημερώνει το αίτημα. Γίνεται δυνατή επίσης η διαγραφή του από το ιστορικό, με το κουμπί Delete.

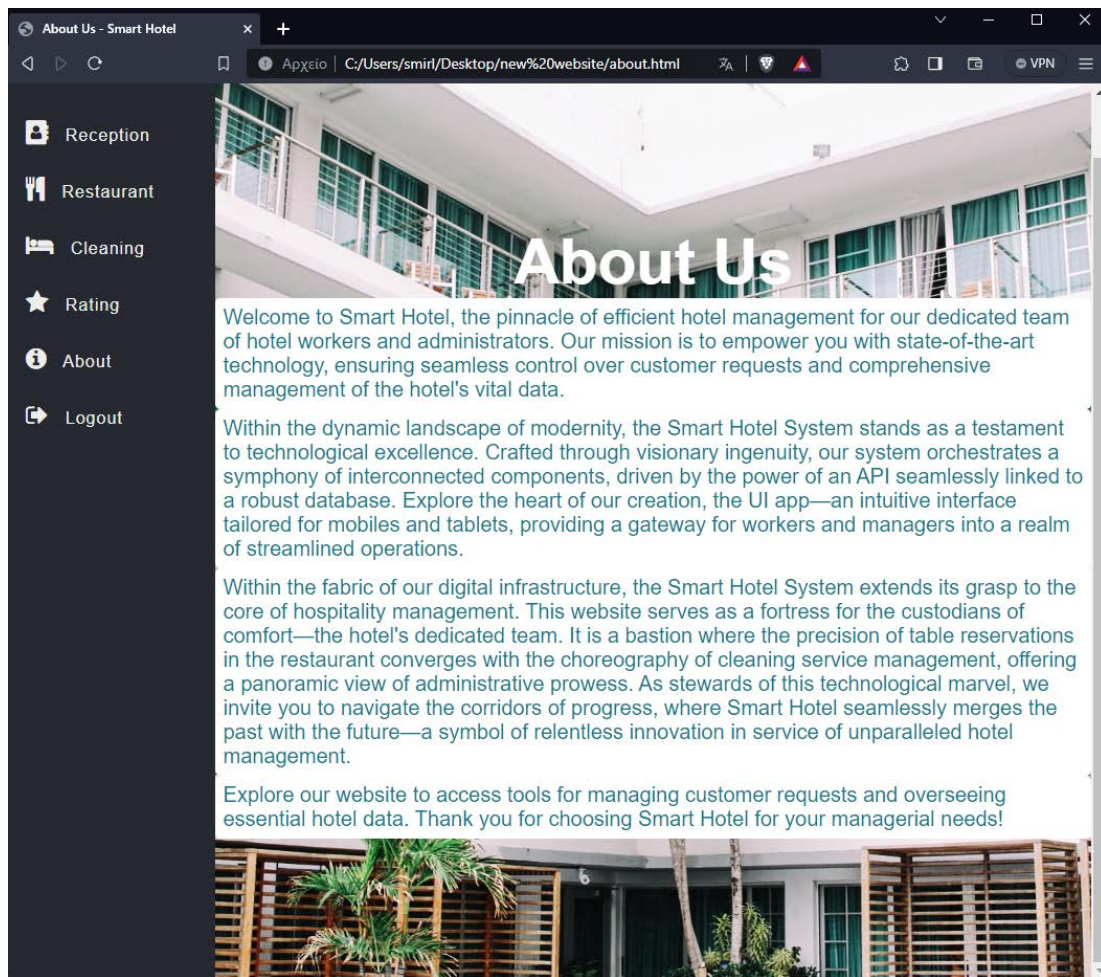
4.2.4 Rating, About και Logout



Εικόνα 4.2.4.1 Η σελίδα Rating

Στη σελίδα Rating παρουσιάζονται οι αξιολογήσεις κάθε δωματίου προς τις υπηρεσίες του ξενοδοχείου. Σε κάθε καρτέλα παρουσιάζεται η αξιολόγηση του κάθε δωματίου ως προς τις υπηρεσίες του ξενοδοχείου.

Στη σελίδα About Us εμφανίζεται κείμενο σχετικά με το σύστημα «SmartHotel» και σχετικά με την ιστοσελίδα. Μπορεί να αξιοποιηθεί μελλοντικά και για συγκεκριμένες οδηγίες χρήσης της εφαρμογής Ιστού.



Εικόνα 4.2.4.2 Η σελίδα About Us

Και τέλος με το κλικ στην επιλογή Logout στο μενού αριστερά, διαγράφεται το cookie που έχει αποθηκευμένα τα στοιχεία χρήστη. Οπότε, «ξεχνιέται» το ποιος είναι συνδεδεμένος στην εφαρμογή ιστού, με αποτέλεσμα να μην μπορούν να ληφθούν και να σταλούν ξανά πληροφορίες, έως ότου ξαναγίνει ορθή σύνδεση κάποιου χρήστη που είναι διαχειριστής.

Συμπεράσματα

Κατά την εκπόνηση της παρούσας εργασίας κατανοήσαμε την αναγκαιότητα του προγραμματισμού ακόμα και στον τομέα του τουρισμού. Αναπτύξαμε λογισμικό για την κατασκευή του Έξυπνου Ξενοδοχείου, μία εφαρμογή που διευκολύνει βασικές υπηρεσίες του τουρισμού, όπως είναι η διαμονή σε ξενοδοχεία. Χρησιμοποιήσαμε τις συγκεκριμένες τεχνολογίες που αναπτύσσονται διεξοδικά στο δεύτερο και τρίτο κεφάλαιο, χωρίς ωστόσο να είναι αυτές οι μοναδικές.

Η χρησιμότητα και η αποτελεσματικότητα της εργασίας γίνεται ακόμα πιο φανερή στο τέταρτο κεφάλαιο, στο Εγχειρίδιο Χρήστη, όπου φαίνεται ξεκάθαρα πώς η χρησιμότητα του λογισμικού μπορεί πολύ έξυπνα να αξιοποιηθεί και να εφαρμοστεί για να κάνει καλύτερη και ευκολότερη τη ζωή των πελατών στα ξενοδοχεία.

Μελλοντικά, η εφαρμογή θα μπορούσε να αποκτήσει επιπρόσθετη λειτουργικότητα ώστε να καλύπτει το κομμάτι της τιμολόγησης των υπηρεσιών και της ηλεκτρονικής πληρωμής. Επιπρόσθετα, με τη χρήση προηγμένων εργαλείων γραφιστικής σχεδίασης, θα μπορούσαν να παραχθούν καλύτερης ποιότητας γραφικά ώστε οι εφαρμογές να παρέχουν ένα πιο φιλικό περιβάλλον χρήσης. Τέλος, σημαντική μελλοντική εργασία είναι η αξιολόγηση της εφαρμογής από τους χρήστες. Στόχος μας είναι η παροχή της εφαρμογής σε ένα ξενοδοχείο προκειμένου να συλλέξουμε αξιολογήσεις και ανάδραση από τους πελάτες και τους υπαλλήλους του ξενοδοχείου.

Εφαρμογή Κινητής Συσκευής και Πληροφοριακό Σύστημα για το Έξυπνο Ξενοδοχείο
Σμυρλής Γεώργιος, Κοσκινάς Χαράλαμπος

Βιβλιογραφία

- [1] DRM, «"New Product Development Glossary". Archived from the original on 13 July 2018. Retrieved 29 October 2006.».
- [2] T. Koetsier, «On the prehistory of programmable machines: musical automata, looms, calculators», *Mechanism and Machine Theory* (Elsevier) 36 (5): 589–603, doi:10.1016/S0094-114X(01)00005-2.
- [3] J. Fuegi και J. Francis, «Lovelace & Babbage and the Creation of the 1843 'notes'». *IEEE Annals of the History of Computing* 25 (4): 16. doi:10.1109/MAHC.2003.1253887..
- [4] R. I. Soare, *Turing Computability: Theory and Applications (Theory and Applications of Computability)*, Springer; 1st ed. 2016 edition (28 Jun. 2016), 2016.
- [5] J. Muir, «A short history of software development,» 4 May 2021. [Ηλεκτρονικό]. Available: <https://blogs.oregonstate.edu/articles/a-short-history-of-software-development/>. [Πρόσβαση 11 Νοέμβριος 2023].
- [6] «Computer History Museum,» [Ηλεκτρονικό]. Available: <https://www.computerhistory.org/timeline/>. [Πρόσβαση 7 Νοέμβριος 2023].
- [7] D. Cassel, 11 Oct 2020. [Ηλεκτρονικό]. Available: <https://thenewstack.io/the-hardware-and-software-used-in-space/>. [Πρόσβαση 7 Νοέμβριος 2023].
- [8] A. K. Ashish Kakar, «A Brief History of Software Development and Manufacturing,» σε *SAIS 2020 Virtual*, 2020.
- [9] Altexsoft, «altexsoft.com,» 2022 Σεπ 10. [Ηλεκτρονικό]. Available: <https://www.altexsoft.com/blog/smart-hotel-infrastructure/>. [Πρόσβαση 2 Νοέμβριος 2023].
- [10] J. Hollander, «HotelTechReport.com,» 19 Ιουλ 2023. [Ηλεκτρονικό]. Available: <https://hoteltechreport.com/news/smart-hotels>. [Πρόσβαση 18 Οκτώβριος 2023].
- [11] Nexus Integra, «nexusintegra.io,» 2023. [Ηλεκτρονικό]. Available: <https://nexusintegra.io/smart-hotels/>. [Πρόσβαση 2 Νοέμβριος 2023].
- [12] E. Ehrli, «devblogs.microsoft.com,» 13 December 2017. [Ηλεκτρονικό]. Available: <https://devblogs.microsoft.com/visualstudio/connect-2017-smarthotel360-demo-apps-and-architecture/>. [Πρόσβαση 2 Νοέμβριος 2023].
- [13] L. Chang, «digitaltrends.com,» 15 Νοέμβριος 2017. [Ηλεκτρονικό]. Available: <https://www.digitaltrends.com/home/marriott-iot-guestroom-lab/>. [Πρόσβαση 23

Οκτώβριος 2023].

- [14] E. Hertzfeld, «hotelmanagement.net,» 16 Νοέμβριος 2017. [Ηλεκτρονικό]. Available: <https://www.hotelmanagement.net/tech/marriott-creates-smart-guest-room-lab>. [Πρόσβαση 26 Οκτώβριος 2023].
- [15] J. Sbounias, «hotelieracademy.org,» 2021, [Ηλεκτρονικό]. Available: <https://www.hotelieracademy.org/5-smart-hotels-that-confirm-the-potentials-of-this-new-hotel-trend/>. [Πρόσβαση 27 Οκτώβριος 2023].
- [16] Microsoft, «What is Xamarin?,» Microsoft, 20 9 2022. [Ηλεκτρονικό]. Available: <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>. [Πρόσβαση 7 Νοέμβριος 2023].
- [17] A. Binstock, «InformationWeek,» 11 June 2011. [Ηλεκτρονικό]. Available: <https://web.archive.org/web/20111207173458/http://www.informationweek.com/news/development/architecture-design/229700276>. [Πρόσβαση 7 Νοέμβριος 2023].
- [18] S. Guthrie, «Microsoft,» 24 Feb 2016. [Ηλεκτρονικό]. Available: <https://blogs.microsoft.com/blog/2016/02/24/microsoft-to-acquire-xamarin-and-empower-more-developers-to-build-apps-on-any-device/>. [Πρόσβαση 7 Νοέμβριος 2023].
- [19] Altexsoft, 8 Φεβ 2019. [Ηλεκτρονικό]. Available: <https://www.altexsoft.com/blog/13-apps-made-with-xamarin-cross-platform-development-in-practice/>. [Πρόσβαση 20 Οκτώβριος 2023].
- [20] J. Clark, «back4app,» [Ηλεκτρονικό]. Available: <https://blog.back4app.com/xamarin-alternatives/>. [Πρόσβαση 20 Οκτώβριος 2023].
- [21] Flask-Pocoo, «Flask.Pocoo.org,» 17 Μάιος 2010. [Ηλεκτρονικό]. Available: <https://web.archive.org/web/20180517082208/http://flask.pocoo.org/extensions/>. [Πρόσβαση 22 Οκτώβριος 2023].
- [22] «Flask.Pocoo.org,» Flask-Pocoo, 1 Σεπτέμβριος 2014. [Ηλεκτρονικό]. Available: <https://web.archive.org/web/20171117015927/http://flask.pocoo.org/docs/0.10/foreword>. [Πρόσβαση 11 Νοέμβριος 2023].
- [23] StackOverflow, [Ηλεκτρονικό]. Available: <https://stackoverflow.com/tags?tab=popular>. [Πρόσβαση 20 Οκτώβριος 2023].
- [24] J. Clark, «back4app,» 2022. [Ηλεκτρονικό]. Available: <https://blog.back4app.com/flask-framework-alternatives/>. [Πρόσβαση 16 Οκτώβριος 2023].
- [25] SQLite, 1 8 2022. [Ηλεκτρονικό]. Available: <https://sqlite.org/mostdeployed.html>.

[Πρόσβαση 22 Οκτώβριος 2023].

- [26] M. L. Gillenson, Βασικές Αρχές Συστημάτων Διαχείρισης Βάσεων Δεδομένων, 2η Έκδοση επιμ., Broken Hill Publishers LTD, 2012.
- [27] SQLite, «About SQLite,» 10 10 2023. [Ηλεκτρονικό]. Available: <https://www.sqlite.org/about.html>. [Πρόσβαση 20 Οκτώβριος 2023].
- [28] Slant, 2023. [Ηλεκτρονικό]. Available: <https://www.slant.co/options/304/alternatives/~sqlite-alternatives>. [Πρόσβαση 8 Νοέμβριος 2023].
- [29] R. C. J. K. Laura Lemay, Πλήρες εγχειρίδιο HTML 5, CSS & JAVASCRIPT, 7η έκδοση επιμ., 2016.
- [30] Douglas Crockford, 27 Ιούνιος 2019. [Ηλεκτρονικό]. Available: <https://www.crockford.com/little.html>. [Πρόσβαση 12 Οκτώβριος 2023].
- [31] ECMA, «ecmascript.org,» 23 Οκτώβριος 2007. [Ηλεκτρονικό]. Available: <https://web.archive.org/web/20090326234420/http://www.ecmascript.org/es4/spec/overview.pdf>. [Πρόσβαση 20 Οκτώβριος 2023].
- [32] MDN, «developer.mozilla.org,» 30 Jun 2023. [Ηλεκτρονικό]. Available: <https://developer.mozilla.org/en-US/docs/Learn/CSS>. [Πρόσβαση 25 Οκτώβριος 2023].
- [33] Μ. Κομβούτη-Βέρου, «Εισαγωγή στο CSS,» Οικονομικό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής, [Ηλεκτρονικό]. Available: Μιχαήλια Κομβούτη-Βέρου . [Πρόσβαση 24 Οκτώβριος 2023].