

ΤΕΧΝΟΛΟΓΙΕΣ ΕΦΑΡΜΟΓΩΝ ΔΙΑΔΥΚΤΥΟΥ ΕΡΓΑΣΙΑ BOOKING

Η εργασία αυτή υλοποιήθηκε από:

Τσουράκης Ορφέας (ΑΜ: 1115201700175)

Ταράτσας Ιωάννης (ΑΜ: 1115201700160)

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1.Εισαγωγή

2.Back-end

2.1 Models

2.2 Controllers

2.3 Repositories

2.4 Security

3.Front-end

3.1 Models

3.2 Services

3.3 Components

3.4 Guards

3.5 App module

4.Bonus

4.1 Recommendation

4.2 Recommendation Start

4.3 Εισαγωγή Δεδομένων στο Dataset

5.ΕΠΙΛΟΓΟΣ

1. ΕΙΣΑΓΩΓΗ

Στην εργασία αυτή υλοποιήσαμε μια εφαρμογή ενοικίασης δωματίων. Στην εφαρμογή έχουμε ενσωματώσει έναν διαχειριστή (Admin), και έχουμε δημιουργήσει τους ρόλους που μας ζητούνται (Tenant, Host) οι οποίοι κάνουν κρατήσεις και καταχωρούν διαμερίσματα αντίστοιχα. Παράλληλα, υπάρχει και η δυνατότητα για κάποιον που δεν έχει λογαριασμό, να κάνει αναζητήσεις και να περιηγηθεί στους χώρους, χωρίς όμως να του δίνεται η δυνατότητα να κάνει κρατήσεις (Guest). Το Bonus κομμάτι με τα προτεινόμενα διαμερίσματα υλοποιήθηκε και έτσι σε κάθε συνδεδεμένο ενοικιαστή εμφανίζονται τα πέντε πιο σχετικά διαμερίσματα ανάλογα με την αλληλεπίδραση του στην εφαρμογή. Στα παρακάτω κεφάλαια θα αναλυθούν οι λεπτομέρειες υλοποίησης του Back-End και του Front-End αντίστοιχα.

2. BACK-END

Ο Server ξεκινάει με το Demo2Application, το οποίο εισάγει τους ρόλους του οικοδεσπότη και του ενοικιαστή(αν δεν υπάρχουν) και δημιουργεί έναν νέο χρήστη με τον ρόλο του admin με τα αντίστοιχα στοιχεία του(username: admin, password:admin)

2.1 Models

Τα models είναι ο κορμός της εφαρμογής, αφού αναπαριστούν όλα τα αντικείμενα. Συγκεκριμένα έχουμε τα εξής μοντέλα:

- Ο χρήστης (User) περιλαμβάνει τα βασικά χαρακτηριστικά ενός χρήστη της εφαρμογής(όπως όνομα, username, φωτογραφία, κωδικό κτλ) και ακόμα έχουμε προσθέσει συσχετίσεις με τους ρόλους του, τα διαμερίσματα που κατέχει αν είναι διαχειριστής, τα διαμερίσματα που έχει νοικιάσει αν είναι ενοικιαστής, τις κριτικές που έχει κάνει και έχει λάβει από άλλους χρήστες, όπως και τα μηνύματα, και τέλος οι αλληλεπιδράσεις του με την εφαρμογή και τα προτεινόμενα για αυτόν διαμερίσματα(bonus).
- Ο ρόλος (Role) περιλαμβάνει το όνομα του ρόλου με βάση το enumeration RoleNames, οι οποίοι εισάγονται κατά το ξεκίνημα της εφαρμογής αν δεν υπάρχουν.
- Το διαμέρισμα (Department) περιλαμβάνει τα βασικά χαρακτηριστικά ενός χρήστη της εφαρμογής(όπως όνομα, διεύθυνση, φωτογραφία μικρή, επιπλέον φωτογραφίες, αριθμό κρεβατιών, μπάνιων, παροχές κτλ) και ακόμα έχουμε προσθέσει συσχετίσεις με τον ιδιοκτήτη, τις κρατήσεις, τις κριτικές του, τα μηνύματα των χρηστών που αναφέρεται, τις αλληλεπιδράσεις των χρηστών για το συγκεκριμένο διαμέρισμα και τους χρήστες που έχει προταθεί(bonus). Όσον αφορά τον τύπο του οι τιμές του ακολουθούν αυστηρά το enumeration DType. Το κόστος υπολογίζεται από δύο παράγοντες: το costPerDay το οποίο είναι το ελάχιστο κόστος κράτησης του διαμερίσματος(χωρίς κανένα άτομο) συν το costPerPerson επί τον αριθμό των ατόμων της κράτησης.
- Η κράτηση (Booking) περιλαμβάνει τις ημερομηνίες που έχει κλείσει ένας ενοικιαστής σε ένα συγκεκριμένο διαμέρισμα και τον αριθμό των ατόμων(ενηλίκων, παιδιών).
- Η κριτική (Review) περιλαμβάνει το κείμενο, τα αστέρια, τον αποστολέα, και τον παραλήπτη ή το διαμέρισμα στο οποίο αναφέρεται αντίστοιχα.
- Η φωτογραφία (Photo) περιλαμβάνει το όνομα, τον τύπο και τα bytes.

- Το μήνυμα(Message) περιλαμβάνει το κείμενο, μια μεταβλητή για να ξεχωρίζει η ερώτηση από την απάντηση, τον αποστολέα, τον παραλήπτη και το διαμέρισμα στο οποίο αναφέρεται αντίστοιχα, και η συσχέτιση του με κάποιο άλλο μήνυμα.
- Η αλληλεπίδραση(Interaction) περιλαμβάνει έναν μετρητή για το πόσες φορές ένας χρήστης έχει αναζητήσει ένα διαμέρισμα

2.2 Controllers

Τα controllers περιλαμβάνουν συναρτήσεις για την επικοινωνία μεταξύ της βάσης(MySql) και του front-end.

- User Controller: **A)** Μέθοδος all η οποία επιστρέφει όλες τις λεπτομέρειες των χρηστών στον διαχειριστή χωρίς τις φωτογραφίες. **B)** Μέθοδος sign up για την εγγραφή των χρηστών, η οποία ελέγχει αν υπάρχουν ήδη χρήστες με ίδιο όνομα ή email, κωδικοποιεί τον κωδικό με το BCryptPasswordEncoder, συσχετίζει τον κατάλληλο ρόλο και εισάγει την φωτογραφία του προφίλ του και επιστρέφει ένα JWT token στο authorization header. **Γ)** Μέθοδος seeProfile η οποία ένα προφίλ αν αυτό υπάρχει. **Δ)** Μέθοδος changePassword η οποία τσεκάρει αν ο χρήστης έβαλε σωστά τον υπάρχοντα κωδικό και κωδικοποιεί τον νέο κωδικό. **Ε)** Η μέθοδος Edit User η οποία κάνει παρόμοιους ελέγχους με την sign up και αλλάζει τα πεδία τα οποία δεν είναι κενά. Στο τέλος επιστρέφει το νέο JWT token αν έχει αλλάξει το username. **ΣΤ)** Η μέθοδος changeMainPhoto ελέγχει αν η φωτογραφία είναι έγκυρη και την ανανεώνει. **Ζ)** Η μέθοδος acceptHost επιτρέπει τον διαχειριστή να δεχτεί έναν χρήστη με την μέθοδο του οικοδεσπότη.
- Department Controller: **A)** Μέθοδος newDepartment για την εγγραφή των διαμερισμάτων, η οποία ελέγχει αν υπάρχουν ήδη διαμερίσματα με ίδιο όνομα, εισάγει την φωτογραφίες, και θέτει τον ιδιοκτήτη με βάση τον συνδεδεμένο χρήστη. **B)** Η μέθοδος Edit Department η οποία επιτρέπει στον ιδιοκτήτη να κάνει παρόμοιους ελέγχους με την newDepartment και να αλλάζει τα πεδία τα οποία δεν είναι κενά. **Γ)** Η μέθοδος getDepartment επιστρέφει το διαμέρισμα με τις φωτογραφίες του και αυξάνει το count του interaction. **Δ)** Η μέθοδος deletePhoto η οποία επιτρέπει στον ιδιοκτήτη να διαγράψει μία από τις επιπλέον φωτογραφίες **Ε)** Η μέθοδος addPhotos η οποία επιτρέπει στον ιδιοκτήτη να προσθέσει μία επιπλέον φωτογραφία. **ΣΤ)** Η μέθοδος changeMainPhoto η οποία επιτρέπει στον ιδιοκτήτη να αλλάξει

την βασική φωτογραφία του διαμερίσματος. **Z)** Η μέθοδος search η οποία επιστρέφει τα εύκαιρα διαμερίσματα βάσει των ημερομηνιών και των φίλτρων και αυξάνει το count του interaction του χρήστη σε κάθε διαμέρισμα. **H)** Η μέθοδος locationFinder η οποία επιστρέφει τις πιο σχετικές τοποθεσίες **Θ)** Η μέθοδος getDepartments επιστρέφει στον ιδιοκτήτη τα διαμερίσματα του. **I)** Η μέθοδος getRecommended επιστρέφει στον ενοικιαστή τα προτεινόμενα διαμερίσματα. **K)** Η μέθοδος getDepartmentsForExport επιτρέπει στον διαχειριστή να εξαγάγει τα δεδομένα των διαμερισμάτων σε μορφή JSON ή XML.

- Booking Controller: **A)** Η μέθοδος book θέτει μία κράτηση στον συνδεδεμένο ενοικιαστή, ελέγχει αν οι ημερομηνίες είναι αποδεκτές(μετά την σημερινή) και ελέγχει αν υπάρχει ήδη κάποια κράτηση στο διάστημα αυτό για το συγκεκριμένο διαμέρισμα. **B)** Η μέθοδος deleteBooking διαγράφει μια κράτηση από το διαμέρισμα και από τον ενοικιαστή. **Γ)** Οι μέθοδοι getBookings και getBooking επιστρέφουν όλες τις κρατήσεις για τα διαμερίσματα ή τις κρατήσεις για κάποιο διαμέρισμα αντίστοιχα, στον ενοικιαστή. **Δ)** Η μέθοδος BookingExists επιστρέφει αν υπάρχει κάποια κράτηση στο συγκεκριμένο χρονικό διάστημα. **Ε)** Η μέθοδος geBookingsForExport επιτρέπει στον διαχειριστή να εξαγάγει τα δεδομένα των κρατήσεων σε μορφή JSON ή XML.
- Review Controller: **A)** Η μέθοδος reviewDepartment επιτρέπει στον συνδεδεμένο ενοικιαστή αν έχει κάνει κράτηση και αν δεν έχει ξανά σχολιάσει να γράψει μια κριτική για το διαμέρισμα. **B)** Η μέθοδος reviewHost επιτρέπει στον συνδεδεμένο ενοικιαστή αν έχει κάνει κράτηση σε κάποιο διαμέρισμα και αν δεν έχει ξανά σχολιάσει τον ιδιοκτήτη να γράψει μια κριτική για τον ιδιοκτήτη του διαμερίσματος. **Γ)** Οι μέθοδοι getDepartmentReviewsForExport και getDepartmentHostForExport επιτρέπουν στον διαχειριστή να εξαγάγει τα δεδομένα των κριτικών για τα διαμερίσματα και τους οικοδεσπότες(αντίστοιχα) σε μορφή JSON ή XML.
- Message Controller: **A)** Η μέθοδος message θέτει τον συνδεδεμένο χρήστη ως αποστολέα και τον παραλήπτη και αν υπάρχει το id του συσχετιζόμενου μηνύματος το θέτει ως απάντηση στο μήνυμα αυτό, αλλιώς το θέτει ως ερώτηση. Αν το μήνυμα είναι κενό δεν αποθηκεύεται. **B)** Οι μέθοδοι getReceivedMessages και getSentMessages επιστρέφουν στον οικοδεσπότη και στον ενοικιαστή τα μηνύματα που είναι

ερωτήσεις. Γ) Η μέθοδος deleteMessage διαγράφει τα μηνύματα και από τους δύο χρήστες και από το συσχετιζόμενο διαμέρισμα.

***Όσον αφορά τις φωτογραφίες έρχονται με την μορφή MultipartFile και συμπιέζονται/αποσυμπιέζονται με τις αντίστοιχες συναρτήσεις που συμπεριλαμβάνονται στην κλάση Util.**

2.3 Repositories

Υπάρχουν Repository για κάθε μοντέλο, τα οποία επιστρέφουν από την βάση τα δεδομένα(κάνοντας ερωτήματα), τα οποία σχετίζονται με τις προδιαγραφές που ορίζουν οι Controllers

2.4 Security

Το Security θωρακίζει την εφαρμογή κάνοντας authentication και authorization. Η κλάση WebSecurity είναι παρόμοια με αυτήν της διάλεξης απλά στην πρώτη μέθοδο configure έχουν προστεθεί κάποιοι έλεγχοι για την πρόσβαση στα μονοπάτια του Server ανάλογα με τον ρόλο ή/και τον χρήστη. Επίσης υπάρχουν κάποια μονοπάτια όπως το sign up στα οποία έχουν πρόσβαση όλοι οι χρήστες. Επιπλέον, στην συνάρτηση CorsConfigurationSource προστέθηκε η δυνατότητα να δέχεται ο Server όλα τα είδη των αιτήσεων.

Οι κλάσεις JWTAuthenticationFilter και JWTAuthorizationFilter για την διαχείριση του JWT token είναι παρόμοιες με αυτές της διάλεξης απλά στην SuccessfulAuthentication επιστρέφεται και ο χρήστης. Επιπλέον στην doFilterInternal στο SecurityContextHolder αποθηκεύονται οι λεπτομέρειες και οι ρόλοι του χρήστη στην εφαρμογή.

Η κλάση UserDetailsServiceImpl βρίσκει τον χρήστη και θέτει τον ρόλο του στα authorities.

Η κλάση GuardUser λειτουργεί ως guard έτσι ώστε να έχει πρόσβαση ένας χρήστης σε ένα μονοπάτι του Server με βάση το id.

Η κλάση AuthenticationFilter επιστρέφει μήνυμα με σφάλμα 401 αν το login αποτύχει.

3. FRONT-END

Το front-end είναι η διεπαφή με την οποία ο χρήστης μπορεί να χρησιμοποιήσει την εφαρμογή και δημιουργήθηκε με την χρήση της Angular. Προστέθηκαν οι εξής βιβλιοθήκες: Bootstrap, jquery, Ol, Country list js, ngx bootstrap, ngx pagination, @ng-bootstrap/ng-bootstrap.

3.1 Models

Τα models και όσα πεδία χρειάζονται είναι αντίστοιχα με αυτά τα οποία υπάρχουν στο Back-end, απλά προστέθηκε το αρχείο user-details.ts το οποίο περιλαμβάνει το id, το token, τους ρόλους, το accepted flag, ενός χρήστη, για να αποθηκευτεί μέσω του authentication service στο local storage.

3.2 Services

Τα services προωθούν μια αίτηση στον server, επιστρέφουν την απάντηση του σε Observable και θέτουν τις παραμέτρους αν χρειάζονται στα urls.

- Authentication Service: Είναι υπεύθυνο για το log in, log out(βγάζει τα στοιχεία του χρήστη από το local storage) και sing up.
- Booking Service: Είναι υπεύθυνο για τις αιτήσεις που σχετίζονται με τις κρατήσεις.
- Department Service: Είναι υπεύθυνο για τις αιτήσεις που σχετίζονται με τα διαμερίσματα.
- Error Service: Έχουμε ένα behavior subject το οποίο θέτει το error μήνυμα για το οικοδοεσπότη.
- Geocoding Service: Στέλνεται αίτηση στον online server arcgis.com που επιστρέφει τα αποτελέσματα σε μορφή JSON.
- Message Service: Είναι υπεύθυνο για τις αιτήσεις που σχετίζονται με τα μηνύματα.
- Review Service: Είναι υπεύθυνο για τις αιτήσεις που σχετίζονται με τις κριτικές.
- User Service: Είναι υπεύθυνο για τις αιτήσεις που σχετίζονται με τους χρήστες.

*Στα Exports τα οποία χρησιμοποιεί ο admin, θέτουμε στα headers το αντίστοιχο content-type και accept header.

3.3 Components

- App: Είναι υπεύθυνο για την λειτουργικότητα του navigation bar της εφαρμογής και εμφανίζει τα αντίστοιχα κουμπιά ανάλογα με τα στοιχεία του χρήστη αν είναι συνδεδεμένος.
- Admin Home: Είναι η σελίδα του διαχειριστή που φέρνει τους χρήστες από την βάση μέσω του αντίστοιχου service και τους εμφανίζει σελιδοποιημένα. Επίσης μέσω του export button πλοηγείται στο app data export component στο οποίο μπορεί να κατεβάσει σε τα δεδομένα της εφαρμογής.
- Choose Page: Είναι η σελίδα επιλογής αρχικής σελίδας όταν ο χρήστης έχει και τους δύο ρόλους.
- Department Details: Είναι η σελίδα του κάθε διαμερίσματος, στην οποία ανάλογα με τις παραμέτρους(ημερομηνίες, αριθμός ατόμων) που έχουν περαστεί στο url από την αναζήτηση και τον ρόλο, εμφανίζει τις πληροφορίες που αντιστοιχούν στον καθένα μαζί με τα αντίστοιχα κουμπιά. Ο ενοικιαστής αν δεν κατέχει το διαμέρισμα και έχει κάνει κράτηση μπορεί να υποβάλει κριτική για το διαμέρισμα αυτό αλλά και για τον οικοδεσπότη. Επιπλέον μπορεί να στείλει μήνυμα στον ιδιοκτήτη μέσω του κουμπιού Ask Host, αλλά και να δει τα μηνύματα που σχετίζονται με το συγκεκριμένο διαμέρισμα μέσω του κουμπιού View messages. Επίσης μπορεί να κάνει επαλήθευση και υποβολή την κράτηση του διαμερίσματος χρησιμοποιώντας το αντίστοιχο κουμπί. **(Τα παραπάνω κουμπιά ανοίγουν ένα παράθυρο Pop up στη σελίδα)**. Αν είναι οικοδεσπότης και κατέχει το διαμέρισμα του δίνεται η δυνατότητα να αλλάξει την βασική φωτογραφία του διαμερίσματος αλλά και να διαγράψει τις υπόλοιπες φωτογραφίες πατώντας το κουμπί πάνω στην φωτογραφία. Για να προσθέσει φωτογραφίες υπάρχει το αντίστοιχο κουμπί add photos. Επίσης με το κουμπί Edit Department πλοηγείται στο component department-edit για την αλλαγή των στοιχείων του. Τέλος και ο οικοδεσπότης μέσω του κουμπιού View Messages, μπορεί να πάει στην σελίδα των μηνυμάτων για το συγκεκριμένο διαμέρισμα.
- Home: Είναι η σελίδα που περιέχει την μπάρα αναζήτησης για τους ανώνυμους χρήστες αλλά και για τους συνδεδεμένους ενοικιαστές. Στο πεδίο Place έχουμε προσθέσει να εμφανίζονται και προτεινόμενες τοποθεσίες ανάλογα με αυτό που γράφει. Όταν πατηθεί για πρώτη φορά το search θα προστεθούν στο url κάποιες παράμετροι που αφορούν την αναζήτηση και θα εμφανιστούν τα αποτελέσματα της αναζήτησης ταξινομημένα και σελιδοποιημένα μαζί με ένα κουτί στο οποίο ο χρήστης θα μπορεί

να επιλέξει επιπλέον φίλτρα(για να εφαρμοστούν τα φίλτρα θα πρέπει να ξαναπατηθεί το search).

- Booking List: Είναι η σελίδα όπου ο συνδεδεμένος ενοικιαστής μπορεί να πλοηγηθεί από το αντίστοιχο κουμπί στην μπάρα πλοήγησης και να δει τις κρατήσεις του σελιδοποιημένα και να τις ακυρώσει.
- Host-home: Είναι η σελίδα του οικοδεσπότη στην οποία μπορεί να δει τα διαμερίσματα τα οποία έχει βάλει για ενοικίαση σελιδοποιημένα και να περιηγηθεί στις λεπτομέρειες τους πατώντας το αντίστοιχο κουμπί. Ακόμα το add department(που βρίσκεται στο navigation bar) τον ανακατευθύνει στη σελίδα του department-register component για την καταχώρηση νέου διαμερίσματος.
- Messages: Είναι η σελίδα όπου ο συνδεδεμένος ενοικιαστής μπορεί να πλοηγηθεί από το αντίστοιχο κουμπί στην μπάρα πλοήγησης και να δει τα μηνύματα του σελιδοποιημένα, να τα διαγράψει και να τα απαντήσει αν είναι οικοδεσπότης .
- User-Details: Είναι η σελίδα προφίλ ενός χρήστη που περιλαμβάνει τα στοιχεία του. Αν είναι ο ίδιος θα μπορεί να αλλάξει την φωτογραφία προφίλ του με το κουμπί πάνω στην φωτογραφία και να αλλάξει τα στοιχεία του και τον κωδικό του αφού έχει πλοηγηθεί στις κατάλληλες σελίδες.
- Auth/login: Είναι η φόρμα για την σύνδεση του χρήστη.
- Auth/signup: Είναι η φόρμα για την εγγραφή του χρήστη.
- Auth/Helpers: Υπάρχουν δυο interceptors(error,JWT) παρόμοια με την διάλεξη.

***Στις φόρμες πριν την υποβολή τους εμφανίζονται μηνύματα λάθους αν δεν έχουν συμπληρωθεί σωστά και μηνύματα λάθους από τον Server.**

***Οι φωτογραφίες υποβάλλονται μέσω νέου αντικειμένου FormData και αν αποστέλλονται μαζί με κάποιο άλλο αντικείμενο π.χ. διαμέρισμα τότε το αντικείμενο το στέλνουμε σαν File της μορφής blob. Για την εμφάνιση των φωτογραφιών χρησιμοποιείται ο DomSanitizer με κωδικοποίηση της φωτογραφίας σε base64.**

***Όσον αφορά τον χάρτη χρησιμοποιείται η βιβλιοθήκη OpenLayers όπου τον δημιουργούμε και θέτουμε τον marker με βάσει τις**

συντεταγμένες χρησιμοποιώντας και click event για την αλλαγή του (αν χρειάζεται).

3.4 Guards

Υπάρχουν guards τα οποία τα χρησιμοποιούμε για την θωράκιση των σελίδων. Συγκεκριμένα :

- Admin guard: Θωρακίζει τις σελίδες στις οποίες μόνο ο διαχειριστής μπορεί να πλοηγηθεί.
- Booking guard: Μόνο ο ενοικιαστής μπορεί να δει τις κρατήσεις του.
- Choose guard: Μόνο αν ο χρήστης έχει δύο ρόλους μπορεί να μπει στην σελίδα επιλογής
- User guard: Μόνο ο χρήστης με το ίδιο id μπορεί να αλλάξει τα στοιχεία του
- Host guard: Μόνο ο accepted οικοδεσπότης μπορεί να πλοηγηθεί στις σελίδες του οικοδεσπότη, αλλιώς αν δεν τον έχει αποδεχτεί ο διαχειριστής τότε πηγαίνει στο error page(οπού εμφανίζεται το μήνυμα ότι δεν έχει εγκριθεί).

3.5 App module

Περιλαμβάνει όλα τα imports και τα routes

4.Bonus

4.1 Recommedation

Η κλάση αυτή υλοποιεί τον αλγόριθμο των διαφανειών αρχικοποιώντας δυο πίνακες όπου το μέγεθος τους είναι συναρτήσει των χρηστών με την παράμετρο K και των διαμερισμάτων με την παράμετρο K. Στην συνέχεια γίνονται έχουμε βάλει όριο 5000 επαναλήψεων στην περίπτωση που το τετραγωνικό σφάλμα δεν γίνεται μικρότερο του 10^{-3} . Σε αυτές τις επαναλήψεις υπολογίζεται το gradient και γίνονται οι κατάλληλες ενημερώσεις στις άγνωστες τιμές. Τέλος επιστρέφεται ο προβλεπόμενος πίνακας.

4.2 Recommedation Start

Περιλαμβάνει την βασική μέθοδο Rec Init, όπου δημιουργεί τον πίνακα των ratings και δημιουργούνται τα διανύσματα των χρηστών με τον εξής τρόπο: Αν ο χρήστης έχει κάνει έστω και μία κριτική, το διάνυσμα του δημιουργείται από αυτές και αν έχει κάνει κράτηση αλλά όχι κριτική σε κάποιο διαμέρισμα τότε μπαίνει ο μέσος όρος όλων των κριτικών. Αν δεν έχει κάνει ούτε κράτηση, ούτε κριτική τότε οι τιμές του διανύσματος δημιουργούνται με βάση την αλληλεπίδραση του με τα διαμερίσματα (Την πρώτη φορά δίνεται η τιμή 3 επειδή είναι ανάμεσα στο 1 και στο 5 ως ουδέτερη τιμή και κάθε φορά που έχει την αντίστοιχη αλληλεπίδραση προστίθεται στην τιμή αυτή μια μικρή τιμή 0.05 μέχρι να φτάσει το 5). Τέλος, αν δεν υπάρχει καμία αλληλεπίδραση δεν επιστρέφεται κανένα προτινόμενο.

4.3 Εισαγωγή δεδομένων από Dataset

Υπάρχει συνάρτηση η οποία συσχετίζει το μοντέλο του dataset με το δικό μας ανοίγοντας και διαβάζοντας τα csv αρχεία και τοποθετώντας τα χρήσιμα χαρακτηριστικά στο μοντέλο μας. (Για την εισαγωγή των δεδομένων στην βάση, τοποθετήσαμε τα csv αρχεία στο resources.)

Οι κλήσεις των παραπάνω βρίσκονται σε σχόλιο και καλούνται στο ξεκίνημα με λίγα reviews γιατί είναι υπερβολικά μεγάλο το αρχείο.

5. ΕΠΙΛΟΓΟΣ

Σημαντικές δυσκολίες που είχαμε ήταν το authentication και authorization μέσω servlets, η εισαγωγή, αποθήκευση και παρουσίαση των φωτογραφιών, η εισαγωγή του χάρτη μαζί με το geocoding και αρκετά προβλήματα με το Lombok λόγω του hash Code. Επίσης αρκετό χρόνο αφιερώσαμε στην μορφοποίηση των σελίδων μέσω css.