**A blue background with black text

AI-generated content may be incorrect.Domain - Model v 0.3**

**GitHub link :** <https://github.com/LampisGiannelis/HomeLink.git>

|  |  |  |
| --- | --- | --- |
| Ονοματεπώνυμο | Έτος | ΑΜ |
| Χαράλαμπος Γιαννέλης | 4ο | 1093341 |
| Άννα Ζαχαράκη | 7ο | 1067507 |
| Ηλίας Κολιτσιδάκης | 6o | 1072597 |
| Σταύρος Κολιτσιδάκης | 5o | 1090063 |
| Ευαγγελία Σταυροπούλου | 7ο | 1067476 |

# **A diagram of a company AI-generated content may be incorrect.**

# **Περιγραφή Κλάσεων**

## **A. Users & External Systems**

1. **User**: Αναπαριστά κάθε χρήστη του συστήματος. Περιέχει βασικά στοιχεία ταυτοποίησης (όνομα, email, τηλέφωνο) και ρόλους (Owner, Tenant).

**Attributes:**

- id: UUID

- name: String

- email: String

- phone: String

- profileStats: UserProfileStats

- accountVerification: AccountVerification

- role: enum { Owner, Tenant }

- ratingProfile: RatingProfile

- wallet: Wallet

**Methods:**

+ getCompletedExperiences()

+ hasReviewed(userId, experienceId): Boolean

+ isVerified(): Boolean

+ requestVerification(): VerificationResult

+ getAvailability(): List<TimeSlot>

+ getWallet(): Wallet

+ getPaymentMethods(): List<PaymentMethod>

+ addPaymentMethod(method: PaymentMethod)

+ getMyListings(): List<Listing>

+ getRentalInterests(): List<RentalInterest>

+ createRentalRequest(tenants, terms)

+ finalizeRental(request: RentalInitiationRequest)

+ sendMessage(chat, content): Message

+ receiveNotification(notification): void

1. **Owner**: Εξειδικευμένος ρόλος του User που μπορεί να δημιουργεί και να διαχειρίζεται αγγελίες (Listing).

**Attributes:**

- id: UUID

- name: String

- email: String

- phone: String

- activeListings: List<Listing>

**Methods:**

+ createListing(): Listing

+ canCreateNewListing(): Boolean

+ getActiveListingsCount(): Int

+ getAvailability(): List<TimeSlot>

1. **Tenant**: Εξειδικευμένος ρόλος του User που αναζητά κατοικίες, υποβάλλει δηλώσεις ενδιαφέροντος και πραγματοποιεί πληρωμές ενοικίου.

**Attributes:**

- id: UUID

- name: String

- email: String

- preferences: UserPreferences

- wallet: Wallet

- activeContracts: List<RentalContract>

- trustScore: TrustScore

- verification: AccountVerification

**Methods:**

+ search(SearchCriteria)

+ applySavedPreferences(SearchCriteria)

+ submitInterest(Listing)

+ selectAvailability(List<TimeSlot>)

+ getCurrentRentContract()

+ getWalletBalance(): Decimal

+ acceptRentalRequest(RentalInitiationRequest):

+ uploadDocuments(List<SubmittedDocument>)

+ recalculateReliability()

+ viewAppointments()

+ declareArrival(appointment, currentLocation): Boolean

1. **Identity Provider**: Αντιπροσωπεύει την εξωτερική υπηρεσία ψηφιακής ταυτοποίησης για έλεγχο εγγράφων.

**Methods:**

+ verifyDocument(userId)

1. **Map Service**: Εξωτερική υπηρεσία χαρτογράφησης για αναζήτηση διευθύνσεων και απόδοση συντεταγμένων.

**Methods:**

+ suggestAddress(): List<AddressSuggestion>

+ getCoordinates(): (latitude, longitude)

+ getAddressSuggestions()

+ getCoordinatesForAddress()

+ renderMapMarker(latitude, longitude):

+ getMapMarkers(List<Listing>)

+ searchRegionSuggestions(String)

+ getDirections(from,to):

+ requestCurrentLocation(): Location

1. **Payment Provider**: Εξωτερικός πάροχος πληρωμών για την επικύρωση στοιχείων κάρτας και εκτέλεση συναλλαγών.

**Methods:**

+ processPayment(amount, method):

+ authorizeTopUp(amount, method)

+ verifyCard(CardDetails)

1. **Email Service**: Υπηρεσία για την αποστολή emails προς τους χρήστες

**Methods:**

+ sendEmail(to, subject, body): Boolean

1. **Database** : οντότητα διαχείρισης βάσης δεδομένων.

**Methods:**

+ saveListing(listing)

+ fetchUserListings(userId)

+ getListingsByOwner(ownerId)

+ saveTemporaryLocation(ownerId, location)

+ getListingById(id): Listing

+ getUserAvailability(userId): UserAvailability

+ saveRentalInterest(interest): void

+ saveRentPayment(payment): void

+ saveReceipt(receipt): void

+ updateContractStatus(contract): void

+ saveTopUpTransaction(TopUpTransaction): void

+ updateWalletBalance(userId, newBalance): void

+ savePaymentMethod(method): void

+ linkMethodToWallet(walletId, methodId): void

+ getInterestsForOwner(ownerId)

+ saveRentalRequest(RentalInitiationRequest): void

+ createRentalNotifications(RentalInitiationRequest): void

+ updateRentalRequest(RentalInitiationRequest): void

+ storeDocuments( List<SubmittedDocument>): void

+ storeNotification(notification): void

+ updateListingStatus(listing): void

+ updateRentalStatus(RentalContract): void

+ deleteRelatedInterests(listingId): void

+ storeContract(RentalContract): void

+ storeNotifications(List<Notification>): void

+ updateContractStatus(RentalContract): void

+ updateListingStatus(listn: Listing): void

+ insertReliabilityScore( ReliabilityScore): void

+ saveNotifications(List<Notification>): void

+ saveMessage(Message): void

+ getChatByContract(contractId):

+ getMessages(chatId): List<Message>

+ storeChatNotification(ChatNotification): void

+ getAppointmentsForUser(userId)

+ updateAppointmentStatus(appId, status): void

+ storeNotification(Notification): void

## **B. Core Use Case Entities**

1. **Listing**: Η αγγελία που δημοσιεύει ο Owner. Περιλαμβάνει τιμή, εμβαδόν, όροφο, αριθμό δωματίων, κατάσταση (ενεργή, ενοικιασμένη), κ.λπ.

**Attributes:**

- id: UUID

- owner: Owner

- property: Property

- location: Location

- photos: List<Photo>

- price: Decimal

- area: Float

- floor: Int

- numberOfRooms: Int

- isShared: Boolean

- maxRoommates: Int?

- status: enum { Active, Rented, Inactive }

- createdAt: DateTime

**Methods:**

+ validateRoommateCount(): Boolean

+ attachPhotos(List<Photo>): void

+ publish(): Boolean

+ setLocation(Location): void

+ isDuplicateLocation(Location): Boolean

+ isMatch(SearchCriteria): Boolean

+ getDetails(): Listing

+ hasInterestFrom(User): Boolean

+ isValidTenantCount(): Boolean

+ markAsRented(): void

+ removeRelatedInterests(): void

+ markAsUnavailable(): void

1. **Experience**: Μια εμπειρία ενοικίασης ή συγκατοίκησης που έχει πραγματοποιηθεί, συνδεδεμένη με μία συγκεκριμένη αγγελία.

**Attributes:**

- id: UUID

- listing: Listing

- participants: List<User>

- startDate: Date

- endDate: Date

- type: enum { Rental, CoLiving }

- isCompleted: Boolean

- rolePerUser: Map<User, String>

**Methods:**

+ isEligibleForReview()

+ getUserExperienceCount(userId)

1. **Account Verification**: Η διαδικασία πιστοποίησης/ταυτοποίησης του χρήστη (email, τηλέφωνο, έγγραφα).

**Attributes:**

- id: UUID

- user: User

- status: enum { Pending, Verified, Rejected }

- verifiedAt: DateTime

- emailVerified: Boolean

- phoneVerified: Boolean

- identityCheck: IdentityCheck

**Methods:**

+ isVerified(): Boolean

+ checkVerificationStatus(userId)

+ checkEmail(): Boolean

+ checkPhone(): Boolean

+ runFullVerification(): Boolean

1. **Rental Interest**: Δήλωση ενδιαφέροντος του ενοικιαστή για μια συγκεκριμένη αγγελία, με τυχόν στοιχεία διαθεσιμότητας.

**Attributes:**

- id: UUID

- listing: Listing

- interestedTenant: Tenant

- preferredSlots: List<TimeSlot>

- finalAppointment: Appointment

- createdAt: DateTime

**Methods:**

+ getInterestedTenants(Listing)

+ isDuplicate(): Boolean

+ save(): void

+ createAppointment(TimeSlot)

1. **Rental Initiation Request**: Προσωρινό αίτημα μίσθωσης που ξεκινά ο ιδιοκτήτης, απευθυνόμενο σε υποψήφιους ενοικιαστές (κατάσταση: «Εκκρεμεί»).

**Attributes:**

- id: UUID

- listing: Listing

- tenants: List<Tenant>

- terms: RentalTerms

- createdBy: Owner

- status: enum { Pending, Accepted, Rejected }

- createdAt: DateTime

- submittedDocuments: List<SubmittedDocument>

**Methods:**

+ sendNotifications(): void

+ save(): void

+ markAsAcceptedBy(Tenant): void

+ attachDocuments(List<SubmittedDocument>): void

+ isVerifiedForAllTenants(): Boolean

+ markAsReady(): void

+ convertToContract(): RentalContract

1. **Appointment :** Κλάση η οποία διαχειρίζεται το ραντεβού για την επίσκεψη κατοικίας.

**Attributes:**

- id: UUID

- listing: Listing

- tenant: Tenant

- owner: Owner

- scheduledAt: DateTime

- isConfirmed: Boolean

- status: enum { Scheduled, Visited }

- confirmedArrival: Boolean

**Methods:**

+ confirm(): void

+ toEmailContent(): String

+ getDetails()

+ markVisited(): void

1. **Rent Payment**: Συναλλαγή ενοικίου που αντιστοιχεί σε συγκεκριμένο μήνα και συνδέεται με σύμβαση μίσθωσης.

**Attributes:**

- id: UUID

- month: Date

- amount: Decimal

- status: enum { Pending, Paid, Overdue }

- transaction: Transaction

- dueDate: Date

- paymentDate: Date

**Methods:**

+ markAsPaid(): void

+ isDue(): Boolean

+ wasPaidOnTime(): Boolean

1. **Top Up Transaction**: Συναλλαγή ανανέωσης υπολοίπου στο Wallet (π.χ. κατάθεση χρημάτων μέσω κάρτας).

**Attributes:**

- id: UUID

- user: User

- amount: Decimal

- method: PaymentMethod

- status: enum { Pending, Success, Failed }

- timestamp: DateTime

**Methods:**

+ submit(): PaymentAuthorization

+ finalize(): void

1. **Chat**: Περιβάλλον συνομιλίας που συσχετίζεται με μία ενεργή μίσθωση (διατηρεί ιστορικό και συμμετέχοντες).

**Attributes:**

- id: UUID

- rentalContract: RentalContract

- participants: List<User>

- messages: List<Message>

**Methods:**

+ getMessageHistory(): List<Message>

+ addMessage(Message): void

+ hasParticipant(User): Boolean

## **C. Identity & Property Details**

1. **Identity Check**: Καταγράφει τα αποτελέσματα ελέγχου επίσημων εγγράφων (π.χ. ταυτότητα, ΑΦΜ) για έναν User.

**Attributes:**

- user: User

- documentType: String

- documentSubmitted: Boolean

- result: VerificationResult

**Methods:**

+ requestVerificationFromProvider(): VerificationResult

+ sendToProvider(): VerificationResult

1. **Verification Result**: Η απάντηση του εξωτερικού φορέα ταυτοποίησης, επιβεβαιώνοντας ή απορρίπτοντας τα έγγραφα του χρήστη.

**Attributes:**

- success: Boolean

- message: String

- checkedAt: DateTime

- timestamp: DateTime

1. **Property**: Περιγράφει το ακίνητο (τύπος κατοικίας, έτος κατασκευής, θέρμανση, ενεργειακή κλάση, δυνατότητα συγκατοίκησης).

**Attributes:**

- id: UUID

- type: enum { Apartment, Studio, House, etc. }

- yearBuilt: Int

- heatingType: String

- energyClass: String

- allowsSharing: Boolean

**Methods:**

+ isCompatibleWithRooms(roomCount, maxRoommates)

1. **Location**: Η γεωγραφική θέση του ακινήτου (διεύθυνση, πόλη, συντεταγμένες).

**Attributes:**

- id: UUID

- address: String

- city: String

- coordinates: (latitude: Float, longitude: Float)

- latitude: Float

- longitude: Float

**Methods:**

+ validateCoordinates(): Boolean

+ fetchFromMapService(String): Location

+ isWithinRadiusOf(otherLocation, radiusInMeters)

+ setFromCoordinates(address, latitude, longitude)

+ validateWithMapService(): Boolean

+ isCloseTo(other: Location, radius: Float): Boolean

1. **Photo**: Εικόνα που συνδέεται με ένα Listing. Περιλαμβάνει πληροφορίες αρχείου (URL, όνομα, τύπο).

**Attributes:**

- id: UUID

- fileName: String

- fileType: String

- fileSize: Int

- url: String

**Methods:**

+ isValidFormat(): Boolean

+ isUnderSizeLimit(): Boolean

## **D. Search & Preferences**

1. **Search Criteria**: Φίλτρα και κριτήρια που θέτει ο χρήστης κατά την αναζήτηση (περιοχή, τύπος, συγκατοίκηση, κ.λπ.).

**Attributes:**

- region: String

- propertyType: String

- allowsSharing: Boolean

- minPrice: Decimal

- maxPrice: Decimal

**Methods:**

+ isValid(): Boolean

+ match(listing: Listing): Boolean

1. **User Preferences**: Αποθηκευμένο προφίλ προτιμήσεων (φίλτρα) που μπορεί να εφαρμόζεται στην αναζήτηση κατοικίας.

**Attributes:**

- user: Tenant

- preferredRegions: List<String>

- preferredTypes: List<String>

- allowsSharing: Boolean

**Methods:**

+ applyTo(criteria: SearchCriteria): SearchCriteria

+ isAvailable(): Boolean

1. **Search Result**: Τα αποτελέσματα μιας αναζήτησης κατοικιών, συμπεριλαμβανομένων των αγγελιών που πληρούν τα κριτήρια.

**Attributes:**

- listings: List<Listing>

- totalResults: Int

- suggestions: List<AddressSuggestion>

- locationMarkers: List<(latitude, longitude)>

**Methods:**

+ generateMarkers(): List<>

+ sortBySuitability(): void

1. **Suitability Score**: Υπολογίζει δείκτη καταλληλότητας για κάθε αγγελία, λαμβάνοντας υπόψη τις προτιμήσεις του χρήστη.

**Attributes:**

-listing: Listing

- score: Float

**Methods:**

+ calculate(userPreferences, listing): Float

## **E. Experiences & Reviews**

1. **Experience History**: Η λίστα εμπειριών ενός χρήστη, μαζί με τις σχετικές αγγελίες, φιλτραρισμένες χρονικά.

**Attributes:**

- user: User

- experiences: List<Experience>

**Methods:**

+ getRecentCompletedExperiences(): List<Experience>

+ filterValidForReview(): List<Experience>

1. **Review**: Αναπαριστά την αξιολόγηση που υποβάλλει ένας χρήστης για έναν άλλο, με αστέρια, χαρακτηριστικά και σχόλια.

**Attributes:**

- id: UUID

- reviewer: User

- reviewedUser: User

- experience: Experience

- rating: Int (1-5)

- characteristics: List<String>

- comment: String?

- visibleOnProfile: Boolean

- createdAt: DateTime

**Methods:**

+ validate(): Boolean

+ submit(): Boolean

1. **Review Form**: Η φόρμα που συμπληρώνει ο χρήστης κατά την αξιολόγηση (επιλογή αστεριών, χαρακτηριστικών, σχόλιο).

**Attributes:**

- stars: Int

- selectedCharacteristics: List<String>

- comment: String?

- visibility: Boolean

**Methods:**

+ isComplete(): Boolean

1. **Rating Profile**: Το προφίλ αξιολογήσεων ενός χρήστη, με μέσο όρο, διακύμανση, χαρακτηριστικά και ρυθμίσεις ορατότητας.

**Attributes:**

- user: User

- averageRating: Float

- totalReviews: Int

- visible: Boolean

**Methods:**

+ updateWithNewReview(Review): void

+ calculateStats(): void

1. **User Profile Stats**: Στατιστικά για τις αξιολογήσεις (average rating, variance, χαρακτηριστικά), χρήσιμα για υπολογισμό σκορ.

**Attributes:**

- averageRating: Float

- totalReviews: Int

- ratingVariance: Float

- mostSelectedCharacteristics: List<String>

- frequentCharacteristics: List<String>

**Methods:**

+ updateAfterReview(review: Review): void

+ calculateVariance(): Float

+ getFrequentTerms(): List<String>

1. **Trust Score**: Γενικός βαθμός εμπιστοσύνης ενός χρήστη, βασισμένος σε αξιολογήσεις, εμπειρίες και συνέπεια.

**Attributes:**

- user: User

- value: Float

- label: String

**Methods:**

+ calculate(UserProfileStats, experienceCount): Float

+ classify(): String

1. **Reliability Score**: Δείκτης φερεγγυότητας για την οικονομική συνέπεια του ενοικιαστή (π.χ. έγκαιρες πληρωμές).

**Attributes**:

- tenant: Tenant

- onTimePayments: Int

- totalPayments: Int

- score: Float

**Methods:**

+ updateScore(): void

## **F. Payments & Financials**

1. **Wallet**: Το ψηφιακό πορτοφόλι του χρήστη, περιλαμβάνοντας υπόλοιπο και συνδεδεμένους τρόπους πληρωμής.

**Attributes:**

- id: UUID

- user: User

- balance: Decimal

- defaultPaymentMethod: PaymentMethod

- paymentMethods: List<PaymentMethod>

**Methods:**

+ hasSufficientBalance(amount): Boolean

+ deduct(amount): void

+ addFunds(amount): void

+ getBalance(): Decimal

+ hasValidMethod(): Boolean

+ addPaymentMethod(method): void

+ getActiveMethods(): List<PaymentMethod>

1. **Payment Method**: Αναπαριστά έναν καταχωρημένο τρόπο πληρωμής (κάρτα, IBAN) που συνδέεται με το Wallet.

**Attributes:**

- id: UUID

- type: enum { Card, BankAccount }

- providerReference: String

- isActive: Boolean

- cardDetails: CardDetails?

**Methods:**

+ validateWithProvider(): PaymentAuthorization

1. **Card Details**: Πληροφορίες κάρτας (αριθμός, ημερομηνία λήξης, CVV) που ο χρήστης εισάγει προσωρινά πριν την οριστική αποθήκευση.

**Attributes:**

- cardNumber: String

- holderName: String

- expiryDate: Date

- cvv: String

**Methods:**

+ isValid(): Boolean

1. **Payment Authorization**: Διαδικασία επικύρωσης της συναλλαγής ή της κάρτας μέσω του Payment Provider (επιτυχία ή αποτυχία).

**Attributes:**

- status: enum { Success, Failed }

- authorizedAt: DateTime

- success: Boolean

- timestamp: DateTime

- providerMessage: String

**Methods:**

+ authorize(amount: Decimal, method: PaymentMethod)

1. **Transaction**: Γενική τραπεζική συναλλαγή (μοναδικό ID, ημερομηνία, κατάσταση).

**Attributes:**

- id: UUID

- amount: Decimal

- status: enum { Completed, Failed }

- relatedTo: enum { TopUp, RentPayment }

- createdAt: DateTime

**Methods:**

+ finalize(): void

1. **Payment Receipt**: Το αποδεικτικό πληρωμής (π.χ. σε PDF), που παράγεται μετά από επιτυχή συναλλαγή.

**Attributes:**

- id: UUID

- transaction: Transaction

- filePath: String

- generatedAt: DateTime

**Methods:**

+ generate(): PaymentReceipt

1. **Payment Form** : Προσωρινό αντικείμενο φόρμας που συλλέγει ποσό και μέθοδο πληρωμής πριν την υποβολή

**Attributes:**

- amount: Decimal

- selectedMethod: PaymentMethod

- cardNumber: String

- holderName: String

- expiryDate: Date

- cvv: String

**Methods:**

+ validate(): Boolean

+ isValid(): Boolean

+ isComplete(): Boolean

+ toCardDetails(): CardDetails

## **G. Rental Contracts & Documents**

1. **Rental Contract**: Αναπαριστά τη συμφωνία μίσθωσης μεταξύ ιδιοκτήτη και ενοικιαστή, με ημερομηνίες, όρους και κατάσταση (ενεργή/ολοκληρωμένη).

**Attributes:**

- id: UUID

- tenant: Tenant

- owner: Owner

- monthlyAmount: Decimal

- startDate: Date

- endDate: Date

- rentPayments: List<RentPayment>

- terms: RentalTerms

- participants: List<User>

- listing: Listing

- contractPDF: ContractPDF

**Methods:**

+ isActive(): Boolean

+ getCurrentDue(): RentPayment

+ registerPayment(payment: RentPayment)

+ finalizeIfAllAccepted(): void

+ generatePDF(): ContractPDF

+ activate(): void

+ hasEnded(currentDate: Date): Boolean

+ completeContract(): void

+ getChat(): Chat?

1. **Rental Terms**: Οι όροι μίσθωσης που ορίζει ο ιδιοκτήτης (διάρκεια, τιμή, τυχόν προκαταβολή).

**Attributes:**

- rentAmount: Decimal

- durationInMonths: Int

- deposit: Decimal

**Methods:**

+ isValid(): Boolean

1. **Submitted Document**: Έγγραφο (ταυτότητα, ΑΦΜ, αποδεικτικό εισοδήματος) που υποβάλλει ο χρήστης για έλεγχο εγκυρότητας.

**Attributes:**

- id: UUID

- type: enum { ID, TaxNumber, IncomeProof }

- fileName: String

- fileSize: Int

- fileType: String

- orderIndex: Int

- status: enum { Uploaded, Validated, Failed }

**Methods:**

+ validateFormat(): Boolean

+ isVerified(): Boolean

1. **Contract PDF**: Το παραγόμενο αρχείο PDF του συμβολαίου.

**Attributes:**

- filePath: String

- generatedAt: DateTime

**Methods:**

+ generateFrom(contract: RentalContract): ContractPDF

+ export(): void

## **H. Notifications & Communication**

1. **Notification**: Ειδοποίηση που δημιουργείται στο σύστημα (π.χ. νέα αίτηση, πληρωμή, λήξη μίσθωσης).

**Attributes:**

- recipient: User

- message: String

- recipient: User

- type: enum { ChatMessage, Rental, Payment }

- status: enum { Read, Unread }

- createdAt: DateTime

1. **Rental Notification**: Εξειδίκευση Notification για αιτήματα ή ενημερώσεις που αφορούν μία μίσθωση (π.χ. έναρξη, αποδοχή).

**Attributes:**

- recipient: Tenant

- message: String

- type: enum { RentalRequest }

- createdAt: DateTime

**Methods:**

+ send(): void

1. **Message**: Κάθε μήνυμα σε ένα Chat, περιλαμβάνοντας αποστολέα, περιεχόμενο, timestamp.

**Attributes:**

- id: UUID

- sender: User

- chat: Chat

- content: String

- timestamp: DateTime

- isRead: Boolean

**Methods:**

+ markAsRead(): void

1. **Chat Notification**: Ειδική ειδοποίηση για νέο μήνυμα στην τρέχουσα συνομιλία.

**Attributes:**

- chat: Chat

- recipient: User

- messagePreview: String

- createdAt: DateTime

**Methods:**

+ send(): void

1. **User Availability**: Αποθηκεύει τα χρονικά διαστήματα (time slots) ενός χρήστη (ιδιοκτήτη/ενοικιαστή) για ραντεβού ή επικοινωνία.

**Attributes:**

- user: User

- timeSlots: List<TimeSlot>

**Methods:**

+ getCommonSlots(other: UserAvailability): List<TimeSlot>

+ addSlots(slots: List<TimeSlot>): void