**A blue background with black text

AI-generated content may be incorrect.Class - Diagram v 1.0**

**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 |

Για λόγους Ευκρίνειας το **Class-Diagram v1.0** βρίσκεται το link του GitHub παραπάνω. Επίσης συνίσταται να δείτε όλες τις κλάσεις και τα attributes από το **domain\_classes.xlsx** αρχείο το οποίο παρουσιάζει αναλυτικά τα πάντα.

|  |  |  |
| --- | --- | --- |
| **Manager Classes** | | |
| **CreateListingManager** | **LocationManager** | **SearchHousingManager** |
| initiateListingCreation() | triggerLocationEntry(listingID) | initiateSearchHousing() |
| checkListingLimit(activeListings) | initilizeLocationModule() | loadUsersPreferences(userID) |
| cancelListingCreation() | handleSearchQuery(query) | applyPreferences(criteria,preferences) |
| validateRoomateCompatibility() | handleSelection(suggestion) | sortListingsByScore() |
| validatePhotos(photos) | onMarkerMoved(newCoordinates) | continueWithoutPreferences() |
|  | onLocationConfirm(coordinates) | executeSearch(preferences) |
|  | checkNearbyListings(listings,coordinates) | broadenSearch() |
|  | terminateLocationDeclaration() |  |
|  |  |  |
| **ExperienceManager** | **IdentityVerificationManager** | **RentalInterestManager** |
| fetchUserExperiences(userID) | initiateVerification(userID) | fetchFullListingDetails(listingID) |
| filterValidExperiences() |  | sendOwnerAvailabilityEmail() |
| getParticipants(ListingID) |  | sendOwnerAvailabilityEmail() |
| hasAlreadyEvaluated(participantID,listingID) |  | createInterestEmail() |
|  |  | createIntersetObject() |
| **RentPaymentManager** | **PaymentMethodManager** | **TopUpManager** |
| initiatePaymentFlow(userID) | triggerAddPaymentMethod() | triggetTopUpTransaction() |
| computeReliabilityScore(delayStats) | triggerAddPaymentMethod() | initiateTopUpFlow(userID) |
|  | notifyCancellation() | validateTopUpForm() |
|  | navigateToCallerUseCase() |  |
| **RentalRequestManager** | **RentalApplicationManager** | **FinalizeRentalManager** |
| validateTenantSelection(listing,user) | loadPendingRequests(user) | fetchtTemporaryRentals(owner) |
| startRentalProcedure() | proceedToDocumentUpload() | terminateContractCreation() |
| startRentalProcess() | handleTermsRejection() | prefillWithValidData(details) |
|  | onUpload(documentSet) |  |

|  |  |  |
| --- | --- | --- |
| **ChatManager** | **AppointmentMananer** | **Rental Termination** |
| startChat(user) | loadScheduledVisits() | initiateRentalTermination(rental) |
| getChatSession() | handleVisitSelection() | handleReturnToCallingUseCase() |
| isRentalExpired(rental) | requestLocationPermission() | ternminateRentalTermination() |
| relateChatMessage(user,rental,text) | confirmArrival() |  |

|  |  |  |
| --- | --- | --- |
| **SIMPLE CLASSES** | | |
| **Listing** | **RentalInterest** | **Message** |
| filterActiveListings() | getRentalInterests(listingID) | createMessage("message") |
| updatelistingStatus(listingID) | getListingInterests(listings) | createErrorMessage("message") |
| getListingsForOwner(ownerID) | getUserIdsFromInterests(users) | createPromptMessage("promt") |
| finalizeListingCreation(listingID) | deleteListingInterests() | createSuccessMessage("Message") |
| fetchListings(preferences) | deleteAssociatedInterests() | createWarningMessage("message") |
| computeSuitabilityScore() |  |  |
| getListingsFromContract(contracts) |  |  |
| getUserListings(userID) |  |  |
| getTypeAndCapacity(listing) |  |  |
| markAsAnavailable(listing) |  |  |
| setListingUnavailable(listing) |  |  |

|  |  |  |
| --- | --- | --- |
| **Photo** | **Location** | **UserPreferences** |
| attachPhotos(photos) | createCoordinates(coordinates) | getUserPreferences(userID) |
| **MapService** | **RentalContract** | **Review** |
| requestMapInit() | getUserRentalContracts(userID) | getExistingEvaluation() |
| getSuggestions(query) | filterCompletedRentalContracts(userID) | validateRequiredFileds() |
| getCoordinates(address) | getActiveContract(userID) | createEvaluation() |
| validateCoordinates(newCoordinates) | getPaymentDeadline() | getRecievedEvaluations(userID) |
| fetchGeolocation(listings) | isValid(contractID) |  |
| generateMarkers() | generateRentalContract(rental) |  |
| requestRouteToProperty() | generateContractPDF() |  |
| getCurrentLocation() | isContractEnded(rental) |  |
| calculateRoute(currentLoc,propertyLoc) | getInvolvedRenters() |  |
| validateArrivalLocation(currentLoc,propertyLoc) | getRental(user) |  |
|  | getRentalParticipants() |  |
| **UserProfile** | **TrustScore** | **EmailService** |
| updateReputation(score,tags) | markAsNewUser(userID) | createCancellationaEmail() |
| isEmailVerified(email) | updateTrustScore(userID,"N/A") | sendEmail(to: ownerEmail) |
| isPhoneVerified(email) | calculateAverageScore(userID) | sendInterestEmail() |
| getUserProfile(userID) | calculateVariance(evaluations) |  |
|  | analyzeTraits(evaluations) |  |
|  | computeTrustScore(evaluations) |  |
|  | getTrustScore(users) |  |
| **UserAvailability** | **PaymentHistory** | **ReceiptPDF** |
| getOwnerAvailability() | getPaymentHistory(contractID) | fetchReceiptForMonth(contractID) |
| createTimeslots() | checkForOutstandingPayments() | createPDF(transaction) |
| compareWith(ownerTimeSlots) | isOwedCurrentMonthOnly() |  |
|  | calculateDelayStats() |  |
|  | updatePyamentHistory() |  |
|  | checkTimeliness(rental) |  |
|  | calculateDelays(rental) |  |

|  |  |  |
| --- | --- | --- |
| **Transaction** | **PaymentMethod** | **Notification** |
| generateTransaction() | hasValidPaymentMethod(userID) | createRentalRequestNotification(rental,users) |
| generateTransactionID() | triggerAddPayment() | notifyAllPartisipants() |
| logFailure(userID,error) | getPaymentMethods(userID) | createChatNotifications(message) |
|  | savePaymentMethod(userID,cardDetails) | sendPush() |
|  |  | createArrivalNotification(to:owner) |
| **RentalTerms** | **Marker** | **Wallet** |
| getPendingRentalRequests() | None | getDefaultPaymentMethod(userID) |
| getPendingRentalRequests() |  | getBalance(userID) |
| markForFinalization(RentalTerms rental) |  | debutAmount(userID,Amount) |
| getAllRentalDetails(rental) |  | getWallet(userID) |
|  |  | updateBalance(userID,amount) |
| **ReliabilityScore** | **ChatSession** | **ChatMessage** |
| updateReliabilityScore() | getSessionMesages() | createNewChatMessage() |
| increaseReliabilityScore() | chatExistsForRental(rental) |  |
|  | createChatSession(rental,participants) |  |
|  | addParticipants(participants) |  |
|  | getSessionMessages() |  |

|  |  |  |
| --- | --- | --- |
| **SCREEN CLASSES** | | |
| **MainScreen** | **CreateListingScreen** | **LocationScreen** |
| onCreateListingClicked() | displayCreateListingScreen() | setLocationTriggered() |
| returnToMainScreen() | showArchiveOptions(List<Listing>) | displaySetLocationScreen() |
| onEvaluateExperienceClicked() | archiveListing(listingID) | inputSearchQuery() |
| getUserPhone(userID) | displayMessage() | displayMessage() |
| checkExistingInterest(userID,listingID) | cancelListingArchive() | displaySuggestions() |
| getInterestDetails(userID,listingID) |  | retrySearch() |
| selectPayRent() |  | selectSuggestion(suggestion) |
| displayMessage() |  | placeMarkers(coordinates) |
| selectTopUpTransaction() |  | confirmLocations() |
| onInterestDeclarationsClicked() |  | dragMarker() |
| onAcceptRentalClicked() |  | cancelLocationDeclaration() |
| onFinalizeRentalClicked() |  |  |
| onChatClicked() |  |  |
| onAppointmentsClicked() |  |  |
| **ExperienceHistoryScreen** | **UserAvailabilityScreen** | **TopUpScreen** |
| displayMessage() | showAvailabilitySelectionScreen() | displayMessage() |
| displayEperienceList() | submitAvailability(timeslots) | selectAlternativeMethod() |
| onParticipantSelected(participantID) | displayCommonTimeslots() |  |
|  | selectCommonTimeslot() |  |
|  | displayMessage() |  |

|  |  |  |
| --- | --- | --- |
| **SearchHousingScreen** | **RentPaymentScreen** | **RentalRequestScreen** |
| displaySearchForm() | displayReceiptDetails(receipt) | displayDeclarations(interests,trustScores)) |
| displayMessage() | terminateRentalPayment() | selectUsers(users) |
| fillSearchCriteria() | addPaymentMethod() | displayValidationResult() |
| acceptFilters() | showPaymentSummary() |  |
| declineFilters() | topUpTranasaction() | **AppointmentScreen** |
| requestBroaderSearch() |  | displayVisits(visitDetails) |
| displaySearchResults(listings) |  | grantPermission() |
| displayMap(markers) |  | manualArrivalSelected() |
| cancelExistingInterest(listingID) |  | selectArrival() |
| onListingSelected(listingID) |  |  |
| displayDetailedListing(listingDetails) |  |  |
| submitInterest(listingID) |  |  |
| displayExistingInterest(interestDetails) |  |  |
|  |  |  |
|  |  |  |
| **RentalApplicationScreen** | **FinalizeRentalScreen** | **ChatScreen** |
| displayRentalTerms(rentalRequests) | showFinalizeOptions() | disableMessageInput() |
| uploadDocuments() | chooseEditContract() | displayChatMessages() |
|  | dislayContractpReview() | displayMessage() |
|  | cancelContractFinalization() | resendMessage(text) |
|  | submitContract(RentalTerms rental) | sendMessage(text) |
|  | selectRentalForFinalization() | displayNewMessage() |

|  |  |  |
| --- | --- | --- |
| **CreateListingForm** | **UploadPhotoForm** | **SearchHousingForm** |
| validateRequiredFields() | removeInvalidphotos() | validateSearchForm() |
| removeInvalidphotos() | uploadPhotos () |  |
| **ReviewForm** | **TopUpForm** | **PaymentMethodForm** |
| createEvaluationForm() | displayTopUpForm(balance,methods) | displayPaymentForm() |
| displayReadOnlyEvaluation(evaluation) |  | fillPaymentMethodForm() |
| setStars(), setTgs(), setCommets(),setVisibility() |  | cancelPaymentMethod() |
| submitEvaluation() |  | highlightFileds() |
| highlightFields() |  |  |
| returnToEdit() |  |  |
| **RentalTermsForm** | **UploadDocumentForm** |  |
| displayRentalTermsForm() | displayDocumentUploadForm() |  |
| validateFormFields(formData) | highlightErrors(errorDetails) |  |
| highlightInvalidFormFields() | checkFileFormatAndSize() |  |
|  | uploadDocuments() |  |

|  |  |
| --- | --- |
| **BankProvider** | **ValidationProvider** |
| submitPayment(transaction) | returnVerificationFailure("EMAIL\_NOT\_VERIFIED") |
| submitTopUpTransaction(transactionID,userID,amount) | returnVerificationFailure("PHONE\_NOT\_VERIFIED") |
| verifyCard(cardDetails) | checkIdentityProof(userID) |
|  | returnVerificationFailure("NO\_ID\_PROVIDED") |
|  | returnVerificationFailure("INVALID\_PROFILE") |
|  | returnVerificationFailure("NO\_EXPERIENCES") |
|  | returnScoreResults(score,label) |
|  | sendForDigitalVerification(documentSet) |

|  |
| --- |
| **ManageDB** |
| getListingsForOwner(ownerID) |
| deleteAssociatedInterests(listingID) |
| saveListing() |
| queryListings() |
| sroteTemporaryLocation(ownerID,coordinates) |
| publishListing() |
| fetchSearchFormData() |
| queryRentalContracts(userID) |
| queryListingsByContract(contract) |
| queryEvaluation() |
| saveEvaluations(evaluation) |
| getUserEmail(userID) |
| getListingDetails(listingID) |
| deleteInterest(userID,listingID) |
| saveUserAvailability() |
| queryOwnerAvailability() |
| getTenandAvailability(userID,listingID) |
| saveInterest(userID,listingID) |
| saveReliabilityScore(userID,score) |
| storeTransaction(transactionID,receipt) |
| getUserID() |
| storeProviderError(userID,error) |
| getListingsByOwner(ownerID) |
| getInterestDeclarations(listings) |
| queryTrustScores(users) |
| getSelectedTenantUsers() |
| saveNotification() |
| getPendingRentalRequests(user) |
| updateRentalStatus() |
| getRentalOwner() |
| updateRentalStatus(user,"Rental Accept") |
| queryTemporaryRentals(owner) |
| updateRentalStatus() |
| updateRentalStatus(rental,"Completed") |
| storeReliabilityScore() |
| storeMessage() |
| storeNotifications() |
| queryScheduledVisits(user) |

## **Λεκτική Περιγραφή Κλάσεων**

### Μanager Classes

1. **CreateListingManager**: Γενική οντότητα που διαχειρίζεται τη δημιουργία αγγελιών από ιδιοκτήτες.
2. **LocationManager**: Γενική οντότητα που διαχειρίζεται την επιλογή και αναζήτηση τοποθεσίας για αγγελίες.
3. **SearchHousingManager**: Γενική οντότητα που διαχειρίζεται την αναζήτηση κατοικιών βάσει προτιμήσεων χρηστών.
4. **ExperienceManager**: Γενική οντότητα που διαχειρίζεται την καταγραφή και αξιολόγηση εμπειριών προηγούμενων ενοικιάσεων.
5. **IdentityVerificationManager**: Γενική οντότητα που διαχειρίζεται τη διαδικασία ταυτοποίησης χρηστών.
6. **RentalInterestManager**: Γενική οντότητα που διαχειρίζεται τη δήλωση ενδιαφέροντος για μια αγγελία από χρήστες.
7. **RentPaymentManager**: Γενική οντότητα που διαχειρίζεται τις πληρωμές ενοικίων και την αξιοπιστία των πληρωτών.
8. **PaymentMethodManager**: Γενική οντότητα που διαχειρίζεται τις μεθόδους πληρωμής των χρηστών.
9. **TopUpManager**: Γενική οντότητα που διαχειρίζεται την προσθήκη χρημάτων στο εικονικό πορτοφόλι του χρήστη.
10. **RentalRequestManager**: Γενική οντότητα που διαχειρίζεται τα αιτήματα επιλογής ενοικιαστών από ιδιοκτήτες.
11. **RentalApplicationManager**: Γενική οντότητα που διαχειρίζεται τις αιτήσεις χρηστών για συμμετοχή σε μίσθωση.
12. **FinalizeRentalManager**: Γενική οντότητα που διαχειρίζεται την ολοκλήρωση και υπογραφή συμβολαίων μίσθωσης.
13. **ChatManager**: Γενική οντότητα που διαχειρίζεται τη συνομιλία μεταξύ χρηστών και ιδιοκτητών.
14. **AppointmentManager**: Γενική οντότητα που διαχειρίζεται τον προγραμματισμό επισκέψεων σε διαθέσιμες κατοικίες.
15. **RentalTermination**: Γενική οντότητα που διαχειρίζεται τη λήξη και τον τερματισμό συμβολαίων μίσθωσης.

### Μanager Classes

1. **Listing**: Οντότητα η οποία αναπαριστά μια διαθέσιμη κατοικία προς ενοικίαση με όλα τα βασικά χαρακτηριστικά της.
2. **RentalInterest**: Οντότητα η οποία δηλώνει το ενδιαφέρον ενός χρήστη για μια συγκεκριμένη αγγελία.
3. **Message**: Οντότητα η οποία αναπαριστά ένα μήνυμα που ανταλλάσσεται μεταξύ χρηστών μέσα από το σύστημα.
4. **Photo**: Οντότητα η οποία περιέχει φωτογραφίες που σχετίζονται με μία αγγελία
5. **Location**: Οντότητα η οποία ορίζει τη γεωγραφική θέση μιας κατοικίας ή χρήστη**.**
6. **UserPreferences**: Οντότητα η οποία περιγράφει τις προτιμήσεις ενός χρήστη κατά την αναζήτηση κατοικίας.
7. **MapService**: Οντότητα η οποία προσφέρει γεωγραφικές λειτουργίες και υπηρεσίες χαρτών.
8. **RentalContract**: Οντότητα η οποία αναπαριστά το νομικό συμφωνητικό ενοικίασης ανάμεσα σε ιδιοκτήτη και ενοικιαστή.
9. **Review**: Οντότητα η οποία περιλαμβάνει αξιολογήσεις και σχόλια χρηστών για εμπειρίες ενοικίασης.
10. **UserProfile**: Οντότητα η οποία περιγράφει τα βασικά προσωπικά στοιχεία και την ταυτότητα του χρήστη.
11. **TrustScore**: Οντότητα η οποία αποτυπώνει την εμπιστοσύνη που αποδίδεται σε έναν χρήστη βάσει της δραστηριότητάς του.
12. **EmailService**: Οντότητα η οποία διαχειρίζεται την αποστολή ειδοποιήσεων και ενημερώσεων μέσω email.
13. **UserAvailability**: Οντότητα η οποία δηλώνει τη διαθεσιμότητα ενός χρήστη για επίσκεψη ή ενοικίαση.
14. **PaymentHistory**: Οντότητα η οποία καταγράφει το ιστορικό πληρωμών ενός χρήστη.
15. **ReceiptPDF**: Οντότητα η οποία περιέχει τα παραστατικά πληρωμών σε μορφή PDF.
16. **Wallet**: Οντότητα η οποία αναπαριστά το εικονικό πορτοφόλι του χρήστη για συναλλαγές.
17. **Transaction**: Οντότητα η οποία περιγράφει μια οικονομική πράξη στο πλαίσιο της πλατφόρμας.
18. **PaymentMethod**: Οντότητα η οποία περιέχει τις αποθηκευμένες μεθόδους πληρωμής ενός χρήστη.
19. **Notification**: Οντότητα η οποία αντιπροσωπεύει μια ειδοποίηση που αποστέλλεται στον χρήστη.
20. **RentalTerms**: Οντότητα η οποία περιλαμβάνει τους όρους και τις προϋποθέσεις μιας ενοικίασης.
21. **ReliabilityScore**: Οντότητα η οποία ποσοτικοποιεί την αξιοπιστία ενός χρήστη στο σύστημα.
22. **ChatSession**: Οντότητα η οποία αναπαριστά μια συνεδρία συνομιλίας μεταξύ χρηστών που σχετίζονται με μια ενοικίαση.
23. **ChatMessage**: Οντότητα η οποία περιέχει το περιεχόμενο ενός μεμονωμένου μηνύματος σε μια συνομιλία.

### UI Screens

1. **MainScreen**: Οθόνη η οποία λειτουργεί ως το κεντρικό σημείο πλοήγησης της εφαρμογής.
2. **CreateListingScreen**: Οθόνη η οποία επιτρέπει τη δημιουργία νέας αγγελίας από ιδιοκτήτη.
3. **LocationScreen**: Οθόνη η οποία επιτρέπει την επιλογή ή αναζήτηση τοποθεσίας για μια αγγελία.
4. **ExperienceHistoryScreen**: Οθόνη η οποία εμφανίζει το ιστορικό εμπειριών και αξιολογήσεων ενός χρήστη.
5. **UserAvailabilityScreen**: Οθόνη η οποία επιτρέπει σε χρήστες να δηλώσουν τη διαθεσιμότητά τους.
6. **TopUpScreen**: Οθόνη η οποία επιτρέπει την προσθήκη χρημάτων στο ψηφιακό πορτοφόλι.
7. **SearchHousingScreen**: Οθόνη η οποία επιτρέπει την αναζήτηση διαθέσιμων κατοικιών.
8. **RentPaymentScreen**: Οθόνη η οποία διαχειρίζεται την καταβολή του ενοικίου και τις σχετικές συναλλαγές.
9. **RentalRequestScreen**: Οθόνη η οποία επιτρέπει στους ιδιοκτήτες να διαχειριστούν αιτήματα ενοικίασης.
10. **RentalApplicationScreen**: Οθόνη η οποία επιτρέπει σε χρήστες να υποβάλουν αίτηση για μίσθωση.
11. **FinalizeRentalScreen**: Οθόνη η οποία καθοδηγεί τη διαδικασία ολοκλήρωσης μιας μίσθωσης.
12. **ChatScreen**: Οθόνη η οποία υποστηρίζει ανταλλαγή μηνυμάτων μεταξύ χρηστών.
13. **AppointmentScreen**: Οθόνη η οποία επιτρέπει τον προγραμματισμό επισκέψεων σε κατοικίες.

### UI Forms

1. **CreateListingForm**: Φόρμα η οποία συλλέγει τα απαραίτητα στοιχεία για τη δημιουργία αγγελίας.
2. **UploadPhotoForm**: Φόρμα η οποία επιτρέπει την υποβολή φωτογραφιών για μία αγγελία.
3. **SearchHousingForm**: Φόρμα η οποία καταγράφει τις προτιμήσεις αναζήτησης κατοικίας από τον χρήστη.
4. **ReviewForm**: Φόρμα η οποία επιτρέπει την υποβολή αξιολόγησης για μια εμπειρία ενοικίασης.
5. **TopUpForm**: Φόρμα η οποία δηλώνει το ποσό και τη μέθοδο για ενίσχυση του πορτοφολιού.
6. **PaymentMethodForm**: Φόρμα η οποία συλλέγει πληροφορίες νέας μεθόδου πληρωμής.
7. **RentalTermsForm**: Φόρμα η οποία περιλαμβάνει τους όρους και τα βασικά στοιχεία μιας ενοικίασης.
8. **UploadDocumentForm**: Φόρμα η οποία επιτρέπει την υποβολή απαιτούμενων εγγράφων για την ολοκλήρωση της ενοικίασης.

### Services Classes

1. **BankProvider**: Πάροχος ο οποίος συνδέεται με εξωτερικό τραπεζικό σύστημα για την επεξεργασία πληρωμών.
2. **ValidationProvider**: Πάροχος ο οποίος υλοποιεί λογική επικύρωσης για δεδομένα και αιτήσεις του συστήματος.
3. **ManageDB**: Κλάση η οποία διαχειρίζεται τη σύνδεση και τις λειτουργίες με τη βάση δεδομένων της εφαρμογής.

## Excel

A screenshot of a computer screen

AI-generated content may be incorrect.Ενδεικτικά στο excel το οποίο βρίσκεται στο GitHub θα βρείτε αναλυτικά τις κλάσεις με τα attributes. Με πράσινο χρώμα επισημαίνονται όσα έχουν υλοποιηθεί σε κώδικα. Γενικότερα μέσω του excel αυτού μπορούμε να δούμε πολύ εύκολα το σύνολο των κλάσεων των μεθόδων και να κατανοήσουμε το project.