



Profile requestOwner, ChangesOnOffer changes): ChangesRequest UserRole MAX OFFERS NUM: int * GuiType - createChangesRequest(HolidayOffer agencyOffer, IMAGES_NUM: int * <<enumeration>> HolidayOffer desiredOffer, String description): ChangesRequest availableImages: ArrayList<URL> * <<enumeration>> + SIMPLE_USER - createDesiredAccommodationOffer(AccommodationType type, + TRAVEL_AGENCY + JAVAFX_GUI searchAccommodationOffers(String destination, Duration int quality, int numOfRooms): AccommodationOffer checkInOutDates, + CMDLINE_GUI + toPersistenceType(): String int numOfRoomsRequired): List<Accommodation> + fromPersistenceType(String role): UserRole + createDesiredTransportOffer(TransportType type, int quality, getAccommodationImages(int accommodationId): List<URL> int numOfTravelers, Route route): TransportOffer generateRandomAccommodationOffers(int lengthOfStay, int Creates instances of numOfRooms): HolidayOffer, - checkRequestValidity(ChangesRequest request): void AccommodationOffer, List<Accommodation> * TransportOffer and HolidayOfferAdapter generateRandomImages(): List<URL> * **DbConnection** ProfileDaoFactory connection: Connection instance: ProfileDaoFactory instance: DbConnection - useMysql: boolean DB CREDENTIALS FILENAME: String Creates instances of getInstance(): ProfileDao + <u>getInstance(): DbConnection</u> ProfileDaoMysgl and loadDaoTypeFromConfigurationFile(): boolean ProfileDaoFileSystem. All Dao classes are associated <u>getConnection(): Connection</u> - createDao(): ProfileDao to the DbConnection Every class that uses the ProfileDao is associated to this AddressChecker + checkAddressExistance(String address): void *= needed to simulate external API



