



```

class Account {
    String email;
    String username;
    String password;
    String name;
    Address address;
    AccountType type;

    1 -- 1 AccountType;
    1 -- * Address;
    Account(String email, String username, String password, String name, Address address, AccountType
    type) {
        this.email = email;
        this.username = username;
        this.password = password;
        this.name = name;
        this.address = address;
        this.type = type;
    }
    Account(){}
    void setEmail(String email) {this.email = email;}
    String getEmail() {return email;}
    void setUsername(String username) {this.username = username;}
    String getUsername() {return username;}
    String getPassword() {return password;}
    void setPassword(String password) {this.password = password;}
    String getName(){return name;}
    void setName(String name) {this.name = name;}

```

```

    Address getAddress() {return address;}

    void setAddress(Address address){this.address = address;}

    void setType(AccountType type) {this.type = type;}

    AccountType getType(){return type;}

}

class AccountType {enum(ADMIN, CLIENT, COOK);}

class Address {

    String street;

    String city;

    String country;

    String postalCode;

    Address(String street, String city, String country, String postalCode) {

        this.street = street;

        this.city = city;

        this.country = country;

        this.postalCode = postalCode;}

    Address(){ }

    void setStreet(String street){}

    String getStreet(){ }

    void setCity(String city){}

    String getCity(){ }

    void setCountry(String country){}

    String getCountry(){ }

```

```

    void setPostalCode(String postalCode){}

    String getPostalCode(){}
```

}

```

class Administrator{

    isA Account;

    Administrator(String email, String username, String password, String name, Address address) {

        super(email, username, password, name, address, AccountType.ADMIN);

    }

    Administrator(){}
```

```

    void deleteAccount() {}

    void reviewComplaint() {}

    void createSuspension() {}

    void approveCook() {}

    pubvoid banAccount() {}

}
```

```

class Card {

    String nameOnCard;

    String cardNum;

    String securityCode;

    String expiryDate;

    1 -- 1 Client;

    Card(String nameOnCard, String cardNum, String securityCode, String expiryDate) {

        this.cardNum = cardNum;

        this.nameOnCard = nameOnCard;

        this.securityCode = securityCode;

        this.expiryDate = expiryDate;}

    Card(){}
```

```

    void setCardOnName(String nameOnCard){}

    String getNameOnCard(){}
```

```

void setCardNum(String cardNum){}

String getCardNum(){}

void setSecurityCode(String securityCode){}

String getSecurityCode(){}

void setExpiryDate(String expiryDate){}

String getExpiryDate(){}
}

class Client{

    isA Account;

    ArrayList<CuisineTypes> preferences;

    Card cardInfo;

    ArrayList<Meal> favMeals;

    HashMap<Meal, Integer> ratings;

    1 -- * Meal;

    1 -- * CuisineTypes;

    Client(String email, String username, String password, String name, Address address, Card card) {

        super(email, username, password, name, address, AccountType.CLIENT);

        cardInfo = card;}

    Client(){}

    void addPreferences(CuisineTypes preference) {}

    ArrayList<CuisineTypes> getPreferences() {}

    void addPayment(Card card) {}

    Card getPayment() {}

    void addFavMeal(Meal meal) {}

    ArrayList<Meal> getFavMeals() {}

    void giveRating(Meal meal, Integer rating) {}

    HashMap<Meal, Integer> getRatings() {}

    void purchaseMeal(Meal meal, Cook cooker) {}

}

```

```

class ClientSignUpScreen{

    Button homeBtn;

    Button registerClientBtn;

    EditText clientFirstName;

    EditText clientLastName;

    EditText password;

    EditText email;

    EditText username;


    EditText postal;

    EditText street;

    EditText city;

    EditText country;


    EditText cardNum;

    EditText securityCode;

    EditText exp;


    FirebaseAuth mAuth;


    1 -- 1 LoggedInScreen;

void onCreate(Bundle savedInstanceState) {}

    void homeNavigator(){}

    void registerClientData(){}


    void textClear(){}

}

class Cook{

```

```

        isA Account;

ArrayList<String> complaintsAgainst;

    ArrayList<Meal> menu;

    HashMap<Meal, Integer> ratings;

    ArrayList<CuisineTypes> cuisine;

    1 -- * CuisineTypes;

    1 -- * Meal;

Cook(String email, String username, String password, String name, Address address) {}

    Cook(){}

    void addCuisineType(CuisineTypes type) {}

    ArrayList<CuisineTypes> getCuisineTypes() {}

    void addComplaints(String complaintMsg) {}

    ArrayList<String> getComplaints() {}

    void addMeal(Meal meal) {}

    void removeMeal(Meal meal) {}

    ArrayList<Meal> getMeal() {}

    void addRating(Meal meal, Integer rating) {}

    HashMap<Meal, Integer> getRatings() {}

}

class CookSignUpScreen{

    Button homeBtn;

    Button registerCookBtn;

    EditText cookFirstName;

    EditText cookLastName;

    EditText password;

    EditText email;

    EditText username;

    EditText postal;

```

```

    EditText street;

    EditText city;

    EditText country;

    FirebaseAuth mAuth;

    1 -- 1 LoggedInScreen;

void onCreate(Bundle savedInstanceState) {}

    void homeNavigator(){}

    void registerCookBtn(){}

    void textClear(){}

}

class CuisineTypes { * -- * CuisineTypes roleName;

enum(ITALIAN, INDIAN, BURGERS, SANDWHICHES, VEGETARIAN, MEAT, DRINKS, BAKERY, VEGAN,
CHINESE, MEXICAN, JAPANESE,

    VIETNAMESE, THAI, KOREAN, CARRIBEAN, FRENCH, BREAKFAST, GREEK);}

class LoggedInScreen{

    FirebaseUser user;

    DatabaseReference reference;

    String userID;

    Button logOutButton;

    * -- 1 MainActivity;

void onCreate(Bundle savedInstanceState) {}

    void logOutOnClick(){}

}

class MainActivity{

    EditText email;

    EditText pass;

    Button cookRegisterBtn;

    Button clientRegisterBtn;

```



```

    Button loginBtn;

    FirebaseAuth mAuth;

    void onCreate(Bundle savedInstanceState) {}

    public void cookRegisterNavigator() {}

    public void clientRegisterNavigator() {}

    public void loginNavigate(){}

    public void loginUser(){}

}

class Meal {

    ArrayList<CuisineTypes> types;

    String name;

    double price;

    Meal(String name, double price, ArrayList<CuisineTypes> types) {}

    Meal(){}

    void changePrice(double price) {}

    double getPrice() {}

    String getName() {}

    void setName(){}

    ArrayList<CuisineTypes> getTypes() {}

    void addType(CuisineTypes type) {}

}

```