***-: Food Save Application :-***

hybrid mobile app

SYSTEM OVERVIEW  
**Choice of Android**Android has been chosen for this project due to open-source nature of the platform as well as the ease of development and deployment. It also has the largest market share and also supports cross platform application development, i.e., developers can develop Android application in Mac, Windows and many UNIX-based operating systems like Ubuntu.

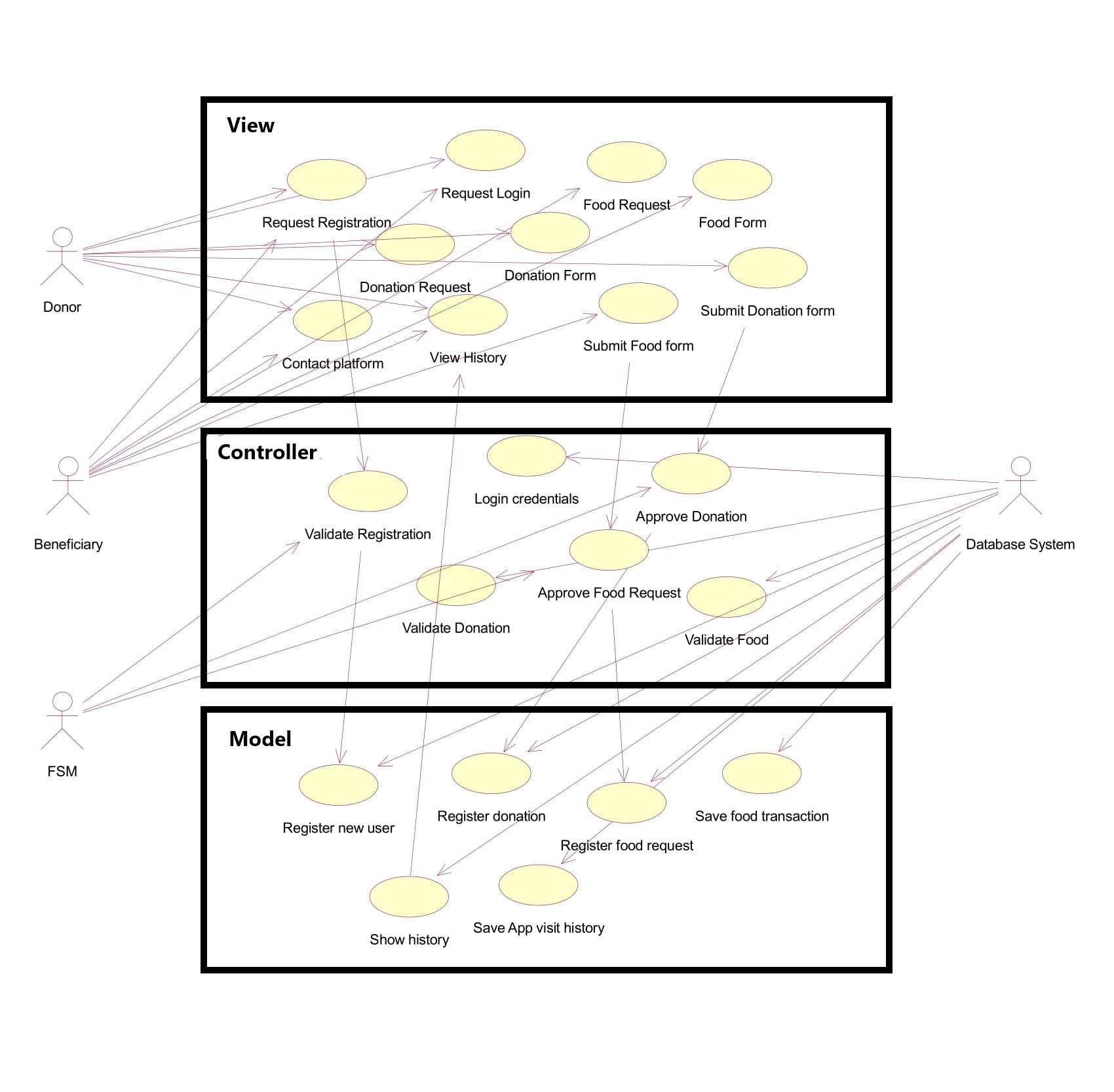
**System Architecture**This application consists of an Android application on the client side and JAVA-MySQL application on the server side. The Android application is the part visible to the user and one it interacts with, while the JAVA/MySQL-based server-side component serves as an interface between the Android application and the database on the server.

The use case for the client-side application is shown in Figure below, showing all the cases available to the user in the application.

|  |  |
| --- | --- |
|  |  |
| Fig. Use case for client-side application | Use case for server-side implementation. |

**MVC Architecture**

The application is a hybrid mobile app having 3 layers of presentation, business logic and data, defining the basis for MVC (Model View Controller) architecture. Specifically for Android mobile app, we will use **Clean Architecture** to allow each layer working separately and independently from others, helping to adopt changes only for that particular layer without disturbing the architecture of whole system.



References:

<https://www.netsolutions.com/insights/mobile-app-architecture-guide/>

<https://www.toptal.com/android/benefits-of-clean-architecture-android>

**Operation Contracts:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**C01- Registration()**

**Contract C01:** Registration

**Operation:** Registration()

**Cross Reference:** Request Registration

**Precondition:** The internet accessibility is mandatory, and the user should not be registered to the system already.

**Postcondition:** After successful registration, an instance of donor class is generated which stores the donor details and donor will be registered to the system as User.

**C02- Login/Logout()**

**Contract C02:** Login/Logout

**Operation:** Login()/Logout()

**Cross Reference:** Request Login/Log out

**Precondition:** User must have internet facility and must be registered before Login.

**Postcondition:** After this, the system services provided to the user and association with database.

**C03- RequestForDonation()**

**Contract C03:** Request for Donation

**Operation:** RequestForDonation()

**Cross Reference:** Request Donation

**Precondition:** Donor must have an internet facility, logged in to the system and donation form is opened in front of user.

**Postcondition:** The instance of donor class is passed to organization(FSM). FSM receives the donor with food item. There is association between Donor and organization(FSM).

**C04 - NotifyDonor()**

**Contract C04:** Notify Donor

**Operation:** NotifyDonor()

**Cross Reference:** Approve Donation

**Precondition:** User must has submitted the donation request form to FSM.

**Postcondition:** There is an association between Donor and organization(FSM) in which organization sent a message of approval or disapproval to donor.

**C05 - ApproveDonation()**

**Contract C05:** ApproveDonation

**Operation:** ApproveDonation()

**Cross Reference:** Approve Donation

**Precondition:** User must have submitted the donation request form to FSM.

**Postcondition:** The instance of donor class is passed to organization(FSM). FSM receives the donor with food item. There is association between Donor and organization(FSM).

**C06 - RequestForFood()**

**Contract C06:** Request for Food

**Operation:** RequestForFood()

**Cross Reference:** Request Food

**Precondition:** Beneficiary is logged in to the system and food request form is opened in front of user.

**Postcondition:** The instance of Beneficiary class is passed to organization(FSM). FSM receives the beneficiary with his valid personal information. There is association between Beneficiary and organization(FSM).

**C07 - NotifyBeneficiary()**

**Contract C07:**  Notify Beneficiary

**Operation:** NotifyBeneficiary()

**Cross Reference:** Approve Delivered Food

**Precondition:** Beneficiary must has submitted the food request form to FSM.

**Postcondition:** There is an association between Beneficiary and organization(FSM) in which organization sent a message of approval or disapproval to Beneficiary.

**C08- ApproveDeliveredFood()**

**Contract C08:** Approve Delivered Food

**Operation:** Approve Delivered Food()

**Cross Reference:** Approve Food Delivery

**Precondition:** Beneficiary has sent the food request toOrganization and organization has declared beneficiary eligible for food.

**Postcondition:** There is an association between Beneficiary and organization(FSM). Instance of Donor class to get foodItem from Food class is passed to FSM class

**C09- ViewHistory()**

**Contract C09:** View History

**Operation:** ViewHistory()

**Cross Reference:** Show History

**Precondition:** Record of all details are added by organization in database correctly.

**Postcondition:** There is an association between history class and organization(FSM). Instance of history class is created by FSM class.