**Software Engineering Mini Project**

**Farmers Buddy**

**Software Requirements Specifications**

Version 2.0

**Team Members**:

1. Praveen Jain – 07IT36
2. Veeresh B – 07IT64
3. Karthik Bellur – 07IT20

# *Index*

**A] UML Diagrams..................................................................................................3**

1. Use Case Diagram

1.1) User : Administrator

1.2) User : Administrator

1.3) User : Administrator

1.4) User : Administrator

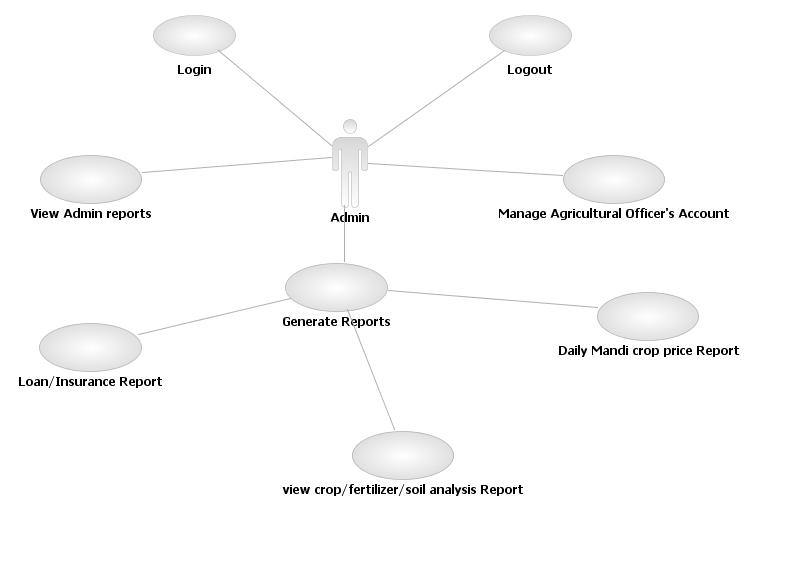
1.5) User : Administrator

**2.**

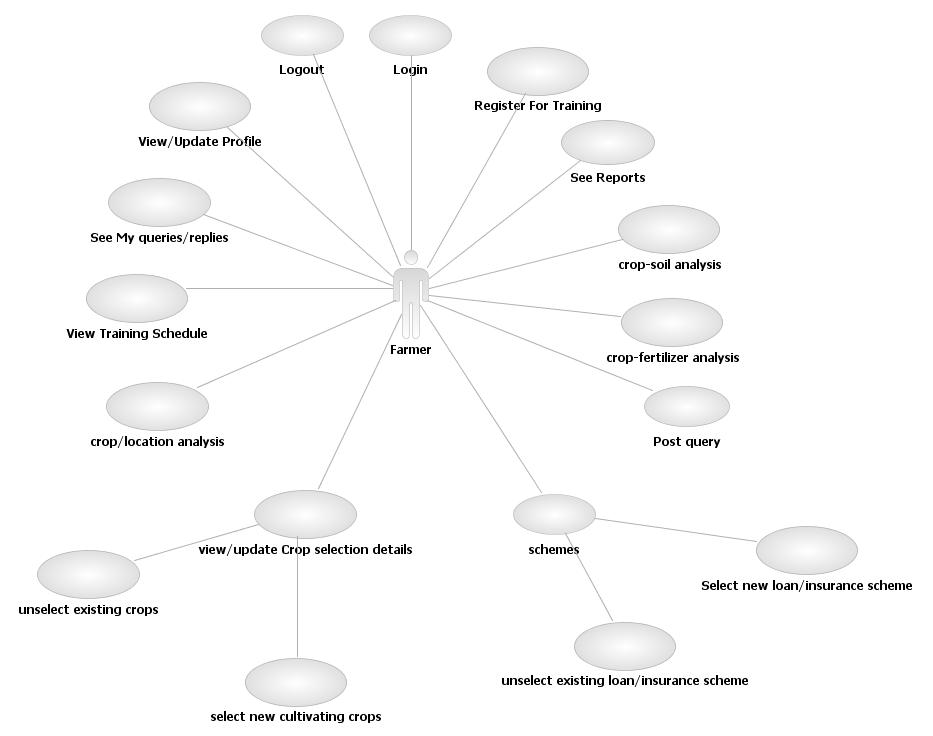
**1. Use Case Diagram**

The diagrams attached are the use-cases of various agents acting on the system.

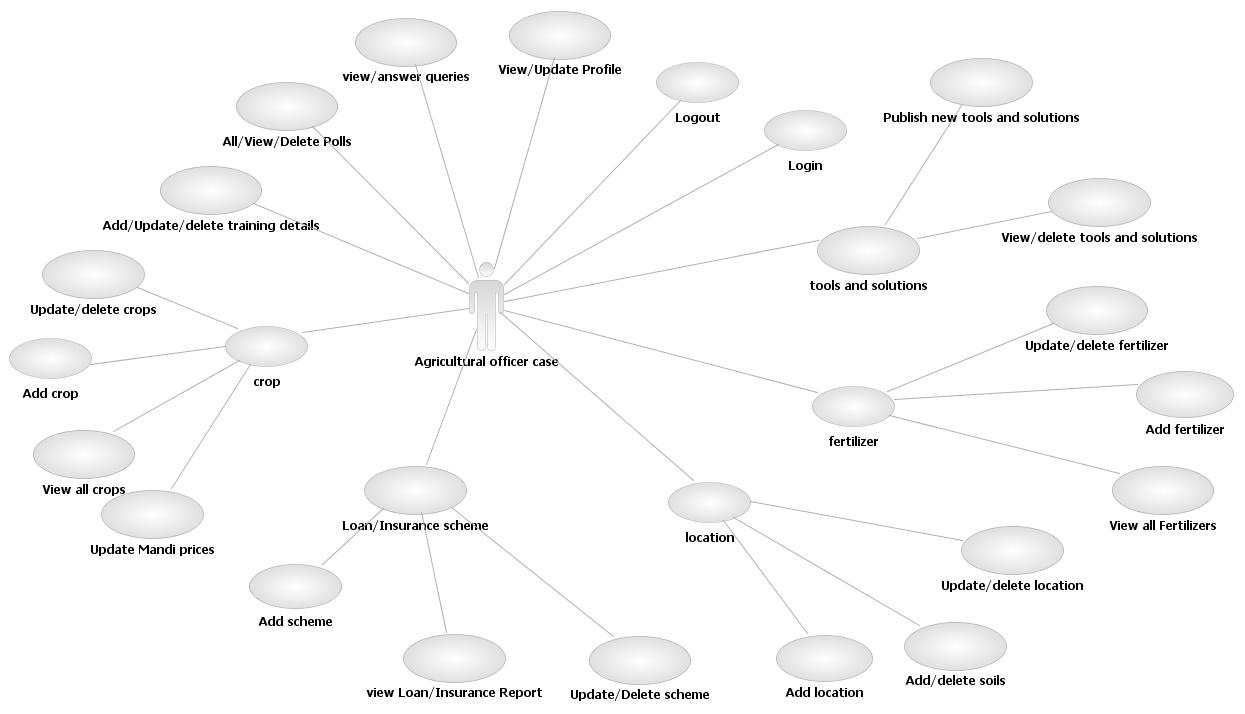
**1.1) User: Administrator**



## 1.2) User: Farmer

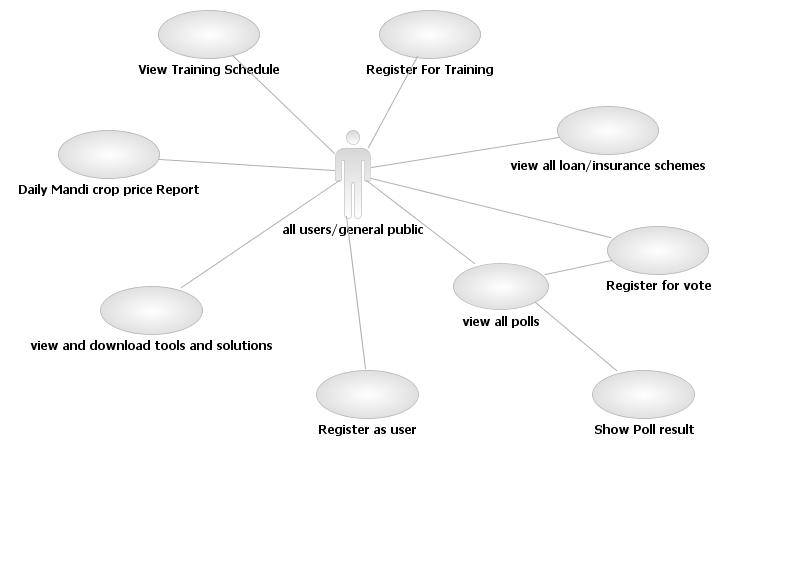


## 1.3) User: Agricultural Officer

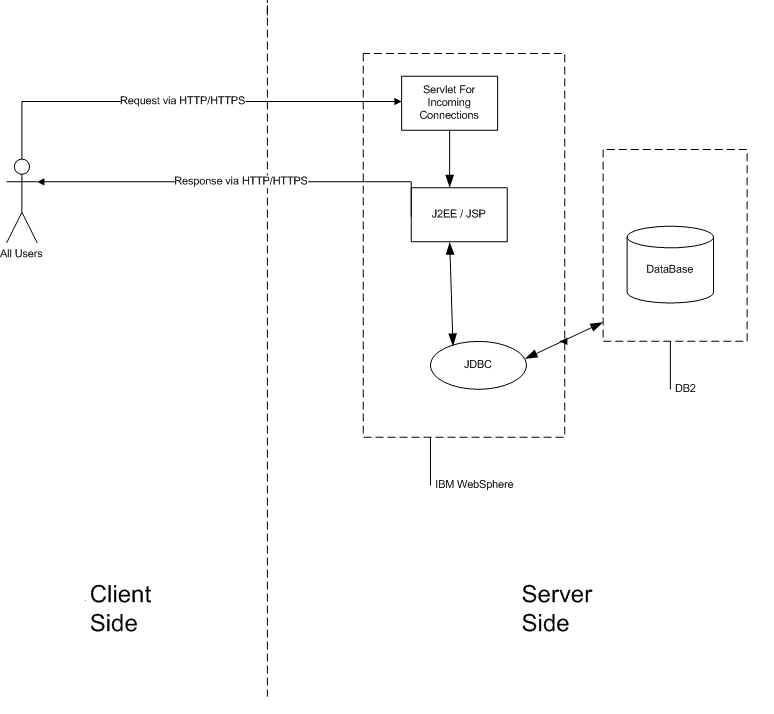


## 1.4) User: Agricultural Students

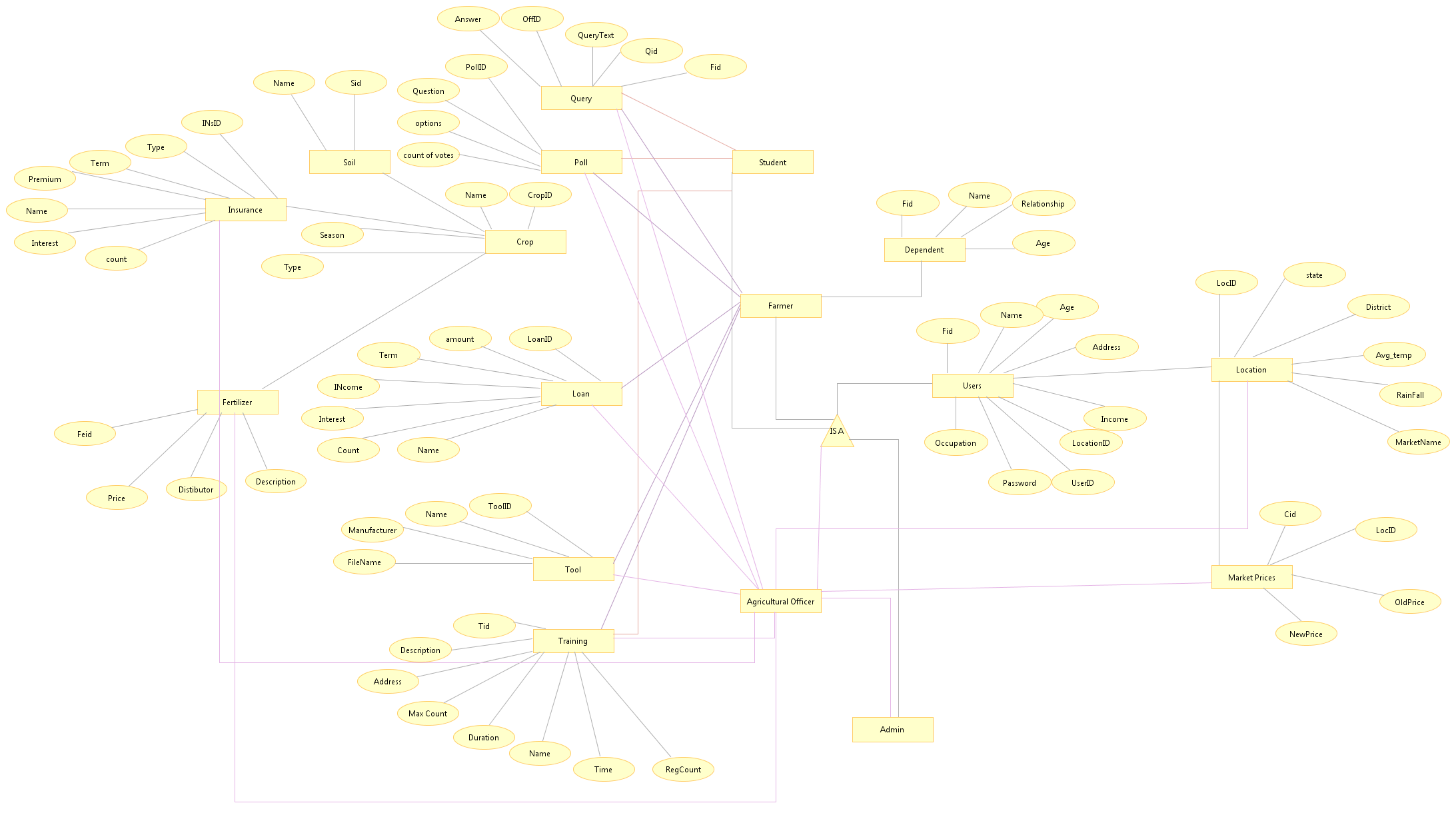
## student.jpg1.5) User: General Public



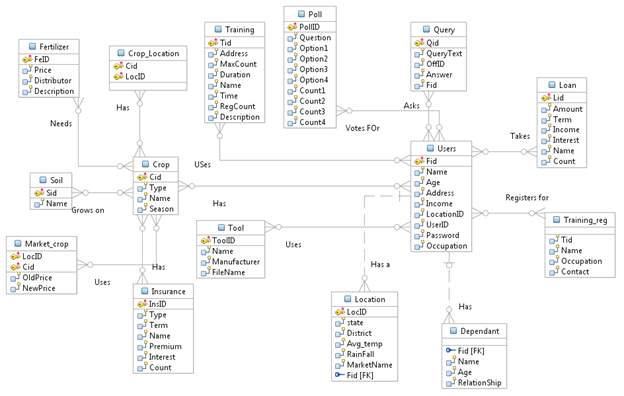
# 2. Architecture Diagram



3. ER Diagram



4. Class Diagram



Each classes mentioned above are implemented as JavaBeans and each JavaBean is associated with a corresponding Action. Action contains the behaviour(methods) of the classes.

1. LoginAction

* doLogin()
* validate()

2. CropAction

* addCrop()
* deleteCrop()
* updateCrop()
* getAllCrops()
* getMyCrops()
* addNewCrop()
* deleteOldCrop()
* getFavFerts()
* addNewFavFert()
* deleteOldFavFert()
* getFavSoils()
* addNewFavSoil()
* deleteOldFavSoil()

3. FertilizerAction

* addFert()
* deleteFert()
* updateFert()
* getFertDetails()
* getAllFerts()

4. InsuranceAction

* addIns()
* deleteIns()
* updateIns()
* getInsDetails()
* getAllInss()
* deleteOldIns()
* addNewIns()
* getAllReportInss()

5. LoanAction

* addLoan()
* deleteLoan()
* updateLoan()
* getLoanDetails()
* getAllLoans()
* deleteOldLoan()
* addNewLoan()
* getAllReportLoans()

6. LocationAction

* addLoc()
* deleteLoc()
* updateLoc()
* getLocDetails()
* getAllLocs()
* deleteOldFavCrop()
* addNewFavCrop()
* getFavCrops()

7. MarketAction

* getPriceOfCrop()
* getMarketPricesOfCrop()
* getCropPricesOfMarket()
* updateCropPrice()

8. PollAction

* addPoll()
* deletePoll()
* getAllPolls()
* getPollResult()
* registerVote()

9. QueryAction

* postQuery()
* getQueryList()
* addReply()
* getAllMyQueries()

10. SoilAction

* addSoil()
* deleteSoil()
* getAllSoils()

11. ToolAction

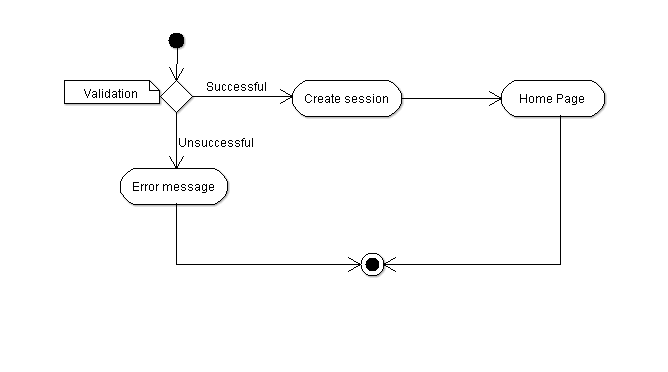
* addTool()
* deleteTool()
* getAllTools()

12. TrainingAction

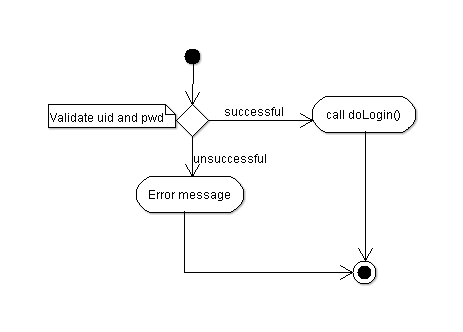
* addTraininingDet()
* deleteTrainingDet()
* getAllTrainingDet()
* updateTrainingDet()
* registerForTraining()

5. Activity Diagram

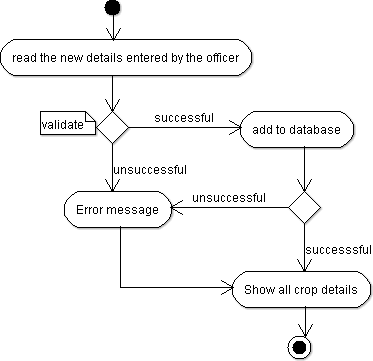
**5.1) doLogin**



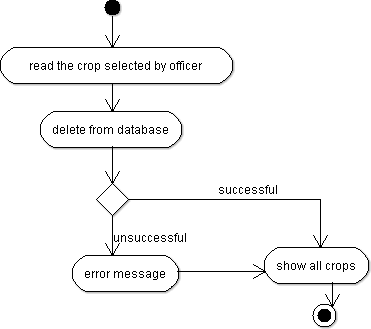
**5.2) validate**

****

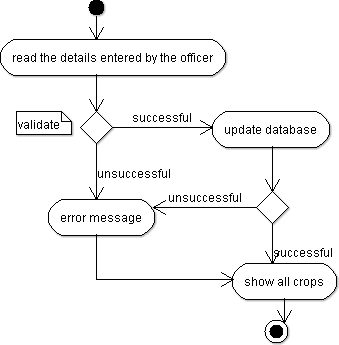
**5.3) addCrop**

****

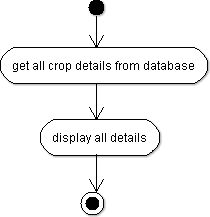
**5.4) deleteCrop**

****

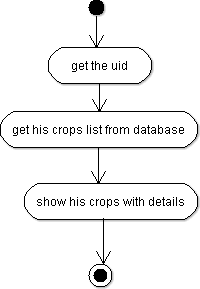
**5.5) updateCrop**

****

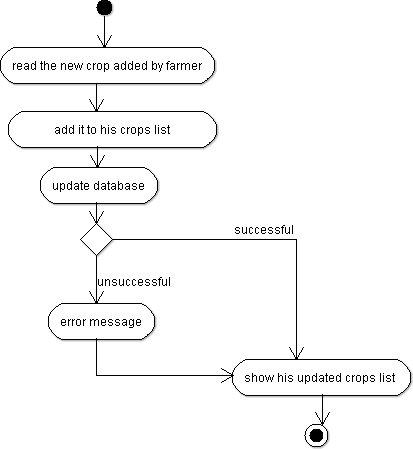
**5.6) getAllCrops**



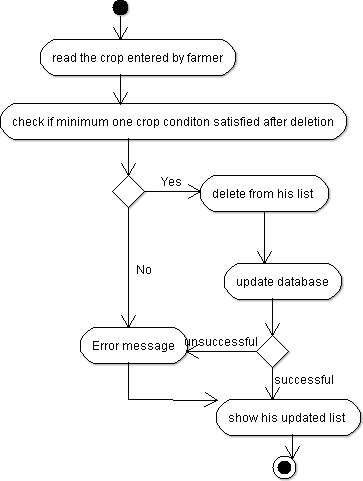
**5.7) getMyCrops**



**5.8) addNewCrop**

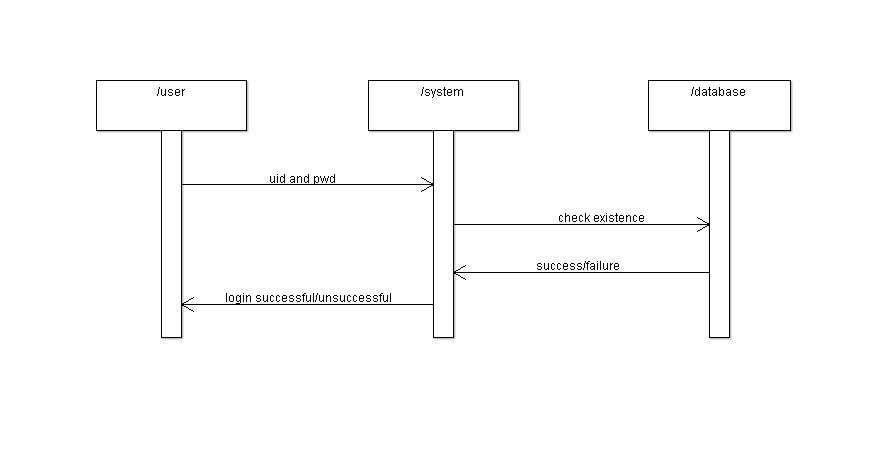


**5.9) deleteOldCrop**

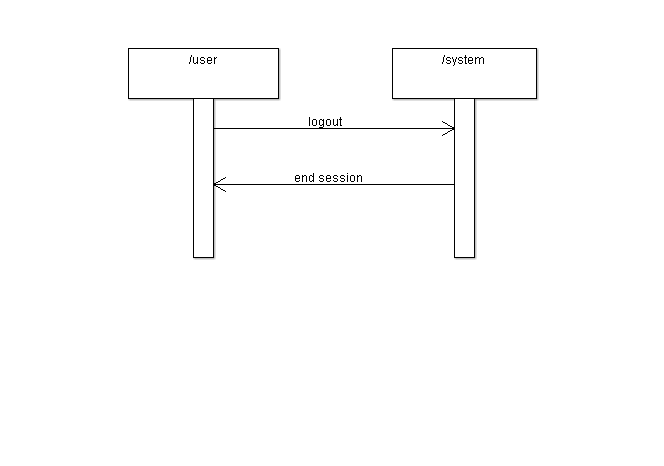


6. Sequence Diagram

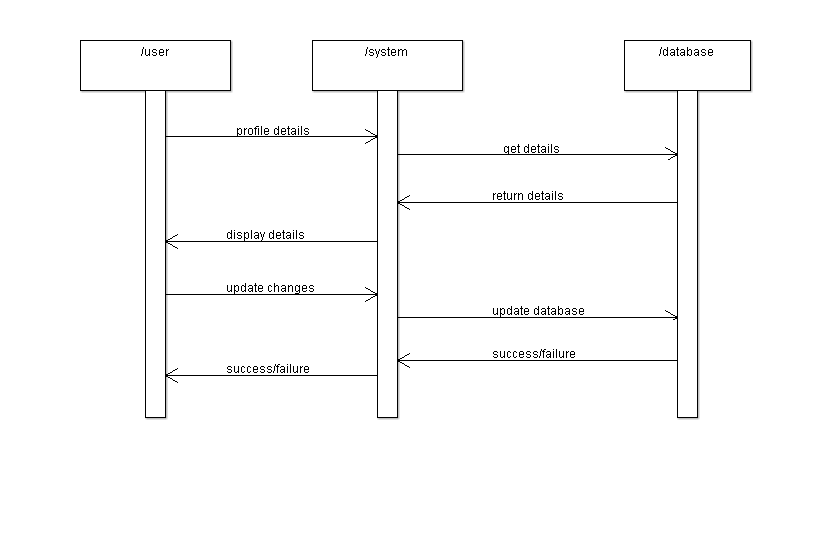
**6.1)login**



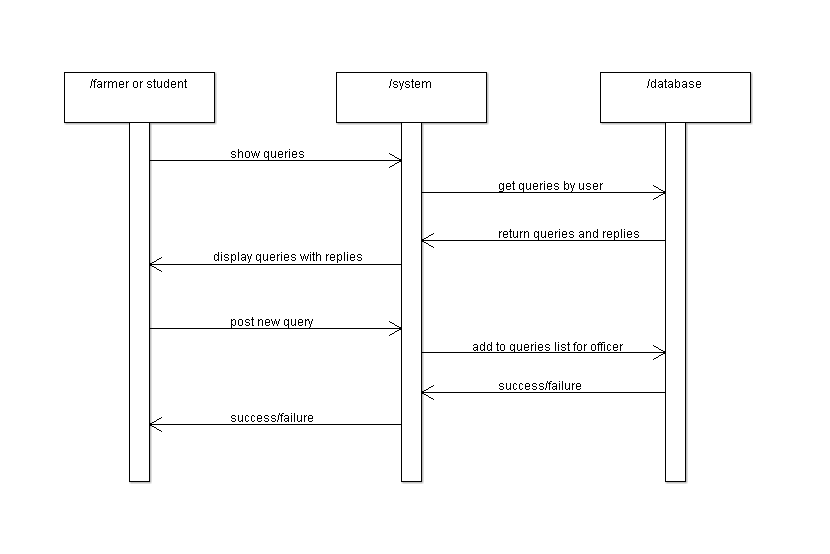
**6.1)logout**



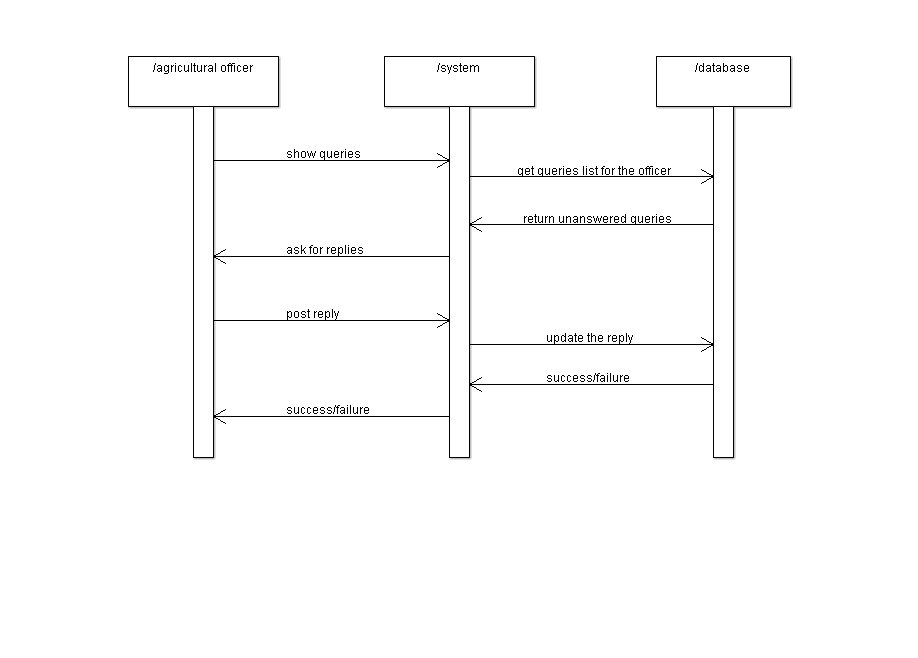
6.3)update profile



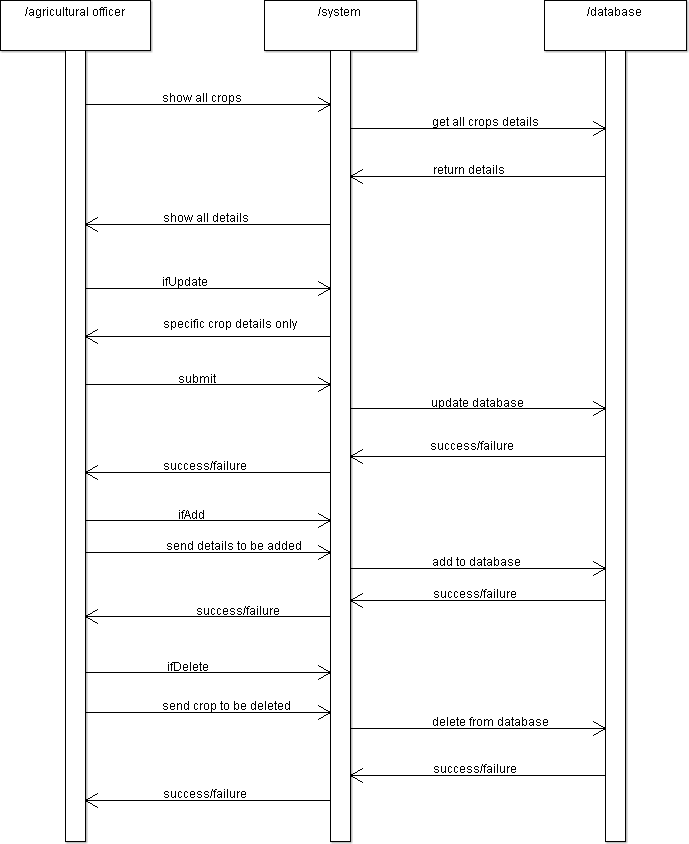
6.4)postquery

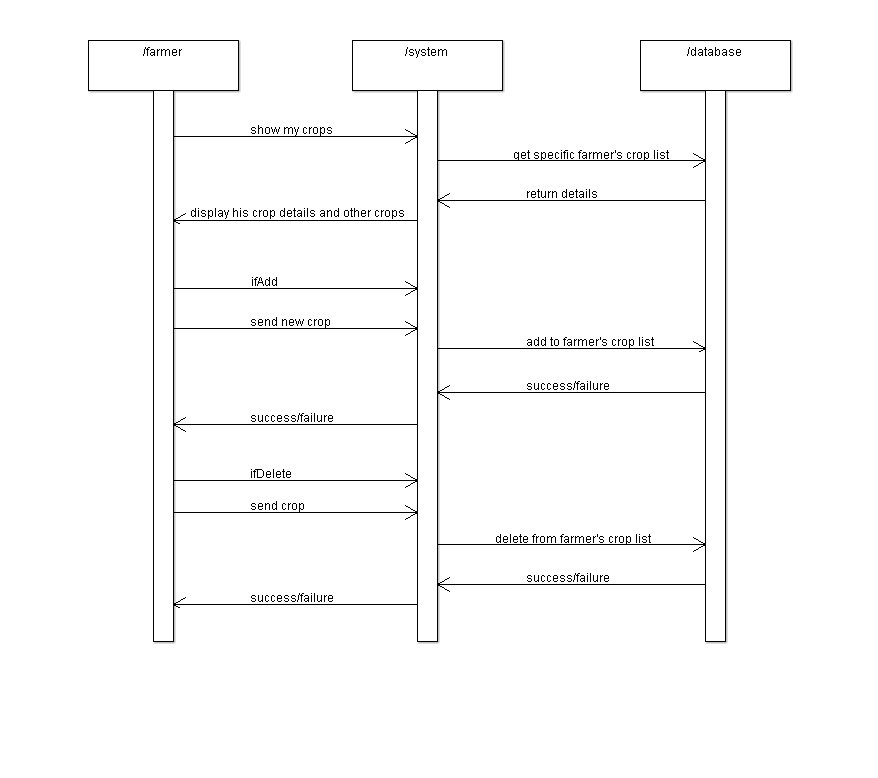


6.5)answer queries



6.6)add/delete/ update crops



6.7) add/delete crops (farmer)

6.8)reports

