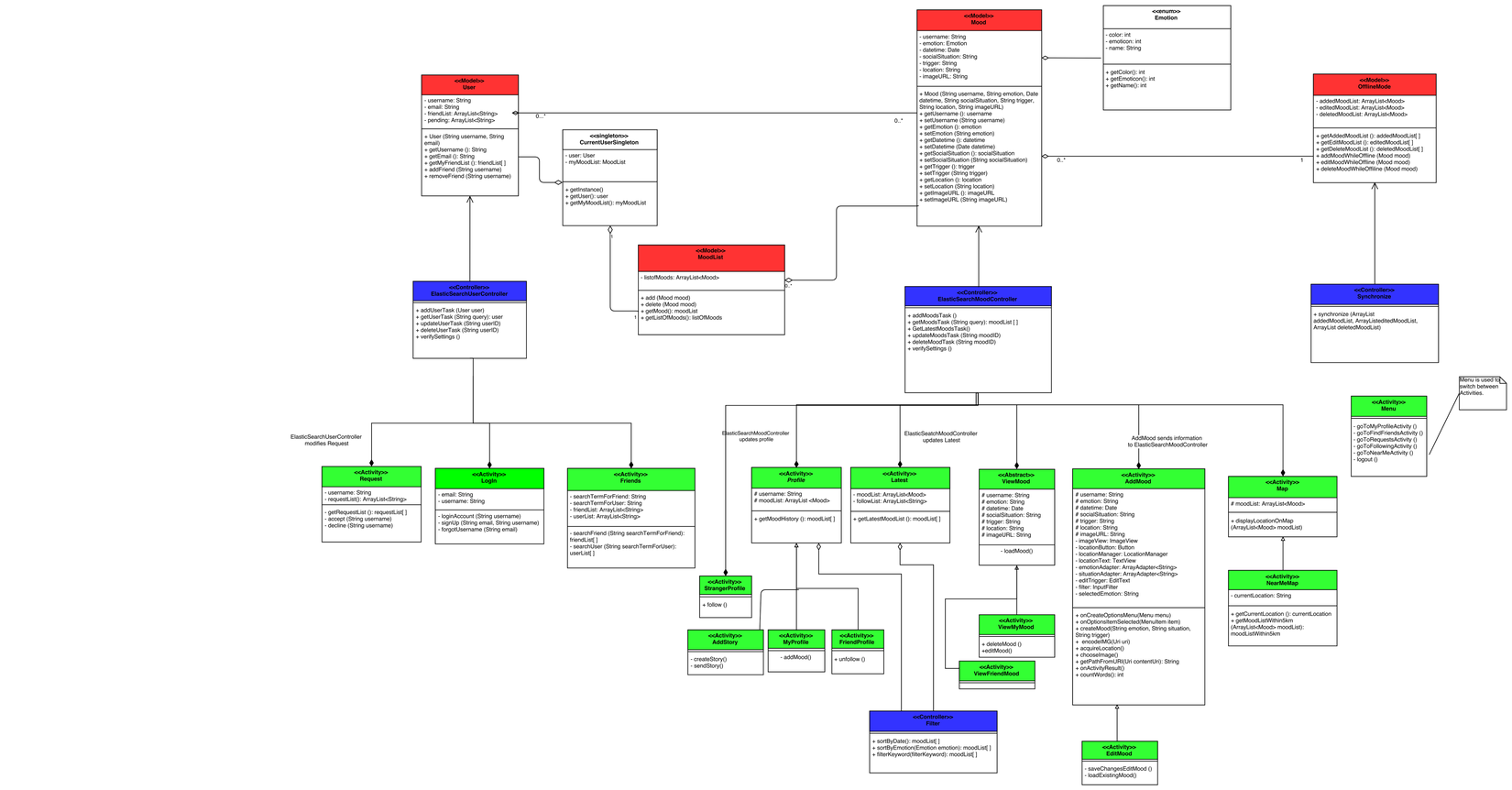
<<Model>>
User
- username: String
- email: String
- friendList: ArrayList<String>
- pending: ArrayList<String>
+ User (String username, String email)
+ getUsername () : String
+ getEmail () : String
+ getMyFriendList () : friendList [ ]
+ addFriend (String username)
+ removeFriend (String username)
0...*
0...*
<<Model>>
Mood
- username: String
- emotion: Emotion
- datetime: Date
- socialSituation: String
- trigger: String
- location: String
- imageURL: String
+ Mood (String username, String emotion, Date datetime, String socialSituation, String trigger, String location, String imageURL)
+ getUsername () : username
+ setUsername (String username)
+ getEmotion () : emotion
+ setEmotion (String emotion)
+ getDatetime () : datetime
+ setDatetime (Date datetime)
+ getSocialSituation () : socialSituation
+ setSocialSituation (String socialSituation)
+ getTrigger () : trigger
+ setTrigger (String trigger)
+ getLocation () : location
+ setLocation (String location)
+ getImageURL () : imageURL
+ setImageURL (String imageURL)
<<enum>>
Emotion
- color: int
- emotion: int
- name: String
+ getColor() : int
+ getEmoticon() : int
+ getName() : int
0...*
1
<<Model>>
OfflineMode
- addedMoodList: ArrayList<Mood>
- editedMoodList: ArrayList<Mood>
- deletedMoodList: ArrayList<Mood>
+ getAddedMoodList () : addedMoodList[ ]
+ getEditMoodList () : editedMoodList[ ]
+ getDeleteMoodList () : deletedMoodList[ ]
+ addMoodWhileOffline (Mood mood)
+ editMoodWhileOffline (Mood mood)
+ deleteMoodWhileOffline (Mood mood)
<<Singleton>>
CurrentUserSingleton
- user: User
- myMoodList: MoodList
+ getInstance()
+ getUser() : user
+ getMyMoodList() : myMoodList
1
1
<<Model>>
MoodList
- listofMoods: ArrayList<Mood>
+ add (Mood mood)
+ delete (Mood mood)
+ getMood() : moodList
+ getListOfMoods() : listOfMoods
0...*
<<Controller>>
ElasticSearchUserController
+ addUserTask (User user)
+ getUserTask (String query): user
+ updateUserTask (String userID)
+ deleteUserTask (String userID)
+ verifySettings ()
<<Controller>>
ElasticSearchMoodController
+ addMoodsTask ()
+ getMoodsTask (String query): moodList [ ]
+ GetLatestMoodsTask()
+ updateMoodsTask (String moodID)
+ deleteMoodTask (String moodID)
+ verifySettings ()
<<Controller>>
Synchronize
+ synchronize (ArrayList addedMoodList, ArrayListeditedMoodList, ArrayList deletedMoodList)
ElasticSearchUserController modifies Request
ElasticSearchMoodController updates profile
ElasticSeatchMoodController updates Latest
AddMood sends information to ElasticSearchMoodController
<<Activity>>
Request
- username: String
- requestList(): ArrayList<String>
- getRequestList () : requestList[ ]
- accept (String username)
- decline (String username)
<<Activity>>
Login
- email: String
- username: String
- loginAccount (String username)
- signUp (String email, String username)
- forgotUsername (String email)
<<Activity>>
Friends
- searchTermForFriend: String
- searchTermForUser: String
- friendList: ArrayList<String>
- userList: ArrayList<String>
- searchFriend (String searchTermForFriend): friendList[ ]
- searchUser (String searchTermForUser): userList[ ]
<<Activity>>
Profile
username: String
moodList: ArrayList <Mood>
+ getMoodHistory () : moodList[ ]
<<Activity>>
Latest
- moodList: ArrayList<Mood>
- followList: ArrayList<String>
+ getLatestMoodList () : moodList[ ]
<<Abstract>>
ViewMood
username: String
emotion: String
datetime: Date
socialSituation: String
trigger: String
location: String
imageURL: String
- loadMood()
<<Activity>>
AddMood
username: String
emotion: String
datetime: Date
socialSituation: String
trigger: String
location: String
imageURL: String
- imageView: ImageView
- locationButton: Button
- locationManager: LocationManager
- locationText: TextView
- emotionAdapter: ArrayAdapter<String>
- situationAdapter: ArrayAdapter<String>
- editTrigger: EditText
- filter: InputFilter
- selectedEmotion: String
+ onCreateOptionsMenu(Menu menu)
+ onOptionsItemSelected(MenuItem item)
+ createMood(String emotion, String situation, String trigger)
+ encodeIMG(Uri uri)
+ acquireLocation()
+ chooseImage()
+ getPathFromURI(Uri contentUri): String
+ onActivityResult()
+ countWords(): int
<<Activity>>
Map
moodList: ArrayList<Mood>
+ displayLocationOnMap (ArrayList<Mood> moodList)
<<Activity>>
Menu
- goToMyProfileActivity ()
- goToFindFriendsActivity ()
- goToRequestsActivity ()
- goToFollowingActivity ()
- goToNearMeActivity ()
- logout ()
Menu is used to switch between Activities.
<<Activity>>
StrangerProfile
+ follow ()
<<Activity>>
AddStory
- createStory()
- sendStory()
<<Activity>>
MyProfile
- addMood()
<<Activity>>
FriendProfile
+ unfollow ()
<<Activity>>
ViewMyMood
+ deleteMood ()
+editMood()
<<Activity>>
ViewFriendMood
<<Activity>>
NearMeMap
- currentLocation: String
+ getCurrentLocation () : currentLocation
+ getMoodListWithinKm (ArrayList<Mood> moodList): moodListWithinKm
<<Controller>>
Filter
+ sortByDate(): moodList [ ]
+ sortByEmotion(Emotion emotion): moodList [ ]
+ filterKeyword(filterKeyword): moodList[ ]
<<Activity>>
EditMood
- saveChangesEditMood ()
- loadExistingMood()