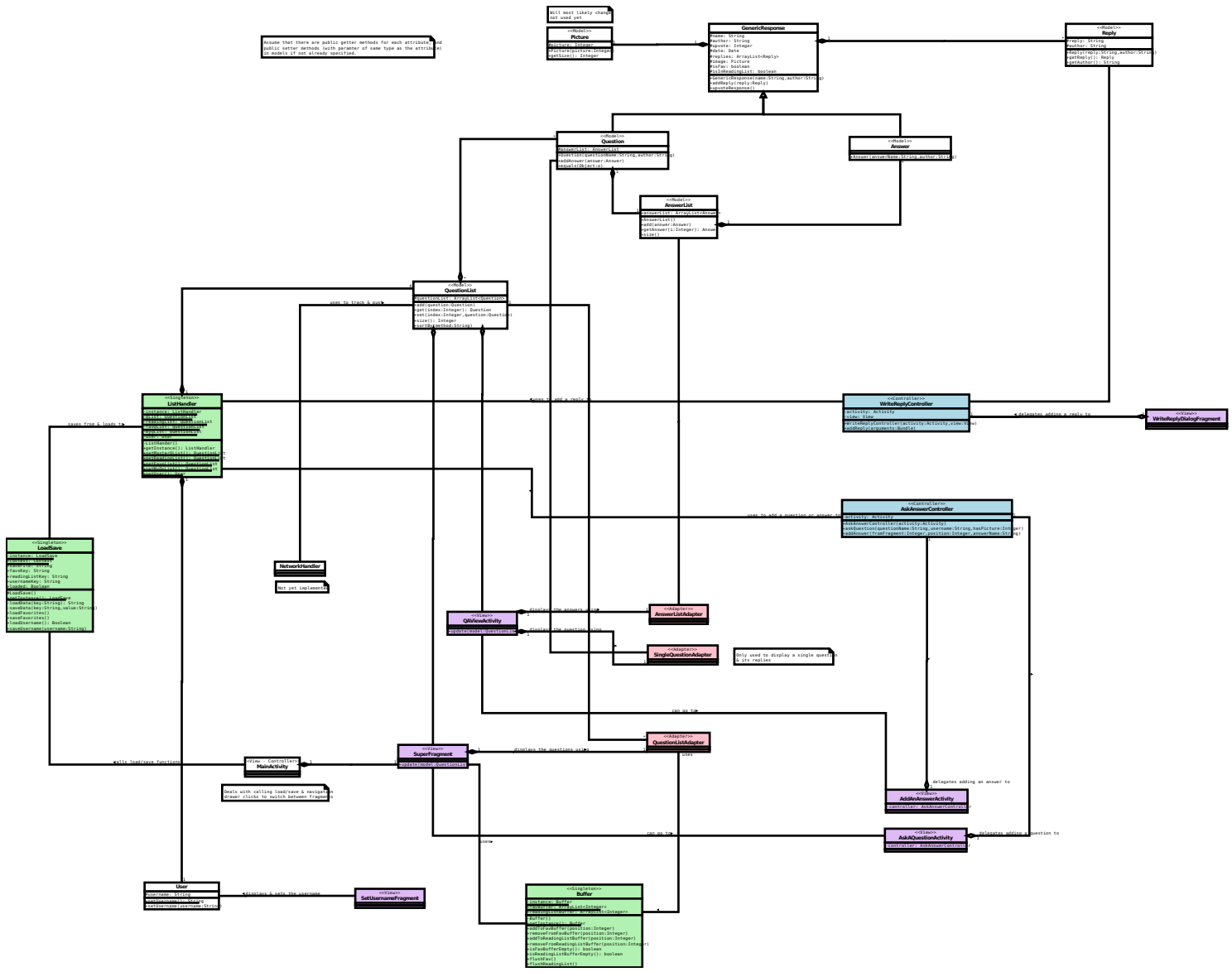
Assume that there are public getter methods for each attribute and public setter methods (with parameter of same type as the attribute) in models if not already specified.
Will most likely change, not used yet
<<Model>>
Picture
-Pictures: Integer
+getPicture(p:Integer)
+getSize(): Integer
GenericResponse
#name: String
#author: String
#date: Integer
#data: Date
#replies: ArrayList<Reply>
#image: Picture
#isFav: Boolean
#isInReadingList: boolean
+GenericResponse(name:String,author:String)
+addReply(reply:Reply)
+upvoteResponse()
<<Model>>
Reply
#reply: String
#author: String
+Reply(reply:String,author:String)
+getReply(): Reply
+getAuthor(): String
<<Model>>
Question
-answerList: AnswerList
+Question(questionName:String,author:String)
+addAnswer(answer:Answer)
+equals(Object o)
<<Model>>
Answer
+Answer(answerName:String,author:String)
<<Model>>
AnswerList
-answerList: ArrayList<Answer>
+AnswerList()
+add(answer:Answer)
+getAnswer(i:Integer): Answer
+size()
<<Model>>
QuestionList
-questionList: ArrayList<Question>
+add(question:Question)
+get(index:Integer): Question
+set(index:Integer,question:Question)
+size(): Integer
+isThere(author:String)
uses to track & sort
<<Singleton>>
ListHandler
-QuestionFile: String
-AnswerFile: String
-ReadingListFile: String
-FavFile: String
-QuestionList
-AnswerList
-ReadingList
-Fav: Map
+ListHandler()
+getInstance(): ListHandler
+getQuestionAt(i:Integer): Question
+getAnswerAt(i:Integer): Answer
+getReadingListAt(i:Integer): Integer
+getFavAt(i:Integer): Integer
+getSize(): Integer
saves from & loads to
uses to add a reply to
<<Controller>>
WriteReplyController
-activity: Activity
-View: View
+WriteReplyController(activity:Activity,view:View)
+addReply(arguments:Bundle)
delegates adding a reply to
<<View>>
WriteReplyDialogFragment
<<Controller>>
AskAnswerController
-Activity: Activity
+AskAnswerController(activity:Activity)
+askQuestion(questionName:String,username:String,hasPicture:Integer)
+addAnswer(fromFragment:Integer,position:Integer,answerName:String)
uses to add a question or answer to
<<Singleton>>
LoadSave
-Instance: LoadSave
-Activity: Activity
-SaveKey: String
+readingListKey: String
+usernameKey: String
+loaded: Boolean
+LoadSave()
+getInstance(): LoadSave
+loadData(key:String): String
+saveData(key:String,value:String)
+loadFavs()
+saveFavs()
+loadUsername(): Boolean
+saveUsername(username:String)
NetworkHandler
Not yet implemented
<<View>>
QAViewActivity
-answerReadingQuestionAt(i:Integer)
displays the answers using
displays the questions using
<<Adapter>>
AnswerListAdapter
<<Adapter>>
SingleQuestionAdapter
Only used to display a single question & its replies
can go to
<<View>>
SuperFragment
-updateUnder(QuestionList)
displays the questions using
<<Adapter>>
QuestionListAdapter
uses
<View - Controller>
MainActivity
with load/save functions
Deals with calling load/save & navigation drawer clicks to switch between fragments
delegates adding an answer to
<<View>>
AddAnAnswerActivity
-controller: AskAnswerController
can go to
<<View>>
AskAQuestionActivity
-controller: AskAnswerController
delegates adding a question to
User
-username: String
-instance: User
+setUsernameAt(username:String)
displays & sets the username
<<View>>
SetUsernameFragment
<<Singleton>>
Buffer
-instance: Buffer
-readingBuffer: List<Integer>
-favBuffer: ArrayList<Integer>
+Buffer()
+getInstance(): Buffer
+addToFavBuffer(position:Integer)
+removeFromFavBuffer(position:Integer)
+addToReadingListBuffer(position:Integer)
+removeFromReadingListBuffer(position:Integer)
+isFavBufferEmpty(): boolean
+isReadingListBufferEmpty(): boolean
+flushFav()
+flushReadingList()