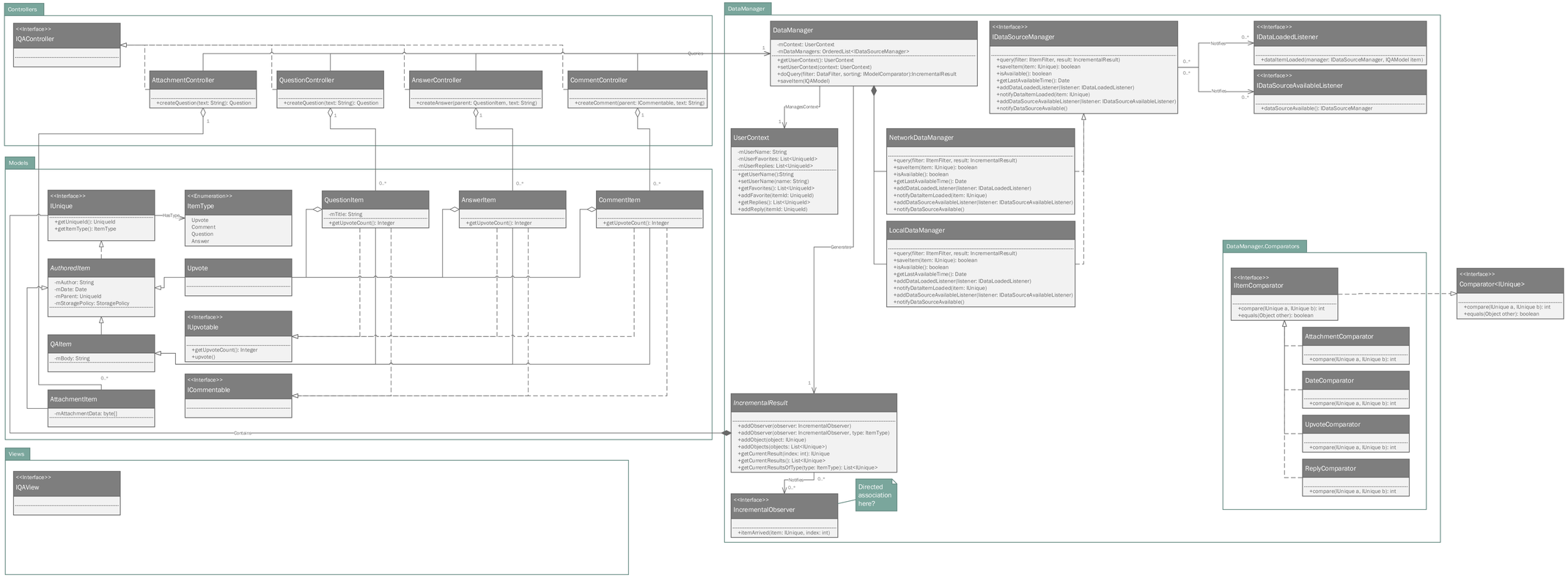
Controllers
<<Interface>>
IQAController
AttachmentController
+createQuestion(text: String): Question
QuestionController
+createQuestion(text: String): Question
AnswerController
+createAnswer(parent: QuestionItem, text: String)
CommentController
+createComment(parent: ICommentable, text: String)
Queries
1
Models
<<Interface>>
IUnique
+getUniqueId(): UniqueId
+getItemType(): ItemType
HasType
<<Enumeration>>
ItemType
Upvote
Comment
Question
Answer
QuestionItem
-mTitle: String
+getUpvoteCount(): Integer
AnswerItem
+getUpvoteCount(): Integer
CommentItem
+getUpvoteCount(): Integer
0..*
AuthoredItem
-mAuthor: String
-mDate: Date
-mParent: UniqueId
-mStoragePolicy: StoragePolicy
Upvote
<<Interface>>
IUpvotable
+getUpvoteCount(): Integer
+upvote()
QAItem
-mBody: String
AttachmentItem
-mAttachmentData: byte[]
<<Interface>>
ICommentable
Contains
Views
<<Interface>>
IQAView
DataManager
DataManager
-mContext: UserContext
-mDataManagers: OrderedList<IDataSourceManager>
+getUserContext(): UserContext
+setUserContext(context: UserContext)
+doQuery(filter: DataFilter, sorting: IModelComparator):IncrementalResult
+saveItem(IQAModel)
ManagesContext
UserContext
-mUserName: String
-mUserFavorites: List<UniqueId>
-mUserReplies: List<UniqueId>
+getUserName():String
+setUserName(name: String)
+getFavorites(): List<UniqueId>
+addFavorite(itemId: UniqueId)
+getReplies(): List<UniqueId>
+addReply(itemId: UniqueId)
<<Interface>>
IDataSourceManager
+query(filter: IItemFilter, result: IncrementalResult)
+saveItem(item: IUnique): boolean
+isAvailable(): boolean
+getLastAvailableTime(): Date
+addDataLoadedListener(listener: IDataLoadedListener)
+notifyDataItemLoaded(item: IUnique)
+addDataSourceAvailableListener(listener: IDataSourceAvailableListener)
+notifyDataSourceAvailable()
Notifies
<<Interface>>
IDataLoadedListener
+dataItemLoaded(manager: IDataSourceManager, IQAModel item)
<<Interface>>
IDataSourceAvailableListener
+dataSourceAvailable(): IDataSourceManager
NetworkDataManager
+query(filter: IItemFilter, result: IncrementalResult)
+saveItem(item: IUnique): boolean
+isAvailable(): boolean
+getLastAvailableTime(): Date
+addDataLoadedListener(listener: IDataLoadedListener)
+notifyDataItemLoaded(item: IUnique)
+addDataSourceAvailableListener(listener: IDataSourceAvailableListener)
+notifyDataSourceAvailable()
LocalDataManager
+query(filter: IItemFilter, result: IncrementalResult)
+saveItem(item: IUnique): boolean
+isAvailable(): boolean
+getLastAvailableTime(): Date
+addDataLoadedListener(listener: IDataLoadedListener)
+notifyDataItemLoaded(item: IUnique)
+addDataSourceAvailableListener(listener: IDataSourceAvailableListener)
+notifyDataSourceAvailable()
Generates
IncrementalResult
+addObserver(observer: IncrementalObserver)
+addObserver(observer: IncrementalObserver, type: ItemType)
+addObject(object: IUnique)
+addObjects(objects: List<IUnique>)
+getCurrentResult(index: int): IUnique
+getCurrentResults(): List<IUnique>
+getCurrentResultsOfType(type: ItemType): List<IUnique>
<<Interface>>
IncrementalObserver
+itemArrived(item: IUnique, index: int)
Directed association here?
DataManager.Comparators
<<Interface>>
IItemComparator
+compare(IUnique a, IUnique b): int
+equals(Object other): boolean
AttachmentComparator
+compare(IUnique a, IUnique b): int
DateComparator
+compare(IUnique a, IUnique b): int
UpvoteComparator
+compare(IUnique a, IUnique b): int
ReplyComparator
+compare(IUnique a, IUnique b): int
<<Interface>>
Comparator<IUnique>
+compare(IUnique a, IUnique b): int
+equals(Object other): boolean