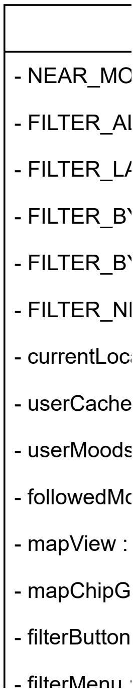
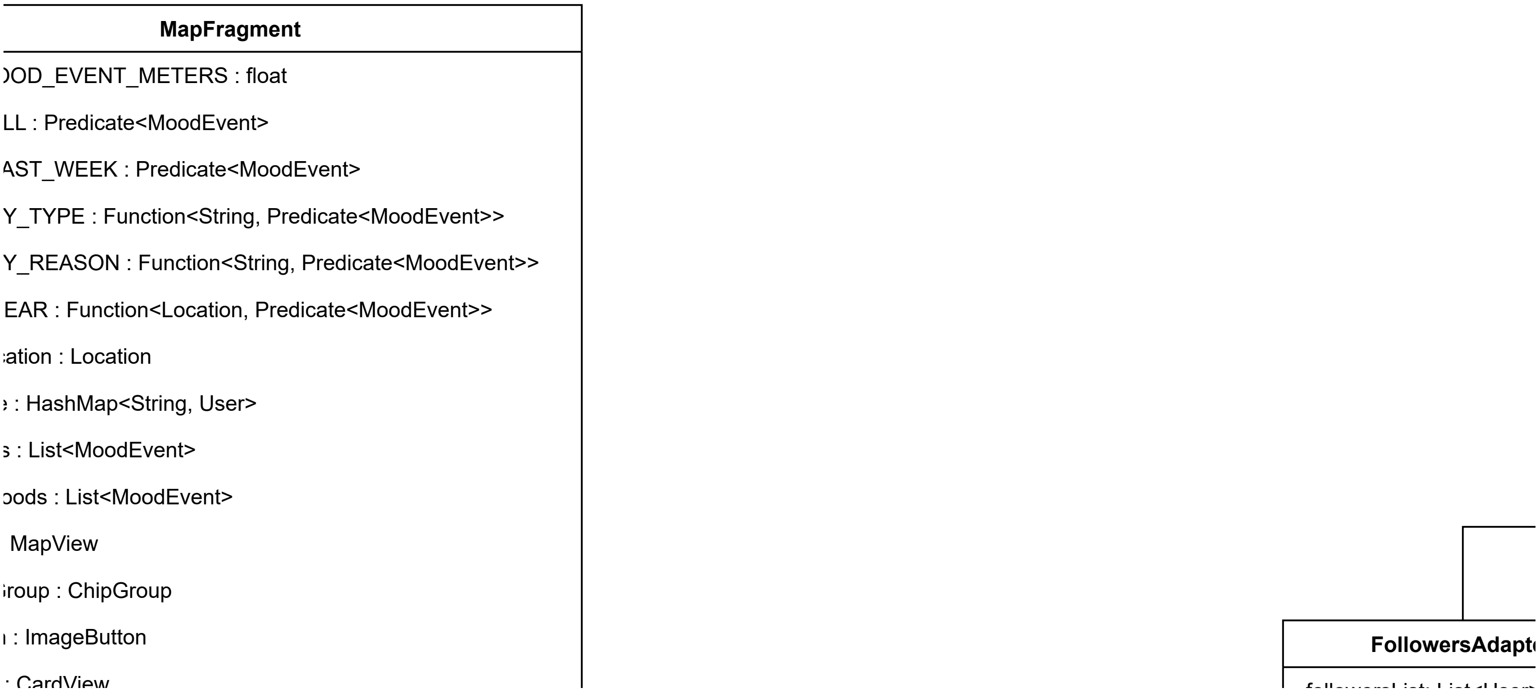

- NEAR_MC
- FILTER_AI
- FILTER_LA
- FILTER_BY
- FILTER_BY
- FILTER_NI
- currentLoca
- userCache
- userMoods
- followedMc
- mapView :
- mapChipG
- filterButton
- filterMenu



MapFragment

OOD_EVENT_METERS : float

LL : Predicate<MoodEvent>

AST_WEEK : Predicate<MoodEvent>

Y_TYPE : Function<String, Predicate<MoodEvent>>

Y_REASON : Function<String, Predicate<MoodEvent>>

EAR : Function<Location, Predicate<MoodEvent>>

ation : Location

e : HashMap<String, User>

s : List<MoodEvent>

oods : List<MoodEvent>

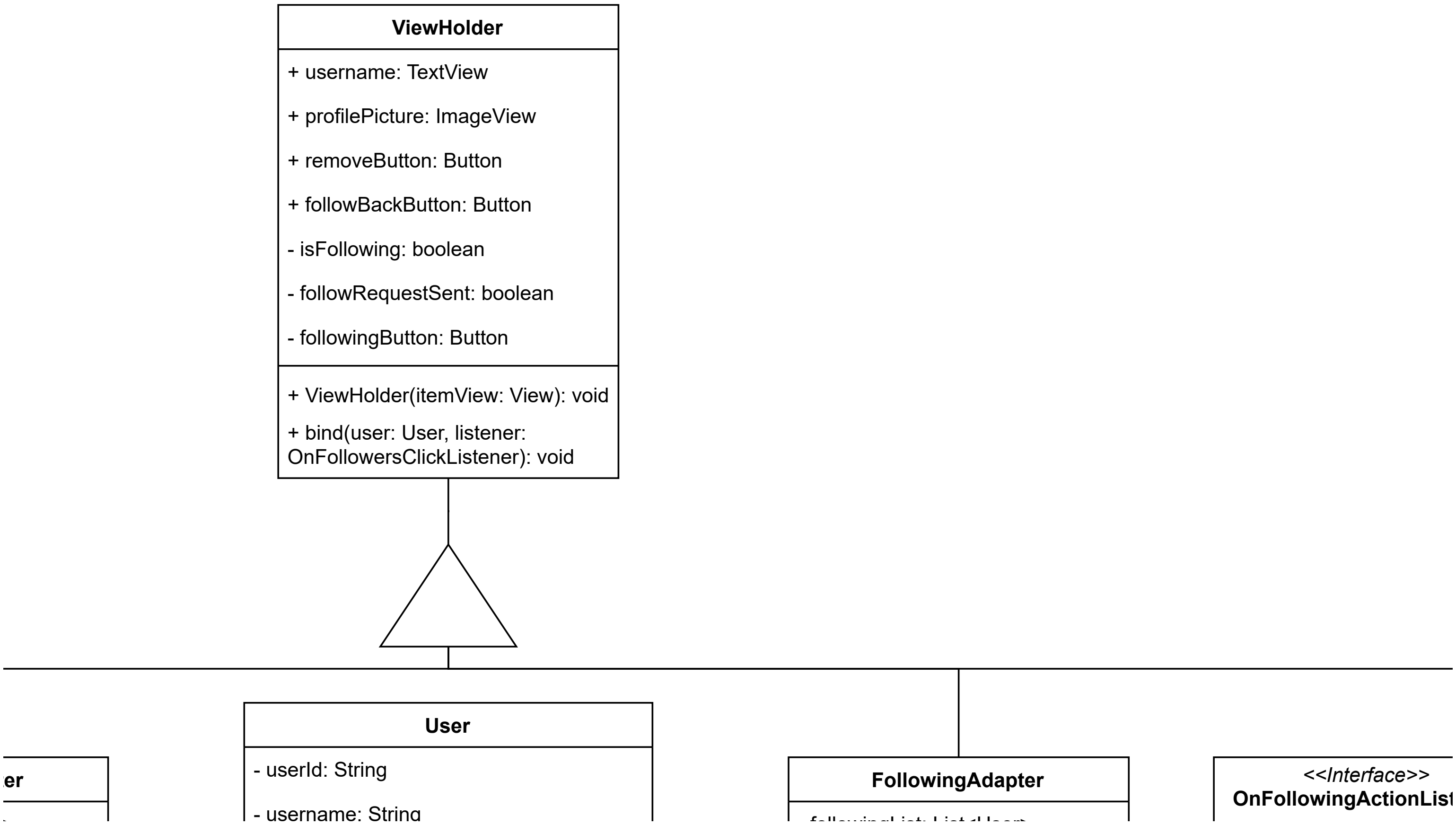
MapView

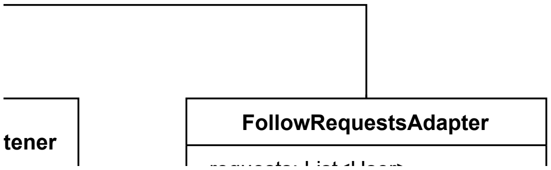
roup : ChipGroup

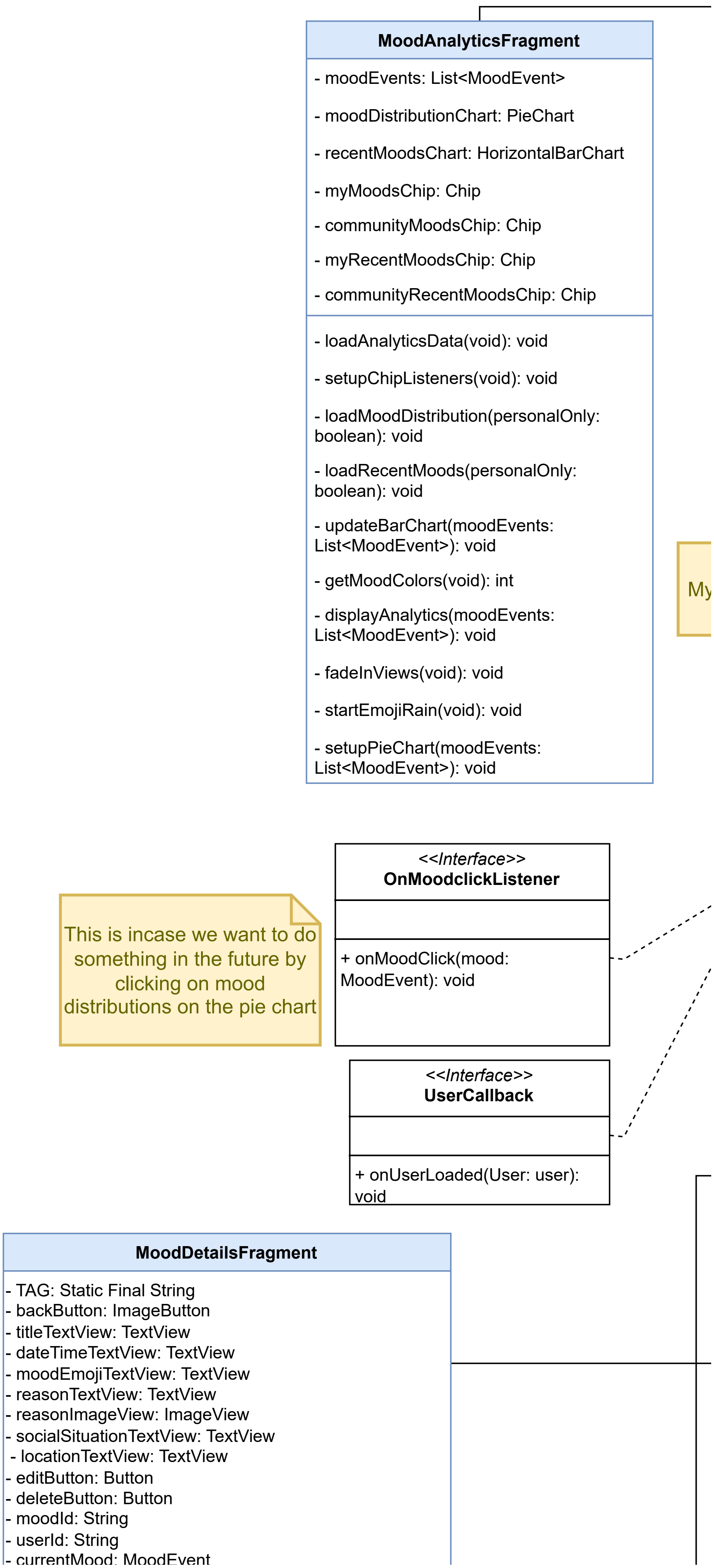
: ImageButton

: CardView

FollowersAdapt


ViewHolder
+ username: TextView
+ profilePicture: ImageView
+ removeButton: Button
+ followBackButton: Button
- isFollowing: boolean
- followRequestSent: boolean
- followingButton: Button
+ ViewHolder(itemView: View): void
+ bind(user: User, listener: OnFollowersClickListener): void
er
User
- userId: String
- username: String
FollowingAdapter
followingList: List<User>
<<Interface>>
OnFollowingActionList


tener
FollowRequestsAdapter
requests: List<User>



MoodAnalyticsFragment

- moodEvents: List<MoodEvent>

- moodDistributionChart: PieChart

- recentMoodsChart: HorizontalBarChart

- myMoodsChip: Chip

- communityMoodsChip: Chip

- myRecentMoodsChip: Chip

- communityRecentMoodsChip: Chip

- loadAnalyticsData(void): void

- setupChipListeners(void): void

- loadMoodDistribution(personalOnly: boolean): void

- loadRecentMoods(personalOnly: boolean): void

- updateBarChart(moodEvents: List<MoodEvent>): void

- getMoodColors(void): int

- displayAnalytics(moodEvents: List<MoodEvent>): void

- fadeInViews(void): void

- startEmojiRain(void): void

- setupPieChart(moodEvents: List<MoodEvent>): void

<<Interface>>
OnMoodclickListener

+ onMoodClick(mood: MoodEvent): void

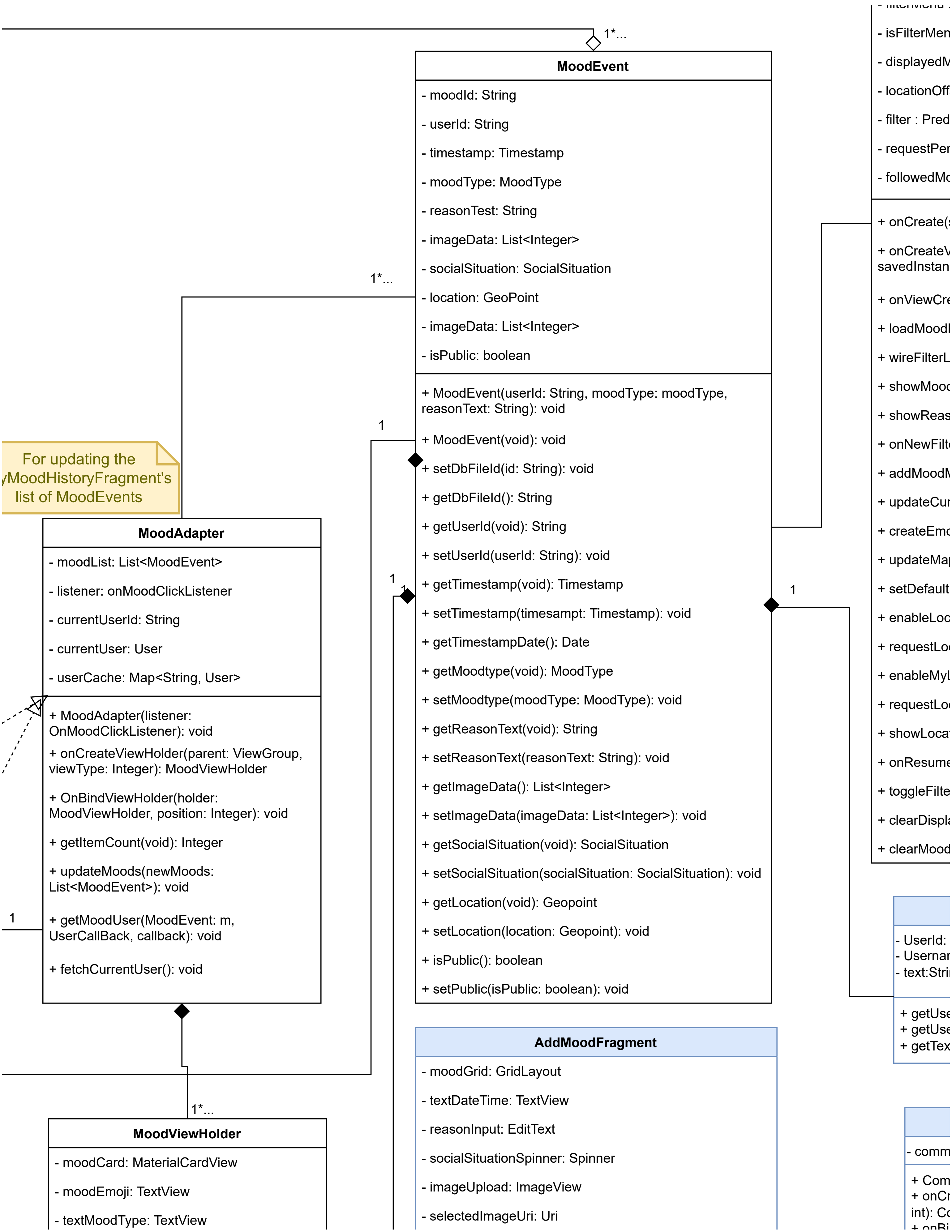
This is incase we want to do something in the future by clicking on mood distributions on the pie chart

<<Interface>>
UserCallback

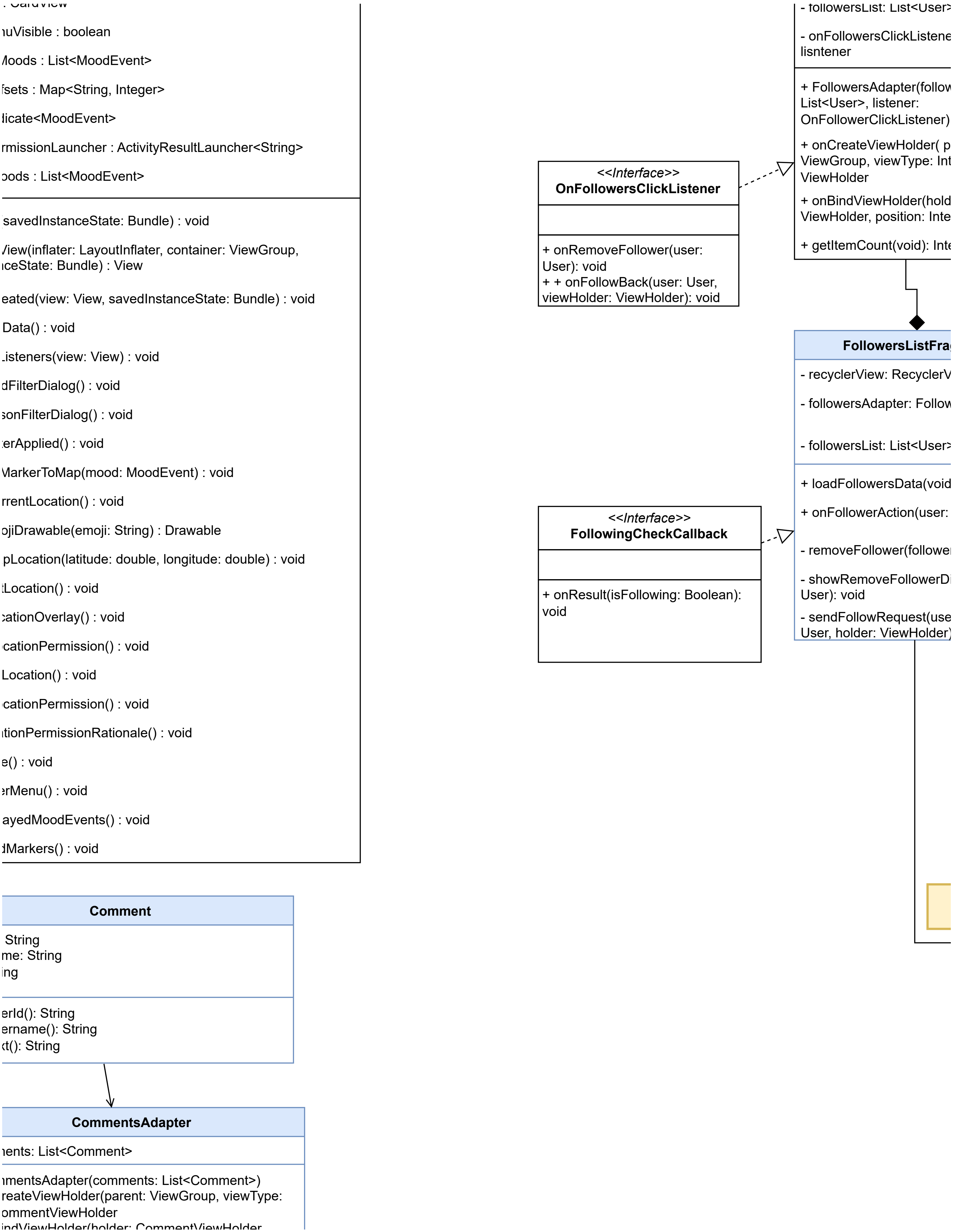
+ onUserLoaded(User: user): void

MoodDetailsFragment

- TAG: Static Final String
- backButton: ImageButton
- titleTextView: TextView
- dateTimeTextView: TextView
- moodEmojiTextView: TextView
- reasonTextView: TextView
- reasonImageView: ImageView
- socialSituationTextView: TextView
- locationTextView: TextView
- editButton: Button
- deleteButton: Button
- moodId: String
- userId: String
- currentMood: MoodEvent



1*...
MoodEvent
- moodId: String
- userId: String
- timestamp: Timestamp
- moodType: MoodType
- reasonTest: String
- imageData: List<Integer>
- socialSituation: SocialSituation
- location: GeoPoint
- imageData: List<Integer>
- isPublic: boolean
+ MoodEvent(userId: String, moodType: moodType, reasonText: String): void
+ MoodEvent(void): void
+ setDbFileId(id: String): void
+ getDbFileId(): String
+ getUserId(void): String
+ setUserId(userId: String): void
+ getTimestamp(void): Timestamp
+ setTimestamp(timesampt: Timestamp): void
+ getTimestampDate(): Date
+ getMoodtype(void): MoodType
+ setMoodtype(moodType: MoodType): void
+ getReasonText(void): String
+ setReasonText(reasonText: String): void
+ getImageData(): List<Integer>
+ setImageData(imageData: List<Integer>): void
+ getSocialSituation(void): SocialSituation
+ setSocialSituation(socialSituation: SocialSituation): void
+ getLocation(void): Geopoint
+ setLocation(location: Geopoint): void
+ isPublic(): boolean
+ setPublic(isPublic: boolean): void
1*...
1
1
1
1
For updating the yMoodHistoryFragment's list of MoodEvents
MoodAdapter
- moodList: List<MoodEvent>
- listener: onMoodClickListener
- currentUserId: String
- currentUser: User
- userCache: Map<String, User>
+ MoodAdapter(listener: OnMoodClickListener): void
+ onCreateViewHolder(parent: ViewGroup, viewType: Integer): MoodViewHolder
+ OnBindViewHolder(holder: MoodViewHolder, position: Integer): void
+ getItemCount(void): Integer
+ updateMoods(newMoods: List<MoodEvent>): void
+ getMoodUser(MoodEvent: m, UserCallBack, callback): void
+ fetchCurrentUser(): void
1
1*...
MoodViewHolder
- moodCard: MaterialCardView
- moodEmoji: TextView
- textMoodType: TextView
AddMoodFragment
- moodGrid: GridLayout
- textDateTime: TextView
- reasonInput: EditText
- socialSituationSpinner: Spinner
- imageUpload: ImageView
- selectedImageUri: Uri
- filterMenu
- isFilterMen
- displayedM
- locationOff
- filter : Pred
- requestPer
- followedMo
+ onCreate(
+ onCreateV savedInstan
+ onViewCre
+ loadMoodI
+ wireFilterL
+ showMood
+ showReas
+ onNewFilt
+ addMoodM
+ updateCur
+ createEmo
+ updateMa
+ setDefault
+ enableLoc
+ requestLo
+ enableMyI
+ requestLo
+ showLoca
+ onResume
+ toggleFilte
+ clearDispl
+ clearMood
- UserId:
- Usernar
- text:Stri
+ getUse
+ getUse
+ getTex
- comm
+ Com
+ onCr
int): Co
+ onBi



: CardView

nuVisible : boolean

/loods : List<MoodEvent>

fsets : Map<String, Integer>

licate<MoodEvent>

rmissionLauncher : ActivityResultLauncher<String>

oods : List<MoodEvent>

savedInstanceState: Bundle) : void

/iew(inflater: LayoutInflater, container: ViewGroup,
iceState: Bundle) : View

eated(view: View, savedInstanceState: Bundle) : void

Data() : void

Listeners(view: View) : void

dFilterDialog() : void

sonFilterDialog() : void

erApplied() : void

MarkerToMap(mood: MoodEvent) : void

rrentLocation() : void

ojiDrawable(emoji: String) : Drawable

pLocation(latitude: double, longitude: double) : void

tLocation() : void

cationOverlay() : void

cationPermission() : void

Location() : void

cationPermission() : void

tionPermissionRationale() : void

e() : void

erMenu() : void

ayedMoodEvents() : void

dMarkers() : void

Comment

String
me: String
ing

erId(): String
ername(): String
xt(): String

CommentsAdapter

nents: List<Comment>

mentsAdapter(comments: List<Comment>)
reateViewHolder(parent: ViewGroup, viewType:
ommentViewHolder
indViewHolder(holder: CommentViewHolder

<<Interface>>
OnFollowersClickListener

+ onRemoveFollower(user: User): void
+ + onFollowBack(user: User, viewHolder: ViewHolder): void

<<Interface>>
FollowingCheckCallback

+ onResult(isFollowing: Boolean): void

- followersList: List<User>

- onFollowersClickListene
lisntener

+ FollowersAdapter(follow
List<User>, listener:
OnFollowerClickListener)

+ onCreateViewHolder(p
ViewGroup, viewType: Int
ViewHolder

+ onBindViewHolder(hold
ViewHolder, position: Inte

+ getItemCount(void): Inte

FollowersListFra

- recyclerView: RecyclerV

- followersAdapter: Follow

- followersList: List<User>

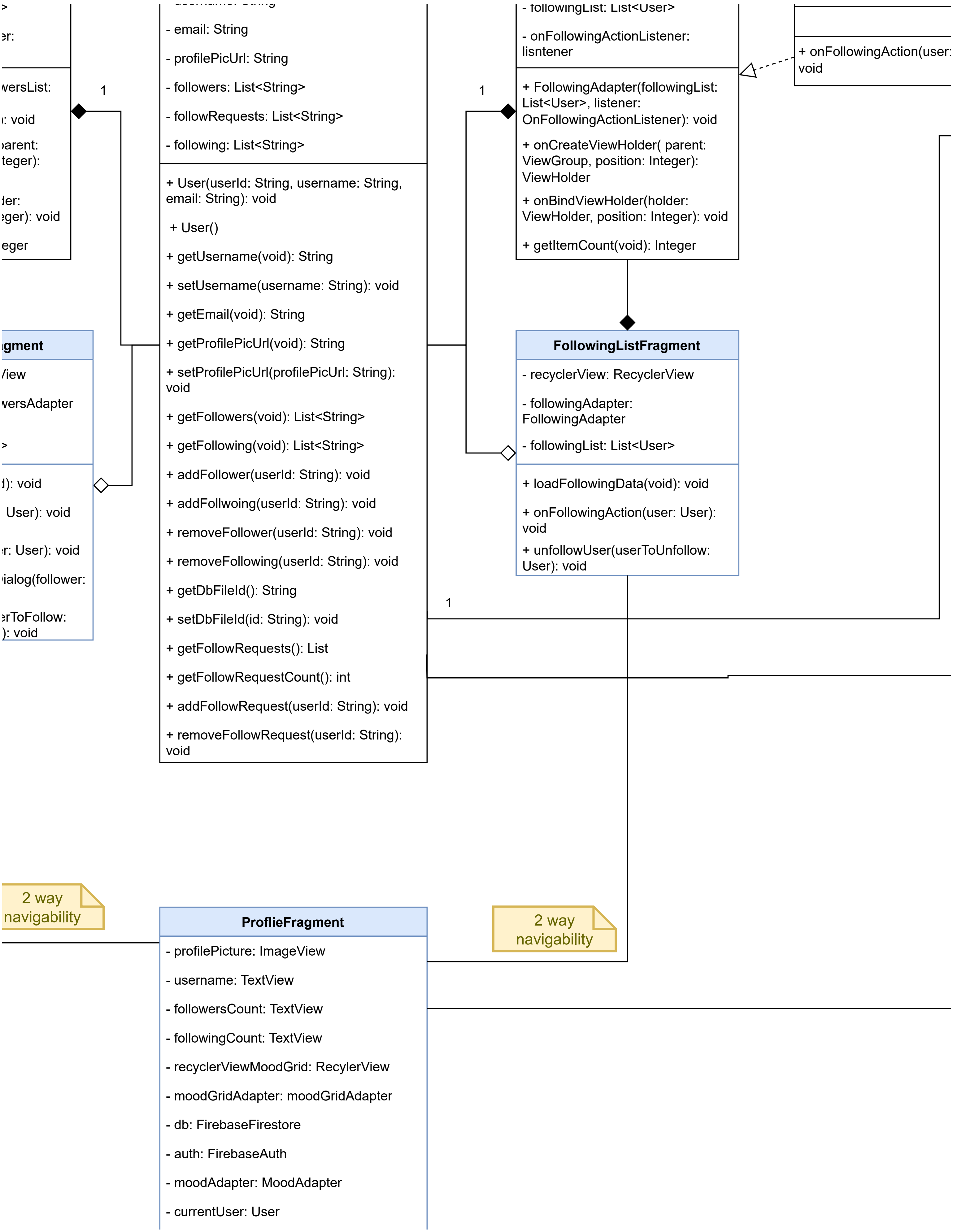
+ loadFollowersData(void

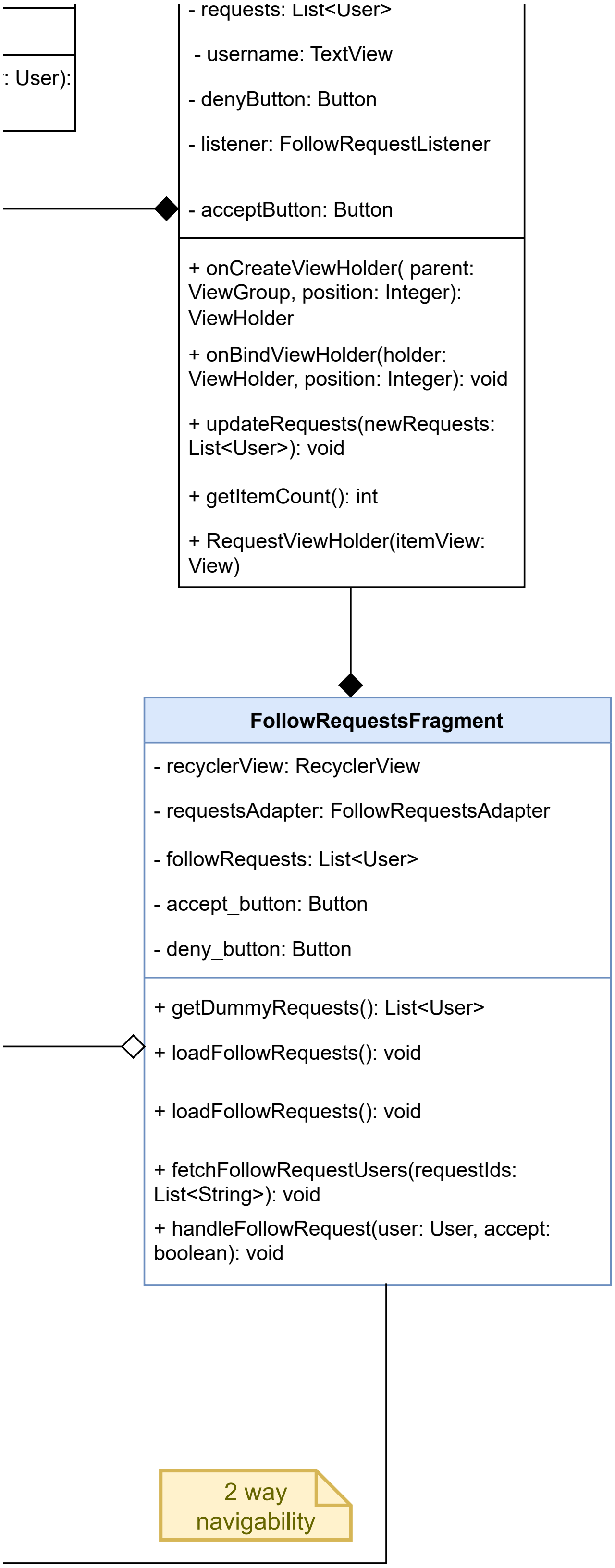
+ onFollowerAction(user:

- removeFollower(followe

- showRemoveFollowerD
User): void

- sendFollowRequest(use
User, holder: ViewHolder)


>
er:
wersList:
: void
parent:
teger):
der:
eger): void
eger
1
- email: String
- profilePicUrl: String
- followers: List<String>
- followRequests: List<String>
- following: List<String>
+ User(userId: String, username: String, email: String): void
+ User()
+ getUsername(void): String
+ setUsername(username: String): void
+ getEmail(void): String
+ getProfilePicUrl(void): String
+ setProfilePicUrl(profilePicUrl: String): void
+ getFollowers(void): List<String>
+ getFollowing(void): List<String>
+ addFollower(userId: String): void
+ addFollwoing(userId: String): void
+ removeFollower(userId: String): void
+ removeFollowing(userId: String): void
+ getDbFileId(): String
+ setDbFileId(id: String): void
+ getFollowRequests(): List
+ getFollowRequestCount(): int
+ addFollowRequest(userId: String): void
+ removeFollowRequest(userId: String): void
- followingList: List<User>
- onFollowingActionListener: lisntener
+ FollowingAdapter(followingList: List<User>, listener: OnFollowingActionListener): void
+ onCreateViewHolder(parent: ViewGroup, position: Integer): ViewHolder
+ onBindViewHolder(holder: ViewHolder, position: Integer): void
+ getItemCount(void): Integer
1
+ onFollowingAction(user: void
gment
View
wersAdapter
>
d): void
User): void
r: User): void
ialog(follower:
erToFollow:
): void
FollowingListFragment
- recyclerView: RecyclerView
- followingAdapter: FollowingAdapter
- followingList: List<User>
+ loadFollowingData(void): void
+ onFollowingAction(user: User): void
+ unfollowUser(userToUnfollow: User): void
1
2 way navigability
ProflieFragment
- profilePicture: ImageView
- username: TextView
- followersCount: TextView
- followingCount: TextView
- recyclerViewMoodGrid: RecylerView
- moodGridAdapter: moodGridAdapter
- db: FirebaseFirestore
- auth: FirebaseAuth
- moodAdapter: MoodAdapter
- currentUser: User
2 way navigability



- requests: List<User>

- username: TextView

- denyButton: Button

- listener: FollowRequestListener

- acceptButton: Button

+ onCreateViewHolder(parent: ViewGroup, position: Integer): ViewHolder

+ onBindViewHolder(holder: ViewHolder, position: Integer): void

+ updateRequests(newRequests: List<User>): void

+ getItemCount(): int

+ RequestViewHolder(itemView: View)

: User):

FollowRequestsFragment

- recyclerView: RecyclerView

- requestsAdapter: FollowRequestsAdapter

- followRequests: List<User>

- accept_button: Button

- deny_button: Button

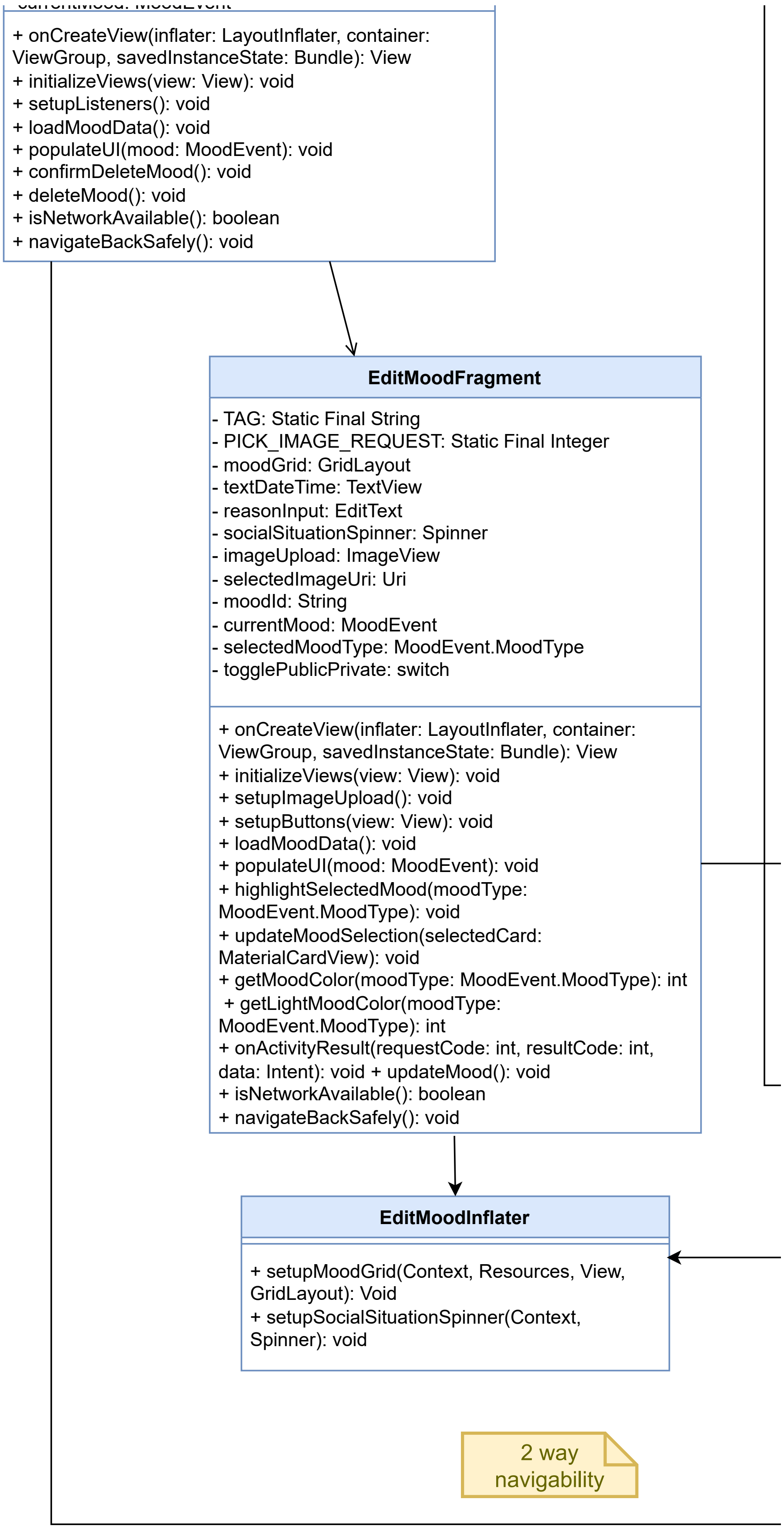
+ getDummyRequests(): List<User>

+ loadFollowRequests(): void

+ loadFollowRequests(): void

+ fetchFollowRequestUsers(requestIds: List<String>): void

+ handleFollowRequest(user: User, accept: boolean): void



- currentMood: MoodEvent

+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View
+ initializeViews(view: View): void
+ setupListeners(): void
+ loadMoodData(): void
+ populateUI(mood: MoodEvent): void
+ confirmDeleteMood(): void
+ deleteMood(): void
+ isNetworkAvailable(): boolean
+ navigateBackSafely(): void

EditMoodFragment

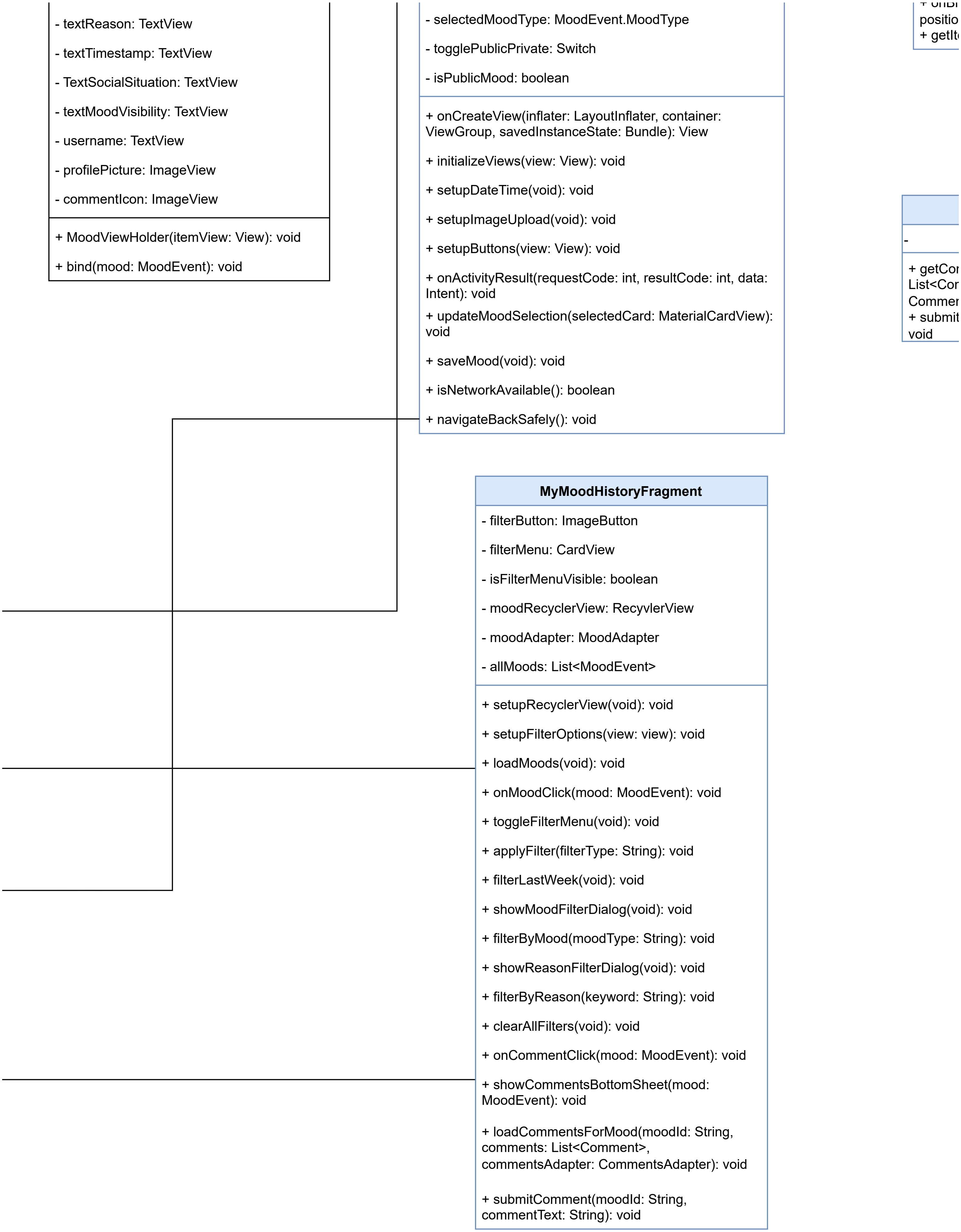
- TAG: Static Final String
- PICK_IMAGE_REQUEST: Static Final Integer
- moodGrid: GridLayout
- textDateTime: TextView
- reasonInput: EditText
- socialSituationSpinner: Spinner
- imageUpload: ImageView
- selectedImageUri: Uri
- moodId: String
- currentMood: MoodEvent
- selectedMoodType: MoodEvent.MoodType
- togglePublicPrivate: switch

+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View
+ initializeViews(view: View): void
+ setupImageUpload(): void
+ setupButtons(view: View): void
+ loadMoodData(): void
+ populateUI(mood: MoodEvent): void
+ highlightSelectedMood(moodType: MoodEvent.MoodType): void
+ updateMoodSelection(selectedCard: MaterialCardView): void
+ getMoodColor(moodType: MoodEvent.MoodType): int
+ getLightMoodColor(moodType: MoodEvent.MoodType): int
+ onActivityResult(requestCode: int, resultCode: int, data: Intent): void + updateMood(): void
+ isNetworkAvailable(): boolean
+ navigateBackSafely(): void

EditMoodInflater

+ setupMoodGrid(Context, Resources, View, GridLayout): Void
+ setupSocialSituationSpinner(Context, Spinner): void

2 way navigability



- textReason: TextView

- textTimestamp: TextView

- TextSocialSituation: TextView

- textMoodVisibility: TextView

- username: TextView

- profilePicture: ImageView

- commentIcon: ImageView

+ MoodViewHolder(itemView: View): void

+ bind(mood: MoodEvent): void

- selectedMoodType: MoodEvent.MoodType

- togglePublicPrivate: Switch

- isPublicMood: boolean

+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View

+ initializeViews(view: View): void

+ setupDateTime(void): void

+ setupImageUpload(void): void

+ setupButtons(view: View): void

+ onActivityResult(requestCode: int, resultCode: int, data: Intent): void

+ updateMoodSelection(selectedCard: MaterialCardView): void

+ saveMood(void): void

+ isNetworkAvailable(): boolean

+ navigateBackSafely(): void

+ onBi
positio
+ getIt

-

+ getCor
List<Cor
Commer
+ submit
void

MyMoodHistoryFragment

- filterButton: ImageButton

- filterMenu: CardView

- isFilterMenuVisible: boolean

- moodRecyclerView: RecyvlerView

- moodAdapter: MoodAdapter

- allMoods: List<MoodEvent>

+ setupRecyclerView(void): void

+ setupFilterOptions(view: view): void

+ loadMoods(void): void

+ onMoodClick(mood: MoodEvent): void

+ toggleFilterMenu(void): void

+ applyFilter(filterType: String): void

+ filterLastWeek(void): void

+ showMoodFilterDialog(void): void

+ filterByMood(moodType: String): void

+ showReasonFilterDialog(void): void

+ filterByReason(keyword: String): void

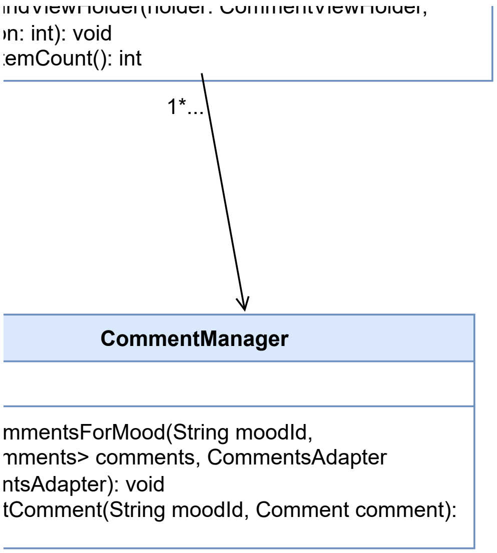
+ clearAllFilters(void): void

+ onCommentClick(mood: MoodEvent): void

+ showCommentsBottomSheet(mood: MoodEvent): void

+ loadCommentsForMood(moodId: String, comments: List<Comment>, commentsAdapter: CommentsAdapter): void

+ submitComment(moodId: String, commentText: String): void


indViewHolder(holder: CommentViewHolder,
on: int): void
temCount(): int
1*...
CommentManager
mmentsForMood(String moodId,
mments> comments, CommentsAdapter
ntsAdapter): void
tComment(String moodId, Comment comment):

- moodEvents: List<MoodEvent>

- filterButton: ImageButton

- isFilterMenuVisible: boolean

- searchResultsRecyclerView: RecyclerView

- searchResultAdapter: SearchResultAdapter

- searchEditText: EditText

- PICK_IMAGE_REQUEST: int

- searchButton: ImageButton

- headerSearchButton: ImageButton

- searchResultsCard: CardView

- filterMenu: CardView

- searchSection: LinearLayout

- moodEvents: List<MoodEvent>

- moodRecyclerView: RecyclerView

- followersSection: View

- followingSection: View

+ onCreateView(inflater: LayoutInflater, container: ViewGroup, savedInstanceState: Bundle): View
+ onCreate(savedInstanceState: Bundle): void
+ onCreateOptionsMenu(menu: Menu, inflater: MenuInflater): void
+ onOptionsItemSelected(item: MenuItem): boolean
+ fetchCurrentUser(): void
+ setUserData(): void
+ loadFollowerAndFollowingCounts(): void
+ setupRecyclerView(): void
+ loadMoodEvents(): void
+ onMoodClick(mood: MoodEvent): void
+ onCommentClick(mood: MoodEvent): void
+ showCommentsBottomSheet(mood: MoodEvent): void
+ loadCommentsForMood(moodId: String, comments: List<Comment>, commentsAdapter: CommentsAdapter): void
+ submitComment(moodId: String, commentText: String): void
+ navigateToFollowersList(): void
+ navigateToFollowingList(): void
+ logoutUser(): void
+ openImagePicker(): void
+ onActivityResult(requestCode: int, resultCode: int, data: Intent): void
+ uploadProfilePicture(imageUri: Uri): void
+ toggleFilterMenu(): void
+ setupFilterOptions(view: View): void
+ filterLastWeek(): void
+ showMoodFilterDialog(): void
+ filterByMood(moodType: String): void
+ showReasonFilterDialog(): void
+ filterByReason(keyword: String): void
+ clearAllFilters(): void
+ performSearch(query: String): void
+ searchExactUsername(username: String): void