

RecordModel

- recordID : String

- timstamp: Date

- photos: ArrayList<Bitmap>

- bodyLocation: BodyLocation

- mapLocation: LatLng

- comment: String

- description: String

- title: String

+ getTimestamp(): Date

+ setTimestamp(Date s)

+ addPhoto(Bitmap b)

+ removePhoto(Bitmap b): Bitmap

+ getPhotos(): ArrayList<Bitmap>

+ getMapLocation(): LatLng

+ setMapLocation(LatLng I)

+ getTitle(): String

+ setTitle(String t)

+ setBodyLocation(BodyLocation b)

+ getBodyLocation(): BodyLocation

+ getComment():String

+ setComment(String c)

+ setDescription(String d)

+ getDescription(): String

+ toString() : String

The body location is a data class that holds an element from the enumerable list of all body parts.

BodyLocation

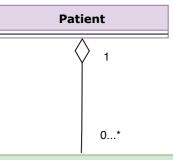
- bodyPart: EBodyPart

+ getBodyPart() : EBodyPart

+ setBodyPart(bodyPart:EBodyPart) : void

<<Enum>>
EBodyPart

- bodyPart: EBodyPart



ProblemModel

- problemID : String

-records: ArrayList<RecordModel>

-title: String

-description: String

-started: DateTime

+ ProblemModel(title:String, description:String)

+ setDescription(description: String)

+ getDescription: String

+ setTitle(title: String)

+ getTitle(): String

+ getRecord(index: integer)

+ getRecords(): ArrayList<RecordModel>

+ setRecords(records:ArrayList<RecordModel>: void

VOIC

+ removeRecord(record: RecordModel):

Boolean

+ removeRecord(index:int) : RecordModel

+ addRecord(record:RecordModel)

+ getDateStarted(): Date

+ toString(): String

ProblemController

 \Diamond

RecordModel

- recordID : String

- timstamp: Date

- photos: ArrayList<Bitmap>

- bodyLocation: BodyLocation

- mapLocation: LatLng

- comment: String

- description: String

- title: String

0..*

+ getTimestamp(): Date

+ setTimestamp(Date s)

+ addPhoto(Bitmap b)

+ removePhoto(Bitmap b): Bitmap

+ getPhotos(): ArrayList<Bitmap>

+ getMapLocation(): LatLng

+ setMapLocation(LatLng I)

+ getTitle(): String

+ setTitle(String t)

+ setBodyLocation(BodyLocation b)

+ getBodyLocation(): BodyLocation

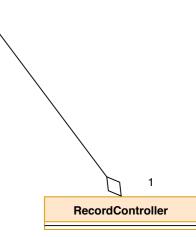
+ getComment():String

+ setComment(String c)

+ setDescription(String d)

+ getDescription(): String

+ toString() : String



ProblemActivityCreation ProblemModel consistency : String **ProblemListFragment** - saveProblem(): void - localList: ArrayList<ProblemModel> - createNotificationChannel(): void - problemListAdapter: ArrayAdapter<ProblemModel> + scheduleNotification(context : Context, delay : long, notificationId : int) - controllerInstance : ProblemController : void + update(): void **ProblemController** -<static> instance : ProblemController -<static> selectedProblem : ProblemModel **PhotoGalleryActivity** - userInstance : UserProfileController · imageView : ImageView[] 1 + problems : ArrayList<ProblemModel> bitmapArray : Bitmap[] + createNewProblem(title: String, description: String): void - counter : int + editProblem(indec : int, newTitle : String, newDescription : String) : void EXTRA_MESSAGE : String + loadProblemData(): void - EXTRA PHOTO: String + saveProblemData(problems : ArrayList<ProblemModel>) : void - imageGet(size : int) : void + setSelectedProblem(index : int) : void imageLoader(bmp : Bitmap) : void + getSelectedProblemRecords(): ArrayList<RecordModel> + getSelectedProblem(): ProblemModel + UpdateSelectedProblemRecord(rm : RecordModel, idx : int) : void + AddSelectedProblemRecordPhoto(b: Bitmap, idx: int): void **ProblemViewActivity PhotoSlideshowPopup** photoList: Collection<PhotoRecord> + initialize(title: String, description: String, started: Date): void - photoListAdapter : ArrayAdapter<PhotoRecord> + displayPhoto(): void This I believe is also deprecated as the problem information is already shown in the record list fragment + deletePhoto(): void

