LoginActivity UserLogin + appTitle: TextView + userId: EditText - UserID: String #onCreate(saveInstanceState: Bundle): void +signUpClick(View view) LoginPresenter +loginClick(View view) -changeActivity(class Activity): void - Userld: string MainThread + tryLogin(UserId):void + onLoginSuccess(User):void + onLoginFail(): void + post(Runnable): void + getInstance(): MainThread + getInstance(): ThreadExecutor SignupActivity + appTitle: TextView - run(): void CreateUserProfile SignUpPresenter + signUpTitle: TextView + cancel(): void <<Interface>> + isRunning(): boolean + userID: EditText - userName: String - userEmail: String - userPhoneNum: String - UserID: String
UserPhoneNum: String
- UserEmail: String UserRepo + userEmail: EditText + onFinished(): void + userPhoneNumber: EditText + execute(): void + patientUserBox: CheckBox - UserType: Boolean userType: Boolean + careProviderBox: CheckBox + CreateNewUser(userName: String, userEmail: String, userPhoneNum: int, userType: Boolean): void + onCUPSuccess():void + onCUPFail(): void # onCreate(savedInstanceState: Bundle): void + loginClick(): void - changeActivity(activity: Class): void AddAssignPatient -UserID: String PatientListPresenter - patients: ArrayList<Patient> - UserId: String ViewPatient CareProviderPatientList + onRemove(Patient: patient) + onAdd(UserId: String) - recyclerView: recyclerView + getAssigned() + onAAPSuccess(ArrayList<Patient>): void - patients: ArrayList + onAAPFail(): void # onCreate(savedInstanceState: Bundle): void + onRAPSuccess(patient):void + onPatientClick(view: View): void + onRAPFail(): void + deleteClick(view: View): void + onGAPSuccess(ArrayList<Patient>): void + onGAPFail():void + onCreateOptionsMenu(menu Menu): void RemoveAssginedPatient + searchClick(view: View) : void + addPatientClick(view View): void - userID: String - changeActivity(class Activity): void PatientRecylerAdapter LEGEND - list: ArrayList<Patients> -Blue: Presenters - deleteBtn: Button ViewCareProvider - presenter: PatientListPresenter -Purple: Views/Activities /- userID: String + onCreateViewHolder(ViewGroup parent, int viewType): ProblemListAdapter.ViewHolder -Orange: List Adapters + onBindViewHolder(ViewHolder, int position): void + getItemCount(): int - Green: Interactors -White: Model EditInfoActivity + appTitle: TextView + signUpTitle: TextView + userName: TextView EditUserController + userEmail: EditText + userPhoneNumber: EditText - userEmail: String EditUserProfile - userPhoneNumber: String - userID: String - userld: String # onCreate(savedInstanceState: Bundle): void - userEmail: String + backClick(): void - userPhoneNum: String - changeActivity(activity: Class): void + onUpdate(userid: String, userEmail: String, userPhoneNumber):void + onEUPSuccess(userEmail: String, userPhoneNumber: String):void + onEUPFail(): void UserProfileContactView + Userld: TextView + UserEmail: TextView + UserPhoneNum: TextView UserProfilerPresenter # onCreate(savedInstanceState: Bundle): void + backClick(): void - Userld: String - changeActivity(activity: Class): void GetLogginUser - userEmail: String - userPhoneNumber: String - UserID: String + getUserInfo(UserId: String): + onGUPSuccess(userId: String):void New_editProblemView + titleText: TextView + titleDate: TextView + titleDescription: TextView + problemTitle: EditText + problemDate: EditText CreateProblem add_editProblemPresenter + problemDescription: EditText + cancelBtn: Button - problemID: String problemTitle: String - problemTitle: String + saveBtn: Button problemDate: Date - problemDate: Date - problemDesc: string - problemDescription: String # onCreate(savedInstanceState: Bundle): void - problemId: String onSaveClick(): void - onCancelClick(): void - changeActivity(class Activity): void + onCreateProblem(problemTitle: String, problemDesc: String, problemDate Date,): void + onEditProblem(problemId: String, problemTitle: String, problemDate: Date, problemDesc: String): void EditProblem problemID: String V- problemTitle: String - problemDate: Date - problemDescription: String <<Interface>> ProblemRepo ViewProblem ProblemList - problems: ArrayList recyclerView: recyclerView - problemListAdapter: ProblemListAdapter ProblemListPresenter - addNewProblemBtn: Button - userId: String # onCreate(savedInstanceState: Bundle): void - problemId: integer + onProblemClick(view: View): void + editProblem(view: View): void + getProblems(userId: String): collection<Problem DeleteProblem + deleteProblem(view: View): void + deleteProblem(problmeld: integer): void + addProblem(view: View): void + onCreateOptionsMenu(menu Menu): void + searchClick(view View): void + deleteClick(view: View): void - changeActivity(class Activity): void **ProblemListAdapter** - list: ArrayList<Problems> - deleteBtn: Button editbtn: Button + onCreateViewHolder(ViewGroup parent, int viewType): ProblemListAdapter.ViewHolder + onBindViewHolder(ViewHolder, int position): void + getItemCount(): int RecordListPresenter **DBController** record: Record problemId: integer RecordListView - DBController: DBController - recordId: integer ViewRecord - JestDroidClient: Client - recyclerView: RecyclerView + getRecords(problemId: integer): collection<Records> - recordListAdater: RecordListAdapter records: ArrayList + deleteRecord(recordId: integer): void DBController(): constructor - addRecordBtn: Button - editRecordBtn: Button + getInstance(): DBController - deleteRecordBtn: Button - searchBtn: ImageButton RecordListAdapter # onCreate(savedInstanceState: Bundle): void list: ArrayList<record> + getRecords(): void - deleteBtn: Button + deletedelete(view View): void - editBtn: Button + editclick(view: View): void + recordClick(view: View): void - changeActivity(class Activity): void + onCreateViewHolder(ViewGroup parent, int viewType): RecordListAdapter.ViewHolder + onBindViewHolder(ViewHolder holder, int position): void RepoFactory + getItemCount(): int + method(type): type RecordViewPresenter - recordId: integer AbstractRepo - problemId: integer recordTile: String #client: JestDroidClient - userName: String #elasticIndex: String #db: SQLiteDataBase <<Interface>> - bodyLocation: bodyLocation RecordsRepo RecordView - bodyPhoto: file #online: Boolean - date: Date + recordTitle: TextView - recordComment: String + insert(): void + recordDate: TextView - recordMapCoordinates: ArrayList<LatLng> coordinates + update(): void CreateRecord + bodyLocationImg: ImageView + delete(): void + bodyPhotoList: ListView + editRecord(problemId: integer, recordId: integer, - recordID: String + commentTitle: TextView recordTitle: String, bodyLocation: bodyLocation, - recordTitle: String + mapImage: ImageButton + allPhotosGrid: GridViewAdapter bodyPhoto: file, date Date, recordComment String recordComment: String xCord: integer, yCord: integer): Void - recordDate: Date + editBtn: ImageButton + deleteRecord(): void + addRecord(problemId: integer, recordId: integer # onCreate(savedInstanceState: Bundle): void recordTitle: String, bodyLocation: bodyLocation, + mapClick(View view): void bodyPhoto: file, date Date, recordComment String + bodyLocationClick(View view): void xCord: integer, yCord: integer): Void + bodyPhotoClick(View view): void + returnBodyPart(xCord: Float, yCord: Float): BodyPart + photoClick(View view): void + editBtnClick(View view): void + fillBodyPhoto(): void DeleteRecord + fillDate(): void + fillAllPhoto(): void - recordID: String GridViewAdapter + fillComment(): void - changeActivity(class Activity): void + context: Context + getCount(): int + getItem(position: int): Object + getItem(position: int): Object EditRecord + getItemId(position: int): long - recordID: String - recordTitle: String /- recordComment: String - recordPhotos: Arraylist - recordGeolocation: Geolocation - recordAuthor: String TakingPhoto + exitCameraBtn: Button **TakePhotoPresenter** + backArrow: ImageView
REQUEST_TAKE_PHOTO: int recordId: integer imagePath: String - Photo: photo - photo: Flle imageFile: File + addPhotoFile(photo: File, recordId: integer) # onCreate(savedInstanceState: Bundle): void - toCamara(): void # onActivityResult(requestCode: int, resultCode: int, data Intent): void - setPhoto(): void Camara - createImageFile(): File - addGrideLines(): void - camara: Camara - savePhotoClick(): void - changeActivity(class Activity): void - mHolder: SurfaceHolder <<Interface>> PhotoRepo + surfaceCreated(holder: SurfaceHolder) void + surfaceDestroyed(holder: SurfaceHolder) void + surfaceChanged(holder: SurfaceHolder, format: int, w: int, h: int) void **PhotoView PhotoViewPresenter** - Photos: ArrayList<photo> + editBtn: Button + deleteBtn: Button - recordId: integer DeleteRecordPhoto + rightArrowBtn: ImageButton + leftArrowBtn: ImageButton + getPhotos(recordId: integer) + image: View - setPhoto(): void + getPhotos(): void - changeActivity(class Activity): \ ViewBodyLocation **PatientGeoLocationView** MapPresenter - bodyLocations: ArrayList + GeoTitle: TextView - xCord: Double + GeoMap: TextView - yCord: Double + saveLocationBtn: Button - recordId: integer + locationDetailsTitle: TextView + locationDetails: TextView + locationPin: ImageView + getLocationsCoordinates(recordId: integer): void + backArrow: ImageView + addGeoLocation(recordId: integer, xCord Double, yCord Double):void + GoogleMapAdapter: AutoGoogleMapAdapter + myGoogleApi: GoogleApiClient + getLocations(): Void + saveLocationClick(): Void + onMapClick(): Void + onPinClick(): Void + onSaveClick(): Void AddBodyLocation BodyLocationPhoto <<Interface>> - bodyLocationID: String + bodyLocationTitle: TextView BodyLocationRepo - bodyLocationUser: String + saveBtn: Button BodyLocationPresenter bodyLocationLable: String + reTakeBtn: Button - bodyLocationFrontImageID: String + image: ImageView - xCord: Double - bodyLocationBackImageID: String + imageFile: File - yCord: Double - xCord: Double - image: File - yCord: Double - recordId: integer getCoordinates(): ArrayList<Float> + getPhotos(): void + setPhotos(): void + addBodyLoactionPhoto(recordId: integer, xCord: Double, yCord: Double, image: File) - changeActivity(class Activity): void DeleteBodyLocation AutoGoogleMapAdapter - bodyLocationID: String @Googlemap API Search SearchPresenter + recyclerView: RecyclerView - record: Record + searchListAdapter: SearchListAdapter - problem: Problem + onCreateOptionsMenu(menu Menu): void + search(keywords: ArrayList<string>, bodyPart: BodyPart, geoLocation: GeoLocation) changeActivity(class Activity): void RecordListAdapter - problems: ArrayList<Problem> - records: ArrayList<Record> + onCreateViewHolder(ViewGroup parent, int viewType): SearchListAdapter.ViewHolder + onBindViewHolder(ViewHolder holder, int position): void + getItemCount(): int **PatientList** CareProvider - Patient :<Patients> + mPatients: ArrayList<Patier + method(): Type + method(): Type User Can only add comment on problems and records - mUserID: String - mPhoneNumber: String Login **BodyLocation** - mEmailAddress: String Problem Record Photo - mProblemID: Integer + method(): Type Patient + mRecordID: Integer bodyLabel: String - mPhotoID: Integer - FrontImageID: String - mTitle: String mTitle: String - mPhotoFile: File + mProblems: ArrayList<Proble - BackImageID: String - mStartDate: Date - mComment: String - mDescription: String - mLocation: Geolocation + method(): Type + method(): Type + method(): Type - mRecords: ArrayList<Record> - mBodyLocations: Set<BodyLocations> - mPhotos: Set<Photo> + method(): Type + contains(bodyPart BodyPart): boolean GeoLocation + contains(ArrayList<String> keywords): bo mLatitude: Double - mLongitude: Double + compareTo(geoLocation GeoLocation): int <<Interface>> Searchable + containsKeywords(ArrayList<String> keywords): boolean + containsBodyPart(ArrayList<BodyPart bodyParts): boolean

+ <u>searchKeywords(</u>ArrayList<Searchable>, ArrayList<String> keywords): ArrayList<Searchable>

+ <u>searchBodyPart(</u>ArrayList<Searchable>, BodyPart bodyPart): ArrayList<Searchable> + searchGeoloaction(ArrayList<Seachable>, GeoLocation): ArrayList<Searchable>

ThreadExecutor