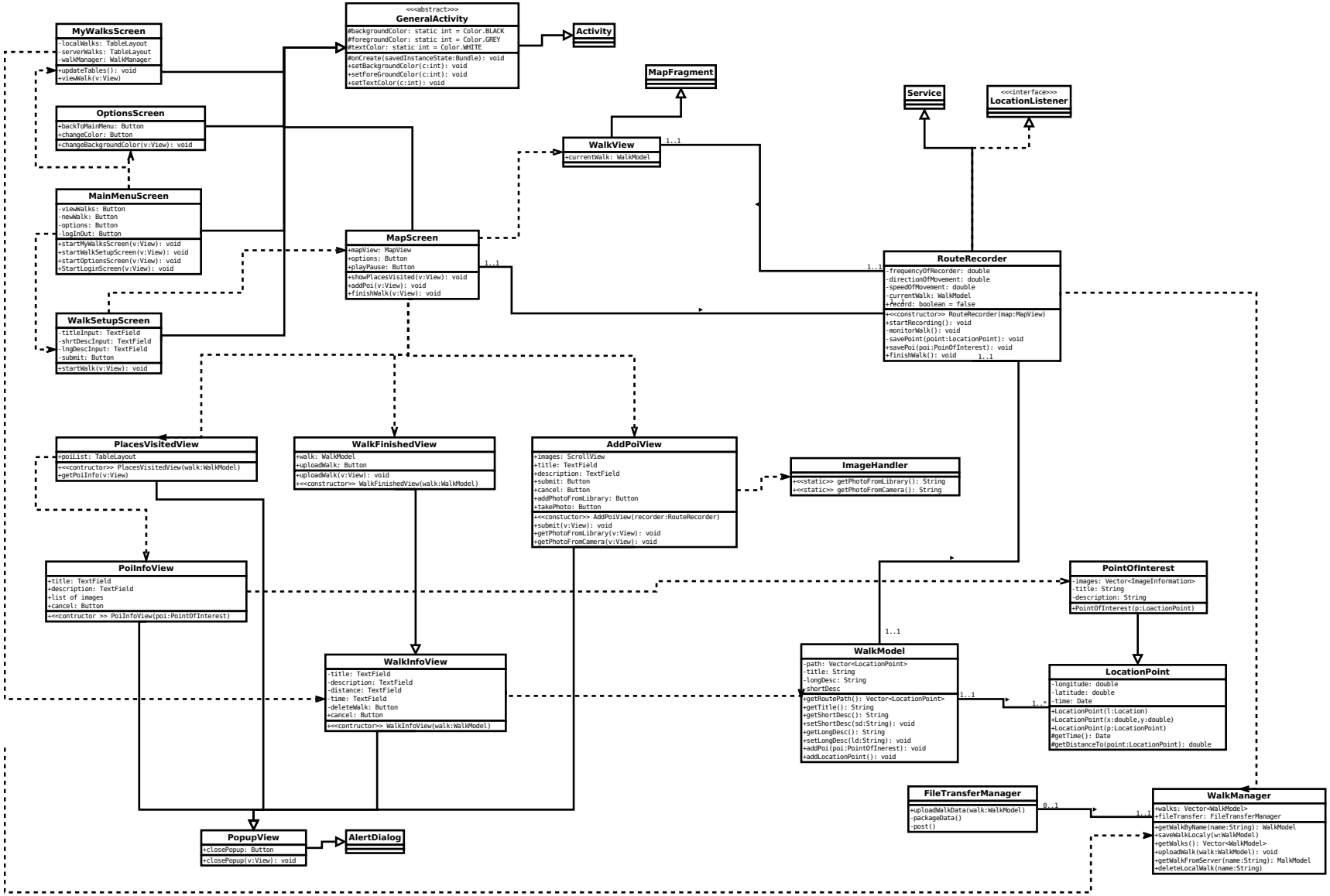
MyWalksScreen
-localWalks: TableLayout
-serverWalks: TableLayout
-walkManager: WalkManager
+updateTables(): void
+viewWalk(v:View)
<<<abstract>>>
GeneralActivity
#backgroundColor: static int = Color.BLACK
#foregroundColor: static int = Color.GREY
#textColor: static int = Color.WHITE
#onCreate(savedInstanceState:Bundle): void
+setBackgroundColor(c:int): void
+setForeGroundColor(c:int): void
+setTextColor(c:int): void
Activity
MapFragment
Service
<<<interface>>>
LocationListener
OptionsScreen
+backToMainMenu: Button
+changeColor: Button
+changeBackgroundColor(v:View): void
WalkView
+currentWalk: WalkModel
1..1
MainMenuScreen
-viewWalks: Button
-newWalk: Button
-options: Button
-logInOut: Button
+startMyWalksScreen(v:View): void
+startWalkSetupScreen(v:View): void
+startOptionsScreen(v:View): void
+StartLoginScreen(v:View): void
MapScreen
+mapView: MapView
+options: Button
+playPause: Button
+showPlacesVisited(v:View): void
+addPoi(v:View): void
+finishWalk(v:View): void
1..1
RouteRecorder
-frequencyOfRecorder: double
-directionOfMovement: double
-speedOfMovement: double
-currentWalk: WalkModel
-record: boolean = false
+<<constructor>> RouteRecorder(map:MapView)
+startRecording(): void
-monitorWalk(): void
-savePoint(point:LocationPoint): void
+savePoi(poi:PoinOfInterest): void
+finishWalk(): void
1..1
WalkSetupScreen
-titleInput: TextField
-shrtDescInput: TextField
-lngDescInput: TextField
-submit: Button
+startWalk(v:View): void
PlacesVisitedView
+poiList: TableLayout
+<<contructor>> PlacesVisitedView(walk:WalkModel)
+getPoiInfo(v:View)
WalkFinishedView
+walk: WalkModel
+uploadWalk: Button
+uploadWalk(v:View): void
+<<constructor>> WalkFinishedView(walk:WalkModel)
AddPoiView
+images: ScrollView
+title: TextField
+description: TextField
+submit: Button
+cancel: Button
+addPhotoFromLibrary: Button
+takePhoto: Button
+<<constuctor>> AddPoiView(recorder:RouteRecorder)
+submit(v:View): void
+getPhotoFromLibrary(v:View): void
+getPhotoFromCamera(v:View): void
ImageHandler
+<<static>> getPhotoFromLibrary(): String
+<<static>> getPhotoFromCamera(): String
PoiInfoView
+title: TextField
+description: TextField
+list of images
+cancel: Button
+<<contructor >> PoiInfoView(poi:PointOfInterest)
PointOfInterest
-images: Vector<ImageInformation>
-title: String
-description: String
+PointOfInterest(p:LoactionPoint)
1..1
WalkModel
-path: Vector<LocationPoint>
-title: String
-longDesc: String
-shortDesc
+getRoutePath(): Vector<LocationPoint>
+getTitle(): String
+getShortDesc(): String
+setShortDesc(sd:String): void
+getLongDesc(): String
+setLongDesc(ld:String): void
+addPoi(poi:PointOfInerest): void
+addLocationPoint(): void
1..1
1..*
LocationPoint
-longitude: double
-latitude: double
-time: Date
+LocationPoint(l:Location)
+LocationPoint(x:double,y:double)
+LocationPoint(p:LocationPoint)
#getTime(): Date
#getDistanceTo(point:LocationPoint): double
WalkInfoView
-title: TextField
-description: TextField
-distance: TextField
-time: TextField
-deleteWalk: Button
+cancel: Button
+<<contructor>> WalkInfoView(walk:WalkModel)
FileTransferManager
+uploadWalkData(walk:WalkModel)
-packageData()
-post()
0..1
1..1
WalkManager
+walks: Vector<WalkModel>
+fileTransfer: FileTransferManager
+getWalkByName(name:String): WalkModel
+saveWalkLocaly(w:WalkModel)
+getWalks(): Vector<WalkModel>
+uploadWalk(walk:WalkModel): void
+getWalkFromServer(name:String): MalkModel
+deleteLocalWalk(name:String)
PopupView
+closePopup: Button
+closePopup(v:View): void
AlertDialog