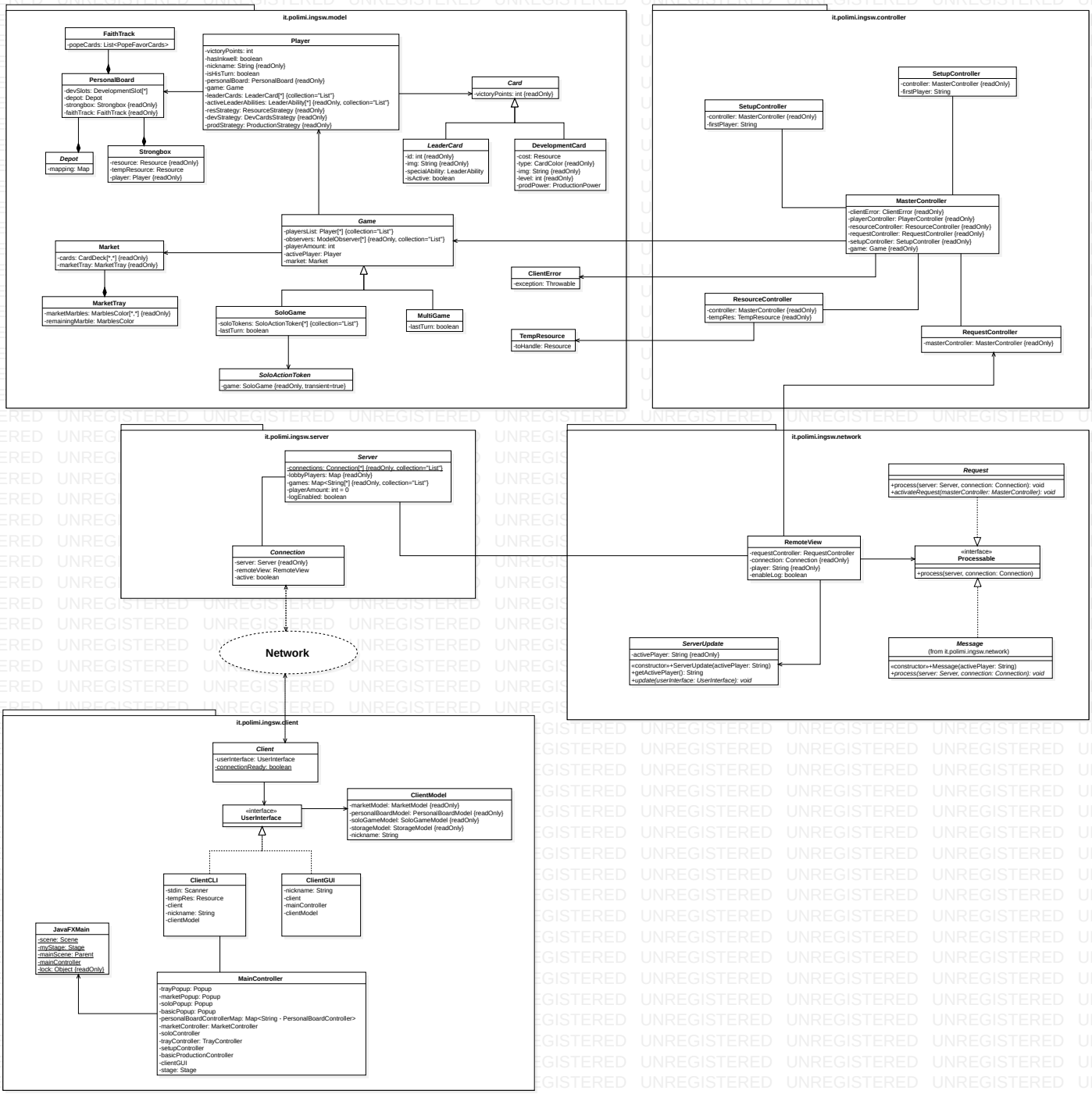
Model.Main


it.polimi.ingsw.model
FaithTrack
-popeCards: List<PopeFavorCards>
PersonalBoard
-devSlots: DevelopmentSlot[*]
-depot: Depot
-strongbox: Strongbox {readOnly}
-faithTrack: FaithTrack {readOnly}
Depot
-mapping: Map
Strongbox
-resource: Resource {readOnly}
-tempResource: Resource
-player: Player {readOnly}
Player
-victoryPoints: int
-hasInkwell: boolean
-nickname: String {readOnly}
-isHisTurn: boolean
-personalBoard: PersonalBoard {readOnly}
-game: Game
-leaderCards: LeaderCard[*] {collection="List"}
-activeLeaderAbilities: LeaderAbility[*] {readOnly, collection="List"}
-resStrategy: ResourceStrategy {readOnly}
-devStrategy: DevCardsStrategy {readOnly}
-prodStrategy: ProductionStrategy {readOnly}
Card
-victoryPoints: int {readOnly}
LeaderCard
-id: int {readOnly}
-img: String {readOnly}
-specialAbility: LeaderAbility
-isActive: boolean
DevelopmentCard
-cost: Resource
-type: CardColor {readOnly}
-img: String {readOnly}
-level: int {readOnly}
-prodPower: ProductionPower
Game
-playersList: Player[*] {collection="List"}
-observers: ModelObserver[*] {readOnly, collection="List"}
-playerAmount: int
-activePlayer: Player
-market: Market
Market
-cards: CardDeck[*,*] {readOnly}
-marketTray: MarketTray {readOnly}
MarketTray
-marketMarbles: MarblesColor[*,*] {readOnly}
-remainingMarble: MarblesColor
SoloGame
-soloTokens: SoloActionToken[*] {collection="List"}
-lastTurn: boolean
MultiGame
-lastTurn: boolean
SoloActionToken
-game: SoloGame {readOnly, transient=true}
ClientError
-exception: Throwable
TempResource
-toHandle: Resource
it.polimi.ingsw.controller
SetupController
-controller: MasterController {readOnly}
-firstPlayer: String
SetupController
-controller: MasterController {readOnly}
-firstPlayer: String
MasterController
-clientError: ClientError {readOnly}
-playerController: PlayerController {readOnly}
-resourceController: ResourceController {readOnly}
-requestController: RequestController {readOnly}
-setupController: SetupController {readOnly}
-game: Game {readOnly}
ResourceController
-controller: MasterController {readOnly}
-tempRes: TempResource {readOnly}
RequestController
-masterController: MasterController {readOnly}
it.polimi.ingsw.server
Server
-connections: Connection[*] {readOnly, collection="List"}
-lobbyPlayers: Map {readOnly}
-games: Map<String[*]> {readOnly, collection="List"}
-playerAmount: int = 0
-logEnabled: boolean
Connection
-server: Server {readOnly}
-remoteView: RemoteView
-active: boolean
Network
it.polimi.ingsw.network
Request
+process(server: Server, connection: Connection): void
+activateRequest(masterController: MasterController): void
RemoteView
-requestController: RequestController
-connection: Connection {readOnly}
-player: String {readOnly}
-enableLog: boolean
«interface»
Processable
+process(server, connection: Connection)
ServerUpdate
-activePlayer: String {readOnly}
«constructor»+ServerUpdate(activePlayer: String)
+getActivePlayer(): String
+update(userInterface: UserInterface): void
Message
(from it.polimi.ingsw.network)
«constructor»+Message(activePlayer: String)
+process(server: Server, connection: Connection): void
it.polimi.ingsw.client
Client
-userInterface: UserInterface
-connectionReady: boolean
«interface»
UserInterface
ClientModel
-marketModel: MarketModel {readOnly}
-personalBoardModel: PersonalBoardModel {readOnly}
-soloGameModel: SoloGameModel {readOnly}
-storageModel: StorageModel {readOnly}
-nickname: String
ClientCLI
-stdin: Scanner
-tempRes: Resource
-client
-nickname: String
-clientModel
ClientGUI
-nickname: String
-client
-mainController
-clientModel
JavaFXMain
-scene: Scene
-myStage: Stage
-mainScene: Parent
-mainController
-lock: Object {readOnly}
MainController
-trayPopup: Popup
-marketPopup: Popup
-soloPopup: Popup
-basicPopup: Popup
-personalBoardControllerMap: Map<String - PersonalBoardController>
-marketController: MarketController
-soloController
-trayController: TrayController
-setupController
-basicProductionController
-clientGUI
-stage: Stage