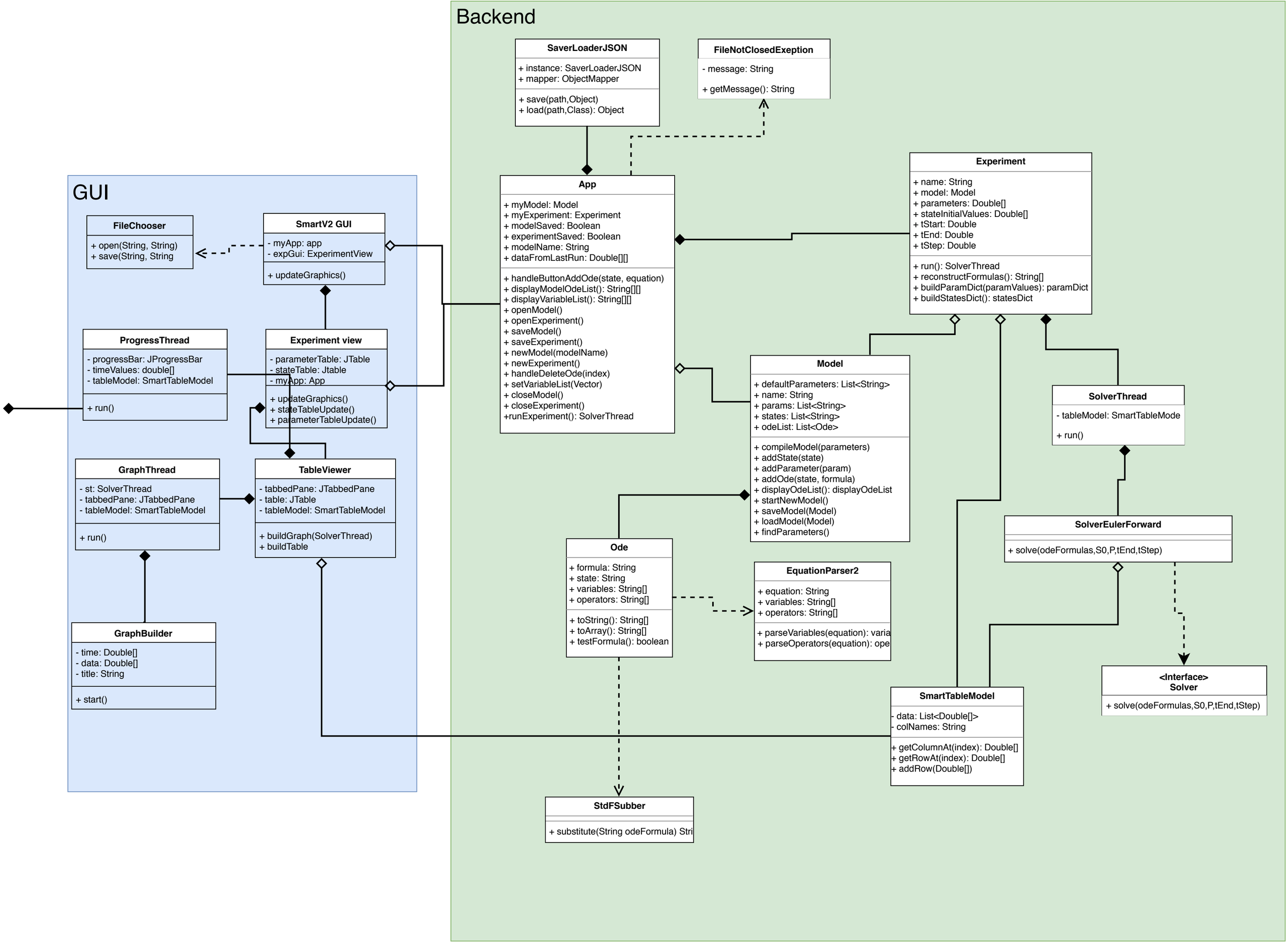
Backend
SaverLoaderJSON
+ instance: SaverLoaderJSON
+ mapper: ObjectMapper
+ save(path,Object)
+ load(path,Class): Object
FileNotClosedExeption
- message: String
+ getMessage(): String
GUI
FileChooser
+ open(String, String)
+ save(String, String
SmartV2 GUI
- myApp: app
- expGui: ExperimentView
+ updateGraphics()
App
+ myModel: Model
+ myExperiment: Experiment
+ modelSaved: Boolean
+ experimentSaved: Boolean
+ modelName: String
+ dataFromLastRun: Double[][]
+ handleButtonAddOde(state, equation)
+ displayModelOdeList(): String[][]
+ displayVariableList(): String[][]
+ openModel()
+ openExperiment()
+ saveModel()
+ saveExperiment()
+ newModel(modelName)
+ newExperiment()
+ handleDeleteOde(index)
+ setVariableList(Vector)
+ closeModel()
+ closeExperiment()
+runExperiment(): SolverThread
Experiment
+ name: String
+ model: Model
+ parameters: Double[]
+ stateInitialValues: Double[]
+ tStart: Double
+ tEnd: Double
+ tStep: Double
+ run(): SolverThread
+ reconstructFormulas(): String[]
+ buildParamDict(paramValues): paramDict
+ buildStatesDict(): statesDict
ProgressThread
- progressBar: JProgressBar
- timeValues: double[]
- tableModel: SmartTableModel
+ run()
Experiment view
- parameterTable: JTable
- stateTable: Jtable
- myApp: App
+ updateGraphics()
+ stateTableUpdate()
+ parameterTableUpdate()
Model
+ defaultParameters: List<String>
+ name: String
+ params: List<String>
+ states: List<String>
+ odeList: List<Ode>
+ compileModel(parameters)
+ addState(state)
+ addParameter(param)
+ addOde(state, formula)
+ displayOdeList(): displayOdeList
+ startNewModel()
+ saveModel(Model)
+ loadModel(Model)
+ findParameters()
SolverThread
- tableModel: SmartTableMode
+ run()
GraphThread
- st: SolverThread
- tabbedPane: JTabbedPane
- tableModel: SmartTableModel
+ run()
TableViewer
- tabbedPane: JTabbedPane
- table: JTable
- tableModel: SmartTableModel
+ buildGraph(SolverThread)
+ buildTable
SolverEulerForward
+ solve(odeFormulas,S0,P,tEnd,tStep)
Ode
+ formula: String
+ state: String
+ variables: String[]
+ operators: String[]
+ toString(): String[]
+ toArray(): String[]
+ testFormula(): boolean
EquationParser2
+ equation: String
+ variables: String[]
+ operators: String[]
+ parseVariables(equation): varia
+ parseOperators(equation): ope
GraphBuilder
- time: Double[]
- data: Double[]
- title: String
+ start()
<Interface>
Solver
+ solve(odeFormulas,S0,P,tEnd,tStep)
SmartTableModel
- data: List<Double[]>
- colNames: String
+ getColumnAt(index): Double[]
+ getRowAt(index): Double[]
+ addRow(Double[])
StdFSubber
+ substitute(String odeFormula) Stri