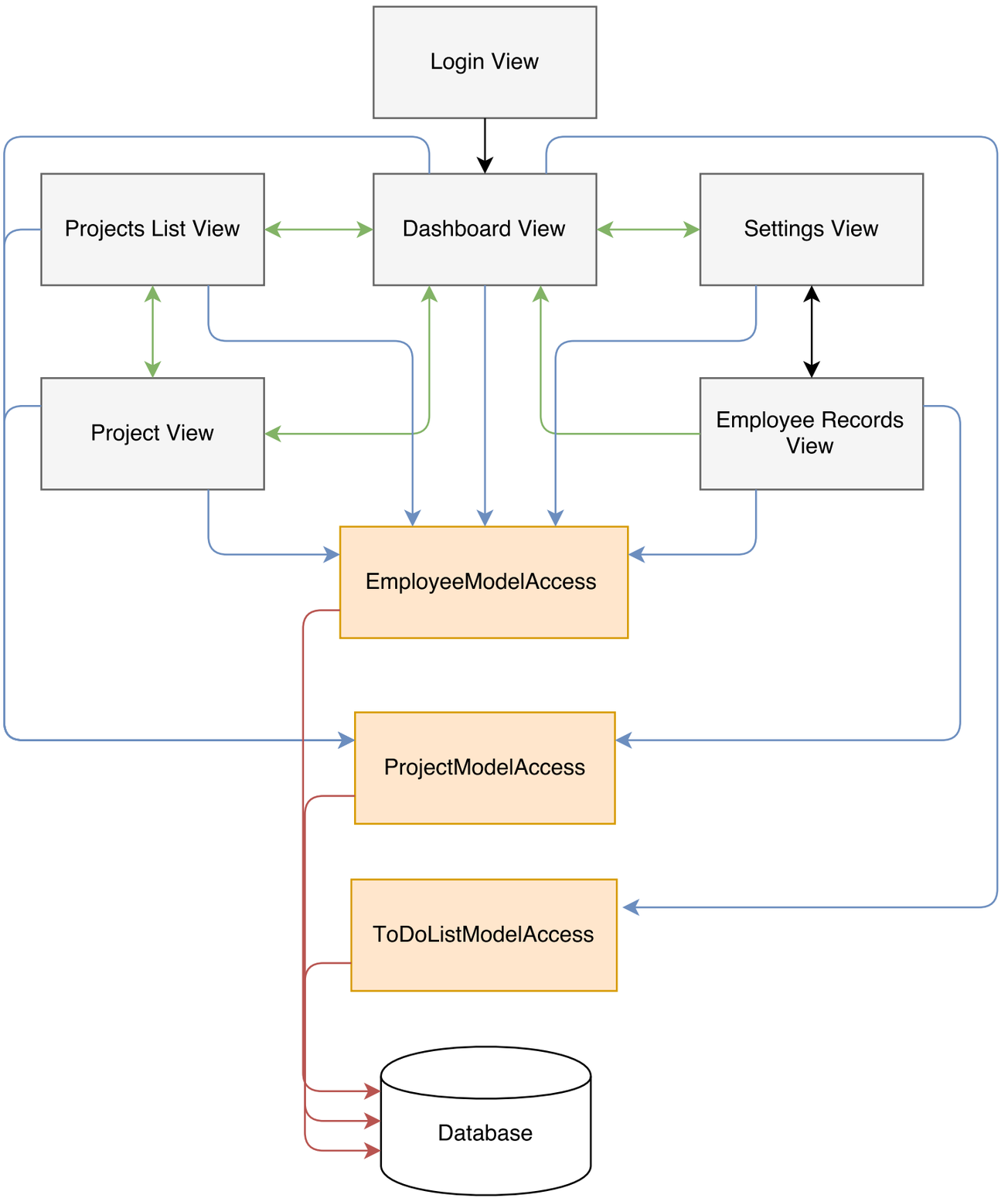
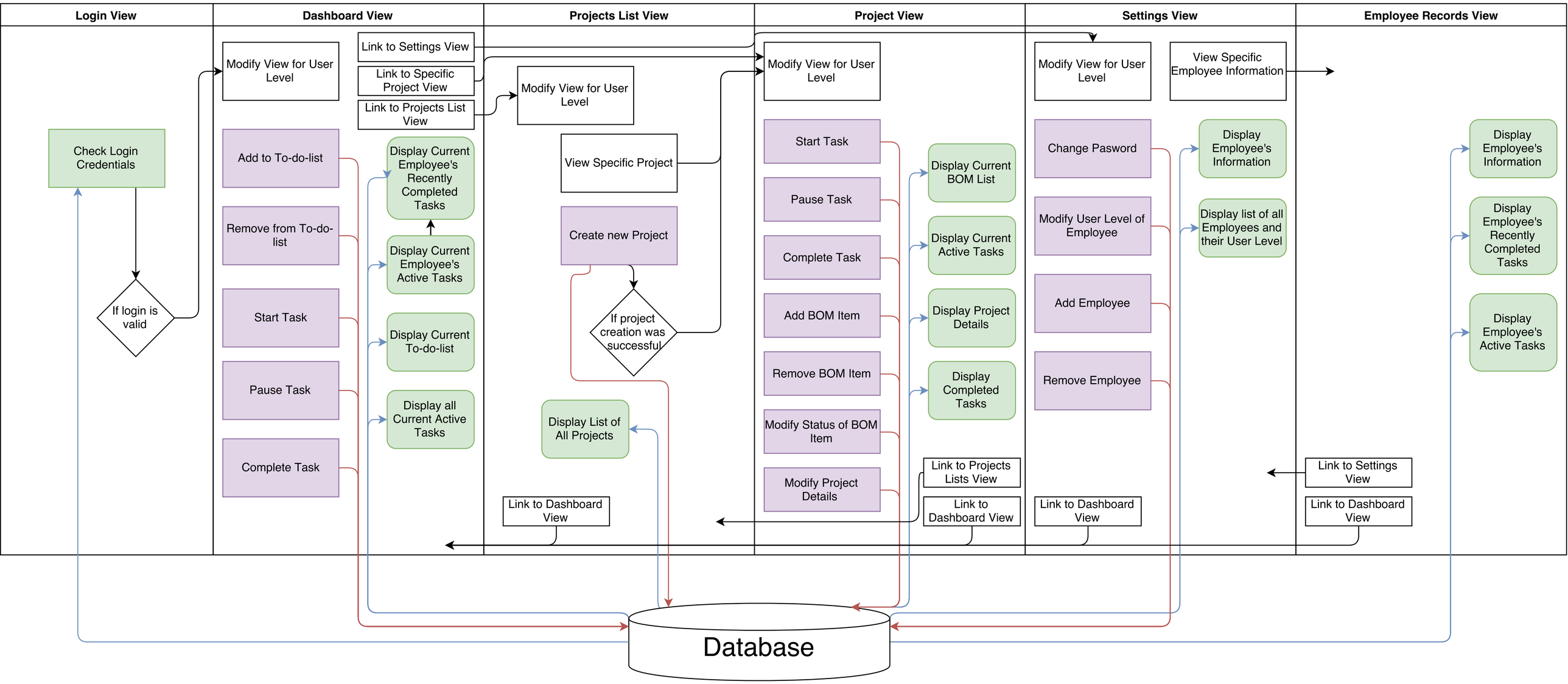
Highest Level Overview of Project Views and Data Access

This diagram shows a high level overview of the software views and how control flows between views. It also shows how the views access data through model access classes. Green lines represent transitions in views, blue lines show access to model access classes, and red lines show access to the database

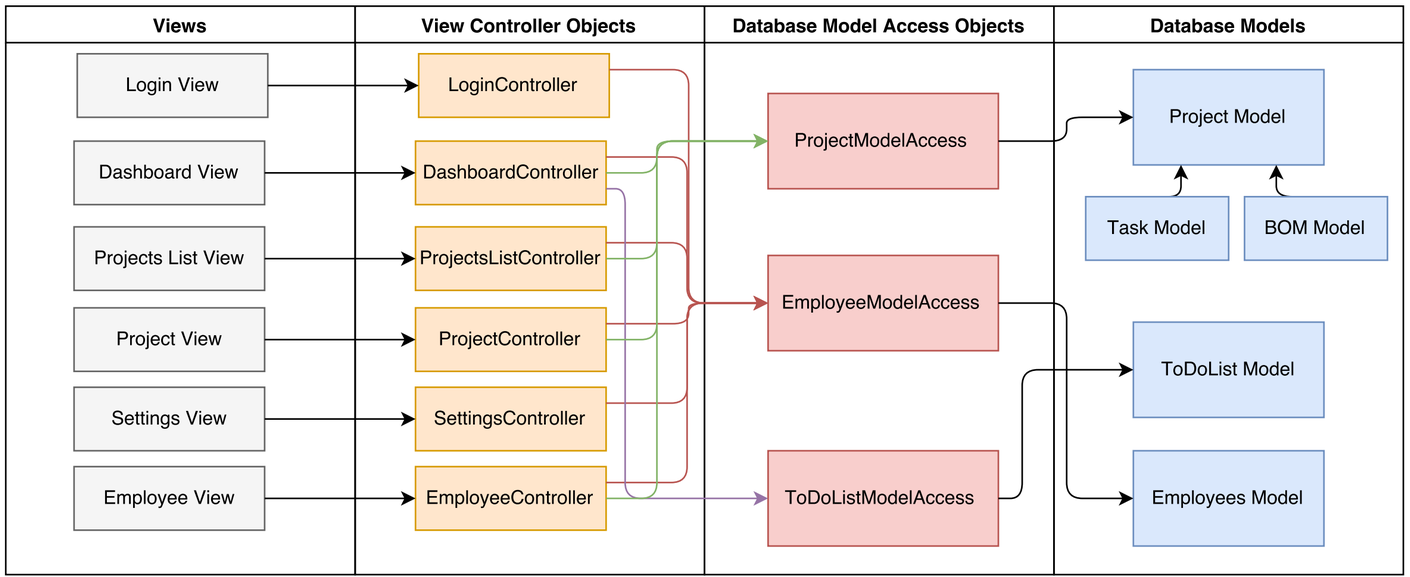


Views and Features Diagram

The diagram below shows the different views that the software will have. It also show what functionality each view will contain.
Purple boxes represent functionality that the user my access. Green boxes represent methods which will display data on the view.
White boxes represent methods for transferring between different views, and diamonds represent automatic transitions between views after successful operations.
Finally, red lines represent writing to the database, while blue lines represent reading from the database.



The diagram below shows how data will be flow between the views and the database.
The view controllers will have database model access objects which will allow CRUD operations on the data.
Data from the database can only be accessed through Model Access Objects.



UML Class Diagrams

These diagrams show all of the object types and class types that will be necessary for the functions of Right-on-Time. Classes which have red highlights are data/model access classes, in other words they act directly on the data in the database. Classes with a grey highlights are view controllers, which modify the user interface and accept user input. Classes with an orange highlight are data objects which carry data between views and the model access classes. There are also 3 enumerated types designated by green highlights. Instead of writing getters and setters for each object, I denoted them by writing "+ Accessor Methods."

ProjectModelAccess

+ ProjectModelAccess()
+ getProjectDetails(projectID : String) : ProjectInfoData
+ getActiveTasksByProjectID(projectID : String) : ArrayList<TaskData>
+ getCompletedTasksByProjectID(projectID : String) : ArrayList<TaskData>
+ getActiveTasksByEmployeeID(employeeID : String) : ArrayList<TaskData>
+ getCompletedTasksByEmployeeID(employeeID : String) : ArrayList<TaskData>
+ getBOMList(projectID : String) : ArrayList<BillOfMaterialData>
+ getProjects() : ArrayList<ProjectInfoData>
+ addBOMItem(BOMObject : BillOfMaterialData)
+ addTaskItem(taskObject : TaskData)
+ addProject(projectObject : ProjectInfoData)

EmployeeModelAccess

+ EmployeeModelAccess()
+ getEmployeeDataByID(employeeID : String) : EmployeeData
+ authenticateLogin(username : String, password : String) : EmployeeData
+ addEmployee(employeeData : EmployeeData) : EmployeeData
+ removeEmployeeByID(employeeID : String)

ToDoListModelAccess

+ ToDoListModelAccess()
+ getToDoList() : ArrayList<ToDoListData>
+ addToDoListItem(toDoListItem : ToDoListData)
+ removeToDoListItem(toDoListItem : ToDoListData)

LoginViewController

- usernameEntry : String
- passwordEntry : String

+ LoginViewController()
+ authenticate(String, String): Boolean

<<enumeration>> BOMStatus

Requested

Ordered

Delivered

<<enumeration>> TaskStatus

Started

Paused

Finished

<<enumeration>> UserLevel

Project Manager

Purchasing

Employee

Administrator

Shop Manager

DashboardViewController

- employeesData : EmployeeData
- toDoListDataObjects : ArrayList<ToDoListData>
- allActiveTaskObjects : ArrayList<TaskData>
- employeeActiveTaskObjects : ArrayList<TaskData>
- toDoListModelAccess : ToDoListModelAccess
- employeeModelAccess : EmployeeModelAccess
- projectModelAccess : ProjectModelAccess

+ DashboardViewController(employeesData : EmployeeData)
+ modifyView(userLevel : UserLevel)
+ displayAllActiveTasks()
+ displayEmployeeActiveTasks(employeeID : String)
+ displayRecentlyCompletedTasks(employeeID : String)
+ displayToDoList()
+ addToDoListItem(priority : String, description : String)
+ removeToDoListItem(itemID : String)
+ startTask(projectID : String, description : String)
+ pauseTask(taskID : String, pauseReason : String)
+ finishTask(taskID : String)

SettingsViewController

- employeesData : EmployeeData
- allEmployeeObjects : ArrayList<EmployeeData>
- currentEmployeeData : EmployeeData
- employeeModelAccess : EmployeeModelAccess

+ SettingsViewController(employeesData : EmployeeData)
+ modifyView(userLevel : UserLevel)
+ displayEmployeesInformation()
+ displayAllEmployees()
+ changePassword(newPass1 String, newPass2 String)
+ modiflyUserLevel(employeeID : String, newUserLevel : UserLevel)
+ addEmployee(name : String, userLevel : UserLevel, username : String)
+ removeEmployee(username : String)

ProjectViewController

- employeesData : EmployeeData
- projectDataObject : ProjectInfoData
- allActiveTaskObjects : ArrayList<TaskData>
- allCompletedTaskObjects : ArrayList<TaskData>
- allBillOfMaterialObjects : ArrayList<BillOfMaterialData>
- employeeModelAccess : EmployeeModelAccess
- projectModelAccess : ProjectModelAccess

+ ProjectViewController(employeesData : EmployeeData, projectID : String)
+ modifyView(userLevel : UserLevel)
+ displayAllActiveTasks()
+ displayCompletedTasks()
+ displayProjectDetails()
+ displayBOMList()
+ startTask(projectID : String, description : String)
+ pauseTask(taskID : String, pauseReason : String)
+ finishTask(taskID : String)
+ addBOMItem(itemDescription : String)
+ removeBOMItem(billOfMaterialID : String)
+ modifyBOMStatus(billOfMaterialID : String, newStatus : BOMStatus)
+ modifyProject(projectName : String, customer : String, projectManager : String, description : String, status : Boolean, budgetedHours : double)

ProjectsListViewController

- employeesData : EmployeeData
- allProjectObjects : ArrayList<ProjectInfoData>
- employeeModelAccess : EmployeeModelAccess
- projectModelAccess : ProjectModelAccess

+ ProjectsListViewController(employeesData : EmployeeData)
+ modifyView(userLevel : UserLevel)
+ displayAllProjects()
+ createNewProject(projectName : String, projectType : String, customer : String, projectManager : String, description : String, budgetedHours : Double)

EmployeeRecordsViewController

- employeesData : EmployeeData
- allTasksForEmployee : ArrayList<TaskData>
- employeeModelAccess : EmployeeModelAccess
- projectModelAccess : ProjectModelAccess

+EmployeeRecordsViewController(employeesData : EmployeeData)
+ displayEmployeesData()
+ displayRecentlyCompletedTasks()
+ displayActiveTasks()

ToDoListData

- ID : String
- dateAdded : String
- priority : String
- Description : String

+ ToDoListData(ID : String, dateAdded : String, priority : String, Description : String)
+ ToDoListData(dateAdded : String, priority : String, Description : String)
+ Accessor Methods

TaskData

- projectID : String
- employeeID : String
- taskID : String
- taskDescription : String
- taskStatus : String
- startTimes : ArrayList<String>
- endTimes : ArrayList<String>
- pauseReason: String
- totalHours : double

+ TaskData(projectID : String, employeeID : String, taskID : String, taskDescription : String, taskStatus : String, startTimes : ArrayList<String>, endTimes : ArrayList<String>, pauseReason: String, totalHours : double)
+ TaskData(projectID : String, employeeID : String, taskDescription : String, taskStatus : String, startTime : String)
+ Accessor Methods
+ calculateTotalHours()

EmployeeData

- name : String
- ID : String
- userLevel : UserLevel
- username : String
- password : String

+ EmployeeData(name : String, ID : String, userLevel : UserLevel, username : String, password : String)
+ EmployeeData(name : String, userLevel : UserLevel, username : String, password : String)
+ Accessor Methods

BillOfMaterialData

- billOfMaterialID : String
- projectID : String
- itemDescription : String
- itemPrice : Double
- status : BOMStatus
- employeeID : String

+ BillOfMaterialData(billOfMaterialID : String, projectID : String, itemDescription : String, itemPrice : Double, status : BOMStatus, employeeID : String)
+ BillOfMaterialData(projectID : String, itemDescription : String, itemPrice : Double, status : BOMStatus, employeeID : String)
+ Accessor Methods

ProjectInfoData

- projectName : String
- customer : String
- projectID : String
- projectType : String
- projectManager : String
- status : Boolean
- currentHours : Double
- budgetedHours : Double
- projectCompleted : Boolean
- projectDescription : String
- currentBudgetInvestment : Double

+ ProjectInfoData(projectName : String, customer : String, projectID : String, projectManager : String, status : Boolean, currentHours : Double, budgetedHours : Double, projectCompleted : Boolean, projectDescription : String, currentBudgetInvestment : Double)
+ ProjectInfoData(projectName : String, customer : String, projectType : String, projectManager : String, status : Boolean, currentHours : Double, budgetedHours : Double, projectCompleted : Boolean, projectDescription : String, currentBudgetInvestment : Double)
+ Accessor Methods