the sake of readability and conciseness: 1. Getters and setters methods are not shown 2. Utility and ViewUtility class are not shown as they serve 3. Some classes do not have any association lines drawn as many classes depend on many classes of another package, as such, package dependenices are drawn instead FIT3077 Assignment 3 Class Diagram **Team Terabytes**Name: Jun Qing Lim (30029937)
Name: Nicholas Wei Ming Yeo (29458021) How to read this UML:

1. <u>Classes/Packages</u> colored as <u>Red:</u> Changes that were made for assignment 3 Acyclic Dependencies Principle is applied in the whole system where there are no cycles between packages. Common Closure Principle is applied in the whole system where classes with similar functionalities are grouped into packages. Common Reuse Principle is applied in the whole system where similar classes are reused GSON is third party library from Google used to deserialize and serialize objects OkHttpClient is third party library used to make HTTP requests javax.swing is Java library to produce GUI components together. - loginModel: LoginModel - loginView: LoginView stores data with ▶ + performLogin(username: String, password: String): Boolean + getExpiringContracts(): List<Contract> + listenLogin(): Void + loginSuccess(user: User): Void DashboardModel

- contractsList: List<Contract>

+ refresh(): Void
+ getDashboardStatus(): DashboardStatus
+ getRenewingContracts(): List<Contract>
+ executeRenewalResponse(contract: Contract, toSign: Boolean): Void
+ getTopFiveContracts(): List<Contract> + getAll(): List<T>
+ get(id: String): T
+ add(object: T): T
+ patch(id: String, object: T): Boolean
+ remove(id: String): Boolean # getAllObjects(endpoint: String, clazz: Class<T[]>): List
getObject(endpoint: String, clazz: Class<T>): T
postObject(endpoint: String, clazz: Class<T>): T
postObject(endpoint: String, object: T): Boolean
deleteObject(endpoint: String): Boolean
patchObject(endpoint: String, object: T): Boolean stores data with ▶ mainPanel: JPanel
confirmSignButton: JButton
rejectButton: JButton
scrollPane: JScrollPane
frame: JFrame + listenViewActions(): Void
- handleRefresh(e: ActionEvent): Void
- handleInitiation(e: ActionEvent): Void
- checkContractRenewals(): Void - displayView(contract: Contract): Void + dispose(): Void - handleRenewContract(e: ActionEvent): Void
- initiateOpenBid(form: BidInitiation): Void
- initiateCloseBid(form: BidInitiation): Void # extractBid(userId: String, type: String): Bid
markBidClose(): Void
+ formContract(selection: Integer): Contract + listenViewActions(): Void
- handleRefresh(e: ActionEvent): Void
- handleSelectOffer(e: ActionEvent): Void
- handleOfferSelection(e: ActionEvent): Void + formContract(selection: Integer): Contract sees message with - expiredContracts: List<Contract>
- activeContracts: List<Contract>
- allContracts: List<Contract>
+ refresh(): Void
+ getTutorsList(selection: Integer): List<String>
+ getContractWithNewTerms(selection: Integer, newTerms: BidInfo, tutorUsername: String): Contract
+ getContractWithOldTerms(selection: Integer, tutorUsername: String): Contract ContractConfirmModel

contractConfirmModel: ContractConfirmModel
mainPanel: JPanel
- confirmSignButton: JButton
- contractDuration: JComboBox
- errorLabel: JLabel
- scrollPane: JScrollPane
- contractDurationLabel: JLabel
- frame: JFrame

- displayView(): Void
+ dispose(): Void - contractConfirmModel: ContractConfirmModel
- contractConfirmView: ContractConfirmView

+ listenViewActions(): Void
- handleSign(e: ActionEvent): Void # initViewTemplate(frameTitle: String, closeAction: Integer): Void
- makeMainPanel(): Void
- makeFrame(frameTitle: String, closeAction: Integer): Void
+ update(): Void
+ dispose(): Void
+ refreshContent(): Void
updateView(): Void
createButtons(): Void
refreshButtons(): Void + listenViewActions(): Void
- handleRefresh(e: ActionEvent): Void
- handleRespond(e: ActionEvent): Void
- handleBuyOut(e: ActionEvent): Void + listenViewActions(): Void
- handleRefresh(e: ActionEvent): Void
- handleRespondMessage(e: ActionEvent): Void

Important note about ommissions in class diagram for