FIT3077 Assignment 2 Class Diagram GSON is third party library from Google used to deserialize OkHttpClient is third party library used to make HTTP Name: Jun Qing Lim (30029937) and serialize objects Name: Nicholas Wei Ming Yeo (29458021) **Disclaimer**: All classes are contained with getters which are omitted in these diagrams for simplicity and ease javax.swing is Java library to produce GUI components LoginView - jPanel: JPanel- frame: JFrame- userField: JTextField LoginController <<abstract>> BasicApi<T> - passwordField: JPasswordField - loginModel: LoginModel - userLabel: JLabel loginView: LoginView <<interface>> ApiInterface<T> passwordLabel: JLabelpanelTitle: JLabelloginButton: JButton rootUrl: String - httpClient: OkHttpClient + listenLogin(): Void + loginSuccess(user: User): Void - gson: Gson + performLogin(username: String, password: String): Boolean + getAll(): List<T> - errorLabel: JLabel + get(id: String): T + add(object: T): T # getAllObjects(endpoint: String, clazz: Class<T[]>): List<1 - displayLogin(): Void # getObject(endpoint: String, clazz: Class<T>): T + patch(id: String, object: T): Boolean + dispose(): Void # postObject(endpoint: String, clazz: Class<T>): T + remove(id: String): Boolean <abstract>>
DashboardView postObject(endpoint: String, object: T): Boolean # deleteObject(endpoint: String): Boolean <<abstract>>
DashboardController DashboardModel # patchObject(endpoint: String, object: T): Boolean - dashboardModel- mainPanel: JPanel- contractPanel: JPanel ------- contractsList: List<Contract> - refreshButton: JButton dashboardModel: DashboardModel _------ initiateButton: JButton dashboardView: DashboardView + refresh(): Void + getDashboardStatus(): DashboardStatus - errorLabel: JLabel ______ # refreshContent(): Void + updateContracts(): Void QualificationApi MessageApi SubjectApi CompetencyApi + getAll(): List<Qualification> + getAll(): List<Message> + getAll(): List<Subject> getAll(): List<Competency> + get(id: String): Qualification + get(id: String): Message + get(id: String): Subject + get(id: String): Competency + listenViewActions(): Void + add(object: Message): Message + add(object: Subject): Subject + add(object: Competency): Competency + add(object: Qualification): Qualification + listenViewActions(): Void - handleRefresh(e: ActionEvent): Void + patch(id: String, object: Subject): Boolean + remove(id: String): Boolean + patch(id: String, object: Qualification): Boolean + patch(id: String, object: Message): Boolea + patch(id: String, object: Competency): Boolean - handleRefresh(e: ActionEvent): Void # bidId: String - handleInitiation(e: ActionEvent): Void + remove(id: String): Boolean + remove(id: String): Boolean + remove(id: String): Boolean - handleInitiation(e: ActionEvent): Void - initiateOpenBid(form: BidInitiation): Void + updateContracts()
- addButtons(): Void + updateContracts(): Void - initiateCloseBid(form: BidInitiation): Void # extractBid(userId: String, type: String): Bid - addButtons(): Void ____ # markBidClose(): Void - formContract(bidInfo: RidInfo): Cont EventListener + selectOffer(selection: Integer): BidInfo - openBidModel: OpenBidModel- mainPanel: JPanel- openBidPanel: JPanel ^ ContractApi + listenViewActions(): Voice + getAll(): List<User> + getAll(): List<Contract> - buttonPanel: JPanel + get(id: String): Bid + add(object: Bid): Bid - offerSelection: JComboBox + get(id: String): User + get(id: String): Contract - openBidModel: OpenBidModel - refreshButton: JButton + add(object: User): User + add(object: Contract): Contract - openBidView: OpenBidView - openBidOffers: List<BidInfo> - selectOfferButton: JButton + patch(id: String, object: User): Boolean + patch(id: String, object: Bid): Boolean + patch(id: String, object: Contract): Boolean _ _ _ - frame: JFrame - timeLeft: JLabel + remove(id: String): Boolean + verify(user: User): Boolean + remove(id: String): Boolean + close(id: String, bid: Bid): Boolean + remove(id: String): Boolean + sign(id: String, contract: Contract): Boolean + refresh(): Void + listenViewActions(): Void - handleRefresh(e: ActionEvent): Void CloseMessageView + selectOffer(selection: Integer): BidInfo + dispose(): Void - updateContent(): Void - handleSelectOffer(e: ActionEvent): Void - refreshContent(): Void - closeBidModel: CloseBidModel - refreshButtons(bidIndex: Integer): Void - updateView(bidInfoList: List<BidInfo>, bid: Bid): Void - closeMessagePanel: JPanel - updateButtons(count: Integer): Void - buttonPanel: JPanel BiddingController - offerSelection: JComboBox refreshButton: JButton respondMessageButton: JButton # initiateContract(contract: Contract): Void - selectOfferButton: JButton - messagePair: MessagePair - frame: JFrame _____ CloseBidController - closeBidModel: CloseBidModel <u>- userApi: UserApi</u> - closeBidOffers: List<MessageBidInfo> + dispose(): Void - closeBidModel: CloseBidModel - mainPanel: JPanel - subjectApi: SubjectApi - updateContent(): Void - closeBidMessages: List<MessagePair> - closeBidPanel: JPanel closeBidView: CloseBidView - contractApi: ContractApi - updateView(): Void + buildBid(userId: String, bp: BidPreference, type: String): Bid - buttonPanel: JPanel <u>- bidApi: BidApi</u> - updateButtons(): Void + update(): Void - - - + - - - - - - - - - - - - - -+ buildContract(bid: Bid, offer: BidInfo): Contract - offerSelection: JComboBox - messageApi: MessageApi - convertObject(m: Message): MessageBidInfo - refreshButton: JButton + listenViewActions(): Void ____ - viewMessageButton: JButton - handleRefresh(e: ActionEvent): Void + viewMessage(selection: int): MessagePair <u>+ getUserApi(): UserApi</u> - selectOfferButton: JButton - handleViewMessage(e: ActionEvent): Void ExpiryService + selectOffer(selection: Integer): BidInfo <u>+ getSubjectApi(): SubjectApi</u> - handleSelectOffer(e: ActionEvent): Void + sendMessage(messagePair: MessagePair, stringMsg: String): Void + getContractApi(): ContractApi - timeLeft: JLabel <u>+ getBidApi(): BidApi</u> <u>+ getMessageApi(): MessageApi</u> + dispose(): Void + checkIsExpired(bid: Bid): Boolean + getCompetencyApi(): CompetencyApi - updateContent(): Void <u>+ getQualificationApi(): QualificationApi</u> refreshContent(): Void ContractView - refreshButtons(bidIndex: Integer): Void - updateView(mbiList: List<MessageBidInfo>, bid: Bid): Void contractPanel: JPanel - updateButtons(count: Integer): Void ContractController stream package acts as a container (of Data class) to - confirmButton: JButton + update(): Void cancelButton: JButton - contractModel: ContractModel deserialize and serialize objects - contractId: String ScrollPane: JScrollPaneframe: JFrame - contractView: ContractView from API automatically. The attributes are obtained from the - createContract(contract: Contract): Void + dispose(): Void
- setDetails(): Void
- refreshButtons(): Void
+ update(): Void + listenViewActions(): Void + acceptContract(): Void - handleAccept(e: ActionEvent): Void + declineContract(): Void - handleReject(e: ActionEvent): Void BidPreference - openOffersModel: OpenOffersModel - mainPanel: JPanel CERTIFICATE DIPLOMA BACHELOR MASTERS PHD - qualification: QualificationTitle - openOffersPanel: JPanel - buttonPanel: JPanel - competency: int - subject: String - preferences: BidInfo - refreshButton: JButton - respondButton: JButton Message MessageAdditionalInfo OpenOffersModel - buyOutButton: JButton - type: String id: StringbidId: String - openOffersModel: OpenOffersModel BidAdditionalInfo - frame: JFrame - toStudent: Boolean - openOffersView: OpenOffersView - myOffer: BidInfo - receiverId: String - openOffers: List<BidInfo> - bidPreference: BidPreference - updateContent(): Void - posterId: String - time: String + listenViewActions(): Void - refreshContent(): Void - bidOffers: List<BidInfo> - dateCreated: Date duration: Integer
rate: Integer
numberOfSessions: Integer
freeLesson: Boolean + refresh(): Void + respond(bidInfo: BidInfo): Void - handleRefresh(e: ActionEvent): Void - refreshButtons(bidIndex: Integer): Void - dateClosedDown: Date - handleRespond(e: ActionEvent): Void - updateView(otherBidInfo: List<BidInfo>, BidInfo myBidInfo, - dateLastEdited: Date - initiatorId: String + buyOut(): Void - handleBuyOut(e: ActionEvent): Void OfferingView subject: Subject + OPEN_BID_MINS: Integer = 3 - day: String
- time: String
- duration: int - sendOffer(bidInfo: BidInfo): Void content: StringadditionalInfo: MessageAdditionalInfo - updateButtons(): Void subjectId: String ____ + CLOSED_BID_DAYS: Integer = 7 + update(): Void V OfferingController - additionalInfo: BidAdditionalInfo offeringPanel: JPanelbuttonPanel: JPanel - messages: List<Message> numberOfSessions: int - offeringModel: OfferingModel - bidSelection: JComboBox - freeLesson: boolean - offeringView: OfferingView <abstract>>
BasicModel - refreshButton: JButton - viewOffersButton: JButton listenViewActions(): Void — — - handleRefresh(e: ActionEvent): Void CloseOffersView + refresh(): Void # userId: String # oSubject: OSubject - handleViewOffers(e: ActionEvent): Void + viewOffers(selection: Integer) Void MessageBidInfo - closeOffesModel: CloseOffersModel - tutorMsgId: String # expiryService: ExpiryService CloseOffersController - updateButtons(count: Integer): Void - content: String - familyName: String
- userName: String - closeOffersPanel: JPanel - buttonPanel: JPanel - studentMsgId: String
- studentMsg: MessageBidInfo - updateView(bidList: List<Bid>): Void + attach(o: Observer): Void - description: String + update(): Void - secondParty: User - refreshButton: JButton - verified: Boolean - closeOffersModel: CloseOffersModel - secondPartyId: String - respondMessageButton: JButton - owner: User - closeOffersView: CloseOffersView isTutor: Boolean - competencies: List<Competency> - qualifications: List<Qualification>- initiatedBids: List<Bid> - subjectId: String - messagePair: MessagePair + dispose(): Void + listenViewActions(): Void - refreshContent(): Void - handleRefresh(e: ActionEvent): Void - numberOfSessions: int - refreshButtons(): Void - handleRespondMessage(e: ActionEvent): Void + refresh(): Void - freeLesson: boolean - updateView(): Void + sendMessage(mbi: MessageBidInfo): Void - expiryDate: Date - updateButtons(): Void - parseMsgBidInfo(mbi: MessageBidInfo): MessageAdditionalInfo + update(): Void subject: Subjectlevel: int lessonInfo: LessonadditionalInfo: EmptyClass - description: String
- competencies: List<Competency>
- bids: List<Bid> <<interface>> Observer to the Subject of # observers: List<Observer> confusion with the + attach(o: Observer): Void Subject in API + notifyObservers(): Void package dependencies are show to illustrate the dependency to produce the views