

DashboardPageController

-borderPane: BorderPane -nameSurnameLabel: Label -emailLabel: Label -ordersExpiringLabel: Label -editAccountButton: MFXButton -ordersHandligButton: MFXButton -proPic: ImageView -badgeExpiringLabelx: Label -warningImage: ImageView -NOTIFICATION_BADGE_SHRINKED_SIZES: int = 30 -MAX_NUM_ORDER_NOTIFIED: int = 99

#initialize(contro: DashboardControl): void ~ordersHandlingButtonAction(event ActionEvent): void ~monthlyReportsButtonAction(event ActionEvent): void ~editButtonAction(event ActionEvent): void ~logoutButtonAction(event ActionEvent): void -displayLoggedOperatorData(): void -displayExpiringOrders(): void

«bind»

+DashboardControl

OrdersPageController

-ordersTable: TableView<Order> -clientColumn: TableColumn<Order, String> -emissionDateColumn: TableColumn<Order, LocalDate> -extraWarrantyColumn: TableColumn<Order, Integer> -isExpressColumn: TableColumn<Order, Boolean> -orderNumberColumn: TableColumn<Order, Integer> -productColumn: TableColumn<Order, String> -quantityColumn: TableColumn<Order, Integer> -supplierColumn: TableColumn<Order, String> -startDatePicker: DatePicker -endDatePicker: DatePicker -filterButton: Button -clientTextField: TextField -borderPane: BorderPane -loadingPane: StackPane

#initialize(control: OrdersHandlingControl): void -handleKeyPressed(event: KeyEvent): void -displayUnfilteredOrders(): ObservableList<Order> +filterButtonAction(event: ActionEvent): void -updateEndDatePickerAccordingToStart(): void -setTableColumns(): void -createExpressCell(): TableCell<Order><Boolean>

«bind»

+OrdersHandlingControl

T: NonLoginControl

«abstract» NonLoginPageController

-minimizeButton: Button -resizeButton: Button -homeButton: Button -backButton: Button -RESIZE MARGIN: int = 5 -MIN WIDTH: int = 1080 -MIN_HEIGHT: int = 720 -MAX_IMG_WIDTH: double = 300 -MAX_IMG_HEIGHT: double = 173

+init(c: T): void

#setResizable(): void -handleSceneResizing(event: MouseEvent): void -handleCursorSwitch(event: MouseEvent): void

-setNavigationButtons(): void

-setButtonIconHoverEffect(): void ~minimizeButtonAction(event: ActionEvent): void ~resizeButtonAction(event: ActionEvent): void

~homeButtonAction(event: ActionEvent): void

+setTableFunctionality<U>(table: TableView<U>): void +setTableFunctionality<U>(table: TableView<U>, columnTitle: String, buttonText: String, action: Consumer<U>): void

-setRowDeselection<U>(table: TableView<U>): void -setColumnSize<U>(table: TableView<U>): void

-addButtonToTable<U>(table: TableView<U>, columnTitle: String, buttonText: String, action: Consumer<U>): void +setRoundImageViewImagesAndPosition(img: Image, imgView: ImageView): void

«bind» +LoginControl

LoginPageController

-emailTextField: TextField

-loginButton: MFXButton

-togglePswVisibility(): void

-passwordTextField: PasswordField

-togglePswVisibiltyButton: Button

-plainPasswordTextField: TextField

#initialize(control: LoginControl): void

~loginButtonAction(event: Action): void

-handleKeyPressed(event: KeyEvent): void

-handleToggleKeyPressed(event: KeyEvent): void

«bind»

+T T: BaseControl

«abstract» BasePageController

«Property»-control: T -ICON_WIDTH: int = 30 -ICON HEIGHT: int = 30 -titleBar: Node +exitButton: Button

#initialize(c: T): void +init(c: T): void +accessor methods() #setDraggable(): void #getIcon(path: String): ImageView ~exitButtonAction(event: ActionEvent): void

ShipmentPageController

-orderExpectedDeliveryDateLabel: Label -orderProductLabel: Label -orderQuantityLabel: Label -orderldLabel: Label -orderClientLabel: Label -orderAddressLabel: Label -shipmentDatePicker: DatePicker -shipmentsTable: TableView<Shipment> -shipmentIdTableColumn: TableColumn<Shipment, Integer> -shipmentDateTableColumn: TableColumn<Shipment, LocalDate> -startingDepositTableColumn: TableColumn<Shipment, String> -transportTableColumn: TableColumn<Shipment, String> -remaningSpaceTableColumn: TableColumn<Shipment, Float> -depositComboBox: ComboBox<Deposit> -transportComboBox: ComboBox<Transport> -driverComboBox: ComboBox<Driver> -submitButton: MFXButton -loadingPane: StackPane

#initialize(control: OrdersHandlingControl): void -displayCompatibleShipments(): ObservableList<Shipment> -setButtonEnablerListener(nodes: Node): void

-resetComboBoxes(): void ~backButtonAction(event: ActionEvent): void

-displayOrderData(): void

-populateComboBoxes(): void

-handleComboBoxNullValue<T>(comboBox: ComboBox<T>): void -setDatePickerLowerBound(): void

-«bind»

+GraphControl

-createDayCell(defaultDate: Date): DateCell -getDayCellStyle(item: LocalDate): String

-setTableColumns(): void -enableSubmitButton(): void

-isFormFilled(): boolean -isShipmentSelected(): boolean

-setSubmitButtonAction(): void

+OrdersHandlingControl

GraphPageController

-moneySpent: Label

-orderNumber: Label -highestCostEmail: Label

-highestCostNameSurname: Label

-highestCostOrderCost: Label

-highestCostOrderId: Label -highestCostAccount: Label

-highestCostOrderProduct: Label

-highestNumberEmail: Label

-highestNumberNameSurname: Label

-highestNumberOrderCost: Label

-highestProductNumberOrderAccount: Label

-highestProductNumberOrderCost: Label

-highestProductNumberOrderId: Label -highestProductNumberOrderProduct: Label

-lowestCostOrderAccount: Label

-lowestCostOrderCost: Label

-lowestCostOrderId: Label

-lowestCostOrderProduct: Label

-lowestProductNumberOrderAccount: Label -lowestProductNumberOrderCost: Label

-lowestProductNumberOrderId: Label

-lowestProductNumberOrderProduct: Label

-mostSpendingPropic: ImageView

-mostOrderingPropic: ImageView -xAxis: CategoryAxis

-yAxis: NumberAxis

-chart: LineChart<String, Number>

-monthComboBox: ComboBox<Month>

-yearComboBox: ComboBox<Year> -montlyReportData: Vbox

-scrollPane: ScrollPane

-PERCENTAGE_DIVISOR: int = 100

-initializeMonthComboBox(): void

-createMonthListCell(formatter: DateTimeFormatter): ListCell<Month>

-initializeYearComboBox(): void

-setInitialAsCapitalLetter(string: String): String

-displayMonthlyReportData(): void

-displayOrderData(): void

-displayMostOrderingAccountData(): void -displayMostSpendingAccountData(): void

-setGraphData(): void

-getAverageLine(ordersData: List<Integer>): XYChart.Series<String><Number> -getOrdersLine(ordersData: List<Integer>): XYChart.Series<String><Number>

-getAverage(listM: List<Number>): Float

-removeLineSymbol(line: XYChart.Series<String><Number>): void

-resetPieChart(): void

-initializeComboBoxes(): void

-setInitialAsCapitalLetter(string: String): String

-setPieChartData(): void

-setPopupOnDataHover(data: XYChart.Data<String><Number>, label: String): void

-setPopupOnDataHover(data: XYChart.Data<String><Number>, label: String): void