Customers
-custID: IntegerProperty
-(Name: StringProperty
-(Name: StringProperty
-(Type: StringProperty
-address: StringProperty
-postcode: StringProperty
-pnumber: StringProperty
-email: StringProperty +Customers(rs : ResultSet)
+getCustID() : int
+CustIDProperty() : IntegerProperty
+setCustID(custID : int) : void -setCustID(custID: int): void
-getName(): String-property
-setCustID(custID: int): void
-getName(): String-property
-setName(): String-property
-setName(): String-property
-setName(): String-property
-setName(): String-property
-setName(): String-property
-setName(): String-property
-setCype(): String
-setCype(): String
-setCype(): String
-setCype(): String
-setAddress(): Associated *setType(postcode : String) : void +getPnumber() : String +pnumberProperty() : StringProperty *setPnumber(pnumber : String) : void +getEmail() : String +emailProperty() : StringProperty *setEmail(email : String) : void -customers 👗 * CustomerAccountsGUIControl -editcustomerLabel : Label -addcustomerLabel : Label Associated -addbuttonOption : Button -editbuttonOption : Button -addbuttonPane : Button -savebuttonPane : Button -makeBookingOption : Button -nameRadio : RadioButton Associated -regRadio : RadioButton -pane : GridPane -tabPane : TabPane -tabPane : TabPane -bookingTab : Tab -searchBar : TextField -fname_field : TextField -lname_field : TextField -address_field : TextField -postcode_field : TextField -pnumber_field : TextField -email_field : TextField -email_field: TextField
-custLat: ObservableLats-String> = FXCollections.observableArrayList("Private","Business")
-customerType: ChoiceBox
-customerTable : TableView-Customers>
-lName: TableColumn-Customers, String>
-lName: TableColumn-Customers, String>
-cType: TableColumn-Customers, String>
-ddress: TableColumn-Customers, String> -postcode : TableColumn<Customers, String> -pnumber : TableColumn<Customers, Integer> -email: TableColumn<Customers, String>
-searchData: ObservableList<Customers> = FXCollections.observableArrayList() -partsTable: TableView-PartsAssigned>
-partsIb: TableColumn<PartsM, Integer>
-partsName: TableColumn<PartsM, String>
-partsDesc: TableColumn<PartsM, String> -partsStock : TableColumn<PartsM, String>
-partsStock : TableColumn<PartsM, Integer>
-partsCost : TableColumn<PartsM, Double>
-recordsTable : TableView<Records> records Table : Table/low-Records, StringsregNumber : Table/Column-Records, Stringsmodel : Table/Column-Records, Stringsmake : Table/Column-Records, Stringseng/Size : Table/Column-Records, Stringsrecords : Table/Column-Records, Stringsrecords : Table/Column-Records, Stringsbooking/Table : Table/Column-Records, Stringsrecords : Table/Strings : Table/Stringsrecords : Table/S -pbookingDate: TableColumn-Records, Integer>
db: SCILteConnection = SOLIteConnection_getInstance()
-customers: Customers = new ArrayList+c()
-reg: TableColumn-RartasAssigned, Stringy--partName: TableColumn-RartasAssigned, Stringy--spcTable: TableColumn-SPCEBooking, Stringy--spcAssigned: TableColumn-SPCEBooking, Stringy-vehicle_part: TableColumn-SPCEBooking, Stringy-vehicle_part: TableColumn-SPCEBooking, Stringy---DeliveryDate : TableColumn<SPCBooking, String> -returnDate : TableColumn<SPCBooking, String> -spcCost : TableColumn<SPCBooking, Integer>
-tbCost : TableColumn<Records, Double> +initialize(url : URL, rb : ResourceBundle) : void +searchByName() : void +openRepairsTab(event : ActionEvent) : void +fillTable() : void +fillPartsTableSearch() : void

+fillPartsTable(): void +fillPartsTable(): void +fillVehicleTableClick(ID: int): void +fillBookingTableSearch(ID: int): void -setValueFactories(): void

