## + start(primaryStage0: Stage): void Session - userName: String - role: String startedTransaction: boolean tick: int - transaction: Transaction · loggedIn: boolean Session() getUserName(): String getRole(): String getTransactionState(): boolean setStartedTransaction(startedTransaction: boolean): void setRole(role: String): void · setUserName(userName: String): void resetSession(): void setLoggedIn(b: boolean): void · isLoggedIn(): boolean getTransaction(): Transaction

**PasswordHash** 

isValidUser(salt: String, hashPassword: String, enteredPassword: String): boolean

getSaltedHashSHA512(password: String, salt: byte[]): byte[]

createHashedPassword(password: String): Password

- createSalt(): byte[]

fromHex(hex: String): byte[]

toHex(array: byte[]): String

App

- sceneManager: SceneManager

- primaryStage: Stage

+ main(args: String[]): void

Database dbConn: Connection - openStatement: Statement + Database(): void + openConn(): int + closeConn(): int + dropAllTables(): int + initialiseSchema(): int setUpInitialItemsAndUsers(): int setUpInitialCashAmounts(): int getCashSummary(): HashMap<String, Integer> + getTotalChange(): double + increaseCashQuantity(currency: String, quantityToUpdate: Integer): int + decreaseCashQuantity(currency: String, quantityToDeduct: Integer): int + queryRecent(): ArrayList<String> insertNewTransaction(): ArrayList<String> + queryRecent(): ArrayList<String> rinsertNewTransaction(status: String, user: String, reason: String): int + queryAllItemsByCategory(category: String): ArrayList<String> queryItemName(itemCode: String): String + queryItemPrice(itemCode: String): Double + queryItemQuantity(itemCode: String): int + queryUsernameAndRole(): HashMap<String, String> queryUsername(): ArrayList<String> + queryRoles(): ArrayList<String> + removeUser(username: String): int + changeRole(username: String, role: String): boolean insertNewUser(username: String, password: String, role: String): int + checkRole(username: String, role: String): boolean + getRole(username: String): String + validateUsername(username: String): int + login(username: String, password: String): int + getCard(username: String): String[] + insertNewCard(username: String, card: String, cvv: String): int + getAllItems(): ArrayList<String[]> ⊦ getItemSoldHistory(): ArrayList<String[]>

SceneManager - database: Database stage: Stage - ownerPortal: OwnerPortal defaultPage: DefaultPage - cashierPortal: CashierPortal - sellerPortal: SellerPortal login: Login - checkoutPage: CheckoutPage inputCashPage: InputCashPage - successfulPage: SuccessfulPage - session: Session - defaultPageController: DefaultPageController # reportsDirectory: String + SceneManager() setUp(): void + switchScenes(scene: Scene): void + getScene(): Scene setStage(stage: Stage): void setDefaultPage(defaultPage: Scene): void getDefaultPageScene(): Scene getClearedDefaultPageScene(): Scene + getOwnerPortalScene(): Scene getCashierPortalScene(): Scene getSellerPortalScene(): Scene + getLoginScene(): Scene getSuccessfulPage(): SuccessfulPage getCheckoutPageScene(): Scene getSuccessfulPageScene(): Scene getInputCashPageScene(): Scene F getDatabase(): Database getSession(): Session setDefaultPageController(dpc: DefaultPageController): void getDefaultPageController(): DefaultPageController

Transaction - sceneManager: SceneManager items: HashMap<String, Integer> - total: double - paid: double - due: double - change: double - changeOrder: ArrayList<String> - currentlyPaid: HashMap<String, Integer> changeAmount: DoubleProperty - totalAmount: DoubleProperty - dueAmount: DoubleProperty + Transaction() reset(): void + addItem(item: String): boolean + addToTotal(n: double): void + getTotal(): double + getItems(): HashMap<String, Integer> calculateTotal(): void ⊦ getChange(): double + getDue(): double removeItem(item: String): void ⊦ calculateChange(): void calculateDue(): void + getChangeOrder(): ArrayList<String> setPaid(paid: double): void + getPaid(): double + getChangeAmount(): DoubleProperty + getDueAmount(): DoubleProperty getTotalAmount(): DoubleProperty ⊦ initialHashMap(): void + addToCurrencyPaid(key: String): void + getCurrentlyPaid(): HashMap<String, Integer> getQuantityPaid(key: String): int setSceneManager(sceneManager: SceneManager): void

