Software Design Document

POS SIMULATOR APPLICATION

Version 1.0
Printed: 27 October,2023

POS SIMULATOR Team:
Indhu R PES1UG21CS237,
Harshitha Krishna R PES1UG21CS227,
Ganta Srilalitha PES1UG21CS200,
Jeshwanth Krishnama Naidu PES1UG21CS24

Table of Contents

1. Introduction

1.1 Purpose

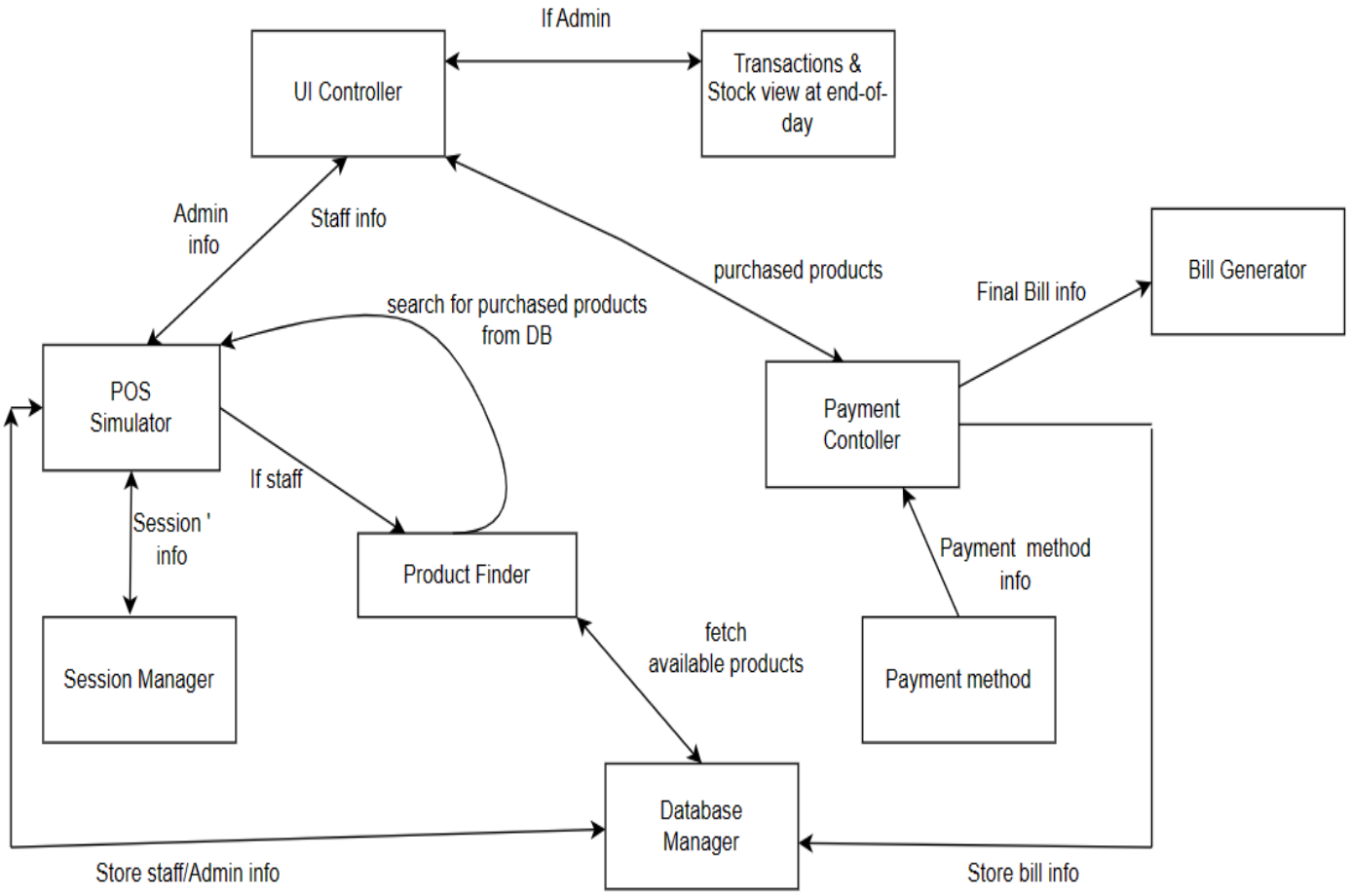
The Software Design Document describes the architecture and system design for POS Simulator.This document is intended for Project Managers, Software Engineers, and anyone else who will be involved in the implementation of the system.

1.2 Scope

The scope of the POS Simulator Web Application design document encompasses the complete description of the system's architecture, design, and components. It includes a detailed examination of the User Interface, Database, Transaction Processing, Reporting, Payment Processing, and Authentication and Security. The document covers the high-level overview of the system's purpose and objectives, defining its boundaries and functionalities. It addresses the interaction between different components, ensuring a comprehensive understanding of their dependencies. Additionally, the scope outlines the extent of the document's reach in providing insight into the system's architecture and design for project stakeholders and developers.

2. System Components

2.1 Decomposition Description



UI Controller:

User Interface Management: Manages the graphical user interface of the POS system, including layout, interactions, and visual elements.

Session Manager:

User Sessions: Tracks and manages user sessions, allowing cashiers to log in, switch users, or end their shifts.
Session Persistence: Ensures that data associated with a session (e.g., an ongoing transaction) is preserved until completion or until the session is closed.

Product Finder:

Product Search: Provides functionality to search for products in the product database based on criteria such as name, category, or barcode.
Product Information: Retrieves and displays detailed product information, including price,

description, and availability.
Quick Entry: Allows cashiers to quickly locate and add products to the current transaction.

Database Manager:

Product Data: Manages the product database, including adding, updating, and deleting product information.
Data Integrity: Ensures the accuracy and consistency of product data.
Data Backup and Recovery: Implements measures to back up and restore the database in case of data loss or system failures.

Transactions & Stock View at End of Day:

Transaction Records: Records and stores details of daily transactions, including sales, returns, and payments.
Inventory Management: Provides an end-of-day stock view, helping businesses track stock levels and identify items that need restocking.
Financial Reports: Generates reports summarizing daily sales and financial data.

Payment Controller:

Payment Processing: Manages the flow of payments, from selecting payment methods to processing payments, and handles any errors or exceptions.
Payment Validation: Validates the payment information provided by customers, ensuring it is accurate and secure.
Payment Confirmation: Confirms successful payments and updates transaction records accordingly.

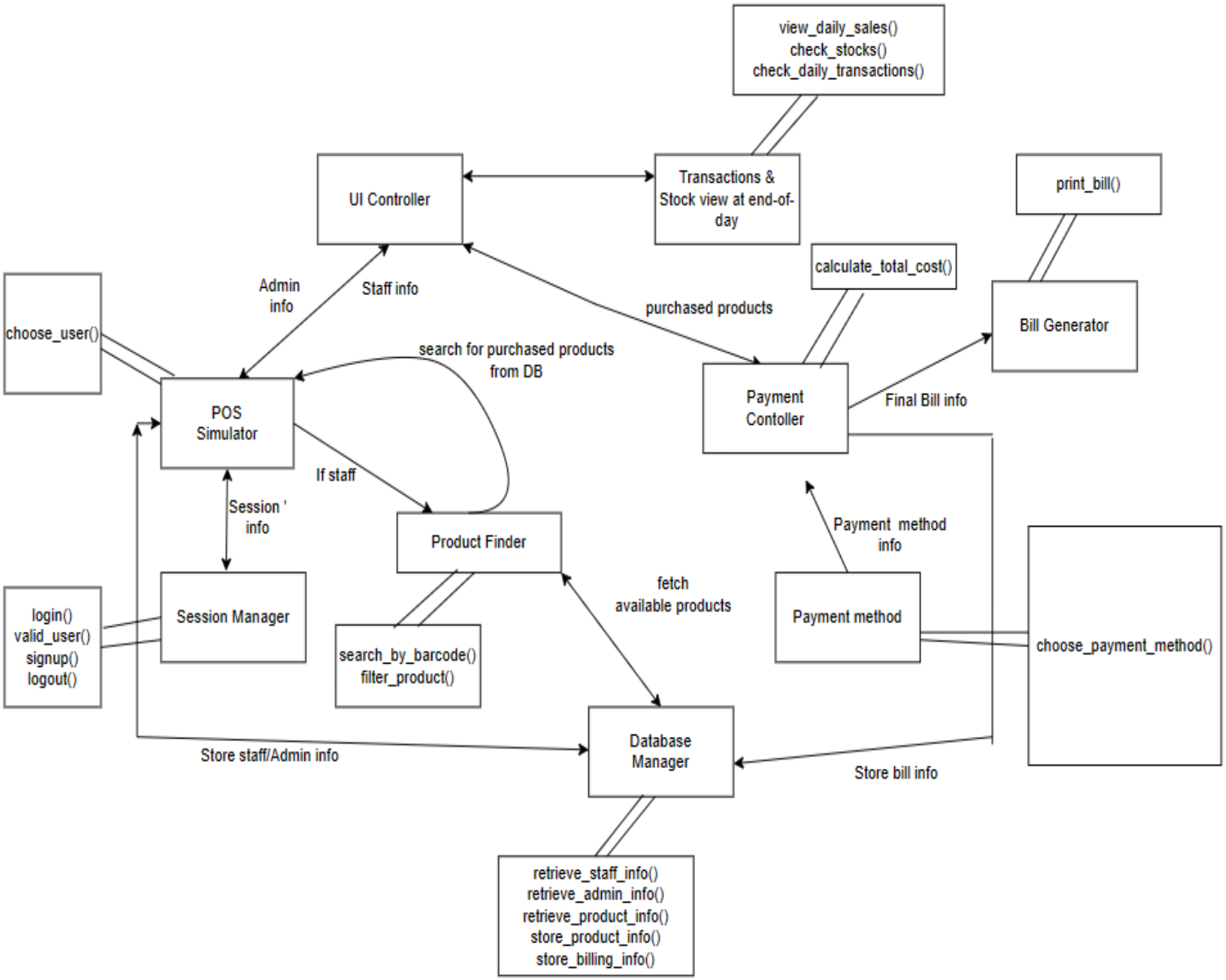
Payment Method:

Payment Options: Represents various payment methods such as cash, credit cards, debit cards, mobile wallets, and gift cards.
Payment Authorization: Handles the authorization process for each payment method and interacts with external payment processors.

Bill Generator:

Receipt Creation: Generates digital receipts for customers, including a breakdown of purchased items, prices, taxes, and total amount.
.

2.2 Dependency Description



- choose_user(): Allows the user to select their role (e.g.staff, admin) in the POS system.
- login(): Handles user authentication by requesting and verifying login credentials.
- valid_user(): Checks if the logged-in user has valid access to the system.
- signup(): Allows new users to create accounts and credentials for the POS system.
- logout(): Logs the user out of the POS system, ending their session.
- view_daily_sales(): Displays the daily sales and revenue statistics.
- check_stocks(): Provides information on the current inventory levels for products.
- check_daily_transactions(): Lists or summarizes the day's completed transactions.
- calculate_total_cost(): Calculates the total cost of selected items in the shopping cart.
- search_by_barcode(): Finds and displays product information by scanning or

entering a barcode.

- filter_product(): Allows users to filter and search for products based on specific criteria (e.g., category, price).
- choose_payment_method(): Lets the customer select the payment method (e.g., cash, credit card, mobile payment).
- print_bill(): Generates a receipt or invoice for the transaction, including itemized details and the total amount.
- retrieve_staff_info(): Fetches information about store staff (e.g., names, roles, contact details).
- retrieve_admin_info(): Retrieves information about the system administrators and their contact details.
- retrieve_product_info(): Gathers product details such as name, price, description, and availability.
- store_product_info(): Allows the addition or modification of product information in the database.
- store_billing_info(): Records and stores transaction and billing details in the system for record-keeping and reporting.