پروژه جامع سیستم مدیریت خط تولید کارخانه

تحلیل، طراحی و نمونه‌سازی سیستم

نام دانشجو: سوگل درس: مهندسی نرم‌افزار / تحلیل و طراحی سیستم‌ها تاریخ: ۲۰۲۶

---

۱) صفحه عنوان

پروژه جامع سیستم مدیریت خط تولید کارخانه

تحلیل، طراحی و نمونه‌سازی سیستم

نام دانشجو: سوگل

درس: مهندسی نرم‌افزار / تحلیل و طراحی سیستم‌ها

تاریخ: ۲۰۲۶

---

۲) چکیده

این پروژه به تحلیل، طراحی و نمونه‌سازی یک سیستم جامع مدیریت خط تولید کارخانه می‌پردازد. هدف اصلی این سیستم، افزایش بهره‌وری، کاهش هزینه‌ها و بهبود فرآیندهای عملیاتی در محیط‌های تولیدی است. سیستم پیشنهادی با مکانیزه‌سازی و اتوماسیون وظایف کلیدی مانند مدیریت موجودی مواد اولیه و محصولات نهایی، برنامه‌ریزی و کنترل خطوط تولید، مدیریت سفارشات خرید و فروش، و همچنین ارائه گزارش‌های تحلیلی، نقش حیاتی در تصمیم‌گیری‌های مدیریتی ایفا می‌کند. در این پروژه، نیازمندی‌های سیستم از طریق تکنیک‌های مهندسی نرم‌افزار شناسایی و مستندسازی شده‌اند. سپس، معماری سیستم در سطوح مختلف با استفاده از نمودارهای Sequence Diagram، Class Diagram، DFD، Use Case و ERD تحلیل و طراحی شده است. معماری سه‌لایه به عنوان رویکرد اصلی طراحی در نظر گرفته شده است تا قابلیت نگهداری، توسعه‌پذیری و مقیاس‌پذیری سیستم تضمین گردد. در نهایت، پیشنهاداتی برای نمونه‌سازی اولیه (Prototype) و ابزارهای مورد نیاز برای پیاده‌سازی ارائه شده

است. این سیستم به کارخانه‌ها امکان می‌دهد تا با مدیریت بهینه‌تر منابع و فرآیندها، مزیت رقابتی خود را در بازار افزایش دهند.

---

۳) مقدمه

در دنیای امروز، رقابت فشرده در بازارهای جهانی، سازمان‌ها را بر آن داشته است تا به دنبال راه‌هایی برای افزایش کارایی، کاهش هزینه‌ها و بهبود کیفیت محصولات خود باشند. کارخانه‌ها به عنوان یکی از پایه‌های اصلی اقتصاد، نقش حیاتی در تولید کالا و تأمین نیازهای جامعه دارند. با این حال، مدیریت کارآمد خطوط تولید، به ویژه در کارخانه‌های بزرگ و پیچیده، چالش‌های فراوانی را به همراه دارد.

روش‌های سنتی مدیریت تولید که اغلب بر پایه اتکا به نیروی انسانی، ثبت دستی اطلاعات و فرآیندهای کاغذی استوار هستند، با مشکلاتی نظیر خطاهای انسانی، کندی در پردازش اطلاعات، عدم شفافیت در وضعیت موجودی، دشواری در برنامه‌ریزی دقیق تولید، و از دست رفتن اطلاعات مهم مواجه‌اند. این مشکلات منجر به اتلاف منابع، افزایش زمان انتظار، و در نهایت کاهش رضایت مشتری می‌شود.

مکانیزه‌سازی و استفاده از فناوری اطلاعات در مدیریت تولید، راهکاری اساسی برای غلبه بر این چالش‌هاست. سیستم‌های مدیریت تولید مدرن، امکان خودکارسازی و هماهنگ‌سازی فرآیندهای حیاتی را فراهم می‌آورند. این سیستم‌ها با قابلیت‌های خود در کنترل موجودی مواد اولیه و محصولات نهایی، برنامه‌ریزی دقیق تولید بر اساس تقاضا، مدیریت سفارشات فروش، و تسهیل فرآیند تأمین مواد اولیه، نقشی کلیدی در ارتقاء عملکرد کارخانه‌ها ایفا می‌کنند.

یک سیستم جامع مدیریت خط تولید کارخانه، با ارائه دیدی یکپارچه و لحظه‌ای از تمام عملیات، به مدیران امکان می‌دهد تا تصمیمات آگاهانه‌تری اتخاذ کرده و منابع را بهینه‌تر تخصیص دهند. این پروژه به بررسی و طراحی چنین سیستمی با رویکردی دانشگاهی و مهندسی می‌پردازد و تلاش دارد تا چارچوبی استاندارد و قابل پیاده‌سازی برای بهبود فرآیندهای تولید در کارخانه‌ها ارائه دهد.

---

۴) شرح سیستم

سیستم مدیریت خط تولید کارخانه، یک سامانه نرم‌افزاری یکپارچه است که برای نظارت، کنترل و بهینه‌سازی تمامی فرآیندهای مرتبط با تولید در یک محیط کارخانه‌ای طراحی شده است. این سیستم با هدف افزایش بهره‌وری، کاهش ضایعات، بهبود کیفیت محصولات و اطمینان از تحویل به موقع سفارشات، به عنوان ستون فقرات عملیاتی کارخانه عمل می‌کند.

ذی‌نفعان سیستم (Stakeholders):

- مدیران تولید: برای نظارت بر عملکرد خطوط تولید، برنامه‌ریزی تولید، مدیریت منابع و تحلیل کارایی.
- مدیران انبار: برای کنترل دقیق موجودی مواد اولیه، محصولات نیمه‌ساخته و نهایی، مدیریت ورود و خروج کالا.
- مسئولین فروش: برای ثبت سفارشات مشتریان، پیگیری وضعیت سفارشات و ارتباط با بخش تولید.
- تأمین‌کنندگان: برای دریافت سفارشات خرید مواد اولیه و ارسال اطلاعات مرتبط.
- اپراتورهای خط تولید: برای دریافت دستورالعمل‌های تولید، ثبت وضعیت پیشرفت کار و گزارش مشکلات.
- مدیران سیستم (System Administrators): برای مدیریت کاربران، تنظیمات سیستم و نگهداری.

ماژول‌های اصلی سیستم:

- مدیریت محصول: تعریف، دسته‌بندی و مدیریت اطلاعات محصولات تولیدی.
- مدیریت مواد اولیه: تعریف، دسته‌بندی و مدیریت اطلاعات مواد اولیه مورد نیاز.
- مدیریت موجودی: ردیابی و کنترل دقیق موجودی مواد اولیه، محصولات نیمه‌ساخته و نهایی.
- برنامه‌ریزی تولید: تعریف برنامه‌های تولید، تخصیص منابع و زمان‌بندی عملیات.
- مدیریت خط تولید: نظارت بر عملکرد خطوط تولید، ثبت وضعیت پیشرفت و گزارش خطاها.
- مدیریت سفارشات خرید: ثبت، پیگیری و مدیریت سفارشات خرید مواد اولیه از تأمین‌کنندگان.
- مدیریت سفارشات فروش: ثبت، پیگیری و مدیریت سفارشات مشتریان.
- مدیریت تأمین‌کنندگان: ثبت و مدیریت اطلاعات تأمین‌کنندگان.
- گزارش‌گیری و تحلیل: ارائه گزارش‌های متنوع از وضعیت تولید، موجودی، فروش و عملکرد.
- مدیریت کاربران و مجوزها: کنترل دسترسی کاربران به بخش‌های مختلف سیستم.

جریان کلی عملیات:

سیستم با دریافت سفارشات فروش، فرآیند برنامه‌ریزی تولید را آغاز می‌کند. این برنامه‌ریزی بر اساس ظرفیت خطوط تولید، موجودی مواد اولیه و زمان‌بندی انجام می‌شود. در صورت نیاز به مواد اولیه، سیستم سفارشات خرید را برای تأمین‌کنندگان صادر می‌کند. پس از تأیید و دریافت مواد اولیه، فرآیند تولید آغاز شده و سیستم وضعیت پیشرفت کار در هر مرحله را رصد می‌کند. اطلاعات تولید و مصرف مواد اولیه به صورت خودکار در بخش مدیریت موجودی به‌روزرسانی می‌شود. پس از اتمام تولید، محصولات نهایی به انبار منتقل شده و آماده ارسال به مشتریان

هستند. سیستم به طور مداوم گزارش‌های مربوط به تمام این فرآیندها را تولید کرده و در اختیار مدیران قرار می‌دهد.

اهداف اصلی سیستم:

- افزایش شفافیت در تمامی فرآیندهای تولید.
- کاهش زمان چرخه تولید.
- بهینه‌سازی مصرف مواد اولیه و کاهش ضایعات.
- بهبود دقت در برنامه‌ریزی تولید و تحویل به موقع سفارشات.
- کاهش خطاهای انسانی از طریق اتوماسیون.
- فراهم کردن داده‌های معتبر برای تصمیم‌گیری‌های استراتژیک.

مزایای سیستم:

- افزایش بهره‌وری کلی کارخانه.
- کاهش هزینه‌های عملیاتی.
- بهبود کیفیت و یکنواختی محصولات.
- افزایش رضایت مشتریان.
- قابلیت انطباق سریع با تغییرات بازار.

دامنه سیستم:

سیستم فعلی بر مدیریت فرآیندهای اصلی خط تولید، انبارداری، سفارش‌گذاری و فروش تمرکز دارد. قابلیت‌های پیشرفته‌تر مانند مدیریت منابع انسانی، نگهداری و تعمیرات پیشگیرانه تجهیزات، یا ادغام کامل با سیستم‌های حسابداری مالی، می‌تواند در فازهای بعدی توسعه به دامنه سیستم افزوده شود.

---

۵) مهندسی نیازمندی‌ها

الف) بازیگران و موارد کاربرد اصلی

بازیگران (Actors):

- مدیر تولید (Production Manager):

  - نقش: مسئولیت اصلی نظارت، برنامه‌ریزی و کنترل فرآیندهای تولید.
  - تعاملات: تعریف محصولات، تعیین استانداردهای تولید، برنامه‌ریزی خطوط تولید، تخصیص وظایف به اپراتورها، نظارت بر پیشرفت تولید، بررسی گزارش‌های انحراف از برنامه، تأیید سفارشات خرید مواد اولیه.

- انباردار (Warehouse Keeper):

  - نقش: مدیریت فیزیکی و سیستمی موجودی مواد اولیه، محصولات نیمه‌ساخته و نهایی.
  - تعاملات: ثبت ورود مواد اولیه و محصولات، ثبت خروج کالا (به خط تولید یا مشتری)، به‌روزرسانی سطح موجودی، انجام شمارش دوره‌ای موجودی، گزارش کسری یا مازاد موجودی.

- مسئول فروش (Sales Representative):

  - نقش: مدیریت ارتباط با مشتریان و ثبت سفارشات.
  - تعاملات: ثبت سفارشات جدید از مشتریان، پیگیری وضعیت سفارشات، دریافت بازخورد مشتریان، اطلاع‌رسانی به بخش تولید در مورد اولویت‌ها.

- تأمین‌کننده (Supplier):

  - نقش: ارائه مواد اولیه مورد نیاز کارخانه.
  - تعاملات: دریافت سفارشات خرید از کارخانه، ارسال تأییدیه سفارش، اعلام تاریخ تحویل، گزارش وضعیت ارسال. (در این سیستم، تعاملات با تأمین‌کننده ممکن است از طریق ایمیل یا پورتال باشد که توسط سیستم تولید یا مستقیماً توسط کارشناس خرید انجام می‌گیرد. در سطح سیستم، تأمین‌کننده به عنوان یک موجودیت خارجی در نظر گرفته می‌شود که اطلاعات مربوط به او در سیستم ثبت می‌شود).

- مدیر سیستم (System Administrator):

  - نقش: مدیریت کاربران، تنظیمات کلی سیستم، پشتیبان‌گیری و بازیابی.
  - تعاملات: ایجاد، ویرایش و حذف حساب‌های کاربری، تعریف سطوح دسترسی، پیکربندی پارامترهای سیستم، نظارت بر عملکرد سیستم.

موارد کاربرد اصلی (Use Cases):

- مدیریت محصول (Manage Product):
  - شرح: امکان ثبت، ویرایش، حذف و مشاهده اطلاعات کامل محصولات نهایی (نام، کد، مشخصات فنی، واحد اندازه‌گیری، هزینه تولید تخمینی).
- مدیریت مواد اولیه (Manage Raw Material):
  - شرح: امکان ثبت، ویرایش، حذف و مشاهده اطلاعات کامل مواد اولیه (نام، کد، مشخصات، واحد اندازه‌گیری، حداقل موجودی).

- مدیریت انبار (Manage Inventory):
  - شرح: ثبت ورود کالا (مواد اولیه، محصولات)، ثبت خروج کالا (به خط تولید، فروش)، مشاهده وضعیت موجودی فعلی، انجام انتقالات بین انبارها (در صورت وجود)، اجرای عملیات شمارش موجودی.
- ثبت سفارش فروش (Create Sales Order):
  - شرح: امکان ثبت جزئیات سفارش مشتری (کد مشتری، لیست محصولات، تعداد، تاریخ تحویل مورد نظر)، بررسی اولیه موجودی یا ظرفیت تولید.
- برنامه‌ریزی تولید (Production Planning):
  - شرح: تعریف برنامه تولید برای دوره‌های زمانی مشخص، تخصیص محصولات به خطوط تولید، تعیین زمان‌بندی تولید.
- کنترل خط تولید (Production Line Control):
  - شرح: نمایش دستورالعمل‌های تولید برای هر خط، ثبت شروع و پایان عملیات تولید، گزارش مصرف مواد اولیه، گزارش محصولات تولید شده، ثبت عیوب و مشکلات در طول تولید.
- ثبت سفارش خرید (Create Purchase Order):
  - شرح: ایجاد سفارش خرید مواد اولیه بر اساس نیازهای اعلام شده توسط بخش تولید یا انبار، انتخاب تأمین‌کننده، تعیین کمیت و تاریخ تحویل.
- مدیریت تأمین‌کننده (Manage Supplier):
  - شرح: ثبت، ویرایش و مشاهده اطلاعات تأمین‌کنندگان (نام، آدرس، اطلاعات تماس، محصولات قابل ارائه).
- گزارش‌گیری (Reporting):
  - شرح: تولید گزارش‌های متنوع مانند گزارش موجودی انبار، گزارش تولید روزانه/هفتگی/ماهانه، گزارش فروش، گزارش کسری مواد اولیه، گزارش انحراف از برنامه تولید.
- پیش‌بینی فروش (Sales Forecasting):
  - شرح: (قابلیت پیشرفته) استفاده از داده‌های تاریخی فروش برای پیش‌بینی تقاضا در آینده.
- مدیریت کاربران (Manage Users):
  - شرح: ایجاد، ویرایش، حذف کاربران سیستم و تعیین سطوح دسترسی آن‌ها.

ب) جدول Backlog نیازمندی‌ها

شناسه	عنوان نیازمندی	شرح	اولویت
REQ-001	ثبت اطلاعات محصول	سیستم باید امکان ثبت نام، کد، مشخصات فنی، واحد اندازه‌گیری و هزینه تولید تخمینی برای هر محصول را فراهم کند.	بالا
REQ-002			بالا

شناسه	عنوان نیازمندی	شرح	اولویت
	مدیریت موجودی مواد اولیه	سیستم باید امکان ثبت ورود و خروج مواد اولیه، به‌روزرسانی خودکار موجودی و نمایش موجودی فعلی را داشته باشد.	
REQ-003	ثبت سفارش مشتری	سیستم باید به مسئول فروش امکان ثبت سفارش مشتری شامل نام مشتری، لیست محصولات، تعداد و تاریخ تحویل را بدهد.	بالا
REQ-004	برنامه‌ریزی تولید	مدیر تولید باید بتواند برنامه تولید را برای محصولات مختلف، در بازه‌های زمانی مشخص، با تعیین خط تولید و کمیت، ثبت کند.	بالا
REQ-005	گزارش موجودی انبار	سیستم باید گزارشی از موجودی فعلی تمامی اقلام انبار (مواد اولیه و محصولات نهایی) را با جزئیات (نام، کد، واحد، مقدار) ارائه دهد.	بالا
REQ-006	ثبت ورود مواد اولیه	انباردار باید بتواند ورود محموله‌های مواد اولیه به انبار را ثبت کند و سیستم به طور خودکار موجودی را به‌روز کند.	بالا
REQ-007	ثبت خروج مواد اولیه به خط تولید	سیستم باید امکان ثبت خروج مواد اولیه از انبار به سمت خط تولید را فراهم کند تا مصرف مواد ردیابی شود.	بالا
REQ-008	ثبت محصولات تولید شده	پس از اتمام تولید یک بچ (Batch)، اپراتور یا مدیر تولید باید بتواند تعداد محصولات تولید شده را در سیستم ثبت کند.	بالا
REQ-009	تعیین حداقل موجودی مواد اولیه	سیستم باید اجازه دهد برای هر ماده اولیه، حداقل میزان موجودی تعریف شود تا در صورت رسیدن موجودی به این حد، هشدار صادر شود.	متوسط
REQ-010	ثبت سفارش خرید مواد اولیه	مدیر تولید یا مسئول خرید باید بتواند بر اساس نیاز، سفارش خرید برای مواد اولیه را با تعیین تأمین‌کننده، کمیت و تاریخ تحویل، ثبت کند.	بالا
REQ-011	نمایش وضعیت سفارش فروش	مسئول فروش باید بتواند وضعیت هر سفارش مشتری (مثلاً در انتظار تولید، در حال تولید، آماده تحویل، تحویل شده) را مشاهده کند.	متوسط
REQ-012	ثبت عیوب تولید	در صورت بروز مشکل یا عیب در خط تولید، اپراتور باید بتواند نوع عیب و تعداد محصولات معیوب را ثبت کند.	متوسط

شناسه	عنوان نیازمندی	شرح	اولویت
REQ-013	مدیریت کاربران	مدیر سیستم باید بتواند کاربران جدید ایجاد کند، دسترسی آن‌ها را مدیریت و حساب‌های کاربری غیرفعال را غیرفعال کند.	بالا
REQ-014	گزارش تولید بر اساس دوره زمانی	سیستم باید امکان ارائه گزارش میزان تولید انجام شده برای یک محصول یا کل محصولات، در بازه‌های زمانی مشخص (روزانه، هفتگی، ماهانه) را فراهم کند.	بالا
REQ-015	مدیریت اطلاعات تأمین‌کنندگان	سیستم باید امکان ثبت، ویرایش و مشاهده اطلاعات تأمین‌کنندگان (نام، آدرس، تلفن، ایمیل، محصولات کلیدی) را فراهم کند.	متوسط
REQ-016	پیش‌بینی ساده فروش	سیستم باید بر اساس داده‌های فروش گذشته، یک پیش‌بینی ساده از فروش در دوره‌های آتی ارائه دهد.	پایین

---

۶) تحلیل سیستم

الف) توضیح متنی Use Case Diagram

در این بخش، روابط بین بازیگران (Actors) و موارد کاربرد (Use Cases) سیستم به صورت متنی شرح داده می‌شود، به جای ارائه نمودار گرافیکی.

- مدیر تولید (Production Manager):
  - این بازیگر با سیستم از طریق موارد کاربرد زیر تعامل دارد:
    - مدیریت محصول: مدیر تولید مسئول تعریف و به‌روزرسانی اطلاعات محصولات است.
    - مدیریت مواد اولیه: تعریف و مدیریت جزئیات مواد اولیه بر عهده مدیر تولید است.
    - برنامه‌ریزی تولید: مدیر تولید به طور مستقیم مسئول برنامه‌ریزی خطوط تولید است.
    - کنترل خط تولید: نظارت بر فرآیند تولید و ثبت نتایج آن (محصولات تولید شده، عیوب) توسط مدیر تولید انجام می‌شود.
    - ثبت سفارش خرید: مدیر تولید (یا شخص مسئول خرید) سفارشات خرید مواد اولیه را ثبت می‌کند.
    - گزارش‌گیری: مدیر تولید برای ارزیابی عملکرد، از گزارش‌های مختلف سیستم استفاده می‌کند.

- مدیریت کاربران: (در برخی سازمان‌ها) ممکن است مدیر تولید نیز دسترسی برای مدیریت کاربران سطح پایین‌تر داشته باشد.

- انباردار (Warehouse Keeper):

  - این بازیگر با سیستم از طریق موارد کاربرد زیر تعامل دارد:
  - مدیریت انبار: انباردار مسئول ثبت تمامی رویدادهای ورود و خروج کالا در انبار است.
  - مدیریت مواد اولیه: (مرتبط با موجودی) انباردار اطلاعات مربوط به موجودی مواد اولیه را با سیستم همگام می‌سازد.
  - گزارش‌گیری: انباردار ممکن است برای کنترل موجودی از گزارش‌های مربوطه استفاده کند.

- مسئول فروش (Sales Representative):

  - این بازیگر با سیستم از طریق موارد کاربرد زیر تعامل دارد:
  - ثبت سفارش فروش: مسئول فروش رابط اصلی ثبت سفارشات مشتریان در سیستم است.
  - گزارش‌گیری: مسئول فروش برای پیگیری وضعیت سفارشات و ارائه اطلاعات به مشتریان از گزارش‌ها استفاده می‌کند.
  - مدیریت محصول: (گاهی) برای دسترسی به اطلاعات محصولات و مشخصات آن‌ها.

- تأمین‌کننده (Supplier):

  - این بازیگر به طور مستقیم با رابط کاربری گرافیکی سیستم تعامل ندارد، اما سیستم با موجودیت تأمین‌کننده از طریق موارد کاربرد زیر در ارتباط است:
  - ثبت سفارش خرید: سیستم پس از ایجاد سفارش خرید، اطلاعات آن را برای پردازش در اختیار مدیر تولید یا مسئول خرید قرار می‌دهد که ارتباط با تأمین‌کننده را آغاز می‌کنند.
  - مدیریت تأمین‌کننده: اطلاعات تأمین‌کنندگان در سیستم نگهداری می‌شود.

- مدیر سیستم (System Administrator):

  - این بازیگر با سیستم از طریق موارد کاربرد زیر تعامل دارد:
  - مدیریت کاربران: اصلی‌ترین وظیفه مدیر سیستم.
  - تنظیمات سیستم: پیکربندی پارامترهای کلی سیستم.
  - پشتیبان‌گیری و بازیابی: (این مورد معمولاً در سطح زیرساخت نرم‌افزار مدیریت می‌شود اما دسترسی به آن از طریق ابزارهای مدیریتی سیستم فراهم است).

روابط بین بازیگران و موارد کاربرد:

- هر بازیگر می‌تواند با یک یا چند مورد کاربرد مرتبط با نقش خود تعامل داشته باشد.
- یک مورد کاربرد می‌تواند توسط یک یا چند بازیگر اجرا شود (مثلاً گزارش‌گیری می‌تواند توسط مدیر تولید و مسئول فروش استفاده شود).
- مواردی مانند "مدیریت محصول" و "مدیریت مواد اولیه" در اصل توسط مدیر تولید تعریف می‌شوند اما اطلاعات آن‌ها در سراسر سیستم استفاده می‌شود.

ب) تحلیل کلاس‌ها

در این بخش، کلاس‌های اصلی سیستم با ویژگی‌ها و مسئولیت‌هایشان شرح داده می‌شوند.

- کلاس: **`Product`** (محصول)
  - ویژگی‌ها: `productID` (شناسه منحصر به فرد)، `productName` (نام محصول)، `description` (شرح)، `technicalSpecifications` (مشخصات فنی)، `unitOfMeasure` (واحد اندازه‌گیری)، `estimatedCost` (هزینه تولید تخمینی).
  - مسئولیت‌ها: نگهداری اطلاعات مربوط به هر محصول نهایی.
- کلاس: **`RawMaterial`** (ماده اولیه)
  - ویژگی‌ها: `materialID` (شناسه منحصر به فرد)، `materialName` (نام ماده)، `description` (شرح)، `unitOfMeasure` (واحد اندازه‌گیری)، `minimumStockLevel` (حداقل سطح موجودی).
  - مسئولیت‌ها: نگهداری اطلاعات مربوط به هر ماده اولیه مورد نیاز.
- کلاس: **`Supplier`** (تأمین‌کننده)
  - ویژگی‌ها: `supplierID` (شناسه منحصر به فرد)، `supplierName` (نام تأمین‌کننده)، `contactPerson` (شخص تماس)، `phoneNumber` (شماره تماس)، `email` (ایمیل)، `address` (آدرس).
  - مسئولیت‌ها: نگهداری اطلاعات مربوط به تأمین‌کنندگان.
- کلاس: **`Warehouse`** (انبار)
  - ویژگی‌ها: `warehouseID` (شناسه منحصر به فرد)، `warehouseName` (نام انبار)، `location` (موقعیت).
  - مسئولیت‌ها: مدل‌سازی فیزیکی یا منطقی انبارها.

- کلاس: **`Inventory`** (موجودی)
  - ویژگی‌ها: `inventoryID` (شناسه منحصر به فرد)، `itemType` (نوع قلم: ماده اولیه یا محصول)، `itemID` (شناسه قلم: `materialID` یا `productID`)، `currentQuantity` (مقدار فعلی)، `lastUpdated` (تاریخ آخرین به‌روزرسانی).
  - مسئولیت‌ها: پیگیری سطح موجودی هر قلم (ماده اولیه یا محصول) در یک انبار مشخص. این کلاس به طور مستقیم با `Warehouse` و `Product` / `RawMaterial` ارتباط دارد.
- کلاس: **`ProductionLine`** (خط تولید)
  - ویژگی‌ها: `lineID` (شناسه منحصر به فرد)، `lineName` (نام خط تولید)، `capacity` (ظرفیت).
  - مسئولیت‌ها: مدل‌سازی خطوط تولید فیزیکی.
- کلاس: **`PurchaseOrder`** (سفارش خرید)
  - ویژگی‌ها: `orderID` (شناسه منحصر به فرد)، `orderDate` (تاریخ سفارش)، `expectedDeliveryDate` (تاریخ تحویل مورد انتظار)، `status` (وضعیت: ثبت شده، تأیید شده، تحویل شده)، `supplierID` (شناسه تأمین‌کننده).
  - مسئولیت‌ها: نگهداری اطلاعات مربوط به سفارشات خرید مواد اولیه. هر سفارش خرید شامل چندین قلم (مواد اولیه و کمیت مورد نیاز) است.
- کلاس: **`SalesOrder`** (سفارش فروش)
  - ویژگی‌ها: `orderID` (شناسه منحصر به فرد)، `orderDate` (تاریخ سفارش)، `deliveryDate` (تاریخ تحویل)، `status` (وضعیت: در انتظار، در حال تولید، آماده تحویل، تحویل شده)، `customerID` (شناسه مشتری - اگر بخش مدیریت مشتری جداگانه وجود نداشته باشد).
  - مسئولیت‌ها: نگهداری اطلاعات مربوط به سفارشات فروش. هر سفارش فروش شامل چندین قلم (محصولات و کمیت مورد نیاز) است.
- کلاس: **`User`** (کاربر)
  - ویژگی‌ها: `userID` (شناسه منحصر به فرد)، `username` (نام کاربری)، `passwordHash` (هش رمز عبور)، `role` (نقش: مدیر تولید، انباردار، مسئول فروش، مدیر سیستم).

- مسئولیت‌ها: مدیریت دسترسی و احراز هویت کاربران سیستم.

- کلاس‌های کمکی (مانند `OrderItem` , `PurchaseOrderItem`): این کلاس‌ها برای نگهداری لیست محصولات/مواد اولیه در سفارشات فروش/خرید استفاده می‌شوند.

  - کلاس: `OrderItem` (قلم سفارش فروش)
  - ویژگی‌ها: `productID` ، `quantity` ، `priceAtOrder` (قیمت در زمان سفارش).
  - کلاس: `PurchaseOrderItem` (قلم سفارش خرید)
  - ویژگی‌ها: `materialID` ، `quantity` ، `unitPrice` (قیمت واحد).

ج) DFD سطح صفر (Context Diagram)

در این سطح، سیستم به عنوان یک موجودیت واحد (System) در نظر گرفته می‌شود که با محیط خارجی (موجودیت‌های خارجی) تبادل داده دارد.

موجودیت‌های خارجی:

1. مدیر تولید:

   - جریان خروجی: اطلاعات برنامه‌ریزی تولید، دستورالعمل‌ها، گزارش‌های عملکرد.
   - جریان ورودی: اطلاعات محصولات، مواد اولیه، وضعیت تولید.

2. انباردار:

   - جریان خروجی: اطلاعات موجودی، درخواست‌های خروج/ورود.
   - جریان ورودی: اطلاعات ورود/خروج کالا، وضعیت موجودی.

3. مسئول فروش:

   - جریان خروجی: جزئیات سفارشات فروش، درخواست‌های وضعیت.
   - جریان ورودی: اطلاعات محصولات، جزئیات سفارشات جدید.

4. تأمین‌کننده:

   - جریان خروجی: سفارشات خرید.
   - جریان ورودی: تأییدیه سفارش، اطلاعات تحویل.

جریان داده سیستم:

- سیستم اطلاعات ورودی از تمام موجودیت‌های خارجی را پردازش کرده و اطلاعات لازم را برای آن‌ها تولید و ارسال می‌کند. به عنوان مثال، مدیر تولید اطلاعات مربوط به محصولات و مواد اولیه را به سیستم می‌دهد و سیستم در عوض گزارش‌های تولید و موجودی را به او ارائه می‌دهد. مسئول فروش سفارشات را به سیستم می‌دهد و سیستم وضعیت تولید آن سفارشات را به او بازمی‌گرداند.

د) DFD سطح یک

در این سطح، سیستم به زیرسامانه‌های اصلی تقسیم می‌شود و جریان داده بین این زیرسامانه‌ها و با موجودیت‌های خارجی نمایش داده می‌شود.

زیرسامانه‌های اصلی:

1. مدیریت انبار (Warehouse Management):

- ورودی: درخواست ورود کالا (از انباردار)، درخواست خروج کالا (از مدیریت تولید/فروش)، اطلاعات مواد اولیه (از مدیریت محصول)، اطلاعات محصولات (از مدیریت محصول).
- پردازش: ثبت ورود/خروج، به‌روزرسانی موجودی، بررسی حداقل موجودی.
- خروجی: وضعیت موجودی (به مدیریت تولید، فروش، گزارش‌گیری)، هشدار حداقل موجودی (به مدیر تولید/خرید).

2. مدیریت سفارش مواد اولیه (Raw Material Order Management):

- ورودی: نیاز به مواد اولیه (از مدیریت انبار/تولید)، اطلاعات مواد اولیه (از مدیریت محصول)، اطلاعات تأمین‌کنندگان (از مدیریت تأمین‌کننده).
- پردازش: ایجاد سفارش خرید، تخصیص تأمین‌کننده، پیگیری وضعیت سفارش.
- خروجی: سفارش خرید (به تأمین‌کننده)، وضعیت سفارش (به مدیر تولید).

3. مدیریت تولید (Production Management):

- ورودی: برنامه تولید (از مدیر تولید)، موجودی مواد اولیه (از مدیریت انبار)، اطلاعات محصول (از مدیریت محصول).
- پردازش: تخصیص وظیفه به خطوط تولید، ثبت شروع/پایان عملیات، ثبت مصرف مواد، ثبت محصولات تولید شده، ثبت عیوب.
- خروجی: درخواست خروج مواد اولیه (به مدیریت انبار)، محصولات تولید شده (به مدیریت انبار)، گزارش‌های تولید (به گزارش‌گیری)، وضعیت تولید (به مدیریت فروش).

4. مدیریت فروش (Sales Management):

- ورودی: سفارشات جدید (از مسئول فروش)، وضعیت تولید (از مدیریت تولید).
- پردازش: ثبت سفارش فروش، بررسی اولیه موجودی/ظرفیت، ارتباط با مدیریت تولید برای زمان‌بندی.
- خروجی: تأیید سفارش (به مسئول فروش)، درخواست خروج محصول (به مدیریت انبار)، وضعیت سفارش (به مسئول فروش).

5. گزارش‌گیری (Reporting):

- ورودی: داده‌های موجودی (از مدیریت انبار)، داده‌های تولید (از مدیریت تولید)، داده‌های فروش (از مدیریت فروش)، داده‌های سفارشات خرید (از مدیریت سفارش مواد اولیه).
- پردازش: جمع‌آوری، فیلتر کردن و سازماندهی داده‌ها.
- خروجی: گزارش‌های متنوع (به مدیر تولید، انباردار، مسئول فروش).

6. مدیریت محصول و مواد اولیه (Product & Raw Material Management):

- ورودی: اطلاعات جدید/اصلاح شده (از مدیر تولید).
- پردازش: ذخیره و مدیریت اطلاعات.
- خروجی: اطلاعات محصولات (به مدیریت انبار، تولید، فروش)، اطلاعات مواد اولیه (به مدیریت انبار، سفارش مواد اولیه).

7. مدیریت تأمین‌کننده (Supplier Management):

- ورودی: اطلاعات جدید/اصلاح شده (از مدیر تولید/خرید).
- پردازش: ذخیره و مدیریت اطلاعات.
- خروجی: اطلاعات تأمین‌کنندگان (به مدیریت سفارش مواد اولیه).

جریان داده بین زیرسامانه‌ها: * مدیریت انبار با مدیریت تولید، مدیریت فروش و مدیریت محصول/مواد اولیه در تعامل است. * مدیریت سفارش مواد اولیه با مدیریت انبار، مدیریت تأمین‌کننده و مدیر تولید در تعامل است. * مدیریت تولید با مدیریت انبار، مدیریت محصول/مواد اولیه و گزارش‌گیری در تعامل است. * مدیریت فروش با مدیریت انبار، مدیریت تولید و گزارش‌گیری در تعامل است. * مدیریت محصول/مواد اولیه با مدیریت انبار، مدیریت تولید و مدیریت فروش در تعامل است. * همه زیرسامانه‌ها داده‌های خود را برای گزارش‌گیری ارسال می‌کنند.

هـ) Sequence Diagram سناریوی ثبت سفارش مواد اولیه

این سناریو، گام به گام تعاملات بین موجودیت‌ها و کامپوننت‌های سیستم هنگام ثبت یک سفارش خرید مواد اولیه را نشان می‌دهد.

سناریو: ثبت سفارش خرید مواد اولیه توسط مدیر تولید.

شرکت‌کنندگان:

- Actor (مدیر تولید): بازیگر انسانی که عملیات را آغاز می‌کند.
- UI (User Interface): فرم یا صفحه‌ای که مدیر تولید از آن استفاده می‌کند (مثلاً فرم سفارش خرید).
- OrderController: مسئول کنترل منطق مربوط به سفارشات.
- InventoryService: سرویس مدیریت موجودی که نیاز به مواد اولیه را بررسی می‌کند.
- SupplierService: سرویس مدیریت تأمین‌کنندگان.
- Database: پایگاه داده برای ذخیره اطلاعات.

ترتیب پیام‌ها:

1. Actor -> UI: "ورود به فرم سفارش خرید"
2. UI -> OrderController: "نمایش فرم سفارش خرید"
3. OrderController -> UI: "نمایش فیلدهای فرم (ماده اولیه، کمیت، تأمین‌کننده)"
4. Actor -> UI: "انتخاب ماده اولیه، ورود کمیت، انتخاب تأمین‌کننده"
5. UI -> OrderController: "ارسال اطلاعات سفارش خرید"
6. OrderController -> InventoryService: "بررسی نیاز به ماده اولیه (اختیاری، ممکن است بر اساس موجودی قبلی باشد)"
7. InventoryService -> Database: "Query برای موجودی ماده اولیه"
8. Database --> InventoryService: "بازگرداندن اطلاعات موجودی"
9. InventoryService --> OrderController: "پاسخ (مثلاً: موجودی کافی است/نیست)"
10. OrderController -> SupplierService: "دریافت اطلاعات تأمین‌کننده (در صورت نیاز)"
11. SupplierService -> Database: "Query برای اطلاعات تأمین‌کننده"
12. Database --> SupplierService: "بازگرداندن اطلاعات تأمین‌کننده"
13. SupplierService --> OrderController: "ارسال اطلاعات تأمین‌کننده"
14. OrderController -> Database: "ذخیره سفارش خرید جدید" (شامل اطلاعات ماده اولیه، کمیت، تأمین‌کننده، تاریخ)
15. Database --> OrderController: "تأیید ذخیره"
16. OrderController -> UI: "نمایش پیام موفقیت آمیز بودن ثبت سفارش"
17. UI --> Actor: "نمایش پیام موفقیت آمیز بودن ثبت سفارش"

و) ERD (Entity-Relationship Diagram)

در این بخش، موجودیت‌های اصلی سیستم، ویژگی‌های کلیدی آن‌ها و نوع روابط بین آن‌ها به صورت متنی شرح داده می‌شود.

- موجودیت‌ها (Entities):
  - Product: (شناسه محصول، نام، مشخصات، واحد، هزینه)
  - RawMaterial: (شناسه ماده، نام، مشخصات، واحد، حداقل موجودی)
  - Supplier: (شناسه تأمین‌کننده، نام، آدرس، تماس)
  - Warehouse: (شناسه انبار، نام، موقعیت)
  - Inventory: (شناسه موجودی، شناسه قلم، نوع قلم، مقدار، تاریخ به‌روزرسانی، شناسه انبار) - این موجودیت وضعیت موجودی یک قلم را در یک انبار خاص نشان می‌دهد.
  - ProductionLine: (شناسه خط، نام، ظرفیت)
  - PurchaseOrder: (شناسه سفارش خرید، تاریخ سفارش، تاریخ تحویل، وضعیت، شناسه تأمین‌کننده)
  - PurchaseOrderItem: (شناسه قلم سفارش خرید، شناسه سفارش خرید، شناسه ماده اولیه، کمیت، قیمت واحد) - این موجودیت اقلام داخل یک سفارش خرید را نگه می‌دارد.
  - SalesOrder: (شناسه سفارش فروش، تاریخ سفارش، تاریخ تحویل، وضعیت، شناسه مشتری - اختیاری)
  - SalesOrderItem: (شناسه قلم سفارش فروش، شناسه سفارش فروش، شناسه محصول، کمیت، قیمت نهایی) - این موجودیت اقلام داخل یک سفارش فروش را نگه می‌دارد.
  - User: (شناسه کاربر، نام کاربری، هش رمز، نقش)
- روابط (Relationships):
  - Product به SalesOrderItem: یک `Product` می‌تواند در چندین `SalesOrderItem` وجود داشته باشد. هر `SalesOrderItem` مربوط به یک `Product` است.
    - نوع رابطه: یک به چند (One-to-Many) از Product به SalesOrderItem.

- RawMaterial به PurchaseOrderItem: یک `RawMaterial` می‌تواند در چندین `PurchaseOrderItem` وجود داشته باشد. هر `PurchaseOrderItem` مربوط به یک `RawMaterial` است.
  - نوع رابطه: یک به چند (One-to-Many) از RawMaterial به PurchaseOrderItem.
- SalesOrder به SalesOrderItem: یک `SalesOrder` شامل چندین `SalesOrderItem` است. هر `SalesOrderItem` بخشی از یک `SalesOrder` است.
  - نوع رابطه: یک به چند (One-to-Many) از SalesOrder به SalesOrderItem.
- PurchaseOrder به PurchaseOrderItem: یک `PurchaseOrder` شامل چندین `PurchaseOrderItem` است. هر `PurchaseOrderItem` بخشی از یک `PurchaseOrder` است.
  - نوع رابطه: یک به چند (One-to-Many) از PurchaseOrder به PurchaseOrderItem.
- Supplier به PurchaseOrder: یک `Supplier` می‌تواند چندین `PurchaseOrder` دریافت کند. هر `PurchaseOrder` برای یک `Supplier` صادر می‌شود.
  - نوع رابطه: یک به چند (One-to-Many) از Supplier به PurchaseOrder.
- Inventory به Warehouse: یک `Warehouse` می‌تواند شامل چندین `Inventory` (رکورد موجودی برای اقلام مختلف) باشد. هر `Inventory` مربوط به یک `Warehouse` است.
  - نوع رابطه: یک به چند (One-to-Many) از Warehouse به Inventory.
- Inventory به Product/RawMaterial: موجودی (`Inventory`) می‌تواند برای یک `Product` یا یک `RawMaterial` باشد. این ارتباط پیچیده‌تر است و معمولاً از طریق فیلدهای `itemID` و `itemType` در کلاس

`Inventory` مدیریت می‌شود. در مدل ERD، می‌توان این را به صورت زیر نمایش داد:

- یک `Product` می‌تواند دارای چندین رکورد `Inventory` باشد. (یک به چند)
- یک `RawMaterial` می‌تواند دارای چندین رکورد `Inventory` باشد. (یک به چند)
- هر `Inventory` مربوط به یک `Product` یا یک `RawMaterial` است. (ارتباط کلی)

- ProductionLine به کار: (این رابطه معمولاً در قالب فرآیندهای تولید و دستورالعمل‌ها تعریف می‌شود و مستقیماً در ERD اصلی نیست، مگر اینکه بخواهیم شیفت‌های تولید یا بچ‌های تولید را مدل کنیم. اما برای سادگی، فعلاً آن را در این سطح در نظر نمی‌گیریم.)

- User به اعمال: کاربران (`User`) با سیستم تعامل دارند و عملکردهای مختلفی را انجام می‌دهند. رابطه بین `User` و سایر موجودیت‌ها معمولاً از طریق لاگ‌های عملیاتی یا مالکیت (مثلاً کسی که یک سفارش را ثبت کرده) مشخص می‌شود که در این سطح، به صورت غیرمستقیم نشان داده می‌شود.

---

۷) طراحی سیستم

الف) معماری سه‌لایه

معماری سه‌لایه یک الگوی طراحی محبوب در توسعه نرم‌افزار است که سیستم را به سه لایه منطقی تقسیم می‌کند: لایه ارائه (Presentation Layer)، لایه منطق کسب‌وکار (Business Logic Layer) و لایه داده (Data Layer). این معماری به جداسازی نگرانی‌ها (Separation of Concerns) کمک کرده و باعث افزایش قابلیت نگهداری، توسعه‌پذیری و تست‌پذیری سیستم می‌شود.

- ۱. لایه ارائه (Presentation Layer):

- وظایف: این لایه مسئول تعامل با کاربر نهایی است. وظیفه اصلی آن نمایش اطلاعات به کاربر و دریافت ورودی از اوست. رابط کاربری گرافیکی (GUI)، وب‌سایت‌ها، یا برنامه‌های دسکتاپ بخشی از این لایه هستند. این لایه هیچ منطق کسب‌وکاری را اجرا نمی‌کند و صرفاً اطلاعات را از لایه منطق کسب‌وکار دریافت کرده و نمایش می‌دهد و ورودی کاربر را به این لایه ارسال می‌کند.

- مزایا: امکان تغییر سریع رابط کاربری بدون تأثیر بر منطق اصلی سیستم. پشتیبانی از پلتفرم‌های مختلف (وب، موبایل، دسکتاپ) با یک منطق کسب‌وکار مشترک.
- نحوه تعامل: از طریق فراخوانی متدهای لایه منطق کسب‌وکار و دریافت پاسخ از آن.

- ۲. لایه منطق کسب‌وکار (Business Logic Layer - BLL):

- وظایف: این لایه قلب سیستم است و تمامی قوانین، منطق‌ها و فرآیندهای کسب‌وکار را پیاده‌سازی می‌کند. این لایه درخواست‌ها را از لایه ارائه دریافت کرده، آن‌ها را پردازش می‌کند (با اعمال قوانین کسب‌وکار)، و در صورت نیاز با لایه داده برای دسترسی به اطلاعات تعامل می‌کند. همچنین، پاسخ‌ها را به لایه ارائه برمی‌گرداند. این لایه مستقل از نحوه نمایش اطلاعات به کاربر عمل می‌کند.
- مزایا: جداسازی منطق اصلی از رابط کاربری و پایگاه داده. امکان استفاده مجدد از منطق کسب‌وکار در چندین رابط کاربری. تسهیل تست منطق کسب‌وکار.
- نحوه تعامل: دریافت درخواست‌ها از لایه ارائه، فراخوانی متدهای لایه داده برای دسترسی به اطلاعات، پردازش داده‌ها و بازگرداندن نتایج به لایه ارائه.

- ۳. لایه داده (Data Layer - DAL):

- وظایف: این لایه مسئول دسترسی و مدیریت داده‌ها است. وظیفه اصلی آن ذخیره، بازیابی، به‌روزرسانی و حذف داده‌ها از پایگاه داده (یا هر نوع ذخیره‌سازی داده دیگر) است. این لایه تمامی عملیات CRUD (Create, Read, Update, Delete) را انجام می‌دهد. این لایه هیچ منطق کسب‌وکاری یا واسط کاربری ندارد.
- مزایا: انتزاع نحوه ذخیره‌سازی داده‌ها. امکان تغییر نوع پایگاه داده (مثلاً از SQL Server به PostgreSQL) بدون تأثیر بر لایه‌های بالاتر.
- نحوه تعامل: دریافت درخواست‌های داده از لایه منطق کسب‌وکار، اجرای کوئری‌ها بر روی پایگاه داده، و بازگرداندن داده‌های خام به لایه منطق کسب‌وکار.

نحوه تعامل لایه‌ها:

- درخواست کاربر از لایه ارائه آغاز می‌شود.
- لایه ارائه درخواست را به لایه منطق کسب‌وکار ارسال می‌کند.
- لایه منطق کسب‌وکار قوانین را اعمال کرده و در صورت نیاز، داده‌ها را از لایه داده درخواست می‌کند.
- لایه داده عملیات لازم را بر روی پایگاه داده انجام داده و داده‌های مورد نیاز را به لایه منطق کسب‌وکار برمی‌گرداند.
- لایه منطق کسب‌وکار با استفاده از داده‌های دریافتی، نتیجه را پردازش کرده و به لایه ارائه بازمی‌گرداند.
- لایه ارائه نتیجه را به کاربر نمایش می‌دهد.

این معماری اطمینان حاصل می‌کند که هر لایه تنها مسئولیت‌های مشخصی را بر عهده دارد و وابستگی بین لایه‌ها حداقل است.

ب) کلاس‌های فاز طراحی

در این بخش، نسخه‌ی دقیق‌تری از کلاس‌های اصلی که در فاز طراحی در نظر گرفته می‌شوند، به همراه ویژگی‌های مهم، متدهای کلیدی و وابستگی‌هایشان تشریح می‌شوند. این کلاس‌ها اغلب با اضافه شدن جزئیات پیاده‌سازی و منطق‌های دقیق‌تر، تکامل می‌یابند.

- کلاس: **`Product`**
  - ویژگی‌ها: `ProductID` (int, PK)، `ProductName` (string)، `Description` (string)، `TechnicalSpecs` (string)، `UnitOfMeasure` (string)، `EstimatedCost` (decimal).
  - متدهای مهم: `CreateProduct()` , `UpdateProduct()` , `DeleteProduct()` , `GetProductDetails()` .
  - وابستگی‌ها: ندارد (موجودیت مستقل).
- کلاس: **`RawMaterial`**
  - ویژگی‌ها: `MaterialID` (int, PK)، `MaterialName` (string)، `Description` (string)، `UnitOfMeasure` (string)، `MinimumStockLevel` (int).
  - متدهای مهم: `CreateRawMaterial()` , `UpdateRawMaterial()` , `DeleteRawMaterial()` , `GetRawMaterialDetails()` .
  - وابستگی‌ها: ندارد.
- کلاس: **`Supplier`**
  - ویژگی‌ها: `SupplierID` (int, PK)، `SupplierName` (string)، `ContactPerson` (string)، `PhoneNumber` (string)، `Email` (string)، `Address` (string).
  - متدهای مهم: `CreateSupplier()` , `UpdateSupplier()` , `DeleteSupplier()` , `GetSupplierDetails()` .
  - وابستگی‌ها: ندارد.

- کلاس: **`Inventory`**
  - ویژگی‌ها: `InventoryID` (int, PK)، `ItemID` (int)، `ItemType` (enum: Material, Product)، `Quantity` (int)، `LastUpdated` (DateTime)، `WarehouseID` (int, FK).
  - متدهای مهم: `AddStock(itemID, itemType, quantity, warehouseID)`، `RemoveStock(itemID, itemType, quantity, warehouseID)`، `GetCurrentStock(itemID, itemType, warehouseID)`، `CheckMinimumStock(itemID, itemType, warehouseID)`.
  - وابستگی‌ها: `Warehouse`، `Product`، `RawMaterial`.
- کلاس: **`PurchaseOrder`**
  - ویژگی‌ها: `OrderID` (int, PK)، `OrderDate` (DateTime)، `ExpectedDeliveryDate` (DateTime)، `Status` (enum: Draft, Approved, Shipped, Received)، `SupplierID` (int, FK).
  - متدهای مهم: `CreatePurchaseOrder()`، `ApprovePurchaseOrder()`، `ReceivePurchaseOrder()`، `GetOrderDetails()`.
  - وابستگی‌ها: `Supplier`، `PurchaseOrderItem`.
- کلاس: **`PurchaseOrderItem`**
  - ویژگی‌ها: `OrderItemID` (int, PK)، `OrderID` (int, FK)، `MaterialID` (int, FK)، `Quantity` (int)، `UnitPrice` (decimal).
  - متدهای مهم: (معمولاً متدها در سطح `PurchaseOrder` مدیریت می‌شوند).
  - وابستگی‌ها: `PurchaseOrder`، `RawMaterial`.
- کلاس: **`SalesOrder`**
  - ویژگی‌ها: `OrderID` (int, PK)، `OrderDate` (DateTime)، `DeliveryDate` (DateTime)، `Status` (enum: Pending, InProduction, ReadyForDelivery, Delivered)، `CustomerID` (int, FK - اگر مدیریت مشتری جداگانه باشد).
  - متدهای مهم: `CreateSalesOrder()`، `UpdateOrderStatus()`، `GetOrderDetails()`.

- وابستگی‌ها: `SalesOrderItem` .

- کلاس: **`SalesOrderItem`**

- ویژگی‌ها: `OrderItemID` (int, PK)، `OrderID` (int, FK)، `ProductID` (int, FK)، `Quantity` (int)، `PriceAtOrder` (decimal).
- متدهای مهم: (معمولاً متدها در سطح `SalesOrder` مدیریت می‌شوند).
- وابستگی‌ها: `SalesOrder` , `Product` .

- کلاس: **`User`**

- ویژگی‌ها: `UserID` (int, PK)، `Username` (string)، `PasswordHash` (string)، `Role` (enum: ProductionManager, WarehouseKeeper, SalesRepresentative, SystemAdmin).
- متدهای مهم: `AuthenticateUser(username, password)` , `GetUserPermissions()` .
- وابستگی‌ها: ندارد (به طور مستقیم، اما نقش او در سطح منطق کسب‌وکار تعیین‌کننده دسترسی‌هاست).

- کلاس: **`ProductionPlan`** (جدید)

- ویژگی‌ها: `PlanID` (int, PK)، `PlanDate` (DateTime)، `StartDate` (DateTime)، `EndDate` (DateTime).
- متدهای مهم: `CreateProductionPlan()` , `AssignToProductionLine()` .
- وابستگی‌ها: `ProductionLine` .

- کلاس: **`ProductionTask`** (جدید)

- ویژگی‌ها: `TaskID` (int, PK)، `PlanID` (int, FK)، `ProductID` (int, FK)، `ProductionLineID` (int, FK)، `QuantityToProduce` (int)، `Status` (enum: Pending, InProgress, Completed, Cancelled)، `StartTime` (DateTime)، `EndTime` (DateTime)، `ConsumedRawMaterials` (List of ConsumedMaterial).
- متدهای مهم: `StartTask()` , `CompleteTask()` , `RecordConsumption()` .
- وابستگی‌ها: `ProductionPlan` , `Product` , `ProductionLine` .

- کلاس کمکی: **`ConsumedMaterial`**
  - ویژگی‌ها: `MaterialID` (int, FK)، `Quantity` (int).

ج) Component Diagram به صورت متنی

کامپوننت دیاگرام، سیستم را به کامپوننت‌های منطقی تقسیم می‌کند و وابستگی‌ها و روابط بین آن‌ها را نشان می‌دهد. این یک دید کلی از ساختار سیستم ارائه می‌دهد.

- کامپوننت‌ها:

1. Inventory Management (مدیریت موجودی):

- مسئولیت: پیگیری و کنترل موجودی مواد اولیه و محصولات نهایی.
- رابط‌ها (Interfaces): `IInventoryService` (برای افزودن/حذف موجودی، دریافت موجودی)، `IWarehouseService` .
- وابستگی‌ها: `Database` .

2. Production Management (مدیریت تولید):

- مسئولیت: برنامه‌ریزی، کنترل و نظارت بر خطوط تولید. ثبت محصولات تولید شده و مصرف مواد.
- رابط‌ها: `IProductionPlanningService` , `IProductionLineControlService` , `IProductionTaskService` .
- وابستگی‌ها: `Inventory Management` , `Database` .

3. Purchase Management (مدیریت خرید):

- مسئولیت: مدیریت سفارشات خرید مواد اولیه و تأمین‌کنندگان.
- رابط‌ها: `IPurchaseOrderService` , `ISupplierService` .
- وابستگی‌ها: `Inventory Management` , `Supplier` , `Management` , `Database` .

4. Sales Management (مدیریت فروش):

- مسئولیت: مدیریت سفارشات فروش مشتریان و پیگیری وضعیت آن‌ها.
- رابط‌ها: `ISalesOrderService` .
- وابستگی‌ها: `Production Management` , `Inventory` , `Management` , `Database` .

5. Reporting (گزارش‌گیری):

- مسئولیت: تولید گزارش‌های متنوع از داده‌های سیستم.
- رابط‌ها: `IReportingService` .
- وابستگی‌ها: `Production Management` , `Inventory Management` , `Sales Management` , `Purchase Management` , `Database Management` .

6. Authentication (احراز هویت):

- مسئولیت: مدیریت کاربران، احراز هویت و مجوزها.
- رابط‌ها: `IAuthenticationService` , `IUserService` .
- وابستگی‌ها: `Database` .

7. Database (پایگاه داده):

- مسئولیت: ذخیره‌سازی تمامی داده‌های سیستم.
- رابط‌ها: (وابستگی برای کامپوننت‌های دیگر).

- نحوه ارتباط کامپوننت‌ها:

  - Authentication با تمام کامپوننت‌های دیگر تعامل دارد تا دسترسی کاربران را بر اساس نقششان کنترل کند.
  - Inventory Management با `Production Management` (برای مصرف مواد) و `Sales Management` (برای بررسی موجودی محصولات) تعامل دارد.
  - Production Management با `Inventory Management` (برای تأمین مواد و ثبت محصولات) و `Sales Management` (برای اطلاع از وضعیت سفارشات) تعامل دارد.
  - Purchase Management با `Inventory Management` (برای افزودن مواد اولیه پس از خرید) و `Supplier Management` (که می‌تواند یک کامپوننت جداگانه باشد یا بخشی از Purchase Management) تعامل دارد.
  - Sales Management با `Production Management` (برای ارسال سفارشات جهت تولید) و `Inventory Management` (برای بررسی موجودی) تعامل دارد.
  - Reporting با تمامی کامپوننت‌های دیگر (که داده تولید می‌کنند) تعامل دارد تا گزارش‌ها را بسازد.
  - تمام کامپوننت‌های منطقی (به جز Database) به `Database` وابستگی دارند.

این تقسیم‌بندی به توسعه‌دهندگان اجازه می‌دهد تا روی بخش‌های خاصی از سیستم تمرکز کنند و نگهداری و به‌روزرسانی آن را آسان‌تر می‌کند.

---

۸) پیشنهاد نمونه نرم‌افزار (Prototype)

نمونه اولیه (Prototype) این سیستم به منظور نمایش قابلیت‌های اصلی و جمع‌آوری بازخورد از ذی‌نفعان کلیدی طراحی خواهد شد. هدف اصلی از ساخت Prototype، اعتبارسنجی مفاهیم طراحی، درک بهتر نیازمندی‌ها و ایجاد یک تصور اولیه از تجربه کاربری است، نه پیاده‌سازی کامل تمامی ویژگی‌ها.

محتوای پیشنهادی برای نمونه اولیه:

- داشبورد اصلی: یک صفحه نمایش خلاصه که وضعیت کلی سیستم را نشان می‌دهد، مثلاً تعداد سفارشات در انتظار، موجودی بحرانی مواد اولیه، وضعیت کلی خطوط تولید (فعال/غیرفعال).
- فرم ثبت محصول: امکان افزودن اطلاعات پایه یک محصول جدید (نام، کد، واحد).
- فرم ثبت ماده اولیه: امکان افزودن اطلاعات پایه یک ماده اولیه جدید (نام، کد، واحد).
- فرم سفارش خرید: قابلیت ایجاد یک سفارش خرید ساده برای یک ماده اولیه از یک تأمین‌کننده مشخص با تعیین کمیت.
- صفحه موجودی انبار: نمایش لیستی از مواد اولیه و محصولات موجود در انبار به همراه مقادیر آن‌ها. امکان فیلتر و جستجو.
- صفحه ثبت سفارش فروش: امکان ثبت یک سفارش فروش ساده شامل انتخاب محصول و تعیین تعداد.
- صفحه گزارش‌ها: نمایش حداقل یک یا دو گزارش کلیدی، مانند "گزارش موجودی فعلی انبار" یا "لیست سفارشات فروش".

ویژگی‌های کلیدی که در نمونه اولیه به صورت نمایشی (Mock-up) یا نیمه‌کاره پیاده‌سازی می‌شوند:

- کاربرپسندی: رابط کاربری باید ساده، شهودی و دوستانه باشد.
- ناوبری آسان: کاربران باید بتوانند به راحتی بین بخش‌های مختلف نمونه اولیه جابجا شوند.
- تأکید بر جریان کار: تمرکز بر نمایش جریان اصلی عملیات (مثلاً از ثبت سفارش فروش تا نمایش آمادگی تولید).

ابزارهای پیشنهادی برای نمونه‌سازی:

1. برای رابط کاربری (UI) و تجربه کاربری (UX):

- HTML/CSS/JavaScript: برای ساخت صفحات وب تعاملی.
- Bootstrap: یک فریم‌ورک CSS محبوب که امکان طراحی سریع و ریسپانسیو را فراهم می‌کند و ظاهر حرفه‌ای به Prototype می‌دهد.
- Figma / Adobe XD: (ابزارهای طراحی واسط کاربری) برای ایجاد ماکت‌های تعاملی و وایرفریم‌های با کیفیت بالا بدون نیاز به کدنویسی. این ابزارها برای نمایش جریان کار و ظاهر نهایی عالی هستند.

2. برای منطق سمت سرور (Backend) و داده (در صورت نیاز به تعاملات پایه‌ای):

- Python Flask / Django: برای ایجاد یک backend سبک با APIهای ساده برای ارتباط با frontend. Flask برای نمونه‌های کوچک‌تر و سریع‌تر مناسب‌تر است. Django برای نمونه‌های پیچیده‌تر و با ساختار منظم‌تر.
- SQLite: یک پایگاه داده سبک و مستقل که برای نمونه‌سازی بسیار مناسب است و نیازی به نصب سرور جداگانه ندارد.

انتخاب ابزار:

اگر هدف صرفاً نمایش واسط کاربری و جریان کار باشد، استفاده از Figma همراه با Bootstrap برای شبیه‌سازی ظاهر صفحات وب، کافی و بسیار کارآمد خواهد بود. اگر نیاز به نمایش تعاملات پایه‌ای سمت سرور (مانند ذخیره یک رکورد ساده) وجود داشته باشد، استفاده از Python Flask با SQLite گزینه مناسبی خواهد بود.

---

۹) ابزارهای پیشنهادی برای رسم نمودارها و پیاده‌سازی

انتخاب ابزارهای مناسب در مراحل مختلف چرخه حیات توسعه نرم‌افزار، نقش کلیدی در افزایش بهره‌وری و کیفیت کار ایفا می‌کند. در این پروژه، ابزارهای زیر برای رسم نمودارهای مهندسی نرم‌افزار و پیاده‌سازی سیستم پیشنهاد می‌شوند:

۱. ابزارهای رسم نمودارها:

- draw.io (اکنون diagrams.net):

- توضیح: یک ابزار رایگان و آنلاین (و همچنین نسخه دسکتاپ) برای رسم انواع نمودارها، از جمله ER Diagram، DFD، Class Diagram، Use Case Diagram، Sequence Diagram و Flowchart. دارای رابط کاربری ساده و قابلیت همکاری (collaboration) بالا.

- مناسب برای: تمامی نمودارهای مورد نیاز این پروژه، به ویژه برای تیم‌هایی که به دنبال راهکار رایگان و دسترسی آسان هستند.

- StarUML:

- توضیح: یک نرم‌افزار دسکتاپ قدرتمند برای مدل‌سازی UML. از طیف وسیعی از نمودارهای UML پشتیبانی می‌کند و قابلیت تولید کد اولیه (code generation) را نیز دارد.
- مناسب برای: پروژه‌هایی که نیاز به مدل‌سازی دقیق UML و قابلیت تولید کد دارند. برای Sequence Diagram، Class Diagram و Use Case Diagram بسیار کارآمد است.

- Visual Paradigm:

- توضیح: یک ابزار مدل‌سازی جامع که فراتر از UML است و شامل مدل‌سازی کسب‌وکار، معماری و پایگاه داده نیز می‌شود. دارای نسخه رایگان (Community Edition) با قابلیت‌های محدود.
- مناسب برای: پروژه‌های بزرگتر و پیچیده‌تر که نیاز به ابزارهای جامع مدل‌سازی دارند. برای ERD و مدل‌سازی معماری مناسب است.

- Figma / Adobe XD:

- توضیح: همانطور که در بخش نمونه‌سازی ذکر شد، این ابزارها برای طراحی وایرفریم‌ها و ماکت‌های بصری رابط کاربری (UI) و تجربه کاربری (UX) ایده‌آل هستند.
- مناسب برای: نمایش نحوه تعامل کاربر با سیستم و طراحی نمای بصری.

۲. ابزارهای پیاده‌سازی:

- Bootstrap (فریم‌ورک CSS):

- توضیح: یک فریم‌ورک frontend محبوب که شامل قالب‌های طراحی شده (CSS, JavaScript) برای تایپوگرافی، فرم‌ها، دکمه‌ها، ناوبری و سایر اجزای رابط کاربری است. امکان ساخت سریع وب‌سایت‌های ریسپانسیو و زیبا را فراهم می‌کند.
- مناسب برای: ساخت سریع رابط کاربری پروژه، چه در مرحله نمونه‌سازی و چه در فاز پیاده‌سازی نهایی.

- Python Flask (میکرو فریم‌ورک وب):

- توضیح: یک میکرو فریم‌ورک وب در زبان پایتون که برای ساخت برنامه‌های وب سبک و مقیاس‌پذیر استفاده می‌شود. کم‌حجم، انعطاف‌پذیر و با یادگیری آسان.

- مناسب برای: پیاده‌سازی Backend سیستم، به ویژه برای نمونه اولیه و پروژه‌هایی که نیازی به پیچیدگی زیاد فریم‌ورک‌های بزرگ ندارند.

- Python Django (فریم‌ورک وب کامل):

- توضیح: یک فریم‌ورک وب قدرتمند و کامل پایتون که با معماری MVT (Model-View-Template) همراه است. شامل ابزارهایی برای مدیریت پایگاه داده (ORM)، احراز هویت، و ساخت API می‌شود.
- مناسب برای: پیاده‌سازی نهایی سیستم، زمانی که پیچیدگی بیشتر شده و نیاز به ساختاردهی قوی‌تر و ابزارهای آماده بیشتری باشد.

توصیه ترکیبی:

برای این پروژه، توصیه می‌شود: * از draw.io برای اکثر نمودارهای UML و DFD به دلیل رایگان بودن و دسترسی آسان استفاده شود. * برای طراحی دقیق‌تر UI/UX و ماکت‌های تعاملی از Figma بهره گرفته شود. * در صورت پیاده‌سازی نهایی، استفاده از Python Flask (برای MVP) یا Django (برای نسخه کامل) همراه با Bootstrap برای frontend، رویکردی کارآمد خواهد بود.

---

۱۰) نتیجه‌گیری

این پروژه با هدف تحلیل، طراحی و ارائه چارچوبی برای یک سیستم جامع مدیریت خط تولید کارخانه انجام پذیرفت. در طول این فرآیند، اهمیت حیاتی وجود چنین سیستمی در دنیای رقابتی امروز و چالش‌های ناشی از روش‌های سنتی مدیریت تولید به خوبی روشن شد. سیستم پیشنهادی با تمرکز بر مکانیزه‌سازی و اتوماسیون فرآیندهای کلیدی از قبیل مدیریت موجودی، برنامه‌ریزی تولید، ثبت سفارشات و گزارش‌گیری، پتانسیل بالایی برای افزایش بهره‌وری، کاهش هزینه‌ها و بهبود کیفیت عملیات در کارخانه‌ها دارد.

با استفاده از تکنیک‌های مهندسی نرم‌افزار، نیازمندی‌های سیستم از دیدگاه بازیگران مختلف شناسایی و مستندسازی شد. سپس، با بهره‌گیری از نمودارهای استاندارد UML و DFD، ساختار و رفتار سیستم در سطوح مختلف تحلیل و طراحی گردید. رویکرد معماری سه‌لایه انتخاب شد تا اطمینان حاصل شود که سیستم دارای قابلیت نگهداری بالا، انعطاف‌پذیری و مقیاس‌پذیری لازم برای پاسخگویی به نیازهای آینده است. کلاس‌های اصلی سیستم با ویژگی‌ها و متدهای کلیدی تعریف شده و وابستگی‌هایشان مشخص گردید. همچنین، مدل‌سازی کامپوننت‌ها و سناریوهای کلیدی با استفاده از Sequence Diagram، دید جامعی از نحوه تعامل اجزای سیستم ارائه داد.

پیشنهادات عملیاتی برای ساخت نمونه اولیه (Prototype) با استفاده از ابزارهای مدرن و همچنین ابزارهای پیشنهادی برای رسم نمودارها و پیاده‌سازی نهایی، راه را برای گام‌های عملی بعدی هموار می‌سازد.

دستاوردهای کلیدی پروژه:

- تدوین یک مدل جامع از نیازمندی‌های یک سیستم مدیریت خط تولید.
- طراحی معماری و ساختار سیستمی با رویکرد مهندسی نرم‌افزار.
- شناسایی چالش‌های طراحی و ارائه راه‌حل‌های مناسب.
- ارائه طرحی برای نمونه‌سازی و پیاده‌سازی سیستم.

قابلیت توسعه آینده:

سیستم طراحی شده به گونه‌ای است که قابلیت توسعه و افزودن ماژول‌های جدید را دارا می‌باشد. این ماژول‌ها می‌توانند شامل مدیریت نگهداری و تعمیرات پیشگیرانه تجهیزات، ادغام با سیستم‌های ERP و حسابداری، ماژول‌های پیشرفته پیش‌بینی تقاضا، مدیریت کیفیت پیشرفته، و همچنین پیاده‌سازی رابط‌های کاربری برای دستگاه‌های IoT در خط تولید باشند.

کاربرد واقعی در کارخانه‌ها:

پیاده‌سازی چنین سیستمی می‌تواند تحول قابل توجهی در نحوه اداره کارخانه‌ها ایجاد کند. با فراهم آوردن دیدی یکپارچه و لحظه‌ای از عملیات، مدیران قادر خواهند بود تا تصمیمات آگاهانه‌تری در مورد تخصیص منابع، زمان‌بندی تولید، مدیریت ریسک و بهبود مستمر فرآیندها اتخاذ نمایند. در نهایت، این سیستم به کارخانه‌ها کمک می‌کند تا ضمن افزایش سودآوری، کیفیت محصولات خود را نیز ارتقا بخشیده و در بازار رقابتی باقی بمانند.

---

۱۱) پیوست

راهنمای رسم نمودارها در draw.io (یا diagrams.net)

در این بخش، راهنمای مختصری برای رسم هر یک از نمودارهای مورد استفاده در این پروژه در ابزار draw.io ارائه می‌شود.

نحوه شروع: 1. به وب‌سایت diagrams.net مراجعه کرده یا نسخه دسکتاپ آن را دانلود کنید. 2. یک نمودار جدید ایجاد کنید یا یک فایل موجود را باز کنید. 3. در پنل سمت چپ، انواع اشکال و الگوها را مشاهده خواهید کرد. برای نمودارهای UML، از بخش "UML" و برای نمودارهای جریان داده از بخش "Flowchart" یا "General" استفاده کنید.

۱. Use Case Diagram (نمودار مورد کاربرد): * شکل بازیگر (Actor): از آیکون "Person" یا "User" در بخش "General" استفاده کنید. * شکل مورد کاربرد (Use Case): از شکل بیضی یا "Use Case" در بخش "UML" استفاده کنید. * روابط: خطوط ساده برای نمایش association (ارتباط)، خطوط فلش‌دار برای include (تلفیقی) و extend (گسترشی).

۲. Class Diagram (نمودار کلاس): * شکل کلاس (Class): از شکل "Class" در بخش "UML" استفاده کنید. این شکل دارای سه قسمت برای نام کلاس، ویژگی‌ها (Attributes) و متدها (Operations) است. * ویژگی‌ها و متدها: به صورت "access modifier name : type" (مثلاً `- privateField : int` یا `+ publicMethod() : void`) نوشته می‌شوند. * روابط: * Association (ارتباط): خط ساده. * Aggregation (تجمّع): خط با الماس توخالی در سمت کلاس کل. * Composition (ترکیب): خط با الماس توپر در سمت کلاس کل. * Inheritance (وراثت): خط با فلش توخالی به سمت کلاس والد. * Dependency (وابستگی): خط چین با فلش به سمت کلاس وابسته.

۳. DFD (نمودار جریان داده): * موجودیت خارجی (External Entity): از شکل مستطیل در بخش "Flowchart" یا "General" استفاده کنید. * فرآیند (Process): از شکل دایره یا "Process" در بخش "Flowchart". * ذخیره داده (Data Store): از شکل دو خط موازی یا "Data Store" در بخش "Flowchart". * جریان داده (Data Flow): از فلش‌ها برای نمایش جهت جریان داده بین اجزا استفاده کنید.

۴. Sequence Diagram (نمودار توالی): * خط عمر (Lifeline): از شکل مستطیل عمودی که در بالای آن نام شرکت‌کننده (Actor یا Object) قرار دارد، استفاده کنید. * پیام (Message): از فلش‌های بین خطوط عمر استفاده کنید. فلش‌های جامد برای فراخوانی همزمان (Synchronous Call) و فلش‌های فلش‌دار برای فراخوانی ناهمزمان (Asynchronous Call). فلش‌های خط چین برگشتی برای بازگشت مقادیر. * فعالیت (Activation Bar): نوار مستطیل باریک روی خط عمر برای نشان دادن زمانی که یک شیء فعال است.

۵. ERD (نمودار رابطه-موجودیت): * موجودیت (Entity): از شکل مستطیل استفاده کنید. نام موجودیت در بالای مستطیل و ویژگی‌ها (Attributes) در داخل آن. * ویژگی کلیدی (Primary Key): معمولاً با خط کشیده زیر آن یا علامت PK مشخص می‌شود. * روابط (Relationship): از خطوطی که بین موجودیت‌ها کشیده می‌شوند، استفاده کنید. برای نمایش کاردینالیتی (Cardinality) از نمادهای Crow's Foot (مثل پای کلاغی) استفاده می‌شود که در بخش "UML" یا "Flowchart" قابل دسترسی هستند (مثلاً ۱:۱، ۱:N، N:M).

۶. Component Diagram (نمودار کامپوننت): * کامپوننت (Component): از شکل "Component" در بخش "UML" استفاده کنید (شکلی شبیه به مستطیل با دو خط عمودی در سمت چپ). * رابط (Interface): از شکل "Interface" (دایره توخالی برای ارائه، نیم‌دایره توخالی برای مصرف) در بخش "UML" استفاده کنید. * وابستگی (Dependency): از فلش‌های خط چین برای نمایش وابستگی بین کامپوننت‌ها استفاده کنید.

با استفاده از این راهنما و کاوش در کتابخانه‌های اشکال draw.io، می‌توانید تمامی نمودارهای مورد نیاز پروژه را به طور موثر رسم کنید.