فاز ۵: طراحی شیءگرای تفصیلی سیستم

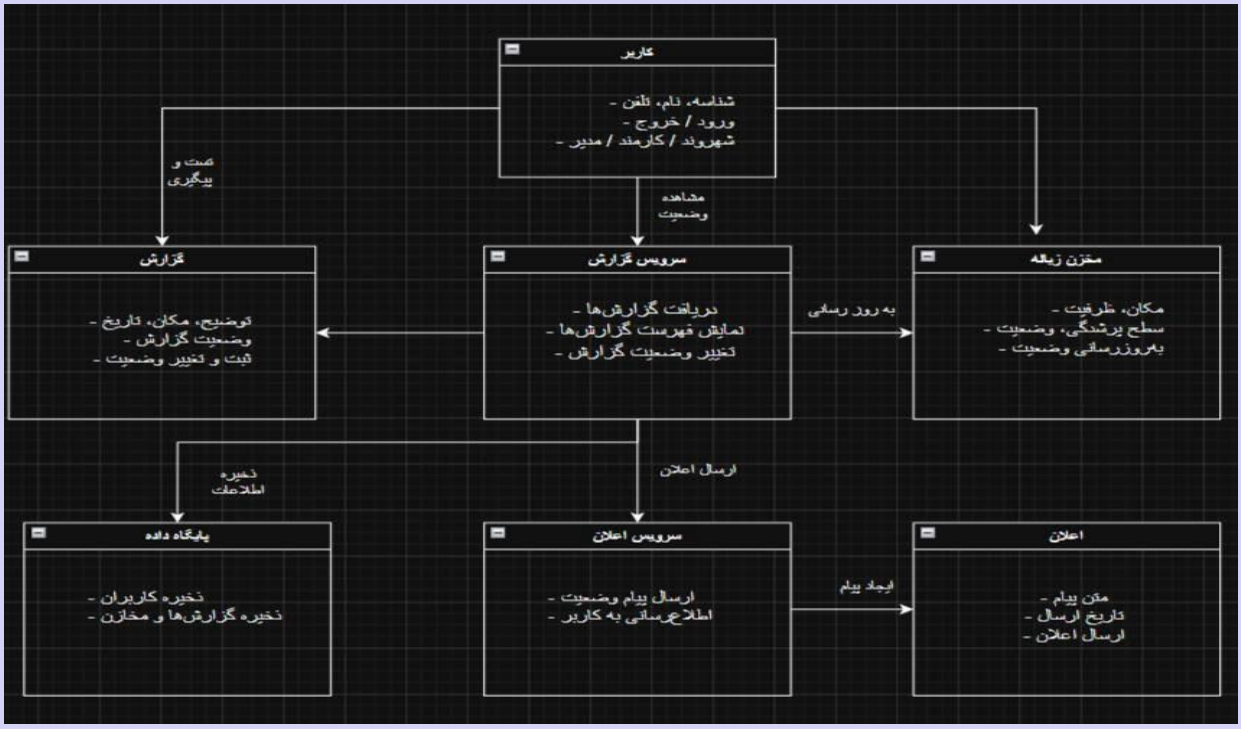
مقدمه

در این فاز، طراحی سیستم از حالت کلی و مفهومی به یک طراحی دقیق‌تر و قابل پیاده‌سازی تبدیل می‌شود. در این مرحله، کلاس‌های اصلی سیستم، ویژگی‌ها، متدها و ارتباط بین آن‌ها مشخص می‌شوند تا ساختار داخلی نرم‌افزار به‌صورت واضح تعریف شود. همچنین برای بهبود کیفیت طراحی، از برخی الگوهای طراحی و اصول شیءگرایی استفاده می‌شود. هدف این فاز آن است که سیستم به‌گونه‌ای مدل‌سازی شود که در مرحله پیاده‌سازی، برنامه‌نویس بتواند با ابهام کمتری کار را شروع کند

نمودار کلاس طراحی تفصیلی

در این بخش، نمودار کلاس تفصیلی سیستم ارائه می‌شود. این نمودار یکی از مهم‌ترین خروجی‌های فاز ۵ است، زیرا نشان می‌دهد سیستم از چه کلاس‌هایی تشکیل شده است، هر کلاس چه ویژگی‌ها و متدهایی دارد، و ارتباط بین کلاس‌ها به چه صورت است.

در این پروژه، کلاس‌های اصلی شامل کاربر، شهروند، پرسنل، مدیر، گزارش، مخزن زباله، اعلان، سرویس گزارش، سرویس اعلان و اتصال به پایگاه داده هستند. این کلاس‌ها به‌گونه‌ای طراحی شده‌اند که هر کدام مسئولیت مشخصی داشته باشند و در کنار هم، عملکرد کامل سیستم را تشکیل دهند



معرفی کلاس‌های اصلی سیستم

کلاس User

این کلاس به‌عنوان کلاس پایه برای کاربران سیستم در نظر گرفته شده است. اطلاعات عمومی هر کاربر مانند شناسه، نام، شماره تماس و نقش در این کلاس قرار می‌گیرد. همچنین عملیات کلی مانند ورود و خروج از سیستم نیز در این کلاس تعریف می‌شود.

کلاس Citizen

کلاس Citizen از کلاس User ارث‌بری می‌کند و نمایانگر شهروندی است که می‌تواند گزارش جدید ثبت کند و وضعیت گزارش‌های خود را مشاهده نماید.

کلاس Staff

کلاس Staff ارث‌بری می‌کند و مربوط به پرسنل شهرداری یا نیروهای اجرایی است. پرسنل می‌تواند گزارش‌ها را بررسی کند و وضعیت مخزن‌ها را به‌روزرسانی نماید.

کلاس Admin

کلاس Admin برای مدیر سیستم در نظر گرفته شده است. این کاربر می‌تواند کاربران را مدیریت کند و گزارش‌های مدیریتی تولید نماید.

کلاس Report

این کلاس مربوط به گزارش‌های ثبت‌شده در سیستم است. اطلاعاتی مانند شناسه گزارش، توضیحات، مکان، وضعیت و زمان ثبت در آن ذخیره می‌شود. همچنین متدهایی برای ایجاد گزارش، تغییر وضعیت و مشاهده جزئیات دارد.

کلاس WasteBin

این کلاس مربوط به مخزن زباله هوشمند است. ویژگی‌هایی مانند شناسه مخزن، موقعیت، ظرفیت، سطح پرشدگی و وضعیت در این کلاس قرار می‌گیرد. با استفاده از متدهای این کلاس می‌توان وضعیت مخزن را بررسی یا به‌روزرسانی کرد.

کلاس Notification

این کلاس برای مدیریت اعلان‌ها استفاده می‌شود. هر اعلان دارای شناسه، پیام و زمان ارسال است و متدی برای ارسال اعلان دارد.

کلاس ReportService

این کلاس یکی از کلاس‌های خدماتی سیستم است و وظیفه مدیریت عملیات مربوط به گزارش‌ها را بر عهده دارد. برای مثال ثبت گزارش، دریافت لیست گزارش‌ها و تغییر وضعیت گزارش در این کلاس انجام می‌شود.

کلاس NotificationService

این کلاس مسئول ارسال اعلان‌ها به کاربران یا پرسنل است. با تغییر وضعیت گزارش یا مخزن، این سرویس می‌تواند پیام مناسب را ارسال کند.

کلاس DatabaseConnection

این کلاس برای مدیریت اتصال به پایگاه داده طراحی شده است. به دلیل اهمیت کنترل اتصال و جلوگیری از ایجاد چند نمونه غیرضروری، این کلاس به‌صورت مناسب و متمرکز در سیستم استفاده می‌شود.

الگوهای طراحی استفاده‌شده

الگوی Singleton

در نظر گرفته شده است. دلیل استفاده از این الگو این است که DatabaseConnection برای کلاس Singleton در این پروژه، الگوی سیستم تنها به یک نمونه اصلی از اتصال پایگاه داده نیاز دارد. اگر چند اتصال غیرضروری ایجاد شود، مدیریت داده‌ها دشوارتر شده و باعث می‌شود فقط یک نمونه از این کلاس وجود داشته باشد Singleton منابع سیستم نیز بی‌دلیل مصرف می‌شوند. بنابراین استفاده از Singleton و همه بخش‌های سیستم از همان نمونه استفاده کنند.

الگوی Observer

برای بخش اعلان‌ها بسیار مناسب است. در این سیستم، هر زمان که وضعیت یک گزارش تغییر کند یا وضعیت یک مخزن به‌روزرسانی شود، لازم است کاربر یا پرسنل مربوطه از این تغییر مطلع شوند. با استفاده از این الگو، سیستم می‌تواند تغییرات را به‌صورت خودکار به بخش اعلان ارسال کند و پیام مناسب برای افراد مرتبط فرستاده شود. این موضوع باعث افزایش هماهنگی بین بخش‌های مختلف سیستم می‌شود.

بررسی طراحی بر اساس اصول SOLID

اصل Single Responsibility

مسئول نگهداری و مدیریت اطلاعات گزارش است، Report در این طراحی، هر کلاس فقط یک مسئولیت اصلی دارد. برای مثال، کلاس NotificationService فقط وظیفه ارسال اعلان را بر عهده دارد. این موضوع باعث می‌شود ساختار سیستم منظم‌تر در حالی که کلاس باشد.

اصل Open/Closed

سیستم به‌گونه‌ای طراحی شده است که بتوان در آینده قابلیت‌های جدیدی به آن اضافه کرد، بدون اینکه نیاز به تغییر اساسی در کلاس‌های اصلی باشد. برای مثال، می‌توان نوع جدیدی از اعلان یا گزارش را به سیستم اضافه کرد.

اصل Liskov Substitution

ارث‌بری می‌کنند. این موضوع نشان می‌دهد که هر یک از این کلاس‌ها می‌توانند User از کلاس Admin و Staff، Citizen کلاس‌های در جایگاه کلاس پایه قرار گیرند و از رفتارهای عمومی آن استفاده کنند.

اصل Interface Segregation

در طراحی این سیستم تلاش شده است که هر بخش فقط به قابلیت‌هایی وابسته باشد که واقعاً به آن نیاز دارد. این کار باعث می‌شود وابستگی‌های اضافی در طراحی کاهش پیدا کند و درک ساختار سیستم ساده‌تر شود.

اصل Dependency Inversion

در این طراحی، وابستگی بین بخش‌های سیستم به‌صورت منطقی و کنترل‌شده در نظر گرفته شده است تا توسعه و نگهداری سیستم در آینده آسان‌تر باشد. این اصل باعث می‌شود تغییر در یک بخش، کمترین اثر را بر سایر بخش‌ها داشته باشد

سرویس	متد	آدرس	توضیح
ثبت گزارش	POST	/api/reports	برای ثبت یک گزارش جدید توسط شهروند
دریافت گزارش‌ها	GET	/api/reports	برای مشاهده لیست گزارش‌های ثبت‌شده
تغییر وضعیت گزارش	PUT	/api/reports/{id}/status	برای تغییر وضعیت گزارش توسط پرسنل یا مدیر
دریافت اطلاعات مخزن‌ها	GET	/api/bins	برای مشاهده اطلاعات مربوط به مخزن‌ها
ارسال اعلان	POST	/api/notifications	برای ثبت یا ارسال اعلان در سیستم

جمع‌بندی

در این فاز، طراحی شیءگرای سیستم به‌صورت جزئی‌تر انجام شد و کلاس‌های اصلی، ویژگی‌ها، متدها و نوع ارتباط بین آن‌ها مشخص های اصلی API استفاده شد. در ادامه نیز SOLID گردید. همچنین برای بهبود کیفیت طراحی، از الگوهای طراحی مناسب و اصول سیستم تعریف شدند تا مسیر پیاده‌سازی نرم‌افزار روشن‌تر شود. در مجموع، خروجی این فاز پایه مناسبی برای شروع پیاده‌سازی نرم‌افزار فراهم می‌کند