طراحی تفصیلی رابطهای برنامه‌نویسی (API Design Specification)

سامانه یکپارچه آموزش آنلاین مدارس

فاز پنجم: طراحی تفصیلی سیستم (Detailed Design)

1. مقدمه

پس از انتخاب معماری ترکیبی شامل معماری کلاینت–سرور، معماری لایه‌ای، معماری خدمت‌گرا (SOA) و معماری رویدادمحور (EDA) در فاز معماری نرم‌افزار، مرحله طراحی تفصیلی با هدف تبدیل معماری مفهومی به مشخصات فنی قابل پیاده‌سازی آغاز می‌شود. یکی از مهم‌ترین فعالیت‌های این مرحله، طراحی رابطهای برنامه‌نویسی کاربردی (Application Programming Interfaces - APIs) میان سرویس‌های مستقل سامانه است.

در معماری خدمت‌گرا، APIها تنها مسیر رسمی ارتباط میان سرویس‌ها محسوب می‌شوند و کیفیت طراحی آن‌ها تأثیر مستقیمی بر امنیت، کارایی، توسعه‌پذیری، قابلیت نگهداری و قابلیت یکپارچه‌سازی کل سامانه خواهد داشت. به همین دلیل، طراحی APIها باید بر اساس استانداردهای شناخته‌شده صنعت نرم‌افزار و با رعایت اصول معماری REST انجام شود.

هدف این سند، تعریف استانداردهای طراحی رابطهای برنامه‌نویسی، تعیین قراردادهای ارتباطی میان سرویس‌ها و ایجاد بستری یکنواخت برای توسعه مستقل زیرسیستم‌های سامانه آموزش آنلاین مدارس است؛ به گونه‌ای که هر سرویس بتواند بدون وابستگی مستقیم به جزئیات پیاده‌سازی سایر سرویس‌ها با آن‌ها تعامل داشته باشد.

2. اهداف طراحی API

رابطهای برنامه‌نویسی طراحی‌شده در این سامانه با اهداف زیر تعریف می‌شوند:

- ایجاد یک قرارداد ارتباطی استاندارد میان تمامی سرویس‌های مستقل سامانه.
- کاهش وابستگی مستقیم میان زیرسیستم‌ها و تحقق اصل Loose Coupling در معماری SOA.
- فراهم نمودن امکان توسعه مستقل هر سرویس بدون ایجاد تغییر در سایر بخش‌های سیستم.
- ایجاد قابلیت استفاده مجدد از سرویس‌ها در ماژول‌ها و کلاینت‌های مختلف.
- پشتیبانی از توسعه همزمان توسط چندین تیم نرم‌افزاری.
- فراهم کردن بستر مناسب برای توسعه نسخه‌های آینده سامانه بدون ایجاد ناسازگاری با نسخه‌های قبلی.
- افزایش قابلیت یکپارچه‌سازی سامانه با سایر سامانه‌های آموزشی، سازمانی و خدمات دولت الکترونیک.

3. اصول طراحی رابطهای برنامه‌نویسی

به منظور حفظ یکنواختی و کیفیت طراحی، تمامی APIهای سامانه بر اساس مجموعه‌ای از اصول معماری تعریف می‌شوند.

3-1. اصل استقلال سرویس‌ها (Service Independence)

هر سرویس مالک دامنه مشخصی از مسئولیت‌های سیستم بوده و تنها از طریق رابطهای عمومی خود با سایر سرویس‌ها ارتباط برقرار می‌کند. هیچ سرویسی مجاز به دسترسی مستقیم به پایگاه داده یا منطق داخلی سرویس دیگر نخواهد بود.

این اصل موجب کاهش وابستگی میان زیرسیستم‌ها و افزایش قابلیت توسعه مستقل هر سرویس می‌شود.

3-2. اصل قرارداد ثابت (Contract First)

تمامی APIها قبل از پیاده‌سازی، به عنوان یک قرارداد رسمی میان سرویس‌ها تعریف می‌شوند.

هر قرارداد شامل موارد زیر خواهد بود:

- آدرس سرویس (Endpoint)
- نوع درخواست (HTTP Method)
- پارامترهای ورودی
- ساختار داده‌های ارسالی
- ساختار پاسخ
- کدهای وضعیت
- قوانین احراز هویت
- شرایط بروز خطا

این رویکرد موجب می‌شود تیم‌های مختلف بتوانند به صورت مستقل فرآیند توسعه را آغاز کنند.

3-3. اصل Stateless Communication

تمامی درخواست‌های ارسالی باید مستقل از درخواست‌های قبلی باشند.

سرور هیچ وضعیت (State) مربوط به نشست کاربران را نگهداری نمی‌کند و تمامی اطلاعات مورد نیاز هر درخواست، شامل اطلاعات احراز هویت، در همان درخواست ارسال می‌شود.

این ویژگی موجب افزایش مقیاس‌پذیری و امکان توزیع بار میان چندین نمونه از هر سرویس خواهد شد.

3-4. اصل Resource-Oriented Design

تمامی APIها بر اساس مفهوم منابع (Resources) طراحی می‌شوند.

نمونه‌ای از منابع اصلی سامانه عبارت‌اند از:

- Users
- Schools
- Classes
- Courses
- Contents
- Assignments
- Exams
- Attendance Records
- Messages
- Notifications
- Reports

هر منبع دارای شناسه یکتا بوده و عملیات مختلف از طریق متدهای استاندارد HTTP بر روی آن انجام می‌شود.

3-5. اصل نسخه‌بندی (API Versioning)

به منظور جلوگیری از ناسازگاری نسخه‌های آینده، تمامی APIها دارای شماره نسخه خواهند بود.

نمونه:

```
/api/v1/users
/api/v1/classes
/api/v1/exams
```

در صورت ایجاد تغییرات ناسازگار، نسخه جدید بدون حذف نسخه قبلی منتشر خواهد شد تا سرویس‌های موجود دچار اختلال نشوند.

4. استانداردهای ارتباطی

تمامی سرویس‌های سامانه بر اساس استانداردهای زیر با یکدیگر ارتباط برقرار می‌کنند.

استاندارد انتخاب شده	ویژگی
RESTful API	معماری API
HTTPS	پروتکل ارتباطی
JSON	قالب تبادل داده
UTF-8	نوع رمزگذاری
JWT Bearer Token	احراز هویت
REST API	نوع ارتباط داخلی
WebSocket	ارتباط بلادرنگ
Event Bus	رویدادهای ناهمزمان

استفاده از این استانداردها موجب سادگی توسعه، قابلیت همکاری میان سرویس‌ها و سازگاری با کلاینت‌های وب و موبایل خواهد شد.

5. استاندارد نام‌گذاری APIها

برای حفظ یکنواختی در کل سامانه، تمامی Endpointها از قواعد زیر پیروی می‌کنند:

- استفاده از اسامی جمع برای منابع.
- استفاده از حروف کوچک انگلیسی.
- استفاده از خط تیره (-) در صورت نیاز.
- عدم استفاده از افعال در آدرس API.
- استفاده از متدهای HTTP برای تعیین نوع عملیات.

نمونه‌های صحیح:

```
/api/v1/users
/api/v1/classes
/api/v1/assignments
/api/v1/exams
/api/v1/notifications
```

نمونه‌های نامناسب:

```
/getUsers
/createClass
/deleteExam
```

کاربرد	متد
دریافت اطلاعات	GET
ایجاد منبع جدید	POST
بروزرسانی کامل	PUT
بروزرسانی جزئی	PATCH
حذف منبع	DELETE

6. استاندارد استفاده از متدهای HTTP

تمامی عملیات سامانه مطابق استاندارد REST از متدهای HTTP استفاده می‌کنند.

این استاندارد باعث خوانایی بیشتر APIها و سازگاری با ابزارهای توسعه خواهد شد.

7. استاندارد قالب پاسخ (Response Format)

تمامی سرویس‌ها پاسخ‌های خود را با قالبی یکنواخت ارائه می‌کنند تا پردازش پاسخ‌ها در کل سامانه ساده و یکپارچه باشد.

نمونه ساختار پاسخ موفق:

```
{
  "success": true,
  "message": "Operation completed successfully.",
  "data": {
    ...
  }
}
```

نمونه پاسخ در صورت بروز خطا:

```
{
  "success": false,
  "errorCode": "AUTH_401",
  "message": "Authentication failed."
}
```

یکپارچگی قالب پاسخ موجب کاهش پیچیدگی در توسعه کلاینت‌ها و افزایش قابلیت نگهداری سامانه خواهد شد.

8. اصول امنیت APIها

امنیت یکی از مهم‌ترین الزامات طراحی رابطهای برنامه‌نویسی در سامانه آموزش آنلاین مدارس است. تمامی APIها باید صرفاً از طریق پروتکل HTTPS در دسترس باشند تا محرمانگی و یکپارچگی داده‌های در حال انتقال تضمین شود.

فرآیند احراز هویت کاربران بر پایه توکن‌های JWT انجام می‌شود و هر درخواست پس از اعتبارسنجی توکن، بر اساس مدل کنترل دسترسی مبتنی بر نقش (RBAC) بررسی خواهد شد. به این ترتیب، هر کاربر تنها به منابع و عملیاتی دسترسی خواهد داشت که متناسب با نقش سازمانی وی تعریف شده است.

علاوه بر این، تمامی ورودی‌ها پیش از پردازش اعتبارسنجی شده و ثبت رویدادهای امنیتی، محدودسازی نرخ درخواست‌ها (Rate Limiting) و مدیریت نشست‌های منقضی‌شده نیز به عنوان بخشی از سیاست‌های امنیتی سامانه در نظر گرفته می‌شوند.

9. جمع‌بندی

اصول و استانداردهای ارائه‌شده در این سند، چارچوب طراحی تمامی رابطهای برنامه‌نویسی سامانه آموزش آنلاین مدارس را تشکیل می‌دهند. رعایت این اصول موجب می‌شود سرویس‌های مستقل سامانه با حداقل وابستگی، بیشترین قابلیت توسعه، امنیت، مقیاس‌پذیری و قابلیت نگهداری طراحی شوند. در ادامه طراحی تفصیلی، رابطهای برنامه‌نویسی هر یک از سرویس‌های کلیدی بر اساس همین استانداردها تعریف و مستندسازی خواهند شد تا امکان پیاده‌سازی مستقیم معماری طراحی‌شده فراهم شود.

10. تعریف رابطهای برنامه‌نویسی سرویس احراز هویت (Authentication Service)

10-1. معرفی سرویس

سرویس احراز هویت مسئول مدیریت ورود کاربران، اعتبارسنجی اطلاعات هویتی، تولید و اعتبارسنجی توکن‌های امنیتی و کنترل نشست‌های کاربران است. تمامی درخواست‌های سرویس‌های دیگر پیش از پردازش، از طریق این سرویس اعتبارسنجی می‌شوند.

این سرویس نقطه ورود اصلی تمامی کاربران سامانه محسوب شده و امنیت کل سامانه به عملکرد صحیح آن وابسته است.

10-2. مسئولیت‌های سرویس

این سرویس وظایف زیر را بر عهده دارد:

- احراز هویت کاربران
- صدور JWT Token
- اعتبارسنجی Token
- Refresh Token
- خروج کاربران (Logout)
- بازیابی رمز عبور
- تغییر رمز عبور
- مدیریت نشست‌های فعال

10-3. Endpointهای سرویس

Endpoint	Method	توضیح
/api/v1/auth/login	POST	ورود کاربر
/api/v1/auth/logout	POST	خروج کاربر
/api/v1/auth/refresh	POST	تمدید توکن
/api/v1/auth/forgot-password	POST	درخواست بازیابی رمز
/api/v1/auth/reset-password	POST	تغییر رمز عبور
/api/v1/auth/validate	GET	اعتبارسنجی توکن

10-4. API ورود کاربران

Endpoint

```
POST /api/v1/auth/login
```

هدف

اعتبارسنجی اطلاعات ورود کاربر و تولید توکن امنیتی.

درخواست (Request)

```
{
  "nationalCode":"1234567890",
  "password":"********"
}
```

پاسخ موفق

```
{
  "success":true,
  "message":"Login successful.",
  "data":{
      "accessToken":"JWT_TOKEN",
      "refreshToken":"REFRESH_TOKEN",
      "expiresIn":3600,
      "role":"Teacher"
  }
}
```

پاسخ ناموفق

```
{
 "success":false,
 "errorCode":"AUTH_401",
 "message":"Invalid username or password."
}
```

مجوز دسترسی

عمومی (نیازی به Token ندارد.)

کدهای وضعیت

Status	توضیح
200	ورود موفق
400	اطلاعات ناقص
401	نام کاربری یا رمز اشتباه
423	حساب کاربری قفل شده
500	خطای داخلی سرور

10-5. تمدید Token

Endpoint

```
POST /api/v1/auth/refresh
```

Request

```
{
   "refreshToken":"REFRESH_TOKEN"
}
```

Response

```
{
   "success":true,
   "data":{
      "accessToken":"NEW_TOKEN",
      "expiresIn":3600
   }
}
```

10-6. خروج از سامانه

Endpoint

```
POST /api/v1/auth/logout
```

Header

```
Authorization: Bearer JWT_TOKEN
```

Response

```
{
 "success":true,
 "message":"Logout completed."
}
```

10-7. اعتبارسنجی Token

```
GET /api/v1/auth/validate
```

Header

```
Authorization: Bearer JWT_TOKEN
```

Response

```
{
 "success":true,
 "data":{
    "userId":205,
    "role":"Student",
    "isValid":true
 }
}
```

10-8. ارتباط با سایر سرویس‌ها

Authentication Service با سرویس‌های زیر تعامل مستقیم دارد:

نوع تعامل	سرویس
دریافت اطلاعات کاربر	User Service
ارسال هشدار امنیتی	Notification Service
ثبت لاگ ورود و خروج	Audit Service
ثبت آمار کاربران	Reporting Service

10-9. الزامات امنیتی

- استفاده اجباری از HTTPS
- JWT با زمان انقضای محدود
- Refresh Token مستقل
- رمزنگاری گذرواژه با BCrypt
- Rate Limiting برای جلوگیری از Brute Force
- ثبت تمام تلاش‌های ناموفق ورود
- قفل موقت حساب پس از چند ورود ناموفق
- اعتبارسنجی تمام ورودی‌ها

11. تعریف رابطهای برنامه‌نویسی سرویس مدیریت کاربران (User Management Service)

11-1. معرفی سرویس

سرویس مدیریت کاربران مسئول ایجاد، نگهداری و مدیریت اطلاعات تمامی کاربران سامانه شامل دانش‌آموزان، معلمان، والدین، مدیران مدارس و مدیر سامانه است. این سرویس هسته اصلی مدیریت هویت کاربران بوده و اطلاعات پایه‌ای موردنیاز سایر سرویس‌ها را فراهم می‌کند.

11-2. مسئولیت‌های سرویس

- ثبت کاربر جدید

- ویرایش اطلاعات کاربر
- حذف کاربر
- مشاهده اطلاعات کاربران
- مدیریت نقش‌ها (Role Management)
- مدیریت وضعیت حساب کاربری
- جستجوی کاربران
- تخصیص کاربران به مدارس و کلاس‌ها

11-3. Endpointهای سرویس

Endpoint	Method	توضیح
/api/v1/users	GET	دریافت لیست کاربران
/api/v1/users/{id}	GET	دریافت اطلاعات یک کاربر
/api/v1/users	POST	ایجاد کاربر
/api/v1/users/{id}	PUT	ویرایش کامل اطلاعات
/api/v1/users/{id}	PATCH	ویرایش بخشی از اطلاعات
/api/v1/users/{id}	DELETE	حذف کاربر
/api/v1/users/search	GET	جستجوی کاربران

11-4. API ایجاد کاربر

Endpoint

```
POST /api/v1/users
```

Request

```
{
  "firstName":"Ali",
  "lastName":"Ahmadi",
  "nationalCode":"1234567890",
  "mobile":"09123456789",
  "email":"ali@example.com",
  "role":"Student",
  "schoolId":12
}
```

Response

```
{
 "success":true,
 "message":"User created successfully.",
 "data":{
    "userId":325
 }
}
```

مجوز دسترسی

فقط مدیر سامانه و مدیر مدرسه

11-5. API دریافت اطلاعات کاربر

```
GET /api/v1/users/{id}
```

Response

```
{
 "success":true,
 "data":{
    "userId":325,
    "firstName":"Ali",
    "lastName":"Ahmadi",
    "role":"Student",
    "school":"Farhang High School"
 }
}
```

11-6. ارتباط با سایر سرویس‌ها

نوع ارتباط	سرویس
احراز هویت کاربران	Authentication Service
اختصاص کاربران به کلاس‌ها	Class Service
ثبت حضور	Attendance Service
شرکت در آزمون	Exam Service
ارسال و دریافت پیام	Messaging Service
تولید گزارش‌های آماری	Reporting Service

12. تعریف رابطهای برنامه‌نویسی سرویس مدیریت کلاس‌ها (Class Management Service)

12-1. معرفی سرویس

سرویس مدیریت کلاس‌ها یکی از مهم‌ترین سرویس‌های دامنه کسب‌وکار در سامانه آموزش آنلاین مدارس محسوب می‌شود و مسئول مدیریت کامل چرخه عمر کلاس‌های آموزشی است. این سرویس وظیفه ایجاد، سازمان‌دهی، نگهداری و مدیریت اطلاعات کلاس‌ها، تخصیص معلمان، ثبت دانش‌آموزان، کنترل ظرفیت، مدیریت وضعیت کلاس و ارائه اطلاعات مورد نیاز سایر سرویس‌های سامانه را بر عهده دارد.

در معماری خدمت‌گرای انتخاب‌شده برای این پروژه، سرویس مدیریت کلاس‌ها مالک تمامی اطلاعات مربوط به کلاس‌های آموزشی بوده و سایر سرویس‌ها از جمله سرویس مدیریت محتوا، تکالیف، آزمون‌ها، حضور و غیاب، پیام‌رسانی و گزارش‌گیری صرفاً از طریق رابطهای برنامه‌نویسی استاندارد با آن تعامل خواهند داشت. این رویکرد ضمن کاهش وابستگی میان زیرسیستم‌ها، توسعه مستقل هر سرویس، مقیاس‌پذیری و نگهداری سیستم را تسهیل می‌کند.

12-2. مسئولیت‌های سرویس

مهم‌ترین مسئولیت‌های این سرویس عبارت‌اند از:

- ایجاد کلاس‌های آموزشی جدید
- ویرایش اطلاعات کلاس
- حذف کلاس‌های غیرفعال
- مدیریت ظرفیت کلاس
- تخصیص معلمان به کلاس‌ها
- ثبت و حذف دانش‌آموزان
- مدیریت سال تحصیلی و نیمسال
- مدیریت وضعیت فعال یا غیرفعال کلاس
- جستجو و فیلتر کلاس‌ها
- ارائه اطلاعات کلاس به سایر سرویس‌های سامانه

12-3. رابط‌های برنامه‌نویسی سرویس

Endpoint	Method	شرح عملیات	سطح دسترسی
/api/v1/classes	GET	دریافت لیست کلاس‌ها	مدیر، معلم
/api/v1/classes/{id}	GET	دریافت اطلاعات کلاس	کاربران مجاز
/api/v1/classes	POST	ایجاد کلاس جدید	مدیر مدرسه
/api/v1/classes/{id}	PUT	ویرایش کامل کلاس	مدیر مدرسه
/api/v1/classes/{id}	PATCH	ویرایش بخشی از اطلاعات	مدیر مدرسه
/api/v1/classes/{id}	DELETE	حذف کلاس	مدیر مدرسه
/api/v1/classes/{id}/students	POST	ثبت دانش‌آموز در کلاس	مدیر، معلم
/api/v1/classes/{id}/students/{studentId}	DELETE	حذف دانش‌آموز از کلاس	مدیر، معلم
/api/v1/classes/{id}/teacher	PUT	تخصیص یا تغییر معلم	مدیر مدرسه
/api/v1/classes/search	GET	جستجوی کلاس‌ها	مدیر، معلم

12-4. API ایجاد کلاس

Endpoint

POST /api/v1/classes

هدف

این رابط برنامه‌نویسی برای ایجاد یک کلاس آموزشی جدید در سامانه مورد استفاده قرار می‌گیرد. اطلاعات پایه کلاس شامل عنوان، پایه تحصیلی، معلم، مدرسه، ظرفیت و سال تحصیلی از طریق این سرویس ثبت شده و شناسه یکتای کلاس ایجاد می‌شود.

نمونه درخواست (Request)

```
{
  "title":"ریاضی دهم",
  "grade":"دهم",
  "schoolId":15,
  "teacherId":48,
  "capacity":35,
  "semester":"1405-1406",
  "academicYear":"1405"
}
```

نمونه پاسخ (Response)

```
{
```

```
  "success":true,
  "message":"Class created successfully.",
  "data":{
      "classId":112
  }
}
```

کدهای وضعیت

Status	توضیح
201	کلاس با موفقیت ایجاد شد
400	اطلاعات ورودی نامعتبر
401	کاربر احراز هویت نشده است
403	سطح دسترسی کافی وجود ندارد
409	کلاس مشابه قبلاً ایجاد شده است
500	خطای داخلی سرور

12-5. API ثبت دانش‌آموز در کلاس

Endpoint

POST /api/v1/classes/{id}/students

هدف

ثبت یک دانش‌آموز در کلاس مشخص‌شده پس از کنترل ظرفیت، اعتبار اطلاعات و جلوگیری از ثبت تکراری.

نمونه درخواست

```
{
   "studentId":325
}
```

نمونه پاسخ

```
{
   "success":true,
   "message":"Student enrolled successfully."
}
```

12-6. API دریافت اطلاعات کلاس

Endpoint

GET /api/v1/classes/{id}

هدف

دریافت اطلاعات کامل یک کلاس آموزشی جهت نمایش در پنل مدیر، معلم یا سایر سرویس‌های سامانه.

نمونه پاسخ


```
{
   "success":true,
   "data":{
      "classId":112,
      "title":"ریاضی دهم",
      "teacher":"محمد رضایی",
      "students":32,
      "capacity":35,
      "semester":"1405-1406",
      "status":"Active"
   }
}
```


12-7. ارتباط با سایر سرویس‌ها

سرویس مدیریت کلاس‌ها با سایر زیرسیستم‌های سامانه تعامل مستقیم دارد. اطلاعات کاربران از سرویس مدیریت کاربران دریافت می‌شود، محتوای آموزشی توسط سرویس مدیریت محتوا به کلاس‌ها اختصاص می‌یابد، سرویس تکالیف و آزمون‌ها از اطلاعات کلاس برای ایجاد فعالیت‌های آموزشی استفاده می‌کنند و سرویس حضور و غیاب وضعیت حضور دانش‌آموزان را بر اساس اطلاعات این سرویس ثبت می‌کند. علاوه بر این، سرویس پیام‌رسانی و اعلان‌ها برای اطلاع‌رسانی به اعضای کلاس و سرویس گزارش‌گیری برای استخراج شاخص‌های آموزشی از داده‌های این سرویس استفاده می‌کنند.

12-8. قواعد تجاری (Business Rules)

به منظور حفظ یکپارچگی اطلاعات، مجموعه‌ای از قواعد تجاری در این سرویس اعمال می‌شود. هر کلاس باید دقیقاً به یک مدرسه تعلق داشته باشد و تنها یک معلم به عنوان مسئول اصلی کلاس تعریف شود. تعداد دانش‌آموزان ثبت‌شده نباید از ظرفیت تعیین‌شده تجاوز کند و هر دانش‌آموز تنها یک بار مجاز به ثبت‌نام در یک کلاس مشخص است. حذف کلاس صرفاً زمانی امکان‌پذیر خواهد بود که هیچ آزمون، تکلیف، جلسه آنلاین یا محتوای فعال به آن وابسته نباشد. همچنین تمامی کلاس‌ها باید دارای سال تحصیلی، نیمسال آموزشی و وضعیت مشخص (فعال، غیرفعال یا پایان‌یافته) باشند.

12-9. ملاحظات طراحی

طراحی این سرویس بر اساس اصل Single Responsibility Principle (SRP) انجام شده است؛ بنابراین کلیه مسئولیت‌های مرتبط با مدیریت کلاس‌ها در این سرویس متمرکز بوده و منطق مربوط به آزمون‌ها، تکالیف، محتوای آموزشی یا حضور و غیاب در سرویس‌های مستقل پیاده‌سازی می‌شود. همچنین مطابق اصل Open/Closed Principle (OCP) ساختار سرویس به گونه‌ای طراحی شده است که قابلیت‌هایی مانند کلاس‌های مشترک، کلاس‌های هوشمند، برنامه‌ریزی خودکار یا مدیریت چندمدرسه‌ای بدون تغییر در هسته اصلی سرویس قابل توسعه باشند. بر اساس اصل Dependency Inversion Principle (DIP) نیز سایر سرویس‌ها صرفاً از طریق APIهای عمومی با این سرویس تعامل داشته و هیچ وابستگی مستقیمی به پایگاه داده یا ساختار داخلی آن ندارند.

12-10. الگوهای طراحی پیشنهادی

در پیاده‌سازی این سرویس استفاده از الگوهای Repository Pattern برای مدیریت دسترسی به داده‌ها، Factory Method برای ایجاد انواع کلاس‌های آموزشی، DTO (Data Transfer Object) برای انتقال اطلاعات میان سرویس‌ها، Facade Pattern برای ساده‌سازی ارتباط با سایر سرویس‌ها و Strategy Pattern برای مدیریت سیاست‌های مختلف ثبت‌نام و تخصیص کاربران پیشنهاد می‌شود.

12-11. جمع‌بندی

سرویس مدیریت کلاس‌ها یکی از سرویس‌های محوری سامانه آموزش آنلاین مدارس محسوب می‌شود و بخش عمده‌ای از فرآیندهای آموزشی بر پایه اطلاعات تولیدشده توسط آن انجام می‌گیرد. طراحی این سرویس مطابق معماری RESTful، معماری خدمت‌گرا (SOA)، اصول SOLID و الگوهای طراحی GoF انجام شده است تا علاوه بر تأمین نیازمندی‌های فعلی، امکان توسعه، نگهداری و مقیاس‌پذیری سامانه در نسخه‌های آینده نیز فراهم شود.

13. تعریف رابطهای برنامه‌نویسی سرویس مدیریت محتوای آموزشی (Content Management Service)

13-1. معرفی سرویس

سرویس مدیریت محتوای آموزشی مسئول مدیریت چرخه کامل منابع و محتوای آموزشی در سامانه آموزش آنلاین مدارس است. این سرویس امکان ایجاد، بارگذاری، سازمان‌دهی، دسته‌بندی، بروزرسانی، حذف و اشتراک‌گذاری انواع محتوای آموزشی را فراهم می‌کند و به عنوان مخزن مرکزی منابع یادگیری مورد استفاده معلمان و دانش‌آموزان عمل می‌کند.

در معماری خدمت‌گرای سامانه، این سرویس مالک تمامی اطلاعات مربوط به محتوای آموزشی بوده و سایر سرویس‌ها از طریق رابطهای برنامه‌نویسی استاندارد به اطلاعات آن دسترسی پیدا می‌کنند. استقلال این سرویس موجب می‌شود مدیریت فایل‌ها، نسخه‌بندی محتوا و توسعه قابلیت‌های جدید بدون ایجاد وابستگی مستقیم به سایر زیرسیستم‌ها انجام شود.

13-2. مسئولیت‌های سرویس

این سرویس مسئول انجام وظایف زیر است:

- بارگذاری انواع محتوای آموزشی
- ویرایش اطلاعات محتوا
- حذف محتوای آموزشی
- مدیریت نسخه‌های مختلف محتوا
- دسته‌بندی و برچسب‌گذاری منابع آموزشی
- اختصاص محتوا به کلاس‌های آموزشی
- کنترل سطح دسترسی کاربران به محتوا
- مدیریت فضای ذخیره‌سازی

- جستجو و بازیابی محتوا
- ارائه اطلاعات محتوا به سایر سرویس‌های سامانه

13-3. رابطهای برنامه‌نویسی سرویس

Endpoint	Method	شرح عملیات	سطح دسترسی
/api/v1/contents	GET	دریافت فهرست محتوا	کاربران مجاز
/api/v1/contents/{id}	GET	دریافت اطلاعات یک محتوا	کاربران مجاز
/api/v1/contents	POST	بارگذاری محتوای جدید	معلم، مدیر
/api/v1/contents/{id}	PUT	ویرایش کامل محتوا	مالک محتوا
/api/v1/contents/{id}	PATCH	ویرایش بخشی از اطلاعات	مالک محتوا
/api/v1/contents/{id}	DELETE	حذف محتوا	مالک محتوا، مدیر
/api/v1/contents/search	GET	جستجوی محتوا	کاربران مجاز
/api/v1/contents/{id}/download	GET	دانلود محتوا	کاربران مجاز
/api/v1/contents/{id}/preview	GET	مشاهده پیش‌نمایش محتوا	کاربران مجاز

13-4. API بارگذاری محتوای آموزشی

Endpoint

POST /api/v1/contents

هدف

ایجاد و ثبت یک محتوای آموزشی جدید شامل فایل، ویدئو، صوت، تصویر، جزوه یا سایر منابع یادگیری و تخصیص آن به کلاس آموزشی مربوطه.

نمونه درخواست (Request)

```
{
  "title":"فصل اول ریاضی",
  "description":"آموزش مبحث مجموعه ها",
  "contentType":"Video",
  "classId":112,
  "fileName":"chapter1.mp4",
  "fileSize":"320MB",
  "tags":["فصل اول","پایه دهم","ریاضی"]
}
```

نمونه پاسخ (Response)

```
{
  "success":true,
  "message":"Content uploaded successfully.",
  "data":{
      "contentId":870,
      "status":"Published"
  }
}
```

کدهای وضعیت

Status	توضیح
201	محتوا با موفقیت ایجاد شد
400	اطلاعات ورودی نامعتبر
401	کاربر احراز هویت نشده است
403	دسترسی غیرمجاز
413	حجم فایل بیش از حد مجاز است
415	نوع فایل پشتیبانی نمی‌شود
500	خطای داخلی سرور

13-5. API دریافت اطلاعات محتوا

Endpoint

GET /api/v1/contents/{id}

هدف

دریافت اطلاعات کامل یک محتوای آموزشی جهت نمایش در پنل کاربران یا استفاده توسط سایر سرویس‌های سامانه.

نمونه پاسخ

```
{
  "success":true,
  "data":{
      "contentId":870,
      "title":"فصل اول ریاضی",
      "contentType":"Video",
      "class":"ریاضی دهم",
      "uploadedBy":"محمد رضایی",
      "uploadDate":"2026-09-15",
      "fileSize":"320MB",
```

```
      "version":"1.2",
      "status":"Published"
  }
}
```

13-6. ارتباط با سایر سرویس‌ها

سرویس مدیریت محتوای آموزشی با سایر سرویس‌های سامانه ارتباط مستقیم دارد. اطلاعات کلاس‌ها از طریق سرویس مدیریت کلاس دریافت می‌شود و هر محتوای آموزشی به یک یا چند کلاس اختصاص می‌یابد. سرویس تکالیف می‌تواند برای هر تکلیف منابع آموزشی مرتبط را فراخوانی کند و سرویس آزمون نیز از محتوای آموزشی به عنوان منبع طراحی سؤالات استفاده می‌کند. همچنین سرویس اعلان‌ها کاربران را از انتشار محتوای جدید مطلع کرده و سرویس گزارش‌گیری اطلاعات مربوط به میزان استفاده، دانلود و مشاهده منابع آموزشی را تحلیل می‌کند.

13-7. قواعد تجاری (Business Rules)

به منظور حفظ یکپارچگی داده‌ها و مدیریت صحیح منابع آموزشی، قواعد زیر در این سرویس اعمال می‌شود:

- هر محتوای آموزشی باید حداقل به یک کلاس آموزشی اختصاص یابد.
- تنها معلم مربوط به کلاس یا مدیر مدرسه مجاز به انتشار محتوا است.
- هر فایل دارای شناسه یکتا و شماره نسخه مشخص خواهد بود.
- حذف محتوا نباید موجب حذف سوابق آموزشی دانش‌آموزان شود.
- تمامی فایل‌ها پیش از ذخیره‌سازی از نظر نوع، حجم و امنیت اعتبارسنجی می‌شوند.
- کاربران تنها مجاز به مشاهده محتوایی هستند که به کلاس‌های آن‌ها اختصاص یافته باشد.
- نسخه‌های قبلی محتوا برای اهداف بازیابی و ممیزی در سامانه نگهداری می‌شوند.

13-8. ملاحظات طراحی

این سرویس مطابق اصل Single Responsibility Principle (SRP) صرفاً مسئول مدیریت منابع آموزشی بوده و هیچ منطق مرتبط با آزمون، تکالیف یا مدیریت کلاس‌ها در آن پیاده‌سازی نمی‌شود. همچنین با رعایت اصل Open/Closed Principle (OCP) امکان افزودن انواع جدید محتوا مانند محتوای تعاملی، واقعیت افزوده (AR)، واقعیت مجازی (VR) یا محتوای مبتنی بر هوش مصنوعی بدون تغییر در ساختار اصلی سرویس فراهم شده است.

بر اساس اصل Dependency Inversion Principle (DIP) نیز محل ذخیره‌سازی فایل‌ها از طریق یک لایه انتزاعی مدیریت می‌شود؛ بنابراین امکان استفاده از فضای ذخیره‌سازی محلی، فضای ابری یا سامانه‌های ذخیره‌سازی توزیع‌شده بدون تغییر در منطق اصلی سرویس وجود دارد.

13-9. الگوهای طراحی پیشنهادی

در پیاده‌سازی این سرویس استفاده از الگوهای Repository Pattern برای مدیریت دسترسی به داده‌ها، Factory Method برای ایجاد انواع مختلف محتوای آموزشی، Strategy Pattern برای مدیریت روش‌های مختلف ذخیره‌سازی فایل‌ها، Adapter Pattern برای اتصال به سرویس‌های ذخیره‌سازی مختلف، DTO (Data Transfer Object) برای تبادل اطلاعات بین سرویس‌ها و Facade Pattern برای ساده‌سازی ارتباط با سرویس‌های وابسته پیشنهاد می‌شود.

13-10. جمع‌بندی

سرویس مدیریت محتوای آموزشی یکی از مهم‌ترین زیرسیستم‌های سامانه آموزش آنلاین مدارس محسوب می‌شود و نقش اساسی در مدیریت منابع یادگیری، پشتیبانی از فرآیند آموزش و تعامل میان معلمان و دانش‌آموزان ایفا می‌کند. طراحی این سرویس بر پایه معماری RESTful، معماری خدمت‌گرا (SOA)، اصول SOLID و الگوهای طراحی GoF انجام شده است تا ضمن حفظ استقلال سرویس، امنیت، قابلیت توسعه، مقیاس‌پذیری و نگهداری آسان در نسخه‌های آینده سامانه تضمین شود.

14. تعریف رابطهای برنامه‌نویسی سرویس مدیریت تکالیف (Assignment Service)

14-1. معرفی سرویس

سرویس مدیریت تکالیف مسئول مدیریت کامل چرخه حیات تکالیف آموزشی در سامانه آموزش آنلاین مدارس است. این سرویس امکان تعریف، انتشار، دریافت، ارزیابی و بایگانی تکالیف را فراهم کرده و ارتباط میان معلمان، دانش‌آموزان و سایر سرویس‌های آموزشی را مدیریت می‌کند. هدف اصلی این سرویس، مکانیزه‌سازی فرآیند تکلیف از زمان ایجاد تا ثبت نمره و ارائه بازخورد است.

در معماری خدمت‌گرای سامانه، سرویس مدیریت تکالیف به صورت مستقل عمل می‌کند و اطلاعات مربوط به کلاس‌ها، کاربران و منابع آموزشی را از طریق APIهای استاندارد سایر سرویس‌ها دریافت می‌کند. این استقلال موجب افزایش قابلیت توسعه، نگهداری و مقیاس‌پذیری سامانه می‌شود.

14-2. مسئولیت‌های سرویس

وظایف اصلی این سرویس عبارت‌اند از:

- ایجاد تکلیف آموزشی
- ویرایش اطلاعات تکلیف
- حذف تکلیف
- انتشار تکلیف برای کلاس
- تعیین مهلت ارسال
- دریافت پاسخ دانش‌آموزان
- مدیریت نسخه‌های ارسالی
- تصحیح و ارزیابی تکالیف
- ثبت نمره و بازخورد
- مدیریت وضعیت تکلیف
- ارائه گزارش عملکرد تکالیف

14-3. رابطهای برنامه‌نویسی سرویس

Endpoint	Method	شرح عملیات	سطح دسترسی
/api/v1/assignments	GET	دریافت فهرست تکالیف	کاربران مجاز
/api/v1/assignments/{id}	GET	مشاهده اطلاعات تکلیف	کاربران مجاز
/api/v1/assignments	POST	ایجاد تکلیف جدید	معلم
/api/v1/assignments/{id}	PUT	ویرایش تکلیف	معلم
/api/v1/assignments/{id}	DELETE	حذف تکلیف	معلم
/api/v1/assignments/{id}/submit	POST	ارسال پاسخ دانش‌آموز	دانش‌آموز
/api/v1/assignments/{id}/grade	POST	ثبت نمره و بازخورد	معلم
/api/v1/assignments/search	GET	جستجوی تکالیف	کاربران مجاز

14-4. API ایجاد تکلیف

Endpoint

POST /api/v1/assignments

هدف

ایجاد یک تکلیف آموزشی جدید و تخصیص آن به کلاس مشخص همراه با تعیین زمان شروع، مهلت ارسال، نحوه ارزیابی و منابع آموزشی مرتبط.

نمونه درخواست (Request)

```
{
  "title":"حل تمرین فصل اول",
  "description":"تمرین‌های صفحات 20 تا 30 کتاب را حل کنید.",
  "classId":112,
  "contentId":870,
  "dueDate":"2026-10-05T23:59:00",
  "maxScore":20
}
```

نمونه پاسخ (Response)

```
{
  "success":true,
  "message":"Assignment created successfully.",
  "data":{
      "assignmentId":540
```

```
  }
}
```

Status	توضیح
201	تکلیف ایجاد شد
400	اطلاعات نامعتبر
401	عدم احراز هویت
403	عدم مجوز ایجاد تکلیف
404	کلاس یا محتوا یافت نشد
500	خطای داخلی سرور

کدهای وضعیت

14-5. API ارسال پاسخ دانش‌آموز

Endpoint

POST /api/v1/assignments/{id}/submit

هدف

ثبت پاسخ دانش‌آموز برای تکلیف تعیین‌شده و ذخیره آن جهت ارزیابی معلم.

نمونه درخواست

```
{
  "studentId":325,
  "fileName":"assignment1.pdf",
  "comment":"پاسخ تمرین‌ها"
}
```

نمونه پاسخ

```
{
  "success":true,
  "message":"Assignment submitted successfully.",
  "data":{
      "submissionId":1280,
      "submitTime":"2026-10-03T18:25:40"
  }
}
```

14-6. API ثبت نمره و بازخورد

Endpoint

POST /api/v1/assignments/{id}/grade

هدف

ثبت نتیجه ارزیابی تکلیف، نمره نهایی و بازخورد آموزشی برای دانش‌آموز.

نمونه درخواست

```
{
  "studentId":325,
  "score":18.5,
  "feedback":"پاسخ‌ها صحیح است، در سؤال سوم دقت بیشتری داشته باشید."
}
```

نمونه پاسخ

```
{
  "success":true,
  "message":"Grade registered successfully."
}
```

14-7. ارتباط با سایر سرویس‌ها

سرویس مدیریت تکالیف برای انجام وظایف خود با چندین سرویس دیگر در تعامل است. اطلاعات کاربران و نقش‌ها از سرویس مدیریت کاربران دریافت می‌شود، اطلاعات کلاس از سرویس مدیریت کلاس‌ها استخراج می‌شود و منابع آموزشی مرتبط از سرویس مدیریت محتوا فراخوانی می‌شوند. همچنین پس از ایجاد یا پایان مهلت تکلیف، سرویس اعلان‌ها پیام‌های لازم را برای کاربران ارسال می‌کند و سرویس گزارش‌گیری نیز داده‌های مربوط به نرخ ارسال، میزان تأخیر، میانگین نمرات و عملکرد دانش‌آموزان را تحلیل می‌کند.

14-8. قواعد تجاری (Business Rules)

به منظور حفظ صحت فرآیند آموزشی، قواعد زیر در این سرویس اعمال می‌شود:

- هر تکلیف باید به یک کلاس آموزشی مشخص وابسته باشد.
- تنها معلم مسئول کلاس مجاز به ایجاد یا ویرایش تکلیف است.
- هر دانش‌آموز تنها یک پاسخ نهایی برای هر تکلیف ثبت می‌کند، مگر آنکه امکان ارسال مجدد توسط معلم فعال شده باشد.
- پس از پایان مهلت ارسال، ثبت پاسخ جدید تنها با مجوز معلم امکان‌پذیر است.
- نمره ثبت‌شده نباید از حداکثر نمره تعریف‌شده برای تکلیف بیشتر باشد.

- تمامی تغییرات مربوط به نمرات و بازخوردها در لاگ‌های سامانه ثبت می‌شوند.

14-9. ملاحظات طراحی

این سرویس بر اساس اصل **Single Responsibility Principle (SRP)** تنها مسئول مدیریت فرآیند تکالیف است و عملیات مربوط به کلاس‌ها، کاربران یا منابع آموزشی در سرویس‌های مستقل انجام می‌شود. همچنین مطابق اصل **Open/Closed Principle (OCP)** ساختار سرویس به گونه‌ای طراحی شده است که قابلیت‌هایی مانند تصحیح خودکار، ارزیابی مبتنی بر هوش مصنوعی، تشخیص مشابهت پاسخ‌ها یا اتصال به سامانه‌های ضدتقلب بدون تغییر در هسته اصلی سرویس قابل اضافه شدن باشد.

بر اساس اصل **Dependency Inversion Principle (DIP)** نیز ارتباط با سرویس‌های کلاس، محتوا، اعلان‌ها و گزارش‌گیری تنها از طریق رابطهای استاندارد API انجام می‌شود و هیچ وابستگی مستقیمی به ساختار داخلی آن‌ها وجود ندارد.

14-10. الگوهای طراحی پیشنهادی

برای پیاده‌سازی این سرویس استفاده از الگوهای **Repository Pattern** جهت مدیریت داده‌های تکالیف، **Strategy Pattern** برای پشتیبانی از روش‌های مختلف ارزیابی، **Factory Method** برای ایجاد انواع تکالیف، **Observer Pattern** برای ارسال اعلان‌های خودکار پس از ایجاد یا پایان مهلت تکلیف، **DTO (Data Transfer Object)** برای تبادل داده میان سرویس‌ها و **Facade Pattern** برای ساده‌سازی ارتباط با سرویس‌های وابسته پیشنهاد می‌شود.

14-11. جمع‌بندی

سرویس مدیریت تکالیف یکی از مهم‌ترین سرویس‌های عملیاتی سامانه آموزش آنلاین مدارس است که فرآیند تعامل آموزشی میان معلم و دانش‌آموز را در حوزه تمرین و ارزیابی مدیریت می‌کند. طراحی این سرویس مطابق معماری RESTful، معماری خدمت‌گرا (SOA)، اصول SOLID و الگوهای طراحی GoF انجام شده است تا علاوه بر پاسخگویی به نیازهای فعلی، امکان توسعه قابلیت‌های پیشرفته آموزشی و نگهداری آسان سامانه در آینده نیز فراهم باشد.

15. تعریف رابطهای برنامه‌نویسی سرویس مدیریت آزمون‌های آنلاین (Examination Service)

15-1. معرفی سرویس

سرویس مدیریت آزمون‌های آنلاین مسئول برنامه‌ریزی، ایجاد، زمان‌بندی، اجرا، نظارت و ارزیابی آزمون‌های الکترونیکی در سامانه آموزش آنلاین مدارس است. این سرویس امکان برگزاری انواع آزمون‌های کلاسی، میان‌ترم، پایان‌ترم، آزمون‌های آزمایشی و آزمون‌های خودارزیابی را فراهم می‌کند و فرآیند کامل آزمون از طراحی سؤالات تا اعلام نتایج را مدیریت می‌نماید.

در معماری خدمت‌گرای سامانه، این سرویس به صورت مستقل توسعه یافته و از طریق APIهای استاندارد با سایر سرویس‌ها ارتباط برقرار می‌کند. اطلاعات کاربران، کلاس‌ها، سؤالات، نمرات و اعلان‌ها از طریق سرویس‌های مرتبط تبادل می‌شود تا وابستگی مستقیم میان زیرسیستم‌ها حذف شده و قابلیت توسعه و مقیاس‌پذیری سیستم افزایش یابد.

15-2. مسئولیت‌های سرویس

وظایف اصلی این سرویس عبارت‌اند از:

- ایجاد آزمون آنلاین
- مدیریت بانک سؤالات
- زمان‌بندی آزمون‌ها
- تعیین مدت زمان آزمون
- تخصیص آزمون به کلاس‌ها
- مدیریت شروع و پایان آزمون
- دریافت پاسخ دانش‌آموزان
- تصحیح خودکار سؤالات تستی
- ثبت نمرات نهایی
- ارائه کارنامه آزمون
- تولید گزارش‌های آماری آزمون

15-3. رابطهای برنامه‌نویسی سرویس

سطح دسترسی	شرح عملیات	Method	Endpoint
کاربران مجاز	دریافت فهرست آزمون‌ها	GET	/api/v1/exams
کاربران مجاز	مشاهده اطلاعات آزمون	GET	/api/v1/exams/{id}
معلم	ایجاد آزمون جدید	POST	/api/v1/exams
معلم	ویرایش آزمون	PUT	/api/v1/exams/{id}
معلم	حذف آزمون	DELETE	/api/v1/exams/{id}
دانش‌آموز	شروع آزمون	POST	/api/v1/exams/{id}/start
دانش‌آموز	ارسال پاسخ‌ها	POST	/api/v1/exams/{id}/submit
دانش‌آموز، معلم	دریافت نتیجه آزمون	GET	/api/v1/exams/{id}/result
معلم	ثبت نمره تشریحی	POST	/api/v1/exams/{id}/grade

15-4. API ایجاد آزمون

Endpoint

POST /api/v1/exams

هدف

ایجاد یک آزمون جدید، تعیین کلاس هدف، زمان برگزاری، مدت آزمون، شیوه نمره‌دهی و مشخصات کلی آزمون.

نمونه درخواست (Request)

```
{
  "title":"آزمون فصل اول ریاضی",
  "classId":112,
  "startTime":"2026-11-05T09:00:00",
  "duration":60,
  "totalScore":20,
  "questionCount":25,
  "examType":"Midterm"
}
```

نمونه پاسخ (Response)

```
{
  "success":true,
  "message":"Exam created successfully.",
  "data":{
      "examId":320
  }
}
```

کدهای وضعیت

Status	توضیح
201	آزمون ایجاد شد
400	اطلاعات نامعتبر
401	عدم احراز هویت
403	عدم مجوز ایجاد آزمون
404	کلاس یافت نشد
500	خطای داخلی سرور

15-5. API ارسال پاسخ آزمون

Endpoint

POST /api/v1/exams/{id}/submit

هدف

ثبت پاسخ‌های دانش‌آموز، پایان جلسه آزمون و ارسال پاسخ‌ها جهت ارزیابی.

نمونه درخواست

```
{
  "studentId":325,
  "answers":[
      {"questionId":1,"answer":"B"},
      {"questionId":2,"answer":"C"},
      {"questionId":3,"answer":"15"}
  ]
}
```

نمونه پاسخ

```
{
  "success":true,
  "message":"Exam submitted successfully.",
  "data":{
      "submissionId":980,
      "submittedAt":"2026-11-05T09:55:18"
  }
}
```

15-6. API دریافت نتیجه آزمون

Endpoint

GET /api/v1/exams/{id}/result

هدف

نمایش نتیجه آزمون، نمره کسب‌شده، درصد پاسخ‌های صحیح و وضعیت قبولی یا نیاز به بازبینی.

نمونه پاسخ

```
{
  "success":true,
  "data":{
      "examId":320,
      "studentId":325,
      "score":18.25,
      "correctAnswers":22,
      "wrongAnswers":3,
      "status":"Passed"
  }
}
```

15-7. ارتباط با سایر سرویس‌ها

سرویس مدیریت آزمون‌ها برای اجرای صحیح فرآیند ارزیابی با چندین سرویس سامانه در ارتباط است. اطلاعات کاربران و نقش‌ها از سرویس مدیریت کاربران دریافت می‌شود و اطلاعات کلاس از سرویس مدیریت کلاس‌ها استخراج

می‌گردد. در صورت نیاز، منابع آموزشی مرتبط از سرویس مدیریت محتوا بازیابی می‌شوند و نتایج آزمون پس از پایان فرآیند ارزیابی به سرویس گزارش‌گیری ارسال می‌شود. همچنین سرویس اعلان‌ها مسئول اطلاع‌رسانی زمان شروع آزمون، یادآوری پایان مهلت و انتشار نتایج آزمون است.

15-8. قواعد تجاری (Business Rules)

به منظور تضمین عدالت آموزشی و صحت فرآیند ارزیابی، قواعد زیر در این سرویس اعمال می‌شود:

- هر آزمون باید به یک کلاس مشخص اختصاص داشته باشد.
- تنها معلم مسئول کلاس مجاز به ایجاد یا ویرایش آزمون است.
- هر دانش‌آموز فقط یک مرتبه مجاز به شرکت در آزمون است، مگر آنکه آزمون مجدد توسط معلم تعریف شود.
- پاسخ‌ها پس از پایان زمان آزمون به صورت خودکار ثبت و ارسال می‌شوند.
- امکان ویرایش پاسخ‌ها پس از ثبت نهایی وجود ندارد.
- نمرات سؤالات تستی به صورت خودکار و سؤالات تشریحی توسط معلم محاسبه می‌شوند.
- تمامی رویدادهای آزمون شامل زمان ورود، خروج، ارسال پاسخ و ثبت نمره در لاگ‌های سامانه ذخیره می‌شوند.
- در صورت قطع ارتباط شبکه، پاسخ‌های ثبت‌شده به صورت موقت ذخیره شده و پس از برقراری ارتباط به سامانه ارسال خواهند شد.

15-9. ملاحظات طراحی

این سرویس مطابق اصل **Single Responsibility Principle (SRP)** تنها مسئول مدیریت فرآیند آزمون بوده و عملیات مربوط به کلاس‌ها، کاربران و محتوای آموزشی در سرویس‌های مستقل انجام می‌شود. مطابق اصل **Open/Closed Principle (OCP)** نیز امکان افزودن قابلیت‌هایی مانند آزمون تطبیقی (Adaptive Exam)، تصحیح مبتنی بر هوش مصنوعی، بانک سؤالات هوشمند، نظارت آنلاین (Online Proctoring) و تشخیص تقلب بدون تغییر در هسته اصلی سرویس پیش‌بینی شده است.

بر اساس اصل **Dependency Inversion Principle (DIP)** نیز ارتباط این سرویس با سایر زیرسیستم‌ها تنها از طریق APIهای استاندارد انجام شده و هیچ وابستگی مستقیمی به ساختار داخلی سایر سرویس‌ها وجود ندارد.

15-10. الگوهای طراحی پیشنهادی

برای پیاده‌سازی این سرویس استفاده از الگوهای **Repository Pattern** جهت مدیریت داده‌های آزمون، **Strategy Pattern** برای پشتیبانی از روش‌های مختلف نمره‌دهی، **State Pattern** برای مدیریت وضعیت آزمون (پیش‌نویس، زمان‌بندی‌شده، در حال برگزاری، پایان‌یافته و آرشیوشده)، **Observer Pattern** برای ارسال اعلان‌های خودکار، **DTO (Data Transfer Object)** برای تبادل اطلاعات میان سرویس‌ها و **Facade Pattern** برای مدیریت تعامل با سایر سرویس‌های سامانه پیشنهاد می‌شود.

15-11. جمع‌بندی

سرویس مدیریت آزمون‌های آنلاین یکی از حیاتی‌ترین زیرسیستم‌های سامانه آموزش آنلاین مدارس محسوب می‌شود و مسئول مدیریت فرآیند ارزیابی الکترونیکی دانش‌آموزان است. طراحی این سرویس بر پایه معماری RESTful، معماری خدمت‌گرا (SOA)، اصول SOLID و الگوهای طراحی GoF انجام شده است تا امنیت، مقیاس‌پذیری، قابلیت توسعه، قابلیت نگهداری و یکپارچگی فرآیندهای ارزیابی در سطح سامانه تضمین شود.

16. تعریف رابطهای برنامه‌نویسی سرویس مدیریت حضور و غیاب (Attendance Service)

16-1. معرفی سرویس

سرویس مدیریت حضور و غیاب مسئول ثبت، مدیریت، پایش و گزارش‌گیری وضعیت حضور دانش‌آموزان در جلسات آموزشی حضوری و آنلاین است. این سرویس امکان ثبت خودکار یا دستی حضور، محاسبه میزان مشارکت دانش‌آموزان، ثبت تأخیر و غیبت، ارائه گزارش‌های آماری و ارسال اطلاعات به سایر زیرسیستم‌های سامانه را فراهم می‌کند.

در معماری خدمت‌گرای سامانه، این سرویس به صورت مستقل توسعه یافته و از طریق رابطهای برنامه‌نویسی استاندارد با سرویس‌های مدیریت کاربران، کلاس‌ها، گزارش‌گیری و اعلان‌ها در ارتباط است. استقلال این سرویس موجب افزایش قابلیت نگهداری، توسعه و مقیاس‌پذیری سامانه خواهد شد.

16-2. مسئولیت‌های سرویس

وظایف اصلی این سرویس عبارت‌اند از:

- ثبت حضور دانش‌آموزان
- ثبت غیبت و تأخیر
- ثبت خروج زودهنگام از کلاس
- ثبت حضور خودکار در کلاس‌های آنلاین
- ویرایش وضعیت حضور توسط معلم
- محاسبه درصد حضور دانش‌آموزان
- تولید گزارش‌های حضور و غیاب
- ارسال اطلاعات حضور به سرویس گزارش‌گیری
- اطلاع‌رسانی غیبت به والدین
- آرشیو سوابق حضور و غیاب

16-3. رابطهای برنامه‌نویسی سرویس

Endpoint	Method	شرح عملیات	سطح دسترسی
/api/v1/attendance	GET	دریافت سوابق حضور و غیاب	کاربران مجاز
/api/v1/attendance	POST	ثبت حضور	معلم
/api/v1/attendance/{id}	PUT	ویرایش وضعیت حضور	معلم
/api/v1/attendance/class/{classId}	GET	مشاهده حضور کلاس	معلم، مدیر
/api/v1/attendance/student/{studentId}	GET	مشاهده سابقه حضور دانش‌آموز	دانش‌آموز، والدین، مدیر
/api/v1/attendance/report	GET	گزارش آماری حضور	مدیر

16-4. API ثبت حضور

Endpoint

POST /api/v1/attendance

هدف

ثبت وضعیت حضور دانش‌آموزان در یک جلسه آموزشی مشخص.

نمونه درخواست (Request)

```
{
  "classId":112,
  "sessionId":45,
  "studentId":325,
  "status":"Present",
  "attendanceTime":"2026-11-18T08:03:20"
}
```

نمونه پاسخ (Response)

```
{
  "success":true,
  "message":"Attendance registered successfully.",
  "data":{
      "attendanceId":845
  }
}
```

کدهای وضعیت

Status	توضیح
201	ثبت حضور موفق
400	اطلاعات نامعتبر
401	عدم احراز هویت
403	عدم مجوز ثبت حضور
404	جلسه آموزشی یافت نشد
500	خطای داخلی سرور

16-5. API دریافت سابقه حضور دانش‌آموز

Endpoint

GET /api/v1/attendance/student/{studentId}

هدف

نمایش سوابق کامل حضور و غیاب دانش‌آموز در تمامی کلاس‌های آموزشی.

نمونه پاسخ

```
{
  "success":true,
  "data":{
      "studentId":325,
      "present":98,
      "absent":2,
      "late":4,
      "attendanceRate":"96%"
  }
}
```

16-6. ارتباط با سایر سرویس‌ها

سرویس مدیریت حضور و غیاب برای انجام وظایف خود با چندین سرویس دیگر تعامل دارد. اطلاعات کاربران از سرویس مدیریت کاربران و اطلاعات کلاس‌ها و جلسات از سرویس مدیریت کلاس دریافت می‌شود. پس از ثبت وضعیت حضور، اطلاعات به سرویس گزارش‌گیری ارسال می‌شود تا شاخص‌های عملکرد تحصیلی محاسبه شوند. همچنین در صورت ثبت غیبت یا تأخیر، سرویس اعلان‌ها پیام‌های لازم را برای دانش‌آموز، والدین و معلم ارسال می‌کند.

16-7. قواعد تجاری (Business Rules)

به منظور حفظ صحت اطلاعات حضور و غیاب، قواعد زیر در این سرویس اعمال می‌شود:

- حضور تنها برای جلسات فعال کلاس قابل ثبت است.
- هر دانش‌آموز در هر جلسه فقط یک رکورد حضور خواهد داشت.
- وضعیت حضور می‌تواند شامل «حاضر»، «غایب»، «با تأخیر» یا «خروج زودهنگام» باشد.
- پس از پایان جلسه، تنها معلم یا مدیر مدرسه مجاز به ویرایش وضعیت حضور است.
- تمامی تغییرات در سوابق حضور در لاگ‌های سامانه ثبت می‌شوند.
- در کلاس‌های آنلاین، حضور اولیه می‌تواند به صورت خودکار بر اساس ورود کاربر به جلسه ثبت شود.

16-8. ملاحظات طراحی

این سرویس مطابق اصل **Single Responsibility Principle (SRP)** تنها مسئول مدیریت اطلاعات حضور و غیاب است و هیچ منطق مربوط به آزمون‌ها، تکالیف یا نمرات در آن پیاده‌سازی نشده است. همچنین مطابق اصل **Open/Closed Principle (OCP)** امکان افزودن قابلیت‌هایی مانند تشخیص حضور مبتنی بر موقعیت مکانی (GPS)، تشخیص چهره، احراز هویت بیومتریک یا تحلیل هوشمند میزان مشارکت دانش‌آموزان بدون تغییر در ساختار اصلی سرویس پیش‌بینی شده است.

بر اساس اصل **Dependency Inversion Principle (DIP)** نیز ارتباط این سرویس با سایر زیرسیستم‌ها صرفاً از طریق APIهای استاندارد انجام شده و وابستگی مستقیمی به ساختار داخلی آن‌ها وجود ندارد.

16-9. الگوهای طراحی پیشنهادی

برای پیاده‌سازی این سرویس استفاده از **Repository Pattern** جهت مدیریت داده‌های حضور، **Strategy Pattern** برای پشتیبانی از روش‌های مختلف ثبت حضور، **Observer Pattern** برای ارسال اعلان‌های خودکار، **DTO (Data Transfer Object)** برای تبادل اطلاعات میان سرویس‌ها و **Facade Pattern** برای ساده‌سازی تعامل با سایر زیرسیستم‌ها پیشنهاد می‌شود.

16-10. جمع‌بندی

سرویس مدیریت حضور و غیاب یکی از سرویس‌های کلیدی سامانه آموزش آنلاین مدارس است که مسئول ثبت و تحلیل وضعیت حضور دانش‌آموزان در فرآیندهای آموزشی است. طراحی این سرویس مطابق معماری RESTful، معماری خدمت‌گرا (SOA)، اصول SOLID و الگوهای طراحی GoF انجام شده است تا ضمن تضمین صحت اطلاعات، امکان توسعه، نگهداری و یکپارچه‌سازی با سایر سرویس‌های سامانه فراهم شود.

17. تعریف رابطهای برنامه‌نویسی سرویس پیام‌رسان داخلی (Messaging Service)

17-1. معرفی سرویس

سرویس پیام‌رسان داخلی مسئول مدیریت تمامی ارتباطات متنی میان کاربران سامانه آموزش آنلاین مدارس است. این سرویس امکان برقراری ارتباط مستقیم و گروهی میان معلمان، دانش‌آموزان، والدین و مدیران را فراهم کرده و بستری امن، یکپارچه و کنترل‌شده برای تبادل اطلاعات آموزشی ایجاد می‌کند.

این سرویس به گونه‌ای طراحی شده است که علاوه بر ارسال و دریافت پیام‌های متنی، قابلیت مدیریت مکالمات، گروه‌های کلاسی، اشتراک‌گذاری فایل‌های آموزشی، ارسال تصاویر و اسناد، ثبت تاریخچه گفتگوها و جستجو در پیام‌ها را نیز فراهم سازد. در معماری خدمت‌گرای سامانه، سرویس پیام‌رسان به صورت مستقل پیاده‌سازی شده و از طریق APIهای استاندارد با سرویس‌های مدیریت کاربران، کلاس‌ها، اعلان‌ها و مدیریت محتوا در تعامل است.

17-2. مسئولیت‌های سرویس

مهم‌ترین مسئولیت‌های این سرویس عبارت‌اند از:

- ایجاد گفتگوهای خصوصی
- ایجاد گفتگوهای گروهی
- ارسال پیام متنی
- ارسال فایل، تصویر و اسناد
- ویرایش یا حذف پیام
- مدیریت اعضای گروه‌ها
- مشاهده تاریخچه گفتگوها
- جستجو در پیام‌ها
- نمایش وضعیت خوانده شدن پیام‌ها
- مدیریت وضعیت آنلاین کاربران
- ارائه اطلاعات پیام‌ها به سایر سرویس‌ها

17-3. رابطهای برنامه‌نویسی سرویس

Endpoint	Method	شرح عملیات	سطح دسترسی
/api/v1/messages	GET	دریافت پیام‌ها	کاربران مجاز
/api/v1/messages	POST	ارسال پیام	کاربران مجاز
/api/v1/messages/{id}	GET	مشاهده پیام	کاربران مجاز
/api/v1/messages/{id}	PUT	ویرایش پیام	فرستنده
/api/v1/messages/{id}	DELETE	حذف پیام	فرستنده
/api/v1/conversations	GET	دریافت گفتگوها	کاربران مجاز
/api/v1/conversations	POST	ایجاد گفتگو	کاربران مجاز
/api/v1/conversations/{id}/members	POST	افزودن عضو	مدیر گروه
/api/v1/messages/search	GET	جستجوی پیام‌ها	کاربران مجاز

17-4. API ارسال پیام

Endpoint

POST /api/v1/messages

هدف

ارسال پیام متنی یا چندرسانه‌ای میان کاربران سامانه یا اعضای یک گروه آموزشی.

نمونه درخواست (Request)

```
{
  "conversationId":205,
  "senderId":48,
  "message":"لطفاً تکلیف جلسه امروز را تا فردا ارسال کنید.",
  "messageType":"Text"
}
```

نمونه پاسخ (Response)

```
{
  "success":true,
  "message":"Message sent successfully.",
  "data":{
      "messageId":9654,
      "sentAt":"2026-11-20T09:45:16"
  }
```

}

کدهای وضعیت

Status	توضیح
201	پیام ارسال شد
400	اطلاعات نامعتبر
401	عدم احراز هویت
403	عدم مجوز ارسال پیام
404	گفتگو یافت نشد
500	خطای داخلی سرور

17-5. API دریافت تاریخچه گفتگو

Endpoint

GET /api/v1/conversations/{id}

هدف

دریافت لیست کامل پیام‌های یک گفتگو همراه با اطلاعات فرستنده، زمان ارسال و وضعیت مشاهده پیام.

نمونه پاسخ

```
{
  "success":true,
  "data":{
      "conversationId":205,
      "messages":[
         {
            "messageId":9654,
            "sender":"محمد رضایی",
            "message":"لطفاً تکلیف جلسه امروز را تا فردا ارسال
کنید.",
            "time":"2026-11-20T09:45:16",
            "status":"Read"
         }
      ]
  }
}
```

17-6. ارتباط با سایر سرویس‌ها

سرویس پیام‌رسان برای احراز هویت و تعیین سطح دسترسی کاربران از سرویس مدیریت کاربران استفاده می‌کند. اطلاعات کلاس‌ها و گروه‌های آموزشی از سرویس مدیریت کلاس دریافت می‌شود تا گفتگوهای گروهی بر اساس ساختار کلاس‌ها ایجاد شوند. همچنین فایل‌های ضمیمه از طریق سرویس مدیریت محتوای آموزشی ذخیره و بازیابی شده و در صورت ارسال پیام جدید، سرویس اعلان‌ها کاربران مقصد را از وجود پیام جدید مطلع می‌کند. اطلاعات آماری مربوط به حجم پیام‌ها و میزان تعامل کاربران نیز به سرویس گزارش‌گیری ارسال می‌شود.

17-7. قواعد تجاری (Business Rules)

به منظور حفظ امنیت و یکپارچگی ارتباطات، قواعد زیر در این سرویس اعمال می‌شود:

- تنها کاربران احراز هویت‌شده مجاز به ارسال پیام هستند.
- هر کاربر فقط به گفتگوهایی که عضو آن است دسترسی خواهد داشت.
- حذف پیام تنها توسط فرستنده یا مدیر سامانه امکان‌پذیر است.
- تمامی پیام‌ها همراه با زمان ارسال و شناسه فرستنده ذخیره می‌شوند.
- حداکثر حجم فایل‌های ضمیمه مطابق سیاست‌های سامانه کنترل می‌شود.
- پیام‌های حذف‌شده در سوابق ممیزی سیستم ثبت خواهند شد.
- تمامی ارتباطات بین کاربران از طریق پروتکل‌های رمزنگاری‌شده انجام می‌شود.

17-8. ملاحظات طراحی

این سرویس مطابق اصل **Single Responsibility Principle (SRP)** صرفاً مسئول مدیریت پیام‌ها و مکالمات کاربران است و منطق مربوط به اعلان‌ها، احراز هویت یا مدیریت کلاس‌ها در سرویس‌های مستقل پیاده‌سازی شده است. همچنین مطابق اصل **Open/Closed Principle (OCP)** ساختار سرویس به گونه‌ای طراحی شده است که قابلیت‌هایی مانند پیام صوتی، تماس تصویری، اشتراک‌گذاری صفحه، واکنش به پیام‌ها، پیام‌های موقت (Ephemeral Messages) یا ترجمه خودکار پیام‌ها بدون تغییر در هسته اصلی سرویس قابل افزودن باشد.

بر اساس اصل **Dependency Inversion Principle (DIP)** نیز تمامی تعاملات این سرویس با سایر زیرسیستم‌ها از طریق APIهای استاندارد انجام می‌شود و وابستگی مستقیمی به ساختار داخلی سایر سرویس‌ها وجود ندارد.

17-9. الگوهای طراحی پیشنهادی

برای پیاده‌سازی این سرویس استفاده از **Repository Pattern** جهت مدیریت داده‌های پیام‌ها، **Observer Pattern** برای اطلاع‌رسانی بلادرنگ، **Mediator Pattern** برای مدیریت تعامل کاربران در گفتگوها، **DTO (Data Transfer Object)** برای تبادل داده میان سرویس‌ها، **Facade Pattern** برای ساده‌سازی ارتباط با سرویس‌های وابسته و **Strategy Pattern** برای مدیریت انواع مختلف پیام‌ها پیشنهاد می‌شود.

17-10. جمع‌بندی

سرویس پیام‌رسان داخلی یکی از سرویس‌های کلیدی سامانه آموزش آنلاین مدارس است که بستر ارتباطی امن، سریع و یکپارچه میان کاربران را فراهم می‌کند. طراحی این سرویس بر پایه معماری RESTful، معماری خدمت‌گرا (SOA)، اصول SOLID و الگوهای طراحی GoF انجام شده است تا علاوه بر پاسخگویی به نیازهای فعلی، قابلیت توسعه، نگهداری، امنیت و مقیاس‌پذیری سامانه در آینده نیز تضمین شود.

18. تعریف رابطهای برنامه‌نویسی سرویس اعلان‌ها (Notification Service)

18-1. معرفی سرویس

سرویس اعلان‌ها مسئول مدیریت، تولید، زمان‌بندی و ارسال تمامی اعلان‌های سیستمی سامانه آموزش آنلاین مدارس است. این سرویس نقش مهمی در اطلاع‌رسانی رویدادهای آموزشی و مدیریتی ایفا می‌کند و کاربران را از رخدادهایی نظیر ایجاد کلاس، انتشار محتوای آموزشی، تعریف تکلیف، زمان برگزاری آزمون، ثبت نمرات، تغییر برنامه کلاس، ثبت حضور و غیاب و سایر رویدادهای مهم مطلع می‌سازد.

در معماری خدمت‌گرای سامانه، این سرویس به صورت مستقل از سایر سرویس‌ها توسعه یافته و بر اساس معماری **Event-Driven** عمل می‌کند. سایر سرویس‌ها پس از وقوع رویداد، یک پیام رویداد (Event) منتشر کرده و سرویس اعلان‌ها با دریافت آن، اعلان مناسب را تولید و از طریق کانال ارتباطی مناسب برای کاربران ارسال می‌کند. این رویکرد موجب کاهش وابستگی میان سرویس‌ها، افزایش مقیاس‌پذیری و بهبود کارایی سامانه می‌شود.

18-2. مسئولیت‌های سرویس

وظایف اصلی این سرویس عبارت‌اند از:

- تولید اعلان‌های سیستمی
- ارسال اعلان درون‌برنامه‌ای (In-App Notification)
- ارسال Push Notification
- ارسال ایمیل
- ارسال پیامک در شرایط خاص
- زمان‌بندی اعلان‌ها
- مدیریت وضعیت خوانده شدن اعلان‌ها
- مدیریت اولویت اعلان‌ها
- ثبت تاریخچه اعلان‌ها
- ارائه گزارش وضعیت ارسال اعلان‌ها

18-3. رابطهای برنامه‌نویسی سرویس

Endpoint	Method	شرح عملیات	سطح دسترسی
/api/v1/notifications	GET	دریافت اعلان‌های کاربر	کاربران مجاز
/api/v1/notifications	POST	ایجاد اعلان جدید	سرویس‌های داخلی
/api/v1/notifications/{id}	GET	مشاهده جزئیات اعلان	کاربران مجاز
/api/v1/notifications/{id}/read	PUT	علامت‌گذاری به عنوان خوانده‌شده	کاربران مجاز
/api/v1/notifications/unread	GET	دریافت اعلان‌های خوانده‌نشده	کاربران مجاز
/api/v1/notifications/history	GET	مشاهده تاریخچه اعلان‌ها	مدیر سیستم

18-4. API ایجاد اعلان

Endpoint

POST /api/v1/notifications

هدف

ایجاد و ارسال یک اعلان جدید برای یک یا چند کاربر بر اساس رویداد ایجادشده در سایر سرویس‌های سامانه.

نمونه درخواست (Request)

```
{
  "title":"آزمون جدید",
  "message":"آزمون فصل اول ریاضی برای روز شنبه برنامه‌ریزی
شد.",
  "receiverIds":[325,326,327],
  "notificationType":"Push",
  "priority":"High"
}
```

نمونه پاسخ (Response)

```
{
  "success":true,
  "message":"Notification queued successfully.",
  "data":{
      "notificationId":5421
  }
}
```

کدهای وضعیت

توضیح	Status
اعلان ایجاد شد	201
اطلاعات نامعتبر	400
عدم احراز هویت	401
عدم مجوز	403
خطای داخلی سرور	500

18-5. API دریافت اعلان‌های کاربر

Endpoint

GET /api/v1/notifications

هدف

دریافت تمامی اعلان‌های مربوط به کاربر جاری همراه با وضعیت مشاهده و زمان ارسال.

نمونه پاسخ

```
{
  "success":true,
  "data":[
      {
        "notificationId":5421,
        "title":"آزمون جدید",
        "status":"Unread",
        "priority":"High",
        "createdAt":"2026-11-25T08:30:15"
      }
  ]
}
```

18-6. ارتباط با سایر سرویس‌ها

سرویس اعلان‌ها تقریباً با تمامی سرویس‌های سامانه در ارتباط است. این سرویس پس از دریافت رویدادهای منتشرشده از سرویس‌های مدیریت کاربران، کلاس‌ها، محتوا، تکالیف، آزمون‌ها، حضور و غیاب، پیام‌رسان و گزارش‌گیری، اعلان مناسب را تولید و از طریق کانال ارتباطی انتخاب‌شده برای کاربران ارسال می‌کند. همچنین اطلاعات مربوط به وضعیت ارسال و مشاهده اعلان‌ها در اختیار سرویس گزارش‌گیری قرار می‌گیرد تا شاخص‌های عملکرد سامانه تحلیل شوند.

18-7. قواعد تجاری (Business Rules)

برای تضمین صحت عملکرد سرویس، قواعد زیر اعمال می‌شود:

- هر اعلان باید حداقل یک گیرنده مشخص داشته باشد.
- اعلان‌های با اولویت بالا بلافاصله پس از ایجاد ارسال می‌شوند.
- اعلان‌های زمان‌بندی‌شده تا زمان مقرر در صف پردازش باقی می‌مانند.
- هر اعلان دارای وضعیت «ارسال‌شده»، «تحویل‌شده» و «خوانده‌شده» است.
- تاریخچه اعلان‌ها به منظور ممیزی و گزارش‌گیری در سامانه نگهداری می‌شود.
- اعلان‌های تکراری ناشی از یک رویداد واحد نباید برای کاربر بیش از یک بار ارسال شوند.

18-8. ملاحظات طراحی

این سرویس مطابق اصل **Single Responsibility Principle (SRP)** تنها مسئول مدیریت اعلان‌ها بوده و هیچ منطق مربوط به کلاس‌ها، آزمون‌ها یا پیام‌رسانی در آن پیاده‌سازی نشده است. همچنین مطابق اصل **Open/Closed Principle (OCP)** قابلیت افزودن کانال‌های ارتباطی جدید مانند پیام‌رسان‌های خارجی، اعلان از طریق شبکه‌های اجتماعی یا سرویس‌های شخص ثالث بدون تغییر در هسته سرویس فراهم شده است.

از آنجا که معماری کلی سامانه ترکیبی بوده و بخش اطلاع‌رسانی بر پایه **Event-Driven Architecture** طراحی شده است، این سرویس از الگوی انتشار و اشتراک رویدادها (Publish/Subscribe) استفاده می‌کند تا رویدادهای تولیدشده توسط سایر سرویس‌ها را دریافت کرده و بدون ایجاد وابستگی مستقیم، فرآیند ارسال اعلان‌ها را مدیریت نماید.

18-9. الگوهای طراحی پیشنهادی

در پیاده‌سازی این سرویس استفاده از **Observer Pattern** برای دریافت رویدادها، **Publisher-Subscriber Pattern** برای پردازش اعلان‌های مبتنی بر رویداد، **Factory Method** برای ایجاد انواع اعلان، **Strategy Pattern** برای انتخاب کانال ارسال، **Repository Pattern** برای مدیریت تاریخچه اعلان‌ها و **DTO (Data Transfer Object)** برای تبادل داده میان سرویس‌ها پیشنهاد می‌شود.

18-10. جمع‌بندی

سرویس اعلان‌ها یکی از سرویس‌های زیرساختی مهم سامانه آموزش آنلاین مدارس محسوب می‌شود که وظیفه اطلاع‌رسانی بلادرنگ رویدادهای آموزشی و مدیریتی را بر عهده دارد. طراحی این سرویس بر پایه معماری RESTful، معماری خدمت‌گرا (SOA)، معماری رویدادمحور (Event-Driven Architecture)، اصول SOLID و الگوهای طراحی GoF انجام شده است تا ضمن تضمین سرعت و قابلیت اطمینان در ارسال اعلان‌ها، امکان توسعه و مقیاس‌پذیری سامانه در آینده نیز فراهم شود.

19. تعریف رابطهای برنامه‌نویسی سرویس گزارش‌گیری و تحلیل (Reporting & Analytics Service)

19-1. معرفی سرویس

سرویس گزارش‌گیری و تحلیل یکی از سرویس‌های راهبردی سامانه آموزش آنلاین مدارس است که وظیفه جمع‌آوری، پردازش، تحلیل و ارائه اطلاعات مدیریتی و آموزشی را بر عهده دارد. این سرویس داده‌های تولیدشده توسط سایر سرویس‌های سامانه را دریافت کرده و آن‌ها را به گزارش‌های عملیاتی، مدیریتی و آماری تبدیل می‌کند تا مدیران مدارس، معلمان، والدین و مدیران سامانه بتوانند بر اساس داده‌های واقعی تصمیم‌گیری کنند.

برخلاف سایر سرویس‌ها که مسئول انجام عملیات تراکنشی هستند، این سرویس تمرکز خود را بر تحلیل داده‌ها، استخراج شاخص‌های کلیدی عملکرد (Key Performance Indicators - KPI)، تولید داشبوردهای مدیریتی و ارائه گزارش‌های تصمیم‌یار قرار می‌دهد.

در معماری خدمت‌گرای سامانه، سرویس گزارش‌گیری هیچ وابستگی مستقیمی به پایگاه داده سایر سرویس‌ها ندارد و تمامی اطلاعات مورد نیاز را از طریق APIها یا رویدادهای منتشرشده توسط سایر سرویس‌ها دریافت می‌کند. این رویکرد ضمن حفظ استقلال سرویس‌ها، امکان توسعه مستقل، مقیاس‌پذیری و پردازش حجم بالای داده‌ها را فراهم می‌کند.

19-2. مسئولیت‌های سرویس

وظایف اصلی این سرویس عبارت‌اند از:

- جمع‌آوری اطلاعات آموزشی از تمامی سرویس‌ها
- تولید گزارش‌های مدیریتی
- تولید گزارش‌های تحصیلی دانش‌آموزان
- تحلیل عملکرد معلمان
- تحلیل میزان مشارکت دانش‌آموزان
- تحلیل حضور و غیاب
- تحلیل نتایج آزمون‌ها
- تحلیل وضعیت تکالیف
- تولید داشبوردهای مدیریتی
- تولید شاخص‌های کلیدی عملکرد (KPI)
- ارائه گزارش‌های آماری
- خروجی گرفتن از گزارش‌ها در قالب Excel، PDF و CSV

19-3. رابطهای برنامه‌نویسی سرویس

Endpoint	Method	شرح عملیات	سطح دسترسی
/api/v1/reports	GET	دریافت فهرست گزارش‌ها	مدیر
/api/v1/reports/student/{id}	GET	گزارش عملکرد دانش‌آموز	مدیر، معلم، والدین
/api/v1/reports/class/{id}	GET	گزارش عملکرد کلاس	مدیر، معلم
/api/v1/reports/teacher/{id}	GET	گزارش عملکرد معلم	مدیر
/api/v1/reports/dashboard	GET	داشبورد مدیریتی	مدیر
/api/v1/reports/export	POST	خروجی PDF/Excel	مدیر
/api/v1/reports/statistics	GET	گزارش‌های آماری	مدیر سیستم

19-4. API دریافت گزارش عملکرد دانش‌آموز

Endpoint

GET /api/v1/reports/student/{studentId}

هدف

ارائه گزارش جامع وضعیت تحصیلی دانش‌آموز شامل نمرات، حضور و غیاب، تکالیف، آزمون‌ها، میزان مشارکت آموزشی و شاخص‌های عملکرد.

نمونه پاسخ (Response)

```
{
  "success": true,
  "data": {
    "studentId": 325,
    "averageScore": 18.42,
    "attendanceRate": "96%",
    "completedAssignments": 28,
    "pendingAssignments": 2,
    "passedExams": 8,
    "failedExams": 0,
    "overallStatus": "Excellent"
  }
}
```

Status	توضیح
200	گزارش با موفقیت تولید شد
401	عدم احراز هویت
403	عدم مجوز دسترسی
404	کاربر یافت نشد
500	خطای داخلی سرور

کدهای وضعیت

19-5. API دریافت داشبورد مدیریتی

Endpoint

GET /api/v1/reports/dashboard

هدف

تولید داشبورد مدیریتی لحظه‌ای برای مدیران مدارس شامل مهم‌ترین شاخص‌های عملکرد سامانه.

نمونه پاسخ

```
{
  "success": true,
  "data": {
    "activeStudents": 2580,
    "activeTeachers": 186,
    "onlineClassesToday": 142,
    "assignmentsPending": 735,
    "todayAttendanceRate": "94%",
    "averageExamScore": 17.8,
    "systemAvailability": "99.98%"
  }
}
```

19-6. ارتباط با سایر سرویس‌ها

سرویس گزارش‌گیری گسترده‌ترین تعامل را با سایر زیرسیستم‌های سامانه دارد. اطلاعات کاربران از سرویس مدیریت کاربران، اطلاعات کلاس‌ها از سرویس مدیریت کلاس‌ها، داده‌های محتوای آموزشی از سرویس مدیریت محتوا، وضعیت تکالیف از سرویس مدیریت تکالیف، نتایج آزمون‌ها از سرویس آزمون، اطلاعات حضور و غیاب از سرویس حضور و

غیاب، آمار پیام‌های آموزشی از سرویس پیام‌رسان و وضعیت اعلان‌ها از سرویس اعلان‌ها دریافت می‌شود. تمامی این داده‌ها پس از پردازش و تحلیل در قالب گزارش‌ها و داشبوردهای مدیریتی ارائه می‌شوند.

19-7. قواعد تجاری (Business Rules)

به منظور تضمین صحت و محرمانگی گزارش‌ها، قواعد زیر اعمال می‌شود:

- هر کاربر تنها به گزارش‌های متناسب با نقش خود دسترسی خواهد داشت.
- اطلاعات گزارش‌ها باید بر اساس آخرین داده‌های ثبت‌شده تولید شوند.
- تمامی گزارش‌ها دارای زمان تولید و شناسه تولیدکننده گزارش هستند.
- گزارش‌های مدیریتی تنها در اختیار مدیران مجاز قرار می‌گیرند.
- اطلاعات شخصی کاربران مطابق سیاست‌های حفظ حریم خصوصی نمایش داده می‌شوند.
- گزارش‌های خروجی نباید شامل اطلاعات محرمانه کاربران غیرمجاز باشند.
- تمامی درخواست‌های تولید گزارش در سامانه ثبت و ممیزی می‌شوند.

19-8. ملاحظات طراحی

این سرویس مطابق اصل **Single Responsibility Principle (SRP)** تنها مسئول پردازش، تحلیل و ارائه گزارش‌ها است و هیچ عملیات تراکنشی مربوط به کلاس‌ها، آزمون‌ها یا تکالیف را انجام نمی‌دهد.

مطابق اصل **Open/Closed Principle (OCP)** نیز ساختار سرویس به گونه‌ای طراحی شده است که در آینده بتوان بدون تغییر در هسته اصلی، گزارش‌های جدید، داشبوردهای اختصاصی، تحلیل‌های مبتنی بر هوش مصنوعی، پیش‌بینی عملکرد دانش‌آموزان و شاخص‌های جدید آموزشی را به سامانه اضافه کرد.

همچنین مطابق اصل **Dependency Inversion Principle (DIP)** این سرویس هیچ وابستگی مستقیمی به پایگاه داده سایر سرویس‌ها نداشته و تمامی اطلاعات را از طریق APIهای استاندارد یا رویدادهای منتشرشده دریافت می‌کند. این رویکرد موجب کاهش وابستگی میان زیرسیستم‌ها و افزایش قابلیت توسعه سامانه خواهد شد.

19-9. الگوهای طراحی پیشنهادی

در پیاده‌سازی این سرویس استفاده از الگوهای **Repository Pattern** برای مدیریت داده‌های گزارش، **Facade Pattern** برای تجمیع اطلاعات سرویس‌های مختلف، **Strategy Pattern** برای تولید انواع گزارش‌ها، **Builder Pattern** برای ساخت گزارش‌های پیچیده، **DTO (Data Transfer Object)** برای انتقال داده‌ها، **Observer Pattern** برای به‌روزرسانی خودکار داشبوردها و **Caching Pattern** برای افزایش سرعت تولید گزارش‌های پرتکرار پیشنهاد می‌شود.

19-10. شاخص‌های کلیدی عملکرد (Key Performance Indicators)

این سرویس شاخص‌های کلیدی عملکرد زیر را به صورت مستمر محاسبه و پایش می‌کند:

- میانگین نمرات دانش‌آموزان

- نرخ حضور و غیاب
- نرخ تکمیل تکالیف
- درصد قبولی آزمون‌ها
- میزان مشارکت کاربران در کلاس‌های آنلاین
- تعداد کاربران فعال روزانه
- نرخ استفاده از محتوای آموزشی
- میزان تعامل کاربران در پیام‌رسان داخلی
- میانگین زمان پاسخ‌گویی سامانه
- دسترس‌پذیری (Availability) سامانه
- نرخ خطاهای عملیاتی
- میزان استفاده از منابع زیرساختی

این شاخص‌ها مبنای تصمیم‌گیری مدیران مدارس، مدیران سامانه و تیم پشتیبانی برای ارزیابی کیفیت خدمات آموزشی و برنامه‌ریزی‌های آینده خواهند بود.

19-11. جمع‌بندی

سرویس گزارش‌گیری و تحلیل یکی از حیاتی‌ترین سرویس‌های سامانه آموزش آنلاین مدارس است که با تجمیع و تحلیل داده‌های تولیدشده توسط سایر سرویس‌ها، اطلاعات ارزشمند مدیریتی و آموزشی را در اختیار ذی‌نفعان قرار می‌دهد. طراحی این سرویس بر پایه معماری RESTful، معماری خدمت‌گرا (SOA)، اصول SOLID و الگوهای طراحی GoF انجام شده است تا ضمن تضمین استقلال، مقیاس‌پذیری، امنیت و قابلیت توسعه، امکان تصمیم‌گیری مبتنی بر داده (Data-Driven Decision Making) را برای مدیران و مسئولان آموزشی فراهم سازد.