مستند مدلسازی کلاس ها

سیستم ردیابی و مدیریت هزینه

فهرست مطالب

روابط بین کلاس ها

روابط وراثت

روابط ترکیب و تجمع

روابط وابستگی

ضمائِم

1. مقدمه

1.1 هدف

هدف از این سند، ارائه مدل های کلاس سیستم ردیابی و مدیریت هزینه است. این سند ساختار کلاس ها، ویژگی ها، متد ها و روابط بین آنها را توصیف می کند و به عنوان راهنمایی برای پیاده سازی سیستم، مورد استفاده قرار می گیرد.

1.2 دامنه کاربرد

این سند شامل مدل های کلاس های سیستم ردیابی و مدیریت هزینه است. این مدل ها، بر اساس نیازمندی ها و موارد کاربری مشخص شده در مستندات پیشین، طراحی شده اند.

1.3 تعاریف و اختصارات

کلاس (Class): یا قالب برای ایجاد اشیاء که ویژگی ها و رفتار های مشابه دارند.

ویژگی (Attribute): داده های مرتبط با یک کلاس.

متد (Method): رفتار های مرتبط با یک کلاس.

رابطه (Relationship): ارتباط بین کلاس ها

وراثت (Inheritamce): یک کلاس از کلاس دیگر، ویژگی ها و متد ها را به ارث می برد.

ترکیب (Composition): رابطه قوی بین کلاس ها که در آن، عمر شیء وابسته به شیء دیگر است.

تجمع (Aggregation): رابطه ضعیف بین کلاس ها که در آن اشیاء، مستقل از هم هستند.

MVC: الگوی معماری Model-View-Controller

DTO: Data Transfer Object

DAO: Data Access Object

2. معماری کلی سیستم

2.1 معماری لایه ای

سیستم ردیابی و مدیریت هزینه از معماری لایه ای (MVC(Model-View-Controller، استفاده می کند.

این معماری سیستم را، به سه لایه اصلی تقسیم می کند:

لایه مدل (Model): این لایه شامل کلاس های موجودیت (Entity)و منطق کسب و کار است. کلاس های این لایه، داده های سیستم را نمایش می دهند و عملیات مرتبط با آنها را، انجام می دهند.

لایه نما (View): این لایه مسئول نمایش اطلاعات به کاربر و دریافت ورودی از کاربر است. این لایه شامل رابط کاربری (UI)، می شود.

لایه کنترلر (Controller): این لایه واسط بین لایه مدل و لایه نما است. کنترلرها درخواست های کاربر را دریافت، پردازش و سپس پاسخ مدل را به لایه نما، برمیگردانند. علاوه بر لایه های اصلی MVC، لایه های دیگری نیز در معماری سیستم، وجود دارند:

لایه سرویس(Service): این لایه بین کنترلرها و مدل ها، قرار می گیرد و منطق کسب وکار پیچیده را، پیاده سازی می کند.

لایه دسترسی به داده (Data Access): این لایه مسئول تعامل با پایگاه داده است و عملیات CRUD (ایجاد، خواندن، بروزرسانی، حذف) را انجام می دهد.

لایه ابزار (Utility): این لایه شامل کلاس های کمکی برای عملیات مشترک مانند اعتبارسنجی، تبدیل داده و مدیریت خطا است.

2.2 الگو های طراحی

در طراحی سیستم ردیابی و مدیریت هزینه، از الگو های طراحی زیر استفاده شده است:

الگوی Repository: برای جداسازی منطق دسترسی به داده از منطق کسب و کار.

الگوی DTO (Data Transfer Object): برای انتقال داده بین لایه های مختلف.

الگوی Singleton: برای کلاس هایی که فقط یک نمونه از آنها نیاز است، مانند مدیریت اتصال به پایگاه داده.

الگوی Factory: برای ایجاد اشیاء مرتبط به یکدیگر

الگوی Observer: برای پیاده سازی سیستم یادآوری و اعلان ها.

الگوی Strategy: برای پیاده سازی روش های مختلف پشتیبان گیری و همگام سازی.

3. نمودار کلاس

3.1 نمای کلی

[اینجا نمودار کلاس کلی قرار می گیرد - به عنوان نمودار 1]

3.2 لایه مدل

[اینجا نمودار کلاس لایه مدل قرار می گیرد - به عنوان نمودار 2]

3.3 لایه سرویس

[اینجا نمودار کلاس لایه سرویس قرار می گیرد - به عنوان نمودار 3]

3.4 لایه کنترلر

[اینجا نمودار کلاس لایه کنترلر قرار می گیرد - به عنوان نمودار 4]

4. توضیحات کلاس ها

4.1 کلاس های موجودیت (Entity)

4.1.1 کلاس User

این کلاس، نماینده کاربران سیستم است.

ویژگی ها:

Id: شناسه منحصر به فرد کاربر (long)

username: نام کاربری (string)

email: آدرس ایمیل (string)

password: رمز عبور رمزنگاری شده (string)

firstName: نام (string)

lastName: نام خانوادگی (string)

firstName: نام (string)

createdAt: تاریخ ایجاد حساب کاربری (Date)

updatedAt: تاریخ آخرین بروزرسانی حساب کاربری (Date)

متد ها:

Register(): ثبت نام کاربر جدید

login(): ورود کاربر به سیستم

resetPassword(): بازیابی رمز عبور

updateProfile(): بروزرسانی اطلاعات پروفایل

4.2.1 کلاس Transaction

این کلاس نماینده تراکنش های مالی (هزینه ها و درآمدها)، است.

id: شناسه منحصر به فرد تراکنش (long)

amount: مبلغ (Date)

date:تاریخ تراکنش (Date)

description: توضیحات (String)

type: نوع تراکنش (TransactionType)

categoryId: شناسه دسته بندی (long)

userId: شناسه کاربر (long)

imageUrl: آدرس تصویر رسید یا فاکتور (string)

isRecurring: آیا تراکنش تکراری است (boolean)

recurringIngo: اطلاعات کاربر (RecurringInfo)

createdAt: تاریخ ایجاد (Date)

updatedAt: تاریخ آخرین بروزرسانی (Date)

متد ها:

create():ایجاد تراکنش جدید

update():بروزرسانی تراکنش

delete():حذف تراکنش

getById():دریافت تراکنش با شناسه

دریافت تمام تراکنش های یک کاربر:getByUser()

دریافت تمام تراکنش های یک دسته بندی:getByCategory()

دریافت تراکنش ها در یک بازه زمانی:getByDateRange()

4.1.3 کلاس Category

این کلاس نماینده دسته بندی های هزینه ها و درآمدها است.

ویژگی ها:

id: شناسه منحصر به فرد دسته بندی (long)

name: نام دسته بندی (String)

color: رنگ (String)

icon: آیکون (String)

parentId: شناسه دسته بندی والد (long)

type: نوع دسته بندی (categoryType)

userId: شناسه کاربر (long)

isDefault: آیا دسته بندی پیش فرض است (boolean)

createdAt: تاریخ ایجاد (Date)

updatedAt: تاریخ آخرین بروزرسانی (Date)

متد ها:

ایجاد دسته بندی جدید:create()

بروزرسانی دسته بندی:() update

حذف دسته بندی:()delete

دریافت دسته بندی با شناسه:()getById

دریافت تمام دسته بندی های یک کاربر:()getByUser

4.1.4 کلاس Budget

این کلاس نماینده بودجه های تعیین شده برای دسته بندی ها است.

ویژگی ها:

id: شناسه منحصر به فرد بودجه (long)

amount: مبلغ بودجه (double)

period: دوره زمانی (Period)

categoryId: شناسه دسته بندی (long)

userId: شناسه کاربر (long)

alertThreshold: آستانه هشدار (double)

startDate: تاریخ شروع (Date)

endDate: تاریخ پایان (Date)

isRecurring: آیا بودجه تکراری است (boolean)

createdAt: تاریخ ایجاد (Date)

updatedAt: تاریخ آخرین بروزرسانی (Date)

متد ها:

ایجاد بودجه جدید:()create

بروزرسانی بودجه:update()

حذف بودجه:delete()

دریافت بودجه با شناسه:getById()

دریافت تمام بودجه های یک کاربر:getByUser()

دریافت تمام بودجه های یک دسته بندی:getByCategory()

دریافت وضعیت بودجه:getStatus()

4.1.5 کلاس Reminder

این کلاس، نماینده یادآوری های تنظیم شده برای پرداخت ها است.

ویژگی ها:

id: شناسه منحصر به فرد یادآوری (long)

title: عنوان (String)

description: توضیحات (String)

date: تاریخ و زمان یادآوری (Date)

isCompleted: آیا انجام شده است (boolean)

recurringInfo: اطلاعات تکرار (RecurringInfo)

notificationMethods: روش های اعلان (List<NotificationMethods>)

userId: شناسه کاربر (long)

transactionId: شناسه تراکنش مرتبط (long)

createdAt: تاریخ ایجاد (Date)

updatedAt: تاریخ آخرین بروزرسانی (Date)

متد ها:

ایجاد یادآوری جدید:()create

بروزرسانی یادآوری:()update

حذف یادآوری:()delete

دریافت یادآوری با شناسه:()getById

دریافت تمام یادآوری های یک کاربر:()getByUser

دریافت یادآوری ها در یک تاریخ:()getByDate

علامت گذاری به عنوان انجام شده:()markAsCompleted

به تعویق انداختن یادآوری:()snooze

4.1.6 کلاس RecurringInfo

این کلاس نماینده اطلاعات تکرار برای تراکنش ها، بودجه ها و یادآوری ها است.

ویژگی ها:

frequency: فرکانس تکرار (Frequency)

interval: فاصله تکرار (int)

startDate: تاریخ شروع (Date)

endDate: تاریخ پایان (Date)

endDate: روز های هفته (برای تکرار هفتگی) (List<DayOfWeek>)

dayOfMonth: روز ماه (برای تکرار ماهانه) (int)

4.1.7 کلاس NotificationMethod

این کلاس، نماینده روش های اعلان برای یادآوری است.

ویژگی ها:

type: نوع اعلان (NotificationType)

value: مقدار (برای ایمیل، شماره تلفن و غیره (String)

isEnabled: آیا فعال است (boolean)

4.1.8 کلاس Report

این کلاس، نماینده گزارش های مالی است.

ویژگی ها:

id: شناسه منحصر به فرد گزارش (long)

name: نام گزارش (String)

type: نوع گزارش (ReportType)

period: دوره زمانی (Period)

startDate: تاریخ شروع (Date)

endDate: تاریخ پایان (Date)

userId: شناسه کاربر (long)

parameters: پارامتر های گزارش (Mp<String, Object>)

createdAt: تاریخ ایجاد (Date)

id: شناسه منحصر به فرد یادآوری (long)

متد ها:

تولید گزارش:()generate

صدور گزارش:()export

اشتراک گذاری گزارش:()share

4.2 کلاس های سرویس (Service)

4.2.1 کلاس UserService

این کلاس، سرویس های مربوط به کاربران را پیاده سازی می کند.

متد ها:

ثبت نام کاربر جدید:(userDTO)register

احراز هویت کاربر:(credentials)authenticate

بازیابی رمز عبور:(email)resetPassword

بروزرسانی پروفایل کاربر:(userDTO)updateProfile

دریافت کاربر با شناسه:(id)getUserById

4.2.2 کلاس TransactionService

این کلاس، سرویس های مربوط به تراکنش ها را، پیاده سازی میکند.

متد ها:

ایجاد تراکنش جدید:(transactionDTO)createTransaction

بروزرسانی تراکنش:(id, transactionDTO)updateTransaction

حذف تراکنش:(id)deleteTransaction

دریافت تراکنش با شناسه:getTransactionById(id)

دریافت تراکنش های کاربر:getUserTransactions(userid)

دریافت تراکنش های دسته بندی:getCategoryTransactions(categoryId)

دریافت تراکنش ها در بازه زمانی:getTransactionsByDateRange(start, end)

ایجاد تراکنش تکراری:createRecurringTransaction(transactionDTO)

4.2.3 کلاس CategoryService

این کلاس، سرویس های مربوط به دسته بندی ها را، پیاده سازی می کند.

متد ها:

ایجاد دسته بندی جدید:createCategory(categoryDTO)

بروزرسانی دسته بندی:updateCategory(id, categoryDTO)

حذف دسته بندی:deleteCategory(id, replacementId)

دریافت دسته بندی با شناسه:getCategoryById(id)

دریافت دسته بندی های کاربر:getUserCategories(userId)

دریافت دسته بندی های پیش فرض:getDefaultCategories()

4.2.4 کلاس BudgetService

این کلاس، سرویس های مربوط به بودجه ها را پیاده سازی می کند.

متد ها:

ایجاد بودجه جدید:createBudget(budgetDTO)

بروزرسانی بودجه:updateBudget(id, butgetDTO)

حذف بودجه:deleteBudget(id)

دریافت بودجه با شناسه:getBudgetById(id)

دریافت بودجه های کاربر:getUserBudgets(userId)

دریافت بودجه های دسته بندی:getCategoryBudgets(categoryId)

دریافت وضعیت بودجه:getBudgetStatus(id)

برررسی و ارسال هشدار های بودجه:checkBudgetAlerts()

4.2.5 کلاس ReminderService

این کلاس، سرویس های مربوط به یادآوری ها را، پیاده سازی می کند.

متد ها:

ایجاد یادآوری جدید:createReminder(reminderDTO)

بروزرسانی یادآوری:updateReminder(id, reminderDTO)

حذف یادآوری:deleteReminder(id)

دریافت یادآوری با شناسه:getReminderById(id)

دریافت یادآوری های کاربر:getUserReminders(userId)

دریافت یادآوری های آینده:getUpcomingReminders(userId)

علامت گذاری به عنوان انجام شده:markAsCompleted(id)

به تعویق انداختن یادآوری:snoozeReminder(id, duration)

پردازش یادآوری های زمان بندی شده:proccessScheduledReminders()

4.2.6 کلاس ReportService

این کلاس، سرویس های مربوط به گزارش ها را، پیاده سازی می کند.

متد ها:

دریافت گزارش کلی:getOverallReport(userId, period)

دریافت گزارش دسته بندی:getCategoryReport(userId, period)

دریافت گزارش مقایسه هزینه و درآمد:getExpenseVsIncomeReport(userId, period)

دریافت گزارش بودجه:getBudgetReport(userId, period)

دریافت گزارش روند:getTrendReport(userId, period)

صدور گزارش:exportReport(reported, format)

اشتراک گذاری گزارش:shareReport(reported, method)

4.2.7 کلاس SynchronizationService

این کلاس، سرویس های مربوط به همگام سازی را، پیاده سازی می کند

متد ها:

همگام سازی داده ها:syncData(userId, deviceId)

دریافت زمان آخرین همگام سازی:getLastSyncTime(userId, deviceId)

حل تعارضات:resolveConflicts(conflicts)

آپلود داده ها:uploadDate(userId, data)

دانلود داده ها:downloadDate(userId)

4.2.8 کلاس BackupService

این کلاس، سرویس های مربوط به پشتیبان گیری را، پیاده سازی می کند.

متد ها:

ایجاد نسخه پشتیبان:createBackup(userId, strategy)

بازیابی از نسخه پشتیبان:restoreBackup(userId, bachupId)

دریافت لیست نسخه های پشتیبان:getBackupList(userId)

حذف نسخه پشتیبان:deleteBackup(backupId)

زمان بندی پشتیبان گیری خودکار:scheduleBackup(userId, schedule)

4.2.9 کلاس NotificationService

این کلاس، سرویس های مربوط به اعلان ها را، پیاده سازی می کند.

متد ها:

ارسال اعلان:sendNotification(userId, notification)

ثبت دستگاه برای دریافت اعلان:registerDevice(userId, deviceToken)

لغو ثبت دستگاه:unregisterDevice (deviceToken)

دریافت ترجیحات اعلان کاربر:getUserNotificationPreferences(userId)

بروزرسانی ترجیحات اعلان:updateNotificationPreferences(userId, preferences)

4.3 کلاس های کنترلر (Controller)

4.3.1 کلاس UserController

این کلاس، API های مربوط به کاربران را، پیاده سازی می کند.

متد ها:

ثبت نام کاربر جدید:()register

ورود کاربر:()login

بازیابی رمز عبور:()resetPassword

بروزرسانی پروفایل:()updateProfile

دریافت پروفایل کاربر:()getUserProfile

4.3.2 کلاس TransactionController

این کلاس، API های مربوط به تراکنش ها را، پیاده سازی می کند.

متد ها:

ایجاد تراکنش جدید:()createTransaction

بروزرسانی تراکنش:()updateTransaction

حذف تراکنش:()deleteTransaction

دریافت تراکنش با شناسه:()getTransactionById

دریافت تراکنش های کاربر:()getUserTransactions

دریافت تراکنش های دسته بندی:()getCategoryTransactions

دریافت تراکنش ها در بازه زمانی:()getTransactionsByDateRange

آپلود تصویر رسید:()uploadImage

4.3.3 کلاس CategoryController

این کلاس، API های مربوط به دسته بندی ها را، پیاده سازی می کند.

متد ها:

ایجاد دسته بندی جدید:()createCategory

بروزرسانی دسته بندی:()updateCategory

حذف دسته بندی:()deleteCategory

دریافت دسته بندی با شناسه:()getCategoryById

دریافت دسته بندی های کاربر:()getUserCategories

دریافت دسته بندی های پیش فرض:()getDefaultCategories

4.3.4 کلاس BudgetController

این کلاس، API های مربوط به بودجه ها را، پیاده سازی می کند.

متد ها:

ایجاد بودجه جدید:()createBudget

بروزرسانی بودجه:()updateBudget

حذف بودجه:()deleteBudget

دریافت بودجه با شناسه:()getBudgetById

دریافت بودجه های کاربر:()getUserBudgets

دریافت بودجه های دسته بندی:()getCategoryBudgets

دریافت وضعیت بودجه:()getBudgetStatus

4.3.5 کلاس ReminderController

این کلاس، API های مربوط به یادآوری ها را، پیاده سازی می کند.

متد ها:

ایجاد یادآوری جدید:()createReminder

بروزرسانی یادآوری:()updateReminder

حذف یادآوری:()deleteReminder

دریافت یادآوری با شناسه:()getReminderById

دریافت یادآوری های کاربر:()getUserReminders

دریافت یادآوری های آینده:()getUpcomingReminders

علامت گذاری به عنوان انجام شده:()markAsCompleted

به تعویق انداختن یادآوری:()snoozeReminder

4.3.6 کلاس ReportController

این کلاس، API های مربوط به گزارش ها را، پیاده سازی می کند.

متد ها:

دریافت گزارش کلی:()getOverallReport

دریافت گزارش دسته بندی:()getCategoryReport

دریافت گزارش مقایسه هزینه و درآمد:()getExpenseVsIncomeReport

دریافت گزارش بودجه:()getBudgetReport

دریافت گزارش روند:()getTrendReport

صدور گزارش:()exportReport

اشتراک گذاری گزارش:()shareReport

4.3.7 کلاس SynchronizationController

این کلاس، API های مربوط به همگام سازی را، پیاده سازی می کند.

متد ها:

همگام سازی داده:()syncData

زمان دریافت آخرین همگام سازی:()getLastSyncTime

آپلود داده ها:()uploadData

دانلود داده ها:()dowloadData

4.3.8 کلاس BackupController

این کلاس، API های مربوط به پشتیبان گیری را، پیاده سازی می کند.

متد ها:

ایجاد نسخه پشتیبان:()createBackup

بازیابی از نسخه پشتیبان:()restoreBackup

دریافت لیست نسخه های پشتیبان:()getBackupList

حذف نسخه پشتیبان:()deleteBackup

زمان بندی پشتیبان گیری خودکار:()scheduleBackup

4.4 کلاس های کمکی (Utility)

4.4.1 کلاس DateUtils

این کلاس، توابع کمکی مرتبط با تاریخ و زمان را، پیاده سازی می کند.

متد ها:

قالب بندی تاریخ:()formatDate

تجزیه تاریخ:()parseDate

دریافت ابتدای روز:()getStartOfDay

دریافت انتهای روز:()getEndOfDay

دریافت ابتدای هفته:()getStartOfWeek

دریافت انتهای هفته:()getEndOfWeek

دریافت ابتدای ماه:()getStartOfMonth

دریافت انتهای ماه:()getEndOfMonth

دریافت ابتدای سال:()getStartOfYear

دریافت انتهای سال:()getEndOfYear

محاسبه رخداد بعدی (برای تکرار):()calculateNextOccurrence

4.4.2 کلاس ValidationUtils

این کلاس، توابع کمکی مرتبط با اعتبارسنجی داده ها را، پیاده سازی می کند.

متد ها:

اعتبارسنجی ایمیل:()validateEmail

اعتبارسنجی رمز عبور:()validatePassword

اعتبارسنجی شماره تلفن:validatePhone()

اعتبارسنجی مبلغ:validateAmount()

اعتبارسنجی تاریخ:validateDate()

4.4.3 کلاس SecurityUtils

این کلاس، توابع کمکی مرتبط با امنیت را، پیاده سازی می کند.

متد ها:

رمزنگاری رمز عبور:hashPassword()

تایید رمز عبور:verifyPassword()

تولید توکن:genderateToken()

تایید توکن:verifyToken()

رمزنگاری داده:encrypt()

رمزگشایی داده:decrypt()

4.4.4 کلاس FileUtils

این کلاس، توابع کمکی مرتبط با مدیریت فایل را، پیاده سازی می کند.

متد ها:

آپلود فایل:uploadFile()

دانلود فایل:dowloadFile()

حذف فایل:deleteFile()

دریافت اندازه فایل:getFileSize()

اعتبارسنجی نوع فایل:()validateFileType

تولید نام فایل:()genderateFileName

4.4.5 کلاس NotificationUtils

این کلاس، توابع کمکی مرتبط با اعلان ها را، پیاده سازی می کند.

متد ها:

ارسال ایمیل:()sendEmail

ارسال اعلان:()sendPushNotification

ارسال پیامک:()sendSMS

زمان بندی اعلان:()scheduleNotification

5. روابط بین کلاس ها

5.1 روابط وراثت

در سیستم ردیابی و مدیریت هزینه، رابطه وراثت در موارد زیر، استفاده می شود:

کلاس TransactionDTO: از کلاس اصلی DTO، ارث می برد.

کلاس IncomeDTO و ExpenseDTO: از کلاس اصلی DTO، ارث می برد.

کلاس BaseEntity: کلاس پایه برای تمام موجودیت ها است که ویژگی های مشترک مانند createdAt, id و updatedAt را، تعریف می کند.

کلاس BaseController: کلاس پایه برای تمام کنترلر ها است که متد های مشترک را، تعریف می کند.

کلاس BaseService: کلاس پایه برای تمام سرویس ها است که متد های مشترک را تعریف می کند.

5.2 روابط ترکیب و تجمیع

در سیستم ردیابی و مدیریت هزینه، روابط ترکیب و تجمیع در موارد زیر، استفاده شده است:

رابطه بین Transaction و User :

یک کاربر می تواند چندین تراکنش، داشته باشد (تجمع، یک-به-چند).

رابطه بین Category و User :

یک کاربر می تواند چندین دسته بندی، داشته باشد (تجمع، یک-به-چند).

رابطه بین Budget و User :

یک کاربر می تواند چندین بودجه، داشته باشد (تجمع، یک-به-چند).

رابطه بین Reminder و User :

یک کاربر می تواند چندین یادآوری، داشته باشد (تجمع، یک-به-چند).

رابطه بین Category و Transaction :

یک دسته بندی می تواند چندین تراکنش، داشته باشد (تجمع، یک-به-چند).

رابطه بین Category و Budget :

یک دسته بندی می تواند چندین بودجه، داشته باشد (تجمع، یک-به-چند).

رابطه بین RecurringInfo و Transaction :

یک تراکنش می تواند یک RecurringInfo، داشته باشد (ترکیب، یک-به-یک).

رابطه بین RecurringInfo و Reminder :

یک یادآوری می تواند یک RecurringInfo، داشته باشد (ترکیب، یک-به-یک).

رابطه بین Reminder و NotificationMethod :

یک یادآوری می تواند چندین روش اعلان، داشته باشد (ترکیب، یک به چند)

رابطه بین Category و Category :

یک دسته بندی می تواند یک دسته بندی والد، داشته باشد و چندین زیردسته بندی، داشته باشد (تجمع، خود ارجاعی)

داشته باشد (تجمع، خود-ارجاعی)

5.3 روابط وابستگی

در سیستم ردیابی و مدیریت هزینه، روابط وابستگی در موارد زیر، استفاده شده است:

رابطه بین کنترلر ها و سرویس ها:

کنترلر ها از سرویس ها، استفاده می کنند.

رابطه بین سرویس ها و مخازن (Repositpry):

سرویس ها زا مجازن برای دسترسی به داده ها، استفاده می کنند.

رابطه بین سرویس ها و کلاس های کمکی:

سرویس ها زا کلاس های کمکی برای عملیت مشترک، استفاده می کنند.

رابطه بین ReminderService و ReminderService :

NotificationService از NotificationService برای ارسال اعلان ها، استفاده می کند.

رابطه بین BudgetService و BudgetService :

TransactionService از TransactionService برای محاسبه وضعیت بودجه، استفاده می کند.

رابطه بین ReportService و ReportService :

TransactionService از TransactionService برای دریافت داده های لازم برای گزارش ها، استفاده می کند.

رابطه بین SynchronizationService و سایر سرویس ها :

SynchronizationService از سایر سرویس ها برای همگام سازی داده ها، استفاده می کند.

6. ضمائم

6.1 انواع شمارشی (Enumerations)

6.1.1 TransactionType

هزینه: EXPENSE

درآمد: INCOME

6.1.2 CategoryType

دسته بندی هزینه: EXPENSE

دسته بندی درآمد: INCOME

هردو: BOTH

6.1.3 Period

روزانه: DAILY

هفتگی: WEEKLY

ماهانه: MONTHLY

سه ماهه: QUARTERLY

سالانه: YEARLY

Frequency 6.1.4

روزانه: DAILY

هفتگی: WEEKLY

ماهانه: MONTHLY

سالانه: YEARLY

NotificationType 6.1.5

ایمیل: EMAIL

اعلان push: PUSH

پیامک: SMS

درون برنامه ای: IN_APP

ReportType 6.1.6

کلی: OVERALL

دسته بندی: CATEGORY

مقایسه هزینه و درآمد: EXPENSE_VS_INCOME

بودجه: BUDGET

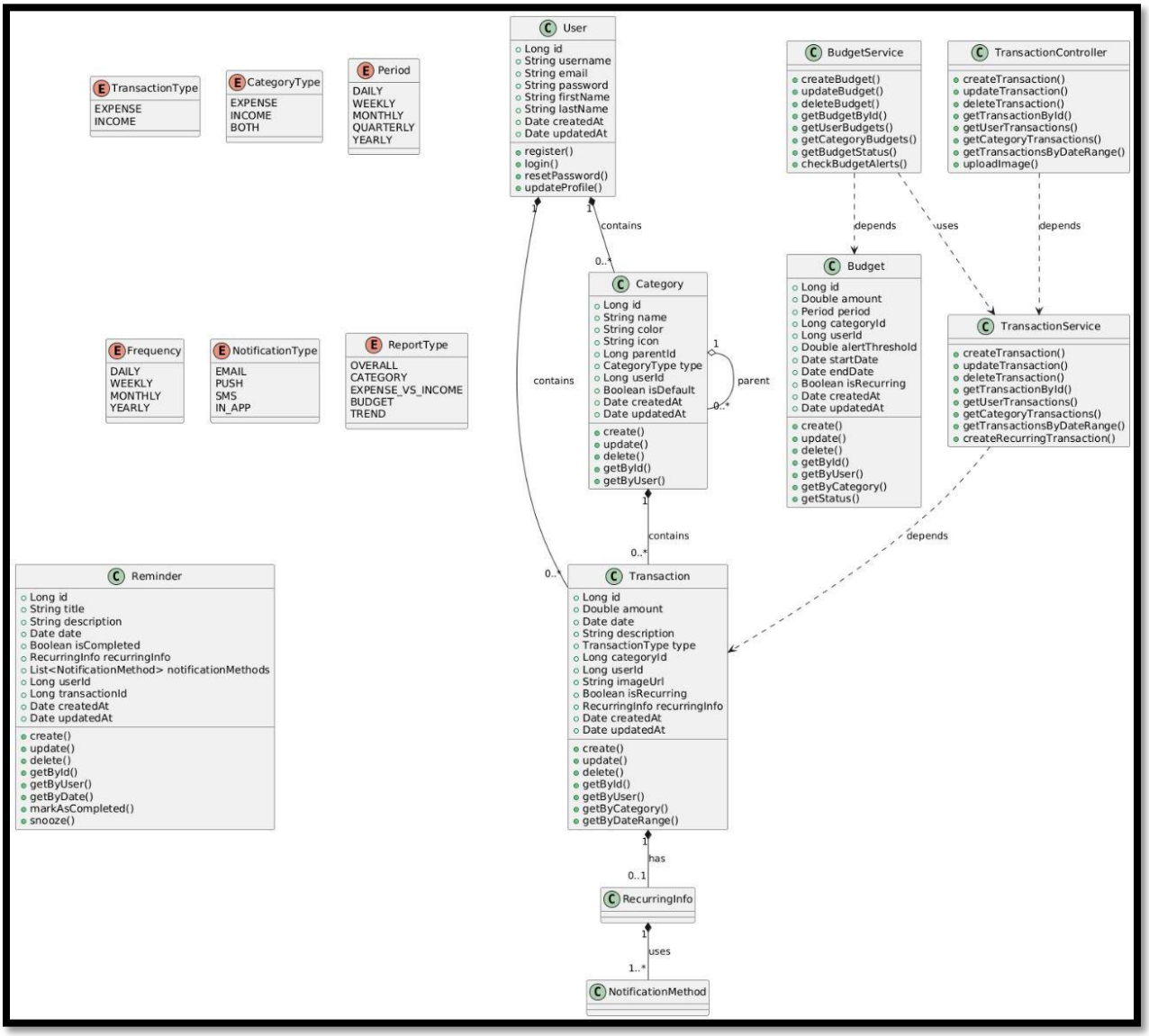
روند: TREND

6.2 جداول پایگاه داده

[اینجا جدول پایگاه داده، قرار می گیرد – به عنوان جدول 1]

6.3 نمودار ER

[اینجا نمودار ER قرار می گیرد – به عنوان نمودار 5]# مستند مدلسازی کلاس ها


TransactionType
EXPENSE
INCOME
CategoryType
EXPENSE
INCOME
BOTH
Period
DAILY
WEEKLY
MONTHLY
QUARTERLY
YEARLY
User
Long id
String username
String email
String password
String firstName
String lastName
Date createdAt
Date updatedAt
register()
login()
resetPassword()
updateProfile()
BudgetService
createBudget()
updateBudget()
deleteBudget()
getBudgetById()
getUserBudgets()
getCategoryBudgets()
getBudgetStatus()
checkBudgetAlerts()
TransactionController
createTransaction()
updateTransaction()
deleteTransaction()
getTransactionById()
getUserTransactions()
getCategoryTransactions()
getTransactionsByDateRange()
uploadImage()
contains
depends
uses
depends
Frequency
DAILY
WEEKLY
MONTHLY
YEARLY
NotificationType
EMAIL
PUSH
SMS
IN_APP
ReportType
OVERALL
CATEGORY
EXPENSE_VS_INCOME
BUDGET
TREND
Category
Long id
String name
String color
String icon
Long parentId
CategoryType type
Long userId
Boolean isDefault
Date createdAt
Date updatedAt
create()
update()
delete()
getById()
getByUser()
parent
contains
Budget
Long id
Double amount
Period period
Long categoryId
Long userId
Double alertThreshold
Date startDate
Date endDate
Boolean isRecurring
Date createdAt
Date updatedAt
create()
update()
delete()
getById()
getByUser()
getByCategory()
getStatus()
TransactionService
createTransaction()
updateTransaction()
deleteTransaction()
getTransactionById()
getUserTransactions()
getCategoryTransactions()
getTransactionsByDateRange()
createRecurringTransaction()
contains
depends
Reminder
Long id
String title
String description
Date date
Boolean isCompleted
RecurringInfo recurringInfo
List<NotificationMethod> notificationMethods
Long userId
Long transactionId
Date createdAt
Date updatedAt
create()
update()
delete()
getById()
getByUser()
getByDate()
markAsCompleted()
snooze()
Transaction
Long id
Double amount
Date date
String description
TransactionType type
Long categoryId
Long userId
String imageUrl
Boolean isRecurring
RecurringInfo recurringInfo
Date createdAt
Date updatedAt
create()
update()
delete()
getById()
getByUser()
getByCategory()
getByDateRange()
has
RecurringInfo
uses
NotificationMethod