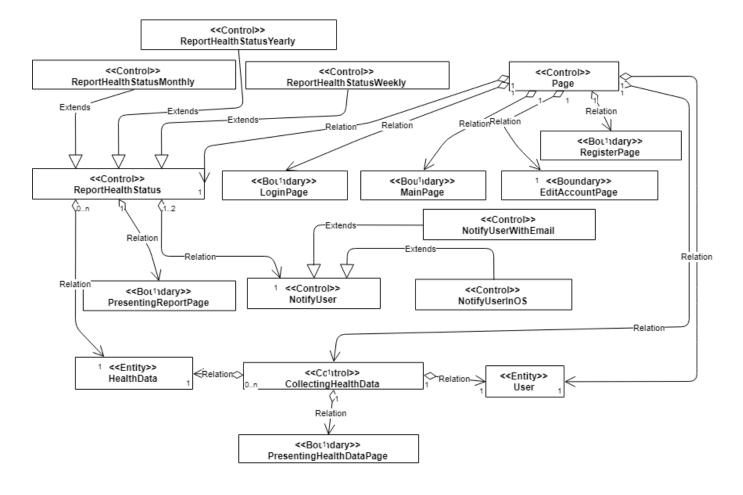
#### Class Diagram- Personal Health Monitoring System

نمودار کلاس در شکل زیر دیده میشود. جزئیات هر کلاس در ادامه آمده است:



#### جزئيات كلاسها:

## <<Boundary>> EditAccountPage + name: Textbox + family: Textbox + email: Textbox

- + password: Textbox + submit: Button
- <<Boundary>> RegisterPage
- + username: Textbox
- + email: Textbox
- + password: Textbox
- + submit: Button
- + draw(): void

## <<Control>> Page

- + presentHealthData(): void
- + reportHealthStatus(): void
- + login(): void
- + register(): void
- + editAccount(): void
- + deleteAcount(): void

#### <<Control>> ReportHealthStatus

- healthData: HealthData[]
- startDate: int

+ draw(): void

- endDate: int
- findData(): void # generateReport(): void # generateStatus(): void
- + setDate(int, int): void

#### <<Boundary>> LoginPage

- + username: Textbox
- + password: Textbox
- + submit: Button
- + draw(): void

### <<Boundary>> MainPage

- + healthData: Button
- + healthStatus: Button
- + editAccount: Button
- + deleteAccount: Button
- + draw(): void

#### <<Boundary>> PresentingReportPage

- healthStatus: String
- analysisResult: int[]
- + startDate: Textbox
- + endDate: Textbox
- + submit: Button
- + setHealthStatus(Sring): void
- + setAnalysisResult(int[]): void
- + drawGraph(): void
- + drawText(): void
- + draw(): void

## <<Control>> ReportHealth Status Monthly

- + generateReport(): void
- + generateStatus(): void

#### <<Control>> NotifyUser

- message: String
- + setMessage(String): void # notify(): void

## <<Control>> NotifyUserInOS

+ notify(): void

## <<Control>> ReportHealth Status Weekly

- + generateReport(): void
- + generateStatus(): void

## <<Control>> ReportHealthStatusYearly

- + generateReport(): void
- + generateStatus(): void

## <<Control>> NotifyUserWithEmail

+ notify(): void

#### Class Diagram- Personal Health Monitoring System

# <<Bot¹ndary>> PresentingHealthDataPage

- healthData: HealthData[]
- + startDate: Textbox
- + endDate: Textbox
- + submit: Button
- + drawGraph(): void
- + drawText(): void
- + getHeathData(): HealthData[]
- + setHeathData(HealthData[]): void
- + draw(): void

# <<Entity>> HealthData

- + heartBit: int
- + bloodPressure: int
- + bodyTemperature: int
- + stepCount: int
- + burnedCalory: int
- + date: int
- + getDate(): int
- + setDate(int): void
- + getHeartBit(): int
- + setHeartBit(int): void
- + getBloodPressure():int
- + setBloodPressure(int):void
- + getBodyTemperature():int
- + setBodyTemperature(int):void
- + getStepCount(): int
- + setStepCount(int): void
- + getBurnedCalory(): int
- + setBurnedCalory(int): void

# <Control>> CollectingHealthData

- startDate: int
- endDate: int
- healthData: HealthData[]
- + setDate(int, int): void
- findData(): void
- + presentData(): void
- + syncingWithWearableDevice(): void
- + syncingWithServer(): void

# <<Entity>> User

- + name: String
- + family: String
- + email: String
- + username: String
- password: String
- sessionTiken: String
- + create(String): Bool
- + edit(String): Bool
- + delete(String): Bool
- + login(String): Bool

Entities	
کاربری که از سیستم استفاده می کند. اطلاعات فردی مثل نام و نام خانوادگی، نام	User
کاربری و غیره در این کلاس وجود دارد.	
اطلاعات حیاتی فرد مثل ضربان قلب، فشار خون، دمای بدن و غیره.	HealthData

Boundaries	
این کلاس فرم ورود به برنامه است. کاربر با وارد کردن نام کاربری و گذرواژه و فشردن دکمه، میتواند وارد برنامه شود.	LoginPage
این کلاس برای ثبت نام در برنامه است.	RegisterPage
این کلاس برای ویرایش حساب کاربری است.	EditAccountPage
صفحه اصلی برنامه است که دکمههای مختلفی از جمله، حذف و ویرایش حساب کاربری، مشاهده دادههای حیاتی، همگامسازی با سرور و ابزار پوشیدنی و همچنین گزارش گیری، در این کلاس وجود دارد.	MainPage
کلاس مربوط به صفحه نمایش خروجی گزارش گیری است که در آن خروجی در یک نمودار نمایش داده می شود و وضعیت سلامت کاربر در قالب یک متن نمایش داده می شود.	<b>PresentingReportPage</b>
کلاس مربوط به صفحه نمایش اطلاعات حیاتی فرد است که این اطلاعات به صورت متن و نمودار نمایش داده میشود.	PresentingHealthDataPage

Control	
این کلاس کنترل کننده صفحات مختلف در برنامه است.	Page
این کلاس وضعیت سلامت کاربر را در بازه زمانی مشخص محاسب <b>Repo</b>	rtHealthStatus
این کلاس وضعیت سلامت کاربر را به صورت هفتگی محاسبه <b>ReportH</b>	ealthStatusWeekly
این کلاس وضعیت سلامت کاربر را به صورت ماهانه محاسبه می کن	ealthStatusMonthly
ReportH این کلاس وضعیت سلامت کاربر را به صورت سالانه محاسبه می کن	lealthStatusYearly
در صورتی که نتیجه گزارش گیری نشان دهنده وجود خطر برای کار آ باشد، این کلاس هشدار لازم را به صورت روزانه به کاربر می دهد.	NotifyUser

## Class Diagram- Personal Health Monitoring System

NotifyUserWithEmail	از این کلاس برای ایجاد هشدار به کاربر ا از طریق ایمیل استفاده میشود. این کلاس از NotifyUser ارثبری دارد.
NotifyUserInOS	از این کلاس برای ایجاد هشدار به کاربر در سیستم عامل اندروید استفاده میشود. این کلاس از NotifyUser ارثبری دارد.
CollectingHealthData	در این کلاس اطلاعات کاربر به منظور نمایش از ابزار پوشیدنی و سرور دریافت و به ذخیرهسازی به سرور ارسال میشود.