

Software Architecture Document

Swapnil Srivastava, Alex Thropp, and Jonas Schulz

Architectural Representation

The architecture in this document will be described using technical memos and architectural views, and well as plain english for those who are less technical.

Architectural Factors

Vision:

For this project we will be designing a piece of wearable technology to track and report the users vitals and health. The system has a few hardware requirements such as it needs to be wearable as some sort of band or watch. Second, it needs to have the necessary sensors to keep track of the users vitals such as their heart rate and oxygen levels. It should also have the correct tools to sense things such as movement and proximity to the ground (gyroscope). Along with the base hardware requirements there are also needs to be internet and bluetooth capable. This will require the correct hardware and software to be successfully implemented. The user interface also needs to be simple and efficient and allow the user to do things such as manipulate settings, keep track/view health and goals, as well, the UI needs to allow the user to do basic things such as view time, date, weather, calendar. If all of these can be successfully implemented we should be able to design and produce a competitive product for the wearable technology world.

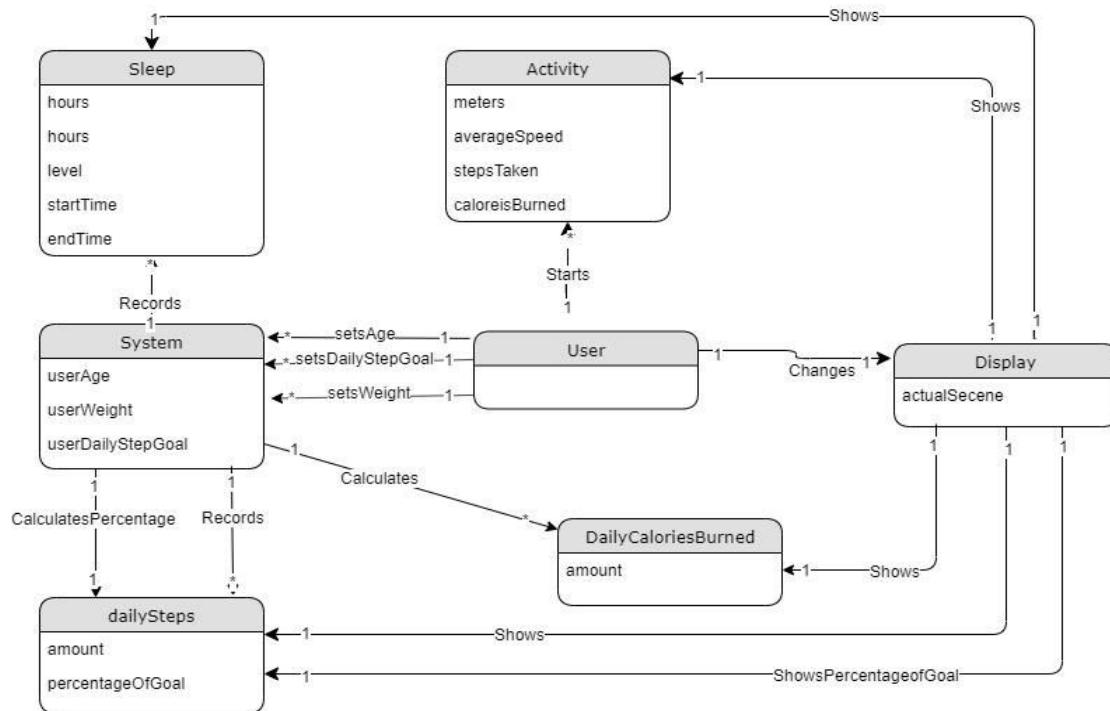
Supplemental Specs

- Bluetooth 5.0
- Latest Gyroscope
- IOS 10 or above (for mobile app connectivity)
- Android Oreo or above (for mobile app connectivity)
- User Agreement

Architectural Decisions

Logical View (Package, Class, and Domain Diagrams)

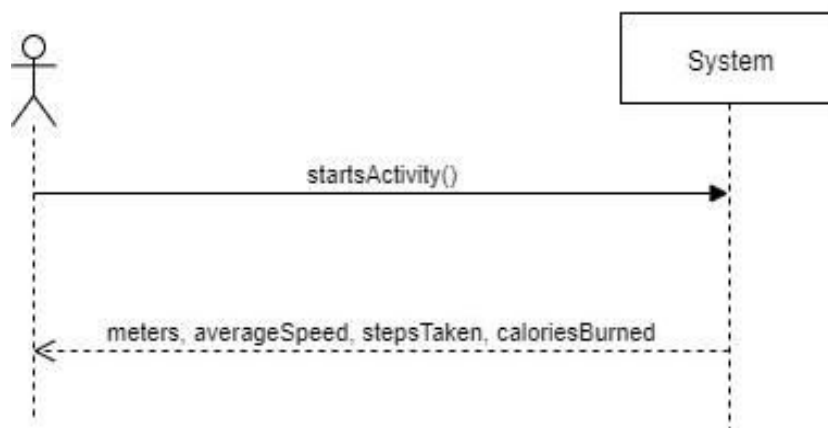
Domain Model Diagram:

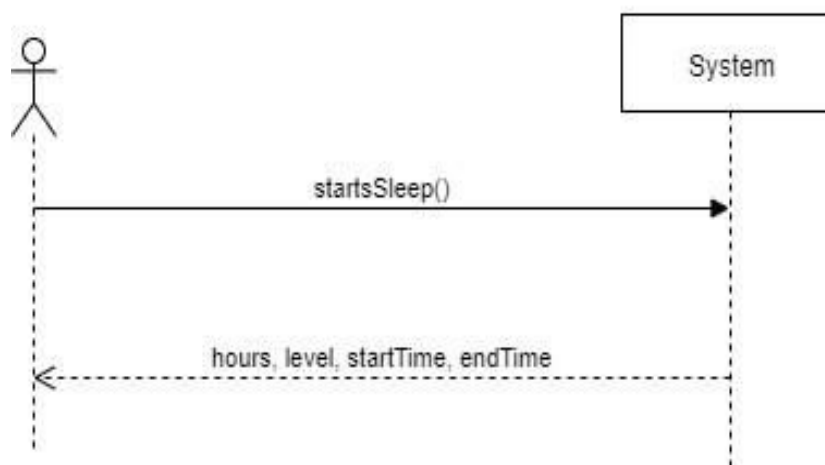
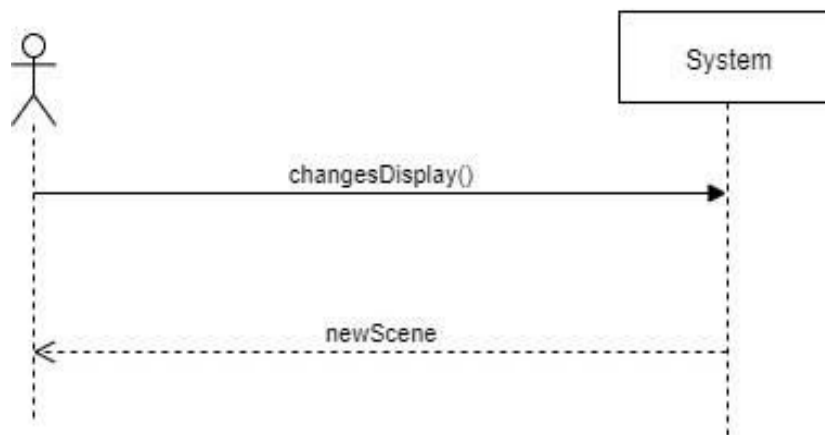
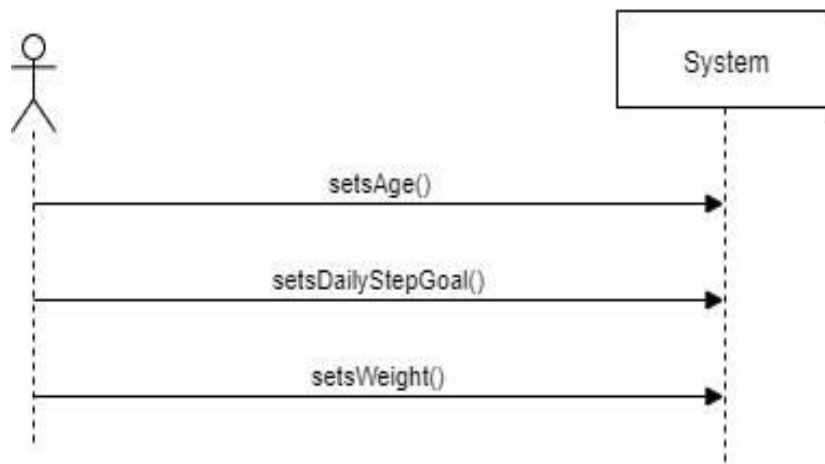


Deployment View (Deployment Diagrams)

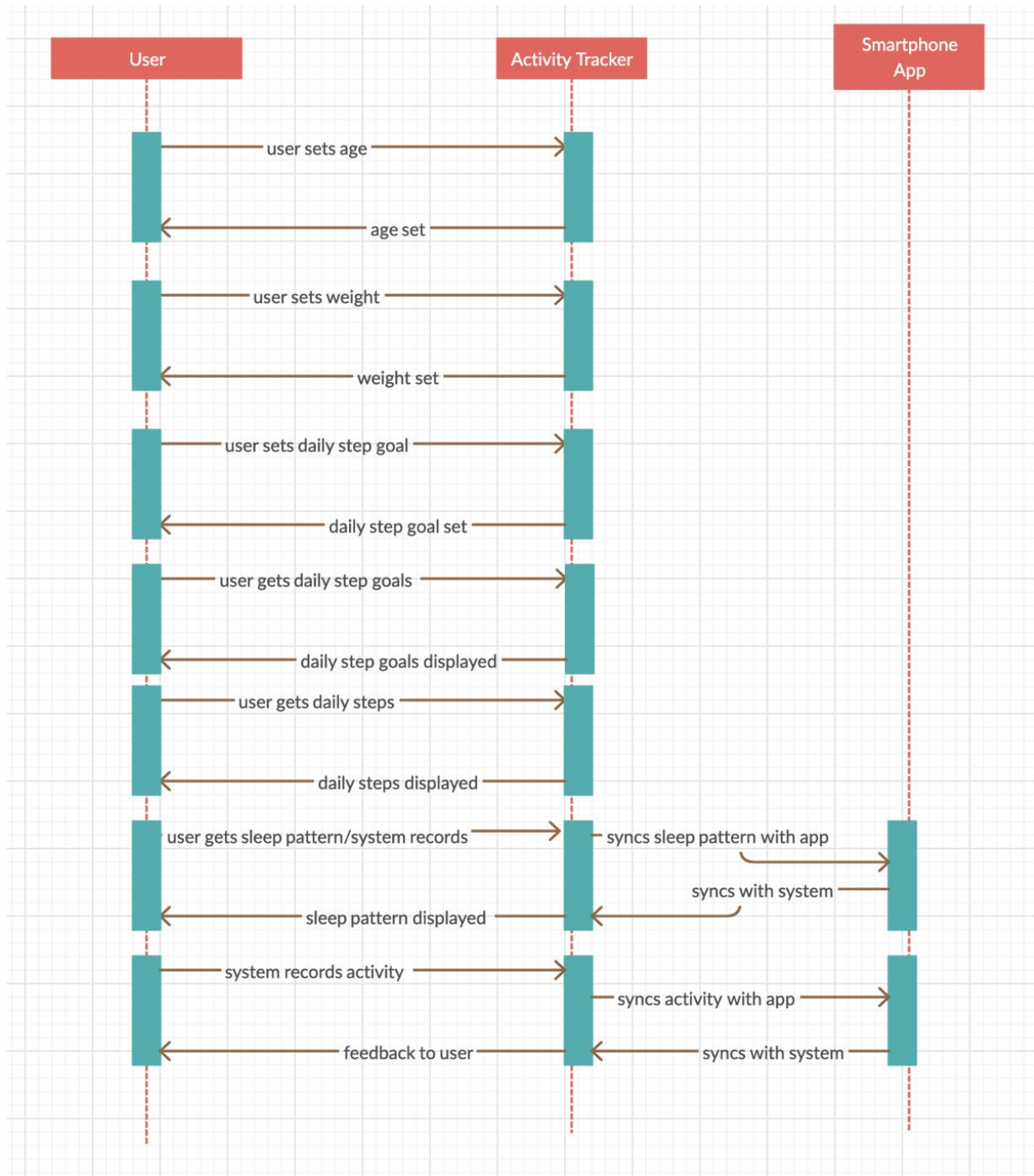
Process View (Class and interaction Diagrams)

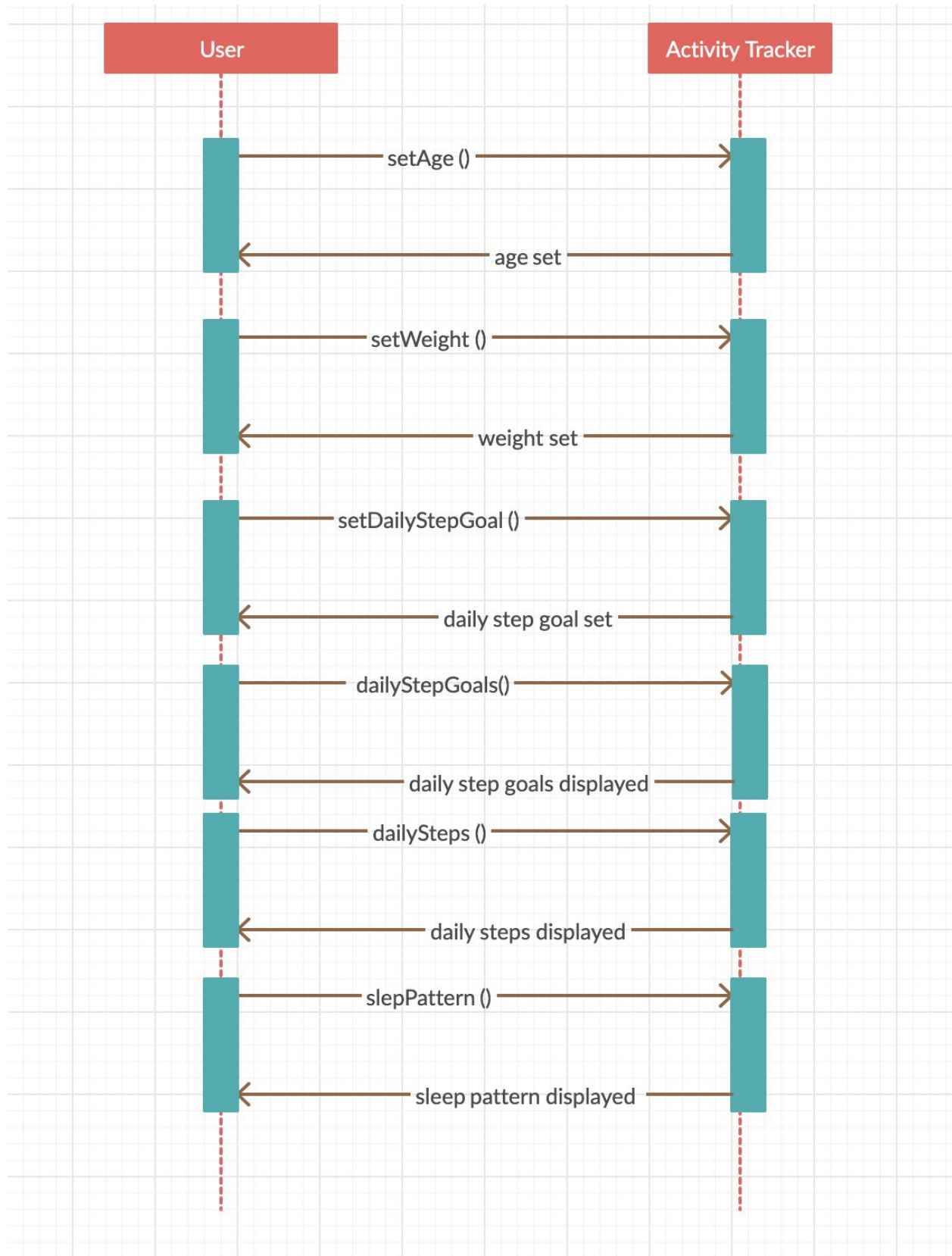
System Sequence Diagrams:





Sequence Diagram:





GRASP patterns and responsibilities of objects

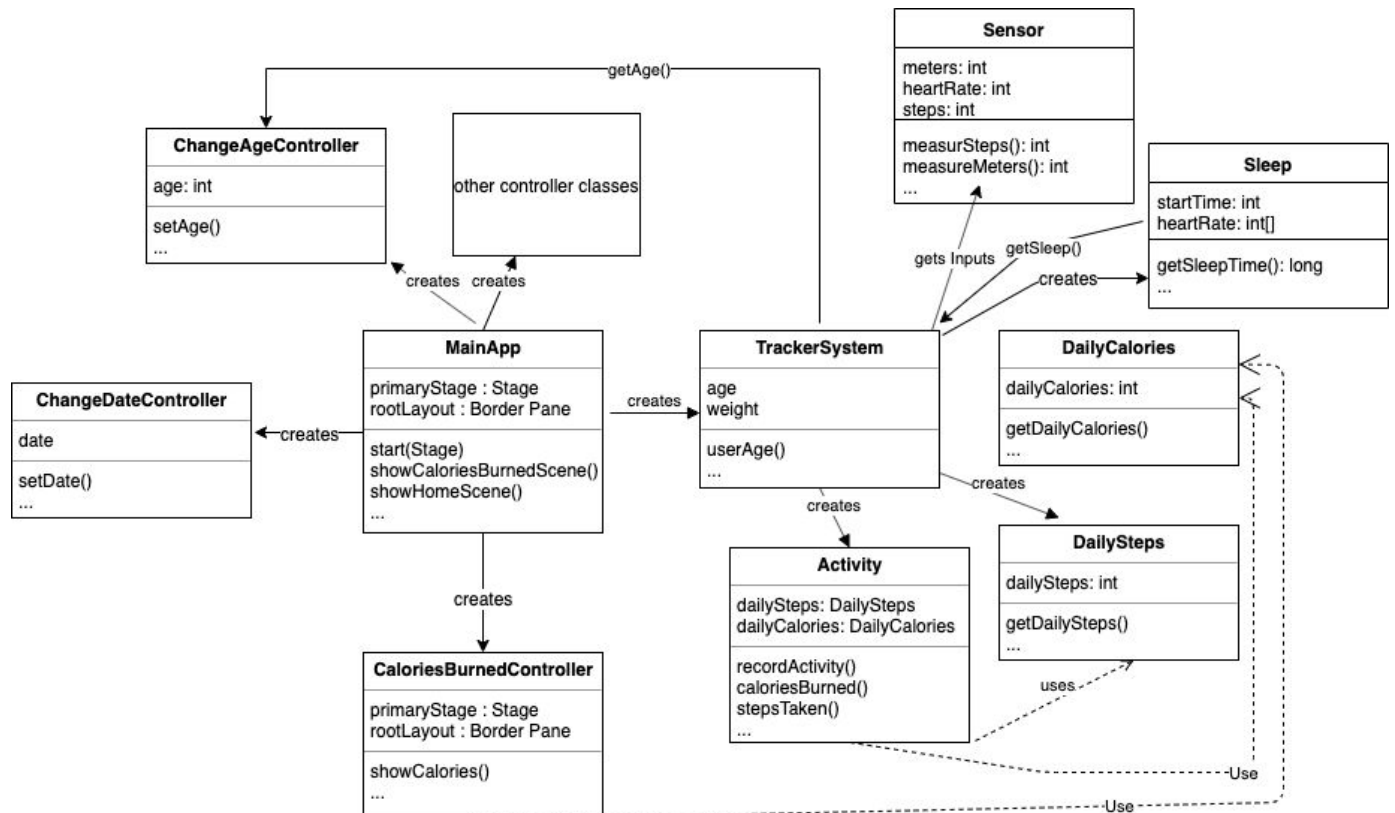
Class	Responsibility	GRASP pattern
MainApp	Starts the program, creates controllers and System, and controls the scene flow	Creator, controller, and pure fabrication
Activity	Saves data about activity	Information expert, low coupling
DailyCalorie	Saves calorie data	Information expert
DailySteps	Saves steps data	Information expert
CaloriesBurnedController	Controls the calorie burned scene	Controller
System	manages activity and user data	Information expert and pure fabrication
ChangeAgeController	Controls change age scene	controller
ChangeDateController	Controls change date scene	controller
Sleep	Saves the sleep information	Information expert

- The system acts as the controller
- The System is responsible for knowing date, time, step count, calories burned, and all other user created instances
- The user is responsible for creating instances of age, weight, step goal, and calorie goal
- The user is responsible for changing the display
- Our System is designed with low cohesion in mind, most classes are independent with little interaction, except basic variable and object handling.

Gang of four patterns:

Class	Responsibility	GoF pattern
MainApp	Starts the program, creates controllers and System, and controls the scene flow	Factory and adapter
Activity	Saves data about activity	Observer and strategy
DailyCalorie	Saves calorie data	Observer
DailySteps	Saves steps data	Observer
CaloriesBurnedController	Controls the calorie burned scene	Facade
TrackerSystem	manages activity and user data	Factory, observer, and facade
ChangeAgeController	Controls change age scene	Facade
ChangeDateController	Controls change date scene	Facade
Sleep	Saves the sleep information	Observer

Class diagram:



Use Case View

Use case 1

Use Case: Recording activity and health

Scope: Activity tracker

Level: User goal

Primary Actor: User of activity tracker.

Stakeholders and Interests:

- User: Wants to record their activity accurately and with ease.

Preconditions: The activity tracker is paired with a phone.

Success Guarantee: Activity is recorded and saved accurately. Heart Rate is monitored and recorded accurately.

Main Success Scenario:

1. User wears the activity tracker
2. The tracker monitors and reports the heart rate of the user.
3. Tracker records the number of steps that the user takes

4. Tracker estimates the number of calories burned by the user based on the number of steps and heart rate.
5. When user sleeps, tracker monitors the quality of user's sleep.
6. Tracker saves the day's recorded activity and reports the activity app on paired phone.

Extensions (Alternative scenarios)

If the watch is not paired before use

1. Watch saves activity and health data for up to 1 week
2. If the phone is paired in that week, the data is shared with the phone
3. If the tracker is not paired for over 1 week, then it only saves data of the past 7 days.

Special Requirements:

- Wearable must have large text and buttons.
- The UI should be clean and easy to interact with
- The activity feedback should be in real time.

Technology and Data Variations: Wearable turned off and on by pressing and holding side button for 3 seconds.

Frequency of Occurrence: Continuous

Miscellaneous Such as open issues: Transferring data from the wearable to the smartphone app.

Use case 2

Use Case: Monitors sleep pattern

Scope: Activity tracker

Level: System goal

Primary Actor: System of activity tracker

Stakeholders and Interests:

- User: Wants to record their activity accurately and with ease.

Preconditions: The activity tracker is paired with a phone.

Success Guarantee: Sleep activity is recorded and saved accurately. Heart Rate is monitored and recorded accurately.

Main Success Scenario:

1. The user falls asleep wearing his activity tracker (that is connected to a smartphone).
2. The phone recognizes the position and location of watch using the gyroscope (to know if user is asleep).
3. The system actively monitors heart rate, O2 levels, and movement during sleep.

4. System publishes daily report of sleep to phone for user to view

Special Requirements:

- Wearable must be on for sleep.
- The sleep activity feedback should be available in real time.

Frequency of Occurrence: Nightly

Miscellaneous Such as open issues: Transferring data from the wearable to the smartphone app.

Use case 3

Use Case: Use wants to view date/time from watch face

Scope: Activity tracker

Level: User goal

Primary Actor: User of activity tracker.

Stakeholders and Interests:

- User: Wants to view information accurately and with ease.

Preconditions: The activity tracker is paired with a phone.

Success Guarantee: Date and time is viewed from watch face.

Main Success Scenario:

1. The user raises the watch to viewing height
2. The user turns on the display.
3. Using the gyroscope the system will look to see if the watched is angled correctly (left to right)
4. The system will display the date and time to the user on the from of the watch face

Special Requirements:

- Wearable must have large text and buttons.
- The UI should be clean and easy to interact with
- The activity feedback should be in real time.

Frequency of Occurrence: Continuous

Miscellaneous Such as open issues: Transferring data from the wearable to the smartphone app.

Use case 4

Use Case: view steps

Scope: Activity tracker

Level: User goal

Primary Actor: User of activity tracker.

Stakeholders and Interests:

- User: Wants to view their activity accurately and with ease.

Preconditions: The activity tracker is paired with a phone.

Success Guarantee: step count viewed on face of watch

Main Success Scenario:

1. The user raises the watch to viewing height
2. The users stick out their arm manipulating the watch to an up to down position. (checked by sys gyroscope)
3. The users turns on the display
4. The step count is displayed on the face of the watch

Special Requirements:

- Wearable must have large text and buttons.
- The UI should be clean and easy to interact with
- The activity feedback should be in real time.

Frequency of Occurrence: Continuous

Miscellaneous Such as open issues: Transferring data from the wearable to the smartphone app.

Use case 5

Use Case: the user wants to access/change the settings

Scope: Activity tracker

Level: User goal

Primary Actor: User of activity tracker.

Stakeholders and Interests:

- User: access or change settings from the watch

Success Guarantee: user reaches settings and can change desired setting.

Main Success Scenario:

1. User turns on the display
2. The user goes to the home screen and selects the settings button
3. Once in the settings menu the user can choose to change date/time as well as the units of measure etc.

Special Requirements:

- Wearable must have large text and buttons.
- The UI should be clean and easy to interact with
- The activity feedback should be in real time.

Frequency of Occurrence: daily

Miscellaneous Such as open issues: Transferring data from the wearable to the smartphone app.

Use case 6

Use Case: user sets age

Scope: Activity tracker

Level: User goal

Primary Actor: User of activity tracker.

Stakeholders and Interests:

- User: Wants to record their age accurately and with ease.

Success Guarantee: age is recorded and saved accurately.

Main Success Scenario:

1. User turns on the display
2. The user goes to the home screen and selects the settings button
3. Once in the settings menu the user can choose to change age
4. The user can select age from an on screen buttons

Special Requirements:

- Wearable must have large text and buttons.
- The UI should be clean and easy to interact with
- The activity feedback should be in real time.

Technology and Data Variations: Wearable turned off and on by pressing and holding side button for 3 seconds.

Frequency of Occurrence: Continuous

Use case 7

Use Case: user sets daily step goals

Scope: Activity tracker

Level: User goal

Primary Actor: User of activity tracker.

Stakeholders and Interests:

- User: Wants to record their daily step goal accurately and with ease.

Success Guarantee: daily step goal is recorded and saved accurately.

Main Success Scenario:

1. User turns on the display
2. The user goes to the home screen and selects the settings button
3. Once in the settings menu the user can choose to change daily step goal
4. The user can select daily step goal from an on screen buttons

Special Requirements:

- Wearable must have large text and buttons.
- The UI should be clean and easy to interact with
- The activity feedback should be in real time.

Technology and Data Variations: Wearable turned off and on by pressing and holding side button for 3 seconds.

Frequency of Occurrence: Continuous

Use case 8

Use Case: user sets weight

Scope: Activity tracker

Level: User goal

Primary Actor: User of activity tracker.

Stakeholders and Interests:

- User: Wants to record their weight accurately and with ease.

Success Guarantee: weight is recorded and saved accurately.

Main Success Scenario:

1. User turns on the display
2. The user goes to the home screen and selects the settings button
3. Once in the settings menu the user can choose to change weight
4. The user can select their weight from an on screen buttons

Special Requirements:

- Wearable must have large text and buttons.
- The UI should be clean and easy to interact with
- The activity feedback should be in real time.

Technology and Data Variations: Wearable turned off and on by pressing and holding side button for 3 seconds.

Frequency of Occurrence: Continuous

Use case 9

Use Case: Download Data

Scope: Activity tracker

Level: User goal

Primary Actor: User of activity tracker.

Stakeholders and Interests:

- User: Wants to view their activity accurately and with ease.

Preconditions: The activity tracker is paired with a phone.

Success Guarantee: Activity is viewed accurately.

Main Success Scenario:

1. The user appropriately wears the activity tracker.
2. The data will be synced daily to smartphone app for user to view later.

Special Requirements:

- Wearable must be connected to phone
- The UI should be clean and easy to interact with
- The activity feedback should be in real time.

Frequency of Occurrence: daily

Miscellaneous Such as open issues: Transferring data from the wearable to the smartphone app.

Operation Contracts

Contract: Create New User

Operation: createNewUser (user: userID)

Cross References: N/A

Pre-conditions:

- System is on

Post-conditions:

- User instantiates a new user (instance creation)

Contract: setAge

Operation: setAge (Age: age)

Cross References: Use Cases: Set age.

Pre-conditions:

- System is on
- User accesses age through settings

Post-conditions:

- User instantiates age. (instance creation)

Contract: setWeight

Operation: setweight (Weight: weight)

Cross References: Use Cases: Set weight.

Pre-conditions:

- System is on
- User accesses weight thorough settings

Post-conditions:

- User instantiates weight. (instance creation)

Contract: setDailyStepGoals

Operation: setsDailyStepGoals (Step Goal: dailyStepGoal,)

Cross References: Use Cases: set daily step goals.

Pre-conditions:

- System is on
- User accesses daily step goals through settings

Post-conditions:

- User instantiates daily step goal. (instance creation)

Contract: recordActivity

Operation: recordActivity()

Cross References: N/A

Pre-conditions:

- System is on
- Pedometer active
- Gyroscope active

Post-conditions:

- Activity is recorded via activity class