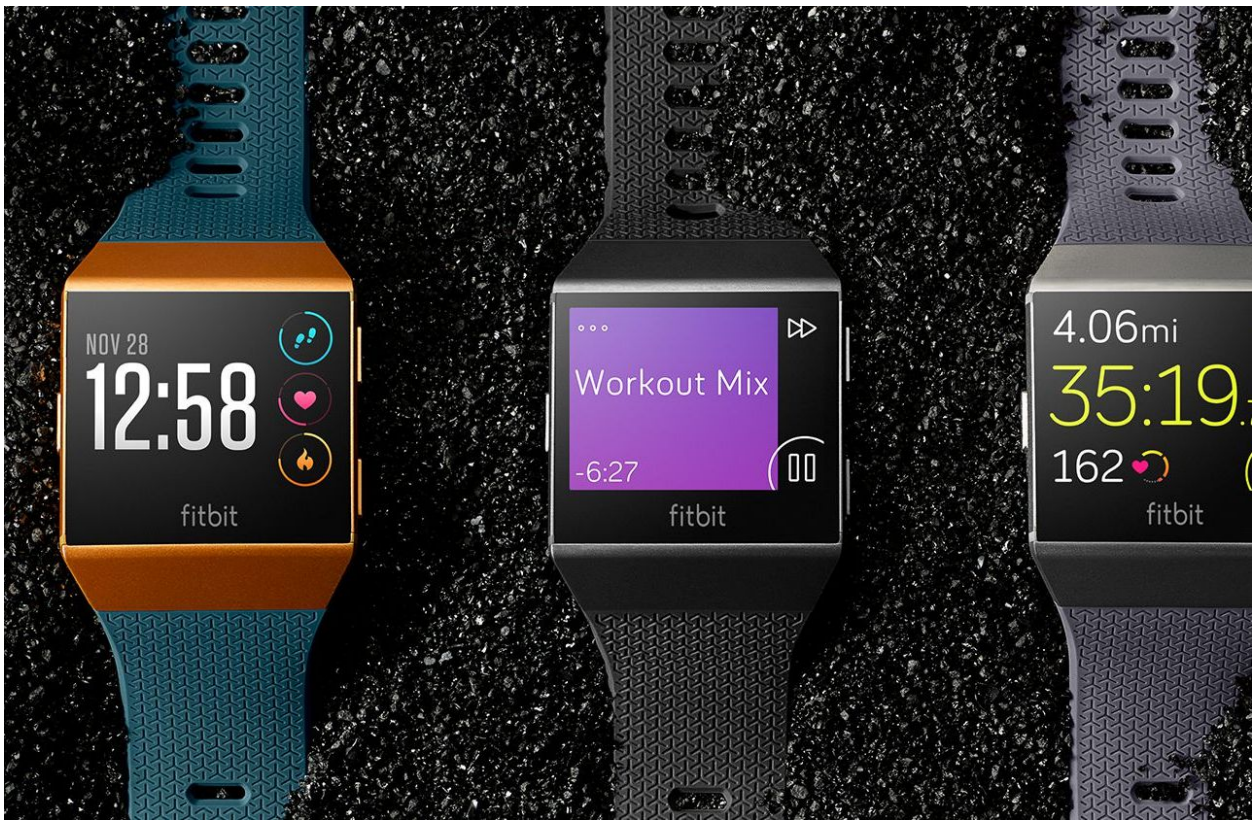


BitFit

Team 5: Lara Brooksbank, Jasmine Mai, and Anna Uhl

Revision History

Version	Date	Description
1.0	09/11/2017	Initial Version
2.0	9/28/2017	Deliverable 2
3.0	10/7/2017	Deliverable 3
3.1	10/16/2017	Deliverable 3 Revision
4.0	10/28/2017	Deliverable 4
5.0	11/09/2017	Deliverable 5
5.1	11/21/2017	Deliverable 5 Revision
6.0	12/2/2017	Deliverable 6

Vision

We envision a robust, accurate, and low-maintenance system that is very affordable and allows users to track lifestyle trends such as activity levels, heart rate, and sleep patterns in order to aid in their personal development.

Glossary

Term	Definition & Information	Format	Validation Rules	Aliases
Account	the personal database of the user containing data obtained through the device			
Application	the user interface on a phone or computer that would allow the user to access the data in their account			
Gyroscope	device that can determine its own orientation			
Heart-Rate	number of heartbeats per unit of time	Integer (unit: beats per minute)	Cannot be 0 or negative	Pulse
Steps	the movement of putting one leg in front of the other	Integer (unit: steps)	Cannot be negative	
Synchronization	the exchange of data between devices			Sync

Use Cases

Scope/System	FitBit
Name	**Tracking Physical Activity**
Level	User-goal
Primary Actor	FitBit
Stakeholders and Interests	FitBit: Provide an accurate documentation of steps & heart rate FitBit User: Document activity accurately
Preconditions	FitBit User must wear FitBit on wrist and have it connected to an account. FitBit needs to be charged & on
Postconditions	Accurate step calculation & heart rate is displayable Data is saved and documented
Main Success Scenario	1. FitBit User wears a charged FitBit 2. FitBit User continues with daily life as the FitBit tracks the number of steps & heart rate through the day 3. FitBit User can check the FitBit their activity levels through the day 4. Data collected throughout the day is at the end of each day & on demand
Extensions	3a. Display data 1. Upward wrist motion 2. Tap on FitBit 3. Press button
Special Requirements	Detection of steps & heartbeat Step number & heart rate display must be visible on request, large font & high contrast display
Variations in Technology and Data	*a. Changes in hardware with the FitBit will still be applicable to the software
Frequency of Occurrence	24/7
Miscellaneous	Hardware necessary for counting steps & measuring heart rate How the software and hardware will interact in order to have an accurate reading

Scope/System	FitBit
Name	**Displaying Time**
Level	User-goal
Primary Actor	FitBit
Stakeholders and Interests	FitBit: Provide a time display FitBit User: Tell time
Preconditions	FitBit User must wear FitBit on wrist and have it connected to an account. FitBit needs to be charged
Postconditions	Time is displayable & correct
Main Success Scenario	1. FitBit User wears a charged FitBit 2. FitBit User is able to view the time on demand
Extensions	2a. Display data 1. upward wrist motion 2. Tap on FitBit 3. Press button
Special Requirements	Time must be visible on request, large font & high contrast display
Variations in Technology and Data	*a. Different timezones *b. Toggle between 12-hr and 24-hr display
Frequency of Occurrence	24/7
Miscellaneous	

Scope/System	FitBit
Name	**Tracking Sleep**
Level	User-goal
Primary Actor	FitBit
Stakeholders and Interests	FitBit: Provide sleep documentation FitBit User: Document Sleep
Preconditions	FitBit User must wear FitBit on wrist and have it connected to an account. FitBit needs to be charged
Postconditions	Approximate sleeping time is displayable Data is saved and documented
Main Success Scenario	1. FitBit User wears a charged FitBit 2. FitBit User continues with daily life as the FitBit tracks the activity 3. If the FitBit User is inactive for an hour and their heart rate is slowed down (Tracking activity use case), sleeping mode will be triggered 4. Whenever the FitBit User moves, sleeping mode will deactivate 5. Sleep log will then be synced with the application as possible sleeping time once the user wakes up
Extensions	
Special Requirements	
Variations in Technology and Data	Changes in hardware with the FitBit will still be applicable to the software
Frequency of Occurrence	24/7
Miscellaneous	How much the heart rate slows down during sleep?

Scope/System	FitBit
Name	**Syncing**
Level	Subfunction
Primary Actor	FitBit
Stakeholders and Interests	FitBit: Provide updated user information & activity FitBit application: Save data from FitBit to be accessible to user
Preconditions	FitBit needs to be charged FitBit must be within bluetooth distance
Postconditions	Data stored on the FitBit is sent to and saved on application
Main Success Scenario	1. Every hour the FitBit will then sync all data to the application 2. Clear out the information that was synchronized
Extensions	*a. During initial setup, the FitBit User will fill out mandatory information in application 1a. If the application is not within bluetooth distance, the data will be stored until next synchronization period 1b. User is able to sync their information on demand 1c. If energy is very low, FitBit will sync before shutting down 1d. If memory is critically low, the FitBit will alert the user and sync automatically.
Special Requirements	Bluetooth hardware
Variations in Technology and Data	*a. Changes in hardware with the FitBit will still be applicable to the software
Frequency of Occurrence	24/7
Miscellaneous	How are we able to implement Bluetooth software with Java?

Operation Contracts

Contract CO1: makeNewActivityTracker

Operation: makeNewActivityTracker()

Cross References: Use Cases: Tracking Physical Activity

Pre-conditions:

- time = 00:00

Post-conditions:

- An ActivityTracker instance *at* was initialized
- A StepTracker instance *st* was initialized
- A HeartRateTracker instance *hrt* was initialized
- *st.stepNum* was set to 0
- ArrayList *hrt.heartRates* was initialized to save heart rate throughout the day

Contract CO2: measureStep

Operation: measureStep()

Cross References: Use Cases: Tracking Physical Activity

Pre-conditions:

- *st.*isOn() = true

Post-conditions:

- *st.currentStep* was set to a value (0 or 1) from the Accelerometer
- *st.stepNum* was incremented if a step was taken

Contract CO3: measureHeartRate

Operation: measureHeartRate()

Cross References: Use Cases: Tracking Physical Activity

Pre-conditions:

- *hrt.*isOn() = true

Post-conditions:

- *hrt.currentRate* was updated with the user's newest heart rate
- *hrt.currentRate* was appended to ArrayList *hrt.heartRates*

Contract CO4: demandSync

Operation: demandSync()

Cross References: Use Cases: Tracking Physical Activity, Tracking Sleep

Pre-conditions:

-

Post-conditions:

- A new instance of SyncSession *ss* was initialized
- *at.stepNum*, *at.heartRate*, *si.SleepTime* was sent to account

Contract CO5: makeNewSleep

Operation: makeNewSleep()

Cross References: Use Cases: Tracking Sleep

Pre-conditions:

-

Post-conditions:

- A new sleep instance *si* has been initialized
- The time at initialization was saved as *si.beginSleepTime*

Contract CO6: endSleep

Operation: endSleep()

Cross References: Use Cases: Tracking Sleep

Pre-conditions:

- New sleep instance *si* has been initialized

Post-conditions:

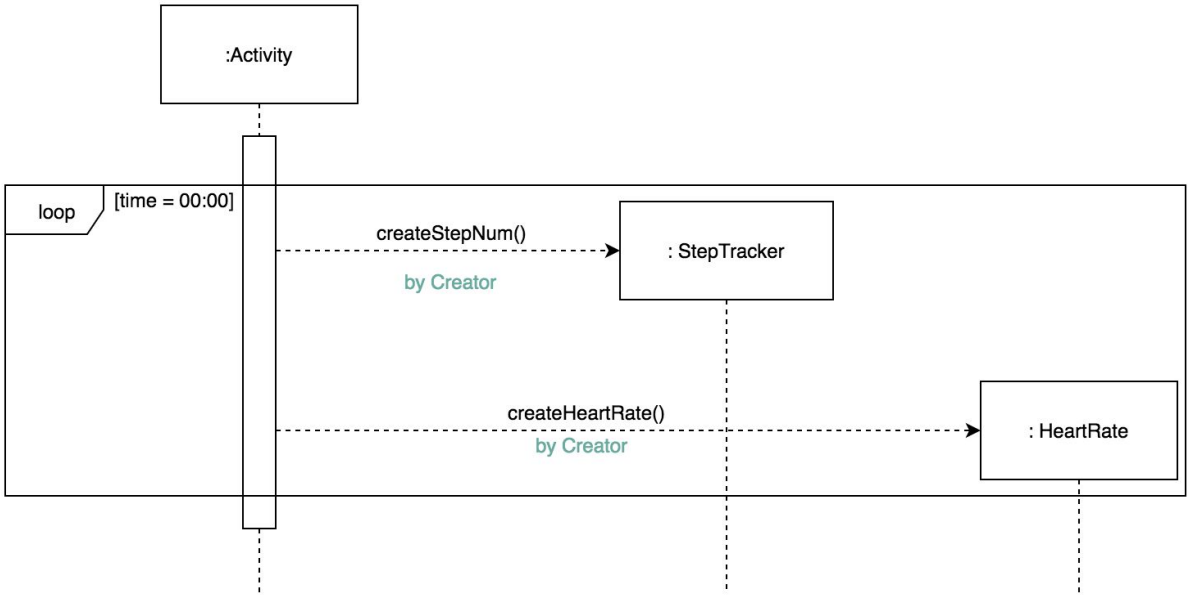
- The time at which the new sleeping period had been ended was saved as *si.endSleepTime*
- The total sleep duration *si.sleepTime* = *si.endSleepTime* - *si.beginSleepTime* was calculated and saved

Testing

The testing that occurred during this project consisted of making sure the clock was producing the correct time using other timed devices as an oracle. We tested to make sure the total time slept that was being produced was accurate. We also tested that the syncing feature was writing to the correct file.

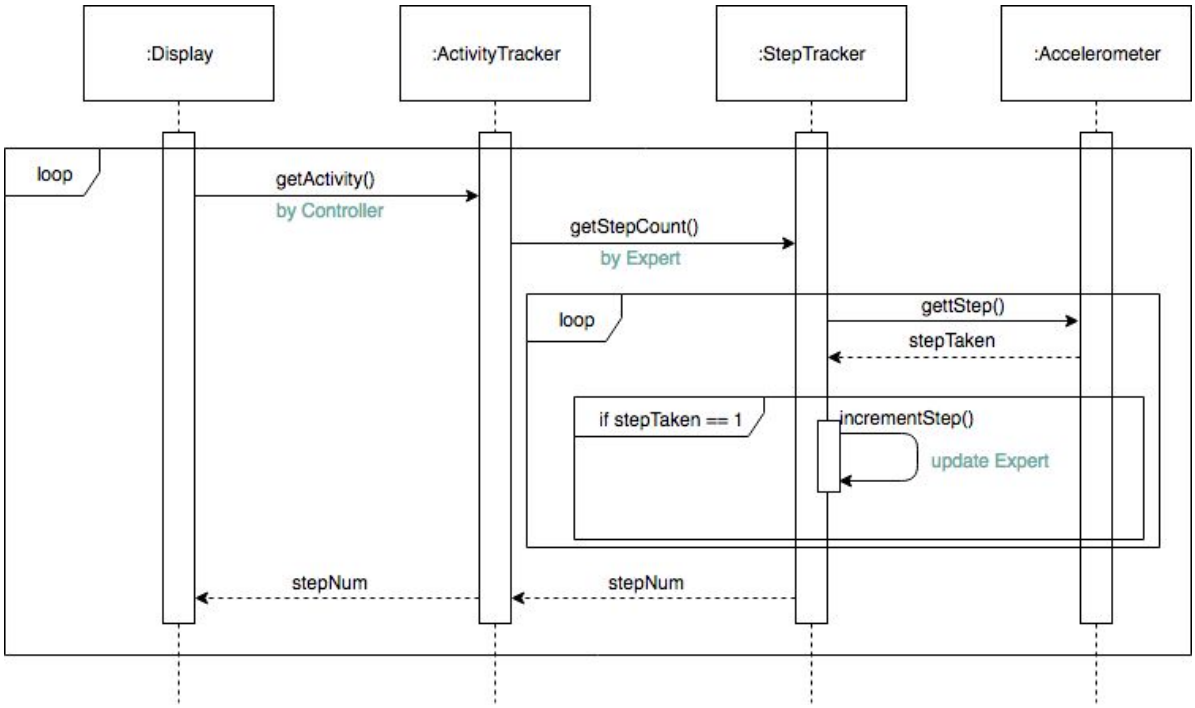
Sequence Diagrams

activityTracker

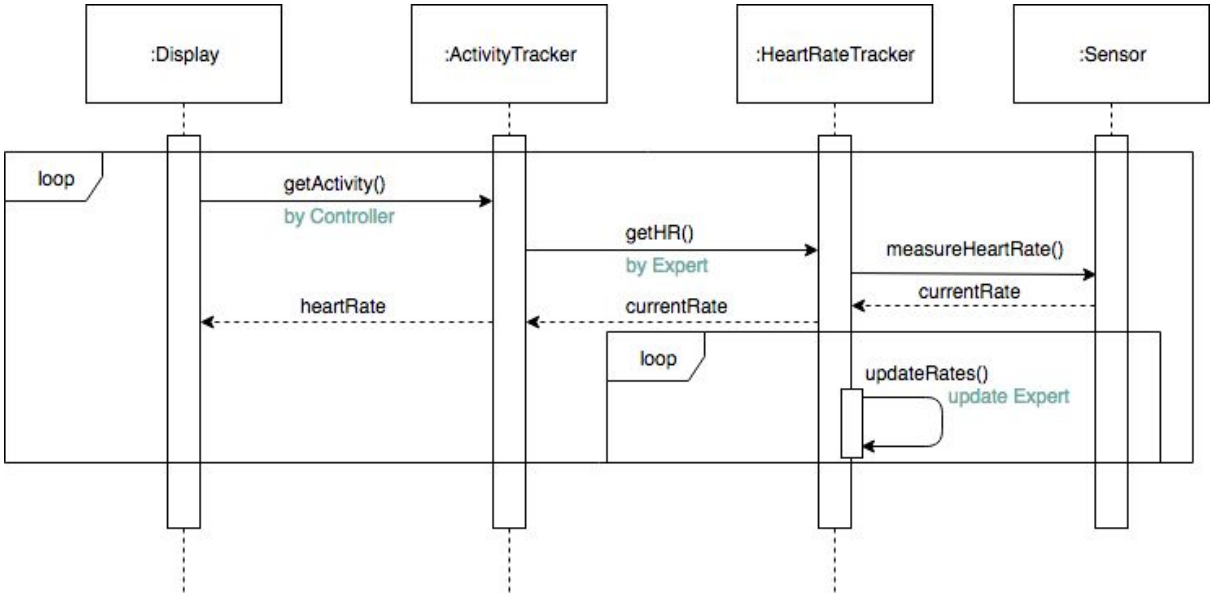


Pure fabrication, it is not necessary, but it adds organization to our application.

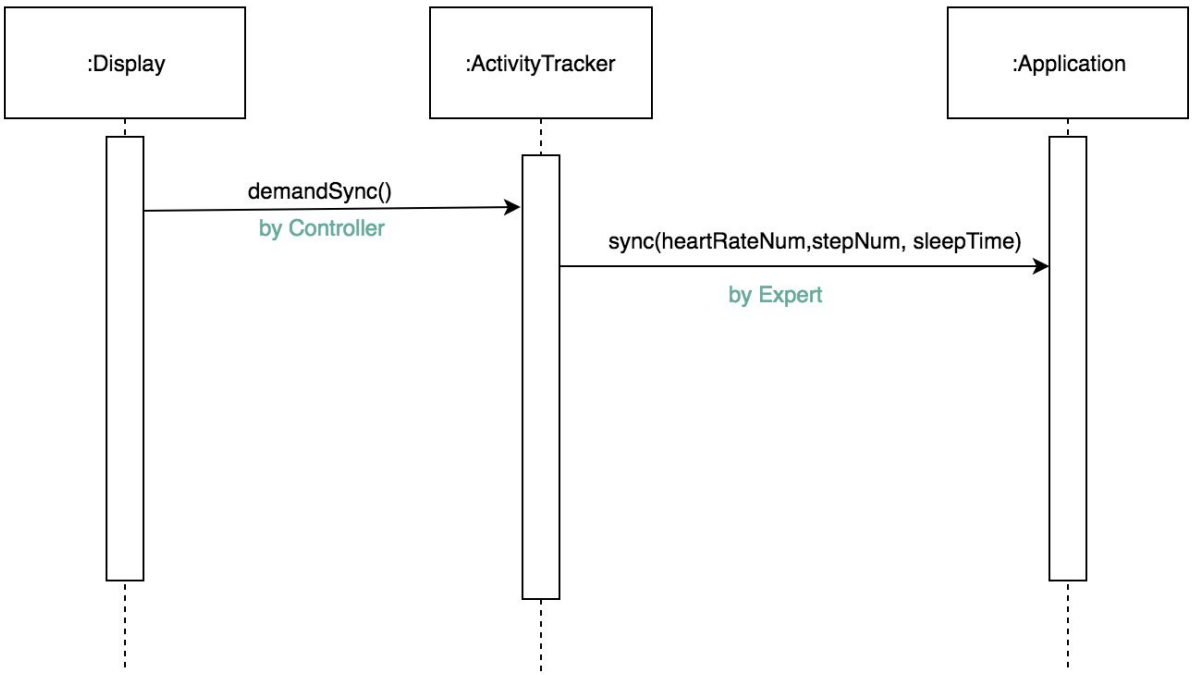
measureStep



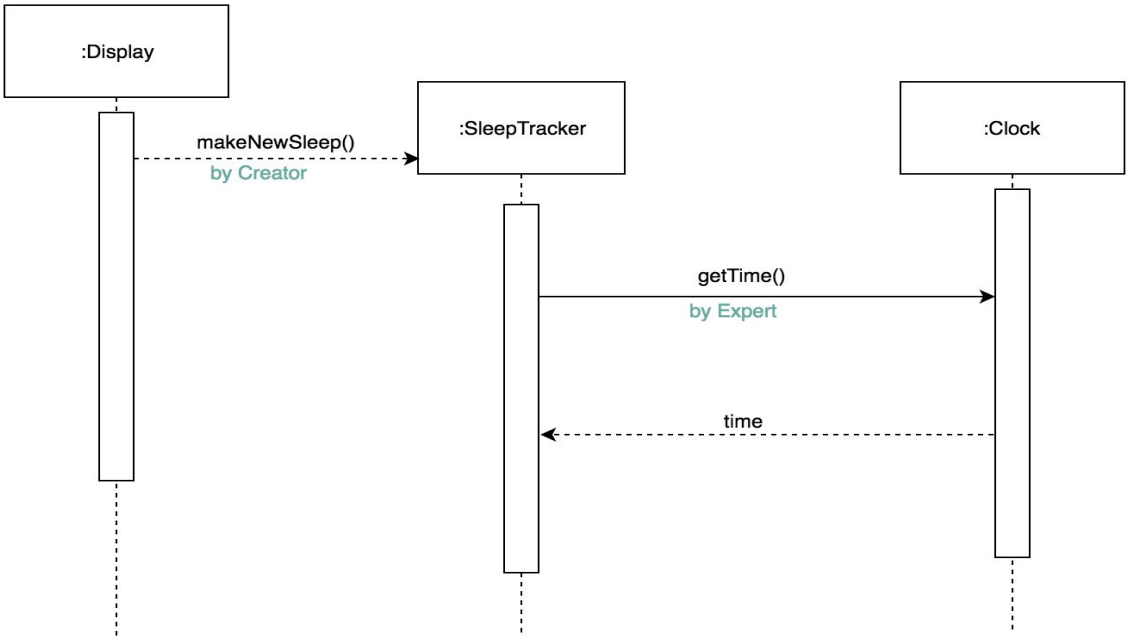
measureHeartRate



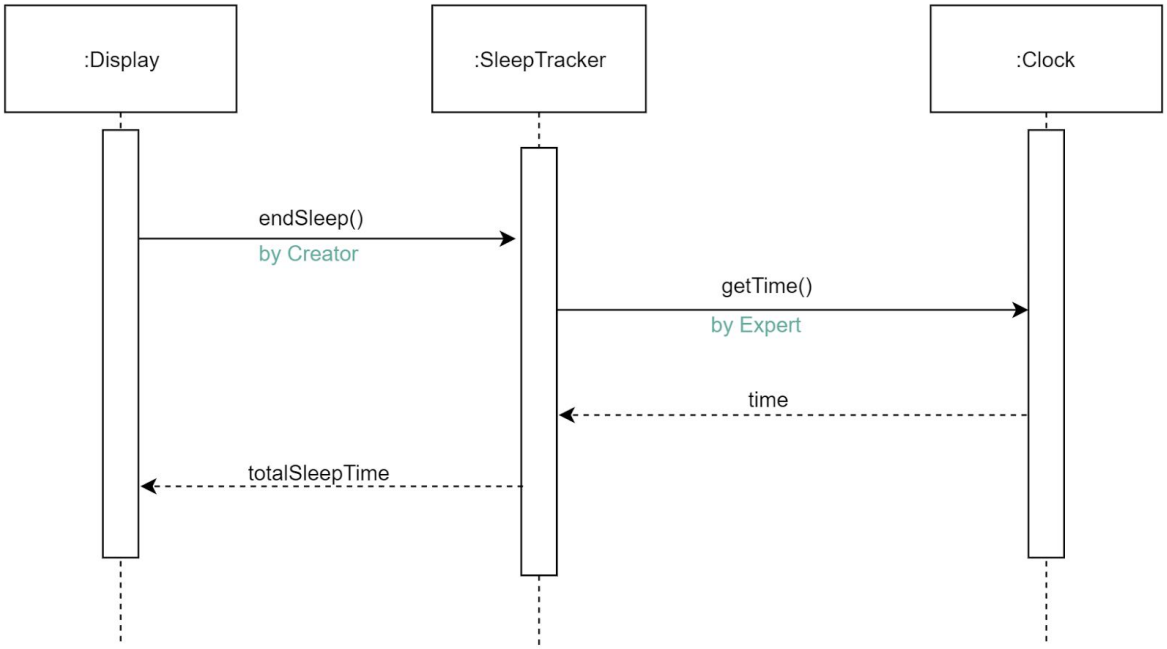
demandSync



makeNewSleep



endSleep



Class Diagram