**Activity Tracker Software**

**Software Architecture Document**

ThunderBits

# Table of Contents

# 1. Introduction

## 1.1. Vision

In today's society, staying fit and healthy is a must, but sometimes life may get in the way and it can be hard to stay healthy. You may need a device to keep it all together, in comes ThunderBits. This is an activity tracker that can be a force to motivate and remind its user to stay fit and healthy. Their goals can be set and met at the click of a button. The vision of this project is to implement an aesthetically pleasing personal health monitor that controls and monitors the users physical activities. Because this device can be used as a watch that displays the date and the time, users will be more inclined to use the other features just by swiping through the different screens. They will be able to see a summary of how long they have slept for that day and their sleeping patterns throughout the duration of its wear. The user will be able to track their steps along with how many calories they burn.

**The major features include:**

1. Time Display
2. Tracks Activity
3. Recognize and track Sleep Patterns
4. Data Transmission

### 1.1.1. Stakeholders

Market: Fitness tracking and not Smart Watch market
User: Dr. X
Key Goals: Tracking and display data with a user-friendly interface.
Problems: Current fitness trackers don’t fully satisfy their customers.

## 1.2. Glossary

Calorie - unit of energy
Clock Face - special type of app that is used to display the date and time.
Display - section of the fitness tracker that will show the data on a screen
ECG - Electrocardiogram. Measures the electrical activity of the heart.
Fitness Tracker - tracks and monitors physical activity, as well as other physiological functions (ex: breathing, sleeping, heart rate, and calorie burn).
Heart Rate - how many beats per minute your heart makes.
MHR - Maximal Heart Rate. Maximum beats per minute your heart should make when operating at highest intensity.
Resting Heart Rate - how many beats per minute your heart makes when you are inactive.
Settings - allows users to change the appearance or behavior of an application.
Sleep Tracker - tracks how much the user slept, quality of sleep, and time spent in REM sleep.

## 1.3. References

*Fostering Engagement with Personal Informatics Systems*

Personal Informatics is a classification of tools that are used to help people collect personally relevant data for the purpose of self-monitoring. Activity trackers are included within this classification, and are incredibly common due to the improvements in cost, design, and ease of use. However, many systems are limited by their ability to encourage the user's engagement, and these systems will in some cases be abandoned after a period of time. This paper draws from various sources to identify the various problems that inhibit the performance of these systems, and then determines seven strategies for the design of such systems. One strategy that stood out the most was in regards to the overall design of the user interface, where ambient awareness is used to reduce the overall "grandness" of the personal informatics system. Ambient awareness is where information is displayed in a manner that does not bring a significant amount of attention to it. Such a strategy will allow the user to see details when necessary, but not so much so that it will distract from everyday life. This would suggest that for our implementation we should consider a design that adds to the user's everyday life, but does not distract them or hinder them from their normal activities [6].

*The Role of Aesthetics and Design: Wearables*

A three-part study was conducted in 2018 where 15 participants were given customised, low-fidelity "activity trackers" based on their own designs, Participants used these prototypes to give insight on their feelings towards their own designs and the current commercial devices. It was found that aesthetics were important to them, and so far, underappreciated with commercial activity trackers, causing a drop in continued user engagement.

This study, like the paper above, found that users enjoyed a product more if the user interface was designed with the actual user in mind. It was suggested that manufacturers should embrace adaptability of their system and allow users to customize the interface, and choose their own designs. This encourages us to consider adding customization to our interface, so that the user will feel more engaged with the software and product. However, this is not a function that should be focused on, if key activity tracker features will be overlooked. Such features are discussed below [5].

*How Do Wearables Track Sleep?*

The majority of activity trackers have three axis accelerometers which allow for the monitoring of how quickly the user moves in any direction. With algorithms, the data is processed to determine what these movements really mean. The American Academy of Sleep Medicine determined in 1995 that actigraphy is a useful tool to pull data regarding a user's sleep cycle. Actigraphy is the use of the accelerometer to track the speed and direction of the user's movements to measure sleep quality. With actigraphy, the three axis accelerometer on the activity tracker can record all of the user's movements, and then using specialized algorithms, the movements can be used to make sense of the user's sleep patterns. Other sleep trackers simply use continuous heart-rate monitoring, to determine based on the user's heart rate whether or not they are asleep or awake, and which stage of sleep they are in. As of now, Fitbit markets that the Charge 3 and Versa have SpO2 (pulse-ox) sensors, but the do not technically do anything at the moment. The implementation of sleep tracking within the activity tracker is one that user's can greatly benefit from [2].

*Pedometer and Step Counter Mobile Apps : How does it work?*

This source was about how pedometers and step counters apps work. It tells how pedometer helps keep track of your steps while doing various activities and capture your distance covered. There are sections that talk about how accurate the count would be, how it affects your phone, and how it works. Step counters work with accelerometer sensors built in it. This provides the x, y, and z axis values where the x increases at a rate depending upon the force of the person's movement. For the problem, the source gives incite on some key steps for implementing a step counter. First you extend the activity to implement *SensorEventListener(),* second get the sensor manager using getSystemService(), third heck for the availability of Sensor and get the sensor using *getDefaultSensor()* functionality of Sensor Manager specifying the Sensor type, fourth heck for the availability of Sensor and get the sensor using *getDefaultSensor()* functionality of Sensor Manager specifying the Sensor type, fifth Implement the logic for displaying or using the Step event in *onSensorChanged()* function called by the system.Check for the time-stamp for duplicate events. Also says that android has a low power step detector feature that developers can use [4].

*Fitbit Heart Rate Monitoring Explained*
In order to make something that can track your heart rate you must first understand how it is used in fitness trackers. This source explains the heartbeat monitor and calories burned application on the Fitbit. It gives you reasons on the uses of a heart rate sensor like it gives you real time heart rate and calorie burn information with a summary of your average heart rate. It also gives incite on how heartbeats are measured, which I think is helpful to know for the problem. It used green LEDs with photodiodes to detect and get accurate beats per minute data  and then uses their company's algorithm to determine accuracy. It also takes in your basal metabolic rate for calories burned. A heart rate monitor can have some problems with fluctuating between high and low heart rate during interval training [1].

*A Look at the Security and Privacy of Fitbit as a Health Activity Tracker*
Since activity trackers are increasing in popularity, there is also a greater need to evaluate the accuracy and security of such software. A study conducted in April of 2019 wanted to determine the accuracy and security risks of the Fitbit Blaze. 24 participants used and evaluated the device, and the study determined that the Fitbit's accuracy was not equivalent to medical grade devices, and that most of the security risks come from potentially fraudulent third part applications. The study pulled from various other research studies where security analysis was done on technical aspects of data collection and transmission between Fitbit devices, mobile applications, and Fitbit databases. Doing this, they found certain security vulnerabilities concerning how the user's data is collected and moved around. A study also found that it could be out-of-date firmware that is a source of this vulnerability. In this regard, we should further discuss the security measures we could take to make sure our implementation is as secure as possible [3].

*Developer for Fitbit*
This is the fitbit developers website where it gives a way to build apps and clock faces using javascript, css, and svg that go on the fitbit. This source can contribute a way to build and customize the clock of the fitbit. It uses Fitbit Software Development Kit (SDK) with Fitbit Studio. This source also gives guides on each feature of it and how to use them. To build the clock face it shows how to do so in both digital or analog. It can also shows how you would implement the settings and to implement GPS API that uses either the GPS receiver thats on it or uses connected GPS from the users phone. This gives you the current location and a way to continuously monitor the users GPS coordinates [7].

*FitbitOS*
This is a high level overview of the Fitbit operating system. It highlights the different features across Fitbit devices. More interestingly, it slightly provides or details use cases for each feature. There's also the ability to look into each device and see the different features made available for each one [8].

*Fitbit Technology*
This resource details Fitbit's technology that power their devices and that has fueled hundreds of research studies. It highlights the three main technologies that Fitbit uses, which are PurePulse, SmartTrack, and Sleep Tracking. For PurePulse, it slightly highlights Fitbit's historical use of and development of their heart rate technology and how they used photoplethysmography to measure blood flow and in turn heart rate. It also details how PurePulse is used within the following features: Resting Heart Rate, Cardio Fitness Level, Heart Rate Zones, and Guided Breathing. In regards to SmartTrack, it highlights the different exercises devices can recognize including walking, running, and outdoor biking. It also details how the user can customize their SmartTrack experience by setting the length of time for each exercise the user wants their device to automatically recognize. Finally, it details the motivations for Sleep Tracking and the different features that rely on Sleep Tracking, which include: Auto Sleep Tracking and Alarms, Sleep Stages, Sleep Schedule, and Sleep Insights [9].

*Competitor Tech: Mi Smart Band*

According to an IDC report published on December 2018, Fitbit is considered the third largest wearable company in shipments as of the third quarter of 2018, behind Xiaomi and Apple. The Mi Smart Band is developed by Xiaomi and is the Global Number 1 Company in wrist-worn wearables for the first quarter of 2019. The Mi Smart Band has several similar features to Fitbit devices including sleep monitoring, heart rate monitoring, workout modes, music controls, and many others. Although this device is the only one developed and sold by Xiaomi, it contains more features that would otherwise be spread across different Fitbit devices, such as swimming mode and swim tracking. Features that the Mi Band has that Fitbits don't have are Do Not Disturb, Find My Phone, and Night Mode. The Mi Band outperforms Fitbit devices in battery performance with batteries lasting up to 20 days while the main Fitbit devices can last up to 4, 5, or 7 days [10].

## 2. Architectural Goals and Constraints

### 2.1. Supplementary Specifications

- Display width and height
- The use of JavaFX
- Supportability: Displayed on laptop only
- Language: English
- **Non-functional Requirements:**
  - Security: Passcode to unlock screen
  - Performance: Smooth interface
  - Reliability: It will operate with no error
  - Usability: Easily understood interface

## 3. Use Case Model

### 3.1. Display Data

| Use Case Section | Comment |
|---|---|
| **Use Case Name** | Display Data to User |
| **Scope** | Activity Tracker Software |
| **Level** | User Goal |
| **Primary Actor** | User |
| **Stakeholders and Interests** | - User:<br>  - Wants accurate time,steps, calories, sleep data displayed<br>- Tracker: Accurate time, steps, calories, sleep data displayed |
| **Preconditions** | - Tracker must be on |

| | |
|---|---|
| **Success Guarantee** | - User can see the data whether it's on their wrist or not. |
| **Main Success Scenario** | - User looks at Tracker.<br>- Tracker displays correct data. |
| **Extensions** | a. User looks at Tracker.<br>b. A non-time screen is displayed.<br>c. User swipes/navigates to specific display.<br>d. Tracker displays correct data. |
| **Special Requirements** | N/A |
| **Technology and Data Variations List** | N/A |
| **Frequency of Occurrence** | Continuous |
| **Miscellaneous** | N/A |

## 3.2. Display Calories

| Use Case Section | Comment |
|---|---|
| ~~**Use Case Name**~~ | ~~Display calories burned to user~~ |
| ~~**Scope**~~ | ~~Activity Tracker Software~~ |
| ~~**Level**~~ | ~~User Goal~~ |
| ~~**Primary Actor**~~ | ~~User~~ |
| ~~**Stakeholders and Interests**~~ | ~~- User: Wants accurate data for calories burned presented~~<br>~~- Tracker: Wants to accurately record the amount of calories burned~~ |
| ~~**Preconditions**~~ | ~~User is identified and "wearing" the activity tracker.~~ |
| ~~**Success Guarantee**~~ | ~~The calories burned data is logged and saved. User's total calories burned are calculated and then updated. Daily calories burned display generated.~~ |
| ~~**Main Success Scenario**~~ | ~~1. User uses the activity tracker and adds input regarding calories burned.~~<br>~~2. Tracker tracks the calories.~~<br>~~3. Software calculates the total daily calories burned and displays it on the appropriate screen.~~ |

| | |
|---|---|
| | 4. ~~User checks the display to see how many calories they burned throughout the day.~~ |
| ~~**Extensions**~~ | a. ~~At any time, User can swipe to a different display.~~ |
| ~~**Special Requirements**~~ | - ~~User interface with swipe capabilities~~ |
| ~~**Technology and Data Variations List**~~ | a. ~~Tracker display determined by User swiping through the interface~~ |
| ~~**Frequency of Occurrence**~~ | ~~Could be continuous~~ |
| ~~**Open Issues**~~ | - ~~Explore data protection and security.~~ |

## 3.3. Display Steps

| Use Case Section | Comment |
|---|---|
| ~~**Use Case Name**~~ | ~~Display steps to user~~ |
| ~~**Scope**~~ | ~~ActivityTracker Software~~ |
| ~~**Level**~~ | ~~User Goal~~ |
| ~~**Primary Actor**~~ | ~~User~~ |
| ~~**Stakeholders and Interests**~~ | - ~~User: Wants accurate data for steps presented~~<br>- ~~Tracker: Wants to accurately record steps~~ |
| ~~**Preconditions**~~ | ~~User is identified and "wearing" the activity tracker.~~ |
| ~~**Success Guarantee**~~ | ~~Step data is logged and saved. User's total daily steps are calculated and then updated. Daily steps display generated.~~ |
| ~~**Main Success Scenario**~~ | 1. ~~User uses the activity tracker and adds input regarding steps walked..~~<br>2. ~~Tracker tracks the steps.~~<br>3. ~~Software calculates the total daily steps and displays it on the appropriate screen.~~<br>4. ~~User checks the display to see how many steps they took throughout the day.~~ |
| ~~**Extensions**~~ | a. ~~At any time, User can swipe to a~~ |

| | |
|---|---|
| | ~~different display.~~ |
| ~~**Special Requirements**~~ | ~~- User interface with swipe capabilities~~ |
| ~~**Technology and Data Variations List**~~ | ~~a. Tracker display determined by User swiping through the interface~~ |
| ~~**Frequency of Occurrence**~~ | ~~Could be continuous~~ |
| ~~**Open Issues**~~ | ~~- Explore data protection and security.~~ |

### 3.4. Display Sleep

| Use Case Section | Comment |
|---|---|
| ~~**Use Case Name**~~ | ~~Display sleep patterns to user~~ |
| ~~**Scope**~~ | ~~ActivityTracker Software~~ |
| ~~**Level**~~ | ~~User Goal~~ |
| ~~**Primary Actor**~~ | ~~User~~ |
| ~~**Stakeholders and Interests**~~ | ~~- User: Wants to see how long and how well they have slept~~<br>~~- Tracker: Displays accurate and informative data on sleep~~ |
| ~~**Preconditions**~~ | ~~User is identified and “wearing” the activity tracker.~~ |
| ~~**Success Guarantee**~~ | ~~Sleep data is tracked and saved. User will be able to see an accurate representation of how they slept the night before and see their fluctuating sleep pattern throughout the users wear~~ |
| ~~**Main Success Scenario**~~ | ~~1. User sleeps and the trackers begins to monitor them~~<br>~~2. User wakes up and the tracker stops monitoring and calculates time slept.~~<br>~~3. User slides the screen to the sleep screen.~~<br>~~4. Tracker displays the number of hours and minutes the user has slept.~~ |
| ~~**Extensions**~~ | ~~a. At any time, User can swipe to a different display.~~<br>~~b. User can click sleep icon and tracker will display a more in-depth reading of users sleep~~ |

| | |
|---|---|
| ~~**Special Requirements**~~ | ~~- User interface with swipe capabilities~~ |
| ~~**Technology and Data Variations List**~~ | ~~a. Tracker display determined by User swiping through the interface~~ |
| ~~**Frequency of Occurrence**~~ | ~~Could be continuous~~ |
| ~~**Open Issues**~~ | ~~- Explore data protection and security.~~ |

## 3.5. Unlock Screen

| Use Case Section | Comment |
|---|---|
| ~~**Use Case Name**~~ | ~~Unlock Screen~~ |
| ~~**Scope**~~ | ~~ActivityTracker Software~~ |
| ~~**Level**~~ | ~~User Goal~~ |
| ~~**Primary Actor**~~ | ~~User~~ |
| ~~**Stakeholders and Interests**~~ | ~~- User: Wants to keep their information protected~~<br>~~- Tracker: Displays a protection lock~~ |
| ~~**Preconditions**~~ | ~~User is identified and "wearing" the activity tracker.~~ |
| ~~**Success Guarantee**~~ | ~~Tracker displays lockscreen~~ |
| ~~**Main Success Scenario**~~ | ~~1. User will turn on tracker~~<br>~~2. Tracker will display a lock screen~~<br>~~3. User puts in correct password and device opens to home screen~~ |
| ~~**Extensions**~~ | ~~a. At any time, User can swipe to a different display.~~ |
| ~~**Special Requirements**~~ | ~~- User interface with swipe capabilities~~ |
| ~~**Technology and Data Variations List**~~ | ~~a. Tracker display determined by User swiping through the interface~~ |
| ~~**Frequency of Occurrence**~~ | ~~Could be continuous~~ |
| ~~**Open Issues**~~ | ~~- Explore data protection and security.~~ |

### 3.6. Change the Time

| Use Case Section | Comment |
|---|---|
| **Use Case Name** | Change the time |
| **Scope** | ActivityTracker Software |
| **Level** | User Goal |
| **Primary Actor** | User |
| **Stakeholders and Interests** | User: Wants to display time.<br>Tracker: Displays time. |
| **Preconditions** | User is identified and "wearing" the activity tracker. |
| **Success Guarantee** | Tracker saves inputted time. |
| **Main Success Scenario** | 1. User will press the settings button.<br>2. Tracker will display the settings window.<br>3. User inputs correct time.<br>4. Tracker saves time. |
| **Extensions** | a. At any time, User can leave the settings window. |
| **Special Requirements** | - User interface with click capabilities. |
| **Technology and Data Variations List** | a. Time display determined by User input. |
| **Frequency of Occurrence** | - Could be continuous |
| **Open Issues** | - N/A |

### 3.7. Input Personal Data

| Use Case Section | Comment |
|---|---|
| **Use Case Name** | Input personal data |
| **Scope** | ActivityTracker Software |
| **Level** | User Goal |
| **Primary Actor** | User |

| | |
|---|---|
| **Stakeholders and Interests** | - User: Wants to keep their information private and input important data<br>- Tracker: Uses data customized for User |
| **Preconditions** | User is identified and "wearing" the activity tracker. |
| **Success Guarantee** | Tracker allows User to input data, and then saves the data. |
| **Main Success Scenario** | 1. User will input data.<br>2. Tracker will save the data.<br>3. Tracker will re-display correct data. |
| **Extensions** | a. At any time, user can exits the data input window. |
| **Special Requirements** | - User interface with input capabilities. |
| **Technology and Data Variations List** | a. User data determined by User input. |
| **Frequency of Occurrence** | - Unlimited |
| **Open Issues** | - N/A |

### 3.8. Input Goal

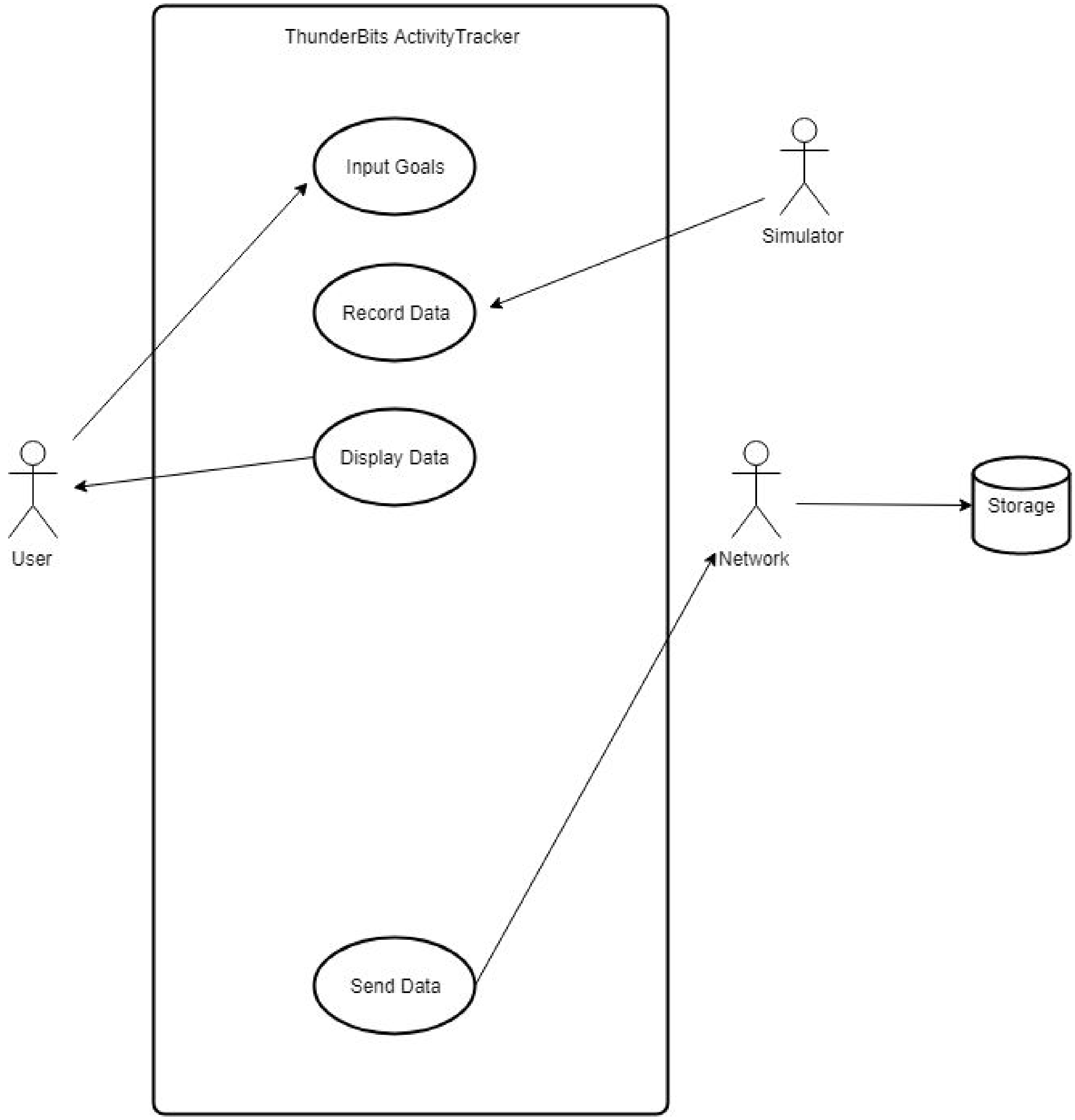
| Use Case Section | Comment |
|---|---|
| **Use Case Name** | Input goal(steps or calories) |
| **Scope** | ActivityTracker Software |
| **Level** | User Goal |
| **Primary Actor** | User |
| **Stakeholders and Interests** | User: Wants to track progress of goals |
| **Preconditions** | User is identified and "wearing" the activity tracker. |
| **Success Guarantee** | |
| **Main Success Scenario** | 1. Move to Steps or Calories Screen<br>2. Set new goal<br>3. Progress shown |

| | |
|---|---|
| **Extensions** | Goal is modified:<br>- Lower goal<br>- Higher goal |
| **Special Requirements** | N/A |
| **Technology and Data Variations List** | N/A |
| **Frequency of Occurrence** | Unlimited |
| **Open Issues** | N/A |

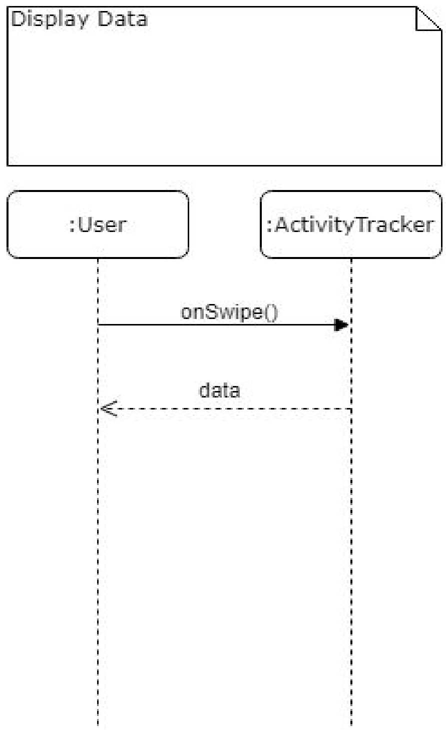
**3.9.**

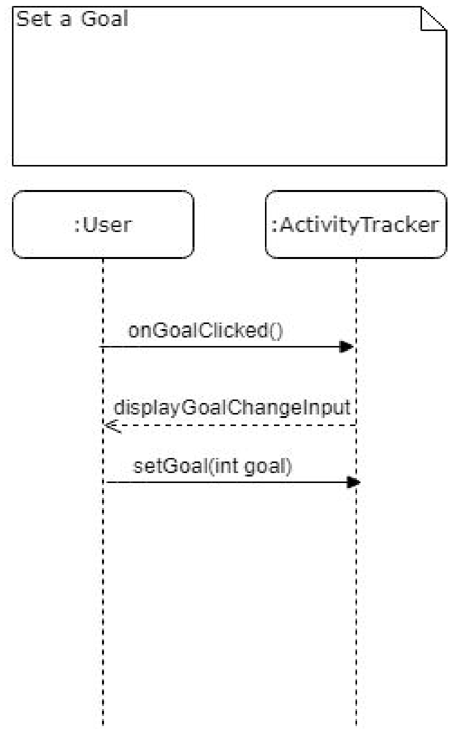
| Use Case Section | Comment |
|---|---|
| **Use Case Name** | Send Data via Network |
| **Scope** | ActivityTracker Software |
| **Level** | User Goal |
| **Primary Actor** | User |
| **Stakeholders and Interests** | User requires the ability to transfer data from Tracker to Cell phone or storage system |
| **Preconditions** | - Data exists<br>- Established connection |
| **Success Guarantee** | - Data must be off tracker and within cell phone or storage system |
| **Main Success Scenario** | 1. Go to the settings menu<br>2. Click on Send data to connected device |
| **Extensions** | If connection fails: reestablish connection and try again |
| **Special Requirements** | Secured network connection |
| **Technology and Data Variations List** | N/A |
| **Frequency of Occurrence** | Unlimited |
| **Open Issues** | N/A |

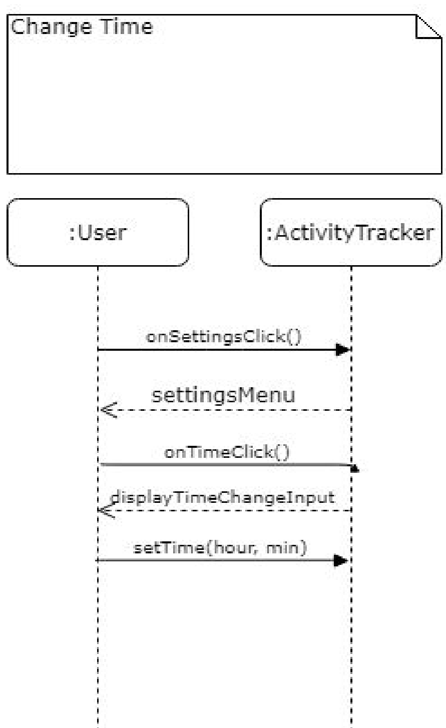
### 3.10. Use Case UML Diagram

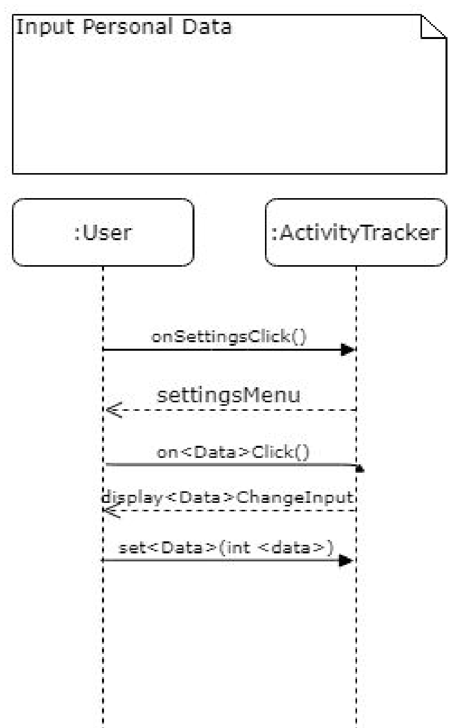


## 3.11. System Sequence Diagrams



## 3.12. Operation Contracts

### System Operations of the Display Data Use Case

Contract CO1: onSwipe

| | |
|---|---|
| Operation: | onSwipe() |
| Cross Reference: | Use Cases: Display Data |
| Preconditions: | The display is swipeable. |
| Postconditions: | - Display was changed.<br>- Data d was displayed. |

### System Operations of the Change Time Use Case

Contract CO1: onSwipe

| | |
|---|---|
| Operation: | onSettingsClick() |
| Cross Reference: | Use Cases: Change Time |
| Preconditions: | The display is on. |
| Postconditions: | - Settings window is initialized. |

Contract CO2: onTimeClick

| | |
|---|---|
| Operation: | onTimeClick() |
| Cross Reference: | Use Cases: Change Time |
| Preconditions: | The Settings window is displayed.<br>The display is on. |
| Postconditions: | - Textbox time was activated. |

Contract CO3: setTime

| | |
|---|---|
| Operation: | setTime(hour,min) |
| Cross Reference: | Use Cases: Change Time |
| Preconditions: | Textbox time is active. |
| Postconditions: | - Time was saved. |

## System Operations of the Set Goal Use Case

Contract CO1: onGoalClicked

| | |
|---|---|
| Operation: | onGoalClicked() |
| Cross Reference: | Use Cases: Set Goal |
| Preconditions: | User is in the data display. Display is clickable. |
| Postconditions: | - Goal Input window was displayed. |

Contract CO2: setGoal

| | |
|---|---|
| Operation: | setGoal(int goal) |
| Cross Reference: | Use Cases: Set Goal |
| Preconditions: | User is in the Goal Input window. Display is clickable. |
| Postconditions: | - Goal was set. |

## System Operations of the Input Personal Data Use Case

Contract CO1: onSettingsClick

| | |
|---|---|
| Operation: | onSettingsClick() |
| Cross Reference: | Use Cases: Input Personal Data |
| Preconditions: | The display is clickable. |
| Postconditions: | - Settings window did appear. |

Contract CO2: on<Data>Click

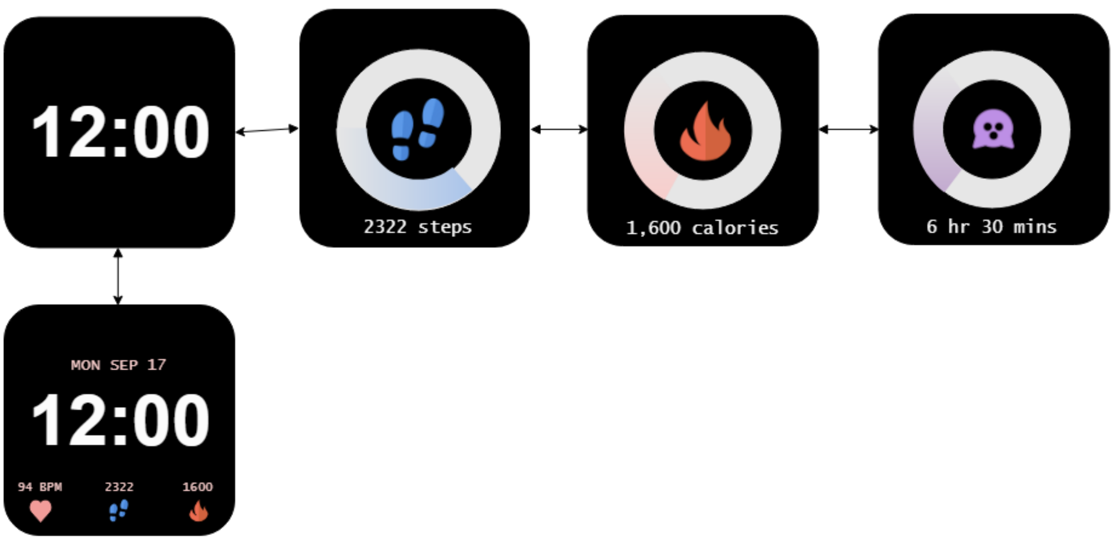
| | |
|---|---|
| Operation: | on<Data>Click() |
| Cross Reference: | Use Cases: Input Personal Data |
| Preconditions: | User is in the settings display. The data inputs are clickable. |
| Postconditions: | - Data input textbox was activated. |

Contract CO3: set<Data>

| | |
|---|---|
| Operation: | set<Data>(int <data>) |
| Cross Reference: | Use Cases: Input Personal Data |
| Preconditions: | Data input textbox is activated. |
| Postconditions: | Data saved. |

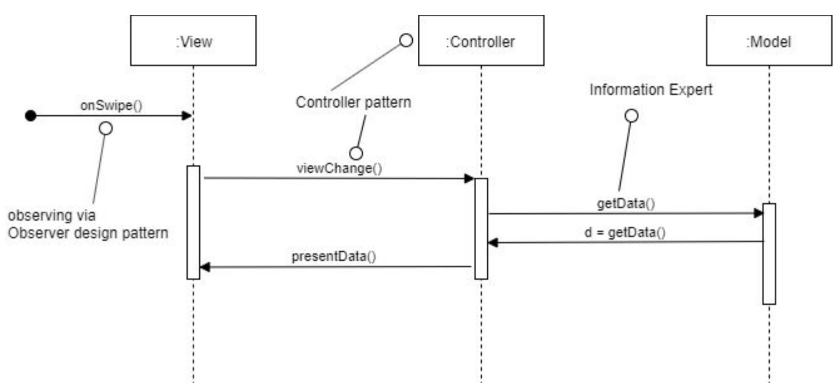
# 4. Design Model

## 4.1. Overview

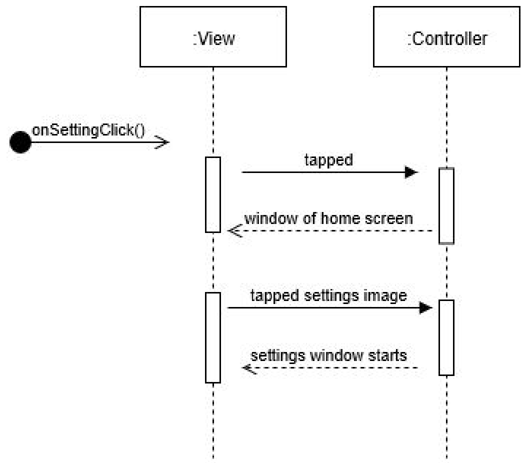



## 4.2. Sequence Diagrams

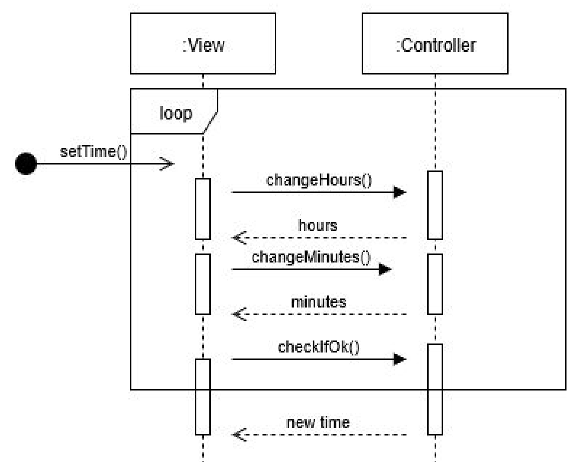
### 4.2.1. Display Data



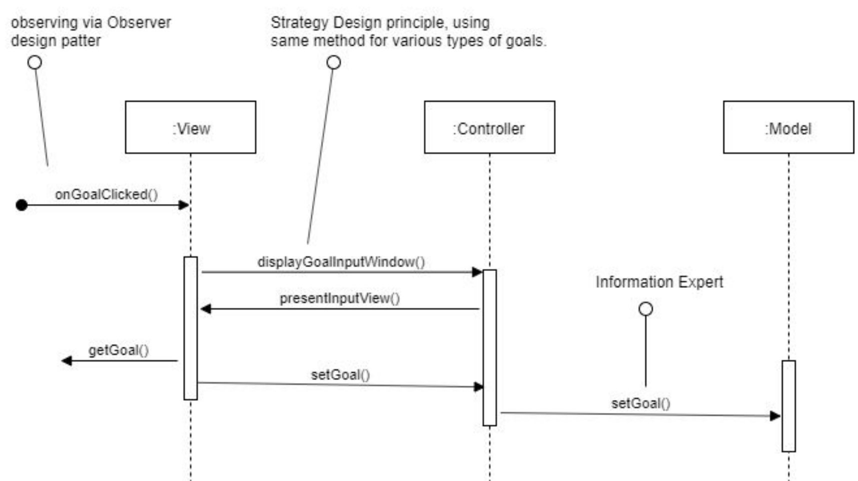
### 4.2.2. Display Settings



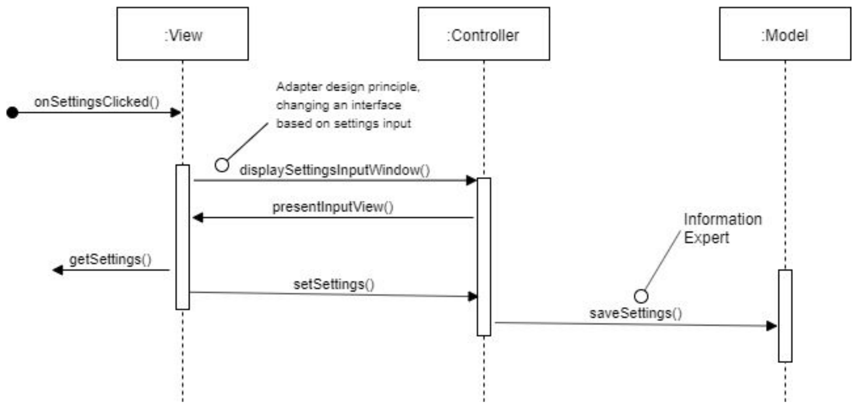

### 4.2.3. Change Time



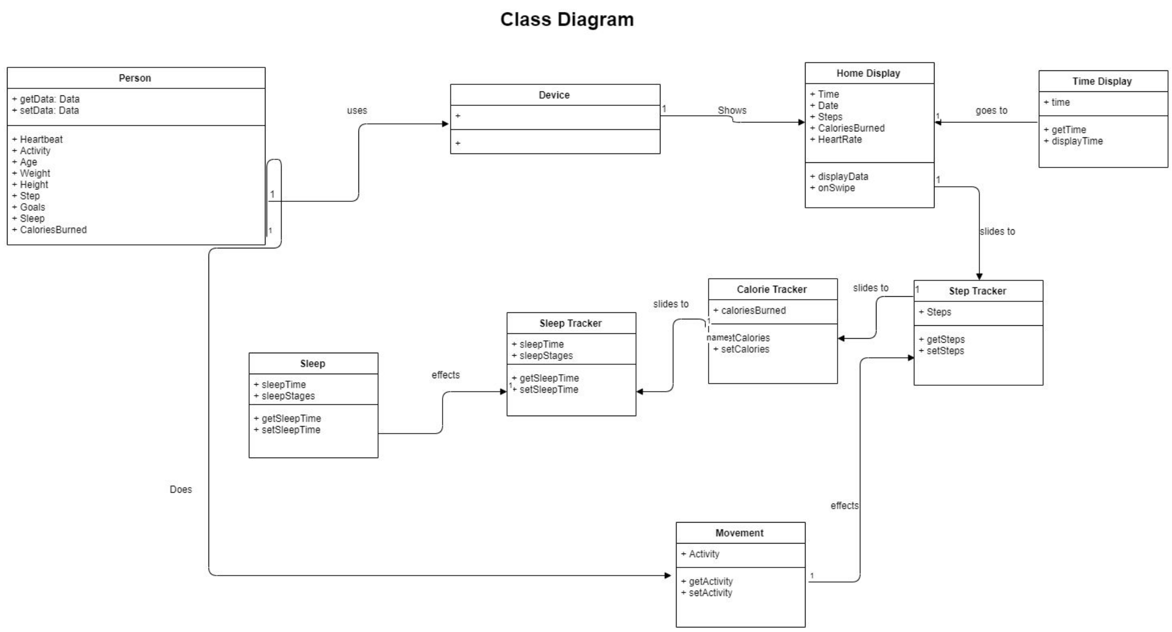

### 4.2.4. Set Goal



### 4.2.5. Input Personal Data

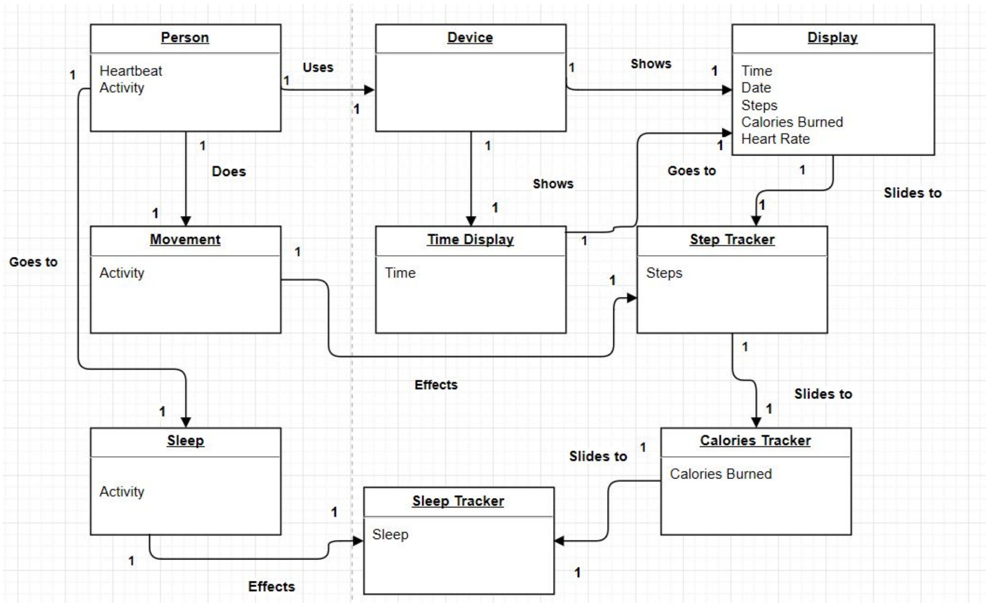



## 4.3. Class Diagram



# 5. Domain Model

## 5.1. Domain Model




## 5.2. Domain Model Description & Justification

This domain model has 9 concepts. It starts with a person (or the user) who both does some type of movement that affects the step tracker program like they decide to go for a jog and then they sleep which affects and sets the sleep tracker into motion. Every now and then the person will want to use the device. Maybe in the middle of the night they want to see just the time, so the person goes to their device and it just shows the time. The person could want an overview of the things things that the device had collected so they go to the main display which shows the time, date, steps, calories burned, and heart rate. If they want more then they will be able to slide and see the step tracker screen which will display their how many steps were made. Next they can slide to the calorie tracker screen that displays how many calories they have burned. They can also slide to the sleep tracker screen to see how long they had slept.

# 6. Bibliography


1. C. Allison, “Fitbit heart rate monitoring explained,” *Wareable*, 10-May-2019. [Online]. Available: https://www.wareable.com/fitbit/fitbit-heart-rate-monitor-guide-330.[Accessed: 02-Sep-2019].
2. C. Lashkari, “How Do Wearables Track Sleep?,” *News Medical Life Sciences*, 27-Feb-2019. [Online]. Available: https://www.news-medical.net/health/How-Do-Wearables-Track-Sleep.aspx. [Accessed: 02-Sep-2019].
3. J. Orlosky, O. Ezenwoye, H. Yates, and G. Besenyi, “A Look at the Security and Privacy of Fitbit as a Health Activity Tracker,” *Proceedings of the 2019 ACM Southeast Conference on ZZZ - ACM SE 19*, pp. 241–244, Apr. 2019.

4. M. Aslam, "Pedometer and Step Counter Mobile Apps : How does it work?," *mohd aslam*, 23-Dec-2013. [Online]. Available: http://mohdaslam.com/pedometer-and-step-counter-mobile-apps-how-does-it-work/.
5. M. Pateman, D. Harrison, P. Marshall, and M. E. Cecchinato, "The Role of Aesthetics and Design: Wearables," *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI 18*, Apr. 2018.
6. R. Gulotta, J. Forlizzi, R. Yang, and M. W. Newman, "Fostering Engagement with Personal Informatics Systems," *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS 16*, pp. 286–300, Jun. 2016.
7. "Fitbit Development: Guides," *Fitbit*, 2019. [Online]. Available: https://dev.fitbit.com/build/guides/. [Accessed: 02-Sep-2019].
8. "Fitbit OS," *Fitbit*, 2019. [Online]. Available: https://www.fitbit.com/fitbitos/. [Accessed: 02-Sep-2019]
9. "Fitbit Technology," *Fitbit*, 2019. [Online]. Available: https://www.fitbit.com/technology. [Accessed: 02-Sep-2019]
10. "Mi Smart Band 4," *Mi*, 2019. [Online]. Available: https://www.mi.com/global/mi-smart-band-4//. [Accessed: 03-Sep-2019]


## Revision History

| Version | Date | Description | Author |
|---|---|---|---|
| 1.1 | 9-14-2019 | Created document | ThunderBits |
| 1.2 | 10-1-2019 | Updated domain model | ThunderBits |
| 1.3 | 10-16-2019 | Updated System Sequence Diagrams, Operation contracts, removed Unlock Screen use case, added stakeholders, captured the following use cases within Display Data: Display Steps, Display Calories, Display Sleep | ThunderBits |
| 1.4 | 11-4-2019 | Added System Diagrams, Class Diagram | ThunderBits |