Activity Tracker Software

Software Architecture Document

ThunderBits

# Table of Contents

[**Table of Contents**](#_ig0snq70ib72) **1**

[**Introduction**](#_by31n94jar3p) **2**

[Vision](#_pqa5brda9xk2) **2**

[Glossary](#_28hadddiyz0i) **2**

[References](#_e5b8bbgg5jlk) **2**

[**Architectural Goals and Constraints**](#_353y3xdkbirm) **5**

[**Supplementary Specifications**](#_b3dvy6eywwnw) **5**

[**Use Case View**](#_12atqorb4k6x) **5**

[Show Time](#_o4s8m2q07zw) **5**

[Show Calories Burned](#_8uydzcb8ubg4) **6**

[Show Steps](#_fqo254pklsl) **7**

[**Bibliography**](#_1e03opxo0tyf) **8**

[**Revision History**](#_b1ffy3t751pd) **9**

# Introduction

## Vision

The vision of this project is to implement an aesthetically pleasing personal health monitor that controls and monitors the users physical activities. The user will be able to see the time and date when the device is powered on and be able to swipe through 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

## 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.

## 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].

# Architectural **Goals and Constraints**

## 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

# **Use Case View**

## Show Time

|  |  |
| --- | --- |
| **Use Case Section** | **Comment** |
| **Use Case Name** | Display time to User |
| **Scope** | Activity Tracker Software |
| **Level** | User Goal |
| **Primary Actor** | User |
| **Stakeholders and Interests** | * User:   + Wants accurate time displayed * Tracker: Accurate time displayed |
| **Preconditions** | * Tracker must be on |
| **Success Guarantee** | * User can see the time whether it’s on their wrist or not. |
| **Main Success Scenario** | * User looks at Tracker. * Tracker displays correct time. |
| **Extensions** | * User looks at Tracker. * A non-time screen is displayed. * User swipes/navigates to time display. * Tracker displays correct time. |
| **Special Requirements** | * Allow user to change the time   + Time zones? * Allow user to switch between Analog or Digital display. |
| **Technology and Data Variations List** | N/A |
| **Frequency of Occurrence** | Continuous |
| **Miscellaneous** | N/A |

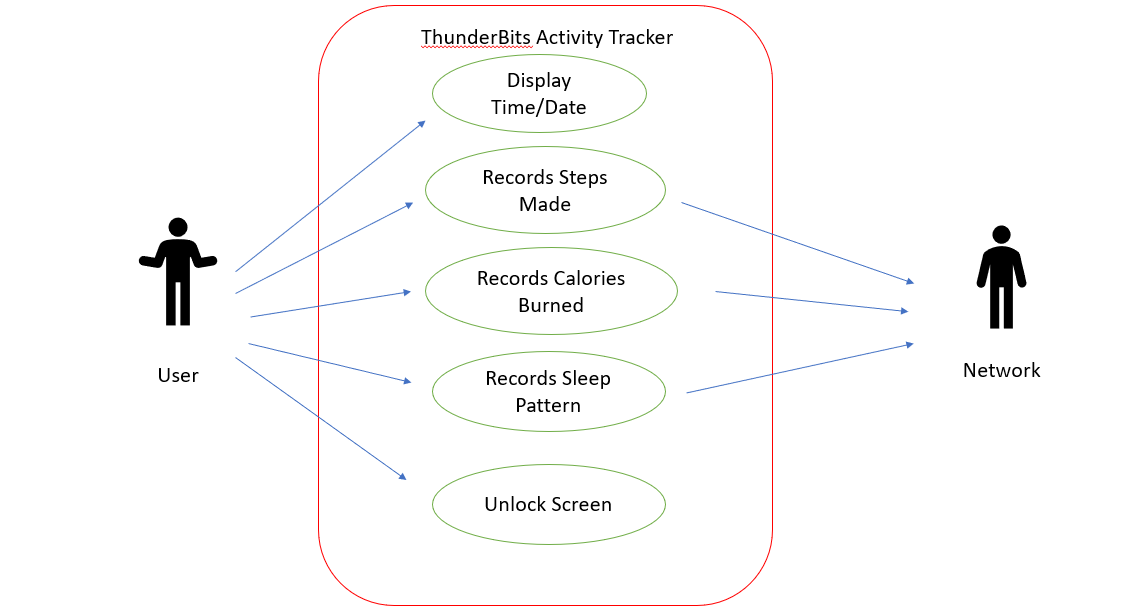
## Show Calories Burned

|  |  |
| --- | --- |
| **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 * 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. 2. Tracker tracks the calories. 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** | 1. At any time, User can swipe to a different display. |
| **Special Requirements** | * User interface with swipe capabilities |
| **Technology and Data Variations List** | 1. Tracker display determined by User swiping through the interface |
| **Frequency of Occurrence** | Could be continuous |
| **Open Issues** | * Explore data protection and security. |

## Show Steps

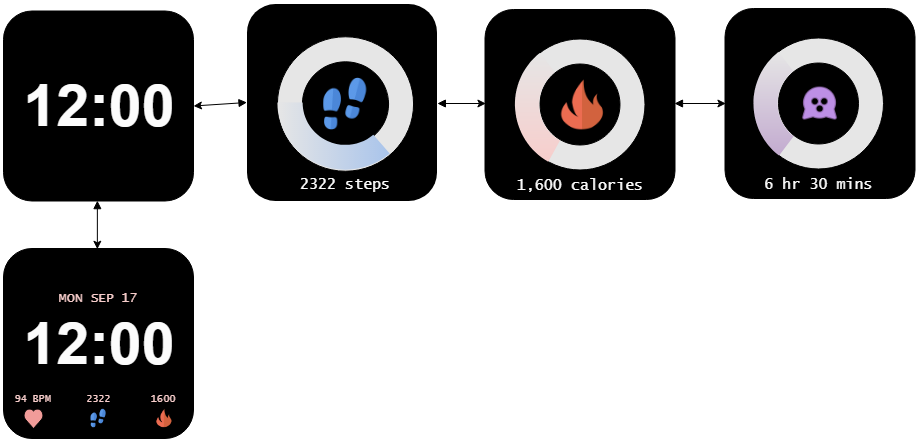
|  |  |
| --- | --- |
| **Use Case Section** | **Comment** |
| **Use Case Name** | Display steps to user |
| **Scope** | ActivityTracker Software |
| **Level** | User Goal |
| **Primary Actor** | User |
| **Stakeholders and Interests** | * User: Wants accurate data for steps presented * Tracker: Wants 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.. 2. Tracker tracks the steps. 3. Software calculates the total daily steps and displays it on the appropriate screen. 4. User checks the display to see how many steps they took throughout the day. |
| **Extensions** | 1. At any time, User can swipe to a different display. |
| **Special Requirements** | * User interface with swipe capabilities |
| **Technology and Data Variations List** | 1. Tracker display determined by User swiping through the interface |
| **Frequency of Occurrence** | Could be continuous |
| **Open Issues** | * Explore data protection and security. |

## Use Case UML Diagram



# Design View

## Overview



# **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 |