**­­StatsWrap™ Software Architecture Document**

Alex Laughlin and Tino Pimentel

**Vision**

For a fitness enthusiast that likes to know every intimate detail of their workout, StatsWrap is an activity tracker that records and graphs all of one’s fitness data in real time. This allows athletes to tailor their workouts with incredible accuracy. Unlike competitors, StatsWrap focuses on keeping your data secure and presenting high fidelity information in a concise live graph.

**Activity Tracker Software FURPS Requirement Model**

**F** – Allows the user(s) to view their activity logs, or reported statistics from the software system. A unique user cannot open multiple sessions. The data being tracked can be reset at any time.

**U** - Software uses a Graphical User Interface to allow users to easily assess with all information provided electronically through a digital screen on the watch. Information on screen is presented clearly with a white background and black text to help colorblind users read their activity logs, etc.

**R** – The software will never fail on its own as long as the watch machine in use has the specified minimum required hardware and software without any lag, hitches, or crashes. Once a user begins using the activity tracker software, their data is stored and calculated within the program. The program can be opened and closed or reset at any given time but will not save any specific user’s data.

**P** – The performance of the activity tracker software can handle many inputs and report several statistics at the same time. Information will be adding to the database of the software dynamically during usage. The software is available to as many watch machines that have the software on them. The maximum throughput is dependent on the speeds of the sensors and processors of the watch machines running the software themselves.

**S** – The usage of java as an object-oriented programming language helps ensure longtime support of our activity tracker software with increasing amounts of devices and maintenance of the code. Developers of the software in the future can add support for different languages and UI screen sizes for when using our software.

**Implementation**

System Requirements:

* + jre 1.8.0\_201
  + Installation of Java 8
  + OS compatible with Java

**Open-Source Software requirements**

The wrist mounted activity tracker system will require Java 8 or newer. Alongside Java 1.8 SE, the system also needs Scenebuilder and JavaFX installed to run the graphical user interface of our software. An operating system such as Windows 7 or higher, Mac OS, or Linux 13.0 installed on the system with our software is recommended for an easier setup.

Gadgetbridge is an open source software for the Android Operating System that can be used in deployment for our activity tracker software to bypass installing previously mentioned Operating Systems, as well as their hardware requirements.

PostgreSQL is an open source database software that would be used to manage the user input data for the activity tracker system. This is an essential component for reporting the graphs of the user stats.

**Purchased Hardware requirements**

The hardware running our activity tracker software must include a light sensor using LED technology. The machine needs to have a minimum of 126 MB of Disk Space for JRE (Java Runtime Engine); 2 MB for Java Update. Minimum of 128MB of RAM. Minimum specified processor to run our software is at least an Intel Pentium 2 266 MHz processor or a comparable processor in performance. A micro-USB cable is required to setup our software initially onto the machine before usage.

**Legal Issues**

**Activity Tracker Inaccuracies:**

One potential is the data being read improperly from the light emitting diodes (LEDs). The results presented on the activity tracker software would be incorrect and may prove unreliable. Customers should not rely on this product to diagnose or prevent diseases.

**Inconvenience of Usage:**

Users of an activity tracker must not strain their eyes looking for information. Users should not use this product while operating heavy machinery or otherwise distracted. Users may receive notifications during activities in which they should not engage with the system. Software should display information concisely to convey information at a glance.

**Glossary**

Account Information - The information stored within the activity tracker software which may include activity logs, alarms, and heartrate.

Activity Log(s) **-** The logged data of all sensors within the Activity tracker

Alarm(s) **–** time-based notifications

Application **–** software designed and written to fulfill a specific task

Calorie **–** the amount of energy required to raise 1 gram of water 1 degree Celsius

Display **-** the area of the watch capable of displaying data

Device API **-** a set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

Component **–** an aspect of a larger entity

Database **–** a structured set of data held in a computer, especially one that is accessible in various ways

Settings **–** options to change how the system looks, works, etc.

Heartrate **–** number of heart beats per minute (bpm)

LED **–** abbreviation for Light emitting diode.

Gait **–** an individual’s manner of walking

Plateau **–** a period of stagnation in a workout regimen, diminished returns on workouts

**Use Cases**

***Use Case (UC1):***

View Graphs

**Scope:**

The current User using the activity tracker software

**Level:**

To allow a user to view their graphs in real time

**Actor:**

User, System

**Stakeholders:**

User: Wants to be able to view their graphs accordingly and properly.

**Preconditions:**

Data exists for system to display graphs

**Postconditions:**

System displays graphs for user, then returns to default screen when user exits.

**Main Success Scenario:**

(UC1main):

1. User chooses view graphs

2. System collects necessary data

3.  System generates live graphs

4. User can easily select which graph to view

5. User can swap graphs with a single button press

6. System updates charts as new data is retrieved

**Extensions:**

(UC1a): UI does not allow for easy graph selection during exercise

(UC1b): System generates graphs with incorrect data

(UC1c): Graphs are not easily readable (unlabeled axes, bad scale, etc.)

(UC1d): Graphs do not update with live data

**Special Requirements:**

User must have logged some data in order to view graphs

**Variations in Tech and Data:**

Users all have unique data, tech is consistent

**Frequency of Occurrence:\**

Many times a day

**Miscellaneous:**

N/A

***Use Case (UC2):***

Set New Activity Goal

**Scope:**

The current User using the activity tracker software

**Level:**

User chooses a new goal value and timeline on which to achieve said goal

**Actor:**

User, System

**Stakeholders:**

User: Wants to be able to set a new activity goal

**Preconditions:**

User must know the goal and timeline to enter the data

**Postconditions:**

System must graph goal line based on input values

**Main Success Scenario:**

(UC2main):

1. User selects “Set New Goal”
2. System prompts user to choose which category to set a new goal in
3. User selects a category (Sleep, steps, heart rate, etc.)
4. System asks user what the new goal should be
5. System asks user when the user wants to achieve this goal by
6. User selects a timeline on which to complete the goal
7. System records new goal and timeline
8. System displays live chart with goal line graphed

**Extensions:**

(UC2a): System does not allow for goal setting

(UC2b): System fails to capture necessary data to graph goal line

(UC2c): Graph is cluttered, incorrect, or unreadable

(UC2d): Goal line is not distinguished from actual values

(UC2e): System does not update live charts with goal line

**Special Requirements:**

User must have a goal in mind and timeline

**Variations in Tech and Data:**

All data is unique to users, tech is consistent

**Frequency of Occurrence:**

Varies, infrequently to multiple times per day

**Miscellaneous:**

N/A

***Use Case (UC3):***

Stop recording data

**Scope:**

The current User using the activity tracker software

**Level:**

User selects stop recording from the interface

**Actor:**

User, System

**Stakeholders:**

User: Wants to be able to stop recording data

**Preconditions:**

User must be already recording data for the stop recording data button to be present

**Postconditions:**

System must not log any data from sensors

**Main Success Scenario:**

(UC3main):

1. User selects “Stop Recording Data”
2. The system alerts the array sensor to stop recording data
3. The system alerts the user that no data is being recorded
4. An icon that represents the Stop Recording Data state is persistent on the interface while the system is not recording data

**Extensions:**

(UC3a): System fails to stop recording data

(UC3b): System fails to notify user that “Stop Recording Data” button press was successful

(UC3c): Icon notifying user of not recording data is not persistent on screen

**Special Requirements:**

User must already be recording data in order for “Stop Recording Data” button to be present

**Variations in Tech and Data:**

All data is unique to users, tech is consistent

**Frequency of Occurrence:**

Varies, infrequently to multiple times per day

**Miscellaneous:**

N/A

***Use Case (UC4):***

Log a new session

**Scope:**

The current User using the activity tracker software

**Level:**

User selects “New Session” from main menu

**Actor:**

User, System

**Stakeholders:**

User: Wants to be able to record new activity session

**Preconditions:**

User must have set up the activity tracker by inputting their basic data (height, weight, resting heart rate)

**Postconditions:**

System must have initialized a new session

**Main Success Scenario:**

(UC4main):

1. User selects “New Session” from main menu
2. System prompts user for name of session
3. System prompts user for which sensors to record data from
4. System prompts user with “Begin Session” and “Cancel” options
5. User selects “Begin Session” and the session is initialized

**Extensions:**

(UC4a): System fails register button press

(UC4b): System fails to prompt user with name of session and data to record

(UC4c): System adds data to existing session or does not record data in new session

**Special Requirements:**

User must have set up basic information (height, weight, age, resting heart rate)

**Variations in Tech and Data:**

All data is unique to users, tech is consistent

**Frequency of Occurrence:**

Varies, infrequently to multiple times per day

**Miscellaneous:**

N/A

***Use Case (UC5):***

View data

**Scope:**

The current User using the activity tracker software

**Level:**

User selects “View Data” from main menu

**Actor:**

User, System

**Stakeholders:**

User: Wants to be able to view logged data

**Preconditions:**

User must have some data logged in the System

**Postconditions:**

System must not have changed any of the data

**Main Success Scenario:**

(UC5main):

* + - 1. User selects “View Data” from main menu
      2. System prompts User for which category they would like to view
      3. User selects a category
      4. System displays data from specified category

**Extensions:**

(UC5a): System fails to register button press

(UC5b): System fails to prompt user with category options

(UC5c): System fails to display or displays incorrect data

**Special Requirements:**

User must have some data logged in System

**Variations in Tech and Data:**

All data is unique to users, tech is consistent

**Frequency of Occurrence:**

Varies, infrequently to multiple times per day

**Miscellaneous:**

N/A

***Use Case (UC6):***

Input data

**Scope:**

The current User using the activity tracker software

**Level:**

User selects “Input Data” from main menu

**Actor:**

User, System

**Stakeholders:**

User: Wants to be able to input new data

**Preconditions:**

User must have set up basic information (age, weight, resting heart rate)

**Postconditions:**

System must have recorded new data input

**Main Success Scenario:**

(UC6main):

1. User selects “Input Data” from main menu
2. System prompts User for name of new data to be entered
3. User selects a name
4. User inputs data to be logged

**Extensions:**

(UC6a): System does not allow new data to be input

(UC6b): System fails to store data input by User

(UC6c): System does not allow access to data after entry

**Special Requirements:**

User must have set up basic information (height, weight, age, resting heart rate)

**Variations in Tech and Data:**

All data is unique to users, tech is consistent

**Frequency of Occurrence:**

Varies, infrequently to multiple times per day

**Miscellaneous:**

N/A

**UML Diagrams**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA close up of a piece of paper

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated**

**Contracts**

**Contract CO1: viewGraphs**

**Operation:**

viewGraphs()

**Cross References:**

Use Cases: View Graphs

**Preconditions:**

User must have some data for System to graph

**Postconditions:**

**-**Several Graph instances were created *(instance creation)*

-The Graphs are representations of the Data *(association formed)*

-The Graphs are displayed on the User Interface

**Contract CO2: Set Activity Goal**

**Operation:**

setNewGoal(goalType, goalValue, goalDate)

**Cross References:**

Use Cases: Set New Activity Goal

**Preconditions:**

-None

**Postconditions:**

**-**A Goal instance was created *(instance creation)*

-The System has assigned the Goal instance a type, value, and end date *(association formed)*

-A Graph instance was created (*instance creation)*

-The Graph is a representation of the Goal (*association formed)*

**Contract CO3: Stop Recording Data**

**Operation:**

stopRecordingData()

**Cross References:**

Use Cases: Stop Recording Data

**Preconditions:**

-The sensor array is recording data

**Postconditions:**

**-**The System ceased to record data *(attribute modification)*

-The System has displayed a confirmation that data is not being recorded

**Contract CO4: Log New Session**

**Operation:**

newSession(name, sensors)

**Cross References:**

Use Cases: Log New Session

**Preconditions:**

- User must have set up the activity tracker by inputting their basic data (height, weight, resting heart rate)

**Postconditions:**

**-**The System has initialized a new session instance (*instance creation)*

-The System has associated the name and sensors to record with to the Session instance (*association formed)*

**Contract CO5: View Data**

**Operation:**

viewData(type)

**Cross References:**

Use Cases: View Data

**Preconditions:**

- User must have some data

**Postconditions:**

**-**The System displayed the data from the Activity Logs (*association formed)*

-The System did not modify the data

**Contract CO6: Input Data**

**Operation:**

inputData(name, value)

**Cross References:**

Use Cases: Input Data

**Preconditions:**

- None

**Postconditions:**

**-**The System created a new Data instance (*instance creation)*

-The System created a new Graph instance *(instance creation)*

-The System populated the Graph instance with the Data instance (*association formed)*

Works Cited

Fitbit­™. “Fitbit Development: Glossary Guide.” *Fitbit*, dev.fitbit.com/build/guides/glossary/.

Fung, Brian. “Is Your Fitbit Wrong? One Woman Argued Hers Was - and Almost Ended up in a Legal No-Man's Land.” *The Washington Post*, WP Company, 2 Aug. 2018, beta.washingtonpost.com/technology/2018/08/02/is-your-fitbit-wrong-one-woman-argued-it-was-almost-ended-up-legal-no-mans-land/.

“Gadgetbridge Alternatives and Similar Apps.” *AlternativeTo*, 26 May 2019, alternativeto.net/software/gadgetbridge/.

Kay, Melissa. “Here's What To Know So You Don't Buy The Wrong Fitness Tracker.” *Gearbrain*, Gearbrain, 16 June 2016, www.gearbrain.com/fitness-tracker-glossary-terms-1862551889.html.

The PostgreSQL Global Development Group. “About.” *PostgreSQL*, 2019, www.postgresql.org/about/.