Logan Sitar & Ryan Ratliffe

31 March 2020

CSCI 360

**Vision**:

Purpose: The purpose of our project is to address the problems surrounding the design of a wearable wrist mounted activity tracker.

Based on the demand for fitness tracking software in wearable wrist mounted devices we are deciding to implement our own solution to this problem; the problem of monitoring activities and sleep while still having the basic functionality of a watch, displaying the time and date. We believe this software will enhance usability for the user in terms of basic functionality and in terms of the fitness tracker. Since we are the significant stakeholders, our vested interest is due to our need to achieve a good grade on the project. Our other stakeholders are the customer, their vested interest lies in receiving a new piece of software that perfectly solves the problem at hand. Any of the stakeholders will be satisfied upon the solving the initial problem stated above. As the developer, our satisfaction as stakeholders will be dampened if the grade we receive on the project is poor. The customer’s satisfaction as a stakeholder will be dampened if their software is poorly functioning and doesn’t fit in seamlessly with their daily lives.

**Use Cases:**

Use case UC1: Tracking steps

Primary Actor: Activity Tracker User

Stakeholders and Interests: User: Wants to easily record and access their step counter.

Preconditions:

User possesses a wrist-mounted activity tracker with appropriate sensors.

The user has set up their device with personal data.

Success Guarantee (Postconditions): Each step is accurately recorded and made accessible

Main success scenario (or basic flow):

The user turns on their activity tracker.

The sensor identifies each step.

The device increments the step counter for the user to see every step.

The device stops incrementing the step counter once it recognizes the user is sleeping.

The number of steps for wake cycle are saved.

The step counter is reset once sleep is recognized, unless the user wants to prematurely

reset with the reset button.

After a sleep cycle, the step counter resumes.

Extensions (or alternative flows):

If steps aren’t being reported consistently, indicate error.

If memory is nearing full capacity, indicate error.

Special requirements: Touch screen UI

Technology and Data Variations List: Steps are recorded by an algorithm that utilizes sensor data.

Open issues: Ensuring the user’s personal data is securely stored.

Use case UC2: Tracking sleep  
Primary Actor: Activity Tracker User  
Stakeholders and Interests: User: Wants to easily record and access their sleep patterns.  
Preconditions:  
 User possesses a wrist-mounted activity tracker with appropriate sensors.   
    The user set up their device with personal data.   
    The user does not have a sleep disorder.  
Success Guarantee (Postconditions): Sleep cycle is recorded accurately.  
Main success scenario (or basic flow):  
    The sensor identifies a lack of steps and recognizes the start of a sleep cycle.  
    The device begins a sleep timer.  
    The device stops incrementing the timer once it recognizes the user is awake.  
    The elapsed time of the users sleep cycle is saved.  
Extensions (or alternative flows):  
    If the user wakes up in the middle of the night and goes directly back to sleep, indicate error.  
    If memory is nearing full capacity, indicate error.  
Special requirements: Touch screen UI, screen dimmer  
Technology and Data Variations List: Elapsed sleep time is recorded by an algorithm that utilizes sensor data.  
Open issues: Ensuring the user’s personal data is securely stored.

Use case UC3: Display Running Time

Primary Actor: Activity Tracker User

Stakeholders and Interests: User: needs to be able to see an accurate and clear reading of time.

Preconditions:

User has purchased a compatible wrist mounted wearable activity tracker.

Software is set up with system time.

Success Guarantee (Postconditions): The time is displayed to the user in either military or standard time.

Main success scenario (or basic flow):

The user turns on the device.

The user inputs their current time in their desired format.

The device displays and constantly updates the time to the user in hours and minutes.

Extensions (or alternative flows): If user inputs invalid time, set the time to 0:00.

Special requirements: Touch Screen UI, clear visibility

Technology and Data Variations List: hardware keeps consistent time reading, even with no connection

Open issues: N/A

Use case UC4: Enter Personal Information

Primary Actor: Activity Tracker User

Stakeholders and Interests: User: user needs to enter accurate information so their results are accurate as well.

Preconditions:

User has purchased a compatible wrist mounted wearable activity tracker.

The users time preferences have been set.

Success Guarantee (Postconditions): The user successfully enters their age, height, weight, gender, and personal goals.

Main success scenario (or basic flow):

The device is powered on.

The user enters their age, weight, and height with up and down arrows.

The user chooses either male, female or other for their gender.

The user enters their goal number of steps per day.

The user enters their goal hours of sleep they want per night.

Extensions (or alternative flows): If the user enters unrealistic values, produce an error message.

Special requirements: Touch Screen UI, wide range of user input

Technology and Data Variations List: N/A

Open issues: The system securely stores and monitors the user’s personal data.

Use case UC5: Connect to device via bluetooth

Primary Actor: Activity Tracker User, Connected Device

Stakeholders and Interests: User: user needs to connect to bluetooth in order to store data.

Preconditions:

User has a device that is bluetooth compatible.

User possesses a wearable wrist mounted activity tracker.

Bluetooth enabled on the external device.

Success Guarantee (Postconditions): The activity tracker connects to the desired device via bluetooth.

Main success scenario (or basic flow):

    The user turns on the activity tracker and external device.

    The user enables bluetooth pairing on the tracker.

    The user navigates through the settings to the connectivity menu.

    The user chooses an external device to connect to.

    The devices are paired.

Extensions (or alternative flows):

If the device doesn’t pop up, restart both devices.

If the connection is weak, prompt the user to move devices closer together.

Special requirements: Touch Screen UI, bluetooth capability

Technology and Data Variations List: bluetooth chip

Open issues: User’s data must be securely passed from one device to another.

Use case UC6: Track Calories Burned

Primary Actor: Activity Tracker User

Stakeholders and Interests: User: user needs to see how many calories he burns per day.

Preconditions:

User possesses a wearable wrist mounted activity tracker.

User has entered all required personal data.

Success Guarantee (Postconditions): The user can see how many calories he has burned in a day.

Main success scenario (or basic flow):

    The user powers on his device.

    The user takes steps.

    The calories burned are calculated based on number of steps and personal data.

    Calories burned are displayed to the user.

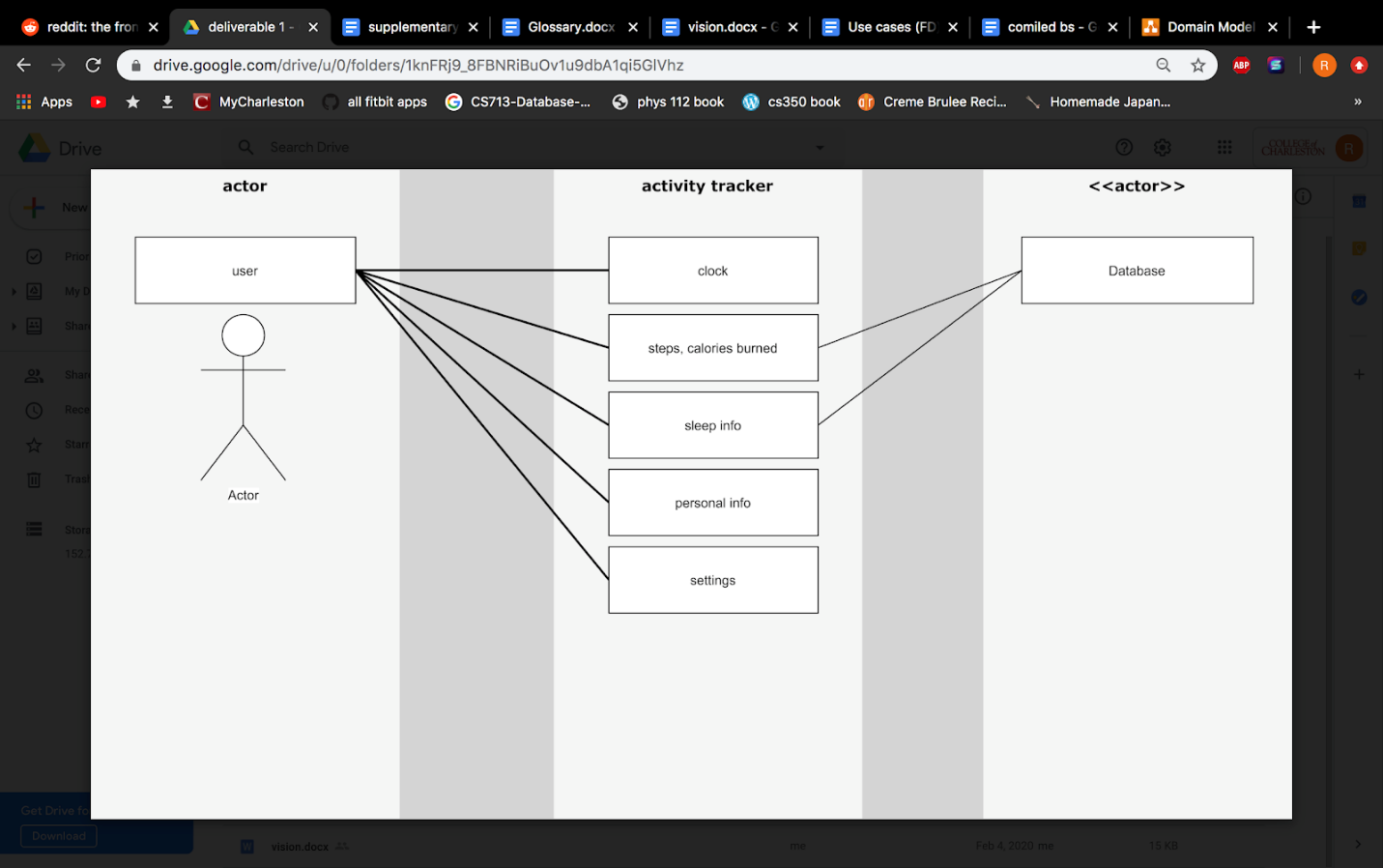
Extensions (or alternative flows): The calculation output is invalid, report calculation error.

Special requirements: Touch Screen UI

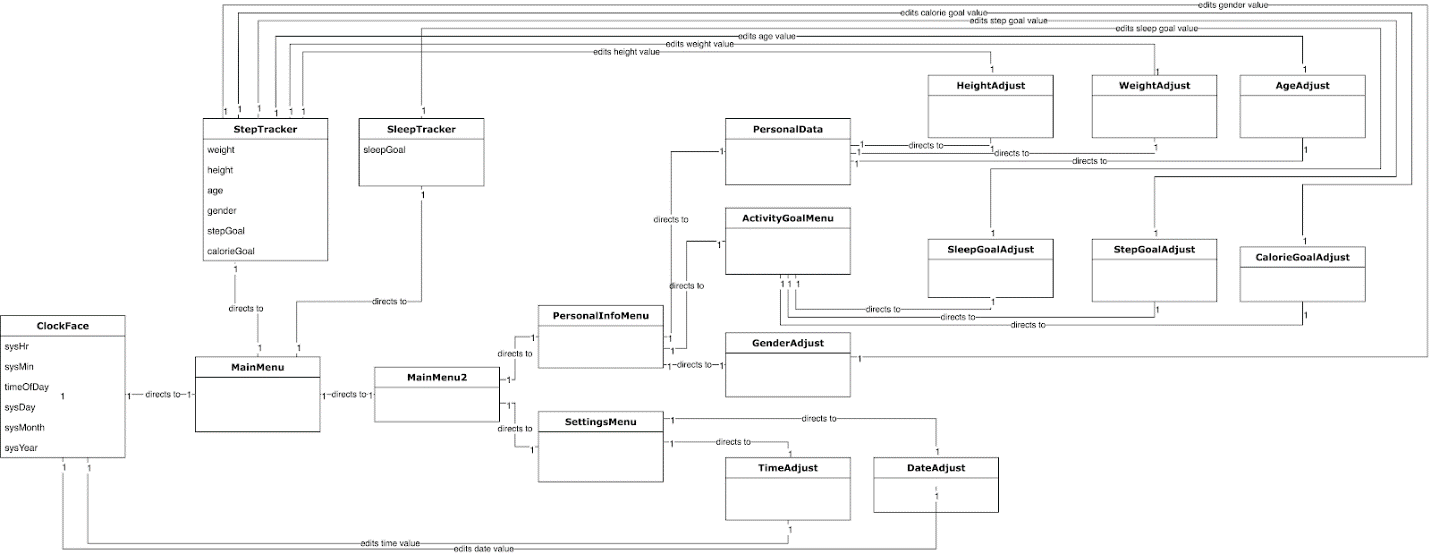
Technology and Data Variations List: algorithms to read and perform operations based off sensor data to determine calories

Open issues: User’s data must be securely stored.

**UML Use Case Diagram:**



**Domain Model Diagram:**

****

**Supplementary Specs:**

Functionality: display/keep clock time, track/access steps, calculate the number of calories burned based on personal entered info, track/access sleep patterns, internet connectivity

Usability: fluidly switch between scenes, access data

Reliability: no memory leaks, storage consuming, accurate sensor readings

Performance: responsive touch interface, seamless transition between activities

supportability + implementation constraints: consistently and easily updated software

purchased components: licensing and legal docs, adobe Illustrator

free open source components: java, JavaFX

interfaces – noteworthy hardware and software interfaces: Eclipse

legal issues: copyright of code, data privacy and security, GNU License

information in domains of interest: the software fulfills a need in the current market

**Glossary:**

Activity Tracker(AT): A wearable device or a computer application that records a person’s daily physical activity, together with other data relating to their fitness or health, such as the number of calories burned, heart rate, etc.

Dimmer: Software that alters the visual perception of something to be less bright

Domain of interest(DOI): A domain is a particular field of thought, activity, or interest, especially one over which someone has control, influence, or rights.

Eclipse: A platform that has been designed from the ground up for building integrated web and application development tooling.

Functionality: The sum or any aspect of what a product, such as a software application or computing device, can do for a user.

General Public License (GNU): A widely used free software license that guarantees end users the freedom to run, study, share, and modify the software.

Graphical User Interface (GUI): A system of interactive visual components for computer [software](https://www.computerhope.com/jargon/s/software.htm). A GUI displays objects that convey information and represent actions that can be taken by the user.

Interface: A device or program enabling a user to communicate with a computer.

Memory Leak: A failure in a program to release discarded memory, causing impaired performance or failure.

Sleep Cycle(SC): A part of our [internal biological “clocks”](https://www.sleepcycle.com/circadian-rhythm/) the regularly occurring patterns of brain waves which occur while we sleep. Sleep cycles typically last around ninety minutes to two hours.

Sleep Disorder(SD): Any disorder that affects, disrupts, or involves sleep. The most common sleep disorder is probably snoring, although it is usually not medically significant. Insomnia, sleep apnea, restless leg syndrome, and sleepwalking are also sleep disorders.

Stakeholder: A person, group or company that is directly or indirectly involved in the project and who may affect or get affected by the outcome of the project.

User Interface (UI): The means by which the user and a computer system interact, in particular the use of input devices and software

**System Sequence Diagrams:**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

\

**Operation Contracts:**

Operation Contract Name: onSelectDeviceButton()  
Cross Reference: Use Case Connectivity:  
Pre-Condition(s): bluetooth connectivity is enabled  
Post-Condition(s):  
- instantiation of a new connected bluetooth device  
- the bluetooth device is associated with the user

Operation Contract Name: setOnAction(StepTracker: class)  
Cross Reference: Use Case Step Tracker  
Pre-Condition(s): the user has already recorded data on the tracker  
Post-Condition(s):   
- A scene is created in Step Tracker class  
- StepTracker.data is set to data.  
- data displayed to user updated  
- data is associated with goals

Operation Contract Name: setOnAction(SleepTracker: class)  
Cross Reference: Use Case Sleep Tracker  
Pre-Condition(s): the user has recorded at least one sleep pattern  
Post-Condition(s):   
- A scene is created in Sleep Tracker class  
- SleepTracker.currentPattern is set to pattern.  
- pattern displayed to user updated  
- pattern is associated with goals

Operation Contract Name: setOnAction(DesiredPersonalInformationClass)  
Cross Reference: Use Case Personal Data  
Pre-Condition(s): User has desired activity wanting to be tracked  
Post-Condition(s):  
- StepTracker.height modified to height associated with settings  
- StepTracker.weight modified to weight associated with settings  
- StepTracker.age modified to age associated with settings  
- StepTracker.gender modified to gender associated with settings  
- StepTracker.goal modified to goal associated with settings  
- SleepTracker.goal modified to goal associated with settings

Operation Contract Name: updateTime()  
Cross Reference: Use Case Display Time  
Pre-Condition(s): user has set time and preferred format  
Post-Condition(s):  
- time is instantiated in ClockFace  
- clockFace.time is set to time  
- time is associated with settings

**Class Diagram:**

A screenshot of a cell phone

Description automatically generated

**Sequence Diagrams:**

A screenshot of a computer

Description automatically generated

All classes are creators because each class creates another activity or class.

**Design Justification**

Our design for our Fitness Tracker’s UI is very simple, we did this because first and foremost, our priority is to be able to read it no matter the circumstances. For our navigation we prioritized being able to fit on any screen your device may have; the buttons are few enough in numbers where even downsizing the screen to older smart watches with smaller screens won’t affect the navigability of the software. We also prioritized being able to easily access the functions of the software you most use, in day to day operation after setup you should only have to click two buttons to see the fitness activity you’ve performed. Our software is inspired by some of the first smartwatches the industry saw, from pebbles to fitbits they all started off with very simple blocky screens that were easy to navigate and required no help of manuals due to its usability. This mentality is easily seen in all screens of the software, with no more than 5 buttons on any screen, easily legible blocky text, and high contrast color, this software is meant for easy seamless use.