CSCI 360 Project
Fall 2019
Instructor: Dr. X (Xenia Mountrouidou)
Activity Tracker
Note: this is a live document, subject to changes. Please check frequently

1. Introduction

The Internet of Things (IoT) has entered our lives for the past years with several cool gadgets that are connected to the Internet. One of the most common is the personal health monitor and Fitbit is one of the most well-known brands. For this semester project you will need to analyze and build software to control a personal activity tracker, similar to a Fitbit. Your software solution should continuously monitor a person's physical activities and vital signs, and motivate people to be more physically active. The final product of this project will be a prototype of a activity tracker software and User Interface, as well as a complete design document with use cases, UML sequence, class, activity, and state diagrams.

You will work in groups of minimum two, maximum four students. A good publication on group collaboration that I recommend for you to read is titled: "All I Really Need to Know about Pair Programming I Learned In Kindergarten". *Keep in mind that you will submit a team review document in the end of the semester. Project work needs to be completed by all partners equally*. Furthermore, you will have a project presentation

where I can ask detailed questions about your work and figure out if you have worked equally. Finally, the whole class will have the same project and plagiarism is impossible in a project like this and will be easily discovered. The project will be uniquely built based on your questions and interviews that you will conduct with the client (me ☺).

2. System Objectives

2.1. The activity tracker system will be designed for operation on a personal computer. It will not be tested on an actual activity tracker device.
2.2. The activity tracker system will abide to the legal, regulatory, behavioral, and sociological aspects of devices of its type.
2.3. The system will not communicate with a mobile application. This operation is left for future expansions.
2.4. The system will show time, activity (steps), and sleep patterns. All inputs will be added by internal simulation pseudo-random counters, not actual sensors.

3. Functional requirements

3.1. Show time
  3.1.1. It must be easy for an individual to check the clock time
  3.1.2. An individual must be able to change the clock time on the activity tracker
3.2. Track steps
  3.2.1. The user must give their data: age, weight, goal, etc.
  3.2.2. The user must be able to retrieve:
    3.2.2.1. Number of steps
    3.2.2.2. Calories burned
3.3. Sleep patterns
  3.3.1. The device should be able to indicate the sleep time and patterns
  3.3.2. The user should be able to retrieve this information easily
3.4. Connection to the internet: the device should be able to send data over a network connection securely.
3.5. Additional features: any additional features will be considered for extra credit

4. Non-functional requirements

4.1. Security
4.2. Performance
4.3. Reliability
4.4. Usability

5. Deliverables

5.1. **Literature review (5 pts):** Literature review (5 pts): This is the part where you need to "do your homework" on what already exists out there before you discuss with the client. You will need to research different sources (publications, blogs, product specifications) and write a proper literature review with a paragraph summary for each source and proper citations (IEEE style).

Submit your work on Oaks and on Github! My github handle is: **mundruid** for you to share your repo with me.

Length: You will need to cite at least 10 different sources. Describe each source briefly and include what it contributes to the solution of the problem, how it contributes this, and any other information that it may give on the design and implementation of election software.

5.2. **D1 - Inception (10 pts):** For this deliverable, you will need to submit the following inception documents for your software such as: Vision, Use-Case enumeration with 2 use cases "fully dressed", use case UML diagram, supplemental specs, and a glossary.

Code: You will need to start working on the design of your User Interface (UI). You may use powerpoint or draw.io. You need to have a good design, multiple screens, clean buttons etc. You may start coding the basic view using Java FX and screen builder. Code submission in this case is not required but encouraged.

Demo: short (min 7 mins, max 15 mins) presentation of vision, use cases, GUI prototype, and any other items you would like to add.

5.3. **D2 - Elaboration (10 pts)**: Submit all use cases fully dressed and domain model. Include a short description of the domain diagram and justification of its components.

5.4.

Code: Code submission will be part of your grade this time. Submit an initial version of your View (UI) using Java FX. Your code will be graded.

Documentation and Revisions: Collect all your documents: vision, supplemental specs, glossary, use case diagram, domain model diagram, and all use cases fully dressed, in **one document,** the Software Architecture Document. At this point you may start revising your previous deliverables, such as use case diagram etc.

Naming and tools: You will need to use **GitHub** for version control**.** You will need to add me to your repo (my github handle is **mundruid**). Include your documentation in the repository. Use Eclipse or IntelliJ to create project named: "TeamMateLastNames-ActivityTracker". Use the package structure com.csci360.activitytracker.

5.5. **D3 - Elaboration (10 pts)**: Submit all the system sequence diagrams and operation contracts required for your election software. You may add these diagrams to your SAD (Software Architecture Document) and submit a single document. Add any revisions you may think are necessary based on my past feedback.

Code: Submit an improved version of your View (UI) using Java FX. Start coding the controller and model parts of the project. You may submit blueprints of classes. *You will need to have at least 30% of your classes fully implemented*.

Demo: short (min 7 mins, max 15 mins) presentation of progress with UI, SSDs, SDs, and any other diagrams you think are worth demonstrating.

5.6. **D4 - Objects (20 pts)**: Identify objects needed to implement your operations. Submit a class diagram for the voting system. Suggest object responsibilities and collaborations. Submit sequence diagrams that include labels that indicate all the object collaborations based on GRASP. Include a short justification of your design choices.

Complete the object design, revise sequence diagrams. Note: it is important to apply the GRASP principles, especially GoF Adapter, Factory, Facade, Singleton, Strategy, observer and GRASP design principles such as Information Expert, Creator, Controller, Polymorphism, etc.

Code: Submit any revisions on View (UI) using Java FX. Populate classes; *at least 60%* of classes should be complete. Include Junit tests for *all classes*. Think about improving the way you store data (file vs database) and include this in your object design. Address important non-functional requirement of security especially when you store sensitive data.

5.7. **D5 - Java code and Testing (65 pts)**: Finalize your code and submit all the test cases that you created using Junit or any other tool. Submit any other test cases that you would like me to run.

What to submit:

1. README file that gives detailed instructions on how to run your code.
2. Create a jar file and submit this on Oaks. Instructions how to create a Jar file using
   eclipse: https://www.cs.utexas.edu/~scottm/cs307/handouts/Eclipse%20Help/jarInEclipse.htm. Instructions how to create a Jar file using
   IntelliJ: https://blog.jetbrains.com/idea/2010/08/quickly-create-jar-artifact/.
3. A zip folder with your code. That is just to make sure that I have a backup with your latest version of the code.
4. The final Software Architecture Document.
5. The powerpoint slides of your presentation.

5.8. **Powerpoint final presentation & Demo (10 pts)**: You will need to present your work in class. This will be a 15 min presentation with 5 mins left for Q&A. The presentation should have professional slides that will present briefly your requirements, analysis, system design, objects, and a demo your prototype.

Tips:
Collaboration tools:
https://www.teamwork.com/,
https://trello.com/


I HACKED INTO YOUR FITNESS BAND AND ANALYZED YOUR DECISION-MAKING UNDER DIFFERENT CONDITIONS.
Dilbert.com @ScottAdamsSays
WHEN YOU ARE HUNGRY, TIRED, OR STRESSED, YOU MAKE TERRIBLE DECISIONS.
9-27-16 © 2016 Scott Adams, Inc. /Dist. by Universal Uclick
HOW OFTEN IS THAT?
ONLY WHEN YOU'RE AWAKE.