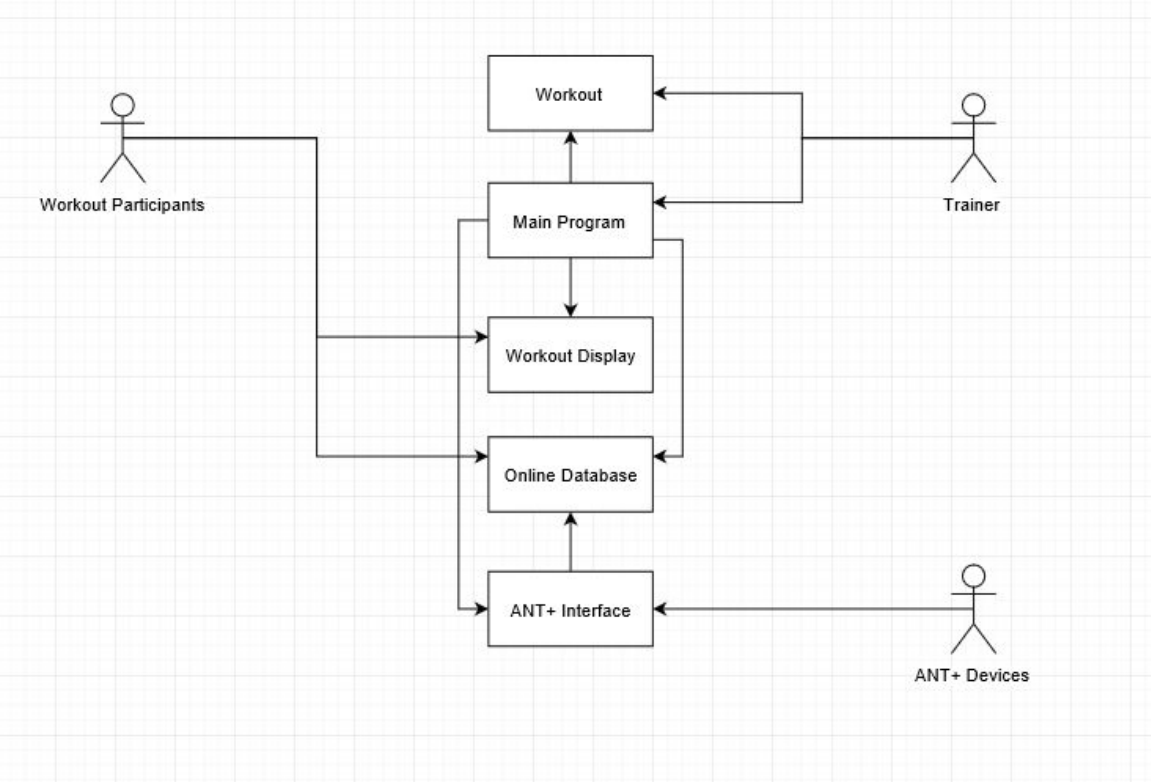
Nathan Franklin, Josh Baker - Group 12: Trainer Tuesday
GitHub: https://github.com/jrb688/Group12_Trainer_Tuesday
D2.2 - Use Cases
Spring 2017 - CS386 Software Engineering - Marco Gerosa
Submitted 3/09/2017

Group 12
Use-Case diagram for whole system



Group 12
Use-Case: Build Workout

1 Brief Description

This use case describes how the trainer ride host uses the program to build a workout

2 Actor Brief Descriptions

2.1 Person designing the workout

3 Preconditions

The option to make a new workout has been chosen from the program's main menu

4 Basic Flow of Events

1. Person enters a name for the workout
2. Person enters description of the workout
3. Person adds segment to workout
4. Person adds more segments to the workout
5. Person saves the workout
6. Program returns to main menu
7. The use case ends.

5 Alternative Flows

5.1 User cancels workout creation

1. If at any point in steps 1-5 the user cancels the workout creation, then
2. The use case resumes at step 6.

6 Subflows

7 Key Scenarios

7.1 Workout created

8 Post-conditions

8.1 Workout saved

The workout is saved in a format that can be read into the program if that workout is selected to run.

9 Special Requirements

9.1 Workout must be saved in order to be used by the system.

Group 12
Use-Case: ANT+ Channel Controller

1 Brief Description

Each ANT+ usb stick of type ‘M’ or type ‘2’ has 8 channels. Each channel can can connect to one ANT+ device. An ANT+ device will send out data for whatever to receive. The ANT+ enabled bike sensor is the master, and the ANT+ USB stick channels are the slaves. multiple slaves can connect to one master. This means one part of the program has to control all of the ANT+ USB sticks and channels.

The program will display data for power meters, cadence and heart rate. Some people have all three sensors, some people only have one, and some people have no sensors.

Each user’s profile will have the serial number for their primary sensor, and their secondary sensors. If someone only brought their secondary sensor, they should still get to be displayed on the screen. If their secondary sensor is found first, it shouldn’t crash the program. If there isn’t enough ANT+ channels for all the devices at the trainer ride, only connect to primary sensors. If a sensor intermittently loses signal, show it on the display that the connection was lost.

2 Actor Brief Descriptions

2.1 ANT+ USB Stick

2.2 ANT+ enabled devices

2.3 User database

3 Preconditions

ANT+ USB sticks are plugged into the computer and readable by the program.

4 Basic Flow of Events

1. The use case begins when the ANT+ Channel controller class object is created
2. Connect to all of the ANT+ USB sticks.
3. Connect to all of the channels on each ANT+ USB stick and save the channels as nodes in a 2d array.
4. Clear all of the channels on all the ANT+ USB sticks.

5. Open the next empty channel to search for power meters.
6. Open the next empty channel to search for heart rate devices.
7. Open the next empty channel to search for cadence sensors.
8. The program tells the controller to quit searching for devices.
9. If there are available channels, fill them with any secondary sensors that were dropped.
10. Send the sensor data to the display (continuously) during the workout.
11. Reset the ANT+ USB sticks upon exiting the program
12. The use case ends.

5 Alternative Flows

5.1 Power meter found

If in step 5 of the basic flow the channel finds a power meter, then

1. Search the device serial number in the database.
2. Check if the user profile for this device is already loaded
3. If the device is primary, open a new user object and add their profile from the database.
4. If the device is secondary, see if the primary device has already been found and add it to the user profile
5. Start showing instantaneous power as received from the sensor.
6. The use case resumes at step 5

5.2 Heart rate sensor found

If in step 6 of the basic flow the channel finds a heart rate sensor, then

1. Search the device serial number in the database.
2. Check if the user profile for this device is already loaded
3. If the device is primary, open a new user object and add their profile from the database.
4. If the device is secondary, see if the primary device has already been found and add it to the user profile, else make a new user object and add their profile
5. Start showing instantaneous power as received from the sensor.
6. The use case resumes at step 6

5.3 cadence sensor found

If in step 7 of the basic flow the channel finds a heart rate sensor, then

1. Search the device serial number in the database.
2. Check if the user profile for this device is already loaded
3. If the device is primary, open a new user object and add their profile from the database.

4. If the device is secondary, see if the primary device has already been found and add it to the user profile, else make a new user object and add their profile
5. Start showing instantaneous power as received from the sensor.
6. The use case resumes at step 7

5.4 No more empty channels

If in step 5,6, or 7 of the basic flow there are no more empty channels, then

1. If there is no channel looking for power sensor, and channel searching for heart rate sensor OR cadence sensor, reset it and look for power meters.
2. Find a user profile with a primary *and* secondary sensor and reset the channel that was being used for the secondary channel.
3. The use case resumes at step 5

6 Subflows

7 Key Scenarios

7.1 ANT+ has enough channels for all devices

7.2 ANT+ does not have enough channels for all devices

8 Post-conditions

8.1 Workout over

reset the ANT+ USB sticks, or they'll stay connected to their frequencies when the program quits.

9 Special Requirements

The trainer ride host should have enough ANT+ USB sticks for the amount of ANT+ sensors at the events.

Group 12

Use-Case: Load Workout

1 Brief Description

Host selects a pre constructed workout for the group, loads it, and starts the workout when participants are ready.

2 Actor Brief Descriptions

2.1 Host

3 Preconditions

1. System must be started
2. Workouts have been created
3. Users devices linked via ANT+

4 Basic Flow of Events

1. The use case begins when the host starts the program.
2. Host selects a workout that has been created.
3. Host sets delay
4. Host sets end time
5. Workout begins
6. The use case ends.

5 Alternative Flows

5.1 No Workouts Created

If in step 2 of the basic flow there are no workouts to be selected by the host, then

1. The host will have the option to create a workout.
2. The host will have the option to select a basic default workout.
3. The use case resumes at step 3

5.2 Invalid Workout File

If in step 2 of the basic flow the host selects an invalid workout file then,

1. Error message displayed to host.
2. Prompt host to select alternative workout file.
3. Host selects a different workout file.
4. The use case resumes at step 3

5.3 Invalid Delay

If in step 3 of the basic flow the host enters a time that has passed then,

1. Error message displayed to host.
2. Prompt host to enter a new start time (delay).
3. Host selects “Ok”.
4. The use case resumes at step 4

5.4 Invalid End Time

If in step 4 of the basic flow the host enters a time that has already passed then,

1. Error message displayed to host
2. Prompt host to enter a new valid end time.
3. Host selects “Ok”
4. The use case resumes at step 5

6 Subflows

6.1 user is prompted to create workout

1. user is sent to workout creator to create a workout.
2. once workout is saved user returns to load workout menu.

7 Key Scenarios

7.1 User loads a workout and the workout begins

7.2 No workout has been created and the user is prompted to create one.

8 Post-conditions

8.1 Workout is loaded and started

9 Special Requirements

9.1 A workout has been created before loading, ANT+ devices detected and connected

Group Participation

Nathan Franklin created the use cases for Build Workout, and ANT+ channel controller.

Josh Baker created the use case diagram, use case for Load Workout, and uploaded to online locations.