**Exposition of user stories and admission of tasks**

**Janou B.M.E Christophe**

Graduation internship   
of ICT & Technology at

Fontys University of Applied Sciences in Eindhoven

7 september 2016

Table of Contents

[1. User story 1: system testability 3](#_Toc461008190)

[2. User story 2: system expandability 3](#_Toc461008191)

[3. User story 3: battery duration 3](#_Toc461008192)

[4. User story 4: easy connection 3](#_Toc461008193)

[5. User story 5: gesture abstraction 3](#_Toc461008194)

[6. User story 6: (optional) appliance control 3](#_Toc461008195)

[7. User story 7: music control 3](#_Toc461008196)

[8. User story 8: switch connection 4](#_Toc461008197)

[9. User story 9: multiple connections 4](#_Toc461008198)

[10. User story 10: click registration 4](#_Toc461008199)

# User story 1: system testability

As a developer I want a testable system so that I can check the system and improve its performance.

* 1. UML design which incorporates design patterns.
  2. the UML design uses the S.O.L.I.D. principles
  3. convert the UML design to code.
  4. fill in the code to add the functionality

# User story 2: system expandability

As a developer I want an expandable system so that I can easily add new features and modules.

* 1. The UML design uses the S.O.L.I.D. principles.
  2. Good use of the solid principles allows for a proper expandability.

# User story 3: battery duration

As an end user I want a device which’s battery lasts longer than 24 hours on a single charge as that I can use my device at least an entire day.

* 1. Put the system when not in use in low power mode.
  2. Measure the current draw and not the usage over time
  3. Optimize code for short awake time
     1. This depend on the connection time
  4. Research batteries and charging techniques.

# User story 4: easy connection

As an end user I want a device to which I can easily wirelessly connect to so that I do not have to hassle with strange configurations.

* 1. Research wireless connection techniques
  2. The device to the connection is made

# User story 5: gesture abstraction

As a developer I want an easy to use API so that I do not have interpret the gestures.

* 1. Interpret and filter (denoise) the raw data.
  2. Extract gestures from the raw data
  3. Make retrieving the gestures easy
  4. Make it reliable

# User story 6: (optional) appliance control

As an end user I want to control my appliances so that I can change the state of my lamp.

* 1. 2 options
     1. Connect the touchpad to an appliance, i.e. switch its network
     2. Use the host to set the control for the lamp i.e. using an app to use the touchpad as a controller

# User story 7: music control

As an end user I want something from which I can control my music so that I can do that without needing my phone.

* 1. The driver needs to switch the basic interpretation to a more specific interpretation.
     1. In Android the default gesture control can be set for volume up/down and next/previous song.

# User story 8: switch connection

As an end user I want to switch the connection of the trackpad from my phone to my computer so that I use the same device on multiple machines.

* 1. All devices require a driver to interpret the protocol.
  2. The protocol can be used on multiple devices.

# User story 9: multiple connections

As an end user I want to connect my wireless headphones and touchpad to my phone at the same time.

* 1. Research on multiple synchronous wireless connections.
  2. If possible with current technologies implement the multiple connections.

# User story 10: click registration

As an end user I want that the trackpad detects the location as well as “clicks” and other gestures so that I can comfortably control my device

* 1. Export the gestures to the host device using a protocol which allows clicks to be sent\
  2. Use the device driver on the systems to interpret the wirelessly received dataset as certain gestures.
  3. Filter false registrations.
  4. 2 contact point registration.
  5. Build a system to easily detect finger up/down.
  6. Struct which contains all the interaction data of 1 or more fingers.

# User story 11:

As a developer I want hardware so that I can use the system

* 1. Pick hardware
  2. Order hardware
  3. Assembly hardware