Pflichtenheft

Virtual Reality for Sensor Data Analysis

Projekt:	Virtual Reality for Sensor Data Analysis 0.1
Autor:	Gero Birkhölzer, Johannes Blank, Alexej Gluschkow, Fabian Klopfer, Lisa-Maria Mayer
letzte Änderung:	May 5, 2017

Contents

1 Purpose — **3**
- 1.1 Mandatory Criteria . . . 3
- 1.2 Desired Criteria . . . 3

2 Product Environment — **4**
- 2.1 Software . . . 4
- 2.2 Hardware . . . 4

3 Product Functions — **5**
- 3.1 General Features . . . 5
- 3.2 VR-World . . . 5
- 3.3 Settings . . . 6

4 Proposed Architecture — **7**
- 4.1 Overview . . . 7
- 4.2 Component Decomposition . . . 7
  - 4.2.1 Services . . . 7
  - 4.2.2 GUI . . . 8
  - 4.2.3 Additional Classes . . . 8

5 Product Data — **9**
- 5.1 VR-World . . . 9

6 User interface — **11**
- 6.1 Structure . . . 11
  - 6.1.1 Start screen . . . 11
  - 6.1.2 VR-Mode . . . 11
  - 6.1.3 Live Data . . . 11
  - 6.1.4 Settings . . . 12
- 6.2 Layout . . . 12

7 Quality Requirements — **13**

8 Test Cases — **14**

9 Development Environment — **15**
- 9.1 Software . . . 15
- 9.2 Hardware . . . 15

10 Project Time Line — **16**

1 Purpose

The software project module in 2017 at the University of Constance focuses on the development of an app for mobile devices.
Especially, this Pflichtenheft intends to describe the structure of an implementation of a virtual reality representation of BLE sensor feedback.

1.1 Mandatory Criteria

M1 The app shall use the Bluetooth adapter of the smartphone to connect to a TI SimpleLink SensorTag device.

M2 The app shall track the position of a TI SimpleLink SensorTag device with up to 30m tolerance.

M3 The app shall visualize the sensors' data and its position using 3D/stereoscopy, more concrete the WebVR framework.

M4 The visualization mentioned in M3 shall be explorable by tilting the joystick of a bluetooth controller.

M5 The VR-World shall consists of at least two different rooms.

M7 The app shall display the stored data inside the VR-World.

M8 The data from the sensor shall have two different representations in the VR-World.

1.2 Desired Criteria

A1 The app could visualize the sensors' data and its position using augmented reality.

A2 The VR-World could represent a whole corridor with more than two rooms.

A3 The app could give a time lapse of the data inside the VR-World.

2 Product Environment

2.1 Software

- Android 5.0 Lollipop or higher

2.2 Hardware

- Bluetooth-enabled Smartphone
- TI SimpleLink SensorTag device
- Victorstar VRBox 2.0
- VR-Park Bluetooth Controller

3 Product Functions

3.1 General Features

F1.1 The app shall use the Bluetooth adapter of the smartphone and the Android.bluetooth library to connect to a TI SimpleLink SensorTag device.

F1.2 The app shall provide a live data view of the sensor feedback in human readable form.

F1.3 The app shall track connected TI SimpleLink SensorTag devices.

3.2 VR-World

The VR-Mode is a 3D view of the world. When entering VR-Mode the user will see a fullscreen 3D world and by pressing the button in the lower right corner he can enter the stereoscopic view of the World. The VR-World is a 3D representation of a real series of rooms.

F2.1 The VR-World shall be able to be viewed inside a web browser and from within the app.

F2.2 While viewing the VR-World the user shall be able to look around using the gyro sensor of his phone to pan the camera around.

F2.3 While the app is not in stereoscopic 3D mode the user shall be able to click and drag to pan the camera around.

F2.4 The app shall be able to move the camera inside the VR-World by using a bluetooth controller.

F2.5 The data fechted from the sensors shall be displayed inside the VR-World.

F2.6 When in VR-Mode, the app shall be in fullscreen mode.

F2.7 The app shall exit the VR-Mode if the user is pressing the "x" in the top right corner of the screen.

F2.8 The app shall be able to switch between stereoscopic 3D and normal 3D mode.

F2.9 The app shall be able to switch from fullscreen VR-Mode to stereoscopic by pressing the button in the lower right corner or by pressing the A-Button on his controller.

F2.10 The app shall be able to exit by pressing the back button on his device or by touching the back button in the top left corner.

F2.11 The app shall be able to switch to the settings screen, while it's in normal 3D mode.

F2.12 The app shall be able to switch rooms if the user pushes the B-Button on his controller.

F2.13 The app shall visualize the position of stored data from the TI SimpleLink SensorTag device.

F2.14 The app shall visualize the given data by the Ti SimpleLink SensorTag, by displaying a point approximately at the sensors stored location, with a number for the value of the the data.

F2.15 The app shall visualize the data by spanning a mesh over all recorded points from the sensor, while the height is the value of the given data.

F2.16 The User shall be able to switch between the two representation by pressing the X-Button on his controller.

3.3 Settings

The user can set the following options:

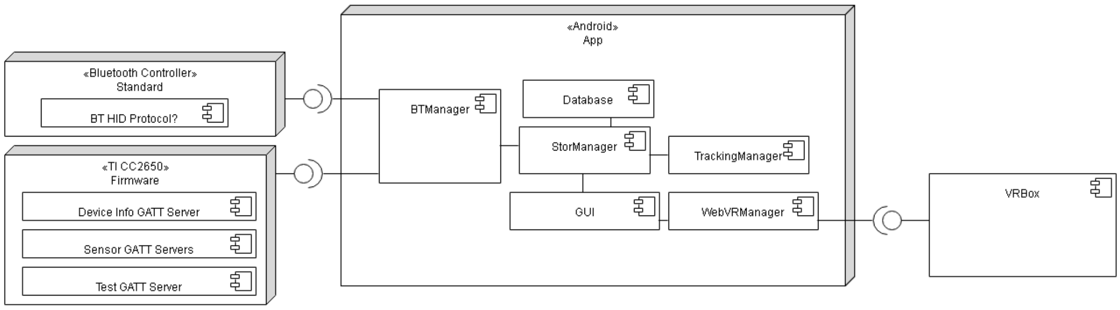
F3.1 The app shall be configuarable so that the user may choose wich data shall be displayed in the VR-World (temperature, etc.).

F3.2 The app shall list the connected devices and a short info about the current setting and state of the TI SimpleLink SensorTag device.

F3.4 The app shall list the results of a Bluetooth scan and present an user interface for controlling the connection of TI SimpleLink SensorTag devices.

4 Proposed Architecture

4.1 Overview

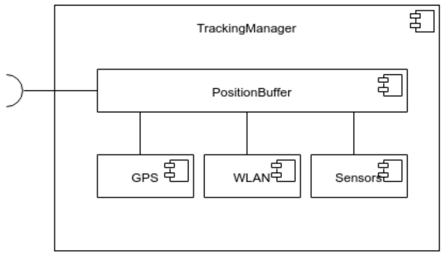


4.2 Component Decomposition

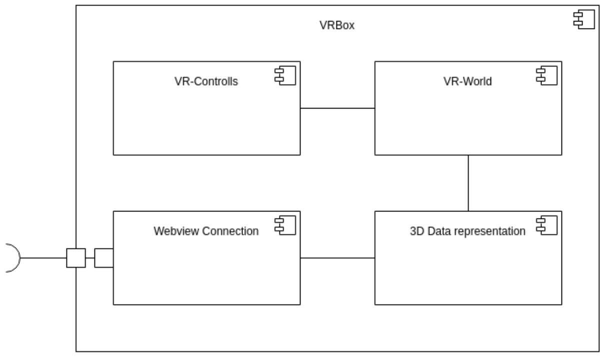
4.2.1 Services

From AndroidDoc:
"A Service is an application component that can perform long-running operations in the background, and it does not provide a user interface".

- **BluetoothManager:** Uses the android.bluetooth and especially the android.bluetooth.le libraries to fetch the sensor data from the TI CC2650.
- **TrackingManager:** Handles the tracking of the cellphone and therefore of the TI SensorTag devices. the current position gets determined by GPS and enhanced by the cellphone sensor and wifi data.



- **WebVRManager:** Uses webview to display the WebVr-World and communicates with the WebVr-Site.
- **StorageManager:** uses a DB (z.B. NeDB or internal stor or similar to safe parsed sensor input, location, settings,...)
- **VRBox:** Handles the display of the Vr-World and the given data from the sensor.



4.2.2 GUI

From AndroidDoc:
"They (Activities) serve as the entry point for a user's interaction with an app, and are also central to how a user navigates within an app (as with the Back button) or between apps (as with the Recents button)".

- **MainActivity** Provides the main startup screen as the main entry point.
- **VRViewActivity** shall provide the WebVR view using the android.webkit library (especially .webview).
- **LiveDataActivity** shall provide a view of the sensor data in human readable form.
- **TISettingsActivity:** Settings screen containing scanning and connecting, connected devices and device settings fragments.
  - **ScanningConnectingFragment** shall show the scanning results, delivered by the SensorTagBluetoothReceiverService and controll to which device to connect to or disconnect.
  - **ConnectedDevicesFragment** shall show a list of all connected devices and a short info about the current setting and state of the TI SimpleLink SensorTag device.
  - **ConnectedDevicesSettingsFragment** shall implement the configuration of the app features of the sensor.

4.2.3 Additional Classes

- **GATT Profiles** (for each sensor one)
- **GATT Sensor Service UUIDs**
- **Parser Functions** because the BLE protocol implemented in the TI CC2650 delivers raw sensor output

5 Product Data

5.1 VR-World

D1.1 Models: The models used to render the VR-World will be saved as .obj files using Blender in /webvr/models/.

D1.2 Textures: As .png files in /webvr/img/.

UUIDs Device Info Service 0000180a-0000-1000-8000-00805f9b34fb

Firmware Revision 00002A26-0000-1000-8000-00805f9b34fb

IR Temprature Service f000aa00-0451-4000-b000-000000000000

IR Temprature Data f000aa01-0451-4000-b000-000000000000

IR Temprature Configuration f000aa02-0451-4000-b000-000000000000

IR Temprature Time Period f000aa03-0451-4000-b000-000000000000

Accelerometer Service f000aa10-0451-4000-b000-000000000000

Accelerometer Data f000aa11-0451-4000-b000-000000000000

Accelerometer Configuration f000aa12-0451-4000-b000-000000000000

Accelerometer Time Period f000aa13-0451-4000-b000-000000000000

Humidity Service f000aa20-0451-4000-b000-000000000000

Humidity Data f000aa21-0451-4000-b000-000000000000

Humidity Configuration f000aa22-0451-4000-b000-000000000000

Humidity Time Period f000aa23-0451-4000-b000-000000000000

Magnetometer Service f000aa30-0451-4000-b000-000000000000

Magnetometer Data f000aa31-0451-4000-b000-000000000000

Magnetometer Configuration f000aa32-0451-4000-b000-000000000000

Magnetometer Time Period f000aa33-0451-4000-b000-000000000000

Optical Service f000aa70-0451-4000-b000-000000000000

Optical Data f000aa71-0451-4000-b000-000000000000

Optical Configuration f000aa72-0451-4000-b000-000000000000

Optical Time Period f000aa73-0451-4000-b000-000000000000

Barometer Service f000aa40-0451-4000-b000-000000000000

Barometer Data f000aa41-0451-4000-b000-000000000000

Barometer Configuration f000aa42-0451-4000-b000-000000000000

Barometer Calibraton f000aa43-0451-4000-b000-000000000000

Barometer Time Period f000aa44-0451-4000-b000-000000000000

Gyrometer Service f000aa50-0451-4000-b000-000000000000

Gyrometer Data f000aa51-0451-4000-b000-000000000000

Gyrometer Configuration f000aa52-0451-4000-b000-000000000000

Gyrometer Time Period f000aa53-0451-4000-b000-000000000000

Movement Service f000aa80-0451-4000-b000-000000000000

Movement Data f000aa81-0451-4000-b000-000000000000

Movement Configuration f000aa82-0451-4000-b000-000000000000

Movement Time Period f000aa83-0451-4000-b000-000000000000

Test Service f000aa64-0451-4000-b000-000000000000

Test Data f000aa65-0451-4000-b000-000000000000 shall equal the test result

Period in tens of milliseconds Configuration: 0: disable, 1: enable; in case of gyrometer or movement sensor: 0: disable, bit 0: enable x, bit 1: enable y, bit 2: enable z

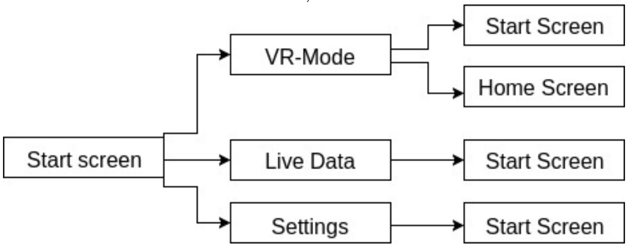
6 User interface

6.1 Structure

A small overview of the menu Structure.

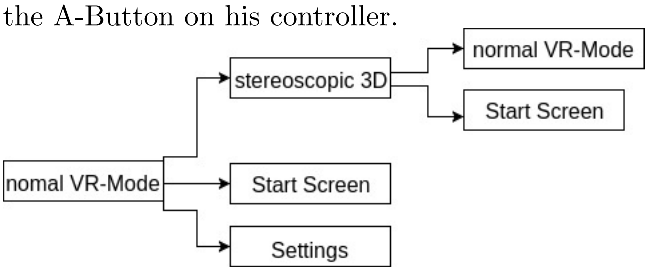
6.1.1 Start screen

The Start screen will be shown when the app is launched, can switch to everything. He can enter the VR-Mode, Live-Data from the sensor or change the settings.



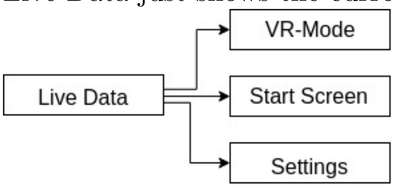
6.1.2 VR-Mode

The VR-Mode launches normally in normal 3D mode from where the user can switch to stereoscopic 3D view by touching the button in the lower left corner or by pressing the A-Button on his controller.



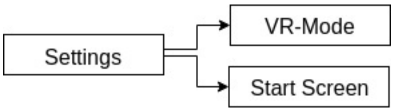
6.1.3 Live Data

Live Data just shows the current live data from the connected sensor.



6.1.4 Settings

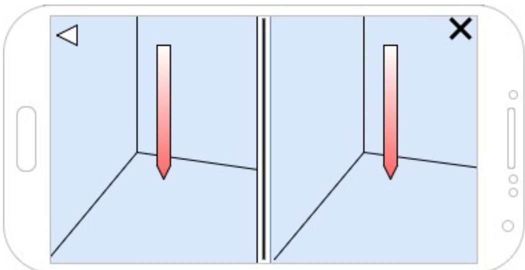
Here the user can select which sensor in range he wants to connect to and some basic settings like switching blue-tooth on and scan for more devices. From the Setting menu the user can switch to VR-Mode without going back to the start screen.



6.2 Layout

A mockup of the Start up screen.

And a mockup of the stereoscopic Vr-Mode.



7 Quality Requirements

	very important	important	less important	lesser important
Functionality				
Adequacy		**X**		
Correctness		**X**		
Interoperability				**X**
Security				**X**
Reliability		**X**		
Usability				
Comprehensibleness			**X**	
Usability			**X**	
Efficiency				
Time response			**X**	
Resource costs			**X**	
Portability				**X**

Functionality All functions should work as intended, but neither the interaction with other programms nor the security of the system is taken into account.

Reliability Errors should be reduced to a reasonable amount.

Usability The App should be usable, but user-friendliness is not stressed during the development.

Efficiency The App should respond in reasonable time to inputs. It also should use reasonable amounts of processor time and storage.

Portability The App will developed for Android without consideration for other operating system.

8 Test Cases

/T0300/ *Look around:* While in normal 3D mode the tester shall click the screen and drag first up to move the camera up. Then move down to move the camera down, then at last left and then right, all the time the camera must follow the movement of the finger. After this the tester shall tilt the phone up to move the camera up, then tilt it down, left and right. The camera shall follow the tilt direction of the phone all the time with no delay.

This test shall be repeated in stereoscopic 3D view. While the clicking and dragging shall not work, the tilting of the phone shall be the only way to pan the camera.

/T0310/ *Move inside VR-World:* While in normal 3D mode the tester shall tilt the joystick on the controller forward and the camera shall move forward. By tilting the joystick backward the camera shall move back, by tilting left the camera shall move left and by tilting right it shall move right. The camera shall allways follow the view point, so forward is allways in the center of the camera.

This test shall be again repeated in stereoscopic 3D view and all functions shall work the same.

/T0320/ *Searching, connecting and disconnecting devices:* While on the TISettings-Activity the tester shall search a TI SimpleLink SensorTag device by pressing the "Scann" button. All devices nearby shall be shown in a list with distinguishable entries. By tapping on a list entry a connection to the device shall be established. By tapping again on the list entry the connection shall be terminated.

/T0330/ *Displaying temperature:* While in normal 3D Mode and a established connection to a TI SimpleLink SensorTag device the tester shall look around. At the position of the device a glowing shere shall be displayed.

This test shall be again repeated in stereoscopic 3D view and shall work the same.

9 Development Environment

9.1 Software

OS	Windows 10, macOS Sierra, Linux Mint 18.1
IDEs	Android Studio, Sensor Controller Studio 1.4.1, Atom, Chrome DevTools
VCS	Git,GitHub
UML-Editor	Enterprise Architekt, MS Visio, draw.io
Zeichensatz	LaTeX

9.2 Hardware

Smartphone	Motorola XT1572
Sensor	TI CC2650STK
VR-Headset	Victorstar VRBox 2.0
Bluetooth-Controller	VR-Park (?)

10 Project Time Line

Week / Final Date	Event / Tasks
25.5.- 1.5.	first research, write Pflichtenheft
2.5.	release Pflichtenheft, project plan, subjects of milestones
2.5.- 8.5.	distribute tasks, decide on design
9.5.- 15.5.	start building
16.5.- 22.5.	
22.5.	*Milestone 1:* Basic functions are implemented (gathering data, display data in app, build a basic 3D world)
23.5.- 29.5.	
30.5.- 5.6.	
6.6.- 12.6.	
12.6.	*Milestone 2:* Gathered data can be displayed in 3D, *intermediate assessment*
13.6.- 19.6.	
20.6.- 26.6.	
27.6.- 3.7.	
4.7.- 10.7.	
11.7.- 17.7.	
17.7.	*Milestone 3:* The app works as wanted :D
18.7.- 24.7.	prepare presentation and usage examples
25.7.	*final presentation*

Possible starting points:
Simple, bad layout
TI official, complex