MASTER'S THESIS

CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF ELECTRICAL

ENGINEERING

DEPARTMENT OF CYBERNETICS

PROTOTYPE OF NOTIFICATION - MONITORING WEARABLE

JANUARY 2018

SUPERVISOR: MGA. ANNA KUTÍKOVÁ

EDUARD FŰZESSÉRY

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Eduard F ü z e s s é r y

Study programme: Biomedical Engineering and Informatics

Specialisation: Biomedical Informatics

Title of Diploma Thesis: Prototype of Notification-Monitoring Wearable

Guidelines:

Main task of this work is to create a prototype, which notifies user about events from smarphone in diferent environments of his work and personal life. Perform user research for defining target group of this prototype, including users needs and problems.
The research should find answers on important question with relation to (smart) wathces/bands, motivation of people to buying these devices and relationship of user to these kind of devices. Based on results of this research identify main issues and their possible solutions. Implement one solution and test it with users.

Bibliography/Sources:

[1] Kuniavsky Mike - Observing the user experience - USA 2003
[2] Disman Miroslav - Jak se vyrábí sociologické znalost - ČR 2002
[3] Contents Convergence Research Center (CCRC) - Lightful user interaction on smart wearables - London 2016

Diploma Thesis Supervisor: MgA. Anna Kutíková

Valid until: the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 13, 2017

Thanks

I would like to thank all the people who helped me with this thesis, from my supervisor, to my family and a lot of hours of consultations with friends and colleagues. Thank you.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date 8.1.2018 ....

Eduard Fűzesséry

Abstract

This work deals with the process of creating a working prototype of a wearable, which is able to measure walked distance and notifies the user on his wrist. Notifications from all users' applications are coming from users' smartphone via Bluetooth Low Energy. This work focuses on finding out users' preferences about their current watches and other smart wearables, their needs and wants on this topic. I have summarized theses preferences and based on this summarization I have created prototype of an environment (wearable, smartphone application) for target group of users based on this research. I have created user requirements on system, low-fidelity prototypes of mobile application, as well as for wearable, for testing purposes and to assigning priorities on the system. The plan was to create MVP = minimal viable product. After prototyping the software, the hardware was chosen for the wearable, main part of the hardware is the SoC (System on a Chip) with CPU and Bluetooth chip supporting Bluetooth Low Energy. For the purposes of this work nRF51822 from Nordic Semiconductor was chosen. After creating the model of a logic board and assembly attempt, was created 3D model and then the model was 3D printed. The solution was then tested with 2 potentional users.

Keywords:
wearable; watch; prototyping; user study; Android application

Abstrakt

Táto práca sa zaoberá procesom vytvorenia prototypu nositeľného zariadenia, ktoré je schopné merať prejdenú vzdialenosť a upozorňovat na notifikácie na jeho zápästí. Upozornenia od všetkých užívateľových aplikácií prichádzajú z užívateľovho smartphonu pomocou Bluetooth Low Energy technológie. Ďalej sa zaoberá zistením užívateľských preferencií o ich aktuálnych hodinkách a iných smart zariadení. Zaujímajú nás aj ich potreby a požiadavky na túto tému. Tieto požiadavky som zosumarizoval a z nich bol vytvorený prototyp pre cieľovú skupinu tohto zariadenia. Boli vytvorené užívateľské požiadavku, low-fidelity prototypy ako nositeľného zariadenia, tak aj aplikácií pre otestovanie a určenie užívateľských priorít na systém. Po naprototypování softwaru bol zvolený hardware na ktorom tento systém má bežať. Bol zvolený SoC (systém na čipe) s CPU a Bluetooth čipom, ktorý podporuje BLE. Pre účely tejto práce bol zvolený čip nRF51822 od Nordic Semiconductor. Po vytvorení modelu základnej dosky a pokuse o osadenie všetkých potrebných komponent bol vytvorený 3D model obalu, ktorý bol vytlačený na 3D tlačiarni. Následne bolo toto riešenie otestované na 2 užívateľoch.

Kľúčové slová:
nositeľné zariadenie; hodinky; prototypovanie; užívateľská štúdia; Android aplikácia

Contents

List of Figures

List of Tables

Chapter 1

Preface

In this work you can find the basic creation of a new wearable product through user research, defining, development and testing.

In first chapter we are trying to define what problems we can see in current market with wearables as well as we are trying to find out what current market offers in wearables. We will take a look at all types of wearables currently available as well as which will be available in near future.

In chapter Analysis and solution proposal we are focusing on defining what users want from their current, or rather wished devices and what features and properties these devices should have. With this we wil find out what problems are these uesrs seeing with their current wearables. We will take a closer look as how this user research was performed. After that we wil take conclusions from this user research and with results we will define most important features their desire, as well as we will define target group for this device. Furthermore we are describing challenges with Bluetooth Low Energy and how it should be implemented.

Next chapter is dealing with implementation challenges for proposed solution, step detection algorithm as well as architecture desicions for the applications. We are looking at the principles of MVVM on Android and B-VIPER on iOS and what particular letters mean.

Last but not least we will be looking at test phase of this diploma thesis and therefore we will take a look at the precision of step detection algorithm as well as user experience from users on wearing the prototype and using the applications.

Chapter 2

Problem and goal specification

People like to know what is happening with their bodies and social life. That is why there is such a big rise in wearable technologies. They provide a lot of information for the user from the perspective of their health thanks to sensor such as pedometer, heart rate monitor, thermometer and so on. They even provide information about their social life, thanks to connection with users smartphones and resending notifications from smartphones to their wearables.

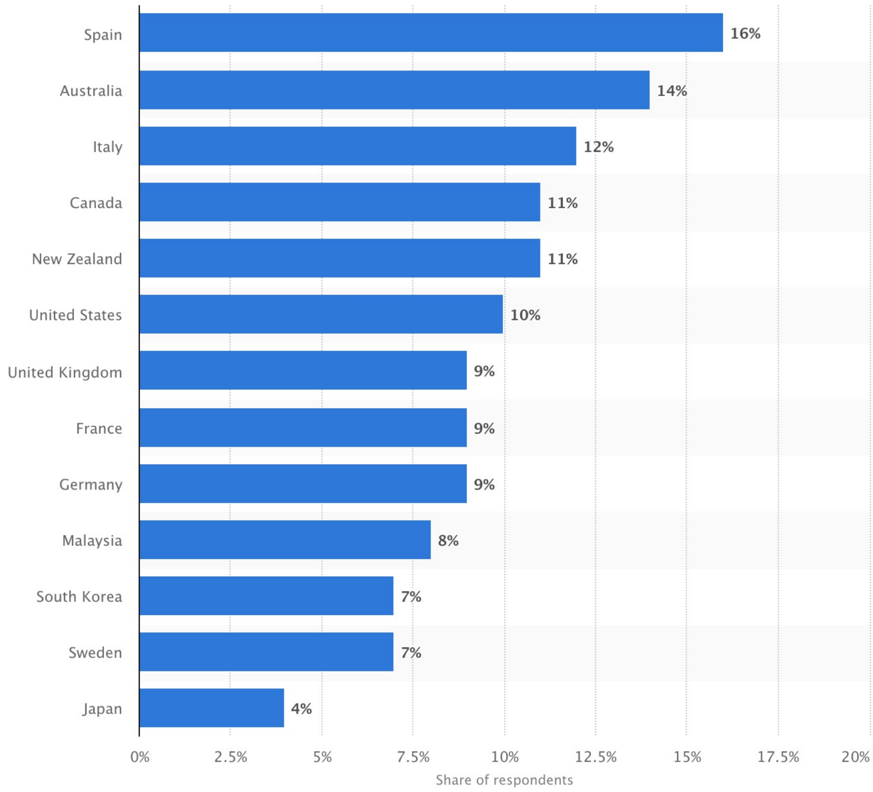

Figure 2.1: Wearable fitness device adoption rate worldwide in 2016, by country [?]

Wearable devices have great adoption rate among people, thanks to the ease of use and additional informations they provide, as seen above.

2.1 Problems with wearables

These devices have many advantages, but they are not flawless. For analysing their flaws we have to take a look at different categories of wearables.

2.1.1 Smart-watch

Smart-watch is the main category of smart wearables when one talks about smart wearables. Smartwatches are on the market from around 2013[3] so they already have few versions behind them and most of the, not just usability, problems are solved. The only big problem they still share between all the manufacturers is battery life. For some people its benefits outweight the need to charge them, almost, every single day and people do not have the problem with charging, because they need to charge their phones, almost every day, too. Another issue with purchasing smart-watch could be its price, which could render these devices as useless for people.

As for features they bring for the end user:

- notifications
- operating system
- touch screen
- thermometer
- pedometer
- heart-rate monitor
- water resistance
- many others

At figure 2.2 we can see one of the many examples of smart-watches currently available at the market. They are made by Apple Inc.

Figure 2.2: Apple Watch

2.1.2 Smart-ring

Smart-ring is the new trend in category of smart wearables. They promise many features in really small package. Among key features of these smart-rings we can place:

- notifications
- pedometer
- heart-rate monitor

- water resistance

- around a week of battery life

Figure 2.3: Motiv ring

Everything in above list looks great for its target group of users, but there are two key “features” not mentioned. These smart rings cost from 150 USD above and most of them are still in developement.

2.1.3 Smart-band

Smart-bands are widely known as fitness trackers. They are relatively small devices with great battery life, but not as many features as smart-watches. Sometimes they contain a touch screen with limited funcionality and as of Q4 2017 they do not have other than proprietary operating system. They offer great compromis between all of the features of smart-watches and smart-rings. Except they are cheaper mainly because there is no issue with space as in smart-rings and at the same time there is no need for processing power as big as with smart-watches. As smart-watches they already iterated over few generations and have the most issues tuned out, except one. They absolutely do not look great on your wrist. God forbid you would like to use your “dumb” watch. You would need to carry two devices on your wrist what is impractical, not good looking and in general annoying.

Figure 2.4: Xiaomi Mi Band 2

2.2 Goal specification

For reasons stated above we have decided to create similar device as smart-band, but with the key exception that the device is attached to your regular watch band. For specification of concrete features needed by the users we will perform user reasearch. This research will help us to check the truthfulness of above stated problems and overall significance of this solution. We will define our features based on this research as well as we will define target group for this product, because we have two main ways of attacking this problem. We can focus on the same market as smart-bands (inexpensive fitness-trackers), or focus on users who really care about their watches.

By the nature of this device we need to make sure it works with smartphones powered by iOS as well as Android, for this reason we will use Bluetooth low energy (BLE) as connection between device and smartphone.

There are few things we have to assume before we will go deeper into this topic:

- this device has other usage as the smart-band, more like classical watch
- we should use materials from classic watches (leather, silicone, metal, ...)
- relatively new product
- there will be specific attachment to watch

Chapter 3

Analysis and solution proposal

For us to get the most relevant results we have to conclude user research on this topic, create a prototype and then test the prototype out with our target group. For creation of a working prototype we need to analyse competition on the market and we have to analyse the proper technologies.

3.1 User research

Has very important role in a development process of new features, or products. It can show stakeholders the current state of a product, or can help them to decide future development. User research is coined term in UX (User experience) field. User research is relatively simple tool for gathering different information about current state of market or product with:

- questionnaires
- interviews
- reviewing other scientific studies.

User research tries to answer questions like:

- Who are your users
- What your users want
- What they purchase
- Where they shop
- What they own
- What they think of your product or brand

3.1.1 Realisation

The research was done with 7 participants defined as young adults. Every interview is concluded below with basic information about a participant. Every interview was divided into few themes, how many watches does participant has, if and how many of smart wearables does he/she has, what are the materials of the devices, what they like or do not like on their smart devices, what are their habits with wearable devices and so on.

- Participiant 1
  - is a young male owning 2 analog watches with stainless steel dial and one gum and one leather bands
  - he got both watches as a gitfs. At the same time he got for his sister smart band for pedometer and fitness tracker funcionality
  - he was thinking about buying a smartwatch for himself, but at that time he did not have nor smartphone supporting BLE nor the finances for these devices. In the meantime he got the 2 watches mentoined earlier and that is why he is no longer seeking this smartwatch
  - he is interested information about his step count and heart rate information during the day, but he did not find suitable solution for him
- Participiant 2
  - is a male, who does not wear any watches already for around 10 years, he gave up on wearing mainly because of the size they add to the wrist
  - currently this participiant has four regular watches with steel dials and leather, silicon and textile bands, he got these watches as gifts
  - he is in the market for smart ring mainly because of discrete health information gathering
  - for this participiant is important to be productive during the day, meaning he not would like to be disturbed, that is why he would appreciate to filter out notifications based on his context/application
- Participiant 3
  - is a young female, which has two different regular watches, currently wearing just one of them, if the outfit is matching to them, because the second one have dead battery and are more sporty
  - currently she is not thinking about buying a smart device, because she thinks she will be influenced by (health) results from this kind of a device (for example, if her step count is lower, then she have to make more steps, or otherwise, if her step count is above average, she can eat more unhealthy), but in overall she would like to have the information about her daily step count

- she thinks that the notification feature would distract her from focusing and she would appreciate to turn off all notifications from this device

- Participiant 4
  - is a young female, she currently owns one regular watch, which she wears all the time (regardless of outfit), she plans to buy second watch in near future, the future watch should have metal narrow band
  - she thought about buying smart-wearable, but she did not like the idea of having two devices on a wrist at the same time
  - the idea of having smartwatch makes no sense for her, because she thinks its feature is richer than she needs and therefore more expensive then they need to be
  - number one feature for her is definitelly the pedometer and calorie counter
  - she does not like the idea of notifications, except running distance

- Participiant 5
  - is a young male, with only two smart-watches and one smart-band (no regular watch), currently using Motorola 360 2nd generation, the main reason why he has so much smart stuff is because of the fact that it just worked for him, but he wanted to try better/more expensive deivces
  - first impulse to buy smart-wearable was the unlock feature with an Android phone and some statistics about health data (pedometer, sleep tracking), his first device was smart-band and on this smart-band was for him one problematic feature, notifications, he had the feeling he needed to pull out his phone more often then before, the band contained color distinguished notifications, he used only this device for sport activities
  - he does not care about time, he just cares about his next appoitnment
  - fun fact: often is happening that he looks at the watch during the conversation people assume that he is in a time press, or bored or something and in fact just someone might be freaking out at Slack

- Participiant 6
  - is a young female currently owning one smart-band with display, before she used normal watch, this smart band was a gift from her boyfriend because she did not pick up her phone (notifications)
  - the main advantage of owning this smart-band is to filter notifications comming from phone
  - the health information provided by smart-band is for her just nice-to-have information, the big advantage of this feature for her is that the smart-band notifies her about making some steps, which she misses from her hourly goal

- Participiant 7
  - is a young female, she has 12 watches from which she almost every watch got as a gift; most of the watches have leather band and metal dial
  - she used Apple Watch for about a year, she stopped using it because of two main issues with this watch:
    * design perspective (impossible to combine with clothes)
    * annoyance of notifications (she turned them off, but display was still lighting up, catching her eye)
  - reasons she used Apple Watch:
    * pedometer, notification about standing up, making steps and so on
    * one can make a call with them
    * she found them cool

3.1.2 Conclusion

The study was conducted on people between 21 - 30 years old with 4 females and 3 male participants. Target group was selected between young people because of suggested target group (active, outfit aware, technically saavy). Based on this research I would divide (future) users of smart-wearables into two main catregories:

1. users which use these devices to filter out notifications and health data is nice-to-have feature
2. users which do not want to be disturbed by these notifications, but they want to have health data available

Interviewed users have many things in common, about what they need/want or use on their smart wearables.

- They want from them good battery life (definitelly more than one day)
- Most of the participants wants them to look cool/stylish, or to easily combine with different outfits.
- Participants would definitelly enjoy enabling/disabling notifications based on their context, (context examples: work/school, free time, working out, running, ...)
- Participants want to see their current step count visible, as well as calories burnt and distance walked

We will focus on the second group with our device, because participants showed bigger interrest in the health data. This decission was important for us to prioritise which features are important for the users. From this research we have concluded basic user needs, from which we will conclude user requierements.

3.2 User requirements

The solution should provide users with health data, based on research and the MVP (Minimal Viable Product) principle, we will implement only accelerometer, because it provides data for pedometer, which is the type of data source most asked for. The solution will provide history of the step data for the user in convinient and natural way. Another major issue with smart-wearable is price-design issue, which our proposed solution should solve by "hiding" the device under users current watch band, participiants were mostly enthusiastic about this solution, which points to real market value of this product.

The solution should provide users with notifications feature and most importantly the users will be able to turn off these notifications based on their preference.

Based on users smartphones choices the solution should support Android as well as iOS operating systems.

3.3 Design process

For this project was desing process executed for two main subprojects. First subproject was the hardware device and second one was smartphone applications (Android, iOS). That is why every following section is divided on these two subscetions.

This proces was heavily inspired by the book Design Thinking (Understand - Improve - Apply)[1]. There is described iterative proces of gathering information from and about users (user research), sketching, designing and testing. These subparts are repetitive through many cycles. These cycles are called desgin sprints, which have reason in bigger teams.

3.3.1 Sketching

Is a process of designing products, applications, UIs etc. It is mainly about comming up with different possible solutions to a problem and putting it on a paper. With applications we had to consider user requirements as well as industry (UI & UX) standards for these devices. Below you can see few of the sketches done with user requirements and standards in mind.

With wearable device the main challange of sketching was to find the optimal location for the device on a watch as well as how to securely attach this device with the possibility of the easiest removal by the user.

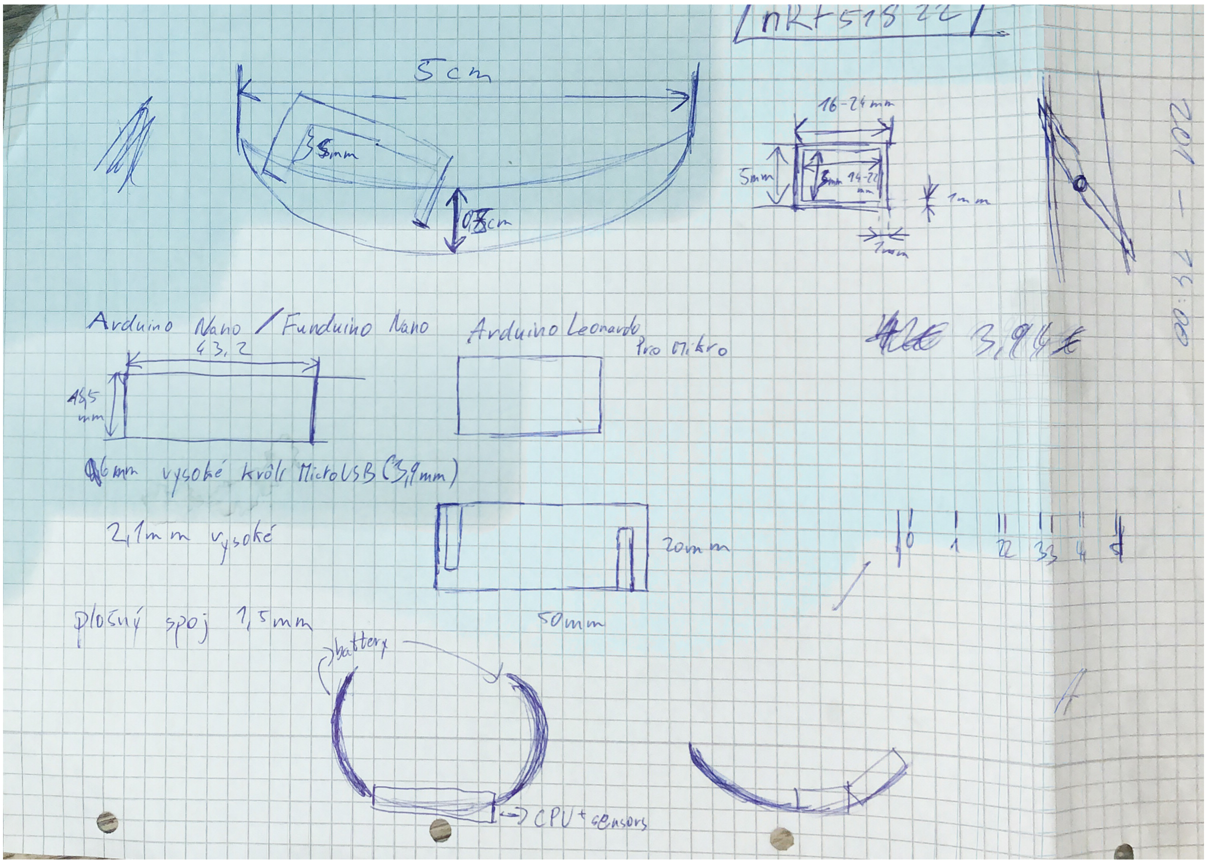

Figure 3.1: Sketching

From sketching we have got two possible solution of the wearable on a watch, first was on the bottom of a hand (where the clip is) and possible solutions of attaching are with adhesive material, magnet clip, system of hooks or with a velcro, last three possibilities going around the strap. The problems with this solutions are that there are many materials of straps as well as many types of clips holding the watch-strap together. Another solution is to place the device under the dial, in general there is more space but this space is not as standardized as on the other side of a hand, circle and square dials with different dimensions. The attachment possibilities are again with adhesive material, magnet (we need to find out if there would be some interference), and sophisticated hooks which would attach to straps attachements.

3.3.2 Low fidelity prototype

Lo-Fi prototype was created for mobile application UI as well as physical desing of the outer shell hardware.

Hardwares outer shell was developed out of a air drying modelling material for testing with users for the maximal possible dimesnions of the device and the correct placement on a wrist-watch. The testing was conducted on 2 participants, which showed interest in helping to bring the solution on the market, in the user study. Both of these participants have shown preference in wearing the device on the bottom of their watches as shown in the figure 3.2. Maximal possible dimensions for comfortable wearing were set to be at 4 cm long, 2 cm wide and 0.8 cm thick.

Figure 3.2: Low fidelity HW prototypes

Figure 3.3: Low fidelity HW prototypes

Applications were designed with program Balsamiq Mockups 3 and below you can see main parts of a low fidelity prototype.

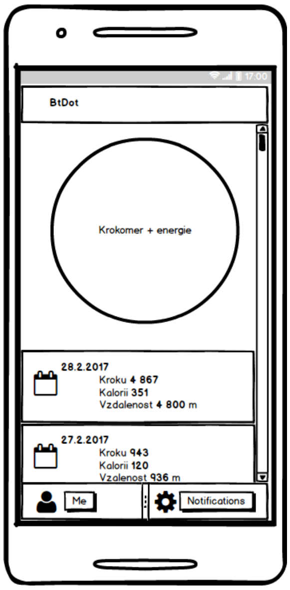

Figure 3.4: Main screen of a SW prototype

Main screens mockup shows big circular progress bar which shows current step count with distance below it and on tap the value is recalcutaled to current burnt calories estimation. This estimation is done based on users sex, height and mainly weight. This values will be recalculated on the fly from todays step count. Below the progress bar is history of your activities for the last 7 days. Everything older than 7 days is currently not available.

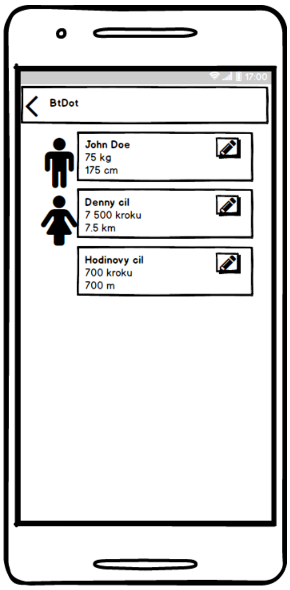

Figure 3.5: About me screen of a SW prototype

On About me screen users can view and adjust theirs goals and measurements.

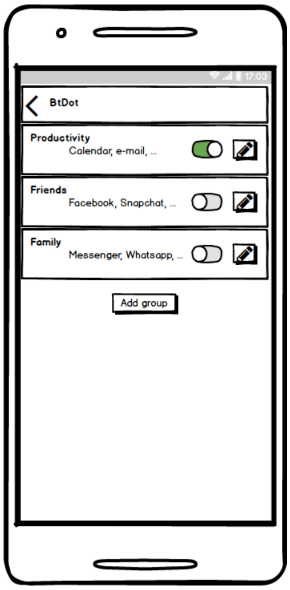

Figure 3.6: Notifications screen of a SW prototype

In user research we have found out that users filter out their notifications, that is why we have provided the low-fi prototype with notification grouping and possibility to quickly change the notifications settings, or rather which application can notify through the wearable.

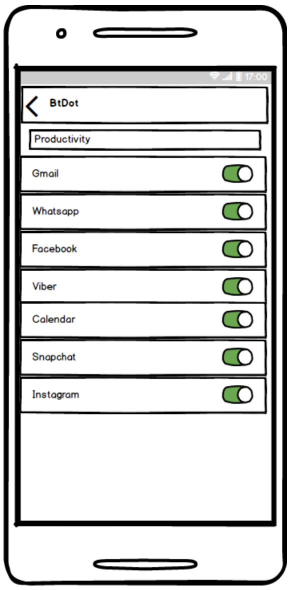

Figure 3.7: Main screen of a SW prototype

More screens are required for example connecting to a device, which will be in simple list form, or screen for setting up goals and parameters of a user.

3.4 Hardware

We can see from user requirements that we will need a few components for realisation such as CPU/MCU, bluetooth capable chip, accelerometer, some way of notificating users about new event (light, haptic) and a board to join all of the elements together. We intend to use components as small as possible for enabling users to wear the final product as secretive as possible. With this in mind we will use componets sized 0402 (0.4 mm × 0.2 mm), because there might be problems with implementation of 0201 (measuring 0.25mm × 0.125mm).

There are more available CPUs/MCUs on the market as SoC (System on a chip), for its widespread use in werable technology we have decided for nRF51822 from company Nordic Semiconductor. This SoC is suitable for more reasons:

- combines CPU with BLE support
- contains 32-bit ARM chip Cortex M0
- has embeded 2.4 GHz transceiver
- this chip is thread safe and run-time protected
- exists large community forum
- Nordic released relatively large amount of examples

Based on the decision using nRF51822 SoC we have to use power supply from 1.8 V to 3.6V, for the usage as long as possible, with as small form factor as possible we are looking at batteries around 200 mAh.

After market research on accelerometers available we have decided for LIS331DLH as it is ultra low-power high performance three axes linear accelerometer with variable output data rates from 0.5 Hz to 1 kHz. This accelerometer has wide supply voltage, similar to our selected SoC 2.16 V to 3.6 V and as mentioned ultra low-power mode consumption down to 10 µA.

For this prototype we have chosen to go with regular white LED light for user notifications.

For all of this possible we will have to design proprietary board and based on current market the board will be deployed by PragoBoard and then assembled by TTC, because working with 0402 components is complicated even under microscope.

3.4.1 Communication with smartphone

The device need to communicate with smartphone via few adresses and calls. In the table below you can see the proposed interface between device and smartphone application. With command request string, string of default response of device and phone and a brief description.

Command	Device/Phone response	Command description
whoami	Smart Watch/NULL	Gets device identification
get_steps	0000 0000/NULL	Receiving step count
set_steps	NULL/0000 0000	Setting step count
reset_steps	Step count reset/NULL	Reseting step count aka set_steps 00000000
get_raw_data_on	X: Y: Z:/NULL	Starts continuous raw data receiving
get_raw_data_off	X: Y: Z:/NULL	Stops continuous raw data receiving
get_time	0000 0000/NULL	Reading internal time 00-HH-MM-SS
set_time	NULL/0000 0000	Sets internal time sync 00-HH-MM-SS
notify_on	Notification ON/NULL	Turns on notification LED

Tabuľka 3.1: Communication table between device and mobile phone

The hardware device needs to be encapsulated in a plastic shell. The optimal solution for this shell is to 3D print around the board and battery. With 3D printing comes a lot of decissions about the material, method and 3D modeling of the shell itself.

3.4.2 Materials

There are more material types to choose from (plastic, metal, glass, ceramic) and for our prototype is the best solution to use plastic. There are 6 most used materials for 3D printing (Acrylonitrile Butadiene Styrene (ABS), Polylactic Acid (PLA), Polyvinyl Alcohol (PVA), Polycarbonate (PC), High-Density Polyethylene (HDPE) and nylon) and we will take a look at the 3 most common materials and their properties, because currently we are creating only a prototype.

3.4.2.1 Acrylonitrile Butadiene Styrene (ABS)

The most common material used for 3D printing with a melting point around 210-250°C[7]. It is the cheapest printing option currently available from the 6 plastic types, At the same time ABS has very good after printing attributes (can be easily sanded, glued and painted).

Of course there are some drawbacks in using this material. First of all this material is non-biodegradable plastic (it is recyclable), secondly it may irritate more sensitive persons during the printing process and the quality of ABS can deteriorate by prolonged exposure to sunlight.

3.4.2.2 Polylactic Acid (PLA)

Is a biodegradable thermoplastic which is derived from renewable resources, such as cornstarch, sugar cane, tapioca roots or even potato starch. This makes of PLA the most environmentally friendly solution in the domain of 3D printing, compared to all the other petrochemical-based plastics like ABS or PVA.

This material can be sanded down and may be painted over with acrylic paint, but some people recommend using a primer. Drawbacks are that it cannot stand too much heat - it becomes soft at around 50°C, other drawback is that PLA material cannot be glued together as easily as ABS, most people get pretty good results with cyanoacrylate (i.e. super glue), but one needs to be very caucios in using this cyanoacrylate.

3.4.2.3 Polyvinyl Alcohol (PVA)

The main feature of this material is that it is water-soluble. In 3D printing is PVA used as a way to provide a structure for an object with overhang issues (while the other material is hardening it can collapse, if there is an overhang), except that is PVA relatively expensive.

3.4.2.4 Conclusion

From available materials is ABS the most viable solution, because the drawbacks of the material (non-biodegrabale and sunlight exposure problem) are the least important in creation of a prototype in comparison with its advantages (painting, gluing and sanding options).

3.4.3 Methods

There are many method for 3D printing objects, in this section we will look at the 3 most common methods Fused deposition modeling (FDM), Stereolithography(SLA) and Selective Laser Sintering (SLS). While there are many more methods such as Digital Light Processing (DLP), Selective laser melting (SLM), Electronic Beam Melting (EBM) or Laminated object manufacturing (LOM) these methods are for specific purpouses or with really high input price or are just really hard to come by.

Most of the methods are using Standard Tessellation Language (STL) files as their datasource.

3.4.3.1 Fused deposition modeling (FDM)

FDM is the method that comes to mind when 3D printing is mentioned. This technology was developed and designed by Scott Crump in 1980s [7] it uses additive technique and it needs supporting elements for overhang parts of a model, created from another type of printing material, which can be dissolvable in for example soap water.

Principle is very easy to understand: The desired 3D object is sliced into layers and software calculates the moves required to build each layer, these layers are then printed from 2 material spools, these materials are heated to liquid state and propeled through extruder towards the surface where the printing material hardens. The extruder can move in X and Y direction and after a layer si fully printed then the slab/surface is lowered for next layer to be printed. To support planned layers the printer may place special supporting material from second material spool.

FDM is used for new product development, model concept and prototyping and even in manufacturing development. This technology is considered to be simple-to-use and environment-friendly. Therefore it is very likely candidate to be used with this project.

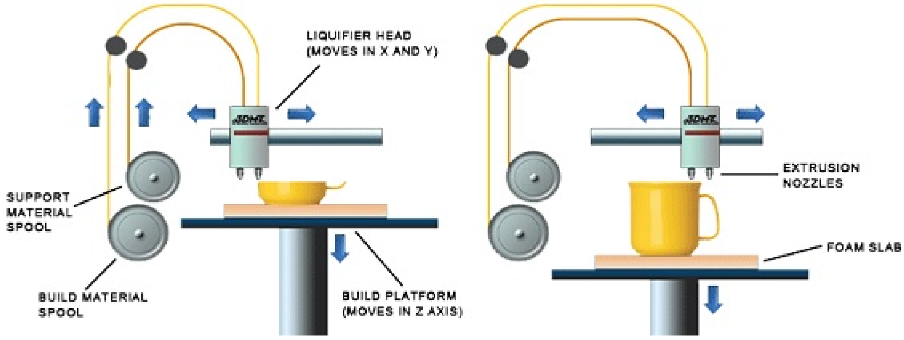

Figure 3.8: Fused deposition modeling principle [7]

Different kind of thermoplastic can be used to print parts. The most common of those are ABS (acrylonitrile butadiene styrene) and PC (polycarbonate) filaments. There are also several types of support materials including water-soluble wax or PPSF (polyphenylsulfone).

3.4.3.2 Stereolithography (SLA)

SLA and SLS can be viewed as very similar methods for 3D printing both are using laser to harden material in similar way, but there are differences in availability of the printing material as well as minimal wall thickness and used material.

SLA is additive technology with liquidified polymer in a basin, which is then hardend with the use of a laser, which can point in X, Y direction. After each layer is hardend the elevator moves the platform down of a layer-thickness difference. This method does need support structures for almost all the final prints to make sure the prints adhere to platform and not float around in liquid polymer. The support structures are computed automatically and they look like thin ribs, with only small tips actually touching the model to save material and printing time.

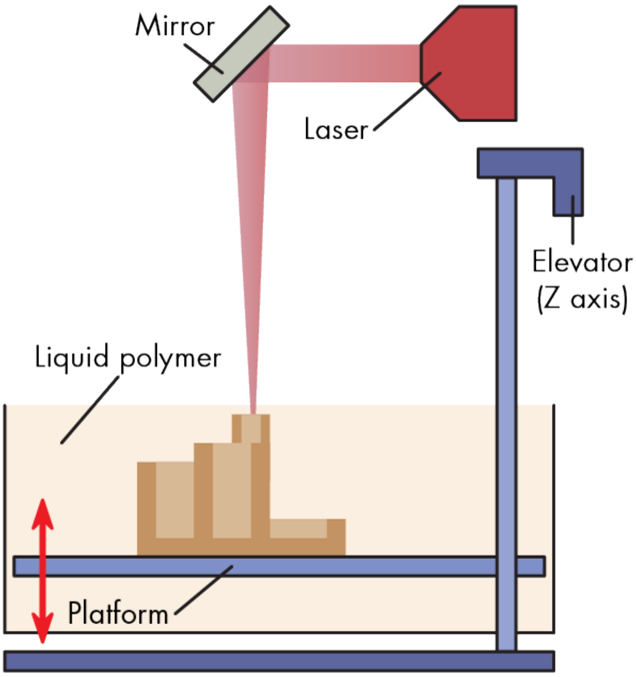

Figure 3.9: Stereolithographyprinciple [6]

3.4.3.3 Selective Laser Sintering (SLS)

As mentioned above this method is similar with SLA in a way that the printing solution is hardened by laser light, but with SLS method we do not need any support structures, because we are printing/hardening powdered material. As with SLA, SLS uses laser to harden in X and Y direction. The Z axis is created with covering the printing plate with new layer of powdered material, with the help of a leweling drum. These layers do not need support, because the powder is used for building as well as supporting of the final print. The printed object needs to be cleaned from the powder with an air gun.

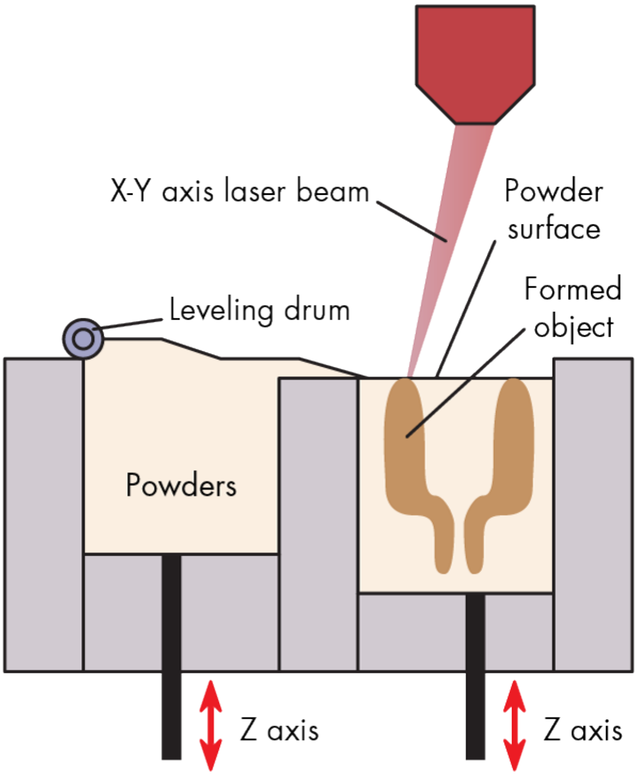

Figure 3.10: Selective Laser Sintering[6]

3.4.3.4 Conclusion

For this prototype with consideration of availability and money will be used the FDM method. As the model does not need any support structures and in general the method is used for prototyping.

3.5 Applications

As mentioned above we need mobile application for users to track their activity levels and to be notified on the device. These applications needs to have ability to persist data as in prototype stage of this project is no server planned. By default they need to have BLE enabled with notifications resending. They need to be UX and UI optimized for the product to be sucessfull. We need architecture for these applications and therefore we will use MVVM for Android and B-VIPER for iOS development, with Rx extensions for enabling easy multithreading capabilities.

3.6 Bluetooth Low Energy

Is a modern communication standard available from Bluetooth (further only BT) 4.0 specification. Currently (Q1 2017) there are 2 main forms of Bluetooth communication Bluetooth Basic

Rate/Enhanced Data Rate (BR/EDR) and Bluetooth with low energy[?]. Every bluetooth chip can communicate either on one of these technologies, or on both of them. Key differences between these standards in terms of use-cases:

- Bluetooth BR/EDR—establishes a relatively short-range, continuous wireless connection, which makes it ideal for use cases such as streaming audio

- Bluetooth with low energy functionality (LE)—allows for short bursts of long-range radio connection, making it ideal for Internet of Things (IoT) applications that do not require continuous connection but depend on long battery life

- Dual-Mode—dual-mode chipsets are available to support single devices such as smartphones or tablets that need to connect to both BR/EDR devices (such as audio headsets) and LE devices (such as wearables or retail beacons)

While each implementation has specific requirements that are detailed in the Bluetooth specification, the Bluetooth core system architecture has many consistent elements. The system includes an RF transceiver, baseband and protocol stacks that enable devices to connect and exchange a variety of classes of data.

Bluetooth devices exchange protocol signaling according to the Bluetooth specification. Core system protocols are the radio (RF) protocol, link control (LC) protocol, link manager (LM) protocol and logical link control and adaptation protocol (L2CAP), all of which are fully defined in the Bluetooth specification.

The lowest three system layers—the radio, link control and link manager protocols—are often grouped into a subsystem known as the Bluetooth controller. This is a common implementation that uses an optional standard interface—the Host to Controller Interface (HCI)—that enables two-way communication with the remainder of the Bluetooth system, called the Bluetooth host.

The primary controller may be one of the following configurations, depending on use case:

- BR/EDR controller including the radio, baseband, Link Manager and optionally HCI

- LE controller including the LE PHY, Link Layer and optionally HCI

- Combined BR/EDR controller and LE controller, with one Bluetooth device address shared by the combined controller

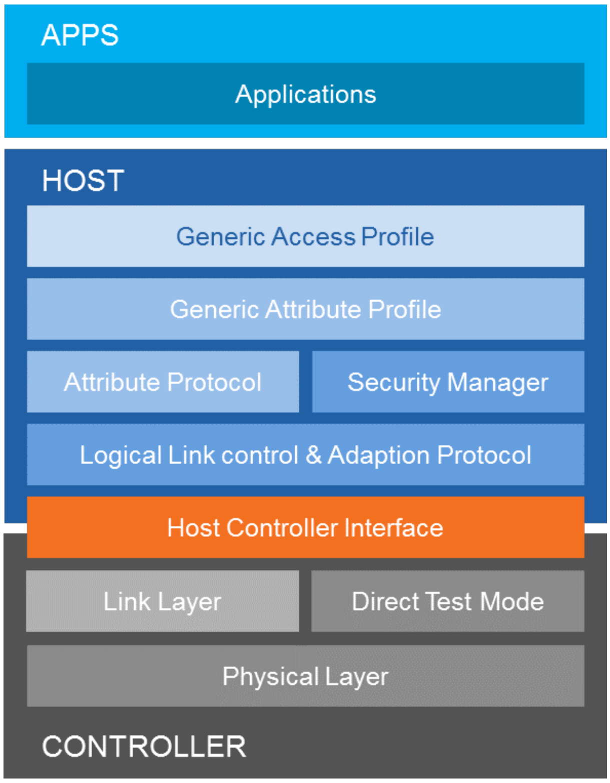

Figure 3.11: Bluetooth Core System Architecture

The Bluetooth specification enables interoperability between systems by defining the protocol messages that are exchanged between equivalent layers. It also enables interoperability between independent Bluetooth sub-systems by defining the common interface between Bluetooth controllers and Bluetooth hosts.

3.6.1 Bluetooth Low Energy (BLE)

Bluetooth channel is split into 40 channels of 2 types:

- Advertising Channels - used for discovering devices, initiating connection and broadcasting data. There are three fixed Advertisement Channels operating on frequencies 2402 MHz (channel 37), 2426 MHz (channel 38) and 2480 MHz (channel 39)
- Data channels - used for communicating data after the devices get connected. There are 37 data channels that use frequencies 2404-2424 MHz (Channels 0-10), 2428-2478 MHz (Channels 11-36)

3.6.1.1 State Machine

The operation of the Link Layer can be described in terms of a state machine with the five states as can be seen on figure 3.12. The Link Layer state machine allows only one state to be active at a time. The Link Layer shall have at least one Link Layer state machine that supports one of Advertising State or Scanning State. The Link Layer may have multiple instances of the Link Layer state machine. Certain combinations of states and roles within multiple state machines in the Link Layer are prohibited.

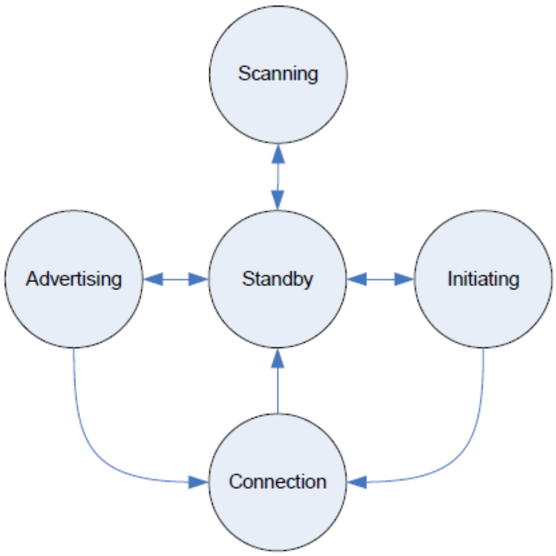

Figure 3.12: Bluetooth State Machine

3.6.1.2 Connection Event

The Link Layer in the Connection State shall only transmit Data Channel PDUs in connection events. The master and slave are supposed to determine the data channel index for each connection event. The same data channel index is going to be used for all packets in the connection event. Each connection event contains at least one packet sent by the master. During a connection event, the master and slave alternate sending and receiving packets. The connection event is considered open while both devices continue to send packets. The slave shall always send a packet if it receives a packet from the master.

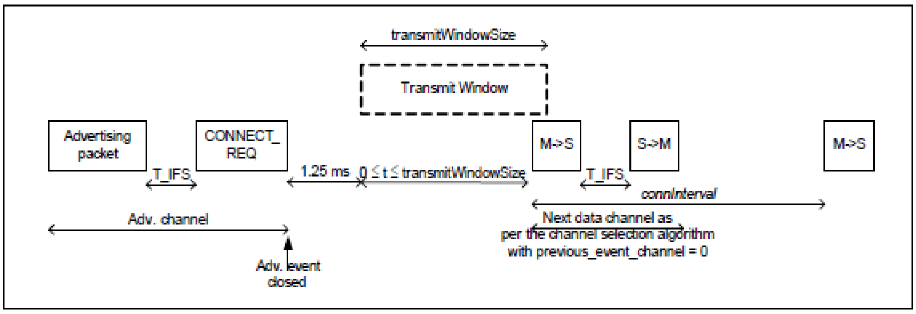

Figure 3.13: Bluetooth Connection Event

3.6.2 Wearable

Wearable is responsible for sending advertising packets to be discoverable by smartphone applications and then respond to application requests, based on proposed table Communication table between device and mobile phone.

3.6.3 Android

From Android 4.3 (API level 18) is BLE supported and these terms are important to know before starting implementing it in Android.

Generic Attribute Profile (GATT) — The GATT profile is a general specification for sending and receiving short pieces of data known as "attributes" over a BLE link. All current Low Energy application profiles are based on GATT. The Bluetooth SIG defines many profiles for Low Energy devices. A profile is a specification for how a device works in a particular application. Note that a device can implement more than one profile. For example, a device could contain a heart rate monitor and a battery level detector.

Attribute Protocol (ATT) — GATT is built on top of the Attribute Protocol (ATT). This is also referred to as GATT/ATT. ATT is optimized to run on BLE devices. To this end, it uses as few bytes as possible. Each attribute is uniquely identified by a Universally Unique Identifier (UUID), which is a standardized 128-bit format for a string ID used to uniquely identify information. The attributes transported by ATT are formatted as characteristics and services.

Characteristic — A characteristic contains a single value and 0-n descriptors that describe the characteristic's value. A characteristic can be thought of as a type, analogous to a class.

Descriptor — Descriptors are defined attributes that describe a characteristic value. For example, a descriptor might specify a human-readable description, an acceptable range for a characteristic's value, or a unit of measure that is specific to a characteristic's value.

Service — A service is a collection of characteristics. For example, you could have a service called "Heart Rate Monitor" that includes characteristics such as "heart rate measurement." You can find a list of existing GATT-based profiles and services on bluetooth.org.

3.6.4 Roles and Responsibilities

Here are the roles and responsibilities that apply when an Android device interacts with a BLE device:

Central vs. peripheral. This applies to the BLE connection itself. The device in the central role scans, or looks for advertisement packets, and the device in the peripheral role makes the advertisement packets. GATT server vs. GATT client. This determines how two devices talk to each other once they've established the connection. To understand the distinction, imagine that you have an Android phone and an activity tracker that is a BLE device. The phone supports the central role; the activity tracker supports the peripheral role (to establish a BLE connection you need one of each—two things that only support peripheral couldn't talk to each other, nor could two things that only support central).

Once the phone and the activity tracker have established a connection, they start transferring GATT metadata to one another. Depending on the kind of data they transfer, one or the other might act as the server. For example, if the activity tracker wants to report sensor data to the phone, it might make sense for the activity tracker to act as the server. If the activity tracker wants to receive updates from the phone, then it might make sense for the phone to act as the server.

3.6.5 iOS

The Core Bluetooth framework provides the classes needed for your iOS and Mac apps to communicate with devices that are equipped with Bluetooth low energy wireless technology. For example, your app can discover, explore, and interact with low energy peripheral devices, such as heart rate monitors and digital thermostats. As of OS X v10.9 and iOS 6, Mac and iOS devices can also function as Bluetooth low energy peripherals, serving data to other devices, including other Mac and iOS devices.

In Bluetooth low energy communication, there are two key players: the **central** and the **peripheral** (much like Android). Each player has a different role. A peripheral typically has data that is needed by other devices. A central typically uses the information served up by a peripheral to accomplish some task. For example, a digital thermostat equipped with Bluetooth low energy technology might provide the temperature of a room to an iOS app that then displays the temperature in a user-friendly way.

Each player performs a different set of tasks when carrying out its role. Peripherals make their presence known by advertising the data they have over the air. Centrals scan for nearby peripherals that might have data they're interested in. When a central discovers such a peripheral, the central requests to connect to the peripheral and begins exploring and interacting with the peripheral's data. The peripheral is responsible for responding to the central in appropriate ways.

3.6.6 Apple Notification Center Service (ANCS)

For enabling our desired behaviour of our wearable we need to take a look at ANCS, which enables us to access notifications from wearable. To access notifications as it is possible in Android is not possible in iOS devices, because of privacy reasons on iOS.

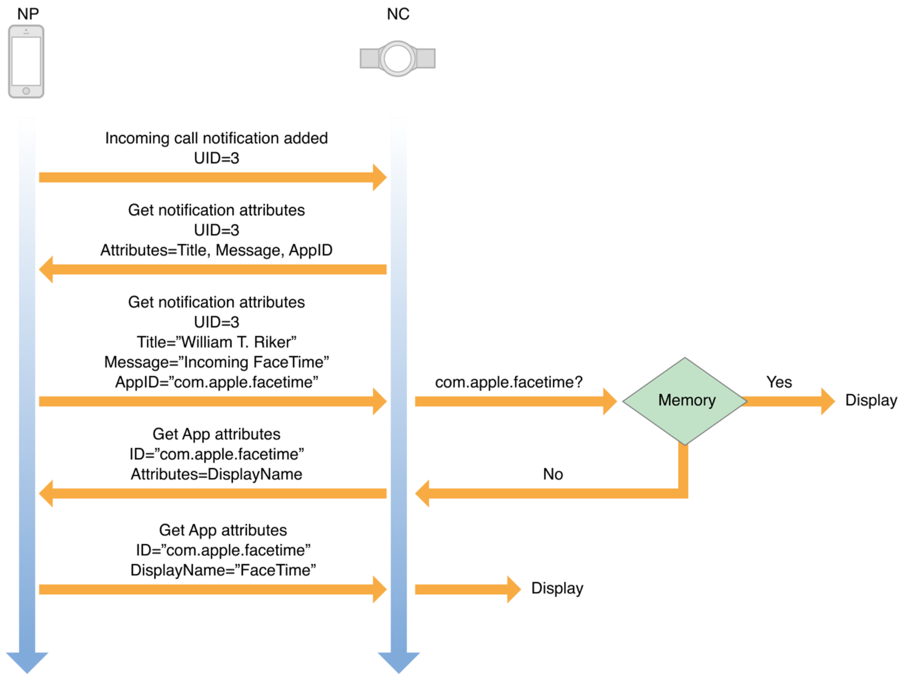

Figure 3.14: Notification attribute retrieval example

On figure above we can see the diagram how are the notifications accessed on iOS device. First on the peripheral (smartphone) is created an event. (Incomming call notification added). This event is send to the central element (wearable), which then needs to get more information about the notification. After that we can show the notification on a werable, or if we need to access the application name, we can ask the peripheral (smartphone) about it and display the notification after that.

3.7 Solution proposal

Based on sub-chapters above we can define our prototype. What features, technologies and for what target group it has to have. We will closer examine these variables in chapter Realisation, if they still make sense for our target group.

We will build a prototype of a smart device attaching to users wrist, with notification features and health data gathering. This device will communicate through Bluetooth Low Energy with smartphones. On these smartphones will run application which will be storing data and possibly send them to server, making rules for your health data and notifications.

3.7.1 HW Device

The device itself will run on nRF51822. The nRF51822 is built around a 32-bit ARM® Cortex™ M0 CPU with 256kB/128kB flash + 32kB/16kB RAM for improved application performance. The embedded 2.4GHz transceiver supports both Bluetooth low energy and the Nordic Gazell 2.4 GHz protocol stack which is on air compatible with the nRF24L series products from Nordic Semiconductor.

The user research concludes that we need to have accelerometer in this device and we will use the LS33DLH accelerometer chip. The accelerometer chip is an ultra low-power high performance three axes linear accelerometer belonging to the "nano" family, with digital I2C/SPI serial interface standard output.

To power these chips we will need a battery with at least 3.5 output voltage and at least 100 mAh for providing users with enough battery life to fullfill their requirement. For enabling notification feature we will need an RGB LED, we chose white light, as this project is only in prototype state.

All of above stated chips will be soldered on the mother board with an 2.4 GHz antena for nRF51822 chip.

3.7.2 Applications

Applications will be the main part of this solution that is why it is important to have a good design of these applications. Low-fidelity protypes should be created in tool Balsamiq 3 and with this tool they should be tested with users.

First we need to define our user requirements on these applications. User requirements can be divided into two categories:

1. functional - are defining, what the application/program should do, what features are expected to be fullfilled

2. general - contectual requirements consequent to users environments

3.7.3 User requirements

We will define our requirements with help of use cases, which we will divide based on the part of solution (HW/applications).

Therefore we have user requirements for 2 programs smartphone application and wearable device.

3.7.3.1 Smartphone application

This section describes user requirements for smartphone applications for both current major platforms (Android, iOS).

1. BLE connection:

   (a) Display the names of nearby BLE devices.

   (b) Allow the user to connect to our wearable supporting BLE.

2. Activity tracking:
   (a) Upon opening the application send request about how many steps were walked.
   (b) Delete steps taken during one day.
   (c) Persist the steps taken during the day.
   (d) Display users daily step count and calories burnt.

3. Notifications
   (a) Notify users wearable about new incomming notification.

4. General
   (a) Allow user to remove all the persisted data
   (b) Display user detail (height, weight)
   (c) Allow user to set his height and weight

3.7.3.2 Wearable device

Similarly we need to define user requirements for the hardware device and its firmware.

1. Be discovered when when battery is connected
2. Detect steps
3. Save steps
4. Remove all detected steps at 3 AM
5. Allow the smartphone application to set internal time.
6. Allow the smartphone application to get detected steps.

Chapter 4

Realisation

This chapter describes what was created based on user requirements from chapter Solution Proposal.

4.1 Hardware

In hardware department were created 2 and a half prototypes, more or less fullfiling proposal.

4.1.1 First prototype

First prototype was using module from external vendor and was mainly used for sotware development. This module contained MCU nRF51822, which has bluetooth chip inside alongside the accelerometer LIS331DLH. This module was then attached to external antenna via standard antenna connector, and programmer board. This prototype did not have its own battery nor notification capabilities. As you can see on image below this prototype was very untestable. For the software development it was enough. Everything was working but I was not completely happy with the result, while it could not be properly tested, because of the antenna and battery from USB programmer.

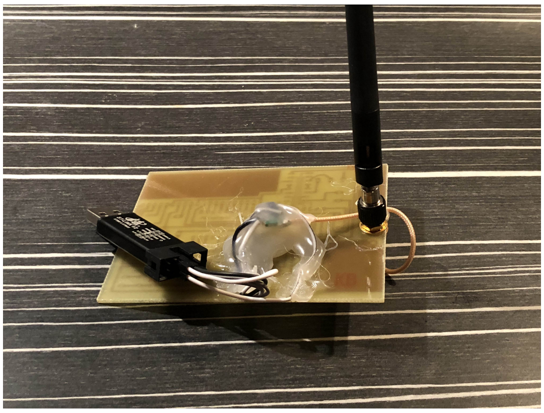

Figure 4.1: Functioning prototype

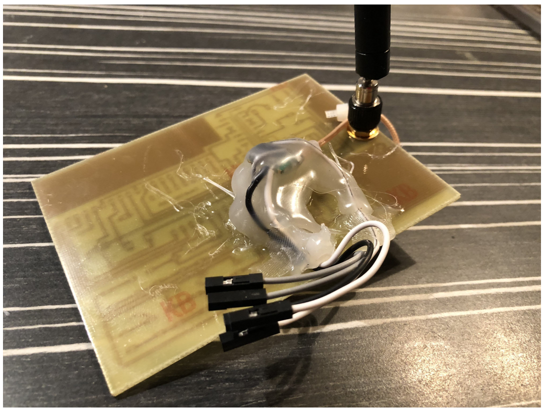

Figure 4.2: Functioning prototype

Figure 4.3: Functioning prototype

4.1.2 Second prototype

Board for second prototype was made on order in PragoBoard and it should be assembled by TTC, after months of fine tuning everything was ordered and shipped to TTC. The result of this work and communication was a failure. Because nobody realized that template for this board and especially for BGA chip was using 0.18 mm openings for paste and the company had limit on 0.28 mm, therefore they couldn't assemble it and every company we asked for assembly said that only chance to finish the job is in Q1 2018. What is a bit late.

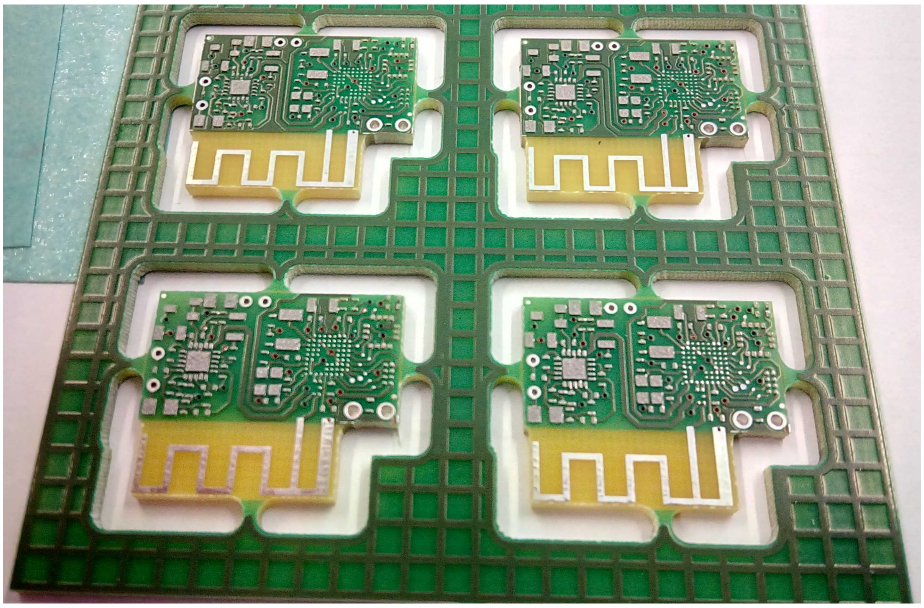

Figure 4.4: Manufactured prototype

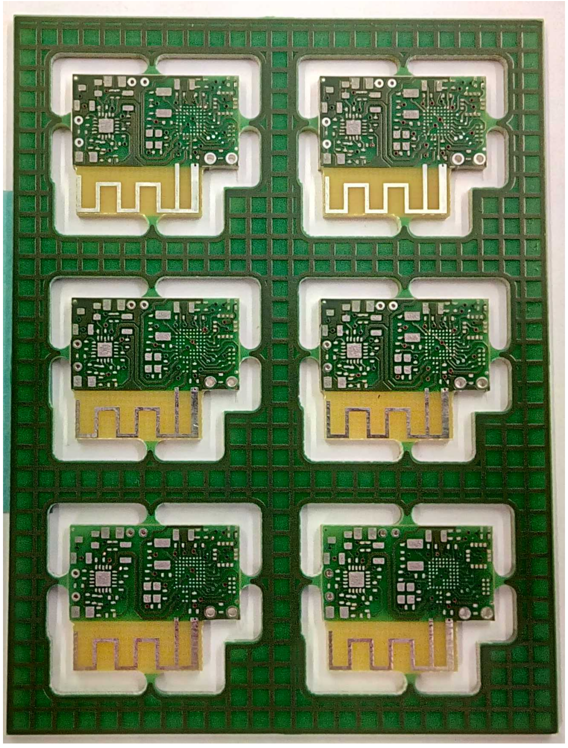

Figure 4.5: Manufactured prototype

4.1.3 Third - final prototype

After fiasco with TTC and 0402 components we tried to assemble the circuit oursleves with modules from first prototype. And that was succesfull, finally! But all the agreements we had with case manufacturer were irrelevant, because of changed physical dimensions. Therefore the tested prototype was not as user friendly as was planned, as we can see on the next figure.

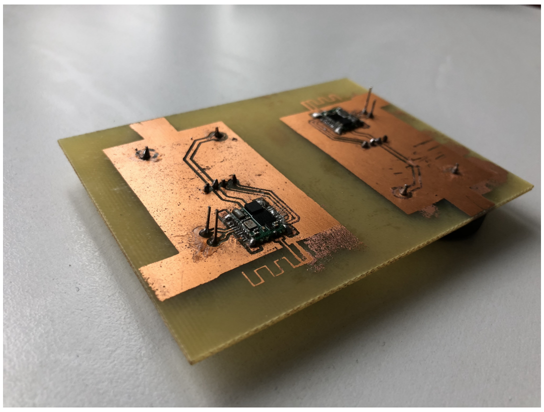

Figure 4.6: Functioning prototype

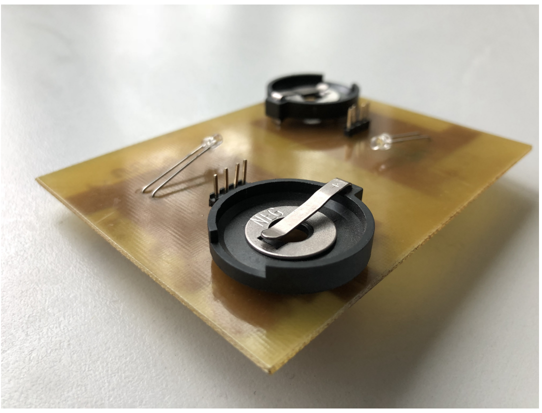

Figure 4.7: Functioning prototype

4.2 Case

Case went through a bit of development as well. First prototype had 2 legs for connecting the device onto the watch band as you can see on the next figure.

Figure 4.8: First case

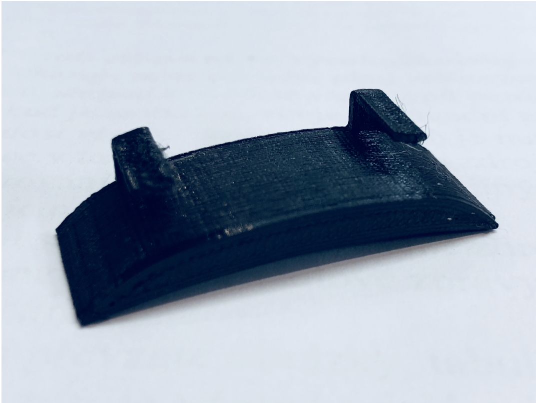

Figure 4.9: First case

This solution showed as very unintuitive and unpleasant for using with a watch, therefore we changed the design for one sturdier leg. This solution proved to be more pleasant to use and it had an advantage of it being able to leave on the band itself.

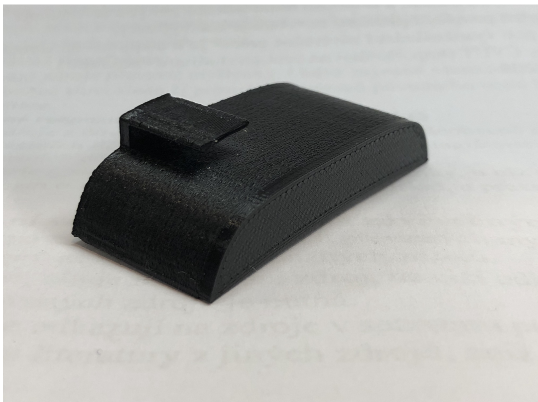

Figure 4.10: Final case

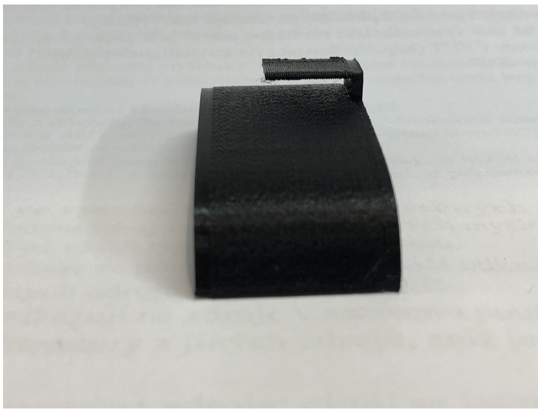

Figure 4.11: Final case

4.3 Software

With this prototype were created 2 different softwares. One in C for wearable based on Nordic examples and second one in Java for Android devices.

The wearable part was created with major help of tutorials for nRF51822 chip, which are handling sending the advertisement packets as well as the connection itself. The parts needed in developing the software for werable ware the responses for smartphone requests, steps detection alogrithm and notification capabilities.

4.3.1 Step detection alogirthm

There are many different step detection algorithms with different precisions and walk models. Different models are able to distinguish between many different activities such as running, walking, swimming and many more. Probably the furthest and most advanced algorithms are inside Apple Watches, because they contain even detection of wheelchair movements. We are going to use very simple model with directional counting of values and a threshold on these values.

Our algorithm works with raw accelerometer data, these data are being read from SPI (Serial Peripheral Interface) input source (accelerometer) and are assigned to variable based on the direction of movement (X, Y, Z). We have to check if we want to measure data, if we do not want these data, then we discard these values. Each value is then compared with threshold of 0x7F hexa, or 127 decimal, if the move is less then our threshold, values for this movement are discarded. The threshold value was defined by experimenting through development, for activity walking. After comparing actual value with threshold we add 1 to counter how many (signed) accelerations were made in this direction. If this counter is greater than "compare_value" (currently set to 8, thtis value was set during testing the algorithm while developing it) then we count this movement as a whole step and reset all these values (except step count). Simultaneusly we are checking the frequency of steps and we are assuming that user will walk with frequency up to 5 steps per second.

We are aware that there are many problems with this algorithm. In my opinion the two most important issues are discarding values while running (user can run faster than 5 steps per second). Next problem is that we are currently not checking full movements, only accelerations in direction and that user can fake the steps rather easily. There is definitelly work left in this algorithm.

4.3.2 Android - MVVM

Architecture is very important aspect in developing applications. Few years ago were these applications only thin clients and therefore the question of architecture was not that important as these applications had only either one easy funcionality or were View of a bigger system with requests. Meanwhile Android as well as iOS applications grew in funcionalities, features and became smart clients, with cached data and/or AI features, or more general with more advance algorithm for data manipulation.

With recent Google release of Live Data package in mind, we have decided for MVVM (Model - View - ViewModel) approach. With this architecture is the application at least a little bit future-proof and ready to be extended with new funcionality.

4.3.2.1 Model

Model in MVVM has a few responsibilities. It abstracts the data source. The ViewModel works with the DataModel to get and save the data either from remote or local repository.

The Model with RxJava/RxKotlin extensions are exposing data easily consumable through event streams— Observables. It composes data from multiple sources, like the network layer, database or shared preferences and exposes easily consumable data to whomever needs it. The Models should hold the entire business logic for particular contexts, or features.

Earlier mentioned Google Live Data package is usable in Model layer as we can see on image below.

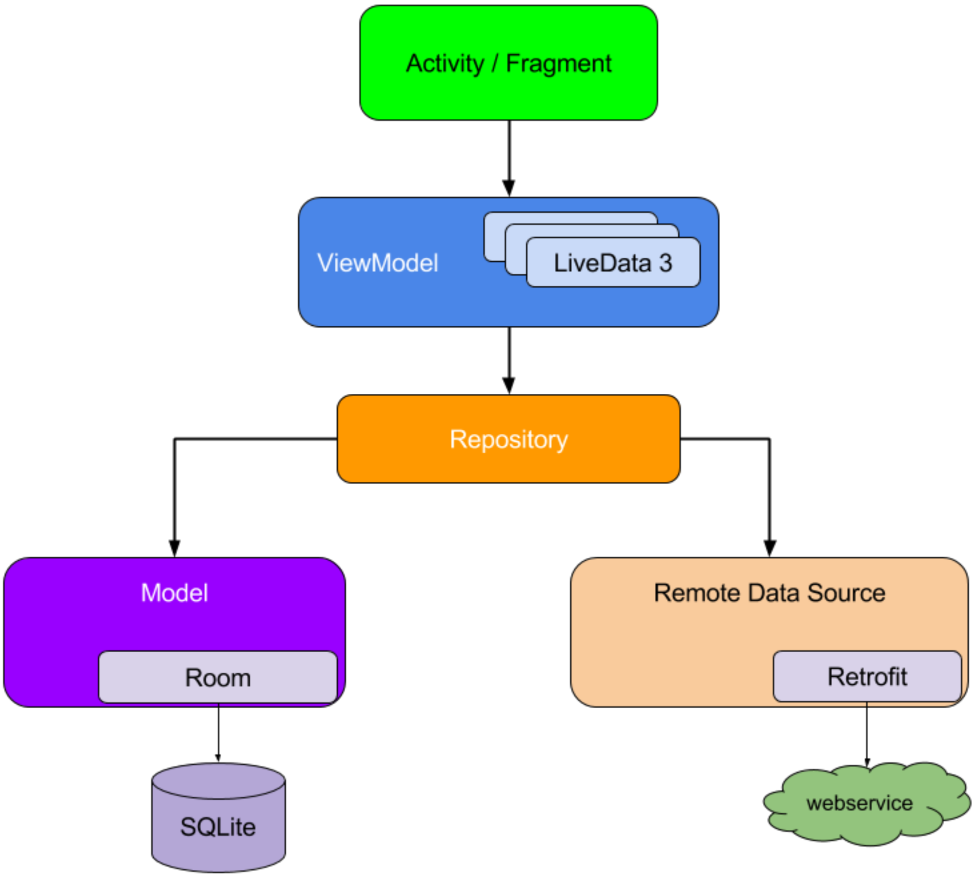

Figure 4.12: MVVM with Android packages [8]

4.3.2.2 View

The View is the actual user interface in the app. It can be an Activity, a Fragment or any custom Android View. For Activities and Fragments, we are binding and unbinding from the event sources on onResume() and onPause().

If the MVVM View is a custom Android View, the binding is done in the constructor. To ensure that the subscription is not preserved, leading to possible memory leaks, the unbinding happens in onDetachedFromWindow.

4.3.2.3 ViewModel

The ViewModel is a model for the View of the app: an abstraction of the View. The ViewModel retrieves the necessary data from the DataModel, applies the UI logic and then exposes relevant data for the View to consume. Similar to the DataModel, the ViewModel exposes the data via

Observables.

The ViewModel should expose states for the View, rather than just events. For example, if we need to display the name and the email address of a User, rather than creating two streams for this, we create a DisplayableUser object that encapsulates the two fields. The stream will emit every time the display name or the email changes. This way, we ensure that our View always displays the current state of the User. We should make sure that every action of the user goes through the ViewModel and that any possible logic of the View is moved in the ViewModel.

4.3.3 iOS - B-VIPER

iOS development is similar story as in Android. Before architecture concerns were raised, many applications had MVC architecture, but in this case it did not mean Model-View-Controller rather Massive ViewController. As in iOS is UIViewController the main UI component, with all the delegates and callbacks. With the release of Swift came possibility to write protocols and with Swift 3 came ability to fully implement protocol oriented programming (POP). In my opinion offers POP more flexibility than classic object programming, because while protocols are like interfaces they offer to implement default behaviour and with that they look more like abstract classes with few advantages, such that protocol can be "implemented" not just by class, but struct as well as enum types. Other advantage is that class, struct or enum can "implement" more than one protocol and therefore you partially achieve multiple inheritance in your code.

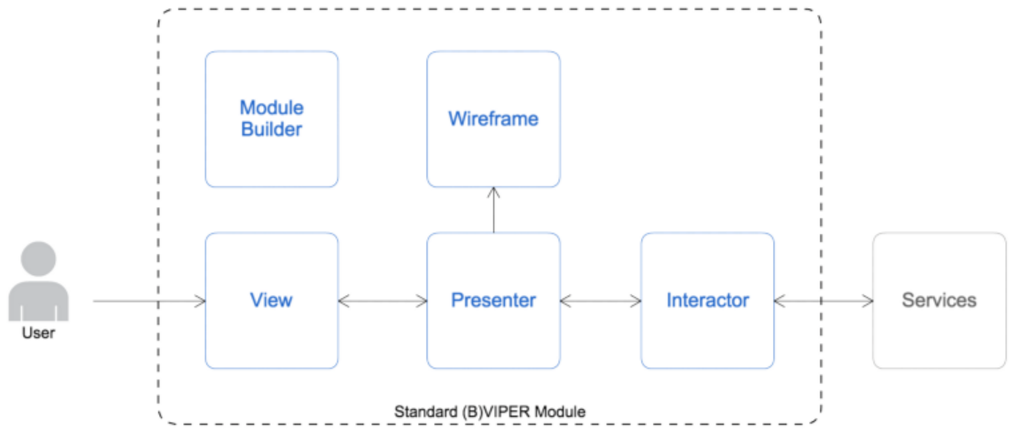

Figure 4.13: B-VIPER base structure [9]

B-VIPER is extended VIPER architecture with builder pattern, and VIPER at itself is View, Interactor, Presetner, Router and Entity. B-VIPER is usable for developing Pods/packages as well as main source code. VIPER itself has origins in Clean architecture, which popularized Robert C. Martin (Uncle Bob). As you can see on the image below clean architecture is often compared to an onion. In its center is entity layer, which contains all enterprise business logic, all models with whose we can work inside this business logic. Next layer (in figure Use Cases) handles all of the application business logic, the main differecne between enterprise logic and application logic is best explained on an example:

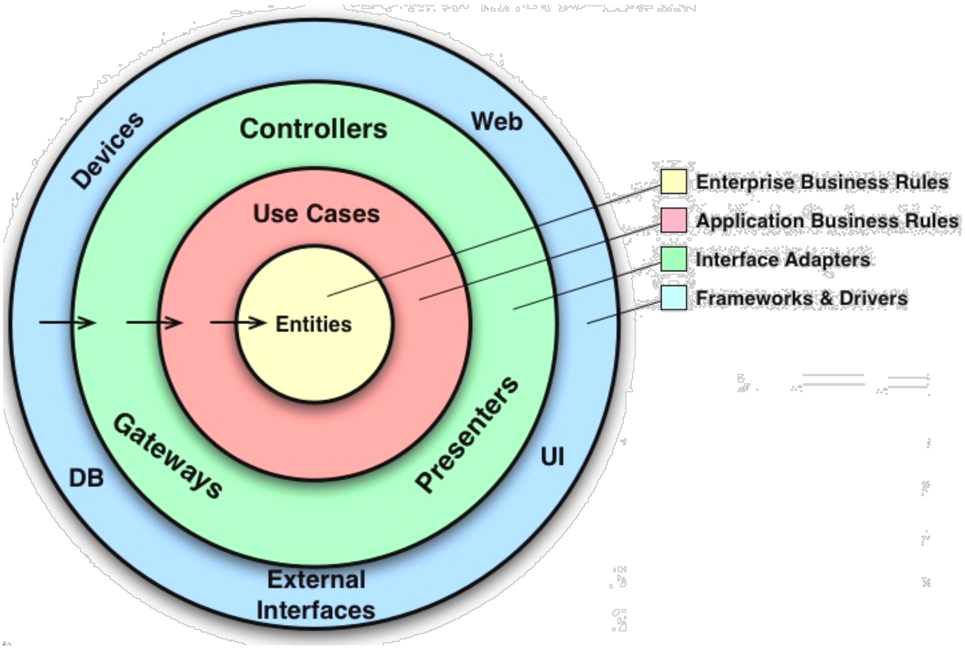

Figure 4.14: Clean architecture [10]

Let us imagine an e-shope enterprise logic is how the payment is processed, when to decrease the stock, what happens if user has insuffiecient funds and so forth. Application business logic is which screen to show based on transaction status inside the enteprise logic.

One before last is layers of presenters, controllers and gateways to another systems with whose we have to communicate. They define the logic for its view, or for API of another system. This layer should have its own models which are testable and therefore independent from the last layer.

Last layer contains platform specific code, by platform we mean the databse SDK, the UI layer, the web API and so forth. These parts of the system can use platform specific models and therefore they do not need to be easily testable, but they should be still testable. In this layer should not be any logic and it should only transfer data from inner (our) system into platform specific code. The outer layer needs to be easily replacable, for example we want to go from SQL to NoSQL databae, without logic and state should be layer parts easily replacable.

1. Builder - itsresponsibility is to instantiate all dependencies and components needed for each particular feature. Builder is the key for reusability, declares what data is needed for the Module to be built, inject all the needed components letting the implementations change easily. A Builder must know the next Module(s) Builder.

2. View - has very simple responsibility from algorithmical standpoint, its main purpouse is to show UX optimized and graphically pleasing user interface, next responsibility is to

set callbacks to users interaction such as buttons, pickers and so on. View should hold reference to its presenter.

3. Presenter - its responsibility is to implement logic for view, when to show some component, or when to disable based on the interators state or as a reaction on event in the features interactor. This layer represents second most outer layer in clean architecture model.

4. Interactor - this layer contains logic for every feature. Every feature should have its own class with own particular dependencies on database, network layer and so on. This layer is "Use Cases" layer in clean architecture model.

5. Entity - model serves more purposes. Firsrt is the real entity model, we imagine under the name Entity and second is implementation of enterprise business logic. Often here are added here services with single responsibility and different mathematical operations. Basically in this layer is done all the heavy lifting.

6. Router - are relatively simple classes which handle what follows after this feature/screen/event is fired. Often there are two niveaus of routers: AppRouter for routing whole application and deciding what to show to user after launching the application and feature Router, which handles wizard navigation, or navigation between features.

4.4 Final prototypes

This section shows final Android prototype with which the users tested with a few comments.

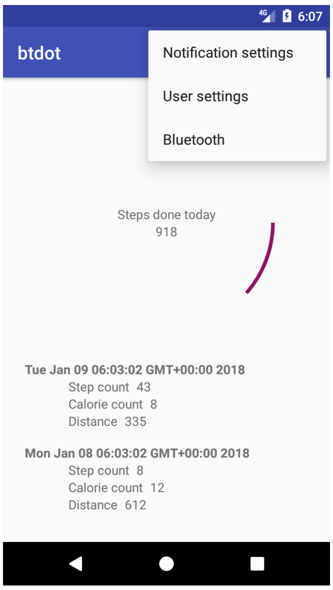

Figure 4.15: Main screen

Main screen contains main dial with steps taken today, while under it we can see data from the last 7 days.

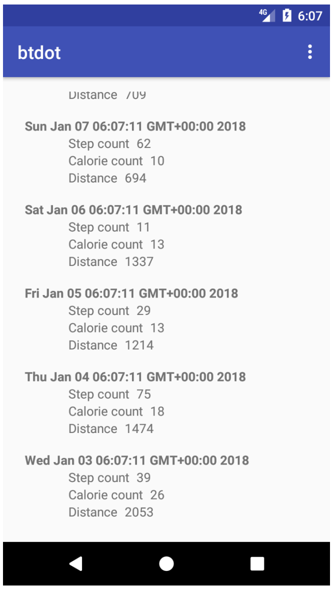

Figure 4.16: Main screen scrolled

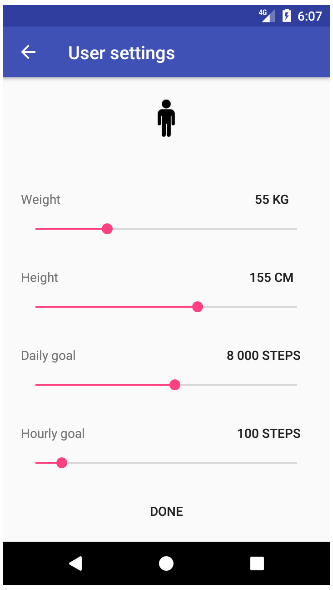

Figure 4.17: User screen

User screen above is scrollable due to possibility of smaller displays.

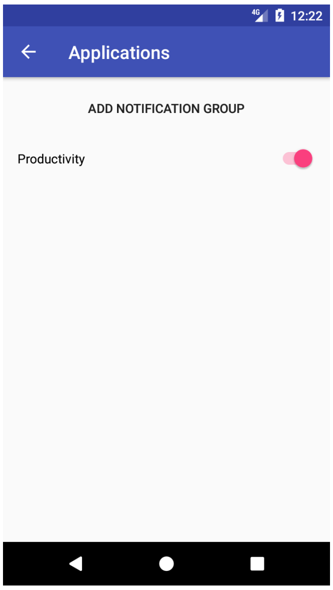

Figure 4.18: Notification group screen

Above we can see the screen for quick toggle of notification groups. While on the next screen we can see all installed applications on current smartphone.

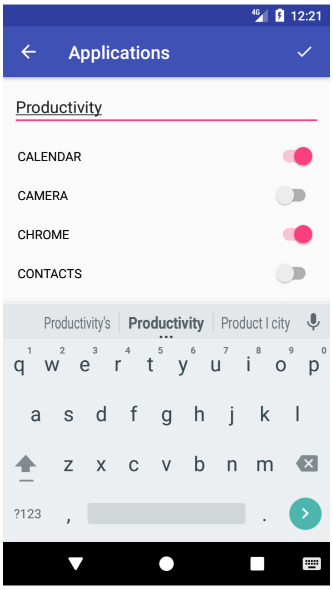

Figure 4.19: Applications screen

Chapter 5

Testing

Testing is very important part of every project. This project needs to utilize user testing of a whole solution as well as basic funcionality of this solution. Main questions we will try to answer are:

- Are users satisfied with general prototypes shape, placement and attachment to the watch? What would they change?
- Would be, in their opinion, better to have the device permanently attached to watch/band or is this solution with removing and reattaching the device for them pleasant?
- Is the applications onboarding process with device attachment and user information setup understandable?
- Do users understand where from which source are shown data?
- How happy are users with the current state of a prototype?

5.1 Usability testing

The test was concluded with 2 users to whom was provided a prototype. One hollow 3D printed model and one functional board with step counting and notification capabilities. We have to take into account status of physical devices and the fact, that users are getting 2 different devices. Users got both devices to take home and use over 2 nights as they please to experience the sensation of owning similar device. With that we got more real feedback on the devices and application.

Before users got the device to take home, they were subjects of classical usability test on mobile application with following tasks:

1. Connect to supported HW device, after starting the application.
2. Fill up all required fields to start using application in its fullest.

3. Create new notification group from which you do **NOT** want to receive notifications (Messenger, WhatsApp)

4. Create new notification group from which you do want to receive notifications (Gmail, Calendar)

5. Edit your hourly step target to 500 steps

6. Change your weight to new value 55 kg.

Before usability testing it is important to get users context, we do not need to do that, because we are using same users as at the begging of the project. Although after every usability testing is important to debrief users and mainly to state that what was tested was the solution and not them and at the same time to try and get any ideas how to make the product better.

Longer term hardware testing will be evaluated based on interviews with users.

5.2 Evaluation

We are using participants 3 and 5 from beginning of the project as they are male and female, one with experience in smart watches and other without it. The participant without the wearable experience was very interested in this project and the other participant was happy with his smart watches, therefore we have results from a potentionally eager user and potentionally pesimistic user of this solution. Both of these users have Android devices. To the user withouth classical watch was lend my watch for testing period.

Before this longer term test we concluded classic usability testing. The tests were concluded in office environment in Prague on 18.12.2017, at this time started the longer term testing, with these two test participants.

5.2.1 Usability test

In first task was only problem that one participant had not turned Bluetooth on and after the dialog about notifying him about he expected that bluetooth was turned on with this dialog, this is a feature of Android and therefore, sadly, we cannot do anything about it.

Second task was pretty straight forward and no problems were discovered. Users were able to get that this information is being saved into device and no further actions were needed to adjust the application.

After completing the first two tasks, got user to the main screen, where they saw empty counter, without any history. Therefore we continued to next assignement. We could see the confusion with task formulation, but participants were able to complete the next two tasks easily after quick explanation about the task formulation.

Participants were pleased with the possibility to set their daily targets, not so about the hourly targets.

The last task was easy for both of them, while it was on the same screen as task before. This task was the start of the debriefing process.

Debriefing continued with post-test interview, where the participants were properly debriefed and asked about possibilities to make this solution better. Their ideas can be united into few statements:

- the concerns about the long term stability of the case attachment to the watch
- they stated that prettier UI for the application would be appreciated
- they missed the possibility to see whole history of steps
- they missed more statistics (graphs and goal reaching)

Chapter 6

Conclusion

At the beginning of the project was a question: Is there a way to enable people to gather health data and receive notification on classical watch? After doing a user research on 7 people we conclude that people want to gather health data and receive notifications, while at the same time they have emotional or style attachments on their classical watch, therefore we can conclude that there is a market for such a device. This statement is backed by fully backed kickstarter project called Smart Buckle.

Next chapter contained informations needed for developing such a solution, what requirements should the device full-fill, what technologies should be used and what is the main target group for this device. Our design process was inspired by book Design Thinking from H. Plattner, Ch. Meinel and L. Leifer. There were created low-fidelity prototypes, which were tested and in this work we could see the final low-fi prototypes.

Realisation was really challenging, because the hardware prototype needed to be created with the help of external companies. This companies were PragoBoard and TTC Telekomunikace, we had long email conversations about specifications so that TTC is able to assemble the components on the board from PragoBoard. After everythnig was created and ordered TTC had a problem with the holes size on a alumiunum template. TTC could not push paste through the holes. Therefore we created plan B, which could not be incorporated into 3D printed case. Android application was created as it is main platform in the current market.

Testing was therefore done in two parts. First was classic usability test with 2 users, tasks and post-test interview, while the second test was longer time testing about using the case. We conclude that there are many challenges before us with this project.

More work is definitelly needed, we need to develop better iOS support, better case design, finish hardware prototype (assemble) and properly test this. After the development we need to focus on materials, marketing and financional stability of this whole project. We will try to get the product to market in Q3 2018, ideally with price up to 50 EUR, what is limit price for the users.

Bibliography

[1] PLATTNER, Hasso, Christoph MEINEL, Larry LEIFER a EDITORS. Design thinking: understand - improve - apply. Heidelberg: Springer, 2011. ISBN 9783642137563.

[2] Wearable fitness device adoption rate worldwide in 2016, by country [online]. USA: Statista, 2016 [cit. 2018-01-01]. Dostupné z: https://www.statista.com/statistics/651681/wearable-fitness-device-adoption-by-country/

[3] Top 5 smartwatches CES 2014 [online]. USA: Forbes, 2014 [cit. 2018-01-01]. Dostupné z: https://www.forbes.com/sites/arieladams/2014/01/13/top-5-smartwatches-ces-2014/#72b2f342737f Core Specifications [online]. USA: Bluetooth SIG, 2017 [cit. 2018-01-01]. Dostupné z:https://www.bluetooth.com/specifications/bluetooth-core-specification

[4] What Material Should I Use For 3D Printing? [online]. USA: 3D Printing for beginners, 2013 [cit. 2018-01-01]. Dostupné z:http://3dprintingforbeginners.com/filamentprimer/

[5] Types of 3D printers or 3D printing technologies overview [online]. USA: 3D Printing from scratch, 2015 [cit. 2018-01-01]. Dostupné z:http://3dprintingfromscratch.com/common/types-of-3d-printers-or-3d-printing-technologies-overview/#fdm

[6] What's the Difference Between Stereolithography and Selective Laser Sintering? [online]. USA: MachineDesign, 2015 [cit. 2018-01-01]. Dostupné z:http://www.machinedesign.com/3d-printing/what-s-difference-between-stereolithography-and-selective-laser-sintering

[7] Smart buckle [online]. USA: Kickstarter, 2017 [cit. 2018-01-01]. Dostupné z:https://www.kickstarter.com/projects/20345353/smart-buckle-convert-any-watch-into-a-smartwatch-i?ref=21x16q&utm_source=A20FB&utm_medium=FB&utm_campaign=TAP&utm_content=D30

[8] It's time to use MVVM as your Android's app base architecture [online]. USA: AndroidPub, 2017 [cit. 2018-01-01]. Dostupné z: https://android.jlelse.eu/its-time-to-use-mvvm-as-your-android-s-app-base-architecture-part-1-cf0ce2542b48?gi=d2aa8f042d47

[9] Architecting Mobile Apps with (B)VIPER Modules - A structured engineering approach for big mobile apps [online]. USA: Travelport Digital, 2017 [cit. 2018-01-01]. Dostupné z: `http://digital.travelport.com/blog/architecting-mobile-apps-with-bviper-modules`

[10] Clean architecture [online]. USA: Jordi Fierro, 2017 [cit. 2018-01-01]. Dostupné z: `http://jordifierro.com/django-clean-architecture`