Master's Thesis

# Touch screen mobile user interface for seniors

*Bc. Tomáš Slavíček*

Praha 2014

Supervisor: Ing. Jan Balata

Czech Technical University in Prague
Faculty of Electrical Engineering, Open Informatics,
Department of Computer Graphics and Interaction

## Acknowledgement

I would like to thank to following people:

- Ing. Jan Balata for supervising my work and for his valuable feedback during the whole work
- Sebastian O'Rill from LG Electronics CZ, s.r.o. for supporting me in the original idea and giving the access to mobile phones for testing and development
- Josef Slavíček for helping me with the graphical design of the prototype
- To all other people who participated in this thesis, for their willingness and patience during the whole process


## Declaration

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing academic thesis.

# Abstrakt

Tato práce se zabývá návrhem mobilního rozhraní pro dotykové telefony uzpůsobeného pro starší osoby. Typický moderní smartphone má velký displej, ale jeho prostředí je poměrně komplikované. Provedli jsme kvalitativní výzkum mezi staršími lidmi. Vyjádřili jsme několik hypotéz, ty jsme ověřili kvantitativním výzkumem. Identifikovali jsme konkrétní schopnosti a omezení starších lidí a shrnuli doporučená pravidla pro přístupnost.

Navrhli jsme funkční prototyp, otestovali několik variant jeho rozložení se seniory a probrali specifické problémy v použitelnosti. Nakonec jsme provedli empirické ověření prototypu a porovnali ho se standardním rozhraním Androidu. Věnovali jsme se také technologii stojící za tvorbou prototypu, jak efektivně použít herní framework pro tvorbu aplikace a jak dosáhnout nízké spotřeby baterie.



## Klíčová slova

senior, důchodce, staří lidé, mobilní, rozhraní, prostředí

## Abstract

This thesis focuses on the problem of designing a user interface for touch screen mobile phones optimized for elderly people. Typical modern smartphone has a large display, but its user interface is fairly complicated. We have performed a qualitative research between old people. Several hypotheses were expressed and verified by a quantitative research. We have identified the specific capabilities and limitations of old people, as well as summarized the recommended accessibility guidelines.

We have designed a working prototype, tested several variants of the layout with elderly and discussed specific usability problems. Finally we did an empiric evaluation of the prototype and compared it with the standard Android interface. Technology behind the prototype was also discussed, how to efficiently use a game framework for creating and app and how to achieve the low battery consumption.

### Keywords

senior, elderly, old, mobile, interface, launcher

# Contents

# 1. Introduction

Majority of phones sold at the beginning of the year 2014 had a touch screen [1]. Typical modern smartphone does not have dedicated hardware buttons to dial a number or to receive a phone call. The only possible way how to interact with the phone is to use virtual on-screen buttons and gestures on the display.

**Touch-based** interface could be confusing for many people, especially for seniors. Three representatives of the major mobile platforms are shown on a Figure 1 (Apple iOS 7, Android 4.4 and Windows Phone 8.0). These interfaces are mostly not optimized for people with vision problems or sensoric-motoric problems. Some accessibility features can be set, but they are not sufficient for most old people. Small icons have a tiny text. The usual color contrast is poor, interface is complicated and does not look consistent between individual screens.

It is assumed that **elderly people** mostly do not want to use simple one-button senior phones. They want to stay in touch with their family, use a camera for taking photos or another useful features like an alarm clock. They want to use their phones without restraints. Classic feature phones with hardware buttons may also not be entirely good for them, they also have very tiny labels on a keyboard and small displays.

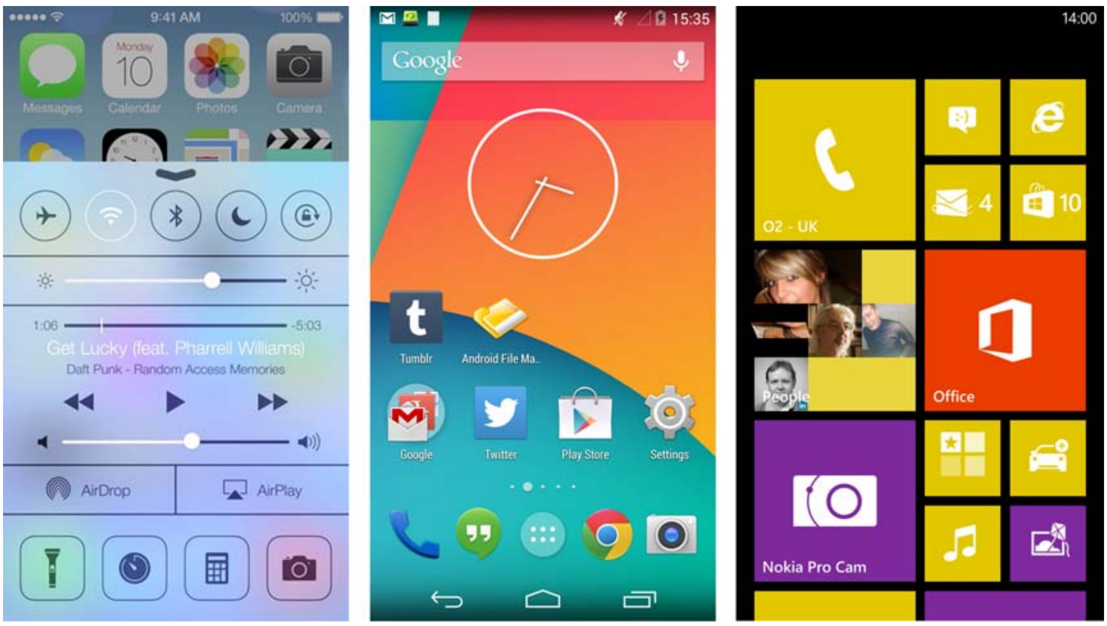


**Fig. 1.** The most common mobile operating systems: iOS 7, Android 4.4, Windows Phone 8.0

## 1.1. Main objectives of the thesis

The **main objectives** of this theses are to research the problem how old people uses their phones. To find the most common usage patterns and recommendations and verify them. To design a new user interface for touch screen phones that will be optimized for elderly users. Working prototype will be created, tested and compared with the traditional Android user interface.

The **ambition** of this work is to design a new set of screens, panels and buttons for a generic touch screen phone. Colors, button positions and animations will lead to a simplification of the most common tasks. Thanks to enlarged buttons, bigger labels and other accessibility features (like a sound or vibration feedback of buttons), more people will be able to access modern technologies. Keyboard will be also enlarged and adjusted to simplify writing and sending messages.

**Interface** will be designed as a complete replacement of a standard phone interface. Even if the user loses or break his phone, he will be able to install this interface on another physical device again, even from a different brand, and use it in the same way.

In summary the main objectives of this theses are:

- To research the problem how old people uses their phones.
- To identify and verify common usage patterns of elderly users.
- To design a working prototype of a new mobile interface and test it with people.
- To compare and measure it with the standard Android interface.

## 1.2. Research methodology

The first part of the research is the **analysis** of the most relevant resources in the literature. What are specific capabilities and limitations of older people. What accessibility guidelines are recommended for them and which solutions already exist. We will go through existing mobile interfaces designed for seniors and compare them. We will discuss what could be possibly made better.

The own **qualitative research** will be performed. We will meet a couple of old people and ask them how they use their phones, what they like and dislike. We will obtain as much as possible real-world experiences from them. On the bases of these interviews, several hypotheses will be expressed.

These hypotheses will be discussed and verified by a **quantitative research**. A questionnaire will be prepared. It will be distributed over the internet, as well as presented personally to another few of elderly. We will try to find out their motivation once again and obtain more information. We will identify specific personas based on their experience.

Based on the hypotheses, the **working prototype** of the application will be designed. We will start from a couple of paper mockups, several variants of layouts will be tested, one specific will be selected. Every each iteration of the prototype will be tested with elderly, the app will be improved and fixed. Specific usability problems will be discussed.

The prototype will be created in a cross-platform game framework, we will discuss our **optimization** during the development, how to efficiently load images and system resources, and how to effectively decrease a battery consumption of the app.

The final prototype will be tested with a couple of elderly once again. We will do an **empirical evaluation**, the quantitative statistical test. The prototype will be compared with the standard Android 4.4 interface. Several people will accomplish very specific given tasks. We will measure their execution time and a number of errors. We will compare how both younger and older people can effectively use this interface.

## 1.3. Organization of the thesis

This theses is divided into 6 sections. In the *second* section (State of the art) is presented the analysis of the most relevant resources about old people and mobile interfaces.

In the *third* section the qualitative research is performed. On the bases of how old people use their phones, several hypotheses are expressed and verified by a quantitative research. In the *fourth* section the main design decisions about user experience are described. The mockups are designed and the final prototype is created.

The *fifth* section contains the summary about all usability tests performed during the process. The prototype is also compared with the Android 4.4. interface, by an empiric evaluation. The final *sixth* part summarizes the outputs of this theses and presents the additional possible future work on this topic.

# 2. State of the art

The first part of our research was to **analyze** the most relevant resources in the literature. We were interested in specific capabilities and limitations of old people, in context to modern technologies, digital devices and touch-screen phones.

Very detailed book is a *Designing Displays for Older Adults* [2]. It is a thorough guide how vision, hearing, cognition and movement of people changes with age. Specific examples how to improve the user interface are shown, for example how to redesign a mail application on a classic feature phone, or the interface of the set-top-box.

Similar to this book is a *Designing for Older Adults* [3]. It provides a detailed summary about the design for old people in general, for example how to improve the healthcare or the work environment. It also describes in detail how to guide the design process and how to perform usability tests with old users.

Very beneficial is an article [4] from year 2012. It summarizes many other articles about touch-screen interfaces for old people, also contains a quick summary of the recommended design guidelines. Authors from South Korea [5] in 2007 did a very extensive research and published a lot of suggestions how phone for seniors should be designed. Another research was made in Malaysia [6] in 2008.

The article [7] (2010) proposes how to design a worth-centered mobile phone design for old users. OldGen [8] (2011) presents a customizable phone user interface for seniors, that is independent on a mobile platform. PhoneAge [9] (2013) is another project that presents the accessible & adaptable solution for smart phones.

An article [10] is focusing on a design and accessibility of mobile web pages. Article [11] presents the recommendations how to design a senior-friendly icons in the menu.

## 2.1. Context of the research

Many articles researched the behavior of old people and described how to design an accessible mobile interface. However many of these studies were not focused on touch screen phones yet. The first Apple iPhone was announced in 2007, many phones still had a small display and a hardware keyboard at this time.

The most comprehensive researches were made in South Korea and Malaysia, countries with very different culture than Europe, and most likely also with the different technology knowledge. What if the results of the research performed in Czech republic in the year 2014 will be different? It will be very interesting to compare the motivations and needs of people and to design the interface exactly by their actual requirements.

The main ambition of this thesis is to find a **real practical improvement** for seniors. They will be able to use their phone easily without restraints, to stay active in their everyday life and to be in touch with their family and friends.

## 2.2. Characteristics of older people

We will cover the most important characteristics of old people in this chapter. It is the summary of all mentioned resources, especially the [2], [3] and [4]. Other recommendations, how to design an interface for seniors, will be covered in the Chapter 4.1.

### Aging of people

The **age** is not a complete indicator of a specific individual's capabilities. The old athlete can still run a marathon in his age. Some abilities decline with age, others remain stable or increase (for example the knowledge). The variability of capabilities widens as people get older, some people can be more vital and active than others.

Familiarity with **technology** usually decreases with age, as well as the **abilities** of old people. Many elderly need to take on reading glasses or use an hearing aid. These two changes often interacts with each other.

When older adults **avoid technology**, it is for many reasons. Probably they have usability problems with the product (it is too complicated for them), or they just do not trust it (for example to use the internet banking).

The common stereotype is, that they do not want to use modern technology at all. It is not true, they want to keep up with technology, understanding of modern inventions make feel them connected to others and to the world in general. They can learn even a complicated technology. If they see a real benefit for them, they will invest their time and they will learn it [3].

### Vision

**Vision** is the most important way how technology can present information to user. Limitations of a human eye depend on age, old users have mostly worse vision. If we live long enough, nearly all of us will have vision problems. Many seniors need to wear bifocal lenses.

**Older eye** receives only a 1/3 of light than that of people in their 20s [3]. The lens of the eye has slightly yellow color, so the colors of the world appear less blue and more yellow. For old people is harder to distinguish between subtle changes of blue and between shades of a red and purple.

The old eye has also a **slower accommodation** between dark and light places. It cannot quickly change the focus, or react to fast-changing brightness. The **visual sharpness** is also worse, they can not see thin lines or focus on hard edges. It is hard for seniors to distinguish between similar icons, especially when there is very low contrast. We should organize menu items into groups to encourage the visual search.

### Hearing

**Hearing** can help a lot with using the user interface. Old people usually expect that virtual keyboard will have the sound feedback. They want to hear a response whether they pressed the button or not, like in the real world.

**Old ears** can not detect very high and very low pitches. Some old persons with age-related hearing loss require at least 90 dB loud sound for notifications (70 dB is recommended for normal young users). In general, the level of an alarm should be at least 10 dB above the background noise, otherwise they will not hear it. The loudness of the audio should be controlled by the user by the visible and easy to use controls.

#### Cognition

When the interface is poorly designed, many people tend to blame themselves rather then the app. The **effective interface** is the one that helps users in completing their goals with a little confusion and the less errors as possible [2]. It is common that even when an app offer a high contrast and large buttons, it can be hard to understand and to use.

It helps a lot, if the user has already a correct **mental model** of the app. It means that the app is similar to something that user already knew in the past. The mental model guides how he thinks that it should work, and how he think it is working inside. People respond better to things that do exactly what they expect. The great designer should predict and capitalize on what people presume and design the app accordingly.

Modern technology have two problems, a **miniaturisation** (everything is smaller) and a "**function creep**" (development of a multi-function systems - the phone is also a digital assistant and a television is also a computer) [3].

#### Movement

The general rule is that old people require about 50-100% more time for a task than adults under 30. This longer **response times** are mostly due to cognitive changes (the time they need to think about it). It is interesting that once the movement is initiated, their response time not much differ from younger people. The most of this time comes from a decision time, not a movement time. We should allow sufficient times for input.

Many interfaces require very **fine movements**. The consequences of an error are frustrating. If user accidentally removes the file, or he is an inaccurate during the drag and drop of the file, he needs to start over again. Problematic for old people are double-clicks on mouse, and the need of precise rotation with the mouse wheel.

**Movement disorders**, like a Parkinson's disease and Arthritis, are mostly caused by a disease, not as the result of an old age. The Parkinson's disease affects approximately 1% of adults over 65 years. Users have a shakiness in hands and a lack of a motor accuracy. The user interface for Parkinson's patients should be designed very differently, than the interface for normal old people.

#### Adaptable interface

We can design **one application** that will be beneficial for everyone. Usability problems are often shared among age groups. When the usability and user experience is improved for older users, it is also improved for younger adults. However increasing usability for some users **may reduce** the usability for others. For example if we include a very loud audio in the interface, younger people will do not want to use it.

We should design an **adaptable interface**, when user can choose the loudness of the sound, the layout, vibration feedback or selected items in the menu. The adaptable interface will work better than any single design.

## 2.3. Analysis of mobile interfaces

There are three types of phones on the market. Classic simple **feature phones** with hardware 3x4 numeric keyboard. They have usually a small display and small labels on buttons. They do not have advanced features, as a high quality camera, but they can last many days on the battery. These phones are on a decline.

The most commonly sold at this time are **touch-screen phones**. They are usually pretty big, they have a 4 inch large display or bigger. They are controlled by touches with finger, swipes and other gestures. The whole user interface is drawn on the display.

The third category are special **senior phones**. They have usually very large buttons, but only a quite small display. They are very simple, they can mostly only send SMS messages and make calls. Most of them are made from a very durable material. They mostly have a special dedicated SOS button for calling help. They also can not take photos or do any advanced features.

### 2.3.1. Accessibility on touch-screen phones

The most widely used **operating systems** for touch screen devices are Android (currently in version 4.4), iOS 7 and Windows Phone 8, see Fig. 1. Some manufactures have their own variations of these interfaces, for example Samsung has its TouchWiz launcher as a modification of the Android interface.

These interfaces are mostly **not optimized** for old people. They uses small and thin fonts, color combinations with a low contrast and many ambiguous pictograms and icons. Some accessibility features can be set, but they are hard to find in the deep settings screens. These changes are mostly insufficient for very old users. For example some labels are enlarged, but not all fonts in the whole interface.

**Apple iOS 7**
: Apple iOS users can use a VoiceOver screen reader [12], it can read loudly what is written under the finger. Users can dictate the text, but only in some specific languages. The font can be enlarged a little, the bold text can be set. User can zoom in the screen, or invert the colors. iOS also supports external Braille displays. However modern iOS 7 itself still uses a very thin and small font and low contrast colors, the interface is not very simple, the system keyboard has very small keys.

**Android 4.4**
: Android operating system has also some enhancements [13]. Larger font can be set, screen can be magnified by a gesture. It has also an integrated screen reader (TalkBack). However Android interface itself is very complicated. It is very hard to use for old people without any alternative user interface installed.

**Windows Phone 8**
: Windows Phone 8 has a couple of similar accessibility features [14]. The high contrast can be selected, the Narrator can be used as a screen reader (only in selected coun-

tries). Some text can be enlarged, the screen can also be magnified. But overall, these modifications does not help user a lot. Text on live tiles and on some other screens will be still very small. Otherwise this system is pretty simple and straightforward to use, even for beginners.

### 2.3.2. Mobile interfaces designed for seniors

A **launcher** is an alternative user interface that can be installed on Android devices. This interface can be downloaded from a Google Play market (the shop with the applications), or even pre-installed on the phone. The launcher can completely change the way how the interface of the phone look like and how the phone is used.

We will go through a couple of **selected interfaces** targeted for seniors. Then we will present a list of other launchers in an alphabetical order.

#### BIG Launcher (free demo, $10 USD) *http://biglauncher.com*

A complex launcher optimized for seniors and people with vision problems. Configurable main screen consists of six big icons (Fig. 2). Lists of contacts, messages or applications are easily accessible and have a very large font. It is possible to change the theme color or the font size. It has the extended support for TalkBack screen reader

This launcher does not run full-screen, senior can easily jump out of the interface by tapping on a status bar. Camera and gallery icons open standard system apps, instead of some optimized screens. The keyboard for SMS messages is small, the interface uses only a standard qwerty keyboard, user needs to find and install a better replacement. It is necessary to scroll with the content on some screens, but the scrollbar is not shown. Virtual back or home buttons are not drawn on a display, user needs to use their hardware alternatives (that are not backlit on some phones).


**Fig. 2.** BIG Launcher, EqualEyes Accessibility, Phonotto

#### EqualEyes (30 day free trial, $60 USD/2-year licence) *http://equaleyes.com*

Very well made suite of accessible mini-applications, almost every phone feature is present in a touch friendly form (Fig. 2). Items in the menu have great contrast, vibration and voice feedback. The large keyboard is used for writing SMS messages. The interface contains a tutorial and a list of other third-party accessible apps. However the interface looks very pure and serious, contacts do not have photos. Actual date, time and a battery value is not easily accessible from the main menu.

#### Protege Launcher SOS (free or $2.76 USD) *id=pt.protegeipl.launcher*

Simple and colorful touch interface for people with less experience (Fig. 3). Some operations are quite complicated, three clicks are needed to display a dial pad. Color contrast of some icons is very poor (white icon on a yellow background), icons also do not have text labels. Back button returns to home screen, instead to previous one.

#### Seniors Phone by Mobili (free) *http://www.seniorsphone.mobi*

Interesting interface for people with bad eyesight (Fig. 3). Screens for calling and sending SMS messages are simplified. Text can be inserted on a large keyboard, or from a couple of predefined sentences. Buttons have distinguishable colors. Other optimized apps can be downloaded (calculator, a torch light). However the advanced setup is quite complex, it is difficult to pin the favorite app or get the list of all apps.

#### Wiser - Simple Launcher by UIU LTD. (free) *http://www.wiser-me.com*

Very fresh and friendly interface (Fig. 3). Lists of the contacts and last calls are simplified, favorite people and applications can be pinned on the screen. However the messaging icon still points to a standard system app, as well as the camera and a gallery icon. The keyboard is also only standard system small, not any optimized.

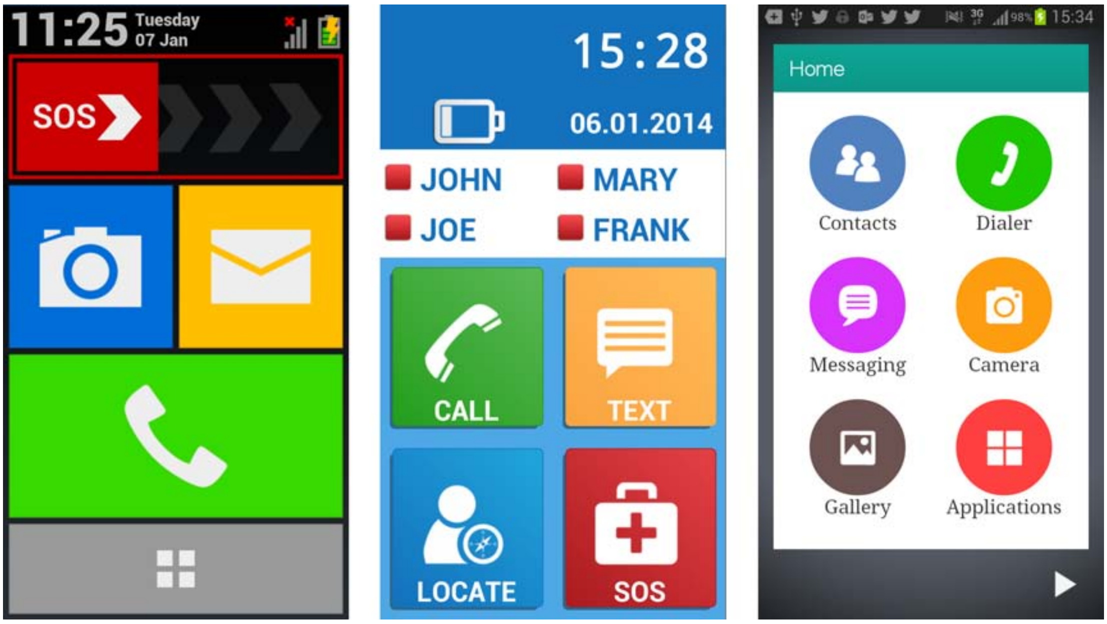


**Fig. 3.** Protege Launcher SOS, Seniors Phone by Mobili, Wiser - Simple Launcher

### 2.3.3. Other launchers for seniors

There are also other launchers for seniors on the market, these are the examples:

**Big buttons (free or $0.99 USD)** *id=com.dialerdev.launcher*

**Easy Launcher - For low vision by Momemi ($0.99 USD)** *id=com.mylauncher*

**EasyCall Launcher (free or $0.99 USD)** *id=com.ElisherArts.EasyCallLauncher*

**Fontrillo - The easy launcher (free demo)** *http://www.fontrillo.com*

**Low Vision by Mobyi Apps (free demo)** *id=com.mobyi.lowvision*

**Phonotto Simple Phone Seniors (free demo, $2.69 USD)** *http://www.phonotto.com*

**Seniors, Elderly and Kids phone (free or $2.99 USD)** *http://www.wise-phone.com/*

**SENIOR Launcher (free or $19 USD)** *id=com.jeejen.family.en*

### 2.3.4. Special phones for seniors

Some phones have already pre-installed a special interface for senior users. Usually we cannot download this interface to our phone, we need to buy the specific brand or type of the phone.

**Fujitsu Stylistic S01** *http://engadget.com/products/fujitsu/stylistic/s01*

This phone has both software and hardware specially designed for seniors. Its display can recognize two levels of a pressure. When user puts the finger on a display, the item is selected. And just only after he presses more, the action is performed.

**GreatCall UI (JitterbugTouch 2)** *http://www.greatcall.com*

**PhoneEasy 740 (Doro)** *http://www.doro.co.uk/experience/products/smartphone*

# 3. Analysis and design

Characteristics of old people were covered in the previous chapter. Existing mobile user interfaces for seniors were described. The ambition of this work is to create a new **mobile interface** optimized for the most common tasks of elderly users. This interface will help them with their everyday problems.

Do they need to take on reading glasses every time they use their phone? Maybe the buttons and labels on the display should be bigger. How often do they use their phone and which features are the most important for them? What do they like and dislike?

There are many questions to consider when developing a new user interface. It is possible to get inspiration from available resources and books, as it was referred in the Chapter 2. However the only way how to verify this information; how to really understand their needs, and even how to discover new findings, is to perform an own qualitative research.

## 3.1. Qualitative research

The main aim of the **qualitative research** is to get an insight into the behavior of people. We want to get interesting new ideas from elderly. If we did not talk with our real consumers, the app will never be optimized for them. We want to understand how they think, what they want and what they really need.

### 3.1.1. Participants and the test setup

Our aim was to meet a group of **different people** with different requirements to observe the most interesting facts. We wanted to meet several people older than 55 years that actually own and use a mobile phone, at least for calling or receiving calls. Approximately one half of people should be older than 70 years.

We were interested in both **males and females**. We also wanted to meet approximately a half of less experienced users and a half of more experienced elderly. They were distinguished by their ability to read/write SMS messages.

We have interviewed **5 old people** in this part of the research. People were interviewed in a semi-structured discussion. Each talk took approximately 60-90 minutes, it was performed at their home. Their age varied from 56 to 76 years (mean age = 67), all were living in a Czech republic. Another few of elderly were also interviewed in the later part of the research (in less formal way).

#### Participants for the qualitative research

**P1:** woman, 74 years, uses a classic senior phone, only for calls
**P2:** woman, 56 years, active person, writes SMS messages
**P3:** man, 59 years, uses phone very actively, has poor eyesight
**P4:** woman, 72 years, pensioner, writes SMS messages
**P5:** man, 76 years, very basic user, only receives calls

### 3.1.2. Screener

People for the qualitative research were selected by the simple screener:

**Age group:**

| | |
|---|---|
| less than 54 years | |
| 55-69 years | (target: 50% of people) |
| more than 70 | (target: 50%) |

**Gender:**

| | |
|---|---|
| male, female | (target: aproximately 50% men, 50% women) |

**Do you have a mobile phone?**

| | |
|---|---|
| yes, no | (target: must be yes) |

**Do you use it for calling or receiving calls?**

| | |
|---|---|
| yes, no | (target: must be yes) |

**Do you write SMS messages from your phone?**

| | |
|---|---|
| yes | (target: 50% = able to write a text on a keyboard) |
| no, only read | (target: 25% = passive user, but can read messages) |
| do not use SMS | (target: 25% = very basic user) |

### 3.1.3. Session guide

Each interview started with a **pre-interview** part. It was important to establish a warm and trustful atmosphere. The session continued in a **semi-structured dialog**. We discussed several topics about mobile usage, accessibility and everyday complications with their mobile phone. It was important to talk about their real experiences with their mobile phones, not what they "think that other seniors do".

After the session the **post-interview** part followed. Participant was thanked for his willingness and assured that all data obtained during the session will be used only for the purpose of this study.

The list of all questions (full session guide) is included in the **Appendix A**. Main topics are summarized here:

**General questions**
The each interview started with general questions where participant lives, how often does he communicate with other people and what are his usual habits. The main aim of this part was to verify information obtained from the screener and open a dialog with the participant.

**Mobile usage**
The next topic was dedicated to everyday usage of the phone. Where does he store his numbers, how does he use call history or how does he handle missed calls.

**Accesibility**
We were interested if he has any problems with a small keyboard, tiny text on the display... And also what does he think about the shape and size of his phone.

**SMS messages**
The next topic was dedicated to SMS messages. What is his opinion on writing text on the phone. What complications does he have with the keyboard. How does he recognize a new unread message?

**Advanced features**
We talked about other possibly handy features. If he uses some reminder or an alarm clock on his phone, or if does he use a flashlight. What features does he miss now and would like to have on his next phone.

**Complications**
We talked about problems what does he make him upset on his phone. Is the menu complicated? Are there any features that he wants to use, but he is not using them?

**Health care**
The next topic was dedicated to health care and SOS functions. How does he think his phone can save his life? How would he call for a help if he need some? Is it important for him?

**Touch screen phones**
The last topic was about touch screen phones. What does he think about them. Does he already tried one? Can he imagine he would use one - and if not - what kind of things he would be afraid of? Why does he think a touch screen phone would not be suitable for him? We have also discussed the other useful abilities of smart phones, like a large display and a camera. Each talk ended with a motivation discussion, what is possible on modern phones today and what could be the advantage for him (for example: a video call with a family, a TV remote or a voice recorder).

### 3.1.4. Analysis of the obtained data

These talks gave us a lot of ideas. Here are the most interesting findings from the qualitative research. Hypotheses will be formulated in the next chapter.

#### Phone usage

Seniors are maintaining contact with only a very few people. They have 5 to 10 close friends to whom they call regularly. The older they are, the fewer friends they have. It seems that very old people talk mostly only with their family relatives, or their doctors and similar persons.

Practically everyone uses a **contact list** in their phone. They have names and numbers saved there, they know how to display the name and call it. These contacts were mostly saved to their phone at the beginning, by younger relatives from their family (a son or grandson), or they saved them there by themselves. However it seems that they all use a contact list mostly for reading, they do not save new names into phone very often.

Some people have a couple of contacts written on a paper (not saved in a phone), they call them by dialing their number on a keypad.

#### Abilities of elderly

Old people are usually not very experienced users. They **can handle** to learn a specific task if they have a motivation for it. For example they might learn how to find a specific contact in a phone, or the last missed call. One participant mentioned that he needs to remove messages from time to time, because his phone has a limited memory. He would not receive any other messages if he did not do it.

They need to **learn the task** slowly, and try it several times. Mostly someone younger is teaching them how to do it. If they want to learn the task by themselves, they want to look into a paper manual with detailed instructions. It does not make sense for them to do complicated tasks "just to try", or to explore other advanced features of their phone. They are really fine with the tasks they already can do and they do not have any motivation to try anything other else.

Very **old people** even do not add any new contacts to their contact list, nor write SMS messages. They really use their phone only for receiving calls and occasionally making calls. They can read a received SMS message, but they are not writing any. They do not expect anything more from their phone. Sometimes they use a flashlight or a radio for listening music, but not any other features.

However it is seen that they do not want to fall behind other people. They **want to keep up** with the modern technology. Mostly they do not feel old yet. They are trying to do the task, even if they know that they are slower. They are glad that they can stay in touch with family and friends, thanks to their phone.

#### Advanced features

Almost all interviewed participants generalized that all **advanced features** in phones are only for younger people: "We are old and we do not need too much features." They said that a phone for old people should be simple, only for calls and messages, without any extra functions. All advanced features should be hidden.

However as the discussion continues, they start to talk about themselves. They would welcome some neat features. It would be really nice for them to have a voice recorder, a camera, calculator or an alarm clock. Another given example was a notepad, or a call recorder. But they would like to have only this one or two selected features, not the others. And each people had it's own favorite features.

It is interesting, for example, that one participant said he certainly does not need a Calendar in his phone. However he revealed later that he would like to be reminded when his friend has birthday or when he has a meeting. It was identified that he does not really understand how could Calendar work in a phone. This function should be named better and designed for them in more simple way. Another participant did not understand what a Call history or List of calls mean.

It seems that older people are sometimes **confused** by the names of items in their phone. Especially when they have more complicated phones. There are many items and similar icons. Some features have very short and not very understandable names.

These people **can not imagine** what these labels mean and therefore they are confused. It is the reason why they are not using these features. Problematic items were for example: Protocols, Organizer, Conversations, WAP or Applications.

### Accessibility and health care

Most of the participants were using classic **feature phones**. They said that their phones are quite small. It is obvious for them that they need to take on prescription glasses every time they want to use the phone. However if they think about it, it is quite annoying for them. They would welcome larger buttons and a more visible text on a display.

They have the number of a doctor or firefighters saved in their phone. But they do not use any speed dial "SOS button" on the phone that is dedicated only for calling help. They did not need to call an emergency assistance yet. But they agree that such a button could be useful for them.

### SMS messages

Some participants write **SMS messages**, even if they do not see well on a keyboard. They already have learned which correct button to push. Although for many people, writing text messages is still very complicated and difficult. They need to press one button many times to write one letter. This is confusing for them and it is one of the reasons they do not write SMS messages. Also some participants said that they just do not have motivation for writing text messages. They would rather talk with another person by voice call.

They want **to be notified** very loudly and visibly on a new SMS or a missed call. The phone should display a large icon of a new message, even if it is turned off. They would welcome some notification light diode. Their phone should notify them with sound many times, as long as they do not read the message.

However some people **did not want** this. Notifications of new messages and missed calls should be very silent for them and non obtrusive. This will be further explored in this research.

### Touch screen phones

They mostly do not know what to think about **touch screen phones**. They think that these phones are mostly only for young people and for playing games. They can not imagine which features could be beneficial for them. They are happy with their actual phone and they do not want to change it. They would need to see and try the phone if there is anything good for them.

However when I shown them what a touch screen phone can do, they were **positively surprised**. They welcomed a large display with big text labels. They would even not need to take on glasses, if the whole interface had large fonts. They liked a camera and viewing photos on the display. The large size of the phone was not a problem, they appreciated it.

It was discovered that touch screen phones aimed for seniors should probably have some rubber bumper or a durable case. They were afraid that it can fall down to ground and broke. The phone without the case was also quite thin for them. They also pressed hardware volume buttons by mistake and were not aware of any mistake.

White text on the black background was more readable for them than black letters on white background. They were not able to focus well on large illuminated display. However it is possible that this will be different on a direct sunlight. All our discussions took place inside a building.

#### Complications

We have also talked about their other **complications** with the phone. Most of the ideas were not directly applicable to our research, it only distracted the discussion from the main topics.

They do not like expensive calls on extra paid numbers. They need to recharge their phone very often. They dislike when they forgot their phone at home. There should be a simple way how to turn off all sounds and turn them on back. One participant was confused by a regular automatic lock of the buttons on the phone.

### 3.1.5. Hypotheses for verification

A couple of interesting ideas came from the qualitative research. We have interviewed 5 different participants. To prove these ideas, we formulated a set of hypotheses:

**H1: Phone usage**
Old people are maintaining contact with only a very few people. Making and receiving calls is the most important task for them. They have contacts saved in a phone and they really use the contact list.

**H2: SMS messages**
Some people write SMS messages. When they do not, it is because they dislike a small keyboard, or it is too complicated for them.

**H3: Accesibility**
Most of the elderly wear prescription glasses. Their phones are small, with a small keyboard. They would welcome larger buttons and bigger text on a display.

**H4: Features**
Seniors use only the most basic features on the phone. They often cannot recognize a feature by the name, so they do not use it. They want to see only selected features they use, other items should be hidden.

**H5: SOS button**
They would like to have one button dedicated only to call help or a family member.

**H6: Notifications**
Phone should notify them very visibly on a new SMS or a missed call.

## 3.2. Quantitative research

We prepared a **questionnaire**. The aim was to prove or deny selected ideas and to gain another information about the target group.

We have received **118 answers** from people older than 50 years (67 of them were older than 60 years). The questionnaire was distributed over the internet to parents and grandparents of our friends, their co-workers and family relatives. It was promoted in the article on SmartMania.cz, on selected Google+, Twitter and Facebook accounts.

Shorter version of the questionnaire was also printed on a paper and distributed to seniors personally. We have met and talked with seven other seniors (living at home, or in the retirement home). It gave us additional insight into their needs and spot between them some differences.

It was quite difficult to get a reasonable amount of answers from old people. Many seniors are not active on the internet or do not use a computer. I would like to thank again to all people who have helped me to get these information.

We wanted to aim the questionnaire only to people older than 60 years. It turned out that also many people from 50 to 59 years were interested in the new simplified mobile interface and they also have problems with their eyes and vision. We were able to compare how different are needs of people younger than 60 years and older.

### 3.2.1. Content of the questionnaire

The full questionnaire is included in the **Appendix B**. It was aimed for people older than 50 years who are using a mobile phone. These were the main topics:

**Phone usage**
How often do they use their phone, how many people do they call regularly and whether they use a contact list in their phone. We also asked who saved them numbers to their phone and how regularly do they add new contacts.

**SMS messages and usability**
Another questions were focused at writing and sending SMS messages. Why they do not write text messages on their phone or what do they dislike on it. Do they see well what is shown on the display? What does having a phone mean for them personally?

**Features of the phone**
The next section was focused on applications. Which particular features do they use on their phone. A selection of names were presented. They were asked which of them do make sense for them (so they can imagine what the particular button will do). Examples of these synonyms were: protocols, calls or call register; messages, conversations and SMS; and features, programs and applications. We wanted to find the most understandable selection of names for our interface.

**A set of statements**
We have also formulated 12 statements based on our hypotheses, previous research papers and other ideas. Selected questions were based on [2]. We wanted to get an opinion on these ideas, if they are also applicable to our target group in Czech republic, in the year 2014. It was possible that widespread of smart phones in previous years and generally better acceptance of technology will slightly change the results.

#### Segmentation questions

The last section was used to split people to particular groups, so we could compare them by their age, gender, way of living and a computer knowledge.

### 3.2.2. Analysis

We got **118** answers from people older than 50 years.

- 50 people from 50 to 59 years (42%)
- 35 people from 60 to 69 years (29%)
- 31 people from 70 to 79 years (27%)
- 1 person older than 80 years and 1 person older than 90 years

Approximately half of them were women (54%), the rest were men. It turned out that some characteristics did not depend significantly on age:

- Nobody uses their phone only for receiving calls. Everyone is making call sometimes, 84% says that they are calling and receiving calls regularly.
- Everyone uses a contact list in their phone. For 81% it is the primary way how to call a contact, so the hypothesis **H1 was confirmed**.
- Rest of the people use a contact list and keypad dialer similarly often, only for 2% of people is the dialer the most used method.
- Only 7% of people did not add any new contact since they have bought the phone. Others are adding new contacts sometimes.
- 74% of people need to wear prescription glasses. It seems that this also does not significantly depend on age. The percentage of 50-59 years old people that wear glasses was almost the same as the percentage of 70-79 years old.
- Only 7 people (6%) were using a hearing aid. Only 4 people said they have fine-motor problems with their fingers and only 3 people were (partially) color-blind.

Full Excel sheet with all results is included on the attached CD.

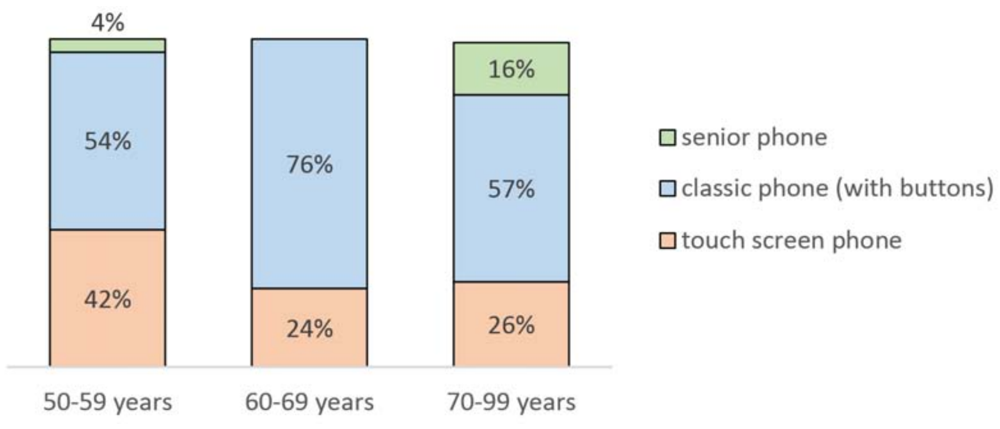


**Fig. 4.** Types of phones old people use

### 3.2.3. Segmentation by age

We have found that other parameters will most likely depend on age. We have split people into three groups by their age:

| | 50-59 years | 60-69 years | 70-99 years |
|---|---|---|---|
| does use a computer (with internet) | 95% | 76% | 78% |
| has internet in a phone | 46% | 18% | 27% |
| writes SMS messages | 98% | 86% | 76% |
| changed a background image / ringtone | 67% | 43% | 24% |
| saved contacts to phone by yourself | 90% | 70% | 56% |
| adds new contacts regularly | 46% | 37% | 16% |

It seems that people between 50-59 years are still very active. Almost everyone uses a computer with an internet connection and almost everyone write SMS messages. Even 46% of 50-59 years old have internet connection in the phone. 67% of them have customized their phone by changing a background image or a ringtone.

When they are older, their participation with technology decreases. Only 56% of people older than 70 years have saved contacts to their phone by themselves (for others the contacts were saved by their family members). Only 24% of them have ever changed a background image or a ringtone in their phone.

But they are still very active in other activities, about 80% of elderly still use SMS messages, at least for reading. Approximately 75% of these people have access to the computer with the internet.

It is interesting that 42% of 50-59 years old people already have a phone with touch screen (Fig. 4). Even approximately 25% of older people use smartphone. However classic feature phones are still used the most between 60-69 years old (76%). Special phones for seniors are used mostly only by people older than 70 years (16%).

| | 50-59 yrs | 60-69 yrs | 70-99 yrs |
|---|---|---|---|
| I do not see well what is on a display | 39% | 58% | 52% |
| I would appreciate a larger font (text size) | 44% | 70% | 32% |
| phone should read loudly what's on a display | 11% | 30% | 48% |
| phone should call help on 1 button press | 18% | 50% | 56% |
| advanced features should be hidden | 14% | 23% | 36% |
| phone should flash/make sounds on new message | 23% | 47% | 54% |

About a half of people do not see well what is shown on a display (it means they selected in a questionnaire that they need to take on prescription glasses to use the phone sometimes, or they see very poorly what is on a display and it is restricting them). This is being a little worse when they are getting older. Hypothesis **H3 was confirmed** in practically all parts.

### 3.2.4. Requirements of older people

Very interesting is the finding who would appreciate a larger font size on his phone. Approximately 40% of 50-59 years old and older than 70 years said they would welcome it. However between the middle 60-69 years group the demand was much greater. 70% of these people wanted to see a larger text. It is most likely because this group is still using classic feature phones with very small display.

Younger people use bigger smartphones, and older people have special senior phones with larger items. Another possible explanation is that many old people had an eye surgery recently, so the people older than 70 years see better than people between 60 and 69 years. Maybe this question was just hard to understand for old people.

In the older group was also much greater demand for some SOS button, which could call them a help or family members on one button press (56%). For younger people this is not so important, only 18% of 50-59 years old people would appreciate that. Hypotheses **H5 was confirmed**, but only for the most old group of people. We will define two different personas in a chapter 3.2.6 based on these findings.

The interesting contradiction was also on an idea that a phone should very visibly flash and make lots of sounds on a new received message. More than 50% of old people would like that, contrary to only 23% of 50-59 years old. For these younger people the phone should be rather unobtrusive and stealthy. The hypotheses **H6 was also confirmed**, but only for the oldest group of people.

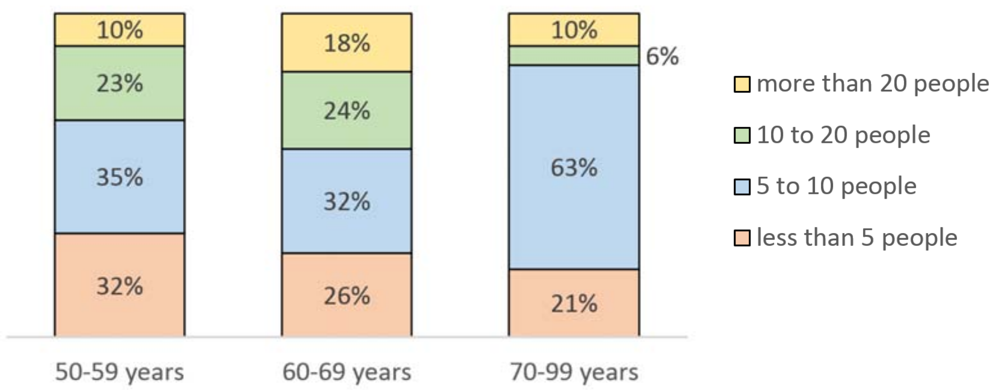


**Fig. 5.** How many contacts do old people call typically

Most of the old people think that their phone contains more features that they could ever utilize. Mostly only elderly with simple senior phones think, that they are using all features of their phone (and it does not even have more features). Approximately half of the people can not understand some names of features in their phone. However only 28% of people think that advanced features should be hidden (36% of people 70-99).

It seems that accidental deletion of items is not a big problem, but 33% of people thinking the opposite is still pretty much. There should be some option to disable removing of messages and contacts from the phone.

**Selected statements (average percentage of people older than 60 years)**

74% My phone contains more features that I could utilize.
53% I can't understand some names of items in my phone.
29% Advanced features should be hidden in my phone, they only confuse me.
33% Sometimes I am afraid I will delete something in my phone by mistake.

53% When I receive a message, my phone should be flashing and making a lot sounds.
58% Sometimes I need to look who called me lately (for example a previous night).
35% I have changed an image on my display, or changed my ring tone.
33% I would like to have a camera or voice recorder in my next phone.

22% I have already used my phone for calling help (and it helped me).
25% My phone should be able to read loudly what is written on a display.
47% I would appreciate a larger font (text size) on my phone.
51% Phone should be able to call help on 1 button press (or contact family members).

What was very surprising that only 21% of old people selected that they already used their phone for calling help (and the phone helped them). In the research [2] they found a much higher representation, almost 70% (of people older than 50 years). Probably it will have something in common with Czech independence, our usual traditions or national health care system.

### 3.2.5. Phone features

The most used features on the phone were of course calling people (100%) and writing SMS messages (87%), see Fig 4. Another favorite features were camera and an alarm clock or reminder. The main advantage of having a mobile phone for people is the ability that they can connect very easily with their family and friends (67%), see Fig. 6.

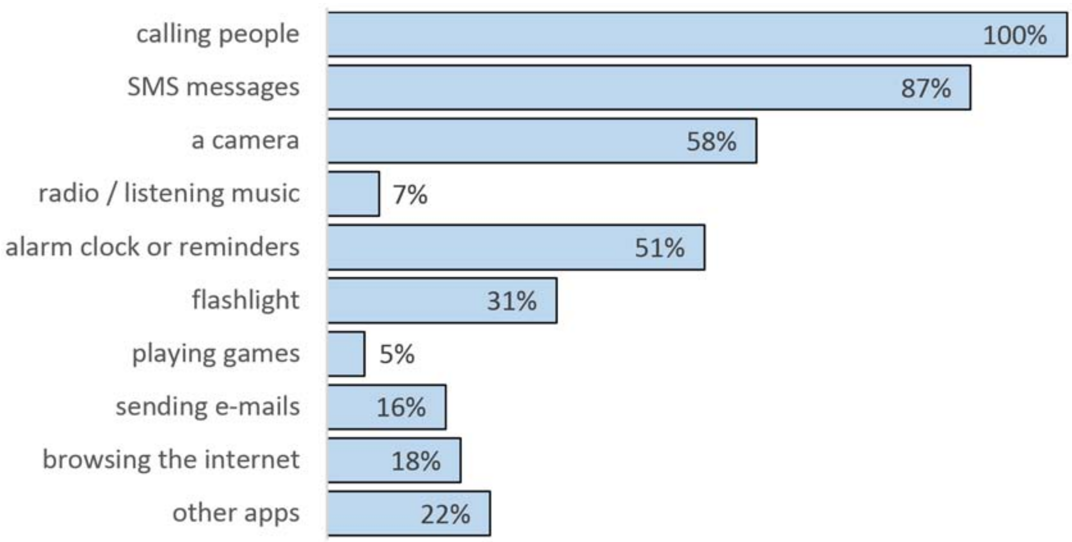


**Fig. 6.** What features do old people (older than 60 years) use on their phone

Seniors really use only the most basic features on their phone. However only 29% of people older than 60 years think that advanced features should be hidden. **Only one part** of hypothesis **H4 was confirmed**. The main disadvantage on writing SMS messages for them is a small keyboard, see Fig. 8, so the hypothesis **H2 was also confirmed**.

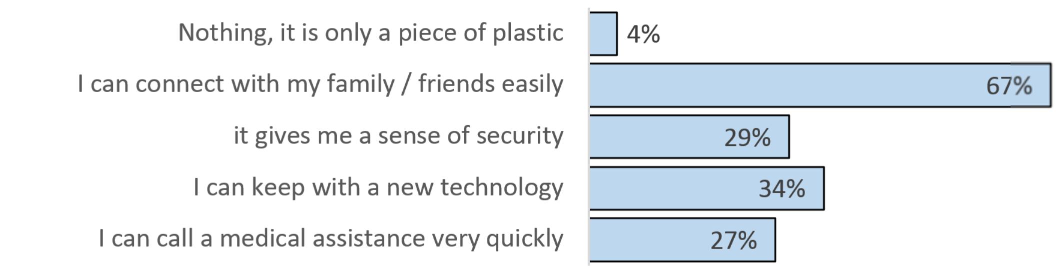


**Fig. 7.** What does a phone mean for old people (older than 60 years)

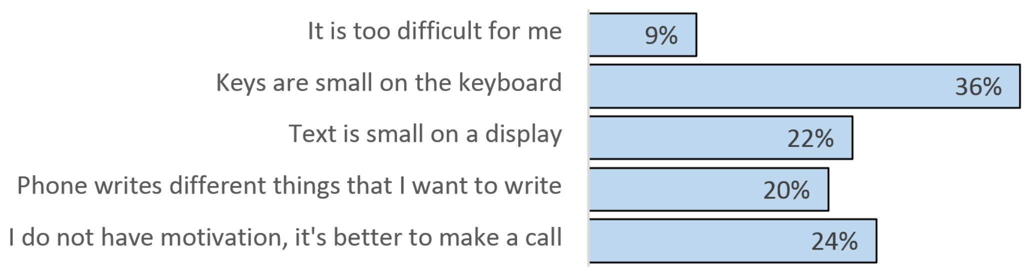


**Fig. 8.** What old (older than 60 years) people dislike on writing SMS

### Names of features

When we compared different names of features (whether they understand what these items mean), these were the results from people older than 60 years:

- 96% calls, 91% call agenda, 33% protocols
- 96% SMS, 93% messages, 44% conversations
- 45% applications, 67% programs

It mean that both names **call** and **call agenda** are perfectly good. **Protocols** (used on some Samsung devices) is not good and meaningful name. It is similar with **SMS** and **messages**, both are making sense. **Conversations** is a worse option.

Both **applications** and **programs** names are not very meaningful for some people. The second one was a little better. Our discussions revealed that many seniors did not understand concept of applications. They did not know that their phone can be extended of new features. They bought a phone "as it is", they had no idea why there is a "store" item in the phone.

### 3.2.6. Personas

Abilities of people were dependent on their age, at least partially (as was noted in the previous chapter), as well as their amount of experience and participation with the technology.

However we were able to identify a couple of other patterns from our discussions. When we looked at the results of the questionnaire, some people preferred specific answers in contrast to other people, who preferred the opposite.

We were able to identify two main personas: **Marie** and **Honza**. They both want to have a phone with easy to use and simplified interface, with bigger items and large visible text. They both write SMS messages, or at least they can read them. They both have numbers of people saved in a phone. And they both have problems with eyes & vision and need to wear prescription glasses sometimes.

But there are some differences:

**Marie, 74 years** is an elderly grandmother and pensioner. Once a month her grandchildren come to her home. They bought her that phone and set it up. Marie has only a few contacts in her phone and still the same names. She is just using that phone. She knows how to start a flashlight or set an alarm. But she does not have a motivation to try another features. She wants to be notified very loudly on a new SMS. She would welcome an SOS button for calling help. She is afraid she could remove something by mistake from her phone. She wants to have a better contrast and bold, very visible font. She prefers a classical multitap ABC keyboard.

**Honza, 62 years** is an active old person, few years before his retirement. He is still using a computer to browse the internet and send e-mails. He is in touch with friends with the same hobby every week. He is using his phone quite actively, once in a while he adds a new contact to his contact list. He is using Skype and sometimes even read e-mails on his phone. He want to personalize his phone, set an image on a background or change the ring tone. His phone should be rather discrete, notifications on SMS messages should not be very loud. He knows qwerty keyboard from a computer and other new phones, he will prefer it (instead of old ABC multitap keyboard).

### 3.2.7. Summary of the analysis

We have performed an own qualitative and quantitative research. We were able to identify needs of old people and we have compared them by their age, abilities and motivations.

In the next chapter we will summarize how exactly the mobile interface should be designed for these people. We will design a couple of paper mockups and create a working prototype.

# 4. Implementation

We have interviewed a couple of old people and performed a quantitative research to verify the ideas. We have gathered information how they use their mobile phones. These findings will be applied to practice now.

How should an optimized mobile interface for seniors look like? What should be the desired user experience? What about the visual design? We will create several paper mockups and then prepare a working prototype. The architecture of the prototype and the implementation details will be covered in the last part of this chapter.

## 4.1. Design principles

The **user interface** for old people should be very simple at the first look and understandable. It should not overwhelm user with lots of information. The interface should look reliable and trustworthy.

### 4.1.1. Navigation

It should be understandable how to navigate between individual screens. Which screen is the home screen and how user can return back to this screen. Similarly, which virtual or hardware buttons will get him back to the previous screen (if we allow it).

The **navigation model** should be consistent in the whole interface. There should be a standardized way for navigating between screens. Ideally all navigation buttons should be drawn on a display, on each screen and on the same specific position. We should not rely on specific hardware buttons in our interface, because each phone is different and some phones even do not have hardware buttons.

Each screen should be very **distinguishable** from others. We could place there some label with the name of the screen (Contacts, Messages). The user must be able to clearly identify which screen is displayed and what he can do with it. It means that the user interface should be consistent and standardized; however any two screens should not look similarly, otherwise people will be confused.

The **structure** of the screens should be very shallow. We should be able to reach every each function of the interface on maximum two or three taps with finger. More complicated structure will be hard to understand for elderly users, they will be lost and confused.

### 4.1.2. Accessibility

All labels and texts in the interface should be presented with a large font. All titles and labels of the buttons should be big and easy to read. The font should be well readable.

Modern thin sans-serif fonts look great, but old users do not prefer them, as they can not read them. The font in our interface should be in a classic style and preferable with a serif typeface. The font with bolder weight or a bigger size would be also better.

A very **high contrast** in the whole interface is very important. All items should be clearly visible. Any complicated images should not be on the background. Black text on a white background (or a white text on a black background) would be the most appropriate for the design.

White interface will also look better on a direct sunlight, compared to the black background that will be better in the normal indoor conditions. As we have discovered, it is possible that old people will preferably select a black interface more often, because it will not shine on them with a light from the display.

### 4.1.3. Text labels and icons

Only limited number of **items** should be located in the menu, 6 to 8 items is enough. Longer lists of items should be split into groups, for example the list of messages, missed calls or contacts. We can split items to categories by date (received today, yesterday, this week...), or by type (received messages, sent).

Every icon in the interface should have a text label. The **pictograms** itself usually have a little meaning to many users. People do not have experience what these icons mean. Especially old users want to be really confident what action it will do, before they click on the button. Meaningful text labels will help them a lot.

**Text labels** should also be short and easily understandable. Every label should be written in a human language, for example "today" or "yesterday" would be certainly better than just a numeric date. Each more complicated screen should have a descriptive label with meaningful instructions (for example: *write a name of the contact here*).

Also names of **system items** should be meaningful and understandable, for example *call agenda* will be better than *protocols*, or *SMS messages* will be better than *conversations* (as it was discussed in the Chapter 3.2.5). We should also encourage the visual orientation of people. Contacts should have photos and items different colors.

### 4.1.4. Buttons and controls

We should use only simple **1-tap** actions on all buttons and items in the whole interface. Long press on the button or double-tap will be less understandable and not intuitive. We should avoid any non-intuitive gestures like the swipes from the side, pinch to zoom, panning and zooming with two fingers... Each action should be simple and obvious. We should not use any slide-out menus and hidden items.

Each **button** should have only one function. Similar buttons with different functions should have a different color and another icon and text. Also the darkness of buttons should be different, to facilitate needs of color-blind users.

Buttons should have a visual response whether they are pressed. They should change their color and make a sound when they are activated, and also make a short vibration feedback. Buttons should also react to taps even if the user holds the phone in his other hand and mistakenly presses the display with his palm.

## 4.2. Paper mockups

We have sketched three different ideas on a paper. Each interface has some characteristics. We will discuss their pros and cons, then select the best ideas from each interface and design one prototype.

### Mockup 1

The **main screen** has a keypad to dial numbers (Fig. 9). It is oriented in a wide 3x4 grid, to save vertical space of the display. There are three **configurable icons** above the keypad. Fourth button opens a menu where other features are hidden. These functional buttons can have a variable length, for example when you dial a number, you will see only two buttons there: *start call* and *end call*.

Each screen has a large **title** at the top. The back button is located at the right top corner on each screen. Content of the screen does not scroll. At the bottom of each screen is the **next button**. It will move user to the next page of the content (or open the next message, or the next contact). At the bottom could be also **secondary button**, to answer the message or perform another action.

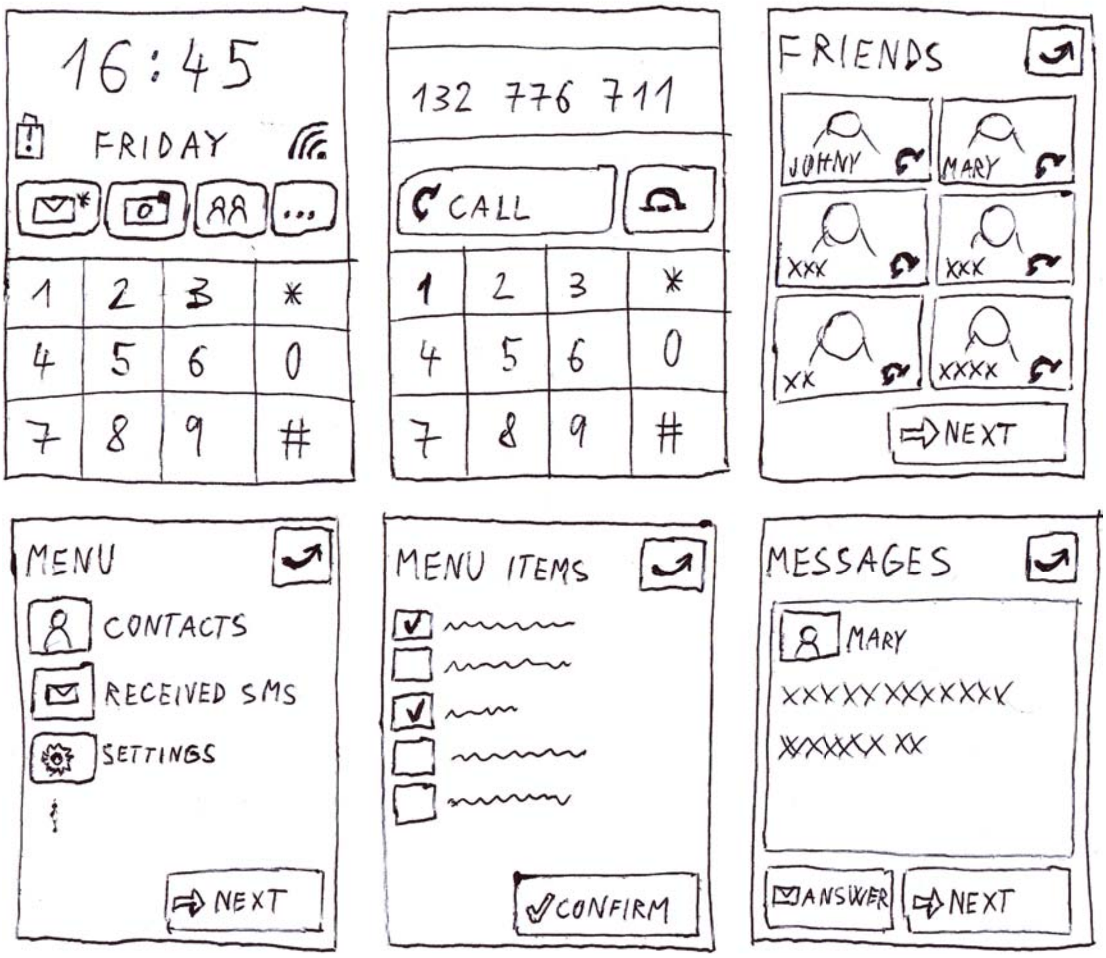


**Fig. 9.** Mockup 1

### Mockup 2

Second mockup does not have any specific menu sub-screen (Fig. 10). All features are accessible from the **main screen**. The grid consists of four fixed buttons (phone, contacts, messages and applications) and five configurable buttons (favorite contacts or applications can be pinned there).

Each sub-screen works like a "hub" that brings together all similar functions. On the **phone** screen you can dial a number, or see the list of previous calls. On the **contacts** screen you can find your favorite contacts, and also switch that to see all contacts.

As well as the **messages** screen, where you can see the list of message threads. If you want to write a new message, you can just click on one button. Message threads are sorted by names of contacts, not split to *received* and *sent* messages.

**Buttons** to secondary features are stacked on the top of each screen, below the top panel. On the **top panel** is shown a name of the screen. On the left side is a back button, on the right side is located the actual time and percentage of the battery.

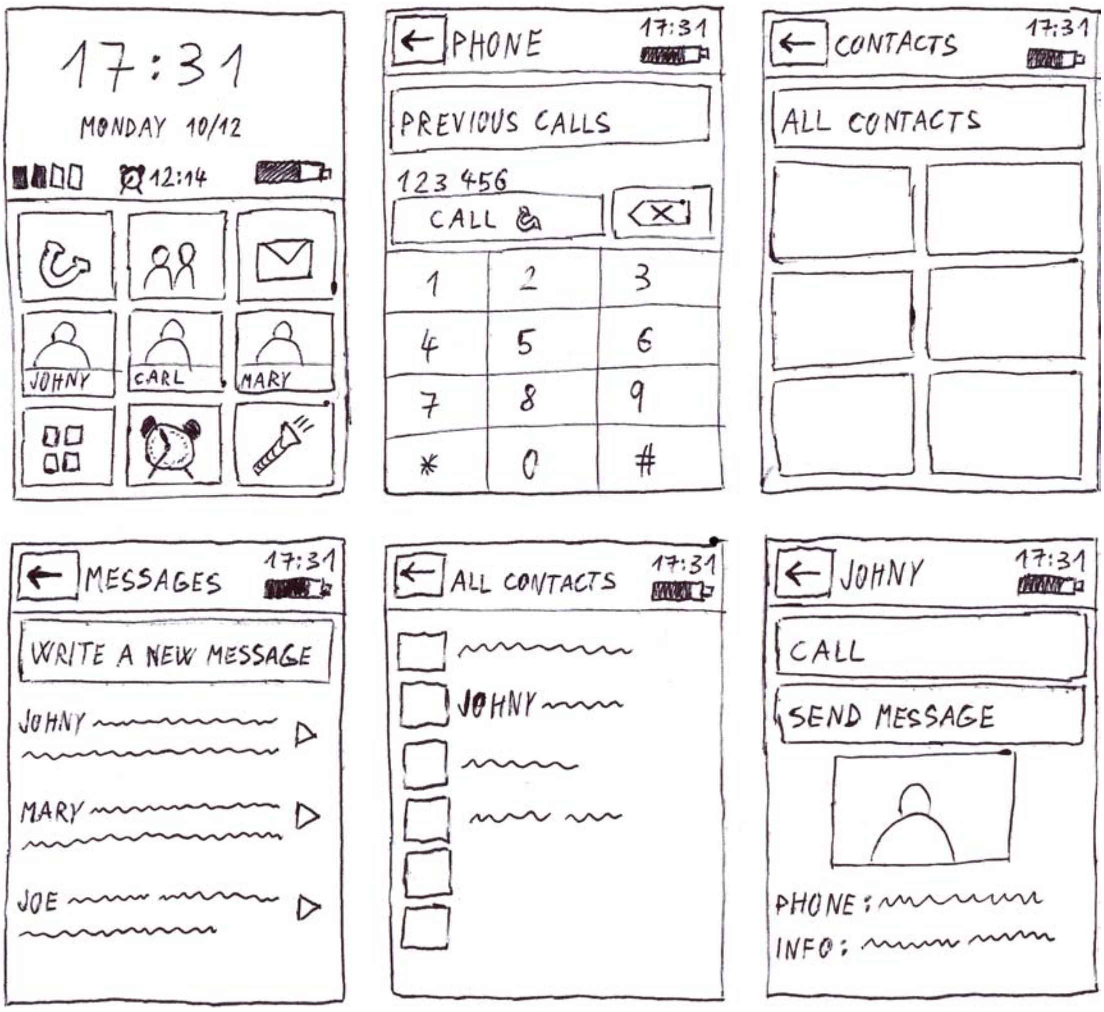


**Fig. 10.** Mockup 2

### Mockup 3

The third mockup gets inspiration from classic feature phones (Fig. 11). The **main screen** consists of a large keypad. Green and red call buttons are located above the keypad, the middle button opens a menu.

This **navigation panel** is used for a consistent navigation across the whole interface. It is displayed on every each screen. Green phone button starts a call, or opens a favorite contacts screen. Red button ends the call, or returns user back to main screen. The middle button is used as a back button (except on the main screen).

The upper part of the interface (above the keypad) essentially corresponds to an old display of a feature phone. When user press a *menu* button, the **keypad** can animate down and expand the view on the display. Similarly when he press a red button, it can animate back to a compact mode. This simple animation can get a clear context to user what is exactly happening and how the interface has changed.

Every screen has a **title on the top** with the name of the screen. There are also some other buttons, for adding a new contact, searching or similar action. More important buttons can be placed at the bottom on the screen (like a *call*, or *send a message*).

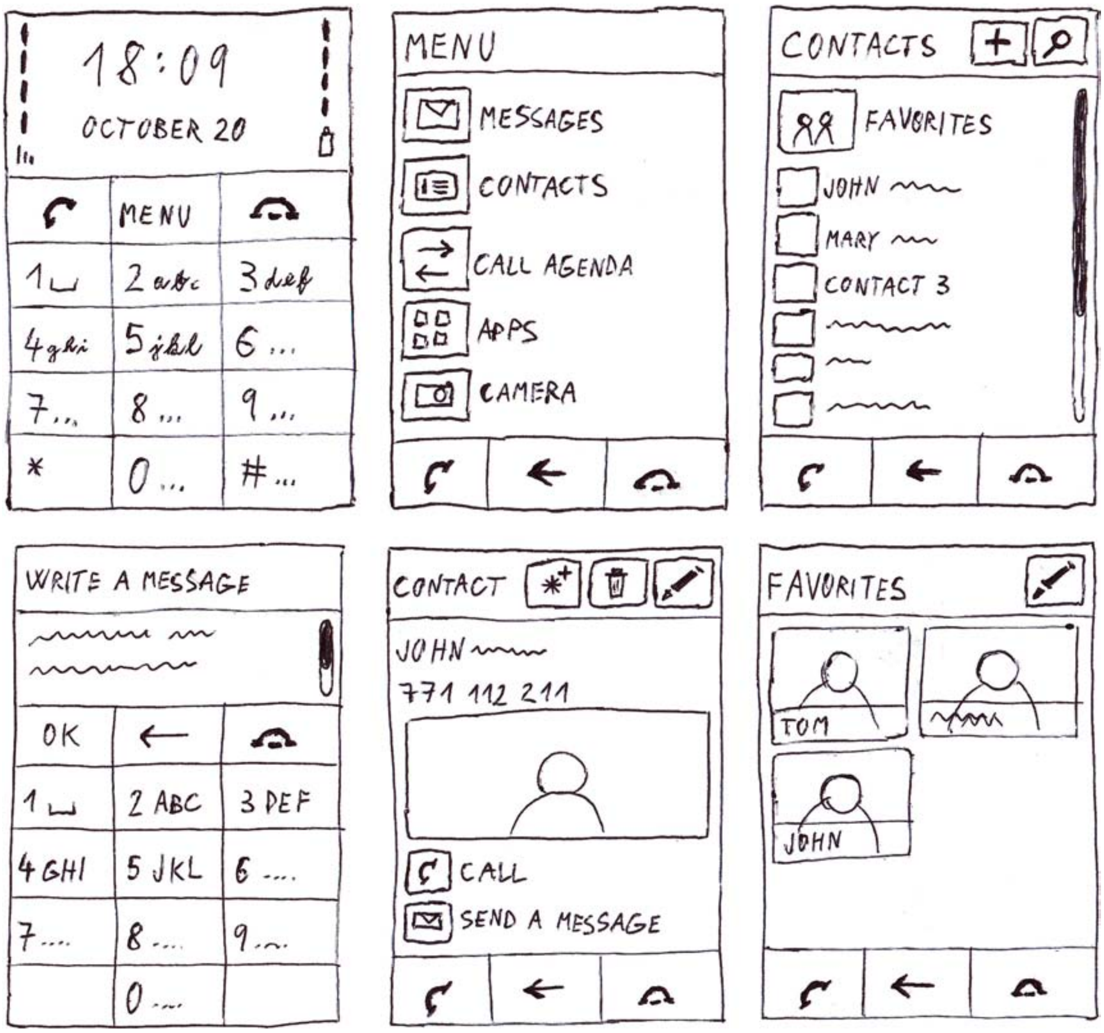


**Fig. 11.** Mockup 3

## 4.3. Problems of mockups

Each mockup had some advantages, but also some problematic characteristics. We will briefly summarize all identified problems of these interfaces:

#### Mockup 1

- Nonstandard wide 3x4 keypad could be confusing for some people. It is known that feature phones with similar keyboards did not sell well.
- Configurable buttons above the keypad changes their meaning and size by context.
- *Back* button at the right top corner is not easily accessible by hand.
- *Next* button takes up a lot of space on a display. It is also possible that this navigation model will not be intuitive. For example, how user can return back to previous content, if he press a *next* button by mistake? Will *back* button return him to previous screen, or to the previous content on this screen?
- Some screens will need more buttons (*edit a contact, remove contact, send a message...*), where we should place these buttons?

#### Mockup 2

- Icons on the main screens are quite small, pictograms do not have text labels.
- *Phone* and *Contacts* hubs are combining together too much features.
- *Phone* screen is overfilled with the content. For example, dialed number is displayed only on a very small part of the display.
- First label on the screen is the name of the screen, however the second label is the name of the next screen (that would be displayed after user clicks on the button).
- The time displayed at the top of each screen is very small.
- There is no home button on any screen for returning back to main screen.
- Favorite contacts are at two locations, on a main screen and on the contacts hub.
- There is no space left for the *settings* icon.
- The interface is, in general, quite complicated.

#### Mockup 3

- Pictograms on the top do not have labels, icons are not understandable.
- Area of the small screen (when the keypad is displayed) is quite small.
- It is not clear when the green button opens a screen with favorite contacts, and when it starts the call.
- The same problem is with the red button. It is unclear from the pictogram that this button will return user to the main screen.
- Link to favorite contacts is listed twice in this interface (in the *contacts* screen and on the green button).

## 4.4. Iterations of prototypes

### Prototype 1

We have designed the first prototype as the combination of mockups 1 and 3 (Fig. 12). The main screen contains a large keypad. Back button is displayed at the top, home button is at the bottom. The red button is not active while there is no call.

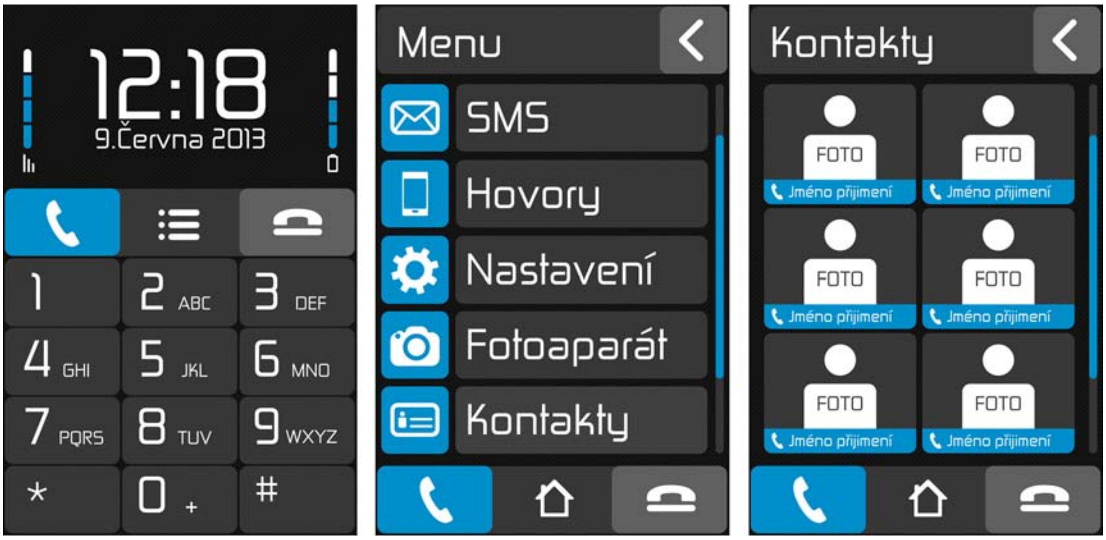


**Fig. 12.** Prototype 1

### Prototype 2

We have tried to invert the colors and change the font (Fig. 13). Back button was moved to the bottom, on the top panel has formed a new place for other buttons. We have tried to design an optimized alarm clock app, although it was not very understandable for old users. You can find all results of tests with seniors in the Chapter 5.

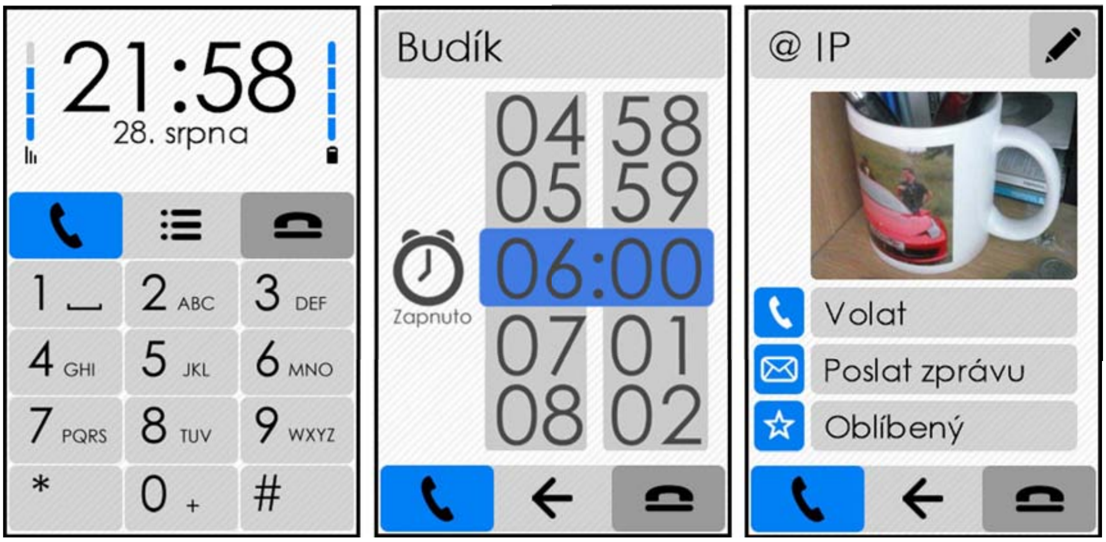


**Fig. 13.** Prototype 2

### Prototype 3

We have tried to add more colors to the prototype (Fig. 14). We have changed the pictogram of the green button to the silhouette of contacts. We have also designed a way how to configure favorite friends. However it was not very understandable for users. Meanings of the pictograms were not clear and the configuration of favorite contacts was complicated. We have described this in the Chapter 5.1.3.


**Fig. 14.** Prototype 3

## 4.5. Final prototype

The **final prototype** was the last iteration of our interface (Fig. 15-18). It was the last fixed and corrected version after all performed usability tests; just before we did the final quantitative empiric evaluation (see the Chapter 5.2). After this statistical evaluation, the prototype was also further developed, but these adjustments had not been covered more in this work.

We have prepared both white and black color variants of the interface. We have increased the contrast and added a new settings for a bold font. We have changed the pictogram of a red button to a home icon and also changed its color to violet. We have added an SOS button on the main screen. Backspace button for deleting last chars of the text has been moved to another position (see the Chapter 5.1.2).

We have changed the *contacts* screen again and the way how to configure favorite contacts. We have added some protective dialog boxes that are shown before the deletion of the item. We have also added notifications on the main screen for new SMS messages and missed calls. Optimized versions of an alarm clock and flashlight apps has been also finished. We have added three types of keyboards optimized for easy text input (see the Chapter 5.1.4).

### 4.5.1. Visual design of the final prototype

Final prototype has a very simple graphics (Fig. 15-18). Black or white **background** has a soft pattern with light stripes. When the high contrast mode is selected, the background is only black or white and the bold font is also selected.

The **font** used in the prototype is Century Gothic. Small size is used for helper texts and in the *contacts* screen. Middle size is used on the keyboard, on smaller menu items and in SMS messages. Largest font is used in the menu and on the top panel.


**Fig. 15.** Final prototype. Main menu and contacts (black theme with high contrast selected)

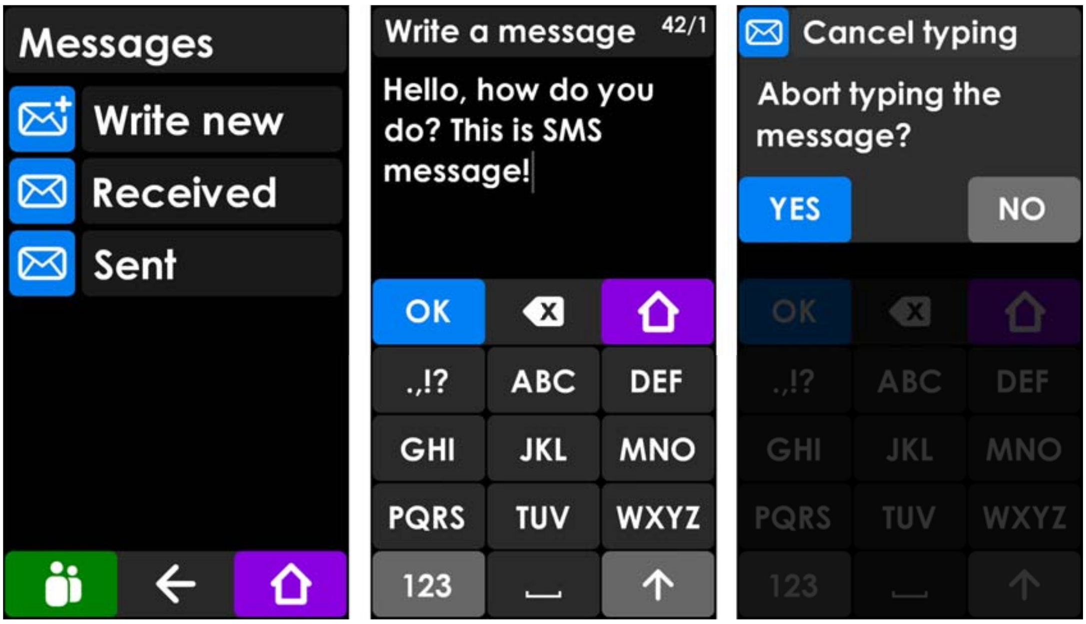


**Fig. 16.** Final prototype. Writing messages (black theme with high contrast selected)

All **buttons** have a bright color and rounded corners. Blue color is used for normal ordinary menu items. Violet color is used on the home button, on unread messages and missed calls. Dark green color is used on the contacts button and bright green on the buttons for starting the call. Red color is used when you need to confirm the deletion of the item, inactive button is gray.

Each button has a simple white **pictogram**. All possible pictograms on a top panel are: OK, Search, Add / Edit / Remove the contact, Save the number and Open the settings screen. Smaller items have several more variants (but they have a text label).

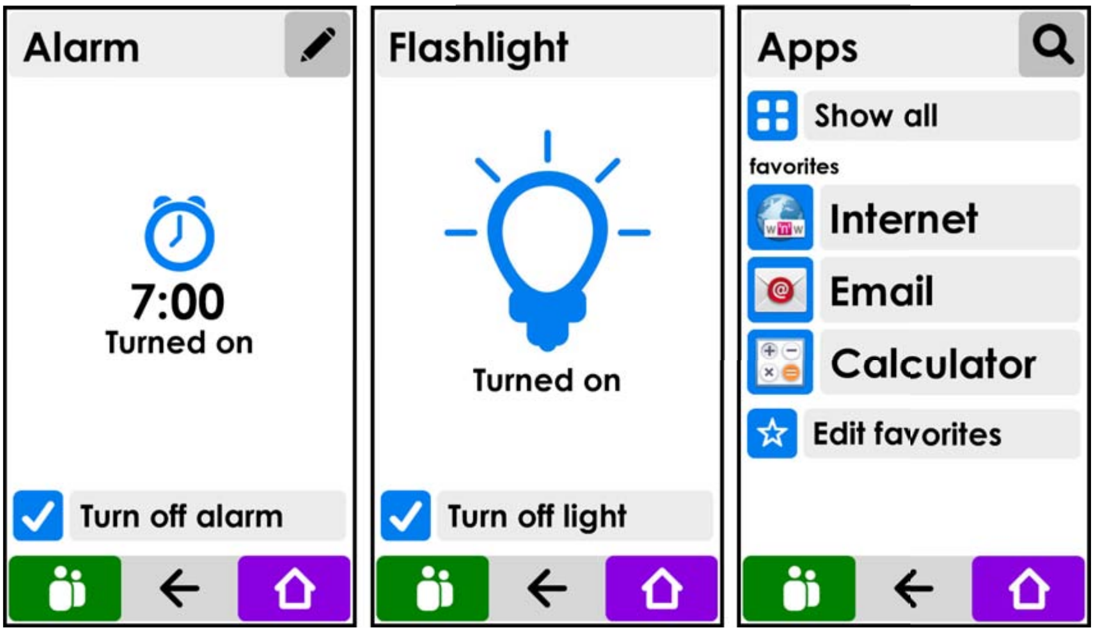


**Fig. 17.** Final prototype. Applications (white theme with high contrast selected)

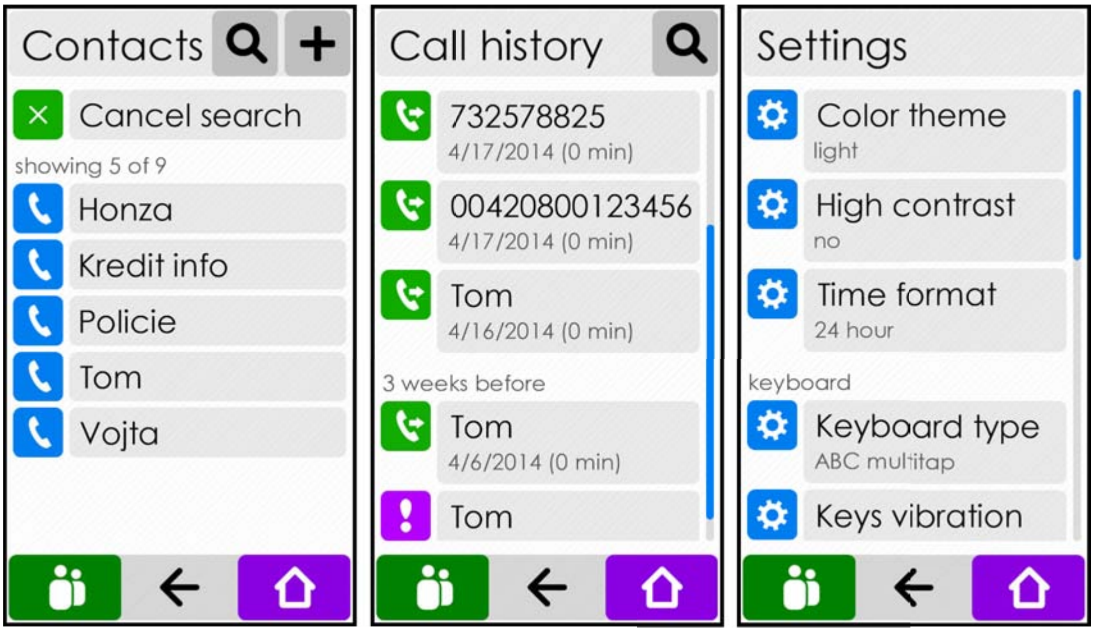


**Fig. 18.** Final prototype. Menu items (white theme with normal contrast selected)

### 4.5.2. Small and large screens

When the keypad is displayed, only **small part** of the screen is available for the content. We can dial numbers or write messages in this mode. If we click on a menu button, the keypad is shifted down with smooth animation and the view is extended.

Only three upper buttons (contacts, back and home) remain visible in the **large screen mode**. In just approximately 300 ms, the content of the small screen smoothly disappear and the big screen is shown (from the transparent state).

**Transition** between two big (or two small) screen has also an animation. If we select an item, last screen is moved to the left, the new screen comes from the right. When we click on a back button, the last screen is moved again from the left (in the direction of the arrow on the back button). This animation is also nice and smooth with transparency. Video of the interface in movement is included on the attached DVD.

### 4.5.3. Menu items

Two types of large screens could be found in the prototype. **Applications** specifically optimized for touch (as an alarm clock or flashlight) and **menu-style screens**. The menu-style screen is every screen with the list of items; for example the list of contacts, list of received messages or the main menu.

**Items** on menu-style screens can have three sizes (Fig. 19). Small (that can be found in a contact list), large (the main menu) and double items (for example the list of missed calls). Each item has a pictogram on the left and a text label on the right.

Each menu-style screen can have a collection of **helper text labels** at the top (for example *Select contacts to add to favorites*).

**The list of items** can also be categorized (messages *received today* or *received yesterday*). Each screen can also have a collection of additional buttons at the bottom (that can scroll with the content if we want, or do not scroll with the content).


**Fig. 19.** Small, large and double menu items

### 4.5.4. Notifications and dialog boxes

When user receives a new message (or get a missed call), the **notification** on the main display is shown (Fig. 20). The icon is flashing from the blue to violet color. Tapping on the notification will open the message, user can also dismiss it by tapping on a little cross at the right. Notifications are stacked over each other; only the last one is shown.

**Dialog boxes** are used for displaying questions (Fig. 20). User can accept or deny the dialog (for example whether he wants to delete the contact, or not). Buttons of dialog boxes are configurable (their colors, as well as their titles on the buttons).

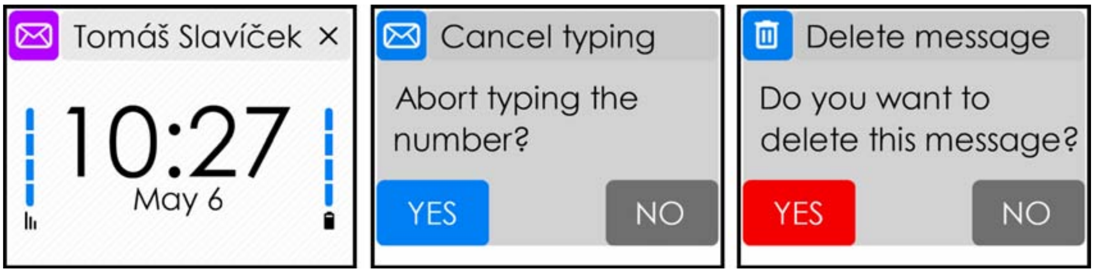


**Fig. 20.** Notifications and dialog boxes

### 4.5.5. Keyboards for text input

We have designed three types of keyboards (Fig. 21). **ABC multitap keyboard** takes an inspiration from classic feature phones. One letter can be written by one, two or more presses on a particular key. Anyone who knew the traditional way of writing messages, can use this method easily. The layout can be switched to numbers, at the bottom there are also *Shift* and *Space* buttons.

Special diacritic chars can be written by an additional presses on buttons: letter Á = 4x pressed ABC button. Correct diacritic chars are automatically selected by the language of the phone (the information is loaded from the Unicode standard).

The second type is an enlarged **sliding version of the qwerty keyboard**. It is the same as on the computer, buttons are large and require only one tap to write one letter. The trick is that only one part of the keyboard is displayed, for other letters user needs to move with the keyboard to the side.

The third type is an **experimental half-screen variant** of something "in between" those two solutions. Only one part of the alphabet is shown on a display. For the second part user needs to click on an arrow on the side. The layout automatically slides to the left with an animation and the remaining letters are shown. See the video 2 on the attached DVD. Evaluation of keyboards can be found in the Chapter 5.1.4.


**Fig. 21.** ABC multitap keyboard, sliding qwerty keyboard, half-screen ABCDE keyboard

## 4.6. Implementation

### 4.6.1. Installation and usage of the prototype

The final prototype can be installed on all devices with **Android 2.3** or newer, on both mobile phones and tablets (although it is designed primarily for phones). It works as a system launcher, you can select it as a replacement of your standard interface (install the app, click on the home button of the phone and choose this prototype). Minimum required display resolution is 320x240 (Fig. 22).

To return back to your **previous interface**, select a *Settings* icon in the main menu of the prototype, then tap on the *System settings* item. Now you can access the system settings, you can slide down with the upper system drop-down bar and also uninstall the prototype. Each version of Android, and each manufacturer, has a little different system settings, so we will not go in more detail here.

You can also select *Apps* in the main menu of our prototype, then click on *Show all apps* or *Edit favorites* and find there your specific application. Previous system launcher will be also located in this list.


**Fig. 22.** Final prototype deployed on real devices

#### Deployment

The **.apk installer** of the prototype is included on the attached DVD, as well as the **source code**. You can just copy the installer file to your phone and run it. You must allow the installation from an unknown sources during the installation.

To be able to **compile the source code**, you will need an active Xamarin.Android licence (it is a paid framework, see the next paragraph) and Xamarin Studio installed (or Visual Studio 2010 and newer). The source code can be compiled on both Windows and Mac OS X operating systems. More detailed documentation how to compile the project is included on the DVD.

### 4.6.2. Technology

The prototype was developed in **C#** using a **Xamarin.Android** framework (formerly named MonoDroid). For the user interface was used a **MonoGame game framework** (with some adjustments) and a custom set of our own helper classes for loading and drawing images, scaling and positioning UI elements and handling the touch (Fig. 23).

This combination of technology is quite unusual for Android development. Typical Android apps are developed in a Java language, or in the native C++. Xamarin and MonoGame are **cross-platform** solutions. The same app, only with small adjustments, can be deployed on all major mobile platforms (Android, iOS and Windows Phone).

The second advantage of this technology is that everything is **hardware accelerated** on the GPU (graphics card). Typical cheap smartphones have a slow CPU (system processor). However their graphics card is very powerful, optimized for games. It can work very efficiently with hundreds of textures in 60 frames per second, even with many transparent images.

Rendering high resolution images on a GPU can very quickly drain the battery of the phone. We have extended the MonoGame framework to be able to **suspend drawing**. When there is no animation on the display (only a still image), graphical buffers are frozen and the game loop is not running. It saves both CPU and GPU cycles and minimize the battery consumption (see more in the Chapter 4.6.5).

**Fig. 23.** Technologies used for developing the prototype

#### Xamarin.Android Framework

**Xamarin** is a cross-platform implementation of Microsoft .NET Framework [15], it is based on an open source project Mono. Developers can re-use their existing C# code and share it across Android and iOS devices to write apps with the native performance.

It is possible to use all the native mobile APIs from the Xamarin. We can use Android APIs, call classes from .NET namespaces, or even bind specific existing Java libraries (for example the Google Analytics library). The source code in **Xamarin.Android** looks almost like a Java code, we can use Android Cursors for loading resources, there is also an Activity screen, the Android manifest is the same. However we can use almost all modern C# 5.0 features in the Xamarin, for example LINQ, Parallel LINQ, reflection or async & await constructions.

The new **Xamarin.Mobile** API also helps with accessing the native resources of devices. One code can be shared for accessing a camera, loading contacts etc. Although this API is still in a beta and it does not have access to all system features.

**Xamarin.Android** apps are compiled just in time on the phone, they run within the Mono execution environment [16]. This virtual machine runs on top of the linux kernel, side by side with the Dalvik virtual machine (standard environment for Java applications). Both environments are written in C language and exposes various APIs to developer's code (Fig. 24).

Xamarin's Mono environment is installed together with the application, it takes up approximately 2.5 MB in the .apk package.

Xamarin.Android is a **paid product**, the basic license cost $299 USD per year (per one developer). The license for students and universities cost $99 USD per year. They also provide a free starter license, but it does not work with the MonoGame framework. The licence can be obtained on their website [15].

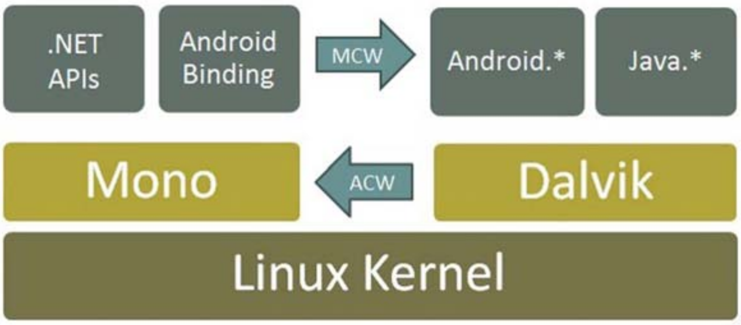


**Fig. 24.** Xamarin.Android architecture

### Frameworks for the user interface

We can use several approaches for drawing user interface on Xamarin.Android. The first option is to use standard **Android XML files** (Fig. 25). These files are compatible, we can just take them from a Java Android project and use them in Xamarin.Android. Standard system Android components and buttons will be drawn, however our interface will work only on Android devices, not on iOS, Windows or other phones.

The second option is to use some cross-platform framework for the user interface. We have used a hardware accelerated **MonoGame framework**. We were able to design all components of the interface from scratch.

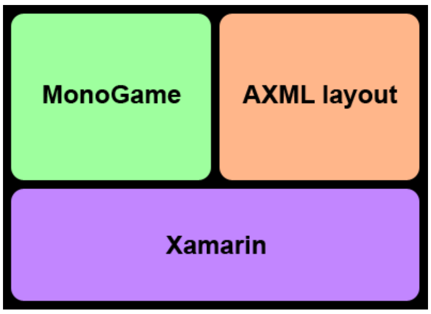


**Fig. 25.** User interface frameworks on the Xamarin.Android

### MonoGame framework

**MonoGame** is a free, open source game cross-platform framework with a very active community [17]. It is based on the Microsoft XNA Framework 4.0. It currently supports Android, iOS, Windows Phone 8 and other platforms like a console Ouya, Playstation Mobile or Linux.

MonoGame uses an **OpenGL ES 2.0** graphics library on Android and the Xamarin framework with C# language as the programming framework (Fig. 26). It is optimized for drawing 2D sprite graphics, as well as rendering not very complicated 3D graphics. Everything is accelerated by the graphics card.

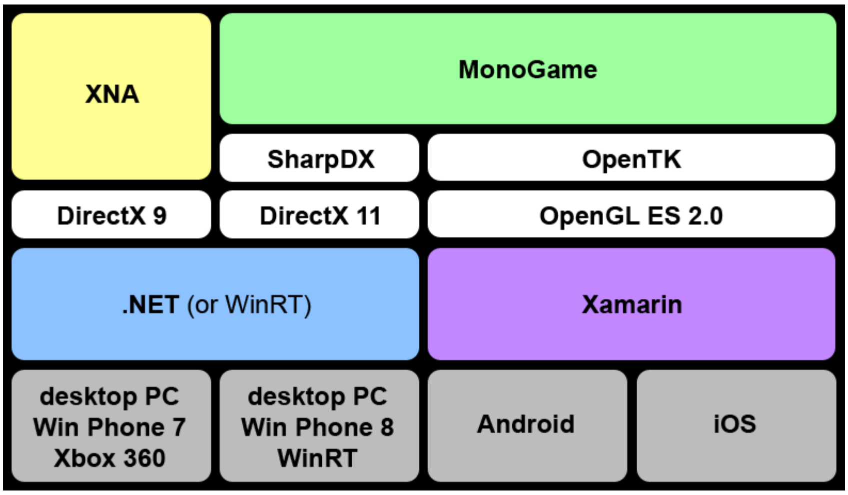


**Fig. 26.** MonoGame and XNA Framework implementation on selected platforms

## 4.6.3. Architecture of the prototype

Typical Android application consists of several activities. **Activity** is the name for one Android screen in the app (for example the main menu, or settings screen). Each activity can start another activity. The previous screen is stopped and the new screen is loaded. When user presses the back button, he is returned to previous activity.

Stopping and resuming activities is slow. We also can not change how these system animations will look like. We want to draw both *small* and *large screens* (with keypad hidden) together, with smooth and transparent transitions. We have designed custom top panels, notification messages and message boxes. Our prototype acts and behaves very differently in comparison to standard Android.

We have implemented our prototype only on **one activity**, with own navigation model. The **Container** object contains two collections of small and large screens (see the simplified class diagram on the Figure 27). The Container has methods for switching screens and animating them. It also holds references to a *Keypad* object, small and large top panels (*SmallTopBanner, TopBanner*) and the *MessageBox*.

The main **Activity1** in our prototype handles system events (for example activation and deactivation of the window) and logs system errors. The **Game1** handles the main application loop, with all our modifications (reloading content after deactivation, freezing graphical buffers, when there is no animation etc.).

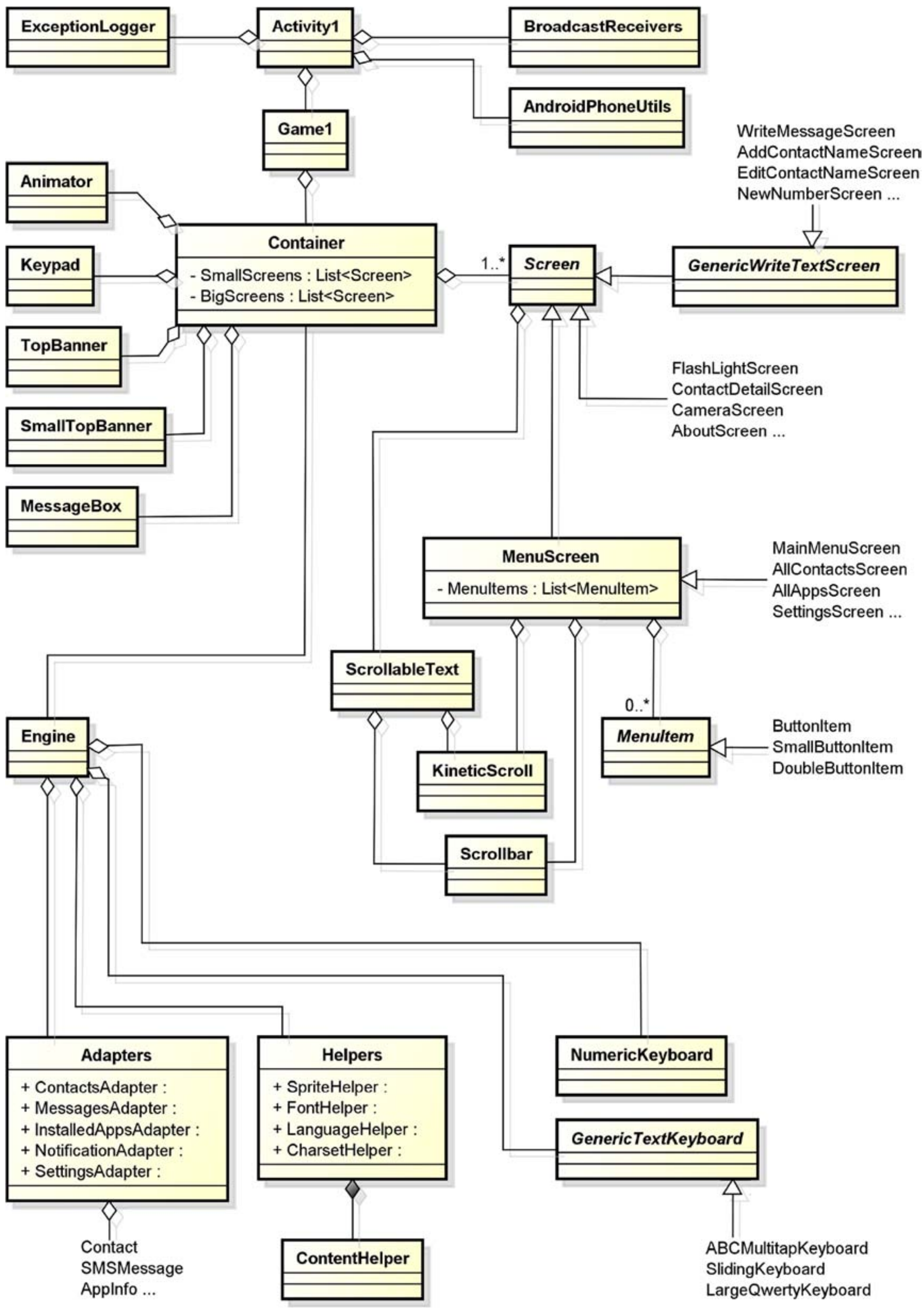


**Fig. 27.** Architecture of the prototype

All screens are inherited from an abstract object *Screen*. **MenuScreen** objects contain a collection of menu items (small, normal, or double sized) and the scroll bar. **GenericWriteTextScreen** objects are used for inserting text and numbers. Other screens are for example a **CameraScreen** or a **FlashlightScreen**.

### 4.6.4. Implementation details

#### Screens and the navigation model

In our prototype we have implemented 22 big screens (main menu, settings menu...) and 12 small dialog windows (with keyboard displayed).

Each screen has several **properties**. We can select whether we want to show the top panel, what title should it have and what types of buttons should be displayed there. We can also choose the type of keyboard for this screen (for inserting text or numbers), or which text will be displayed on the OK button. Inherited screens (like the screen for the main menu) are just setting these parameters in their constructor.

If the screen is configured that it contains the *Add* and *Search* buttons at the top, corresponding methods *PressedAddButton* or *PressedSearchButton* are automatically called when user taps on each button.

Every screen has also a **Tag** property. When it is changed, the method *TagChanged* is called. The screen can handle the incoming object. For example when it receives the contact or message, it can display it. This principle is also used for navigation between screens, we can also pass the Tag object there.

#### Efficient loading images in MonoGame

Typical way how to load images into MonoGame is to use the **Content Pipeline**. It is an integrated mechanism for loading images and fonts into the project. Images are converted on a PC into **.xnb files** during the compilation time. These files are copied into the phone and loaded in runtime. The advantage is that some devices do not have a standard file system (for example the Xbox 360), so they can load images this way. The problem is that these images are converted into an uncompressed format. Loading of them is very slow, they also take up a lot of space.

We have used a similar method that Jake Poznanski proposed [18]. We are loading textures **directly from .png** files. However we had to extend this method, we can not premultiply alpha in runtime, because it was causing graphics artifacts on some mobile graphics cards. We have used a PNG-Alpha-Premultiplier by maxme [19].

We are **caching textures** during the runtime, so one specific file name is loaded to application only once. The textures are not loaded all at once at start up, but we **load them lazily** just before we need to draw them. When the app does not have enough memory, it automatically **unloads** old unnecessary textures. We use the large heap settings on Android phones (with the Android 4.0 operating system or newer).

When the app is deactivated, we need to hook on the **DeviceReset** event and **reload** all textures from memory (as soon as the graphics device is already prepared - we need to wait for it at an Update method). We cache both textures and their byte values, to be able to restore them very quickly. We are using a similar method to one described in the article that we wrote on a Nokia Wiki [20].

### Selecting an image quality by DPI

In the typical Android application, **image quality** of textures is selected automatically. Several folders are prepared for developers (*hdpi*, *ldpi*, *mdpi*, *xdpi*) and images are chosen by the display resolution of the phone and the DPI (pixel density in dots per inch).

This method does not work in MonoGame. We have used a similar approach. We have prepared images in a *75%* resolution, *100%*, *150%*, *225%* and *337%* (Fig. 28). The correct image quality is chosen by our classes **SpriteHelper** and **ContentHelper** by the width of the display. The basic 100% resolution corresponds to 480x320 pixels, 225% is a 1280x720 and 337% is Full HD (1920x1080).

We have prepared a **pixel perfect graphics** for the most common display resolutions. Unusual sizes of displays get the nearest value. We do not take into account the DPI of the display, only the resolution. A bigger display will have the same number of buttons (but they will be bigger). The whole user interface will be correctly stretched.

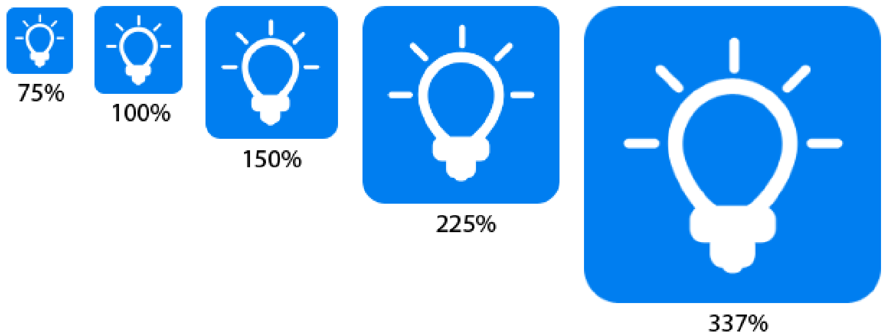


**Fig. 28.** Quality of images used in the prototype

### Loading system resources

For loading system resources from the phone (contacts, SMS messages) we have written a set of helper adapters. These classes will be different on Android, iOS and other devices. **ContactsAdapter** is used for loading a list of contacts, loading the photo of the contact and the number. **InstalledAppsAdapter** is used for loading a list of applications and **MessagesAdapter**, similarly, for loading SMS messages.

These resources are loaded on the second thread. When the corresponding *MenuScreen* is opened, loading of the items is **started on the background**. When all data are fetched, the screen is redrawn. Old menu items are replaced by new menu items. The position of the scrollbar is restored, as well as the list of selected items (if there are check boxes on the screen). The access to collections is automatically **locked** (in *MenuScreen-CacheMechanism*) to prevent deadlocks between threads.

We are using simple Cursor queries by **ContentResolver.Query** calls. Cursors are used on Android for simple queries on the system data. We are opening and closing these cursors by ourselves. We are not using automatic **CursorLoader** objects, because they are managed by the Activity itself (and they also start another background threads), it was not 100% optimal. We were running out of memory on some devices.

### 4.6.5. Optimizing a drawing loop

MonoGame uses a **game loop** for refreshing user interface. Methods *Update* and *Draw* are called up to 60 times per second. In the *Update* method the touch input is obtained and positions of objects in the scene are actualized. In the *Draw* method the whole scene is rendered. This execution model is very convenient for games. The typical game is always in motion and objects are moving. Fixed time step helps developer with creating animations and also with computing in-game physics.

However this model is not very appropriate for **applications**. They are mostly static, with no animations. We do not need to redraw a static image too often. We will need to modify the game loop, if we still want to take full advantage of the game framework, to have the whole interface accelerated on a graphics card and to use a lot of transparency for animations between screens.

#### Measurements

You can see on Figures 29-31 that a traditional game loop in MonoGame is very **resource consuming** on Android devices. Three left charts visualize the CPU load of our prototype, running with the default MonoGame game loop. Only one static page of the main menu is displayed, all other system apps in the phone are turned off. We have measured the values using the **Android Debug Monitor** (monitor.bat) from the Android SDK.

The orange sectors display the **CPU load** of our app (in percents), blue sectors show the additional system overhead. Our prototype takes almost 50% of the processor load on both three phones! *Note:* most likely it is not the real half of the maximum speed of the CPU, for example the LG L4 was automatically under-clocked to 166 MHz at this moment, its normal speed is a 1001 MHz. Although it is certainly a higher load that we would expect.

#### Optimizations

The first improvement will be to skip all *Draw* calls, when nothing has changed on a display. In the MonoGame we can call a **SuppressDraw** method from the *Update*. It will skip the next *Draw* call, until the *Update* is called again. We can see on the middle charts that it is a very fast and useful modification. The CPU load has decreased to approximately 20-30%. However we can improve it even better.

We do not need to call an *Update* method at all, when nothing has changed in scene. We will completely stop the game loop. It will require some modifications of the MonoGame source code. This will be our new execution model:

- *Update* and *Draw* will be called only once at startup, then game loop will stop
- The game loop will be **started**:
  - when user taps on a display
  - OR when asynchronous event has occurred (SMS message was received...)
- Both previous events can start an animation. The game loop will be **stopped**:
  - when user released the finger from display
  - AND when the animation has stopped

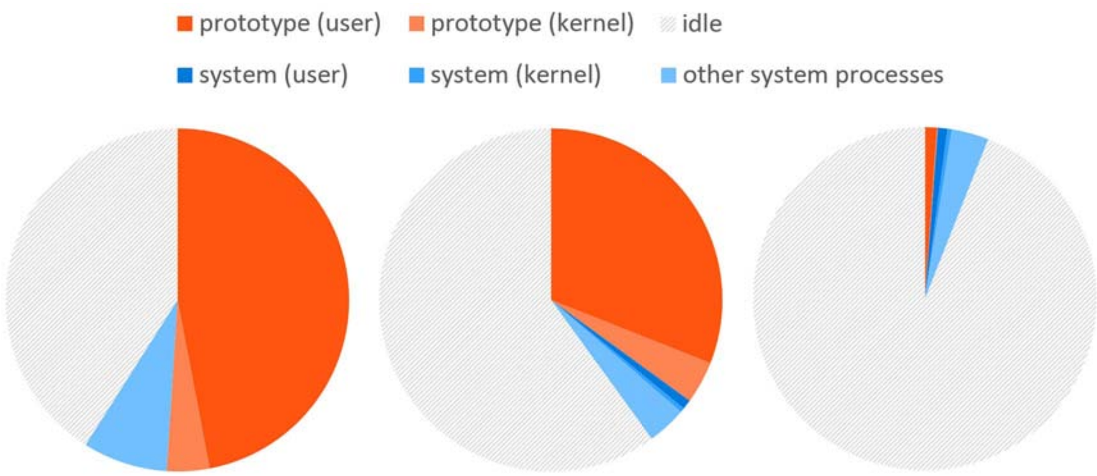


**Fig. 29.** CPU load on LG L4: default game loop, SupressDraw called, our implementation

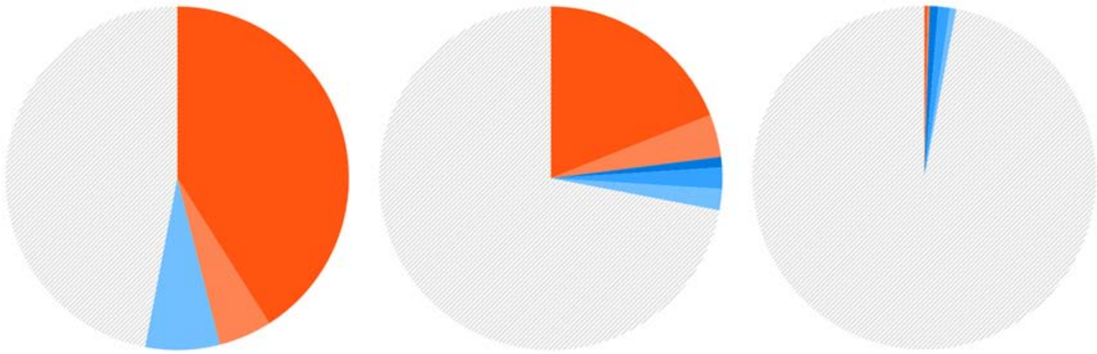

**Fig. 30.** Samsung Galaxy S3: default game loop, SupressDraw called, our implementation

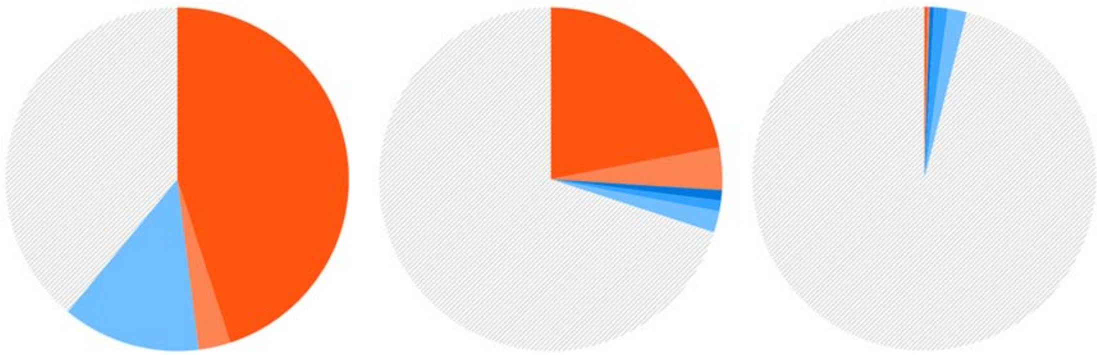

**Fig. 31.** LG Nexus 5: default game loop, SupressDraw called, our implementation

### Optimized game loop

We have extended a **TouchPanel** object: It notifies the game loop when a finger was pressed. We have also modified the **Game.Tick** method, so it can return the correct time from the last drawn frame. The game loop is controlled by the **GameView.Pause** and **GameView.Resume** calls. For some asynchronous events we have used the **System.Timers.Timer**, for example for redrawing the clock on the main screen.

This method has proven to be very effective. It only takes approximately 0.5% of the CPU load on a static scene, which is 100 times better than the default implementation of the game loop.

**Animations** are rendered with full CPU power in standard 60 frames per second, but after they stop, the app is switched back into the "frozen mode" and the screen is redrawn only again when needed.

# 5. User tests

We have designed three paper mockups and created a working prototype of the new interface. In this chapter we describe and summarize all performed usability tests.

Prototype **was tested** with old users. We performed these tests at their homes, not in the usability lab. This method has proven to be very effective for us. It allowed us very fast iterations of the interface. After each several sessions the prototype was corrected and enhanced, we were able to test it again with the same and also with different people.

Second part of this chapter describes the **empirical evaluation** of the final prototype. It was compared with the standard Android 4.4 interface. 14 people accomplished 6 very specific tasks on both interfaces. Their execution time was measured and number of mistakes counted. We have compared how both younger and older users can effectively use this interface.

## 5.1. Usability tests

Three paper mockups, as well as three iterations of prototypes (see the Chapter 4.2), were tested with old people. We have met approximately 10 different people in several informal sessions.

These were typical tasks to test:

- Add a new contact to the contact list.
- Find a specific contact in the contact list and call him.
- Add a contact to favorites, and remove it from favorites.
- Read the last received SMS message.
- Write a message and send it.
- Look who called you.
- Start a specific app.

### 5.1.1. Problems of prototypes

We have tested the prototype on a Samsung Galaxy S3 (4.8" AMOLED 720p display) and a Galaxy S3 mini (4" AMOLED 800x480). These were the main identified problems:

- Participants wanted to touch the scrollbar and drag with it. It was not possible in the first prototypes, the only way how to move with items, was to move with the content itself. We have fixed it in the final prototype. We have also added a search button to each screen, so users can filter any long content.
- The position of the SOS button in the Prototype 3 was not good, people touched it by mistake. The button looked almost exactly like the home button and even had the same red color. We have moved the SOS button to the bottom.

### Pictograms

- Pictogram for the menu button was not understandable. We have removed the icon and instead wrote there a text label: *MENU*.
- Some pictograms on the top bar were confusing, participants did not understand their meaning. We have simplified them and also got rid of many of them. We have found that these pictograms should also have text labels.
- Pictograms of missed and received calls in the *call list* screen were very similar. We have changed their color as well as the pictograms itself.

### Buttons

- People were pressing virtual buttons on borders of the display by mistake (when they held the phone in hand). We have added a little non-active gap there.
- A hardware back button was not active on our prototype (only the software back button, drawn on a display). People wanted to use both.

### Menu items

- Some items in the menu used to do two things. In the message list, tapping on the message icon started writing a message to this contact. However taping on the right part of the item opened the contact's detail. This behavior was confusing for people, so we have set only one action for each item.

### SMS messages

- When they selected the specific contact and then wanted to send him a message, they needed to click on OK button once, and then confirm the number again. We have simplified it. In this case only the SEND button is shown and it directly sends the message.
- Users could not select a specific contact after they wrote the SMS message; only to write a number. We have added a button there (Fig. 32).


**Fig. 32.** Selecting a number after writing an SMS message

### 5.1.2. New navigation model

**Navigation buttons** were little problematic for some people. They did not understand what the "hamburger icon" of the menu mean. We have changed it, and wrote there just a *MENU* label.

However much more confusing for them were doubled **backspace buttons** in the SMS messages (see the left bottom image in the Fig. 33). Middle button was used to cancel the text input, right button was used to remove the last char. People were mixing them up, the middle back button looked much more related to the same-colored gray keyboard, than a bright violet button.

We have changed the **navigation model**, the right button is always a home button (active or inactive). The middle button is always a step back to previous screen, or the deletion of the last char. The left button opens the contacts screen, calls the contact or sends the message. We have changed the meaningless OK label to a "Send".


**Fig. 33.** The old and new navigation model in our prototype

### 5.1.3. Configuration of favorite contacts

#### The first variant

We have designed three variants of the **favorite contacts** screen. Configuration of this screen was very problematic for people. The first prototype had a *plus* or *minus* pictogram on the contact detail screen (Fig. 34). When they clicked on the pictogram, the contact was added to the favorites screen. Participants did not see any connection between this pictogram and the favorites screen

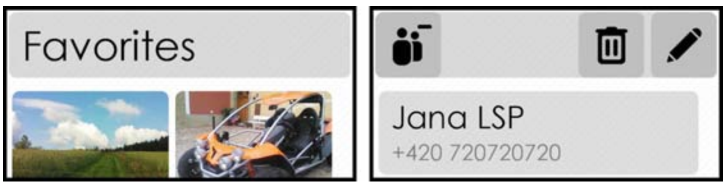


**Fig. 34.** The first variant of configuration of favorite contacts

### The second variant

In the second prototype we have moved the configuration directly to the favorites screen (Fig. 35). Tapping on the *plus* icon opened a screen with check boxes, when user could select his contacts. When he already had some contacts on the favorites screen, this screen had another icon (the icon of four check boxes). Tapping on this icon redirected him to the edit screen, where he could uncheck some contacts and click on OK, or click on a *plus* to add another contacts.

The most confusing was the *plus* pictogram. The same icon was used on *all contacts* screen for adding a new contact. People were mixing together these two screens and did not understand a difference.


**Fig. 35.** The second variant of configuration of favorite contacts

### Final implementation

We have merged together both *favorites* and *all contacts* screen to one screen. When user selects Contacts in a menu or clicks on a green navigation menu, the same screen is opened. The final implementation of *contacts* screen is shown on a Fig. 38.

The final **contacts screen** has two simple pictograms at the top panel, to *add contact* and *search contacts*. The primary view on this screen is the view with favorite contacts. We can change the view by *Show all contacts* button, or by tapping on a search. These two add/search buttons are always there.

The *Edit favorites* button is used for the configuration of **favorite contacts**. It opens the edit screen, where the selected contacts are located (with check boxes). On this screen there is also an *Add favorites* button.

This variant of the configuration was pretty intuitive for people. However we have discovered in the final evaluation (see the Chapter 5.2.9) that some people also wanted to add a contact to favorites from its detail. They simply overlooked the *Edit favorites* button on favorites screen and they were lost. We should add some *Add to favorites* button back to the contact detail.

## 5.1.4. Evaluation of keyboards

We have designed three types of keyboards for a text input, as it was noted in the Chapter 4.5.5. The first **ABC multitap** keyboard was pretty straightforward and easy to use for users, who were already writing SMS messages on a classic feature phone. They appreciated large buttons and did not almost have any problems with it.

### Sliding qwerty keyboard

For some people the **sliding qwerty** keyboard was better. They were rather slow in writing messages, but they appreciated the simplicity (so they do not need to press one button more times). They did not have much problems with the scrolling. They liked that the interface is working reliably (the recognition whether they want to scroll or write the letter). The scrolling was a little weird for some users, but altogether they were enjoying it.

The *qwerty* layout was rather complicated for some people. They preferred the *abcde* arrangement. Some people also wanted to see this type of keyboard in a landscape mode (and without the scrollbar). We did not implement the landscape mode yet, but we will consider it for the future work.

### Half-screen ABCDE keyboard

The third variant was rather experimental. **Half-screen keyboard** was very complicated. I had too many combinations of the layout that users were making mistakes, even after they got used to it. The alphabet was divided on two sub-screens. Numbers have also 2 or 3 sub-screens. Diacritics, or alternative alphabet (for example Cyrillic layout) added also another 2 or 3 sub-screens.

People were loosing a context where they are, whether they should press an arrow button to the right, or press the bottom button to switch the layout. The animation of the movement has been slightly helping them, but they still were not very successful.

### Comparison of keyboards

It seems that each person preferred one **specific keyboard**. In the final app it would be nice to have a dialog after the start, so every user can easily select which keyboard he would like to use (Fig. 36).

It will be very interesting to **compare these keyboards** more and measure the real efficiency how fast people can write on them. We tested in our evaluation only the basic ABC multitap keyboard. We wanted to measure the efficiency of the whole interface. We needed to "lock" as many variables as possible to have the most accurate results.


**Fig. 36.** Cyrillic alphabet; numeric layout on a sliding and a half-screen keyboard

## 5.2. Evaluation

We will use an **empirical research** method inspired by [21]. Mackenzie used this method in 2007 for evaluation of text entry techniques. He was comparing two types of touch-based keyboards, how effectively people can write on them.

In our evaluation we will compare our prototype with the standard Android 4.4. interface. For this technique it is very important to very precisely describe all test conditions. Which phone will be used for testing, how it will be set up and which exact instructions will people get. The research should be repeatable, all parameters directly measurable. Mackenzie talks in his course [22] how to select right testable questions and also how to correctly evaluate tasks.

### 5.2.1. Participants

We have tested the prototype with **14 participants** (Fig. 37). We gave them 6 simple, but very specific tasks (like a call this number: +420 800 123456). At first we have tested it with several less experienced people older than 70 years. However these tasks were very difficult for them (see Chapter 5.2.6) and measured values were not comparable.

We have extended the research and we have tested the prototype with both younger and older people. The youngest one was 19 years old, the oldest one was 62. Mean age of all 14 people was 38 years (with the standard deviation 14). Execution times of all people were compared in the graph by their age.

Our participants did not need to own or use a touch screen phone. We have only checked that they did not used our prototype before yet, as well as pure Android 4.4 interface (they have seen it for example in an article, but they did not use it as their primary phone yet).

**Fig. 37.** Testing a prototype

### 5.2.2. Procedure

At the start of each session we will show each participant briefly, how to use the interface. We will let him to try the interface himself. Then we will present him 6 tasks, measure his execution time and count his number of mistakes.

Each person will perform tasks on both interfaces (within-subjects design). Half of people will start with our prototype, half of them will start with the Android interface. Test will be performed in their familiar environment, like at their home or in the pub.

We are measuring these parameters:

- execution time in seconds for every task
- whether they finished the task (yes / no)
- how many errors did they have

Primary goal is to measure **the execution time** of tasks. However participants should not be stressed, the testing is not the race. They should be fine and relaxed and perform tasks as naturally as normal. If they do not finish the task, it is not their fault, especially Android interface is quite complicated for beginners. It is expected that especially old people will have problems.

As a secondary goal we are looking again for **usability problems** of our prototype. If more people have the same problem with one button, can not find a specific item or have problems with one particular task; we will take a note about it. We will fix the interface after all participants performed the test.

Every participant will get the same **instructions**. During the test we will not give them any additional advice. The time will be measured in seconds (by stopwatch) from the first touch on a display, to final state, when the task was finished (he called the number or sent that SMS). Maximum time for one task is 90 seconds (this value was determined experimentally).

#### Errors

An **error** is every unwanted action - when the participant clicked on a wrong button and the interface did something else. For example he wanted to open a detail of a contact to send him a message, but it started a phone call (by mistake). Or he wrote a wrong letter in SMS message.

We count both conscious and unconscious errors. For example he does not even need to notice that he wrote a mistake in the text. We do not advice him to correct it. When the participant is just taping through the interface and exploring screens, these are not errors. Some tasks can be performed in multiple ways.

The **count of errors** is only an informational value for us. Participants should avoid mistakes, but they should not worry much about them. Their primary goal is to try to finish the task. When they do not find a solution, they can skip the task and ask for help. We will show them how to do it.

### Post-test discussion

We will talk shortly with each participant after the test. We want to see their opinion on these questions:

- how does he like each interface
- what does he like or dislike on them
- which interface would be better for him

### 5.2.3. Test setup

We are testing on a Nexus 5 phone (4.95” TFT FullHD display) and a Samsung Galaxy S3 (4.8” AMOLED 720p display). They both have almost the same size and a large high quality display.

Both phones are in their factory settings (after the hard reset), they do not have a SIM card inserted. WiFi and 3G connection is also turned off. Brightness of the display is set to automatic. The language of the phone is set to Czech (the native language of participants).

**Our prototype**
It has the default settings. Dark theme is selected, with the normal contrast (not that higher one). Classic multitap 3x4 keyboard is selected. Vibrations are set on a minimum, but sound of the keys is turned on.

**Android 4.4**
Also in the default settings. Google Now screen is not visible, because the phone is not connected with any Google account. Icons of the "phone app" and messages are located at the left bottom.

Android 4.4 interface can be found for example on Nexus 5 devices. It is the main software that Google ships to manufactures for Android devices, without any modifications from Samsung, LG and other brands. Not many people have direct experience with it, so it is an ideal interface for our test.

**Each phone has some items inserted:**

- There are 7 contacts in each phone: *Hasici, Honza, Jana, Jirka, Kredit info, Policie, Tom, Vojta*
- 3 contacts are pinned as favorites: *Hasici, Jana, Tom*
- There are a couple of mised calls in the call history. The last was an outgoing call.
- There are no received or send messages in the phone.

### 5.2.4. Learning part

At the beginning, we will show to each participant the main screen of the phone and where is the home button located (how he can return back to main screen). We will show him the dialer and how to dial a simple number (for example 1234), also the screen with favorite contacts (both Android and our prototype have the similar screen). The main menu of the phone will be also shown to him.

We will **not** show him explicitly, how to dial a "+" on a keyboard, or how the keyboard for text input looks like. We will not reveal him, where is the list of last calls, how to add a new contact to contact list and where is the alarm clock located in a phone.

### 5.2.5. Tasks

Each participant will get these 6 questions, sequentially one by one.

#### Task 1: Call the number +420 800 123 456

The first task is very simple. We have shown him, where is the dialer. He needs to just find and open it and dial the number. The trick is with the "+420" at the beginning. For writing a plus symbol he needs to press # button three times on our prototype. On the Android he needs to long press the 0 button.

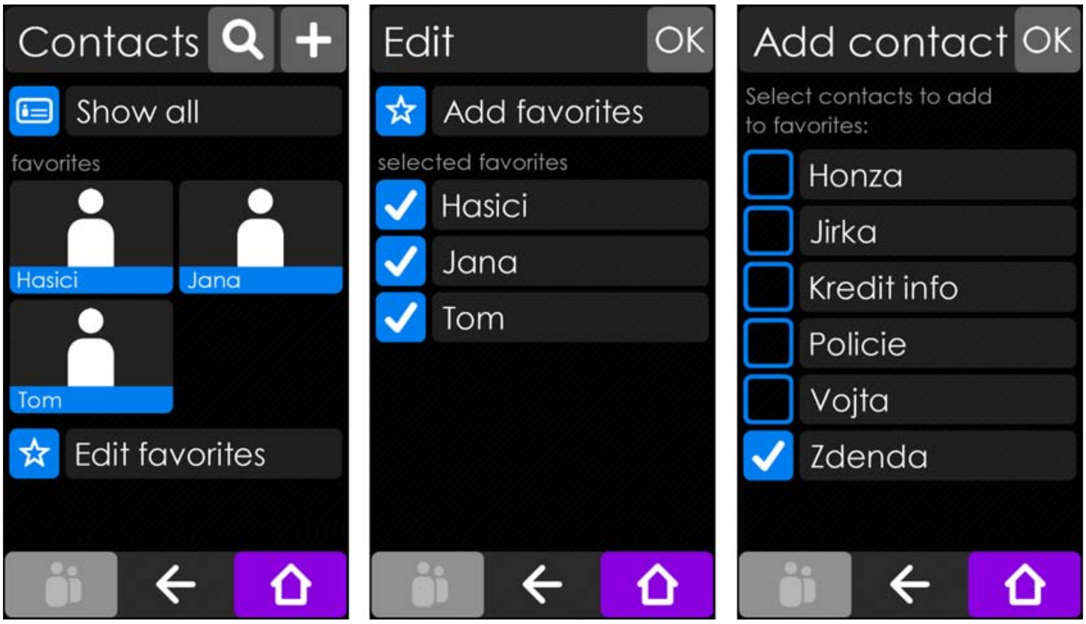


**Fig. 38.** Adding a contact to favorites on our prototype

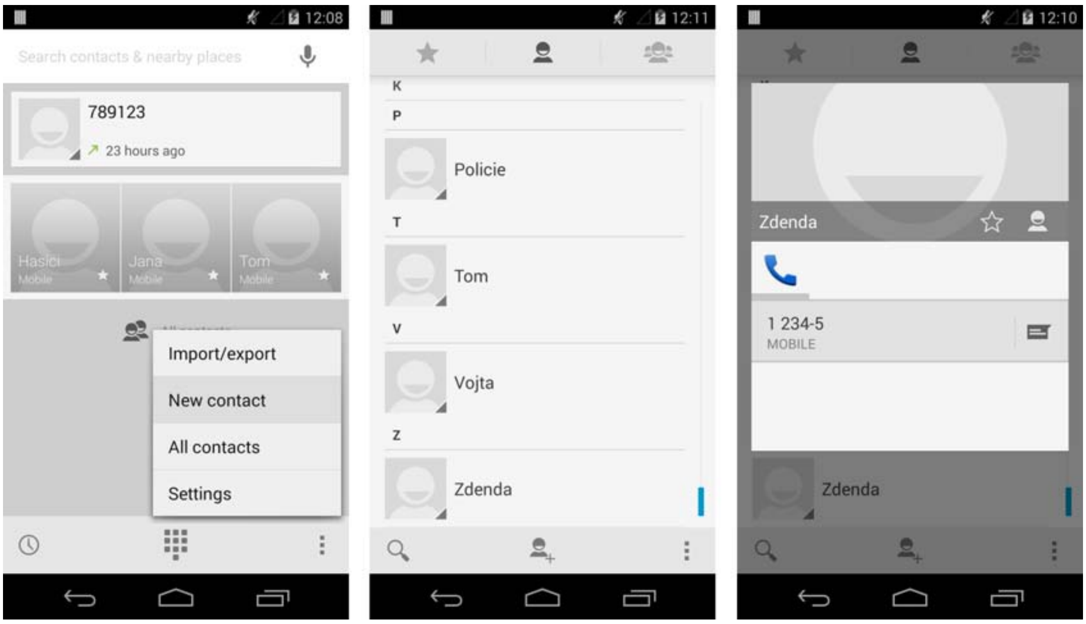


**Fig. 39.** Adding a contact to favorites on Android 4.4

### Task 2: Add a new contact *Zdenda* to your contact list and call him

A fairly common task. He needs to add a new contact to their phone (Fig. 42). He needs to remember, where the list of contacts was located and to find the *Add contact* button. He also needs to write a name *Zdenda* on a text keyboard correctly, insert the number, save it and call this contact.

We tell him that it is not important, which number he will insert. He should set something like a 1234. If he wants to insert a full number (like a 123 456 789), he can. We just pause a stopwatch for a while to make it comparable. When he forgot to insert any number, we will remind him that he should insert some.

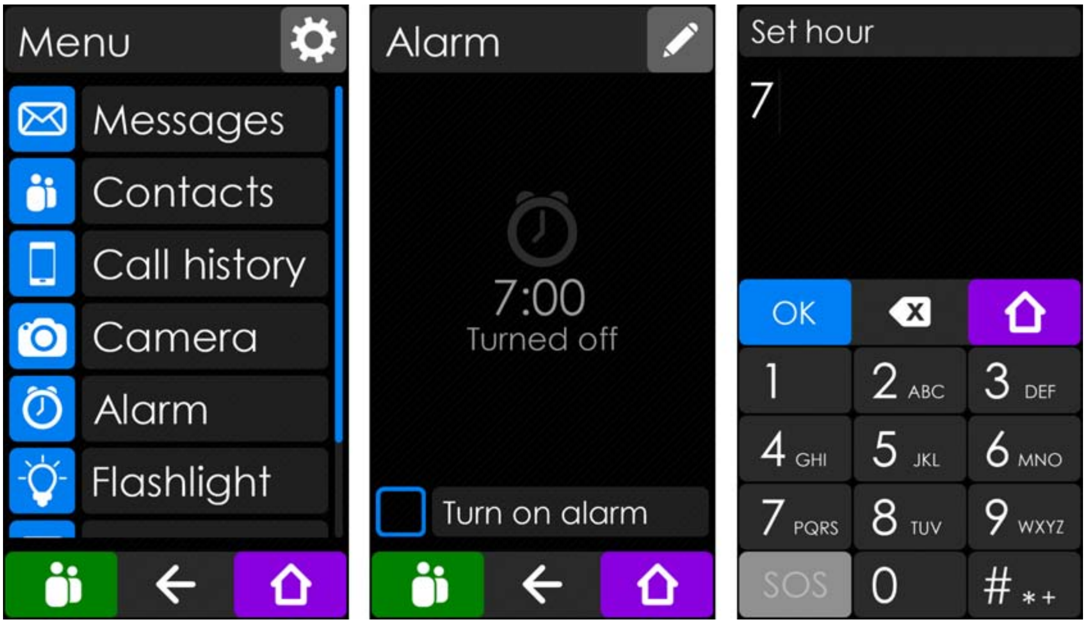


**Fig. 40.** Setting an alarm clock on our prototype

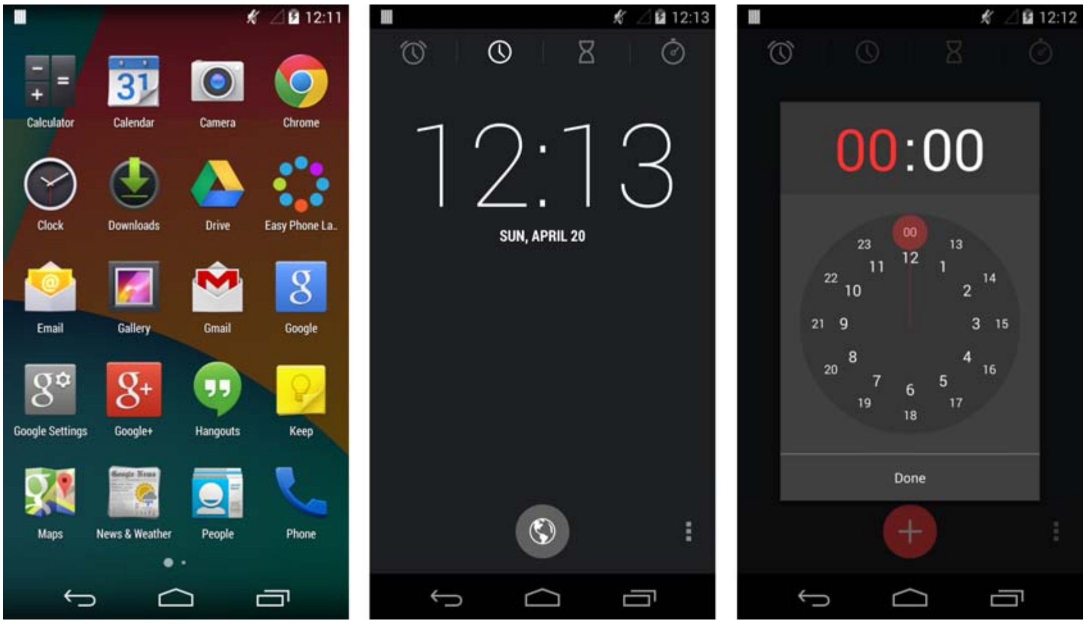


**Fig. 41.** Setting an alarm clock on Android 4.4

#### Task 3: Find who called you last time and call him back

The goal of this task is to find and open the screen with call agenda, then click on some item, except the most new one. The last item was an outgoing call.

#### Task 4: Add the contact *Zdenda* to favorites (speed dial)

This task was pretty easy to understand, but quite hard to perform (Fig. 38-39). They knew, where is the screen with favorite contacts (we have shown it to them). They needed to find the contact *Zdenda* and add him to this screen. In our prototype they needed to click on "Edit favorites". In the Android interface they had to to open a detail of a contact and click on a "star" icon.

#### Task 5: Write an SMS message to *Zdenda*: "Ahoj, prijedu v 6 hodin" and send it

This was a short test of a text entry. The goal was to write a Czech version of a message "Hello, I will come at 6 pm". They needed to find how to write a comma, and how to write a number. There were two possible ways how to do the task, to write a new message and then select a contact, or to select a contact and then write him a message.

#### Task 6: Set an alarm clock to 6:40 in the morning

This task tested how they can find a specific feature in their phone they did not used yet. At the our prototype they needed to open a menu, then select an *Alarm*.

At Android they needed to find a *Clock* icon in the main menu and click on the first tab on the alarm icon. Second part of this task was to find a way how to add or edit an alarm and set the correct time (Fig. 40-41).

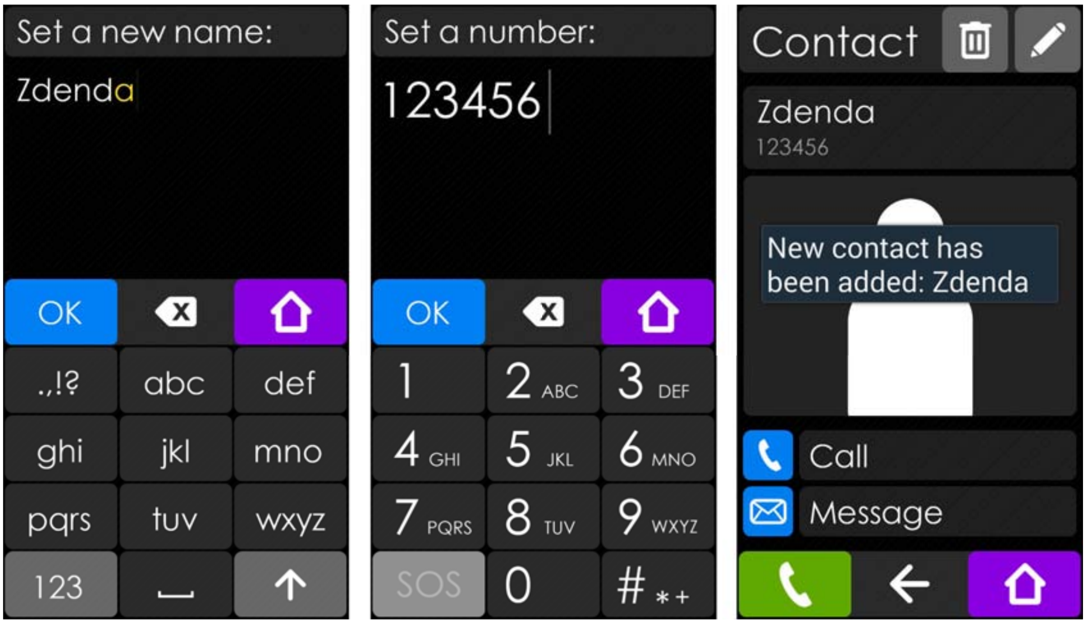


**Fig. 42.** Adding a new contact on our prototype

### 5.2.6. Testing with less experienced seniors

We tried to perform these tasks with **4 less experienced seniors** that were older than 70 years. However these tasks were very difficult for them. One participant, for example, did not know that a phone number can contain a "plus" sign, he was confused and did not understand what we wanted from him to do.

For users, which do not use SMS messages or never added any new contact to their phone, these tasks just **did not make any sense**. They would not even know how to do it on their phone. They were interested in these tasks and they wanted to try them. But their execution time was very slow. They stopped many times as they were thinking carefully what to do. Measurement times were not comparable.

They usually need to **become slowly familiar** with the environment and learn every task very slowly. They need to try every task many times with our instructions how to do it. When we did not want to tell them more information during the test, they were lost and confused. We were loosing a trust from them.

They were able to finish only **two or three tasks** on our prototype. We needed to change the interface to high contrast and a bold font, because some of them were not able to read the normal text. The Android 4.4 interface was there only for the illustration, they were not able to read on them anything. They also did not have any motivation why to try the same tasks on the Android (and to fight with it).

They would like to learn slowly one or two basic specific tasks, ideally with the help of the paper manual. Too many tasks in a row were confusing them.

Although we were not able to measure the execution times, these sessions were still very beneficial for us. We got a lot of usability feedback, as it was discussed in the Chapter 5.1.

### 5.2.7. Testing with all people

We have extended the research and tested **people of all ages**. The youngest one was 19 years old, the oldest one was 62. Mean age of all 14 people was 38 years (with the standard deviation 14).

For the purposes of the research, it would be definitely better to test only similar people with the same age. However seeking for these people would have been very time consuming and also outside the scope of this work. At least we were able to compare whether the execution times and number of errors depend on age.

### 5.2.8. Analysis

All measured values are included in the **Appendix C**.

On the Fig. 43 is the graph of **execution times of tasks**. On the axis X is the age of the participant. On the axis Y is the sum of execution times of all tasks, one marker for each interface and each user. Blue squares are times measured on our prototype, orange circles are times measured on the Android 4.4 interface.

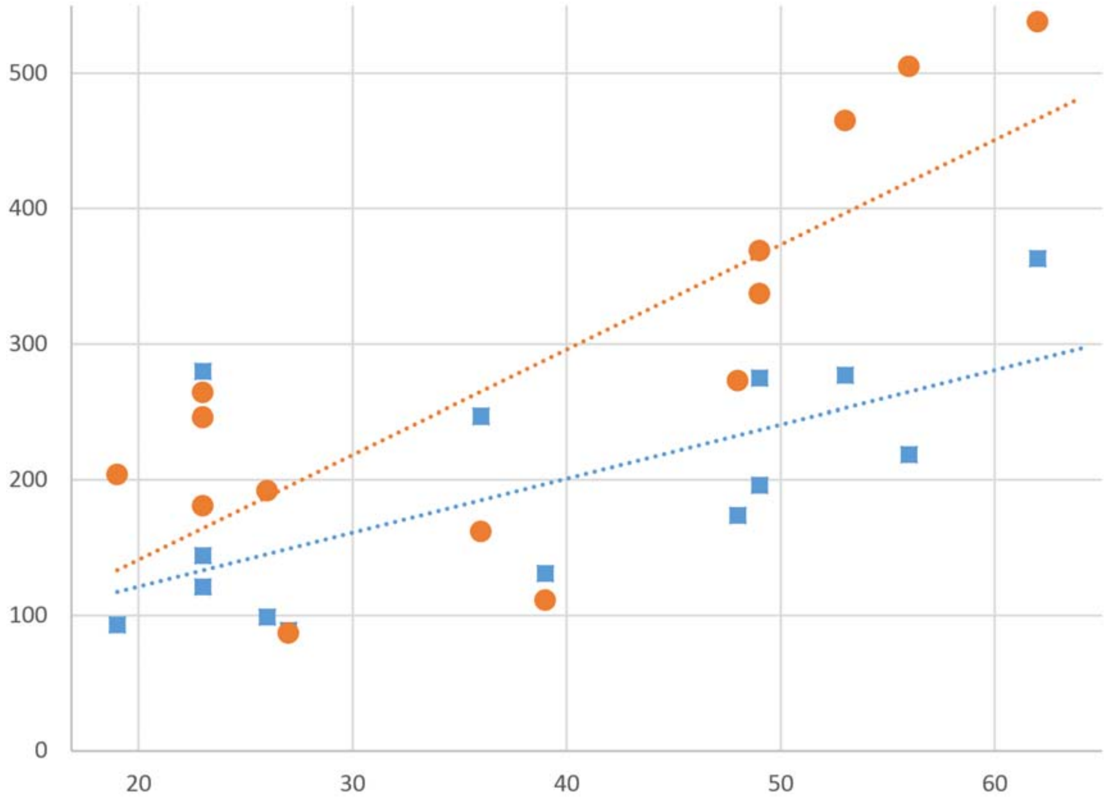


**Fig. 43.** Sum of execution times of tasks by age. Blue squares: our prototype, orange: Android.

Faster **execution times** were performed mostly on our prototype. For example: 19 year old participant finished all six tasks on our interface in 93 seconds (1.5 minutes), compared to Android that took him 204 seconds (3.4 minutes). The oldest participant was able to finish all tasks on our prototype in 363 seconds (6 minutes), the Android interface took him 9 minutes (or he did not finished some tasks, 90 seconds was a maximum time per each task).

Blue and orange dotted lines are linear **approximations of the trends**. It can be recognized that our interface was on average better. There is also very visible fact that older people were much slower in executing tasks that younger people (approximately 3-4 times worse).

However some people finished tasks faster on the Android interface. 36 and 39 years old participants were partially familiar with the Android interface. The 36 years old participant also did not manage to **find a call agenda** (Task 3) in our prototype. It was a pretty easy task (as the average time for other people was 12.9 seconds), but he spent all 90 seconds searching for it. He was incorrectly looking for it on the contacts screen, not on the other extra call agenda screen located in the menu.

Another problematic part of our interface was an **alarm clock screen**. Many people found where is this screen very quickly, but were not able to set a correct time. Our settings was split to minutes and hours separately. User had to select an hour, and then click on the OK button to select minutes. Participants wanted to fill both hours and minutes on one screen, or write some kind of a semicolon. 4 people from 14 had problems with that. For other people the average time was only 12.7 seconds (Fig. 44).

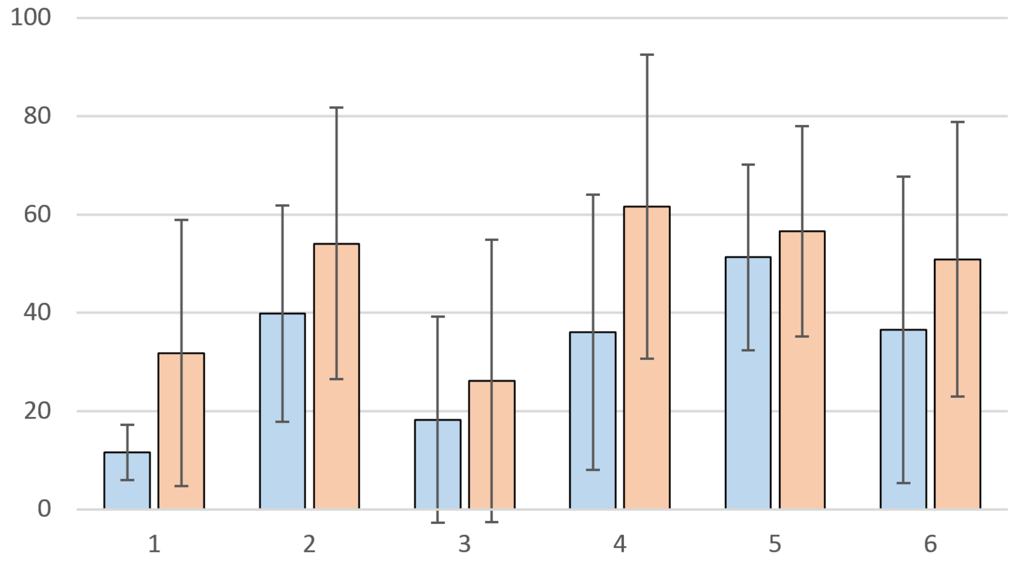


**Fig. 44.** Average execution time of tasks. Blue: our prototype, orange: Android 4.4.

It is clearly visible on Fig. 44 that average times for each task were very random. Each pair of the columns displays the average execution time, for each task. Left blue columns are values from our prototype, orange are from the Android interface.

The black lines are standard deviation lines. For example the shortest time of the first task was 6 seconds (for our prototype) and the longest was 27 seconds (with standard deviation 5.62, see the **Appendix C** for all values). Shortest time for the first task on the Android was 9 seconds and the longest was 90 seconds (standard deviation 27.1).

### Improvements

We also tried to compare only values from people younger than 40 years, and then make **another comparison** of people older than 40 years. However the standard deviations were not better, numbers of their problematic tasks were very similar. Both younger and older people had some problems with some specific tasks, compared to other people that were successful in these tasks.

It would be good to perform these tests again, but with only one selected specific age group, for example 50-59. We could also show them explicitly the each task. Measured times would be better and more comparable (with lower standard deviations). It will improve the internal validity of the test. However it will also narrow the insight into the real behavior of people. We could measure only how fast they can click, not whether they are able to finish the task by itself.

### 5.2.9. Evaluations of errors

Average numbers of errors for each task are shown on a Fig. 45. For example the first task was performed by 14 people, they together made 5 errors on our prototype, and 9 errors on the Android interface.

The most common mistakes for each task (on our prototype) were:

**Task 1: Call the number +420 800 123 456**
All participants found the plus button on a keyboard, but some people were trying to long press this button, which did nothing. Especially younger people were used to long press the buttons, older people did not have problems with that.

**Task 2: Add a new contact Zdenda to your contact list**
Icon of a plus button (at the right top corner) was very hard to find for some users. It would be better to add a large text item to the contact list *Add new contact*, or to add text labels for the upper icons.

**Task 3: Find who called you last time and call him back**
There were no problems with this task.

**Task 4: Add the contact Zdenda to favorites**
Many people were trying to add a contact to favorites from its detail, but the correct way was to open a screen with favorites and click on *Edit favorites* button. Favorite contacts are integral part of our experience and their settings must not confuse people. There should also be another button *Add to favorites* on the contact detail screen.

**Task 5: Write an SMS message to Zdenda and send it**
The waiting time for a new letter was very long (if user wanted to write a letter E after D). Proposed 1500 ms was too much, even for seniors, it should be shortened to 700 ms or less. Nobody used tapping on the display to accept the actual letter.

**Task 6: Set an alarm clock to 6:40 in the morning**
Users wanted to write a semicolon and fill both hours and minutes on one screen. These screens should not be split, or we should add some helper text labels.

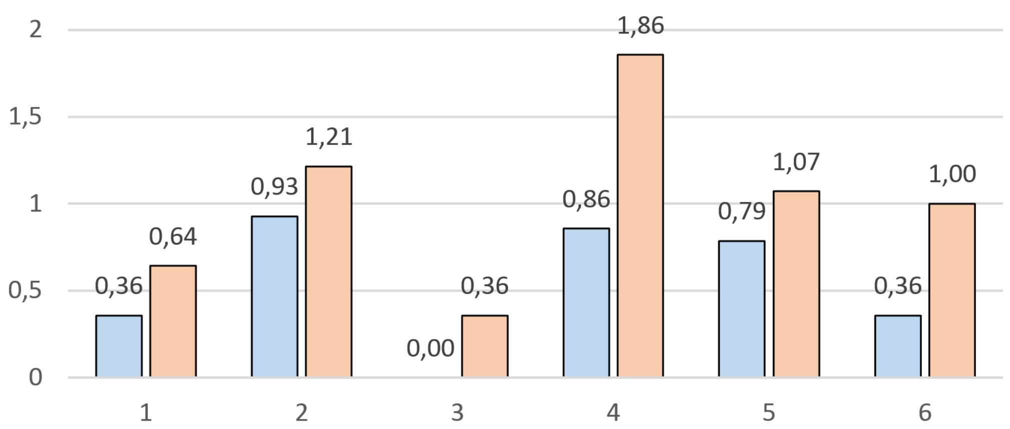


**Fig. 45.** Average numbers of errors per user / task. Blue: our prototype, orange: Android 4.4.

# 6. Conclusion

The main objectives of this thesis were to research the problem how old people use their phones. What is their **motivation**, what are their specific capabilities and limitations. To design a new mobile interface optimized for elderly users, create and test the prototype.

We have performed both **qualitative and quantitative research**. We have found and verified how elderly use their phones and what they like and dislike. We were able to identify specific personas and summarize the recommended guidelines for the optimized interface for old users.

We have used all these findings to **design a working prototype**. We have started from a couple of paper mockups, designed several iterations of the layout and tested them with old people. We have discussed the specific usability problems and fixed them in the final prototype.

Finally we did an **empiric evaluation**, a statistical test with 14 people. We have measured how efficiently they can use the interface and compared their execution times and numbers of errors with the standard Android interface.

We have also talked about the **technology** and discussed how the prototype was developed. We have modified the game framework to be able to load images efficiently, and to draw them with the lower battery consumption.

## 6.1. Future work

We have focused in this research primarily on the **visual part** of the interface and the primary user experience. It will be interesting to explore also the **sound**, reading text aloud and navigating in the interface by voice commands. The prototype could be extended to meet the needs of blind users, or people with Parkinson's disease. We could try to support larger sizes of fonts, or the landscape layout.

We have proposed **three types of keyboards** in chapters 4.5.5 and 5.1.4, but we did not evaluated them in much detail. It will be interesting to test them with people again, compare their speed and the number of errors (similarly, as we have tested the whole interface with an empiric evaluation).

Our **empiric evaluation** method could be also extended. As we have written in the Chapter 5.2.8, it will be appropriate to perform another evaluation with more people, with the similar age and maybe with more restricting conditions.

However, we were able to design a **real working prototype** that can really run on mobile phones. Old people liked it and they were saying that they would even use it on their phones. We should really take an inspiration from this prototype, and based on its properties, create the final app.

# Appendix A.

# Session guide for the qualitative study

All questions of the session guide for the qualitative research are included here. Participants were interviewed in a semi-structured discussion, we talked about their daily usage of the phone, what features they use and what they like and dislike:

**General questions**

Please, tell us something about where you live.
Can you describe your usual work day?
How many friends do you have here? Do you live with your family?
How often do you visit your friends/family?
Do you call them by phone occasionally?
Do you take your phone with you, when you go out? (for shopping...)
Are you happy that your mobile can be always with you?
Did you have a classic stacionary landline phone before? (not mobile)
How did you mantain a contact with your family before?

**Mobile usage**

How many people do you call occasionally?
Do you have their numbers saved in your phone? Or written on a paper?
Did you save these numbers to phone by yourself?
Do you sometimes dial a number by pressing digits on a keypad?
How do you recognize who is calling you? (when the phone rings)
What if the phone stops ringing (you have a missed call). How do you recognize who was calling?
Do you look sometimes into the history of last calls?

**Accesibility**

Do you see well what is shown on the display on your phone?
Do you need to take glasses for some operations?
What about the keyboard. Is it large and readable enough?
Can the buttons be pressed easily?
Are you satisfied with the phone size and shape?
How could be the shape better? More thick, or large?

**SMS messages**

Do you send SMS messages from your phone sometimes?
What is your opinion on sending text messages? Would you rather call?
Do you handle writing messages on your phone well? Is it complicated?
Is it the keyboard large enough?
If you do not write SMS messages... Why? What is the main problem?
Can you at least read a received message?
How do you recognize that you have received a new message?
Do you receive an unwanted messages sometimes? (advertisement...)
Do you erase received messages? Why? Is it important for you?

### Advanced features

Do you send or receive emails on your phone?
What other features do you use on your phone? An alarm clock?
Do you use a reminder, organizer or a calendar on your phone?
Are these features even available on your phone?
What if you want to note or save some information? (a shopping list)
Do you save this note to your phone, or somewhere else?
Would you welcome a voice recorder in your phone? (a dictaphone)
Do you use any other features on your phone? Some applications?
Which feature do you miss on your phone?
How about the camera? Taking photos, showing pictures to friends...
What should the most perfect phone be able to do?

### Complications

Does your phone annoys you sometimes? Making you upset.
In which situations? Is it frequent?
How often do you need to connect your phone on a charger?
Do you think your phone is too complicated? Its menu and items.
Are there any features you would like to use, but you are not using them?
How it should be done better?
What other features would you improve on your phone?

### Healthcare

Have you ever used your phone for calling a healthcare assistance?
How do you think your phone can save your life?
Which people should your phone contact if you need help? Your family?
How would you call for a help if you need some? To press some SOS button?
What if the phone identifies when you fall down on a floor, for example?
What if your phone reminds you when you need to take a medical pill?

### Touchscreen phones

What do you think about modern touchscreen smartphones?
Have you already tried some device with a touch screen?
Can you imagine you would use one? If your daughter gives it to you.
If not, why would you be afraid? Why do you think it is not for you?
How would you benefit from a large screen?
What if this phone has a large text, great contrast, simple buttons...
Large display can display nice photos. Would you use phone as a camera?
What about making a video call with your family on the phone?
What other features should modern smartphone have?

# Appendix B.

# Questionnaire for the quantitative study

The full questionnaire for the quantitative part of the research is included here. It was offered only to people older than 50 years that currently use a mobile phone.

**How do you use your phone?**
only for receiving calls
sometimes I call someone
I am calling and receiving calls regularly

**How many people do you call typically?**
less than 5 people
5 to 10 people
10 to 20 people
more than 20 people

**Who saved numbers to your phone?**
myself
somebody else (a son, a grandson...)

**Do you add new numbers to your phone?**
sometimes I add a new person to my phone
I add new people regularly
no (I have the same list of people in my phone since the beginning)

**How do you call people?**
I primarily use a list of saved people in my phone
I mostly dial numbers on a numeric keypad
I use both variants

**SMS messages**
do not use SMS messages
I can read a received message
sometimes I write a message to someone
I am sending SMS messages regularly

**What do you dislike on writing SMS messages? *(you can select multiple items)***
it is too difficult for me
keys are small on the keyboard
text is small on a display
my phone writes different things that I want to write
I do not have the motivation - it is really better to make a call
I have no problems

**What type of phone do you use?**
classic phone (with buttons)
special senior phone
a phone with a touch screen

**Your phone**

I can see well what is on the display
sometimes I need to take on prescription glasses
I see very poorly on the display, it is restricting me

**What does a mobile phone mean for you?** ***(you can select multiple items)***

nothing, it is only a piece of plastic
I can connect with my family and friends easily
it gives me a sense of security
I can keep with a new technology
I can call a medical assistance very quickly

**Which features do you use on your phone:** ***(you can select multiple items)***

calling people
SMS messages
a camera
radio / listening music
alarm clock or reminders
flashlight
playing games
sending e-mails
browsing the internet
other apps (maps, navigation, Skype...)

**Do you understand these names of features?** ***(you can select multiple items)***

protocols
calls
call agenda
programs
SMS
conversations
messages
applications

**Please select all the statements you agree with:** ***(you can select multiple items)***

My phone contains more features that I could utilize.
I can't understand some names of items in my phone.
Advanced features should be hidden in my phone, they only confuse me.
Sometimes I am afraid I will delete something in my phone by mistake.

When I receive a message, my phone should be flashing and making a lot sounds.
Sometimes I need to look who called me lately (for example a previous night).
I have changed an image on my display, or changed my ring tone.
I would like to have a camera or voice recorder in my next phone.

I have already used my phone for calling help (and it helped me).
My phone should be able to read loudly what is written on a display.
I would appreciate a larger font (text size) on my phone.
My phone should be able to call help on 1 button press (or contact family members).

## Segmentation questions

**Your age group**

50 - 59 years
60 - 69 years
70 - 79 years
more than 80 years

**Gender**

male
female

**How do you live?**

alone
with a partner
with family (with a daughter, grandchildren...)
in a shared care (a retirement home)

**Your experience with technology** ***(you can select multiple items)***

do not use a computer
I use a computer (without the internet)
I use a computer with the internet
I have internet in my phone

# Appendix C.

# Measurement from the empirical research

All measured values from the empirical research are included here. We have measured the execution time and a number of errors of 14 people on our prototype and on the Android 4.4:

**Selected tasks**

1. Call the number +420 800 123 456
2. Add a new contact Zdenda to your contact list
3. Find who called you last time and call him back
4. Add the contact Zdenda to favorites
5. Write an SMS message to Zdenda and send it
6. Set an alarm clock to 6:40 in the morning

**Execution time: Easy Phone**

| *Age* | 19 | 23 | 23 | 23 | 26 | 27 | 36 | 39 | 48 | 49 | 49 | 53 | 56 | 62 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task 1 | 10 | 15 | 10 | 8 | 6 | 7 | 6 | 8 | 9 | 27 | 11 | 20 | 12 | 13 |
| Task 2 | 16 | 22 | 39 | 27 | 17 | 17 | 20 | 20 | 61 | 52 | 53 | 70 | 61 | 82 |
| Task 3 | 7 | 11 | 25 | 17 | 23 | 7 | 90 | 5 | 7 | 16 | 13 | 7 | 22 | 5 |
| Task 4 | 17 | 45 | 71 | 26 | 9 | 19 | 38 | 49 | 17 | 90 | 11 | 12 | 11 | 90 |
| Task 5 | 33 | 44 | 45 | 33 | 31 | 30 | 72 | 30 | 47 | 69 | 72 | 78 | 51 | 83 |
| Task 6 | 10 | 7 | 90 | 10 | 13 | 9 | 21 | 19 | 33 | 21 | 36 | 90 | 62 | 90 |

**Execution time: Android 4.4**

| *Age* | 19 | 23 | 23 | 23 | 26 | 27 | 36 | 39 | 48 | 49 | 49 | 53 | 56 | 62 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task 1 | 31 | 11 | 23 | 14 | 13 | 13 | 12 | 9 | 13 | 27 | 44 | 55 | 90 | 90 |
| Task 2 | 63 | 66 | 35 | 46 | 21 | 15 | 21 | 21 | 48 | 90 | 61 | 90 | 90 | 90 |
| Task 3 | 6 | 9 | 18 | 10 | 15 | 2 | 13 | 6 | 4 | 22 | 31 | 50 | 90 | 90 |
| Task 4 | 47 | 36 | 30 | 67 | 90 | 20 | 29 | 5 | 90 | 90 | 90 | 90 | 90 | 88 |
| Task 5 | 39 | 59 | 43 | 66 | 42 | 24 | 42 | 41 | 39 | 50 | 77 | 90 | 90 | 90 |
| Task 6 | 18 | 65 | 32 | 61 | 11 | 13 | 45 | 29 | 79 | 90 | 34 | 90 | 55 | 90 |

**Number of errors: Easy Phone**

| *Age* | 19 | 23 | 23 | 23 | 26 | 27 | 36 | 39 | 48 | 49 | 49 | 53 | 56 | 62 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| Task 2 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 2 | 2 | 1 | 1 | 0 | 3 |
| Task 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Task 4 | 1 | 2 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 2 |
| Task 5 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 2 | 0 | 1 | 3 |
| Task 6 | 0 | 0 | 2 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

**Number of errors: Android 4.4**

| *Age* | 19 | 23 | 23 | 23 | 26 | 27 | 36 | 39 | 48 | 49 | 49 | 53 | 56 | 62 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Task 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 | 1 | 2 |
| Task 2 | 1 | 2 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | 2 | 0 | 0 | 4 | 4 |
| Task 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 1 |
| Task 4 | 3 | 3 | 3 | 3 | 3 | 3 | 1 | 0 | 0 | 3 | 2 | 0 | 2 | 0 |
| Task 5 | 0 | 0 | 2 | 2 | 0 | 2 | 0 | 2 | 0 | 0 | 1 | 0 | 4 | 2 |
| Task 6 | 0 | 1 | 1 | 2 | 0 | 0 | 2 | 2 | 2 | 0 | 0 | 1 | 1 | 2 |

# Appendix D.

# Contents of the DVD

The enclosed DVD has the following structure:

**Final prototype - APK installation**
Installation file of the final prototype (it can be deployed on Android 2.3 devices or newer, see the Chapter 4.6.1).

**Final prototype - screenshots**
Screenshots of the final prototype.

**Final prototype - source codes**
The source codes of the prototype (written in Xamarin / MonoGame).

**Prototype iterations - video**
Videos of the user interface of selected iterations of the prototype.

**Text of thesis - latex sources**
LaTeX source code of this Thesis.

**Research documents**
All important documents about the **research**, mostly in Czech language.
The **questionaire** (blank questions, the results in a graph and raw data in CSV).
Notes from the **literature review**.
The results of the **empiric evaluation** (both Microsoft Excel files and data in CSV).

**The text of this work and a readme.txt document is also included on the DVD.**

# Bibliography


[1] Gartner. *Smartphone Sales Surpassed Sales of Feature Phones*. 2014. URL: http://www.gartner.com/newsroom/id/2665715 (visited on 05/11/2014).

[2] Richard Pak and Anne McLaughlin. *Designing Displays for Older Adults (Human Factors & Aging Series)*. 1st ed. CRC Press, Dec. 2, 2010. ISBN: 1439801398.

[3] Arthur D. Fisk et al. *Designing for older adults: Principles and Creative Human Factors Approaches*. 2nd ed. Boca Raton, FL, USA: CRC Press, 2009.

[4] MunaS. Al-Razgan et al. "Touch-Based Mobile Phone Interface Guidelines and Design Recommendations for Elderly People: A Survey of the Literature". In: *Neural Information Processing*. Ed. by Tingwen Huang et al. Vol. 7666. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 568–574. ISBN: 978-3-642-34477-0. DOI: 10.1007/978-3-642-34478-7_69. URL: http://dx.doi.org/10.1007/978-3-642-34478-7_69.

[5] Hyunjeong Kim et al. "Contextual Research on Elderly Users' Needs for Developing Universal Design Mobile Phone". In: *Proceedings of the 4th International Conference on Universal Access in Human Computer Interaction: Coping with Diversity*. UAHCI'07. Beijing, China: Springer-Verlag, 2007, pp. 950–959. ISBN: 978-3-540-73278-5. URL: http://dl.acm.org/citation.cfm?id=1766311.1766421.

[6] Hazrina Hassan and Mohd Hairul Nizam Md Nasir. "The Use of Mobile Phones by Older Adults: A Malaysian Study". In: *SIGACCESS Access. Comput.* 92 (Sept. 2008), pp. 11–16. ISSN: 1558-2337. DOI: 10.1145/1452562.1452564. URL: http://doi.acm.org/10.1145/1452562.1452564.

[7] Karen Renaud and Judy van Biljon. "Worth-centred Mobile Phone Design for Older Users". In: *Univers. Access Inf. Soc.* 9.4 (Nov. 2010), pp. 387–403. ISSN: 1615-5289. DOI: 10.1007/s10209-009-0177-9. URL: http://dx.doi.org/10.1007/s10209-009-0177-9.

[8] Alex Olwal, Dimitris Lachanas, and Ermioni Zacharouli. "OldGen: Mobile Phone Personalization for Older Adults". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 3393–3396. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979447. URL: http://doi.acm.org/10.1145/1978942.1979447.

[9] Bessam Abdulrazak et al. "PhonAge: Adapted SmartPhone for Aging Population". In: *Inclusive Society: Health and Wellbeing in the Community, and Care at Home*. Ed. by Jit Biswas et al. Vol. 7910. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 27–35. ISBN: 978-3-642-39469-0. DOI: 10.1007/978-3-642-39470-6_4. URL: http://dx.doi.org/10.1007/978-3-642-39470-6_4.

[10] Yoko Asano et al. "Tips for Designing Mobile Phone Web Pages for the Elderly". In: *Proceedings of the 12th International Conference on Human-computer Interaction: Interaction Design and Usability*. HCI'07. Beijing, China: Springer-Verlag, 2007, pp. 675–680. ISBN: 978-3-540-73104-7. URL: `http://dl.acm.org/citation.cfm?id=1772490.1772567`.

[11] Y. Batu Salman, Young-Hee Kim, and Hong-In Cheng. "Senior 2014; Friendly icon design for the mobile phone". In: *Digital Content, Multimedia Technology and its Applications (IDC), 2010 6th International Conference on*. Aug. 2010, pp. 103–108.

[12] Apple. *iOS Accessibility portal*. 2014. URL: `https://www.apple.com/accessibility/ios/` (visited on 05/11/2014).

[13] Google. *Android Accessibility portal*. 2014. URL: `https://support.google.com/accessibility/android/#topic=6004781` (visited on 05/11/2014).

[14] Microsoft. *Windows Phone accessibility information*. 2014. URL: `http://www.windowsphone.com/en-us/how-to/wp8/settings-and-personalization/accessibility-on-my-phone` (visited on 05/11/2014).

[15] Xamarin. *Native iOS, Android, Mac and Windows Apps in C#*. 2014. URL: `http://xamarin.com/` (visited on 05/12/2014).

[16] Xamarin. *Android architecture under the hood*. 2014. URL: `http://docs.xamarin.com/guides/android/under_the_hood/architecture/` (visited on 05/12/2014).

[17] MonoGame. *Cross-platform game framework*. 2014. URL: `http://monogame.codeplex.com/` (visited on 05/12/2014).

[18] Jake Poznanski. *Speeding up XNA Content Load*. 2014. URL: `http://jakepoz.com/jake_poznanski__speeding_up_xna.html` (visited on 05/12/2014).

[19] Maxme. *PNG-Alpha-Premultiplier tool*. 2014. URL: `https://github.com/maxme/PNG-Alpha-Premultiplier` (visited on 05/12/2014).

[20] Tomáš Slavíček. *UI Framework for XNA and MonoGame with high DPI support*. 2013. URL: `http://developer.nokia.com/community/wiki/UI_Framework_for_XNA_and_MonoGame_with_high_DPI_support` (visited on 05/12/2014).

[21] I. Scott MacKenzie. "Evaluation of text entry techniques". In: *Text entry systems: Mobility, accessibility, universality* (2007). San Francisco, CA: Morgan Kaufmann., pp. 75–101.

[22] Steven J. Castellucci I. Scott MacKenzie. "Empirical Research Methods for Human-Computer Interaction". In: *CHI2013, April 27 – May 2, 2013, Paris, France* (2013). ACM 13/04 – 13/05.