Multi User Support for Senior Citizen Visual Guidance System

University of Oulu
Department of Information Processing Science
Master’s Thesis
Canrong Deng
March 4, 2016
Abstract

Due to the population aging phenomenon, the working-age population will decline in the future while the seniors’ quality of life can decline. A lot of research has been done and many of researchers figure out modern ICT and mobile technology needs to shoulder more responsibility in elderly care. The benefit of science and technology should be brought to senior citizens’ life.

Moreover, the smartphone ownership rate has been increasing in recent years, especially the percentage of Android phone ownership. Because elders will take smartphone everywhere, installing an application in senior citizen’s Android phone is a good way to track their movements. The caregivers could give guidance remotely to elders based on their location and surrounding environment scene by using mobile technology of this kind.

The multi user support for senior citizen visual guidance system consists of five components: Android phone, Pupil headset, Indicator-based Glasses, OldBirds and web server part. The system uses the Pupil headset to capture senior citizen’s font view and transfer it to OldBirds part which is controlled by caregivers. Furthermore, the Indicator-based glasses are used to show the guidance orders given by remotely caregivers. The web server part is transfer station between Android phone and OldBirds part.

Google Cloud Messaging service has already been integrated into this system so that caregivers can directly give guidance orders to senior citizens about when and where to go.

For future work I suggest, from the hardware perspective, to connect Pupil headset’s world camera and eye camera together by a USB hub. Furthermore, I recommend to combine Pupil headset and Indicator-based glasses together to remove the need for Bluetooth connection.

Keywords
Senior citizen, Android, OldBirds, Real-time video

Supervisor

University Lecturer Ari Vesanen
Professor Petri Pulli
Foreword

I start working on this thesis at the beginning of June, 2015. This research follow previous studies made by Raappana and Korvala (2015); Pulli et al. (2012); Firouzian, Asghar, Tervonen, Pulli, and Yamamoto (2015). The multi user support for senior citizen visual guidance system aimed at transfer Pupil headset’s camera view to the OldBirds part in real time and the caregiver can directly send guidance messages to old citizen’s Indicator-based Glasses.

I spend about 4 months on coding and faced a lot of technology problems on developing this system. After several discussions and brainstorming, my supervisors, helped me to make sure the orientation. I would like to express my gratitude to they for the useful comments, suggestions and engagement through the learning process of the master thesis.

At last, I want to say thanks to all the people who helped me especially my supervisors University Lecturer Ari Vesanen and Professor Petri Pulli.

Canrong Deng

Oulu, March 4, 2016
Abbreviations

GCM (Google Cloud Messaging) A service provided by Google to send message from server to Android, iOS or Chrome apps

UDP (User Datagram Protocol) The UDP uses a no handshaking transmission model within a minimum of protocol mechanism to deliver datagram.

TCP (Transmission Control Protocol) The TCP is a handshaking transmission model within a number of mechanisms to provides ordered, reliable, and error-checked transformation stream delivery service.

Pupil Headset (Pupil Mobile Eye Tracking Headset) A smart wearable device developed by Pupil Labs

SLS (Selective Laser Sintering) SLS is an manufacturing technique that uses high power laser to substitute plastic, metal, ceramic, or glass powders defined by a 3 dimensional model.

USB (Universal Serial Bus) It is an industry standard that defines data cables which currently managed by the USB Implementers Forum

UVC (USB video device class) It is a USB device category which capable of streaming video like still-image camera, digital camcorders, webcams, transcoders, and analog video converters, which defined by USB Implementers Forum.

AT (Assistive Technology) It includes a series of devices which helping people with disabilities. AT assisted people to accomplish tasks that they had dramatic difficulty complete.

ADL (Activity of daily living) It is a word used in healthcare stand for the things people normally do.

NAT (Network Address Translation) Because of the IPv4 address hits its limit, network address translation hides an entire IP address space behind a single IP address.
Contents

Abstract ................................................................. 2
Foreword ................................................................. 3
Abbreviations .......................................................... 4
1 Introduction ............................................................. 6
  1.1 Purpose of the study ............................................. 6
  1.2 Motivation ......................................................... 6
  1.3 Prior research .................................................... 7
  1.4 Research question ............................................... 8
  1.5 Research methods ............................................... 9
  1.6 Structure ........................................................ 10
2 Related Research ...................................................... 11
  2.1 Video streaming ............................................... 11
  2.2 Android platform ............................................. 13
  2.3 Pupil Headset .................................................. 14
  2.4 Open source software ....................................... 16
3 Implementation ......................................................... 18
  3.1 Requirements .................................................. 18
  3.2 Android Tool .................................................. 19
    3.2.1 Media streaming client ................................ 21
    3.2.2 GCM client ............................................... 23
  3.3 Web server ..................................................... 25
  3.4 OldBirds ........................................................ 28
    3.4.1 Media streaming server ................................ 28
4 Findings .............................................................. 29
5 Conclusion ............................................................ 33
References .............................................................. 35
Appendix A. Sequence diagram for registering user ............... 38
Appendix B. Sequence diagram for message sending .............. 39
Appendix C. Sequence diagram for video streaming .............. 40
Appendix D. Use case diagram ..................................... 41
1 Introduction

The population aging phenomenon is occurring all over the world due to rising life expectancy and/or declining fertility rates. In the European Union zone, the elderly (65+) will increase 75% and the very elderly (80+) will rise by 175% in 2050. However, the working-age (15-64) population will decline by 15% in next 35 years. Furthermore, the old-age dependency ratio (the number of people 65 and over relative to those between 15 and 64) of 2050 is supposed to double to 54% compared to 2014, which means the EU will have two persons of working ages for every senior citizen (Carone & Costello, 2006).

The senior would like to maintain an independent and free lifestyle (Mynatt & Rogers, 2001) and many elders would like to live on their own (Smith, 1990). Farber, Shinkle, Lynott, Fox-Grage, and Harrell (2011) reported that majority of seniors prefer aging in place which means remaining at home as long as possible even if they will need day-to-day health care service. As much as 82% of the elderly would like to still live in their current residence (Farber et al., 2011). In the United States, the rate of older adults in their 50s who live alone is about one quarter and the rate increases to one-third when considering elder people in their 60s. Furthermore, 40% of the seniors live alone in their 70s and 60% of elders live alone in their 80s and over (Donahue & John, 2014).

Due to lack of sufficient labor force and because of seniors’ own desires, modern ICT and mobile technology as the most promising substitution need to shoulder more responsibility in elderly care. Taking into account the current situation and technology barrier in tele-care of senior citizen, the system described in this thesis provides help for seniors who would like to live alone and get assistant from remote caregivers.

1.1 Purpose of the study

Senior citizens want to live alone while their safety problems are still unsolved. In the market, majority of the products provide only indoor services but not outdoor service.

Purpose of this study is to design teleguidance for elders and study the current technology that could help senior walking around the zone around their house and get necessary survival assistance. This thesis focuses on developing a safe connectivity between the remote caregiver and the senior citizen.

1.2 Motivation

Assistive technology (AT) can work as an effective supplement to informal and formal care for those who live alone or educated seniors. Besides, the most of AT was used for movement activities (walking, transferring, and going outside), and
AT often used for more than one activity. (Agree, Freedman, Cornman, Wolf, & Marcotte, 2005)

Miskelly (2001) mentioned a video-monitoring equipment which can link up to 3 cameras and start a video call to your relatives. Miskelly (2001) also wants to develop a movement detector which can show exactly where you are to your caregivers through integrated network.

This thesis illustrates a navigation system which combines Android and Unity-3d platforms together to provide safe and reliable service to senior citizens’ and their caregivers. This system can be used in the outside environment to support seniors’ movement requirements. The safety field for movement and physical activity can be set by caregivers.

The caregivers in the service center provide observation and follow up service to senior citizens through OldBirds interface. This interface provides caregivers with browsing opportunities, abilities to see and review picture and video taken by camera of senior citizen’s Android phone, map observation, and communication abilities.

\subsection{1.3 Prior research}

The idea of providing guidance service for senior citizens is not new. For instance, Strothotte, Petrie, Johnson, and Reichert (1995) developed a MoBIC travel aid to help the blind and elderly traveller independent movement by providing point guide information. The MoBIC system consists of two parts: the MoBIC PreJourney system to help user planning journeys, and the MoBIC Outdoor system to execute this plan during the journeys.

Goodman, Gray, Khammampad, and Brewster (2004) use a mobile computer which leads the senior along a route using photographs of landmarks with audio and text instructions to design a pedestrian navigation system. This system was tested by the seniors who feel their sensory, cognitive and movement abilities have been declined.

Sorri, Leinonen, and Ervasti (2011) proposed new technologies should be the right substitution of nurse. The reasons are the seniors would like to keep on their existing lifestyle and lack of enough human labor. Also, directly moving to unfamiliar environments such as hospitals or nursing homes will cause negative effect to elders (Bart, Elizabeth, & Lynne, 2006). Sorri et al. (2011) also believe that their findings can serve as a proof that seniors can benefit from high-tech applications.

Although the latest technologies may not currently be available in real life, they are being researched in the lab in many countries (Mynatt & Rogers, 2001). Firouzian et al. (2015) mentioned new indicator-based glasses which aimed at helping senior citizens who suffered from memory loss. The indicator-based glasses can receive the bit array data through Bluetooth connection from Android application. By analysis the data from the smart phone, the colors, frequency, brightness and blinking time of light in the indicator-based glasses can be set in time.

Based on Firouzian et al. (2015)'s research finding, Raappana and Korvala (2015)
developed their Open Visual Guidance System for Mobile Senior Citizen. In that system, indicator-based glasses worked as guidance tool to the senior citizen. They developed the Android application and a web server to transfer GPS location information to OldBirds part. Furthermore, they combined Linphone application to help the caregiver to give phone call from the OldBirds to smart phone.

The OldBirds is a web application that works as a communication medium between seniors and caregiver. In the last version of OldBirds, it shows the location of elderly citizen and their health condition. Besides, OldBirds is able to simulate the home, outdoor, and shopping center scene to help caregiver to take actions. Furthermore, OldBirds embedded communication tool to send text message or phone call to seniors. (Raappana & Korvala, 2015)

Pulli et al. (2012) mentioned a Mobile Augmented Teleguidance-based Safety Navigation Concept for Senior Citizens which intended to increase awareness of both, the target users and their relatives or other people in charge, of a situation of the target users, and thus to ensure a feeling of safety.

One concern about the introduction of high technology is that seniors refuse to accept new things. However, the survey pointed out to the contrary. Senior citizens would like to use AT if the benefits of using technology outweigh the costs of such use (Mitzner et al., 2010). Mitzner et al. (2010) also mentioned that focusing on the benefits of technology in education and training program can promote the adoption of caretaker technology.

According to the SeniorNet (www.seniornet.org) which provides nonprofit computer and Internet education for older adults and seniors, researchers found that senior citizens are willing and capable to learn new computer skills.

### 1.4 Research question

Intention of this system is to provide a safe and reliable guidance system for senior citizen’s outdoor movement by using modern ICT technology to provide a real-time navigation service. From this, the research questions of this thesis are shown below:

- Is it possible to provide guidance for multiple seniors through web server?
- Is it possible to support real-time streaming function for OldBirds application?

The web server is using Flask-RESTful API to design a transfer station to transport data between Android tool and OldBirds. Because the multi-user OldBirds part is designed by others which can send messages to web server, the web server is responsible for moving guidance command from web server to Android tool. The Android phone gets service from mobile virtual private network which is situated behind Network Address Translation (NAT). NAT traversal technology is required to be used in this system to transport information to Android phone.

As Figure 1 shows, the NAT device will give every individual device an independent IP address. Every phone is able to send messages to the Internet by using NAT.
device. However, the Internet needs to find a way to traverse NAT to send messages to individual phone.

![Network Address Translation structure](image)

**Figure 1.** Network Address Translation structure

With the Android platform, the Pupil headset can transfer the real-time streaming to the OldBirds side and the caregivers can send the guidance direction back to the senior’s glasses.

For the first time use, the senior needs to register and enter username, password, and Linphone account in the Android phone which will transfer these data to the server part.

### 1.5 Research methods

Design science research (DSR) is easier to be understand as a "paradigm" (in Kuhn (2012) sense) than a research method. According to Hevner and Chatterjee (2010), the designer solves human problems via creating artifacts in order to bring new knowledge to the scientific community. The key principle of DSR is to master a design problem and its answer in the process of building and application of an artifact.

Besides, other ICT research methods are regularly linked with DSR such as action research, and case studies. Indeed, there are various DSR patterns and they all follow generic design cycle proposed by Takeda, Veerkamp, and Yoshikawa (1990) in 1990.

However, I chose the model made by Peffers et al. (2006) considering the fact that it is easier to understand and use. In the first step of process, the researchers need to identify the research question and mark its priority in correct order. Later, the researchers determine the objectives which can be quantitative or qualitative for the solution (Peffers et al., 2006). In this step, the researchers are required to obtain knowledge of the state of the problems and current answer and their utility.

In the third phase, the researcher needs to design and develop the artifact that could be used to solve this problem. After that, the artifact needs to be demonstrated by using this artifact to solve the entire or a part of problem. In the fifth step, the
artifact will be evaluated by relevant metric and analysis technology. In this step, the researchers will decide whether to restart from second or third step or continue to communication step. (Peffers et al., 2006)

Eventually, the researcher publishes the whole process which requires knowledge of the disciplinary culture. The publication can either be academic or professional as the nominal structure of an empirical research papers. (Peffers et al., 2006)

The whole process does not require to run in sequential order. As Figure 2 shows, it is convenient to start from any step and move on. A problem-centered approach will start from step 1 while the object-centered solution will begin at step 2. Furthermore, the design and development centered means would be initiated from step 3.

![Figure 2. Design science research process (DSRP) model(Peffers et al., 2006)](image)

The main contribution of this research is to provide a technology assistance for senior citizens, especially for those who are easily lost. The research helps the caregivers know exactly the location of elders and their view sights. To conclude, the research creates a ”proof-of-concept” of a multiuser real time video streaming system for senior citizen’s outdoor activity.

### 1.6 Structure

The structure of this paper is following the DSRP model of Peffers et al. (2006). In the first section, this paper demonstrates the problem domain, prior research, and research methods. Next in the section 2, it illustrates the related research. In the section 3 the implementation of system is shown and in section 4 the findings are presented. In section 5 the research is concluded.
2 Related Research

This section is a narrative literature review which briefly introduces the technology that is used in this Senior Citizen Visual Guidance System.

First two subsections present video streaming and Android platform which are the third party services provided by public commercial companies. The following subsection describes Pupil Headset, commercial smart glasses sold by Pupil Labs in Germany. Because we are using third party package when developing this system, the related open source protocol will be demonstrated in the fourth subsection.

2.1 Video streaming

Jones and Schumacher (1992) introduced a system which can transfer and distribute signal through electrical lines and even got patents for that. But the system could not be used in practice due to its expensive input and weakness of performance. In the later 1980s, personal computers became popular in common family because of the cheap price and improved hardware. However, due to the lower bandwidth and diverse transfer methods, media streaming technology was not available in that time.

In the early 1990s, the idea of streaming media had a dramatic growth from a novel concept to a practical tool. This phenomenal growth happened due to the growth of video coding technologies and Internet. (Conklin, Greenbaum, Lillevold, Lippman, & Reznik, 2001)

In the late 1990s, bigger network bandwidth and access to internet emerged. Microsoft, Adobe, and Apple started to develop streaming media market and introduce their own streaming format to users.

In this research, Wowza Streaming Engine is responsible for transforming video from the Android phone to OldBirds part. As the official website mentions, Wowza Streaming Engine provides flexible and customizable service for customers. (Wowza, 2015c)

Hong, Fong, and Fong (2002) researched the Quality of service control for improving the quality of video through Internet at the receiver side. Besides Zhang, Zhu, and Zhang (2005) also studied the Quality of service control for the quality of video delivery through wireless internet connection. Furthermore, Stankovic, Hamzaoui, and Xiong (2003) developed a live video streaming system over packet network and wireless channels which have been tested to be a good and robust system.

These above studies are implemented in a Personal Computer environment while Herrero and Vuorimaa (2004) created the way to transfer video to handheld device. Herrero and Vuorimaa (2004) use Multimedia Broadcast Multicast Service and Digital Video Broadcasting - Handheld technologies to evaluate their research.
Meggers, Strang, and Park (1997) conducted a research around video streaming for mobile users. In this research, authors claimed that even in low bandwidth fixed network environment, their gateway is applicable to transfer the video to mobile users.

Playback clients and devices that work with Wowza Streaming Engine, include Microsoft Silverlight player, 3GPP mobile phones, IPTV set-top boxes (Amino, Enseo, Roku, Streamit and others), the Adobe Flash player, Apple QuickTime Player and iOS devices (iPad, iPhone, iPod Touch), and game consoles such as Wii, Xbox, and PS3. (Wowza, 2015b)

Real-time streaming uses many different protocols and variable transmitted frame rate, and therefore the arrival time can be controlled. Timing references have to be embedded in the stream in order to achieve synchronism. (Mantoro, Ayu, & Jatikusumo, 2012)

Although the Wowza Streaming Engine support Live Video Streaming, Live Transcoding, IP Camera Streaming, Audio Only Streaming, and Video on Demand, in this thesis, only the Live Video Streaming system that Wowza uses in version 4 will be depicted.

The Wowza Streaming Engine in the middle of Figure 3 works as a transfer station between the recorder and watcher. People can use either Gocoder or camera with encoder to record life event. The live video is compatible with multiple devices such as TV, Gaming, Computer, Tablet, Mobile, and Over The Top. (Wowza, 2015b)

The version 4 includes a new web-based graphical interface which interacts with the server through a REST API. This version also brings full support for MPEG-DASH and support for additional captioning formats. Besides, it could integrate with JAVA API enhancements,Wowza Streaming Cloud service, and HEVC/H.265 video over Apple HLS. (Wowza, 2015b)

The variable encoder method supported by Wowza Streaming Engine is shown in the Table 1. The live streaming depends on a continuous live stream input which means the unexpectedly disconnection and reconnection need to be taken into consideration when choosing encoding inputs. (Wowza, 2015b)
Table 1. Live Streaming Compatible: Encoding Inputs

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Video</th>
<th>Audio</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTMP</td>
<td>H.264, VP8, VP6,</td>
<td>AAC, AAC-LC, HE-AAC+, v1 &amp; v2, MP3, Opus,</td>
</tr>
<tr>
<td></td>
<td>Sorenson Spark®,</td>
<td>Vorbis</td>
</tr>
<tr>
<td></td>
<td>Screen Video v1 &amp; v2</td>
<td></td>
</tr>
<tr>
<td>RTSP/RTP</td>
<td>H.265, H.264, VP9, VP8</td>
<td>AAC, AAC-LC, HE-AAC+, v1 &amp; v2, MP3, Speex, Opus, Vorbis, Opus, Vorbis</td>
</tr>
<tr>
<td>MPEG-TS</td>
<td>H.265, H.264, VP9, VP8</td>
<td>AAC, AAC-LC, HE-AAC+, v1 &amp; v2, MP3, Opus, Vorbis, AC-3 (Dolby® Digital), E-AC-3 (Dolby Digital Plus)</td>
</tr>
<tr>
<td>ICY</td>
<td></td>
<td>AAC, AAC-LC, HE-AAC+, v1 &amp; v2, MP3</td>
</tr>
</tbody>
</table>

With service, similar capabilities can be provided by Unified Origin, Adobe Flash Media Server, Red5 Media Server and Helix Universal Server. After publishing the version 4, Wowza Streaming Engine won the 2014 Streaming Media European Readers’ Choice Award (Best Streaming Innovation) in London. (Wowza, 2015b)

2.2 Android platform

The main smart phone operating systems are: Android, IOS, Symbian, Windows Phone and Blackberry OS (Hall & Anderson, 2009). In this thesis, the operating system been chosen is Android because of the existing code and whole navigation system design.

Android is an operating system currently bought by Google in 2005, based on Linux kernel (Karch & Nickelson, 2011) and obey material design developed by Google. Android’s source code is released under open source license. The majority of Android software were under the Apache software license, Version 2.0 while there are some exceptions like linux kernel is under GNU General Public License, version2 (Android Developers, 2015d). Thus, for the applications developed in Android platform, it is legal to use it for commercial use and mods. However, you need to mention the original author’s contribution in your own documentation and keep all the rights of original authors. Besides, anyone can use its code without asking permission from the original authors according to the Apache 2.0.

The architecture of Android is as Figure 4 shows. The Linux kernel works as a low-level abstraction layer while the applications form the top layer. Developers have the same access to the framework APIs as the Android own application developers. Besides, their application can call other application directly without switching applications (Android Developers, 2011).
All the Android applications are written in Java and compiled to Java bytecode. But it will be translated into DEX bytecode in order to run in Dalvik virtual machine. (Enck, Octeau, McDaniel, & Chaudhuri, 2011)

The developers of Android application can self-publish their application to the Android market while the Apple developers have to submit their application to Apple and wait for Apple’s approval. From the architectural perspective, Android application is more safe because every application is started in a new process with an instance of Dalvik virtual machine and cannot access other application’s data without permission. However, iPhone applications can access system resource by default, which means iPhone user needs to trust Apple will completely guarantee their user data safety. (Butler, 2011)

2.3 Pupil Headset

Kassner, Patera, and Bulling (2014a) introduced Pupil because existing commercial systems are based on closed source hardware and software. They wanted to develop open source and accessible eye trackers for the academia and industry.

Figure 5 shows the view of Pupil headset when the senior wearing it. Because the frame is designed based on the 3D scan of the human head, the headset is flexible and lightweight for old citizen’s head movement (Kassner, Patera, & Bulling, 2014b).
The Pupil headset consists of three parts: frame, world camera mount, and eye camera mount. The frame of Pupil headset is made by Selective Laser Sintering (SLS) which suits better for rapid manufacture process than normal injection molding. This means that proof-of-concept can quickly provide product in actual production. (Kassner et al., 2014a)

The Pupil headset choose the USB cameras which comply to UVC standard and linked to computer platform through the High speed USB 2.0. As Kassner et al. (2014a) designed, any other UVC compliant camera can be used as substitute camera to Pupil headset. Furthermore, the Pupil headset can extend its camera by adding more UVC compliant camera as world camera mount or eye camera mount. (Kassner et al., 2014a)

In the Figure 6, the left picture is the right view of world camera mount and its connection to the frame. While the right one is eye camera mount and its connection to the frame. Both of these mounts can be adjusted to the user’s eye movements anywhere and at any time.

**Figure 5.** Front and right view of Pupil headset (Kassner, Patera, & Bulling, 2014b)

**Figure 6.** The scope of movement of the mount (Kassner, Patera, & Bulling, 2014a)
2.4 Open source software

Open source software (OSS) is a software which opens its source code under a license in which the copyright holder provides the rights to study, distribute, and change the software to anyone and for any purpose (Laurent, 2004). OSS is usually developed in a public collaborative way.

The innovative open source collaborative way produces more diverse scope of design perspective than normal company development way. A report by the Standish Group pointed out the value of OSS is about 6% of the world’s software market (Rothwell, 2008).

Considering the amazing value of OSS, the developers need to protect their copyrights from the legal aspect. The license accompanied with OSS is the weapon to protect software engineer’s copyrights. Every open source license needs to be approved by the Open Source Initiative (OSI), pass the Open Source Initiatives’s license review process.

Popular and widely used licenses are mentioned below:

- Apache License 2.0
- BSD 3-Clause "New" or "Revised" license
- BSD 2-Clause "Simplified" or "FreeBSD" license
- GNU General Public License (GPL)
- GNU Library or "Lesser" General Public License (LGPL)
- MIT license
- Mozilla Public License 2.0
- Common Development and Distribution License
- Eclipse Public License

This thesis focuses on Apache License 2.0 and GNU General Public License and will introduce them in detail. The Apache License is free license written by Apache Software Foundation. After the success of the version 1.0 and 1.1, the Apache Software Foundation published version 2 in January 2004. The purpose of Apache license is to encourage sharing of code and protect author’s copyrights. Furthermore, it is also compatible with GPL version 3 according to their official document (Apache, n.d.).

The rules one needs to follow when redistributing OSS under Apache License 2.0 are quite simple and easy. Firstly, you need to give other code holders a copy of Apache license. Furthermore, any changes in the code files need to be mentioned in notice document. Thirdly, in the derivative works that distribute your original code, all copyright, patent, trademark, and attribution notices from the Source form of the Work retained by you, excluding those notices that do not pertain to any part
of the Derivative Works. Lastly, if your work contains a notice file, any derivative works that you distribute need to include a copy of Apache License in that notice file. Even though you can change or add your copyright statement in the notice file, you cannot modify the original Apache License at all. Apache License is commercial friendly license and you can change and modify the original code to fulfill commercial or open source desire. (Apache, 2004)

As for the GNU General Public License, it is famous because it works as the copyleft license of Linux kernel. Compared with BSD and Apache License which encourage derivation of code, GPL is created to encourage code free distribution and open source. That’s why we can use variable free Linux operating system today, including commercial company’s product and personal or organizational developed software. (GPL, 2007)

When choosing GPL, the regulations are as follows. To begin with, the project needs to obtain a “notice” file about what you have edited and the exact date. Besides, the “notice” file needs to mention that it is published under GPL and “keep intact all notices”. The most important thing is once a software uses the product under GPL, then it must be open source and free, that is infectious. It is safe to use GPL derivate work as a stand-alone product but not for product which requires secrecy. If you want to put it as a commercial product, you need to accompany new and specific features with other license like BSD or Apache License 2.0. (GPL, 2007)
3 Implementation

In this section, the Multi User Support for Senior Citizen Visual Guidance System will be illustrated in detail and the main purpose is to tightly connect the caregiver using OldBirds parts with seniors wearing pupil headset. Later on it was decided that the Indicator-based glasses will be integrated to this system to support the guidance direct function.

The Visual guidance system consists of five components: Android phone, Pupil headset, Indicator-based Glasses, OldBirds and web server part. The implementation can be divided into two parts based on their installation environment. The android tool is an application installed in the Android smart phone. However the web server and OldBirds are established in the server computer. The use case diagram is as shown in Appendix D.

3.1 Requirements

Requirements of this system originate from the initial meeting with Prof. Pulli and from earlier studies Raappana and Korvala (2015); Pulli et al. (2012); Firouzian et al. (2015). The system needs to transfer Pupil headset’s camera view to the OldBirds part in real time and the caregiver can directly send guidance messages to old citizen’s Indicator-based Glasses. The system must be robust and reliable and support multi-user function. The system must not overuse the phone so that it can be used for other purpose as well. Table 2 summarizes all the requirements.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Video libstreaming</td>
<td>The system must enable to transfer the video of user’s front view to the caregiver.</td>
</tr>
<tr>
<td>2</td>
<td>Pupil headset</td>
<td>The system needs to enable Pupil headset to support video record.</td>
</tr>
<tr>
<td>3</td>
<td>Guidance Command</td>
<td>The caregiver can send direction command from the server to senior’s Indicator-based Glasses as long as the user can connect to the Internet.</td>
</tr>
<tr>
<td>4</td>
<td>Multi-user</td>
<td>One caregiver can take care of several seniors simultaneously by using this guidance system.</td>
</tr>
<tr>
<td>5</td>
<td>Indicator-based Glasses</td>
<td>The system needs to be combined with Indicator Glasses program developed by Firouzian.</td>
</tr>
<tr>
<td>6</td>
<td>Robust and reliable</td>
<td>The system must not overuse the phone so that it can be used for other purposes as well.</td>
</tr>
</tbody>
</table>
Park and Jayaraman (2003) marked five domains of requirements in the design of wearable technologies. That is wearability, durability, usability in field, maintainability, and performance metrics to support wearable technology’s functionality. This visual guidance system is designed based on their requirements of wearable technology.

The visual guidance system is made up of five parts, Indicator-based Glasses, Pupil headset, Android tool, web server, OldBirds. The Android tool including the existing Android operation system and this visual guidance system’s Android part which connects Indicator-based Glasses and Pupil headset through a USB data cable or Bluetooth. The web server connects Android Tool by UDP and GCM while linked with OldBirds through HTTP protocol. The web server consists of Python server part, GCM, and Wowza Streaming Engineering. The OldBirds is designed as a webpage that interacts with the caregiver.

Figure 7. Architecture diagram of Visual Guidance System

Figure 7 shows the architectural design of this visual guidance system and introduces the components of the artifact. The sequence diagrams of this system are listed in the Appendix A, B, and C.

3.2 Android Tool

The Android tool is an application installed in Android phone. In the Android tool implementation, there are five packages where each package provides different functions for VGSAActivity. Figure 8 is the package diagram of Android Tool. The package android.samples.com.gcmquickstart supports Google Cloud messaging and package com.semegiant.usb recognizes UVCCamera when it is attached. Besides, the package net.majorkernelpani.streaming provides services for real time streaming function and oulu.university.smartglasses handles the Indicator-based glasses.
Figure 8. Package diagram of Android Tool
### 3.2.1 Media streaming client

UVC camera is a series of devices that are capable of streaming video like still-image cameras, digital camcorders, webcams. The latest version 1.5 is described by non-profit corporation, USB Implementers Forum, in Video Class 1.5 document set.

The com.sernegiant.usb package uses libusb100.so, libuvc.so, libuvcCamera.so which are written in C. By using these three files to communicate with UVCCamera, Android tool is able to access the UVCCamera and get video buffers.

According to the Pupil Labs, the Pupil headset is not limited to use Pupil software, any other software that supports the UVC interface can be used in developing Pupil related project (Kassner et al., 2014a).

In order to realize video live streaming, the system uses the libstreaming package which also supports spydroid-ipcamera open source project. Both of them are developed by fyhertz@gmail.com and spydroid-ipcamera is published in the Google Play in free version and no ads version. Considering the fact that all of them are using Android’s own camera as capture tool, our system needs to use UVCCamera package.

The normal process of real time video transferring is video capture -> compression ->transfer -> decompression ->play. The Figure 9 shows the architecture of real time video streaming part of visual guidance system.

![Figure 9. Video streaming architecture](image)

There are three methods to encode data from the peripherals: 1) with the MediaRecorder API and a simple hack. 2) with the MediaCodec API and the buffer-to-buffer method which requires Android 4.1. 3) With the MediaCodec API and the surface-to-buffer method which requires Android 4.3.

Because the MediaRecorder API would cause some serious jitter, even though ParcelFileDescriptor tries to compensate this jitter, this method still causes problems in specific phones. Besides, the surface-to-buffer method is not compatible with UVCCamera package recording method. In the visual guidance system, I finally chose the MediaCodec API with buffer-to-buffer method.

The package methods are also diverse. The libstreaming package supports H.264 and H.263 for video while AMRNB and AAC for audio. In reality, the format didn’t influence the real time video quality.

In the libstreaming project, fyhertz provides three examples to show how to use libstreaming package in developing your own Android project. In the example 1, the Android phone works as a RTSP server by using net.majorkernelpanic.streaming.rtp.RtspServer class and wait for the libstreaming server, Wowza streaming engine, to request a stream. As shown in Figure 10, using Android phone as video capture client, user can watch real time video in PC by opening rtsp://your rtspserver ip:8086
in vlc software.

![Pull model diagram](image)

**Figure 10. Pull model**

The example 2 provides an instance of signaling the session using SDP through other protocols than RTSP protocol. Personally, I think RTSP protocol is good enough to support our requirements and ignore the second example.

The third example is as shown in Figure 11. The Android phone works as RTSP client by using net.majorkernelpanic.streaming.rtsp.RtspCilent to push streams to libstreaming server, Wowza streaming engine, through Internet. After that, user could watch realtime video in the PC through open rtsp://your wowzastreaming ip:1935/live/test.stream in vlc software.

![Push model diagram](image)

**Figure 11. Push model**

The system needs to consider the privacy of senior citizens and further development requirement about supporting multi-camera. Therefore, this system chooses example 3 and using Android phone as RTSP client.

All the Java files are stored at net.majorkernelpanic.streaming package. In the VGSActivity, one needs to set initial value of Session and RtspClient variable for using libstreaming package.

```java
//initial session
mSession = SessionBuilder.getInstance()
    .setContext(getApplicationContext())
    .setAudioEncoder(SessionBuilder.AUDIO_AAC)
    .setVideoEncoder(SessionBuilder.VIDEO_H264)
    .setSurfaceView(mSurfaceView)
    .setCallback(VGSActivity.this)
    .setCamera(mUVCCamera)
```
23

.mbuild();

//initial RtspClient
mClient = new RtspClient();
mClient.setSession(mSession);
mClient.setCallback(VGSActivity.this);

3.2.2 GCM client

Google Cloud Messaging (GCM) is a free service provided by Google Inc to help developers send data between client apps and servers. GCM provides downstream and upstream function between client apps and servers. The client apps include Android device app and iOS client app. (Android Developers, 2015a)

As shown in Figure 12, the GCM includes a client app, a GCM Connection server which is maintained by Google, an app server in your server part that links to the connection server through XMPP or HTTP protocol. A client app is enabled with GCM service by installing Google play service SDK. To downstream or upstream message, the app needs to register with GCM connection server and get an unparallel registration token. (Android Developers, 2015a)

The GCM connection server works as transfer station which accepts downstream messages from the app server and send them to a client app. Besides, the connection server support XMPP protocol can also accept upstream messages from the client app and forward them to your app server. For the app server, it has to implement HTTP or XMPP protocol for exchanging messages with GCM connection server. (Android Developers, 2015a)

One needs both a client implementation and a server implementation to realize GCM function. The GCM client needs to include at least two activities. One is registering activity that obtains registration token from Google Instance ID service. The other is receiving activity that accepts messages from web server through GCM service. (Android Developers, 2015e)

The GCM client relies on Google Play services SDK to realize GCM function. Therefore, in the main activity’s onCreate() method and its onResume() method, apps need to check appropriate Google Play services APK before using Google play services features. (Android Developers, 2015e)
private boolean checkPlayServices() {
    GoogleApiAvailability apiAvailability = GoogleApiAvailability.getInstance();
    int resultCode = apiAvailability.isGooglePlayServicesAvailable(this);
    if (resultCode != ConnectionResult.SUCCESS) {
        if (apiAvailability.isUserResolvableError(resultCode)) {
            apiAvailability.getErrorDialog(this, resultCode,
                PLAY_SERVICES_RESOLUTION_REQUEST).show();
        } else {
            Log.i(TAG, "This device is not supported.");
            finish();
        }
        return false;
    }
    return true;
}

As shown in Figure 13, the GCM client needs to register with GCM connection server before it starts working. It needs to receive registration token and send it to the web server. To get the unique registration token, the registering activity calls instanceID.gotToken(). The registration token may become invalid if

- The client app unregisters with GCM.
- The client app is uninstalled manually
• The registration token expires (for example, Google might decide to refresh registration tokens, or the APNS token has expired for iOS devices).

• The client app is updated but the new version is not configured to receive messages.

For the occasions above, the web server needs to remove existing registration token and stop using it to send messages (Android Developers, 2015b)

### 3.3 Web server

The web server works as a transfer station between Android tool and OldBirds. Requirements for the web server are as Table 3 shows.

<table>
<thead>
<tr>
<th>Function</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Register</td>
<td>UDP listen</td>
<td>The Android Tool sends user registration data to the web server.</td>
</tr>
<tr>
<td>Update registration token</td>
<td>UDP listen</td>
<td>Because the registration token will be refreshed in some conditions, web server needs to update it to keep connection reliable.</td>
</tr>
<tr>
<td>Update location</td>
<td>UDP listen</td>
<td>Get user location from Android tool</td>
</tr>
<tr>
<td>Send control information</td>
<td>PUT, POST</td>
<td>Caregiver sends control information to web server and web server sends it to certain Android tool.</td>
</tr>
<tr>
<td>Show all the locations</td>
<td>GET, GOT</td>
<td>Transfer all the users’ information like name, location to the OldBirds.</td>
</tr>
<tr>
<td>Set caregiver</td>
<td>POST, PUT</td>
<td>Specify the caregiver for users.</td>
</tr>
<tr>
<td>Caregiver registration</td>
<td>PUT, POST</td>
<td>Registering caregiver account.</td>
</tr>
</tbody>
</table>

To fulfill the above requirements, the incoming UDP messages follow the next JSON format:

1. JSON format for registering user

```json
{
  'type': 'register',
  'data': [
    {
      'name': 'value',
      'passwd': 'value',
      'sip_address': 'value',
      'token': 'value'
    }
  ]
}
```
(2) JSON format for updating registration token
{
    'type': 'update',
    'data': [
        {
            'name': 'value',
            'passwd': 'value',
            'token': 'value'
        }
    ]
}

(3) JSON format for updating location
{
    'type': 'location',
    'data': [
        {
            'name': 'value',
            'passwd': 'value',
            'latitude': 'value',
            'longitude': 'value',
            'heading': 'value'
        }
    ]
}

The PUT, POST, GET, GOT actions are supported by the Flask platform which is a lightweight Python web framework based on Werkzeug and Jinja 2 (Ronacher, n.d.). The following tiny example shows how to use Flask:

```python
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

if __name__ == "__main__":
    app.run()
```

After running this tiny example, the website http:localhost:5000/ will show "Hello World!”. And by visiting special website, user can choose PUT, POST, GET, GOT action to send or get data from that webpage.

In order to store permanent data like user and caregiver information, the service is using SQLite. SQLite is a SQL (Structured Query Language) database engine which puts its source code in the public domain(SQLite, n.d.). Figure 14 is the ER(Entity-relationship model) diagram of the database.
The User table consists of id, name, passwd, gps_latitude, gps_longitude, gps_heading, caregiver_id, registration_token, and sip_address. The id is the primary key of User table and name is the username. Besides, passwd is the passwd of user and caregiver_id is the foreign key of Caregiver table. Furthermore, gps_latitude, gps_longitude, gps_heading stand for the location of user. In addition, registration_token is obtained from Android tool and sip_address is the Linphone account.

The Caregiver table consists of id, name, pwd, sip_address. The id is the primary key of Caregiver table and name is the username. Moreover, pwd is the password of the Caregiver and sip_address is the Linphone account.

The purpose of storing sip_address in the database is to help caregivers send text messages or make phone calls to senior citizens which is achieved through Linphone application. Even though Linphone application is not used in my visual guidance system, this variable needs to be kept for further development.

In this system, I try to use HTTP method to support downstream message function which transfers data from App server. The detailed implementation is shown in “send control information” function. The GCM connection server can transport up to 4 kb of data or 2 kb notification to the client app more accurately. The GCM connection server handles all variable messages and transfers them to and from the target client app.(Android Developers, 2015a)

According to Android GCM official website, if the developer tries to use HTTP method to downstream messages, the developer needs to issue a POST request to https://gcm-http.googleapis.com/gcm/send in the below format.(Android Developers, 2015c)

```
Content-Type:application/json
Authorization:key=API_KEY
{
    "to" : registration token,
    "notification" : {
        "body" : "great match!",
        "title" : "Portugal vs. Denmark",
        "icon" : "myicon"
    },
    "data" : {
        "Nick" : "Mario",
        "Room" : "PortugalVSDenmark"
    }
}
```
The API_KEY is comes from the Google Developers Console and registration tokens are gotten from the Android tool. The Android tool will deal with these data or notification as request.

### 3.4 OldBirds

The OldBirds part is a web application which simulates real life scenes for caregiver. This web application is designed in Unity 3D combined with OpenStreetMap. (Raappana & Korvala, 2015)

OpenStreetMap is a free editable map of the world which is a volunteered project ([https://github.com/openstreetmap](https://github.com/openstreetmap)). In order to integrate with the OldBirds project, Raappana and Korvala (2015) use open source project named UnitySlippyMap written in C# by jderrough ([https://github.com/jderrough/UnitySlippyMap](https://github.com/jderrough/UnitySlippyMap)).

#### 3.4.1 Media streaming server

I did not add much code in OldBirds part only some code for video playing. So the media streaming server just provides a window for video playing and not for connections or maps.

The HTTP server can be run on any computer as long as the Wowza Media Server is visible from it, it does not have to run on the same computer as Wowza.

The index.htm page is embedded with flash player and will open a TCP connection to RTMP server of Wowza. That is to say the web page is able to receive the stream even if the browser is behind a NAT.

Therefore, in the further development, the OldBirds developers just need to add the JWPlayer to OldBirds webpage. The Unity 3D official community provides a method to illustrate how to do add JWPlayer to OldBirds webpage.
4 Findings

This section is used for illustrating the design cycle of this system and demonstrating its findings, problems and the evaluation.

The first design cycle is considering Raappana and Korvala (2015)’s master thesis and analyses my research question based on their code. In their thesis, they mentioned they were unable to migrate Pupil headset into their system due to the code below which belongs to Linephone’s Mediastreamer2. The AndroidCameraConfigurationReader5.java and AndroidCameraConfigurationReader9.java under java.src.org.linphone.mediastream.video.capture.hwconf regulate the Linphone API to only use font and back cameras of Android phone.

static public AndroidCamera[] probeCameras() {
    List<AndroidCamera> cam = new ArrayList<AndroidCamera>(Camera.getNumberOfCameras());
    for(int i=0; i<Camera.getNumberOfCameras(); i++) {
        CameraInfo info = new CameraInfo();
        Camera.getInfo(i, info);
        Camera c = Camera.open(i);
        cam.add(newAndroidCamera(i, info.facing == Camera.CameraInfo.CAMERA_FACING_FRONT, info.orientation, c.getParameters().getSupportedPreviewSizes()));
        c.release();
    }
    AndroidCamera[] result = new AndroidCamera[cam.size()];
    result = cam.toArray(result);
    return result;
}

As we can see from this snippet, Linphone sets the Android front camera as default and it is only possible to use the internal cameras of smart phones. The Linphone system is an Open source VOIP project where source code and documentation is easily available. In the beginning, I planned to rewrite the Mediastreamer2 of Linphone to let it suite for external UVC camera. While after I discussed with my supervisor, we came to the conclusion that this method is not suitable for further development. If Mediastreamer2 changed, Linphone application is not the original one but a derived work. That is to say, we need to change Medastreamer2 API every time when new Linphone version published.

Finally, by choosing a third party library to comply the Pupil headset to our software, our software can access the Pupil headset’s world camera. Because the Pupil headset’s eye camera is using non-UVC camera, the third party library is not able to access it.

In a study on eye camera project for getting course credit it was suggested to write a lower layer C library to access eye camera. However, I propose changing world camera and eye camera and only keep the Pupil headset’s frame.

Moreover, the eye camera used in this study has already lost its Infra-Red (IR) filter, so that the image sensor will be able to "see” the IR light. If we need to change
these two cameras, we need to keep attention on the IR filter which is a thin piece of coated glass.

Because our Android application uses an intent filter to discover USB devices as they’re connected, it will automatically request permission explicitly in the application before it connects to the device.

In order to get the permission to use a usb camera, Android application needs to request permission by using an intent filter. Accompanied by this intent filter, the application needs to specify the Product ID (PID) and vendor ID (VID) in the @xml/device_filter file.

The USB-IF (usb.org) is responsible for assigning VID which is a 16-bit vendor number while the Product ID is assigned by the manufacturer. The VID/PID combination must be definitely unique all over the world.

If the application uses an intent filter to discover USB devices as they are connected, it will automatically receive a permission. If not, the application needs to request permission explicitly before connecting to the device. The intent filter is as follows.

```
<activity ...>
  <intent-filter>
    <action android:name="android.hardware.usb.action.USBDEVICE_ATTACHED"/>
  </intent-filter>
  <meta-data android:name="android.hardware.usb.action.USBDEVICE_ATTACHED"
    android:resource="@xml/device_filter" />
</activity>
```

Because we use Pupil headset which is developed by a small lab who did not afford the VID fees ($5000). For the further developed consideration, it is hard to decide which brand of camera will be used to capture video. Thus, requesting permission notification in Android part is acceptable for now.

In order to access the Pupil headset’s world view camera, I use the UVCCamera package (https://github.com/saki4510t/UVCCamera) developed by t_saki@serenegiant.com. t_saki also publish his application in Google Play store named UVCStreamer.

The UVCStreamer is a popular open source code application in Google Play store. There are still some other applications like CameraFi, Webeecam that also focus on UVCCamera video recording.

After few tests, I found out that Pupil headset’s world camera does not work in all Android phones. That is to say, not all types of Android phones can detect Pupil headset’s world camera by UVCStreamer application. The original code of UVCStreamer is quite large and I deleted the extra code and only kept 4 files in com.seregiant.usb package.

In the second cycle, the problem that needed to be solved was transferring the real-time video recorded by Pupil headset to the OldBirds part. By searching the web, I found a useful application called spydroid-ipcamera which can stream the camera and microphone of your phone to your browser or to VLC. The most amazing part is the author open sources its code and provides detailed documentation in the Github (https://github.com/fyhertz/spydroid-ipcamera).
However, the spydroid-ipcamera is using Android’s own camera to design application and therefore the spydroid-ipcamera cannot directly use Pupil headset’s camera. After research on its documentation, I found out that the spydroid-ipcamera use libstreaming API to transfer real time video to another client.

Therefore, I chose to use libstreaming API in this system. Nevertheless, if a system needs libstreaming API in commercial product, fyhertz mentioned that developers need to contact him before applying it into commercial product.

The original code of spydroid-ipcamera also need to edit to suite som.seregiant.usb package’s code. In order to spit the original class and derived work, the original class is kept in sub-package while the derived work are just list under net.majorkernelpanic.streaming package.

However, my opinion is that the UVCCamera package is currently the best way to implement our video streaming. The problem of detecting UVCCamera is caused by the fact that the different Android ROMs limit USB Host and Accessory access. If this system needs to suite all the Android phones, the Android phone needs to be rooted, which seems not possible for senior citizens. But at least it works on majority of Android phones.

The authors of Kassner et al. (2014a) suggest on their wiki page, they suggest to use UVC Camera in Pupil headset. However, they use Logitech C525/C512 for the Pupil headset’s world camera and Microsoft HD-6000 for the Pupil headset’s eye camera. According to the USB Video Class Linux device driver, Logitech C525 is a standard UVC camera while Logitech C512 and Microsoft HD-6000 is non-UVC camera.

Furthermore, the Linux kernel starting at version 2.6.22 will trigger a bug in first and second generation Logitech webcams because the Linux kernel includes a USB audio bug fix.

First and second generation Logitech webcams, moreover, also face firmware bug which makes the camera somewhat unstable. The exact impact of the bug on a particular user can’t be predicted.

Since April 2014, Android device is mainly using Linux kernel version 3.4 or 3.10. If we need to change the Pupil headset’s world and eye cameras in the future, we need to think about the compatibility with Linux kernel.

In the third cycle, after trying to send information to web server, I found out that the carrier will not provide public IP address to the common users. Therefore, the web server cannot directly send data to the Android phone. In Raappana and Korvala (2015)’s thesis, they suggest to use TCP instead of UDP.

However, after test in reality, I found out it will not work. So I use GCM service to downstream message to Android phone. The code to realizing GCM function is written by myself and kept in android.samples.com.gcmquickstart package.

In the future development, the GCM service also can support upstream message to server and therefore the Android tool does not need to use UDP methods to upstream messages, which will help this system to be more reliable.
The evaluation of system was done by Park and Jayaraman (2003)’s five domains of requirements in design of wearable technologies. Because the Pupil headset and Indicator-based smart glasses are embedded in the normal glasses, the wearability is acceptable for senior citizens. The durability rely on the Pupil headset quality. In the section 2, related research, Karch and Nickelson (2011) mentioned they using SLS which dramatically flexible, durable, and light weight.

The usability in field of this system depend on the Android phone’s battery capacity. Up to now, the latest Android Nexus, Nexus 6p, contains 3450 mAh battery. That is to say Nexus 6p’s battery capacity is 2.5 times as large as that Nexus one, which is the first Nexus phone. Thus, with the hardware development, the Android phone will install higher capacity battery in the future.

Because this system consists of Android tool, web server, OldBirds which still is developing and upgrading, the maintainability of this system is guaranteed. The critical point of performance metrics of this system is the latency in video streaming. The other parts like GCM message transfer, USB connection, Bluetooth control’s delay are negligible. Because the sequence data of real time video is too large, the delay of video streaming depend on the Internet speed.

The visual guidance system built in this paper attempts to improve senior citizen’s life quality. The observations about the functionality of this visual guidance system show that it works reliably enough to some extent for safe use.

Because I implement GCM and Wowza in this system, this system can transverse NAT if needed. However, if this system needs to be applied in practice, one needs to purchase domain name or public IP address. If this system obtains a public IP address, the implementation process can be optimized.

Without comprehensive user testing, this system’s safety cannot be guaranteed, but it can be used as a “proof-of-concept” to illustrate that this kind of system can be put into practice. It must be stated that there are many aspects of the application that need to be improved, but I think that it is possible that the guidance system will eventually be used as a commercial program.
5 Conclusion

Because of the rising life expectancy and/or declining fertility rates, the population aging phenomenon happens all over the world. Besides, the seniors prefer aging in place to living with children or professional nurse. However, the safety problems of living alone in older ages are not addressed nowadays.

The purpose of this study is to provide an assistive technology to help senior citizens’ outside activity. This system can be used by caregivers for giving guidance command to elders remotely. According to the study of Mitzner et al. (2010), senior’s acceptance of AT in their daily life is higher than we expect and therefore the point of introducing AT to senior’s daily life is to guarantee application’s wearability, durability, usability in field, maintainability, and performance metrics (Park & Jayaraman, 2003).

The study is following the design science research process of Peffers et al. (2006). The expected result of design science research process is an artifact addressing problems that are related to real life problems. In this research, that is a system which can help the caregivers know exactly the location of elders and their view sight.

This paper wanted to figure out what actions developers need to take when they want to develop a system to provide guidance service for seniors in commercial way. This paper gives a brief introduction of prior research and related research which provides general knowledge about this system. In the section 3, it provides detailed demonstration about the technical implementation and requirements. The findings section shows the DSR process of this research in order.

In order to make the system safe enough for normal life use, I tested that the delay of messages and data transmission is negligible while the video transferring depends on the Internet speed. In the 4G environment, the video transferring is quite quick and effective. But in practice, the 4G environment is not stable, it will automatic change to 3G environment which supports at least 200 kbit/s. However, in the 3G environment, the delay may be obvious.

To answer the first question “Is it possible to provide guidance for multiple seniors through web server? ”, this system uses successfully GCM service to send guidance command to seniors.

To answer the second question “Is it possible to support real-time streaming function in OldBirds application? ”, this system uses Wowza streaming engine to support video streaming function. The pupil headset provides the world camera to capture video and Android tool to record video.

For the future work, I have a couple of ideas from where to start. Firstly, the part of code of Indicator-based glasses should be completed. Therefore, for the upstream message function, it can be changed to use GCM service to overcome the NAT barrier for the future development. If the further development does not want to use GCM service, the ngrok application can used to solve the NAT problem but then the UDP listen methods need to be transformed into TCP methods.
The further requirement is the system need to support two cameras: world camera and eye camera of Pupil headset, recording and transferring. Therefore, world camera and eye camera need to be connected by a USB hub.

I also recommend that someone combined Pupil headset and Indicator-based glasses in hardware layer. The Bluetooth connection consumes quite much electricity. If the Indicator-based glasses can transfer data through USB the usability of this system will be improved.

After completing all this, it would be good to test whole system with elderly citizens in order to examine this system’s functionality in senior’s normal life. It would be better to support multiple-camera of Pupil headset in the future. The system supports many cameras if needed, but it can transfer video from one camera only at a time. To conclude, this paper describes a “proof-of-concept” system to support senior citizens’ outdoor activity.
References


Appendix A. Sequence diagram for registering user
Appendix B. Sequence diagram for message sending
Appendix C. Sequence diagram for video streaming
Appendix D. Use case diagram