Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Degree programme of Computer Science and Engineering

Ville Saarinen

# Bridging Capture and Enjoyment: Removing the Transfer Phase from the Photography Workflow

Master's Thesis
Espoo, June 4, 2010

Supervisor: Prof. Tomi Männistö, Helsinki University of Technology
Instructor: Risto Sarvas, D.Sc. (Tech.), HIIT

Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Degree Programme of Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

| | |
|---|---|
| **Author:** | Ville Saarinen |
| **Title of thesis:** | Bridging Capture and Enjoyment: Removing the Transfer Phase from the Photography Workflow |
| **Date:** | June 4, 2010 **Pages:** 8 + 64 |
| **Professorship:** | Software Engineering **Code:** T-76 |
| **Supervisor:** | Prof. Tomi Männistö |
| **Instructor(s):** | Risto Sarvas, D.Sc. (Tech.) |

With the advent of digital photography and the proliferation of cell phone cameras, photography has become an everyday activity, leading to a dramatic increase in the amount of photos that people take. Even though the photo capturing technology has improved, the methods and tools with which consumers handle this increase in photo volume have not improved fast enough. Many consumers find moving photos from device to device to be unnecessarily difficult and time-consuming.

For this thesis, we built a prototype of a service using existing technologies that automated the transferral of photos from a capture device — a Nokia N900 cell phone — to an online service that the users could use to browse and share their photos. The prototype was deployed to a group of test users for a trial period. Their experiences and feedback were collected with the help of interviews.

Based on our results, we concluded that there is a demand for such a service. The prototype that we built did not have enough functionality as to being a viable commercial service, but with improvements users would generally have been willing to pay for it. Giving end-users control over their photos and building their trust are central themes that are critical for such a service.

| | |
|---|---|
| **Keywords:** | snapshot photography, mobile phone, cell phone, GPRS, 3G automatic transfer, web service, digital photography |
| **Language:** | English |

# Acknowledgements

I was very fortunate in being able to write this thesis. The topics that I touch on - photography, mobility, online consumer services - are ones that I had a passion for even before I started the project. Now that the thesis is complete, I can confidently say that my passion for these topics has grown.

I want to thank the following people: Mikko, Hanno, Markku and everyone else at Futurice who created this opportunity for me; the TUMPS team for delivering an awesome product; Jenni for the emotional support; FUTUKESÄ, Bögelit and Pöheikön Pojut for some much-needed mental time off every now and then; all of the courageous testers; Professor Männistö for agreeing to supervise this with such a tight schedule; and Taina Hyppölä for everything she did for the DI2010 project.

Finally, I'd like to give my biggest thanks to my supervisor, Risto Sarvas, without whose expertise and advice the quality of this thesis would be nowhere near the level of where it is today - not to mention ready on time.

I'm looking forward to learning by doing and by figuring things out myself from now on. 19 years of formal education feels enough to last me a lifetime. But you never know...

Espoo, June 3rd, 2010

Ville Saarinen

# Abbreviations

| | |
|---|---|
| API | Application Programming Interface |
| AWS | Amazon Web Services |
| CF | Compact Flash |
| DSLR | Digital Single-Lens Reflex camera |
| GUI | Graphical User Interface |
| HTTP | Hypertext Transfer Protocol |
| REST | Representational State Transfer |
| SD | Secure Digital |
| SMS | Short Message Service |
| TUMPS | The Ultimate Mobile Photo Sharing solution |
| UI | User Interface |
| URL | Uniform Resource Locator |

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Photography has been used to capture moments for over a century. Photos are used for both social and non-social functions: creating bonds between people, communicating with each other, reminiscing, documenting and simply for aesthetic value.

In this thesis we focus on the growing trend of snapshot photography, also known as domestic or family photography. Snapshot photography is photography that is performed by ordinary people with consumer cameras in everyday situations. Other forms of photography include art photography and photojournalism, but these will not be considered. [17]

With the advent of digital photography in the 1990s, people have no longer been limited by a financial factor (cost of film) with regards to the quantity of photos taken. Memory cards allow for the storage of hundreds or thousands of photos, and Moore's law has further reduced the price of these memory cards year after year. There is no longer a direct cost associated with each release of the shutter. This has led to an explosion in the amount of photos that are taken.

Due to this increase in the amount of photos that need to be filtered and sorted, tools have been developed to help the photographer in organizing their collection and selecting their best photos. Digital photography has also given the photographer the power to edit their photos. This power used to be only in the hands of the laboratory or machine that developed and printed the photo. By being able to fully control what happens to the photo from the moment of capture to the moment of actually consuming the photo, it is possible for the photographer to more accurately portray their intent and message to whoever views the photo.

Before digital photography, sharing was done physically. Hard copies of photos were made, and either sent to others or viewed in a social setting. The format was either single photos, a pile of photos or an organized photo album. Digital photography — and especially the internet — removed this physical constraint, since photos could be shared and viewed digitally on a computer or camera. One could instantly send a photo to the other side of the world, to relatives or friends.

Photography as a business model has also been changing over the past century and a half. Before the 20th century, when the portable camera was developed by Kodak and people were given the power of easily capturing photos themselves, the business model was to charge users for the camera, the film onto which moments were captured and the development process. Once the Polaroid instant camera was released, the user no longer needed to pay for the development process, eroding away one part of the photography business model. Digital cameras have removed yet another factor that the users need to pay for: film. Finally, cell phone cameras — by giving the end user a camera for "free" — have removed the last of the three factors that used to make up the photography business model: the camera hardware itself. [17]

Even though the traditional business model for snapshot photography has been gradually eroded away, the technological advances have opened up new business opportunities. Consumers face an ever-increasing amount of choice: what type of camera do I get, where do I store my photos, what do I do with my photos? These choices increase complexity for the end user. The photography workflow can now consist of photo capture, filtering, transferring to other platforms, editing, printing and sharing. Too much complication leaves many users unsatisfied, and many of their casual snapshots end up unutilized. My goal is to build a product or service that simplifies this process. If the product is successful, consumers will be willing to pay for it, and it might emerge as a new business model built around photography. Whether this is through directly paying for the product or by other methods is to be seen.

In a user's photography workflow, the transfer phase — the phase during which photos are transferred off of the camera — is one that could be considered a holdover from the film days. Back then, it was simply impossible not to physically transfer the photos off of the camera in order to get them processed (or, in the case of Polaroid, in order to view and share them). At the dawn of the digital photography era, this still held true: the generated photo files needed to be transferred along with the storage medium or over a physical connection. Current advances in technology, however, have allowed

us to question the need for such a phase, where the user needs to explicitly perform the transferral. Wireless technology and the internet have given way for the possibility to automate this phase, allowing the user to only spend effort in directly achieving their photographic goals.

Ubiquitous computing provides us with a vision where different types of devices can directly communicate with each other. In the case of photography, this could mean that photos are automatically transferred to some central repository that is backed up, that certain photos flow to digital picture frames, other photos are automatically shared on social networking platforms, and so on. Studies have shown that users greatly value the ability to send photos directly from device to device, without having to have an intermediate machine such as a computer [15].

After researching users' photography goals and workflow, I believe that the removal of the explicit transfer stage from the photography workflow is something that would help users in achieving their photographic goals more easily. It is also something that can be achieved with current widespread technology and could possibly lead to a service or technology that consumers would be willing to pay for.

With this thesis, I want to answer the following questions with regards to a product or service that removes the transfer phase from the photography workflow:

- What are the needed core features and functionality?
- How can such a service or product be implemented?
- What kind of business model can be built around such an implementation? Does the business model affect the implementation?

With the help of a group of students from the Helsinki University of Technology, we were able to build a prototype of a service that removes the explicit transfer phase for photos taken with a Nokia N900 cell phone. The service, TUMPS (The Ultimate Mobile Photos Sharing solution), was designed, implemented and deployed to a group of test users that tested TUMPS for about a month. Based on interviews performed after the trial period, we were able to make the following conclusions:

- The automatic transfer of photos from mobile phones is a feature that most users want.

- Building such a service is possible with current, off-the-shelf technology and infrastructure.
- The automatic transfer of photos did not significantly affect the photography behavior of users.
- Nokia N900 users do not consider the image quality provided by the cell phone camera to be good enough so as to replace ordinary digital cameras to capture all the events in their lives.
- Further development, such as a better mobile client and better ways of sharing photos online, would be required if TUMPS were to be turned into a commercial service.
- The service must ensure that the photos of users do not end up in the wrong hands and that the user is given control as to which photos are sent online.

# Chapter 2

# Snapshot Photography

## 2.1 Enjoying Photos

In order for us to help users with their casual photography by improving their photography workflow, we need to understand the reasons why people take photos. In other words, what is their ultimate purpose with the photo? What do they wish to achieve with it? If we want to build a solution that improves the workflow, that solution must also support the photographer in achieving this goal.

Research has shown that the main purposes for snapshot photography are social reasons. According to Sarvas and Frohlich, even though digital photography, the internet, and cellular phones have changed users' photography workflows, the reasons behind taking photos still remain the same [17]. Their work outlines three major *social functions*:

- *Social bonding.* The act of taking a photograph with someone can create a social bond between the two people, even if the photo is never looked at. By posing with another person, you are communicating to each other that there is some sort of bond between the two of you, that there is a reason why both of you will appear in the resulting photograph together. [17, 14]

- *Demonstrating cultural and group membership.* By taking pictures of yourself or another person in a certain context or place, you are communicating how the person in the photograph belongs in that context. Chalfen says that one is "demonstrating and adherence to appropriate models of social organization and kinship", that "a sense of belonging

and security is developed and maintained". By taking these pictures, you want to tell others (and possibly remind yourself in the future) what you value, what social group you belong to and where you are located in a cultural context. [17, 14]

- *Capturing visual histories.* Documenting certain aspects of one's life is a common reason for photography. This can happen, for example, at home, on trips or at different events. These visual histories can be used later for reminiscing, helping one remember what has happened and reliving past experiences with other people. The benefit from this type of function does not need to be tied to the future: this form of documentation can be sent across the globe to tell a story to, say, relatives. [3]

Although these social functions have existed in photography since the beginning of the 20th century, the internet has opened up new social aspects. We are no longer tied to a physical object that needs to be shared in the same physical location or sent via mail to whomever the recipient might be. One can, for example, demonstrate cultural membership by posting photos to different social networks (such as Facebook[1]), where the ensuing discussion and collective feedback with regards to the photos take place. Photos become a sort of "social object"[2], an object around which people gather and the basis for why they communicate with each other at that moment.

Photos can also have non-social or *individual functions*:

- *Personal task.* Digital cameras can be used to capture information from the real world and stored to be used later on [21, 11]. For example, one might want to take a picture of a map with their cell phone so that they can look at it later, instead of having to rely solely on their mental image. Services such as Evernote[3] are meant for storing these types of images, can can help the user organize, categorize and make the most out of these types of images.

- *Personal reflection.* During our everyday lives, we come across scenes that, in one way or another, mean something to us on a personal level. It might be a beautiful sunset that you want to save for its aesthetic value, or it might be a scene that evokes certain feelings. These photos

[1] `http://www.facebook.com/`

[2] Engeström, Jyri. *What makes a good social object*, `http://www.zengestrom.com/blog/2007/08/what-makes-a-good-social-object.html`

[3] `http://www.evernote.com/`

are stored so that they can be viewed by the photographer later on, hopefully evoking the same emotions as when the scene was captured. [21, 11]

## 2.2 The Photography Workflow

Once a photograph is taken, the photographer usually performs some other actions on the photo before it reaches its final destination. The path that the photo takes and the actions that are performed on it can be considered a "flow" through the digital photography ecosystem. This workflow isn't linear and constant — different people will have different workflows, and a single user may have multiple different workflows, depending on the type of photo that is taken or camera that is used.

Kirk[12] defines the following stages in the workflow:

- *Capture.* The moment when the photo is taken.
- *Edit on camera.* An optional phase. The user reviews photos taken during the capture phase and removes unwanted images. Unwanted images consist of poor quality photos as well as photos that another person (possibly appearing in the shot) pressures the camera owner to remove. Removing photos from cameras was also much more prevalent at the beginning of the digital photography era due to the fact that memory sizes were small, and only a limited number of photos would fit on the camera. Currently, however, storage capacity have increased so much that it is no longer a leading cause for image removal.
- *Download.* If the photo is to be taken off of the capture device, this is a mandatory phase. This phase is initiated due to a couple of reasons: the camera's memory is full and needs to be cleared, an event has taken place and the photographer wants to see the photos or the photographer needs a certain image at that moment. The downloading phase can be done with multiple different technologies, most of which are reviewed in the following section.
- *Edit on computer.* This optional step is done before photos are organized and filed. Some users like to remove unsuccessful photos and possibly perform some minor post-processing on the photos, but most of the editing is done during a later editing phase.

- *File pictures.* This phase in the workflow usually had users arranging the photos in folders on their computer hard drive according to the event or download date. Camera manufacturers (and other software vendors, such as Adobe[4]) often include software with the camera that automatically takes care of handling the photo files, giving the user better tools with which to sort and organize their photos. However, many users still perform this task manually.

- *Back-up.* This optional phase includes the copying of photos to another storage medium (such as CDs or an external hard-drive) in case the main storage location for the photos is corrupted or destroyed.

- *Pre-share edit.* This editing phase is for many the one that requires the most work. Before photos are shared, the best ones are selected and these are then optionally edited in some basic ways: rotated, cropped, red-eye removal is performed, etc. The number of photos and the amount of editing that is performed depends on how the photos are to be shared (for example, e-mail, by physical disc, social networks).

- *Printing pictures.* Another optional step. For some, the final destination for their digital photos is to produce physical versions of the photos for storage, archival and sharing. Often, filtering is performed in order to select only the best photos to print. Cost is associated with printing, so users might not want to print an excessive amount of photos.

- *Share.* The actual act of sharing the photos. This could be physically sharing printed pictures, viewing digital photos with other people in the same location or sending photos digitally over various means.

The above can be roughly grouped into three stages: capture and edit on camera form the pre-download stage; download, edit on computer, file pictures and backup form the at-download stage; and printing pictures, pre-share edit and sharing form the pre-share stage. Although they can all be performed together, users would often perform these stages separately from each other. [12]

As mentioned before, the above is only a generalization of what the photography workflow looks like for many people. However, one might just as well have their camera as the final destination for their photos. Or one might share photos directly from the capture device, as is often the case with cell phones.

[4] `http://www.adobe.com/`

Due to developments in ubiquitous computing, changes to the workflow described above are likely in the future. Users currently find certain phases, such as transfers, to be difficult [15]. If that phase is difficult enough, it may lead to a situation where users will simply skip that step. In the worst case scenario, this ends their photography workflow and results in the user not achieving their photography goal for those images.

Currently, there are considerable difficulties when transferring photos from mobile phones. A large portion of users never end up moving their photos off the phone, primarily due to the fact that the users did not know how it was done [15]. By automating this step, users would not need to learn anything new and, from their point of view, "skip" this challenging step in the workflow.

By automating the transfer to some central storage location, we would solve another problem that users often have: photos enter their personal collection from multiple different devices, and due to this, the collection often becomes fragmented. [15]

## 2.3 A Comparison of Transfer Methods

In order to achieve the desired enjoyment from photos, most often the photos need to be transferred off of the capture device. Since the beginning of the digital photography era, this phase has been a physical one: the user had to transfer the photos with the use of some type of physical media (disk, flash memory) or connect a cable between the source device and destination (USB).

With the application of newer technologies, different ways of transferring photos have been possible. Some of these, such as Bluetooth and Wi-Fi, have removed the need for a physical medium over which the photos are moved.

What many of these methods have in common is that they require explicit action from the user — a decision to actually perform the transferral from one device to another. Some technologies have automated this step, but at the cost of working only in a limited physical area (i.e. within range of a cable or line of sight), requiring demanding initial configuration or forcing changes to the photography workflow of the user.

We are going to discuss the strengths and weaknesses of these different transferral methods in order to select the most suitable one that would allow for

the removal of the explicit transfer stage with minimal initial configuration. A critical comparison will be performed based on difficulty of setup, amount of effort and time required to perform the transferral (in other words, ease of use), technology availability and working range.

### 2.3.1 Universal Serial Bus

Universal Serial Bus — most commonly referred to as USB — is a specification for wired communication between a host (usually a computer) and other devices. The original specification was introduced in 1995. Widespread usage came only after the introduction of the 1.1 version of the standard, which was released in 1998. USB 1.1 has a theoretical maximum transfer speed of 12Mbit/s. [18]

The USB 2.0 standard was introduced in 2002 and offered a 40-fold increase in maximum theoretical speed, up to 480Mbit/s. Actual transfer speeds, however, are never as high as theoretical transfer speeds, but the speed can be considered as sufficient for the transferral of photographs. [18]

USB was designed to support plug and play and be hot-swappable, meaning that using it should be as easy as plugging the cable into both devices before starting the transfer process and unplugging them once finished [18]. This makes USB easy to use, since simply plugging the capture device into the destination device is usually enough in order to initiate the transfer of photos (once configured correctly).

The maximum length of USB cables is 3 meters for USB 1.1 and 5 meters for USB 2.0 [19]. In practice, this means that the source and destination devices need to be in close proximity to each other in order for the transfer to occur.

The initial configuration required for USB is usually minimal, depending on the target device. Most operating systems (Windows[5], for example) come with built-in tools that handle the transferral with a few steps. This can also be automated, meaning that all photos are transferred as soon as the camera is plugged in.

One of the advantages of USB is that several standards related to digital photography exist that use USB as the transfer protocol. Two examples are PictBridge and Picture Transfer Protocol (PTP). PTP is a protocol aims at allowing different devices to exchange photos without the need for additional drivers. It is standardized under ISO15740 [10]. PictBridge offers the possi-

[5]Downloading pictures from your digital camera, `http://www.microsoft.com/windowsxp/using/digitalphotography/takeit/transfer.mspx`

bility to send photos directly from a digital capture device to a printer and print them [4].

Most current digital cameras and many mobile phones are equipped with USB ports, making USB a viable transfer method for these devices.

### 2.3.2 Flash memory

Flash memory is a form of permanent memory that comes in many different form factors. Flash memory is built on semiconductor chips and has no moving parts, making it less susceptible to damage due to physical movement than hard disks [2]. Practically all digital cameras and many mobile phones use removable flash memory as a storage medium, with some cameras also containing integrated, non-removable flash memory.

Flash memory that was targeted for consumer electronic devices started appearing in the mid-90s. The first removable flash memory formats included SmartMedia, CompactFlash (CF), MultiMediaCard (MMC), SecureDigital (SD) and Memory Stick. Nowadays, CF and the various different forms of SD are the most commonly used flash memory formats with regards to photography. DSLR cameras most often use either CF or SD cards, while smaller cameras use the SD format. With cell phones, even smaller forms of SD (mini SD and micro SD) are often used.

The maximum capacity of flash memory is related to the form factor. During the first half of 2010, the largest available capacity on the market was 64GB for CF cards and 32GB for SD, microSD[6] and miniSD cards (using the SDHC standard). By using a rather large estimate of 5MB per image (an approximate file size for a 10+ megapixel image), these memory cards fit about 16000 and 8000 images, respectively. It must be noted, however, that the average image size depends a lot on what type of camera is being used. Professional and advanced amateur photographers shoot their photos using RAW format, which can take up 2-3 times more space, while the cameras in cell phones are often only capable of 2-5 megapixels and thus take up less space. It is fair to say that storage capacity is no longer a bottleneck in photography.

Since flash is the ubiquitous storage media on digital photography devices, it follows that all data transfer in photography is limited by the read speed

[6] Sandisk first to ship 32 gigabyte microSDHC card, `http://www.sandisk.com/about-sandisk/press-room/press-releases/2010/2010-03-22-sandisk-first-to-ship-32-gigabyte-microsdhc-card`

of the flash memory itself, and also by the maximum speed of the transfer channel. Which one is the bottleneck? As of 2010, the maximum theoretical read speeds are:

- 720Mbit/s for CF[7]
- 240Mbit/s for SD[8]
- 80Mbit/s for miniSD and microSD [9]

The maximum theoretical throughput of some common transfer channels are:

- 480Mbit/s for USB 2.0
- 400Mbit/s for Firewire 400
- 800Mbit/s for Firewire 800

When using USB 2.0 or Firewire 400 and a high speed CF card, the transfer channel becomes the bottleneck, but in all other cases the speed of the flash memory determines how high the maximum transfer speed is. As with USB 2.0, the maximum transfer speed for flash memory is sufficient to not be a limiting factor when transferring photos.

The initial configuration that is required for flash memory is minimal and comparable to that of USB. The flash memory is attached to a reader, after which it shows up on the target device as a storage device. The photos can be copied or moved off the drive manually, or it can be done automatically with software. Most current operating systems support flash storage devices natively.

### 2.3.3 Infrared/Bluetooth

Infrared (IR) technology was among the first wireless data transfer technologies available for consumers. It uses the infrared portion of the light spectrum to transmit data. The protocols and specification used to transfer

[7] `http://sandisk.com/products/dslr/sandisk-extreme-pro-compactflash-card`

[8] `http://sandisk.com/products/dslr/sandisk-extreme-sdhc-cards-`

[9] Panasonic launches worlds first Class 10 SDHC cards, `http://www.dpreview.com/news/0905/09052102panasonicclass10sdhc.asp`

data over IR is governed by the Infrared Data Association, or IrDA[10]. IR was a standard feature on many older laptops and certain mobile phones.

IR is a technology that works best over short (under a meter) distances and requires direct line of sight between transmitter and receiver. Maximum transfer speeds depend on the IrDA category used: 115.2kbit/s for SIR (Serial Infrared), 1.152mbit/s for MIR (Medium Infrared) and 4Mbit/s for FIR (Fast Infrared). [13]

IR has mostly been supplanted by other wireless technologies in consumer devices, such as Bluetooth and Wi-Fi. USB can also be considered a better alternative to IR, since the devices need to be close to each other with IR in any case, and USB is faster, generally more reliable and is supported by more devices.

Bluetooth, which could be considered as one of the successors to IR, is an open wireless standard that uses radio waves to exchange data. Originally invented by Ericsson in 1994, the technology is currently managed by the Bluetooth Special Interest Group[11].

The range of Bluetooth depends on the class of device used. The approximate range is 100 meters for class 1, 10 meters for class 2 and a meter for class 3. The maximum transfer speed depends on the version of the Bluetooth specification used. Bluetooth v1.2 has a maximum speed of 1 Mbit/s, v2.0 + Enhanced Data Rate (EDR, released in 2004) has a maximum speed of 3 Mbit/s and v3.0 + EDR (released in 2009) has a maximum speed of 24 Mbit/s [6]. Compared to wired connections (USB and flash memory), Bluetooth (and especially Infrared) are a lot slower. Using the maximum realistic transfer speed for Bluetooth v2.0 — currently the most prevalent version available — it could take up to 20 seconds to transfer a single 4 megabyte photo. With tens or hundreds of photos, this transfer time adds up and becomes a limitation of the transfer method.

Bluetooth support was officially added to Windows with Windows XP Service Pack 2 in 2004.[12] On the Mac, it has been supported since Mac OS X 10.2, released in 2002.[13] Using Bluetooth before these releases required the use of third party drivers, making the use of Bluetooth somewhat more difficult than, say, USB.

[10] `http://www.irda.org/`

[11] `https://www.bluetooth.org/`

[12] Bluetooth Wireless Technology FAQ - 2010, `http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/Bth_FAQ.docx`

[13] Apple Introduces "Jaguar", the Next Major Release of Mac OS X, `http://www.apple.com/pr/library/2002/jul/17jaguar.html`

Another reason why Bluetooth is sometimes considered hard to use is the fact that it uses rather stringent security measures [20]. Before devices can exchange data, they need to be paired, which is not always a straightforward process. Once paired, however, the exchange of data is usually a simple process. Some software, such as ProximitySync[14], even allow the transferral process to start automatically when the devices come within range of each other.

Most current laptops and cell phones have built-in support for Bluetooth, along with some desktop computers and a few digital cameras. For digital cameras and computers that do not support Bluetooth, adapters can be added which add support for it.

### 2.3.4 Wi-Fi

Wi-Fi is a term that describes products that use the IEEE 802.11 standard for wireless local area network (WLAN) communication. In order for a device to be able to be labeled as Wi-Fi compatible, it needs to conform to the IEEE standard and be certified by the Wi-Fi Alliance. The technology behind Wi-Fi was developed during the 1990's, and the non-profit Alliance was formed in 1999 in order to establish and enforce standards for interoperability between device manufacturers.[15]

Compared to Bluetooth, Wi-Fi has a much larger operating distance. The IEEE 802.11 standard defines many different versions of the standard, of which 802.11a, 802.11b, 802.11g and recently standardized 802.11n are most commonly used. The operating distance and speed depend on which version is used. 802.11b and 802.11g, currently the most commonly used ones, use the unregulated 2,4GHz frequency and have a range of about 30 meters when indoors and about 90 meters outside. This distance can be dramatically increased with specialized antennas. 802.11a, which uses the 5GHz range, has a slightly shorter operating distance. 802.11n, the newest version of the standard, uses new techniques to further extend the operational distance.

The maximum theoretical data throughput for each version of the standard can be found in Table 2.1. [5]

These speeds can be considered fast enough as to not hinder the user experience when transferring photos, especially when using 802.11g or faster connections. With a 54Mbit/s connection, a 4 megabyte photo will be trans-

[14] `http://www.braddolman.com/proximitysync/ProximitySync.html`
[15] Wi-Fi Alliance, `http://www.wi-fi.org/organization.php`

Table 2.1: Maximum Throughput of Different 802.11 Versions

| Version | Speed |
|---|---|
| 802.11a | 54 Mbit/s |
| 802.11b | 11 Mbit/s |
| 802.11g | 54 Mbit/s |
| 802.11n | 600 Mbit/s |

ferred in less than a second.

Practically all current laptops and many cell phones support Wi-Fi. However, few digital cameras support it without additional adapters. An example of such an adapter is the Eye-Fi[16], which functions similarly to a SD memory card but also has a built-in Wi-Fi chipset and antenna. It allows the user to take pictures onto the included memory, and as soon as it enters within range of a known Wi-Fi network, it transfers the photos to either a computer or different online services, automating the transfer phase. Professional photographers that use DSLRs also have different accessories that send photos over a Wi-Fi network as soon as they are captured (such as the Canon WFT-E5[17]).

One requirement for Wi-Fi usage is the personal Wi-Fi infrastructure that the user needs to setup. A Wi-Fi access point needs to be installed, and the computer (or other destination device) needs to be accessible from the wirless network. Setting up the software on the destination device can also be a non-trivial task.

### 2.3.5 GPRS/3G

All the previously discussed data transfer channels are under the control of the end user. By this I mean that the transferral has been possible over a connection that is private to the user, and goes directly from the capture device to the target device (possibly over a private network). General Packet Radio Service (GPRS) is the first of these channels that performs the transferral over a public network: first over the mobile operator's network, then possibly over the internet.

Before the introduction of GPRS, data communication over a cellular GSM network was achieved with circuit switched data calls. Users were billed based on how long they were connected. GPRS, which is a packet switched

[16] `http://www.eye.fi/`

[17] `http://cpn.canon-europe.com/content/product/accessories/wft_e5.do`

service, can be considered "always on" and users are charged based on the amount of data that they transfer.

GPRS technology falls under the GSM 2.5G (between second and third generation) category. The maximum transfer speed depends on the location of the user and the quality of the mobile network, but ranges from 32.0 to 80.0 kbit/s (8.0 to 20.0 kbit/s upstream). EDGE (Enhanced Data rates for GSM Evolution), a GSM 3G technology, increases this speed to 236.8 kbit/s (59.2 kbit/s upstream), approximately tripling the bandwidth when compared to GPRS.

3G is a family of standards that, in addition to including EDGE, includes UMTS, which is often synonymous with "3G" in Europe. Maximum UMTS throughput is 384 kbit/s with R99 devices, 14.4 Mbit/s (5.76 Mbit/s up) with HSPA and 56 Mbit/s (22Mbit/s up) when using HSPA+ [1, 7]. The maximum speed drops if the user is in motion. While 3G and especially GPRS are significantly slower than Wi-Fi, since users are not tied to a physical location during the transfer process, the slower speed does not matter quite as much. The transferral can happen while users move about freely.

Except for some remote areas, GPRS has a large coverage of populated areas globally, and reaches across most of Europe, for example. 3G coverage is concentrated mostly around the most densely populated areas, such as cities. EDGE coverage falls in between 3G and GPRS.[18]

In addition to using a public network, GPRS/3G differs from the previous communication channels in that using it costs something. When the use of GPRS and 3G became popular, mobile operators charged by the kilobyte or had prepaid packages of a certain number of megabytes. Currently, most operators offer an "unlimited" package, charging the user for a flat rate (usually between 10-30 euros per month) that allows them to use the data connection as much as they want. In practice, many operators set a limit (such as 5 gigabytes) to how much data can actually be transferred with the "unlimited" data plan.

Practically all current cell phones sold have GPRS capabilities. Most mid- to high-end phones also have 3G capabilities, making data transfer over mobile networks available to just about everyone with a mobile phone. With the current proliferation of unlimited data connections, an increasing amount of people utilize the internet capabilities of their mobile phones, including

[18] GSM world coverage map, Edition 2009/A, `http://www.coveragemaps.com/gsmposter_world.htm`

sharing photos. Many popular services, such as Facebook[19] and Flickr[20], make the uploading of photos a simple process, allowing users to easily send photos from anywhere. However, all of these still require explicit actions from the user.

Unlike Wi-Fi, the user does not need to install and administer the infrastructure that GPRS needs. However, GPRS settings need to be configured in the cell phone before it can be used, but many operators have the ability to automatically send or install these settings onto the phone of the user. Once properly configured, the internet connection is always available when the user is within the operator coverage area. After this, the ability to transfer photos is dependent on the software and built-in features on the mobile phone as well as the online service where the photos are sent to. For some services (such as the N900's built-in Flickr and Facebook sharing), setting it up is trivial, but some software or services have a difficult setup process, requiring software installation and configuration.

### 2.3.6 Summary

Technological advances with regards to transferring photos off of the capture device have allowed for a faster and easier transfer phase. What used to require a wired connection, possibly complicated explicit user actions and take many seconds per image can currently be performed wirelessly, with minimal user interaction and at the speed of multiple photos per second. Although the size of photos has also increased, the maximum throughput of the different available transfer methods has increased at an even greater rate, leading to a reduction in the required time and effort when moving photos off of the capture device.

We have mostly only mentioned the theoretical maximum transfer speeds of the different methods. In practice, the maximum speed is rarely (if ever) reached due to interference and protocol overhead.

When comparing the speed of the most current and prevalent transfer methods (USB 2, flash memory readers, Wi-Fi, Bluetooth 2.0 and 3G), only Bluetooth and 3G can be considered slow enough as to make a noticeable difference to the user experience. Since Bluetooth requires the capture device to be in relatively close proximity to the destination device, a better option would most probably be a faster wired connection (such as USB 2.0). Bluetooth is better suited for either low-bandwidth devices (e.g. mouse and keyboard) or

[19] `http://www.facebook.com/`
[20] `http://www.flickr.com/`

Table 2.2: Summary of Transfer Methods

| Method | Max. Range | Speed | Initial Setup | Transferral | Notes |
|---|---|---|---|---|---|
| USB | 5 meters | Fast | Minimal | Plug in, automatic | Tangible |
| Flash memory | 5 meters (USB) | Fast | Minimal | Plug in, automatic | Requires flash memory reader, tangible |
| Bluetooth | 10 meters | Slow | Setup and pairing | Automatic with software | Supported by few cameras |
| Wi-Fi | 30 meters | Medium | Infrastructure and setup | Automatic with e.g. Eye-Fi + software | Requires home Wi-Fi network |
| GPRS/3G | Operator coverage | Slow | Software installation | Manual | Cell phone only |
| Thesis prototype (3G/Wi-Fi) | Operator coverage | Slow (3G) + Medium (Wi-Fi), constant transfer | Minimal | Automatic | Depends on a third party service, cell phone only |

ones that greatly benefit from being wireless (e.g. headsets). 3G connections, on the other hand, are not tied to a specific location and work in most urban areas, with a satisfactory fallback to EDGE or GPRS. This means that even though the transferral of (especially large) photos can take some time, the user does not need to wait in one location while the transferral occurs. Since the connection is often constantly open, we can perform the image transfer automatically right after capture, if needed.

The downside of 3G is that using the connection incurs a cost for the user as well as using more power, which can cause a significant drain on the device's batteries. However, especially due to the increasing use of smartphones that require a data connection in order to be fully functional, consumers are buying an increasing amount of mobile data packages that allow them to transfer large amounts of data with a constant monthly fee.

Although the 3G connection is available for cell phone cameras, dedicated digital cameras currently cannot connect to a 3G network. However, with products such as the Eye-Fi becoming more and more popular, it is likely that a product that enables a 3G connection with an SD or CF card will be developed in the future.

If the goal is to create a system where no action at all is required from a user in order to transfer photos from a capture device, wired connections can be ruled out since they require the user to connect the wire between the devices. The upside of wired connections — especially USB 2.0 — is that USB connectors are ubiquitous on capture devices and computers, require very little initial setup and when transferring photos using it, users get a tangible "feel" for the process: they are connecting the wire in order to accomplish one thing and know exactly what is going to happen. Compared to wireless, where the definitive point at which photos are transferred may be uncertain, user expectations are more easily matched when using wired connections. This is why having the option to use a wired connection is a plus, even if the user might have some sort of wireless transfer option.

Wireless connections, on the other hand, only require the user to be within range before the connection can be automatically started. However, current systems require a significant setup effort and possibly the installation and administration of an infrastructure, as is the case with a personal Wi-Fi network. If the Wi-Fi network already exists and is configured correctly, Eye-Fi is an example of a product that makes the transferral of photos relatively painless after an initial setup.

If we want to remove the transfer phase regardless of the location of the user, however, then a 3G or GPRS connection is the best choice. Currently, no widespread services or solutions exist that use this communication channel to automatically send photos from the capture device.

Based on these choices, we decided to implement a service that utilizes GPRS or 3G to automatically transfer photos when the user is not within reach of a Wi-Fi network with internet access, but switches over to Wi-Fi once possible. This gives us maximum available bandwidth without tying the user to any location, while still giving us the ability to transfer photos practically as soon as they are taken. It is also important to make the initial setup process to be as painless and smooth as possible, allowing even the most non-technical people to be able to start using it. If setting up the service is too difficult, the potential users will simply give up and ditch the thought of using it.

# Chapter 3

# Research Question and Methodology

## 3.1 Research Objectives

With the proliferation of cell phone cameras, many users are experiencing a problem, especially once they replace their cell phone with a newer one: most of the photos taken with the camera are found only on the cell phone. This is partly due to the difficulty of transferring photos off of the cell phone, leading to users either never transferring their photos elsewhere or doing it on rare occasions. [15]

With this thesis, I want to build and validate a solution for this problem. The solution should be user-centric — that is, it should be easy to take into use, should require minimal or no effort to use and should allow the user to accomplish what they want to achieve with their photos. This would allow the target user group to be as extensive as possible and would not be limited to technologically-oriented users who are prepared to spend more effort in setting up and using it. The target user group is an "average" consumer, one who takes photos in order to document their life and to capture unique, one-off situations.

Even though this solution focuses on cell phone cameras, the solution will hopefully be general enough so that it can be expanded to include photos taken with normal digital cameras in the future.

This solution will be validated by building a prototype that will be tested by a group of users. The experiences, usage data and feelings will be collected (with the help of themed interviews) and analyzed, resulting in a list of

improvements that can be made to the prototype.

One way for the solution to become widespread and reach as many users as possible is by commercializing it. In order for this to happen, a viable business plan must exist. Another objective for this thesis is to try to find and validate (with the help of the test users) such a business model, which can be used to turn the solution into an actual consumer product down the road.

## 3.2 Research Question

With this thesis, I want to answer the following questions:

- What are the needed core features and functionality of a service or product that removes the explicit transfer phase of a user's photography workflow?
- How can such a service or product be implemented?
- What kind of business model can be built around such an implementation? Does the business model affect the implementation?

## 3.3 Contribution of Research

This thesis will contribute the following to the field of photography and digital service creation:

- Validation of the hypothesis that the removal of the explicit transfer phase helps casual photography.
- The design, implementation and validation of a prototype that demonstrates one way of achieving the above.
- Suggestions as to how to improve upon the prototype so as to better achieve its goal.
- Suggestions on how the prototype could be turned into a profitable service.

## 3.4 Research Methodology

### 3.4.1 Previous Work

Previous work will be studied in order to gather data about people's photography habits: why they take photos, what their photography workflow is like and what they want to accomplish with their photos. This should give us a general hypothesis whether removing the transfer phase will benefit users. Understanding the reasons for taking photos should also help us in selecting features that are needed in the prototype that we wish to develop, implement and test. Research material is selected from multiple fields: human-computer interaction, social sciences, information technology and information sciences.

A design-science in information systems[8] based approach will be used to test this hypothesis. A prototype of an actual product will be developed iteratively, deployed to test users, used in realistic conditions and results from actual usage will be collected and analyzed.

Our research will also help us in selecting the most appropriate technology and architecture with which to build the prototype.

### 3.4.2 Prototype Design and Implementation

Based on our research, we will design and implement a prototype for a service that removes the transfer phase from the casual photography workflow. The implementation will be done by a group of students from the Helsinki University of Technology (Aalto University) for Futurice, a digital service development company based in Helsinki.

The goal of the implementation is to create a prototype that is advanced enough so as to be usable by a group of test users. The prototype should have just the basic features so that the users can automatically transfer their photos as well as do rudimentary tasks with them. The selection of features for the prototype will depend on exploratory prototype construction — that is to say, the prototype will be developed in iterations, with usage testing done concurrently, and feature selection (which is done between iterations) will be guided by the focused main overall goals as well as experiences gathered from the unfinished prototype usage.

### 3.4.3 User Testing

Once the service prototype is good enough, it will be deployed to a group of test users. For the user testing, we want the testing to be as realistic as possible — we want the prototype to be used by people who use their camera phone as their main, everyday phone and take pictures with it. The service will be installed on their phones and they will use their phone normally for a certain amount of time, after which the users will be interviewed about the experience.

Hyysalo says that one can have structured, themed or open interviews [9]. Structured interviews have a rigid set of preselected questions, while open interviews are more akin to discussions about the selected topic. Themed interviews are in between, with preselected questions but the interviewer can ask additional questions based on answers or comments. The more open an interview is, the harder it is to analyze answers, since extracting data from the unstructured answers is more difficult. It is also harder to compare answers from different people if the questions are not exactly the same.

For our interviews, themed interviews will be used. This will allow us to react to the answers and opinions of the interviewees, which could bring to light new factors that hadn't been thought of beforehand. It will also allow for expanding on a subject, if so desired.

To further be able to compare the actual behavior of testers before and after taking the service into use, an online questionnaire will be given to all testers. The goal of the questionnaire is to gather information about the testers' photo sharing behavior before starting to use the prototype. The questionnaire will be made up mostly of multiple choice answers, giving us quantitative data with which to compare the users.

# Chapter 4

# Results

## 4.1 TUMPS System

TUMPS, or The Ultimate Mobile Photo Sharing solution, is a project that was done on behalf of Futurice by a group of nine students from the Helsinki University of Technology for the course T-76.4115 - SW Development Project. Futurice is a software solution provider that specializes in digital web and mobile service creation. The project lasted close to six months, during which the students planned and implemented the service iteratively.

With the project, we wanted to create a service that would move all of the photos taken with a cell phone camera online to a web interface with which the user would be able to organize, view and share the photos. Taking the service into use should be easy enough that even non-technical users could achieve it.

The student group consisted of six developers and three "software experts": a project manager, a quality assurance manager and a software architect. The developers were roughly divided into three pairs: one pair was responsible for the server, one for the web user interface and one for the mobile client. However, these divisions weren't to be definitive — the team members helped each other, if needed. I acted as the client or product owner, giving feedback to the group, setting the project priorities and deciding what features should be implemented.

### 4.1.1 Project Goal

The goal of the project was simplified to a prioritized list of three requirements:

1. The service has to automatically transfer all photos taken with a cell phone to the web service, where selected photos can be shared to at least one social service.
2. The installation process needs to be as simple as possible.
3. Using the web UI must be intuitive and users should be able to achieve their photographic goal with it (within the service's capabilities).

This prioritized list was clearly communicated to the team along with the reasoning behind it. It was important that they understand the "why?" of the project, since it allowed them more power as individuals when deciding on what or how to implement something. Understanding the purpose of the project would hopefully also motivate the team members to produce high quality work.

The most important requirement was the main goal of the project: to remove the need to explicitly transfer photos away from the capture device. We wanted the user to not have to change their behavior in any way when taking photos, which could only be achieved by silently transferring all the photos automatically. There also needed to be some sort of way that the user could achieve what they set out to do with capturing these photos, meaning that simple organizing, selection and sharing tools needed to be available.

The simplicity of the initial install phase of many consumer services is one aspect that is often overlooked. Every action that the user needs to perform at this point is a potential step where the user can decide that the installation is not worth the effort, causing them to abort the installation process and leading to one less user of the service. We wanted to make the installation of TUMPS as simple as possible, with very few steps. Since TUMPS automates the transfer phase for the user by automatically transferring photos in the background, once installed, very little effort is required to continue using the service.

Figure 4.1 is a theoretical portrayal of the target effort needed to use TUMPS when compared to manually transferring photos. Initially, some effort is required in order to install TUMPS, but after that, all the user needs to focus on is the selection and sharing of photos. Compare this to the current

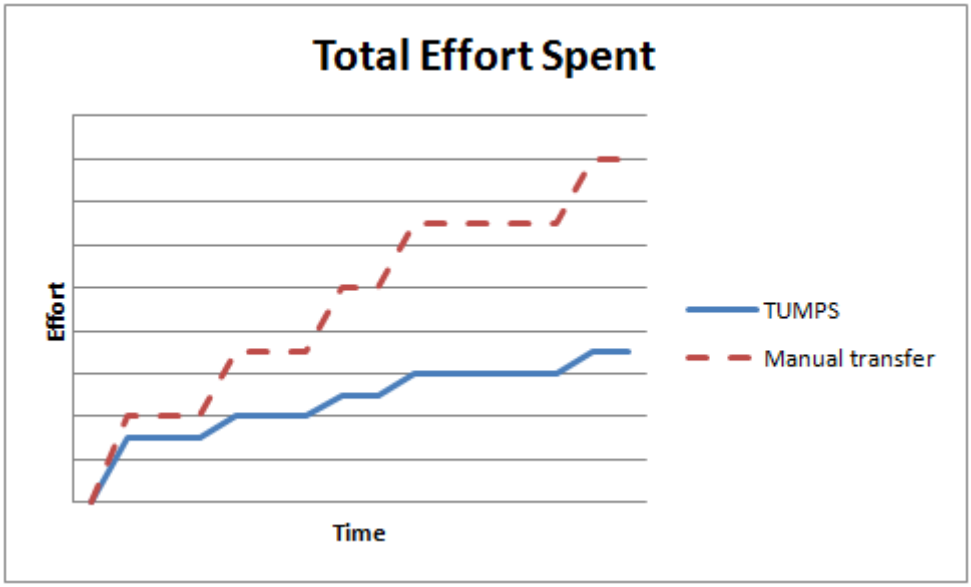


Figure 4.1: Total effort spent in transferring, selecting and sharing photos

de facto, where the user needs to perform the transferral in addition to the selection and sharing phases.

The third goal of the project was a high-caliber user experience with regards to the web service. As with any consumer web service, usability is extremely important. However, the main focus of this project was removing the photography transfer phase, not duplicating yet another photo sharing website. Being able to organize and share (and thus enjoy) photos depends partly on a usable web experience, but without a working end to end implementation of the primary goal, it has no use. In addition, the web interface is a component of the whole service that can be improved or overhauled afterwards, if the whole concept is deemed useful.

Some non-functional goals were also set for the implementation team. One was that we wanted to use new technologies, on the mobile, web and development side. One reason for this was because if a TUMPS-like service is ever commercialized, it will take some time before it is available to the public, during which the currently high-end phones will most likely be more common within the general public. With technological advances and Moore's law, the prices of smart-phones (i.e. advanced phones that allows one to surf the net, run applications and do a multitude of other tasks in addition to using it as a phone) will continue to drop, at some time passing the price barrier that will allow it to enter the realm of the average consumer.

Another reason for choosing the newest technologies, especially with regards to the web and development side, was that since the project didn't have high market or customer pressure (owing to not having a commercial client), Futurice was able to use the project as a sort of test-bed for new technologies with low risk. Many development frameworks or environments claim to improve productivity with the help of new tools, infrastructure, programming languages and libraries, and we wanted to put these claims to a test. This increase in productivity would also hugely improve the probability of success for the project, since the resources (amount of work hours) available was only on average about one month of work per team member. In practice, since the course required some mandatory non-development work and most of the technologies required some research and studying, the amount of actual productive work time was quite a bit less.

Four out of the nine TUMPS team members were not Futurice employees. Due to not wanting to unnecessarily set up access to the company intranet for outsiders, using a cloud-hosted development environment (version control from Bitbucket, virtual servers from Amazon Web Services) were among these new technologies used. This gave an added benefit that the team could develop the service from any location, in addition to being in control of the services themselves.

### 4.1.2 Feature Planning

#### Mobile

As a mobile platform, we ended up choosing the Nokia N900. At the time, it had been just released to the public, and was Nokia's flagship smartphone. For a mobile phone, the Linux-based operating system, Maemo, was brand new and could be seen as Nokia's next generation mobile phone OS. It also allowed Linux-like development, meaning that applications could be developed using a familiar programming language (C++) instead of an esoteric, proprietary programming language. Since the OS was based on Linux, it meant that application developers could easily install their own software and even modify the operating system, if need be. The downside of choosing the N900 was that few people — and thus potential testers — owned it, and since it was still a rather technical device, the target market of the phone was more technically-oriented than the average consumer.

For the mobile client, we decided early on that we would not develop a graphical user interface. Once the software is installed, it works in the background

and transfers all of the photos that were taken on the device, including all photos that existed beforehand. Omitting the GUI served multiple purposes. First of all, the we wanted to minimize the risk that we would not succeed in creating a functional mobile client by removing all non-essential features. Developing for a mobile platform — especially one that the development team members had no experience in — is more generally more difficult that developing web services with current technologies. We wanted to make sure that the primary goal of project was achieved as reliably as possible from the mobile device point of view (i.e. transferring all the photos automatically off of the device with no user intervention). If, during the project, we achieve this with time to spare, the addition of a GUI could be considered.

The second purpose of removing the GUI was that it possibly made the end-user focus (or in this case, remove the focus) on the fact that truly no actions are required on their part in order to move the photos off of the capture device. Users are used to having to open software or perform some action to do this, so removing the ability to perform any actions should reinforce the idea of the primary goal of the project in the minds of the user. Users should ultimately forget that some sort of transferral happens in the background. Instead of thinking of the cell phone memory as a separate storage space from the online service, it transforms the web UI into a "window" through which the user sees the contents of their mobile phone image folder.

The downside of not having a GUI on the mobile client is that users would not get feedback on whether the client was working or not, and if all the photos had been sent. In addition to providing feedback for the user, the GUI could have received input from them as well, allowing the possibility to control the TUMPS client somehow. For example, a toggle that turns syncing on and off would have been possible.

When the actual transferral of photos takes place is something that needed to be decided. The team was not sure if it was possible to somehow hook into the phone's photo application in such a way that it notifies TUMPS whenever a photo is taken, so the approach that was decided upon is that the software scans the location where photos are stored at certain intervals. Whenever it finds new photos, it uploads them in a batch. An interval of five minutes for these checks was decided upon, since it shouldn't use up system resources very much while still retaining the "nearly instantaneous transferral" target of the service.

One fact that we needed to take into consideration was that users might travel abroad, meaning that either the cell phone had to be able to disable data roaming (in order to prevent substantial roaming data charges) or the

software had to be able to support it.

### Web

For the web UI, feature selection and design was accomplished through user stories. I gave the team a couple of stories that needed to be somehow accomplished by the service, and they had free reign on how the user interface would achieve it and how it was implemented. The main user story follows.

> During a week, the user takes a couple of pictures at school on Monday. From Wednesday to Thursday, the user is out on a trip during which lots of photos are taken. Finally, on Friday, the user logs in to the web UI for the first time that week. The user should be able to easily browse through the couple of photos that were taken at school on Monday. The trip photos should also be somehow grouped together so that they can be viewed as one batch or event. The user then decides to share all of the trip photos except for two on Facebook, as well as selecting only the top three photos for sharing with Flickr.

Based on the user stories, the team initially drafted a wireframe for the web UI.

Once the wireframe was reviewed and accepted, the team started making more realistic drafts of the UI that included graphical elements.

Carefully reviewing each UI draft was very important to the success of the project. As can be seen from the first draft (Figure 4.3), the team had included many small features to the UI that were not directly related to the user stories and, more importantly, the main goals of the project. These seemingly small features (such as photo titles, editing and tagging) would have required a lot of work, and would have been distractions from the most important work required in order for the project to succeed. The drafts were discussed and multiple iterations were made until I was happy enough with them.

After the fourth UI draft, the web interface was good enough for me and I told the team that they could implement the service with the fourth draft as a guideline. I believed it catered to the main user story and would allow the required actions to be performed with one omission: at this point in the project we wanted to be on the safe side and make sure that sharing photos to the most popular social networking site, Facebook, worked fully. Only

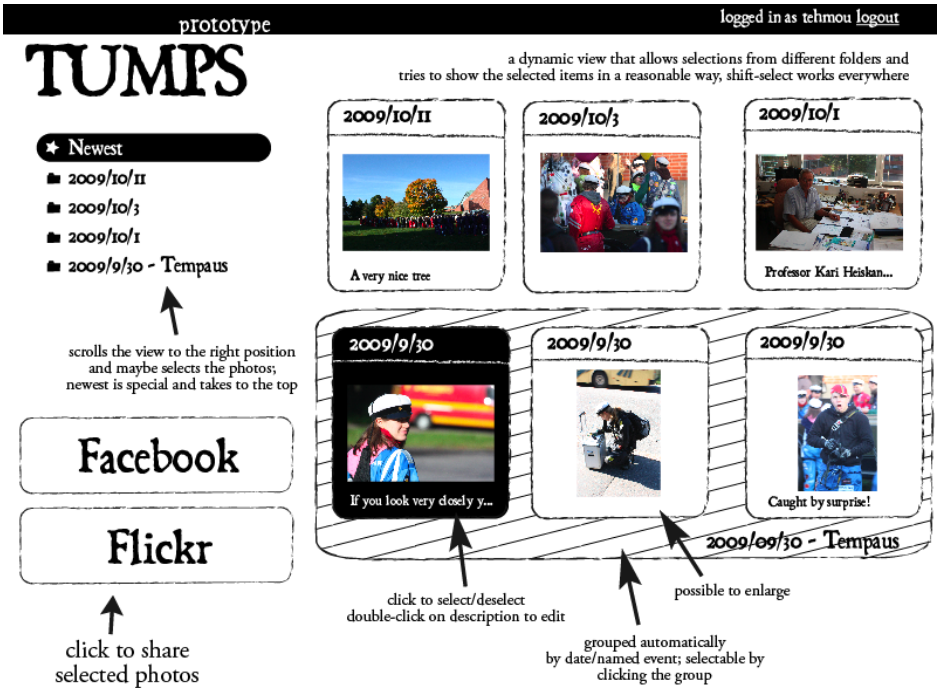


Figure 4.2: Web UI wireframe

after this worked would we add support for other services such as Flickr. This can be seen from the missing Flickr sharing button in the fourth UI draft in Figure 4.4.

Going through the different draft versions with the team members proved to be more important than I initially thought. It was extremely easy at this early stage in the project to add features, and we needed to constantly remind ourselves of the primary goals of the project. Whenever something new was added, we needed to ask ourselves: "Does this contribute directly to our goal? Can we achieve the goal without it?" Moving as much as possible out of scope was important. With such a limited budget, we had to build a service that really focused on one thing and one thing only, but did it extremely well.

The web UI for the installation process for TUMPS was quite a bit simpler than the rest of the web UI. For this, we again used a user story:

> The user opens the installation page of TUMPS with a browser. They enter their mobile phone number and press the Submit button, after which their mobile phone receives a message with an installation link and their password for the web UI. Upon open-

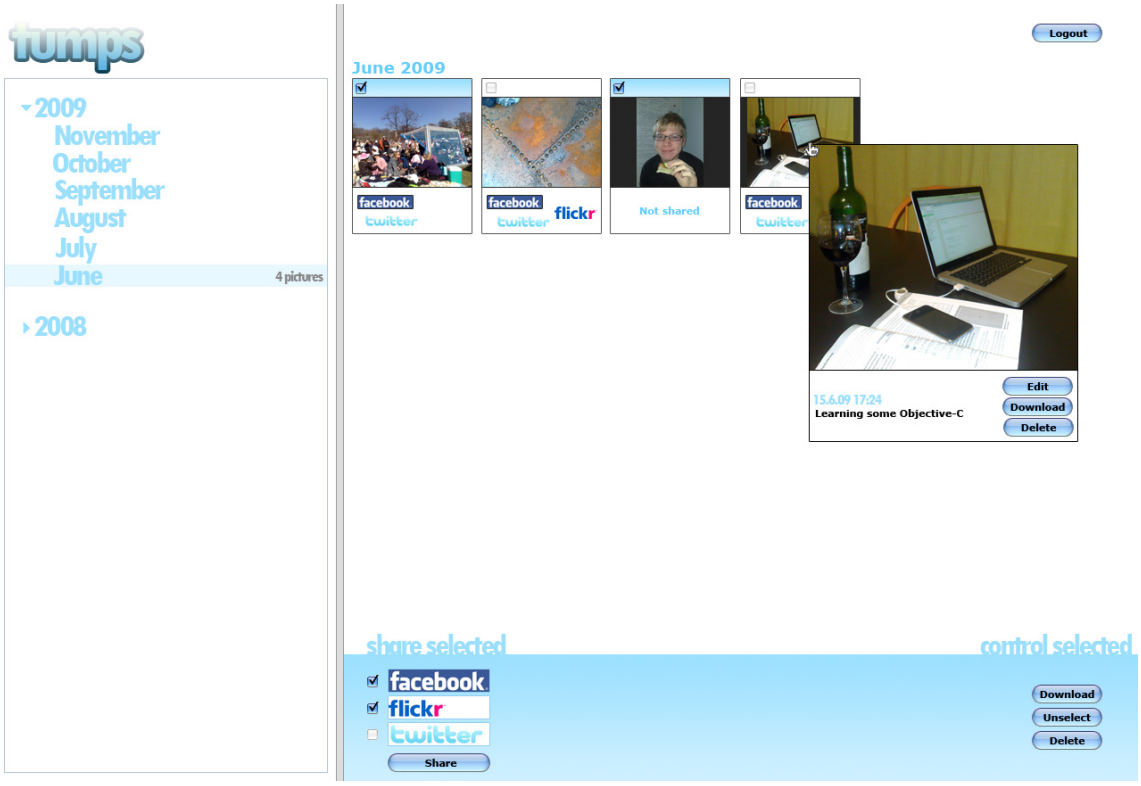


Figure 4.3: First draft of the web UI

> ing the link, the user is prompted to install the software. After agreeing to this, the software is installed and started, and no more actions are required from the user. All past and future photographs are transferred to the web service.

Implementing this required overcoming some technical hurdles (SMS sending, user-customized installation package), but from a GUI point of view, the only part that needed to be implemented was the web page where the user enters their phone number. The design of the page was made by the team and I supplied the instruction text for it.

I believe that by limiting the target of the project to three relatively simple goals and by showing how these goals help the end-user through user stories, we really helped team members understand the reason why such a service was being built. It also improved their ability to make decisions for themselves with regards to feature details and backlog management.

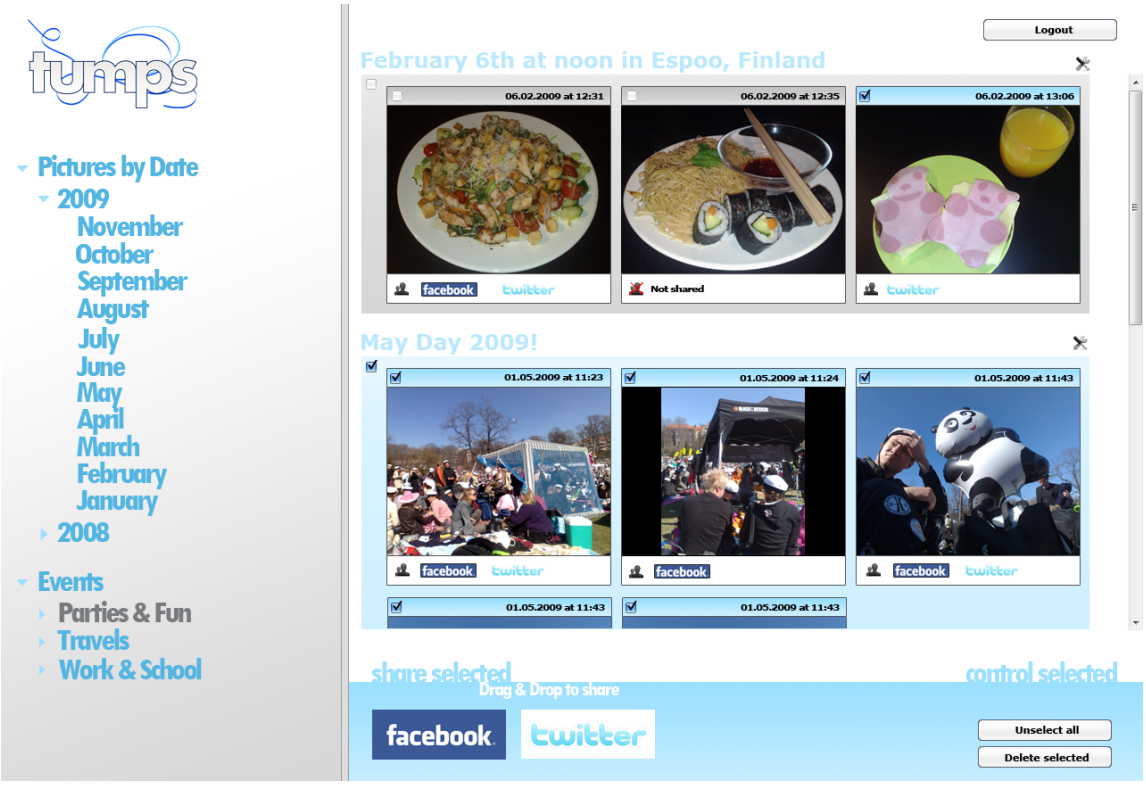


Figure 4.4: Fourth draft of the web UI

### 4.1.3 Architecture and Technologies

The development team decided to split the software that existed on the web server into two parts: the UI, and a backend server that handled the actual photo storage and processing logic. The backend server handled all user management as well. This had multiple benefits:

1. Photo uploading, viewing, manipulating and sharing could be done through a single interface, decreasing complexity and point of failures.

2. The web UI could be hosted on another server than the backend, improving scalability.

3. By decoupling the user interfaces (mobile and web) and the actual service offered by TUMPS, either the uploader or the web UI could be completely replaced without having to touch the server logic.

4. If the server API is made open to the public, anyone can extend the service by creating their own uploader, photo management tool or ser-


Figure 4.5: An enlarged photo in the fourth draft of the web UI

vice to which TUMPS could share images to. For example, one could write a desktop client without having to change anything in the server.

The backend server implemented a RESTful API for all photo operations. REST is a software architecture style that uses different resources (in this case, URLs) upon which actions are performed (HTTP methods) [16]. Once authenticated, the mobile client simply uses HTTP PUT to send images, while the web client uses the appropriate actions in order to view, organize and share photos.

An additional benefit from using a RESTful HTTP interface relates to the vision of ubiquitous computing. One of the many challenges of this vision is the communication between different devices. When designing a device or service that is to be used in such an ecosystem, how can one guarantee that the device can communicate with all current and future devices? By using a standardized (HTTP) interface, other devices will have an easier time communicating with and supporting TUMPS.

The development and test server for TUMPS were cloud services from Amazon Web Services. AWS allowed us to instantiate a new server with minimal effort, dynamically increasing or decreasing computing power and memory, if needed. As a web server, the popular Apache Tomcat Java server was used. Instead of coding the web server with Java, however, a more modern approach was tried. The team chose the Lift web development framework, which uses the Scala programming language. Scala is a language that compiles into Java byte code (thus being able to run on Tomcat) but generally requires fewer lines of code when compared to traditional Java.

For the actual web UI, since the photo handling was done by the server and had a clean API, the team decided to make the UI completely client-side with the SproutCore framework. While the web UI seems like a web page, it is in fact a client-side application that uses HTML5 technologies to deliver an enhanced user experience while performing all operations through the server's RESTful API. The amount of effort from the development team that was required to implement features such as drag and drop was minimal. The downside of using SproutCore was that it required a relatively new web browser that supported the HTML5 technologies (Firefox 3, Safari 3 or Chrome).

One difficult technical hurdle to overcome during the project was related to the second project goal: having an installation process that is extremely simple and requires minimal effort. We decided that one way of achieving this is to preconfigure the installation package with the user's credentials, which would remove the need for the user to enter any text on their mobile phone during installation — a huge usability win when considering the small physical or virtual keyboards that current cell phones have.

Unfortunately, this installation package pre-configuration was not directly compatible with the proper way to install software on an N900: through the package manager, apt-get. Apt-get uses so-called catalogs or repositories, which contain the software installation packages of most software that is available for the N900. All the users download and install software from the same catalogs, meaning they can't be personalized (since everyone downloads the same installation package). We wanted to use apt-get for the software installation, since without it our one-click installation would be much more difficult (and as a added bonus, the TUMPS software could easily be updated by the system and uninstalled by using the N900's software uninstallation feature). This limitation was overcome by the fact that users can add catalogs to the phone. Each time a new user entered their phone number on the installation web page, the server would create a new user account, a new

catalog just for this user, make a preconfigured installation package and place it in the new catalog. This solution has one glaring problem: over time, as new users sign up, the number of single-user catalogs adds up. The team decided to solve this problem by removing the catalog once the user logs in to the web UI for the first time.

The installation process also required the sending of a message to the phone of the user, which contained a link to the installation package. Although using e-mail would have been easy, not everyone has their phone configured to receive their e-mail messages. However, every cell phone user can receive SMS messages, which the team ended up using in order to have the largest possible user coverage. Sending SMS messages was achieved by using a service called Clickatell[1], an SMS gateway that converted HTTP-requests sent by the server to SMS messages.

For version control, the team used Bitbucket[2], a cloud service that offered mercurial version control. Team members pulled the source from Bitbucket, worked on it on their own machine, committed the changes then pushed these commits back to Bitbucket.

By installing a continuous integration service onto their AWS development server, the team was able to link their version control directly with their development server, automatically building the newest version of the software whenever someone committed new code. Unit tests were also run at this point, quickly notifying the team if something was wrong with the newest commit.

The overall architecture of the system is shown in Figure 4.6.

[1] `http://www.clickatell.com/`
[2] `http://bitbucket.org/`

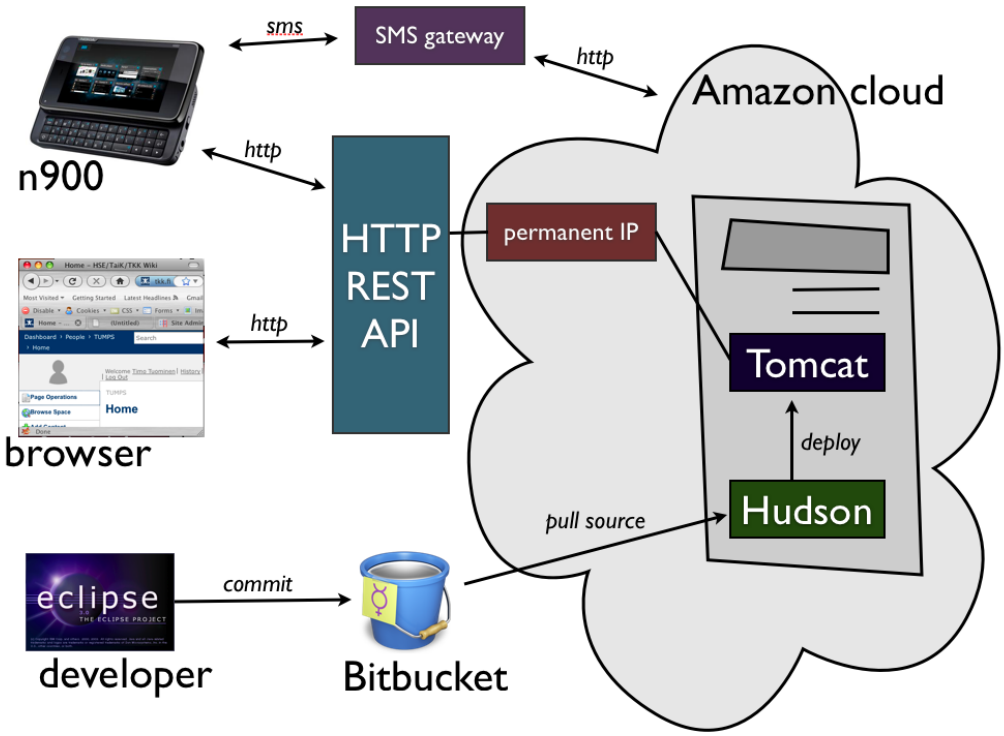


Figure 4.6: The TUMPS software architecture

### 4.1.4 TUMPS Usage

The installation process for TUMPS is started by opening the installation web page with any browser. The web page can be seen in Figure 4.7. Once the user enters their phone number, TUMPS sends them two text messages. The first one includes a URL to the customized installation package and the second one includes the URL and password for the web UI.

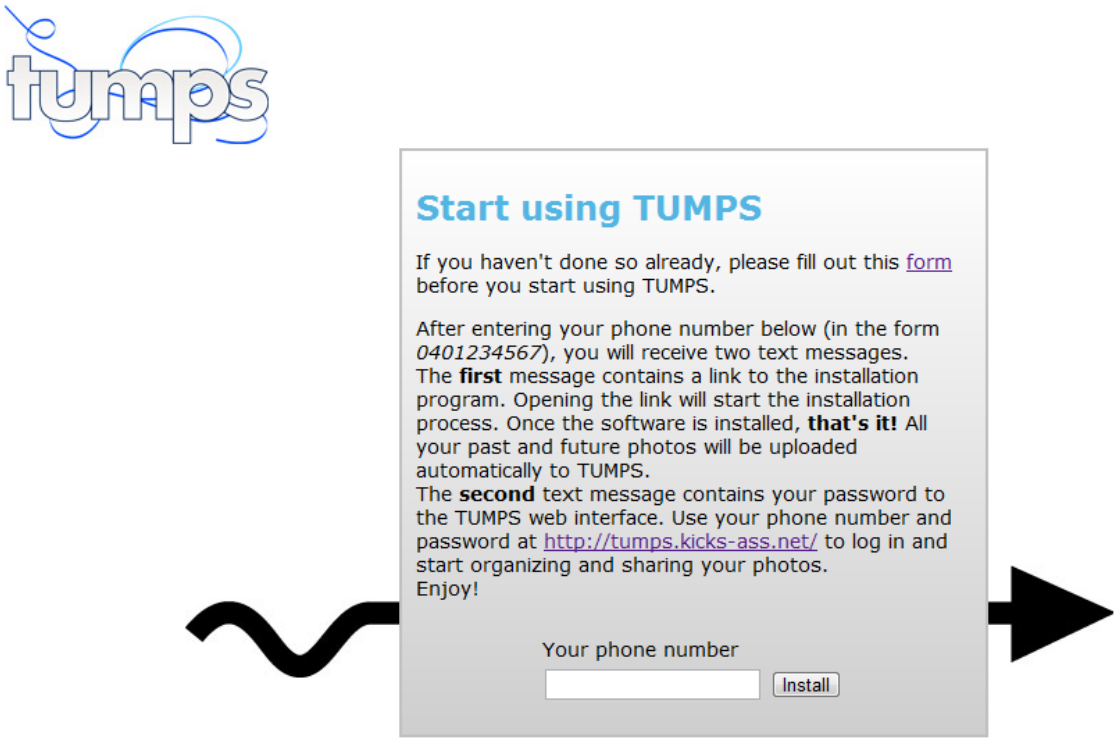


Figure 4.7: The TUMPS installation page

Once the user opens the installation package URL on their phone, they are asked to add the custom catalog that contains the installation package to their list of catalogs. After agreeing to this, they are asked if they want to install the actual software. Once the user also agrees to this, TUMPS is installed and the software is started. It automatically starts transferring all existing photos to the TUMPS server. The software is installed as a service, so it will continue sending photos, even if the phone is restarted.

When the user wants to view and share their photos, they log in to the TUMPS web UI with their phone number and password that they received with the second SMS message.

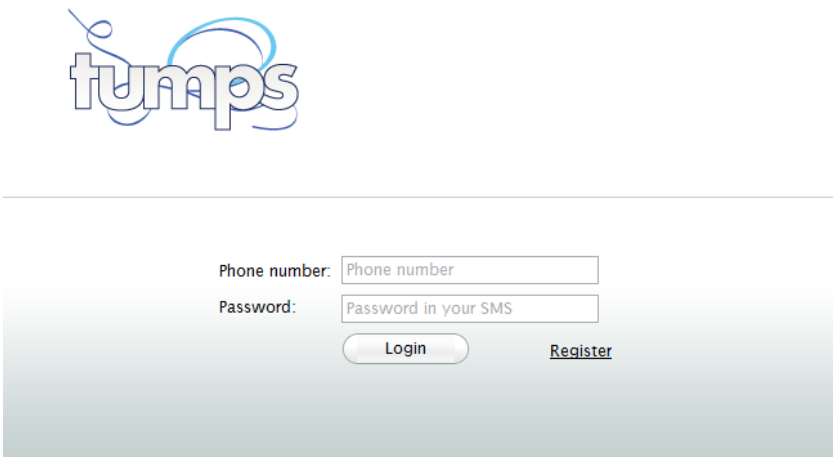


Figure 4.8: The TUMPS login page

Upon logging in, they are presented with a stream of photos, with the newest ones displayed on top. The stream uses an "infinite scroll" technique, meaning that older photos are loaded dynamically as the user scrolls down the page instead of loading all the user's photos at once.

The user is able to narrow down photos by selecting a year or a month from the left-hand pane. The right pane will then only show photos that were taken during that time frame, with the newest ones on top.

Events can be used to further organize photos. An empty event can be created, after which photos are added to it by dragging and dropping them onto the event name, which is located at the bottom of the left-hand pane. All photos that are not associated with an event can be seen by selecting "Ungrouped photos".

By double-clicking on a photo, a larger version can be viewed (as seen in Figure 4.10). The user can view photos as a slideshow by clicking the Next and Previous buttons.

The only service that users can share photos to is Facebook. If the user wants to share photos, the photos are selected (Ctrl or Shift clicking if multiple photos are to be shared), after which the Share to Facebook button is pressed. If it is the first time the user is sharing photos, they are asked to log in to Facebook and give access right for TUMPS to publish photos for them. This setting is remembered and needs to be done only once.

TUMPS does not provide baseline web services features, such as the ability to

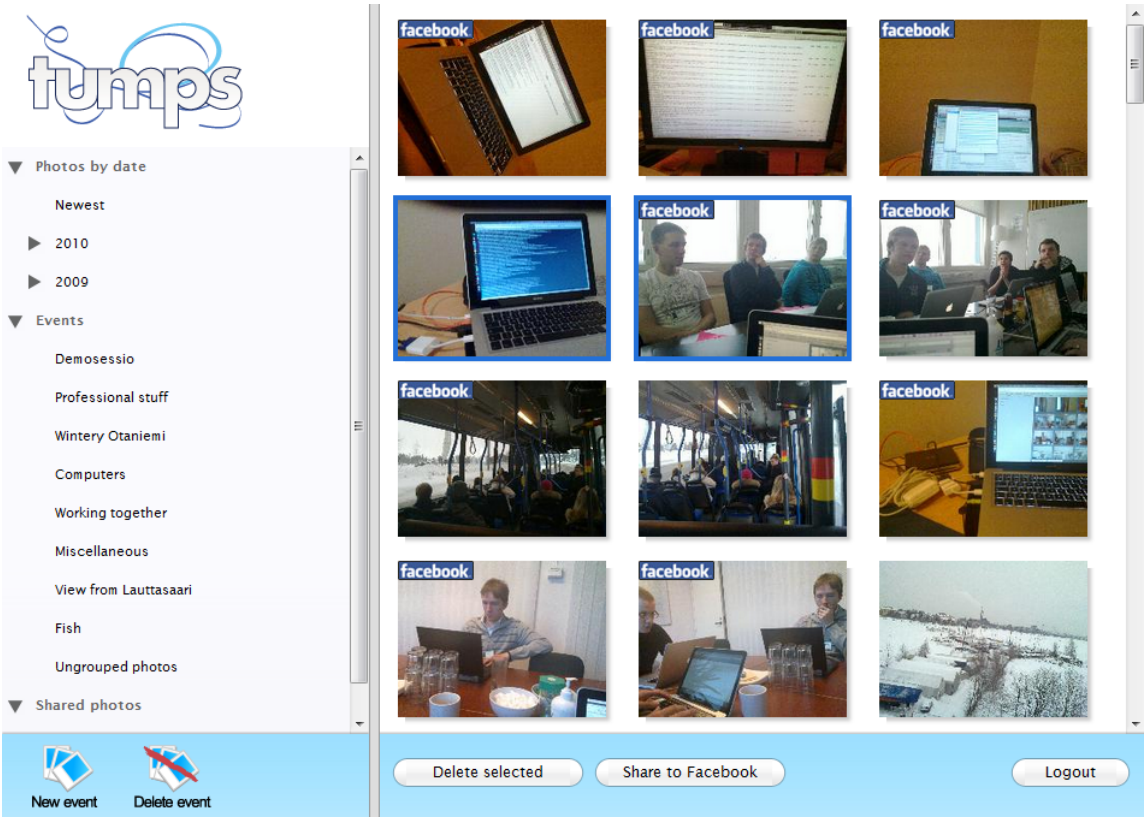


Figure 4.9: The TUMPS web UI

change a user's password or delete their account (actually, deleting an account is possible through the API, but is not implemented in the user interface). This was a conscious decision — all of the team's effort was spent focusing on the three project goals. If TUMPS were to be released to the public, a lot of work would be needed to implement these features that end-users expect as a given.

## 4.2 User Studies and Testing of TUMPS

In order to validate the outcome of our TUMPS project, the service was deployed and tested by a group of users. The goal was to see if our assumptions about the usage were valid, if the system had some critical problems and to see what improvement would be needed in order to make TUMPS a viable consumer service. We also wanted to find out users' feelings related to issues such as privacy and trust, which are important for a service that

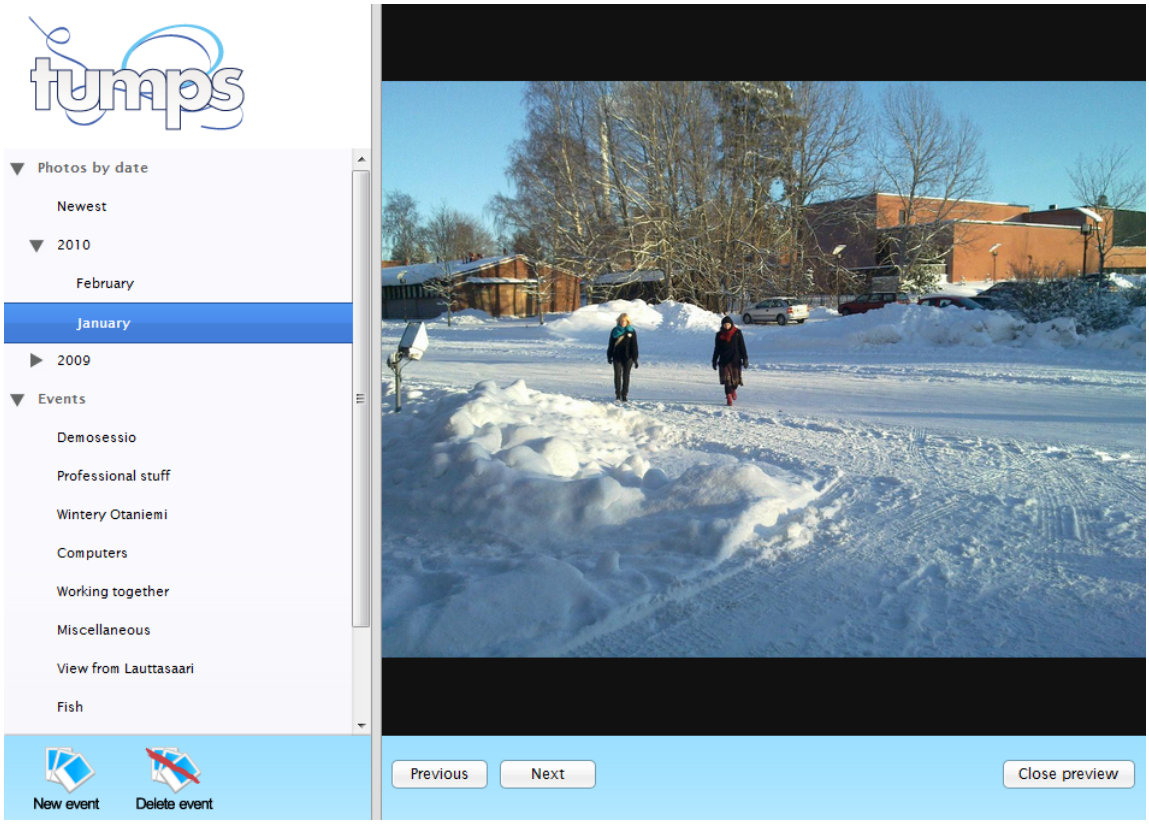


Figure 4.10: Viewing a single image in TUMPS

handles private objects such as photos. Finally, we wanted to try to gauge the reaction of end-users if TUMPS was a paid service.

### 4.2.1 User Test Group

Testers for TUMPS were gathered with the use of the social networking site Facebook. With a status update, I asked my contacts for volunteers that owned an N900 and an unlimited GPRS data plan. The unlimited data plan was required since I didn't want to incur any costs for the users, and using TUMPS would require the transferral of many megabytes of data (the actual amount depending on the number of pictures taken). I also asked coworkers at Futurice that use an N900 as their primary phone to participate as testers.

Although the ideal situation would have been to get a diverse test group as possible, this was not achieved due to two main reasons:

1. Most people who own a Nokia N900 are not representative of an "av-

erage" mobile phone user. The price (currently about 500 euros in Finland without an operator package) places it as a premium smartphone and well above the price of an average phone. Many of the testers also mentioned that the reason for buying the phone was the ability to tinker with it (due to the underlying Linux operating system), making them technologically more adept than average mobile phone users.

2. People in my Facebook circle of friends as well as my coworkers are mostly in the approximate same age group as myself (20-30 years old) and have a similar educational and geographical background (current or former students from the Helsinki University of Technology or living near Helsinki, Finland).

I ended up with a group of 13 testers. Out of these 13, two people were unable to get the service working at all and seven stopped using it at some point before the end of the test period. Out of these seven, three were not available for an interview. All in all, ten people were interviewed, with two of them having no actual experience using the software due to technical problems, and thus were not asked questions related to actual TUMPS usage. Nine out of the ten interviews were performed face to face and one was done with a Skype[3] audio call.

### 4.2.2 Interview Analysis Method

All of the interviews were recorded. After the interview, the interview was played back and notes were made on all the answers and points brought up by the interviewee. These were divided into the following topics:

- Tester background and photography habits
- TUMPS installation
- TUMPS mobile software
- TUMPS web UI
- Usage amount
- Improvement ideas
- Privacy

[3] `http://www.skype.com/`

- Expectations and reactions
- Future usage and payment

Once these summaries were completed, they were gathered into one spreadsheet that allowed for an easier visualization and analysis of certain question answers. For example, the "Used events" and "Shared to Facebook" were two columns that depicted the usage of two web UI features, and contained each user's usage information for those features. Complaints about technical problems, new features requests and general comments were each added to their own column. Once this spreadsheet was compiled, summarizing interview results was relatively easy.

### 4.2.3 Results

#### Pre-TUMPS Questionnaire

Two out of the ten people interviewed said that they don't use the social networking site Facebook at all or rarely, while the rest answered that they use it daily or constantly. Four people said that they mostly use it to read what their friends are doing, while six said that they also use it to post content about themselves. When asked how often they share their photos online publicly, one user said daily, five answered monthly and the rest answered yearly. Eight out of ten answered that these were one-off, situational photos.

For cell phone photo usage, one user said that they take photos daily, four said weekly and four said monthly. One user replied that they never take cell phone photos, but this was most likely due to the fact that they received their N900 phone at the same time as they started using TUMPS. Of these cell phone photos, eight answered that they had shared or downloaded only 1-30% of these onto their computer, while the rest answered that they hadn't shared or downloaded them at all.

#### Photography Behavior

Out of the ten people interviewed, everyone owned and used another camera in addition to the N900. Five owned a DSLR, four owned a "pocket camera" and one used both.

Everyone used the N900 to capture unique, one-off situations — photos that they would want to share with other people because they were in some way

special (funny, unique, beautiful). Only two out of the ten people used the N900 camera in another manner: one used it as a replacement for pocket cameras (i.e. using it as the main camera to document a trip or experience) while another tester used it also as a sort of document scanner, taking pictures of text or photos instead of manually entering them in with the phone's keyboard.

Prior to using TUMPS, different methods were used to transfer the photos off of the cell phone. Six out of ten never transferred all of their photos off of the phone, transferring only selected ones with various methods: e-mail (3 users), built-in Facebook sharing (2 users), built-in Flickr sharing (2 users) and USB (2 users). Two users performed periodic backups using USB, sharing photos straight to Facebook with the built-in sharing tool. One user had a homemade script that automatically backed up all their photos to their own server periodically. Another user started using TUMPS as soon as they obtained the N900 and used TUMPS as the only way of transferring photos elsewhere.

Out of the ten interviewed, only half of them processed the photos in some way. Adobe Photoshop Lightroom[4] was the most popular tool, with four users (all of them DSLR owners as well) and Apple iPhoto[5] had one user.

### TUMPS Usage, Feelings and Expectations

Out of the eight people interviewed that were able to use TUMPS, only half of them continued using the software (i.e. had the N900 software installed) until the end of the month-long test period. Two people used it for less than a day, while the two other ones stopped using it after about two weeks. One of the testers that used it for less than a day uninstalled the software as soon as they realized that TUMPS sent all photos to the server, including ones taken before installing the mobile software. One of the testers that used it for about two weeks had to uninstall it because they needed to save battery power, and there was no way to disable photo sending temporarily. Since there was no way to reinstall the software easily, they stopped using TUMPS after the uninstallation. The other two testers that stopped using it did it due to a software bug that caused a connection dialog to pop up every five minutes if the phone did not have an Internet connection.

The three users that were unable to participate in an interview all used TUMPS for less than one day before removing the software from their N900.

[4] `http://www.adobe.com/products/photoshoplightroom/`
[5] `http://www.apple.com/ilife/iphoto/`

Two of them stopped using it due to the aforementioned connection dialog bug, while one user was unable to get the mobile software to upload all of their photos.

Out of the eight testers that were able to use TUMPS, five said that having TUMPS installed made them think about the fact that their photo would be sent somewhere while they were taking a photo. Out of the three who said that they didn't think about it, two said it was due to the fact that they never took really personal photos with their cell phone. However, even though 5/8 testers said that they think about it, none of them said that having TUMPS installed would change their photography behavior (two testers said that it "might"). This can partly be confirmed by quantitative data: out of the eight testers, shooting frequency was not affected when comparing amount of photos taken before and after installing TUMPS.

When logging in to the service for the first time, five out of eight testers felt a sense of surprise. Three felt a positive surprise, either by the fact that their content was already available or by the fact that TUMPS transferred all photos, not just ones taken after installation. However, the fact that all photos were transferred caused one person to be neither positively nor negatively surprised, but simply was not expecting it and didn't know what to think. For one tester, however, this surprise was so negative and unexpected that they immediately uninstalled TUMPS from their phone.

Half of the testers used the web UI about once a week. For the other half, they used it only a couple (1-3) of times. Only one user used the Event-feature available, while only two people shared photos to Facebook from TUMPS (both of these testers would not have shared the photos to Facebook without TUMPS).

Certain features of TUMPS received explicit compliments from the interviewees. The idea of the service overall received four compliments. The ease of installation and the fact that the web UI was much more like a piece of desktop software rather than a web page also received many compliments, with three each.

### Future Usage

When asked if the users would continue using TUMPS with the current technical problems/bugs fixed, five said they would continue using it, one said they would if it was possible to block certain photos from being transferred and two said that they would not since TUMPS does not currently offer any use cases that are not already possible with the N900's photo sharing

feature. Both testers that were unable to get TUMPS to work also said that they would use the service if the current technical hurdles were overcome.

After asking the users if they were willing to pay a couple of euros per month for the service, only one said that they would, one said they might and the other eight people interviewed said that they probably wouldn't. A few mentioned that if the service was more polished and had more features that were applicable to them, they would consider paying a small sum for it. When given the hypothetical situation that a cell phone operator would provide TUMPS as a free service with an unlimited GPRS/3G data plan and asked whether it would affect their decision to buy the data plan, six out of ten said that it might — on the condition that they were comparing it to data plans offered by other operators and other factors (especially price) were equal. The rest of the people interviewed said that it would not affect their decision. It must be noted, however, that this was a situation was purely hypothetical, since all of the interviewees already had an unlimited data plan. Most of them mentioned the fact that a smart-phone (such as the N900) practically requires an unlimited data plan in order to extract all of the value from it, and that their decision to subscribe to an unlimited data plan is based mostly on this (and possibly other factors) rather than being able to use TUMPS.

Interviewees were also asked whether they would use a TUMPS desktop software that automatically sent photos that were placed in a certain folder to the web server. Three said they would use it (mostly as a backup solution for their photos), three said that they might use it and four testers said that they would not use it. Concerns were expressed about the fact that they have thousands of photos on their computer, and that this feature would require a lot of bandwidth and space on the server.

### Technical Problems

Certain technical problems or bugs received a lot of attention from users. The most commented problem was the connection dialog that sometimes popped up every five minutes for the user (seven out of ten people). Another common problem was the fact that the catalog that the software was installed from was removed from the server once the user logged in to the web UI, resulting in an error message being displayed on the user's phone when it tried to check for software updates. In order to fix the problem, the user had to remove the catalog from the phone.

One small technical problem that had a surprisingly large impact was the

fact that the login form was not made with conventional HTML, but rather Javascript. Four people commented on it, partly because they could not login by pressing enter, but more importantly it did not allow their browser to use the login credentials storing feature. This meant that since changing one's password was not a feature that was implemented in TUMPS, whenever they wanted to login to the TUMPS web UI, they had to find the original SMS that contained their password.

A surprising technical problem that was uncovered was the fact that using the TUMPS web UI was not really possible using the N900 web browser. Accessing the web interface using an N900 was not a use case we had expected — since all the users' photos are already on the phone and sharing to Facebook is possible directly from it, we had not anticipated people trying to log in with their phones. However, three people had tried using the web UI from their phones, unsuccessfully.

One final technical problem of note was related to browser incompatibility. Some users tried using TUMPS with the Opera browser, which is not supported by the framework (Sproutcore) used to build the web UI. Two users also mentioned problems with the Firefox web browser (the scrollbar would disappear).

### Improving TUMPS

During the interviews, many improvement suggestions and new feature requests were given. Of these, the most requested one was a GUI for the mobile software (eight out of ten interviewees). This would serve multiple functions: as a status indicator, showing when all photos have successfully been transferred (or when an error has occurred). Six users also wanted to have the ability to turn the transferral on or off, with the possibility to mark certain photos as private (i.e. they are never sent to the server). For one user, the inability to block the sending of certain photos was a deal-breaker, while the inability to temporarily disable the transferral of all photos was a deal-breaker for another user.

For the web UI, different destinations where photos could be shared to was the most requested new feature. The following services were mentioned: Flickr (7 votes), Twitpic[6] (3 votes), Picasa[7] (2 votes), Worpress[8] and other blogging platforms (1 vote) and SFTP/FTP (1 vote). A surprisingly large number of

[6] `http://www.twitpic.com`
[7] `http://picasaweb.google.com/`
[8] `http://wordpress.org/`

people (7 interviewees) wanted the ability to easily share either one picture or a group of pictures by simply getting some sort of URL that they could send to whomever they wanted. Two people also wanted the ability to send photos be e-mail. Six people would have liked a Download-button, giving them the ability to easily transfer selected photos to their computer, while two users would have liked this to have been automatically done somehow for all photos.

Table 4.1 contains all the other improvement suggestions given by the users.

Table 4.1: Test User Improvement Ideas

| Feature | Count |
|---|---|
| (Web) Basic editing (rotate and crop) | 3 |
| (Web) Public API | 2 |
| (Web) Photo tagging | 2 |
| (Web) Collaborative albums | 1 |
| (Web) After logging in, not all photos would be shown | 1 |
| (Web) Utilizing photo GPS data (e.g. showing photos on a map, search by location) | 1 |
| (Web) Changing "Event" to "Photo type" | 1 |
| (Web) More options when sharing to e.g. Facebook (visibility, photo text) | 1 |
| (Web) Search | 1 |
| (Web) Printing service integration | 1 |
| (Web) A GMail-inspired "Archive" button for photos | 1 |
| (Mobile) A confirmation dialog once the installation is successful | 2 |
| (Mobile) Integration with the existing camera app (for example by adding an on/off button for transferrals) | 2 |
| (Mobile) The ability to change the folder that TUMPS monitors for new photos (would allow the user to store images on an external memory card instead of the phone's built-in memory) | 2 |
| (Mobile) Changing TUMPS to not open a connection if no internet connection is open | 2 |
| (Mobile) Reinstallation support | 1 |
| (Mobile) After installation, ask the user whether they want existing photos to be transferred or not | 1 |
| (Mobile) A "force sync now" button | 1 |
| (Mobile) A persistent tagging mode (all photos taken are tagged with the currently active title) | 1 |
| When uninstalling the mobile software, the ability to remove the online account as well | 1 |

# Chapter 5

# Discussion

## 5.1 Test Result Limitations

When analyzing the results of the user testing, some things need to be taken into consideration.

1. Since the test users do not represent a varied selection from different demographics of users, not all results can be said to apply generally to "average" consumers. The test users were all about 20-30 years old and most of them either current or past students of the Helsinki University of Technology. Also, since the N900 is far from an average consumer cell phone (being a relatively expensive smart-phone that uses a new mobile operating system), they can be considered more technologically minded and belong to the early adopters consumer group. However, certain results from these tests may be extrapolated to other types of users. Since the price of cell phones are constantly being pushed downwards, the early adopters and high-end of today could very well represent the average users of tomorrow.

2. The trial period was only for one month. Especially the analysis of the photo organization and searching process is something that would, in an ideal case, have a large volume of photos from a long period of time.

3. Interviewees might be skewed as to wanting to "please" the interviewer by praising the tested service rather than giving criticism. However, I tried to alleviate this by (truthfully) mentioning at the beginning of the interviews that I had not personally built the service and that they should be as truthful as possible during the interview.

## 5.2 Effect on Photography Behavior

Even though the quality of cell phone cameras has constantly been improving over the past decade, it was clear from our tester group that their cell phones have not replaced dedicated digital cameras as a tool to capture and document their lives. For every member of the test group, the main use of the N900 camera was capturing (and often sharing) one-off situations — situations or scenes that otherwise would have passed, but for the fact that the participant had a camera with them at all times in the form of a cell phone. When going on, say, a trip that they want to document, all of them had another camera that they would use. However, one of the testers said that the N900 was good enough to take pictures outdoors, but low-light performance was not acceptable. The camera on the N900 could even be considered as a high-end cell phone camera, with features such as a 5 megapixel sensor, autofocus and a flash. Lower end cell phones, which are much more common, have cameras that are of even lower quality (fewer pixels, no autofocus or flash).

Based on this behavior, we can assume that users did not use their cell phones for one of the social functions of photography: capturing visual histories. Photos were mostly taken for the other functions mentioned in Section 2.1.

We can expect the quality of cell phone cameras to continue improving. For example, Nokia recently announced a new cell phone, the N8, to be released in the fall of 2010, that has a built-in 12 megapixel camera. Sample images appear to be of very high caliber. However, this is only speculation, and real world usage will determine if it is an example of a cell phone camera that crosses the boundary in order to be "good enough" to replace dedicated digital cameras.

Until the quality of cell phones cameras increase to an acceptable level, people will continue using dedicated digital cameras for storing visual memories. Removing the explicit transfer phase for those cameras is currently possible with solutions such as the Eye-Fi, which automatically sends the photos over Wi-Fi to the user's computer (or online services). The next possible step would be to remove the physical limitation of having to be within the range of certain Wi-Fi networks by replacing Wi-Fi with GPRS/3G as the communication channel. Technically, this is currently possible, but no products are currently available to the general public that offer this.

Since cell phones are still mostly used to take situational photos (due to the fact that they're constantly with us), for our test group, TUMPS did not alter their photography behavior or enable them to do things that would

have otherwise been impossible or difficult. The Nokia N900 already has built-in sharing features that allow photos to be shared to popular online services (such as Flickr and Facebook), and for one-off photos these features are easy enough to use and require little effort on behalf of the user. Only two users said that TUMPS made them share photos that they otherwise would not have shared.

Having TUMPS installed didn't seem to have an incentivizing effect with regards to taking more pictures. The frequency at which photos were taken didn't increase noticeably for a single tester. Since one of the main selling points of TUMPS is the ability to automatically transfer even a large number of photos off of the capture device, I believe that users will receive the most benefit once their cell phone's camera crosses the "good enough" barrier for quality and they start using their cell phone as their primary, documenting camera. This will increase the number and frequency of photos taken with it, and thus automating the transfer phase will lessen the amount of effort required much more than TUMPS currently does.

Even though the main feature of TUMPS didn't directly benefit test users' photography habits, TUMPS offered certain benefits that the users liked: it worked as a backup for their photos, it allowed them to easily transfer their photos to their computer and it allowed them to show their photos to other people from a larger screen. This can be seen from the fact that eight out of the ten people interviewed said that they would continue using TUMPS.

## 5.3 Expectations, Trust and Control

For a service where users store lots of personal items — such as TUMPS — meeting expectations and thus building trust is critical. Once the user's trust is broken — by doing something other than what was expected, for example — it can be incredibly difficult to build it back again. For this reason, the operation of the service must align with user expectations as closely as possible. In order to achieve this, multiple things can be done: the operation of the service can be explained using text and video, the user interaction flow should be made as natural and obvious as possible and unexpected situations need to have clear instructions on what has happened and what the user can do from then on.

Explicitly explaining what something does is not always an effective method of guiding the expectations of users. Instructional texts are often only skimmed or skipped completely. As an example, the following text was clearly

visible on the installation page for TUMPS:

> After entering your phone number below (in the form 0401234567), you will receive two text messages.
>
> The first message contains a link to the installation program. Opening the link will start the installation process. Once the software is installed, that's it! All your past and future photos will be uploaded automatically to TUMPS.

Even though it is clearly stated that TUMPS will upload both new and existing photos, four out of eight users said that they were not expecting old photos to appear in the web UI when they first logged in. For two users this was a positive surprise and thus their trust toward the service was not affected negatively, but for one user the shock was so big that they immediately uninstalled TUMPS from their mobile phone. This indicates that users did not read the provided instructional text or then forgot the message once they started using the service. We cannot thus rely only on text to guide the users' expectations. However, showing the text is a different matter (such as a popup on the cell phone while installing the mobile software) might prove to be more effective.

Trust is also closely related to privacy issues. Users often take personal photos with their cell phones, and they must trust TUMPS enough to store these personal items in a secure and reliable manner. Leaking some of these photos to unintended parties is probably one of the worst things TUMPS could do, since users would not be able to trust the service with handling their personal items anymore. One interviewee suggested that TUMPS could get a security audit from a third party, partly validating the security and privacy practices of the service. Giving the site a polished and professional look would most likely also help in building the trust of users. One tester mentioned that it might be a good idea to look at the visual techniques that banks and existing online backup providers use to give an initial impression of reliability.

An interesting viewpoint with regards to trust was given by one interviewee. They said that they would trust the service more if it had a lot of users, even though it would be a much more valuable target to hackers at that point. They had the mindset that "if so many people trust the service with their photos, then I guess I can trust it as well". Of course, getting the service to be used by so many people in the first place requires other methods to build the trust of the users.

Another suggestion that would greatly increase privacy and trust toward TUMPS that was given by one interviewee is if each user would get their

own (virtual) computer, thus shielding their content from other users of the service. In practice, this is currently not feasible due to high cost — if few users are barely willing to pay a couple of euros a month for a service such as TUMPS, they would most probably not be willing to pay the tens (or hundreds) of euros per month that their own server would cost. However, with the declining prices of computing power, this option might be viable in the future.

Moving program logic and user data online — into the "cloud" — is a trend that has been gathering support from users for a while. Services such as Dropbox, Google Docs[1] and the many online backup solutions are successful services with lots of users, even though the end-user is storing possibly private data in the servers of these service providers. This loss of control is, however, outweighed by the many benefits that storing data in the cloud provides: the ability to access the data from anywhere where an internet connection is available, the ability to access them from different internet-enabled devices, interoperability with other online services and not having to bother about technical details such as file handling and backups.

TUMPS is an expansion of this trend into the realm of photography. If users trust services with the handling and storage of their documents and files, it is not far-fetched to believe that they would also be prepared to move their photo collection into the cloud. Of course, there are certain photos that users will never want to lose complete control of: very personal photos and photos that have confidential material in them. A service such as TUMPS needs to take this into consideration if they do not want to lose the customers that have these types of photos. However, for most users, these photos are rarer than the photos that they are willing to trust the service with, and can thus gain the benefits that are associated with moving data into the cloud.

## 5.4 TUMPS 2.0

### 5.4.1 Fixes

Based on user experiences and suggestions, we can compile a list of problems that need to be fixed with TUMPS and new features that would greatly improve TUMPS as a service.

One major problem that would need to be fixed is the logic and method with which the mobile client transfers the photos. Currently, when it finds new

[1] `http://docs.google.com/`

photos, the N900 software always tries to open a connection in a way that gives the user a popup dialog if no connection is open. This needs to be changed so that it either doesn't give a popup message and connects, or then waits until a connection is already open before sending the photos.

The mobile software had some bugs related to finding and uploading photos. For some of the testers, the software would upload only a few photos or in two cases, none at all. One bug that was discovered was triggered if the user had recorded a video: the software would stop uploading new photos upon reaching it. Other reasons for the upload not working were not further investigated and the root cause of the problem was not found. This is something that would need to be worked on in order to get TUMPS working for all N900 users.

Another important problem to fix is the method with which a unique installation package was delivered to each user. The current method — creating a catalog or repository just for that user, then removing it from the server once the user logs in — resulted in users getting a "Catalog not found" error once the phone tried to update its software. Having the user manually remove the catalog was not the only downside of this approach: once the catalog was removed, TUMPS lost the ability to automatically update the mobile software. Two solutions for this come to mind:

1. Do not remove the catalog from the server once the user logs in. There are two downsides to this: a permanent catalog is made for each user, resulting in wasted space on the server and if an update to the mobile software is made, a customized version of the updated software needs to be deployed to each of these catalogs.

2. Make a single installation package for the software and deploy it to a common public catalog and then personalize it once the installation is started with data from an external source. This data might come from the server, or optionally we could add one step to the installation process and ask for the user's mobile phone number. The additional effort required for installation by the end-user might be outweighed by the benefits when comparing it to the current personalization method.

Users would also have liked some sort of message once the software is installed, telling them that the software was successfully installed and operational. A couple of users tried searching for some sort of TUMPS application, since they did not realize that the software was already running once installed.

The web UI had a number of minor but annoying issues. These include the non-standard login form (preventing the browser from storing login credentials), incompatibility with certain web browsers, not being able to rotate photos to the correct orientation and seemingly random errors with the Event feature. Further usability testing with a varied arsenal of web browsers would be needed in order to locate all of these bugs and fix them.

### 5.4.2 New Features

In addition to fixing these technical problems, users wished for additional features that would have made TUMPS a more useful service for them. The most requested feature was some sort of GUI for the mobile phone software. Users wanted it for different reasons, but the most common one was the ability to see the status of photo uploading. Being able to see what the software is doing, if all photos have been uploaded and if there are any errors give the service a level of transparency that many users would have wanted. This transparency also helps in building trust, as the service no longer works "by magic". On the other hand, hiding the technical transferral details from the user is one of the goals that we want to achieve with TUMPS, but in order for this to work, it must "always work".

A second feature that the mobile GUI would allow is the temporary disabling of transfers. Some users were traveling abroad or needed to save battery life and wished that TUMPS could have been turned off for these periods.

The third most requested feature for the mobile GUI was the ability to block the sending of certain photos. Users had some sensitive photos (work-related or personal) that they did not want to send anywhere, no matter how secure. Again, giving the user more control of what is being sent (and when) is very important in building the trust of the user.

On the web side of the service, users would have received the most benefit from being able to send photos to more services and thus share them with others in different ways. The N900 already has good sharing tools for Facebook (the only service that was implemented in TUMPS) and thus the sharing capability of TUMPS was used by only two people. The new suggested sharing destinations for TUMPS included services such as Flickr, TwitPic, Picasa, blogging platforms and as e-mail attachments. Users would have also liked it if they could have downloaded the original images easily to their computer.

One common use case for the testers was sharing one or multiple photos with a group of people as effortlessly as possible. Instead of multiple steps involved in sharing to some other service, this could have been implemented

as a simple feature in TUMPS: the user selects the photo(s) they want to share and clicks on a "Generate URL" button. This creates a gallery with only the selected photos that can be accessed with the URL that the user receives. The user then simply sends that URL to whomever they want to share the photos with, be it with an e-mail, social networks or chat services. It would also be possible to "expire" this URL in case the user wants to disable it.

One of the weaknesses of having only a month-long testing period is that the need to find and organize photos from that time period is minimal, especially if the amount of photos taken during that period is not very high. Finding the photo that one is looking for is possible by simply browsing through the chronological photo stream, or by limiting photos to a certain time period. This can be seen from the fact that only one user used the organizational feature that was provided: Events. A couple of recommendations were given by testers on how it could be improved. Two testers would have liked tagging functionality instead, which allows the user the ability to tag photos with an arbitrary amount of textual tags, which can be searched for later on. These tags could then be also used when exporting the photos to services that support tagging, such as Flickr. One user would have liked "Events" to have been changed to "Photo Type", arguing that over time, he would have way too many Events listed.

An interesting feature suggestion that would help in organizing photos was a Gmail-inspired "Archive"-button. Since TUMPS is a service that automatically gets new content, users might be most interested in content that has appeared since their last visit to the web UI. Upon logging in, this new content is handled in some way (photos are tagged and a selection of photos are shared to certain services, for example), after which they can be archived and hidden from the user. Of course, the user can access these photos again by selecting that appropriate period in time or by searching. This feature would also solve one problem reported by a test user: they often showed friends photos using the web UI, but didn't want everyone to be able to see his whole photo stream once logged in.

An increasing amount of cell phones are equipped with GPS capabilities, especially high-end smart-phones. The N900, which has a GPS receiver, can automatically add location information (geotags) to the metadata of photos. Utilizing this data in photo sorting and organizing is one possible improvement feature for TUMPS. Photos could be, for example, displayed on a map, or users could search for photos that were taken within range of a certain location.

## 5.5 Possible Business Models

Results from the test user interviews were rather clear: people were not willing to pay for TUMPS in its current form (even with technical problems fixed), since it the automatic transfer didn't help them enough that they would consider the saved effort to be worth their money. TUMPS also did not provide many features that were not already possible with the N900, namely sharing one-off photos to Facebook. Only one user out of ten said that they would be willing to pay for it, while another said that they might be willing to pay.

In order to maximize the potential customer base, all of these would greatly benefit if the TUMPS mobile client was expanded to other mobile platforms as well. With some platforms (Maemo, S60, Android), automatically sending the photos is possible, but with others (such as S40 and iPhone), having an application running constantly in the background is not possible. With these devices, a TUMPS uploader program would need to be explicitly opened before new photos could be uploaded to the server. Once the software is opened, however, the transferral can begin automatically.

TUMPS was also missing features that are must-haves for consumer products. These include proper user management, error logging and graceful fallback, feedback mechanisms, documentation and billing services. While they are not directly related to the goal of TUMPS, they are features without which TUMPS can not be deployed to a general audience.

Three different ways come to mind with which a company could generate revenue with TUMPS, which are explained in the following sections.

### 5.5.1 An Improved TUMPS

This method is probably the most obvious, but is in no way a guaranteed success. By developing and improving TUMPS in a way that brings enough benefit to potential users, it might be possible to cross the barrier that currently prevented the test users from wanting to pay for the service.

There are a few different directions in which TUMPS could be taken. One is to generally expand what TUMPS offers, adding features such as image editing, locational features and social features (commenting, collaborative albums, friend lists) in addition to optimizing the current automated uploading and sharing of photos. The target would be to provide many features, of which most people find at least something useful for them. This method

would make TUMPS a direct competitor of existing services such as Flickr. Competing against proven, high-quality services can be an extremely difficult position to be in, however.

A better approach, in my opinion, would be to focus on one or two central features and polish them to work extremely well, making TUMPS the best available service for those features. Instead of trying to be good at many things, TUMPS would be exceptional at one or two. The goal would be to make the user experience as painless and fluid as possible, letting the user achieve what they want to do with minimal effort. An example of this would be to make TUMPS a central repository for all of a user's photos, one that provides good tools to send the photos to other services that perform the other functions that a user wants to perform with their photos (edit them, discuss them, display them to a large crowd, print them, etc.). This would require a desktop client that transfers photos taken with other cameras, similar to the client used by the cloud storage service Dropbox[2]. Out of the ten interviewees, four said that they would use and two said that they might use such a feature. Of course, this is an example of only one of the areas that TUMPS could focus on.

Revenue from TUMPS would most likely some from a subscription model, where users pay a monthly fee to use it. It might also be possible to offer a "pro" version, that offers additional benefits but costs more. For example, if a PC client is made that uploads photos taken with other cameras, the basic version could offer a certain amount of storage while the pro version offers an unlimited amount of storage. Professional photographers that have a much larger collection of high-megapixel photos would be one of the target audiences for the pro version.

With many cloud services (e.g. Dropbox), users can sometimes have some trouble visualizing the benefits. The only way to truly see if a service is useful for a user is to use it. That's why it might be a good idea to offer TUMPS as a free trial for some period of time. Offering a trial period lowers the barrier of entry for many potential subscribers. After, say, two months of use, the user should be able to answer the question "do I get x euros per month worth of benefit from this service?". Flickr, which offers both a free and a Pro version of their service, limits the number of photos that free users can upload per month and the number of their latest photos are visible to the public. If they upgrade to the pro version, they can upload an unlimited amount of photos and all of their photos are visible. This creates an interesting dependency on the Pro version for professional users with lots of photos: if they decide that

[2] `http://www.dropbox.com/`

they want to stop paying for the Pro version, many of the links to photos that they had previously shared with other people will suddenly stop working.

Although eight out of ten people said that they wouldn't pay for TUMPS, quite a few mentioned that they would consider it if TUMPS offered more features, even though they were unable to necessarily specify what these features were. One user said that if the service was simply "nice to use overall", they might pay for it. This hilights the importance of a great user experience, and how the financial success of a service might depend on the user interface and overall feeling.

### 5.5.2 Offering TUMPS As An Incentive

If getting TUMPS users to pay directly for the service ends up not being feasible, another way TUMPS could generate revenue is indirectly. Instead of charging for the service, a company could charge for another product and offer TUMPS as a free service when paying for the other product.

An example of this is a mobile operator that wants to get more unlimited data plan subscribers. Since TUMPS practically requires an unlimited data plan, offering TUMPS with such a plan would be a logical pairing. Since the operator owns the transfer channel, they could possibly improve TUMPS by utilizing the mobile subscriber data in different ways. By placing the TUMPS server inside their own network, the traffic generated by TUMPS would not have to leave their internal network, reducing traffic costs as well. This example has its problems, however: not a single test user said that having TUMPS as a "free" service alongside an unlimited data plan would make them subscribe to the plan — the reason for subscribing to the plan comes from other needs, such as owning a smart-phone. However, six out of ten said that it might affect their operator decision if all data plans were otherwise nearly identical across different operators.

### 5.5.3 Wait For Changes in User Behavior

The third way with which revenue could be generated from TUMPS depends on an outside force: cell phone camera technology would need to advance to a level that changes the behavior of consumers. If the cameras on cell phones become good enough to capture all the photos that the user would want to take (i.e. visual histories), the amount of photos taken with it would increase, and thus the amount of effort that would be saved by automatically transferring photos off of the camera would grow. Consumers might then be

prepared to pay for a service (TUMPS) that would give them an increased amount of benefit.

The fact that camera quality will improve over time is nearly a given — however, whether the benefit from TUMPS ever surpasses the payment barrier for average consumers is a factor that is still unclear.

# Chapter 6

# Conclusions

Previous research has shown that users find the transfer phase of their photography workflow to be difficult and unnecessary [15]. In some cases — especially with cell phones — the operation is so tedious that the majority of photos taken with the camera are never transferred off of it.

With the use of current, off-the-shelf technology, it was possible to build a prototype of a system that automatically transfers all photos taken with a Nokia N900 cell phone camera to an online service, called TUMPS. From the web UI, users could flip through, search for and share their photos.

After using the service for a month, feedback from test users was clear: most (8 out of 10) of the users benefited from the service and would have continued using it, even though it currently provided minimal sharing features.

The usage of the service did not significantly alter the photography behavior of users. The image quality of cell phone camera photographs is still not good enough so that people would use it as their sole camera to document their lives. Once this barrier is crossed, however, and all of a user's photos travel through the service, the full potential of automatic photo transferral might be realized — manually transferring photos through a computer would simply become a step that is longer be required.

In order to make a commercial service out of TUMPS, it would need to be further developed. The main improvement targets would be the mobile client and additional online services where photos could be shared to. Revenue for such a service could come either from directly charging for it, or by offering TUMPS alongside another product (such as data plans for operators).

When building a service such as TUMPS, privacy, trust and control are key issues that need to be addressed throughout the product. Users take personal

photos, and in order for them to trust a service with their personal objects, they need to believe that their photos will not fall into the wrong hands, that the service is stable and has lasting power and that in the end, the users themselves have complete control as to where their photos are transferred.

The implemented solution to our stated research problem brings us one step closer to the vision of ubiquitous computing. Even though, technically, TUMPS doesn't help photo capture devices communicate directly with other types of devices, TUMPS moves a user's photos to a location that is accessible (with the right credentials) from any internet-enabled device, with a standardized (HTTP) protocol. This would allow, for example, internet-enabled digital picture frames to instantly and automatically show pictures that the user has taken, no matter their geographical location. The picture frame and the camera would not have to know anything about each other, either, since the linking element (TUMPS, in this case) acts as a sort of universal photo storage location that speaks a language that most devices can understand.

# Bibliography


[1] Alkintzis, A. P. K. S., Ors, C. H. A. D. F., Azhyannur, R. A. P., and Otorola, M. WLAN-GPRS Integration for Next-Generation Mobile Data Networks. *Ieee Wireless Communications*, October (2002), 112–124.

[2] Bez, R., Camerlenghi, E., Modelli, a., and Visconti, a. Introduction to flash memory. *Proceedings of the IEEE 91*, 4 (huhtikuu 2003), 489–502.

[3] Chalfen, R. *Snapshot Versions of Life*. Bowling Green State University Popular Press, 1987.

[4] CIPA. White Paper: Digital Photo Solutions for Imaging Devices. `http://www.cipa.jp/english/pictbridge/DPS_WhitePaper_E.pdf`, 2003.

[5] Gast, M. S. *802.11 Wireless Networks: The Definitive Guide*, 2 ed. O'Reilly Media, Inc, 2005.

[6] Group, B. S. I. Bluetooth Specification Documents. `http://www.bluetooth.com/English/Technology/Building/Pages/Specifcation.aspx`.

[7] Halonen, T., Romero, J., and Melero, J. *GSM, GPRS and EDGE performance: evolution towards 3G/UMTS*, 2 ed. John Wiley and Sons, 2003.

[8] Hevner, A., March, S., Park, J., and Ram, S. Design science in information systems research. *Mis Quarterly 28*, 1 (2004), 75–105.

[9] Hyysalo, S. *Käyttäjä tuotekehityksessä*. Taideteollinen korkeakoulu, Helsinki, 2009.

[10] ISO. ISO 15740:2008 - Photography – Electronic still picture imaging – Picture transfer protocol (PTP) for digital still photography devices. `http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=45344`, 2008.

[11] Kindberg, T., Spasojevic, M., Fleck, R., and Sellen, A. The ubiquitous camera: An in-depth study of camera phone use. *IEEE Pervasive Computing* (2005).

[12] Kirk, D., Sellen, A., Rother, C., and Wood, K. Understanding photowork. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06* (2006).

[13] Knutson, C. D., and Brown, J. M. *IrDA Principles and Protocols: The IrDA Library, Vol. 1.* MCL Press, 2004.

[14] Musello, C. *Images of information: still photography in the social sciences.* J. Wagner Ed. Sage Publications, Beverly Hills, 1979.

[15] Neustaedter, C., and Fedorovskaya, E. Understanding and improving flow in digital photo ecosystems. In *Proceedings of Graphics Interface 2009* (2009), pp. 191–198.

[16] Richardson, L., and Ruby, S. *RESTful Web Services.* O'Reilly Media, Inc, 2007.

[17] Sarvas, R., and Frohlich, D. *From Snapshots To Social Media, The Changing Picture Of Domestic Photography.* Springer Verlag, 2011.

[18] Seebach, P. Standards and specs: The ins and outs of USB. `http://www.ibm.com/developerworks/power/library/pa-spec7.html`, 2005.

[19] USB-IF. USB Info: Frequently Asked Questions - USB Cables, Connectors, and Networking. `http://www.usb.org/about/faq/ans5`.

[20] Vainio, J. Bluetooth security. `http://www.mowile.com/bluesec.pdf`, 2000.

[21] Van House, N., Davis, M., Ames, M., Finn, M., and Viswanathan, V. The uses of personal networked digital imaging. *CHI '05 extended abstracts on Human factors in computing systems - CHI '05* (2005), 18–53.