P1 Project

Passport photo program

Group B205

Software

Department of Computer Science

Aalborg University

The 20th of December 2011

AALBORG UNIVERSITET

Faculty Office for Engineering and Science
Strandvejen 12-14
Phone 96 35 97 31
Fax 98 13 63 93
http://tnb.aau.dk

Title:
Passport photo program

Theme:
Programmatic image editing

Project period:
P1, fall 2011

Project group:
B205

Participants:
Dag Toft Børresen Pedersen
Christian Jødal O'Keeffe
Niels Brøndum Pedersen
Aleksander Sørensen Nilsson
Mette Thomsen Pedersen
Rasmus Fischer Gadensgaard
Kasper Plejdrup

Supervisor:
Karsten Jahn

Secondary supervisor:
Marion Berg Christiansen

Finished: 20/12-2011

Synopsis:

In this report it will be documented that there is a problem with passport photos and explained why the rules of passports are needed in the modern community. These chapters will lead up to a statement of the problem which will be used in the development of a solution.
The solution will be a program based on the programming language C, so there will not be any user interface. There will be a chapter describing how the structure of the program will be, and the reason behind the construction. There will also be a section about the reflections made, when the program was being made.
One of the last chapters will concern the testing and conclusion of the finished program. What could have been done better to the program and why does it not work, if that is the case. This chapter will also concern the reflection on the product to see if it actually solves the problem or why it does not solve the problem.
The product of the project is a program which helps the user to crop and scale a passport photo as well as asking the user questions regarding the passport photo rules from the police.

Source of front page picture:
http://www.frederiksberg.dk/Borgerservice/PasOgKoerekort/Pas.aspx

Date: Aleksander Sørensen Nilsson

Date: Christian Jødal O'Keefe

Date: Dag Toft Børresen Pedersen

Date: Kasper Plejdrup

Date: Mette Thomsen Pedersen

Date: Niels Brøndum Pedersen

Date: Rasmus Fischer Gadensgaard

Prolog

This project has been made by Rasmus Fischer Gadensgaard, Christian Jødal O'Keeffe, Casper Plejdrup, Aleksander Sørensen Nilsson, Niels Brøndum Pedersen, Mette Thomsen Pedersen and Dag Toft Børresen Pedersen from Denmark. We are group B205. The project began on 14.10.2011 and was finished on the 20.12.2011, at the University of Aalborg, and were we are studying Software.
Most of us come from Tech College Aalborg, but some come from other colleges. We all chose University of Aalborg because we like the PBL model. PBL stands for Problem Based Learning and it is a unique learning method that is only used at University of Aalborg and Roskilde University Centre (RUC).
The goal of this project is to make us better programmers with the programming language C and make us better at working in groups of 5 to 7 people. The project is about passport photos and our main questions are: Why are so many passport photos being discarded and what can be done about it?

Contents

Introduction 1

This project is based on the subject "image editing" with a focus on passport photos. From this project a product should emerge that will try and deal with a problem within the subject's field. When the product is done, it should be tested and based on the test it will be possible to evaluate the product's capability at solving the problem.
It is a requirement that the program is made in the program language C. The program will therefore be a command line based program.
A part of the project is to document a problem and analyse it, to find the reason behind the problem. When the analysis is done, the product should be designed to handle the problem and thus solve it.
This is the essential of problem-based learning: to find a problem and then solve it and hopefully gaining some knowledge in the process. For this to be an effective model all the group members must participate in the process and acquire knowledge by working with the project.

Problem Analysis 2

This project is based on the idea that the rules regarding passport photo is the reason why many people get their photo rejected at the civil service department. What can be done to make it easier for people getting their passport photo approved? To answer this and find a solution to this problem it is necessary to take a closer look at the rules, to understand why these rules are so strict and why there is a need for these rules in the first place.

2.1 The reason for passports

Going back to medieval time, the state or kingdom at that time was the ruling body and restricted the "means of violence" from its citizens. The reason behind this restriction was to be able to control its population better by giving out special permissions to certain individuals. They acted as the will of the state. The state did not want its citizens to leave its boundaries without its explicit permission to do so. This permission usually had the form of a document of some kind, that identified the person as a representative of the state and that he was under the protection of the state.
This was the first passport-like document that existed though it lacked identification pictures. The reason behind this restriction was because the state was afraid that its citizens might get influenced by outside forces. The state feared that this influence might affect the traveler in a bad way, be it getting robbed or incited to go against the state. A side effect to this restriction was that it also served to identify individuals to their particular state [Torpey, 2000].

A little funny anecdote from the Hebrew Bible Nehemiah 2:7-9 making mention of a passport-like document: "I also said to him, “If it pleases the king, may I have letters to the governors of Trans-Euphrates, so that they will provide me safe-conduct until I arrive in Judah? And may I have a letter to Asaph, keeper of the royal park, so he will give me timber to make beams for the gates of the citadel by the temple and for the city wall and for the residence I will occupy?” And because the gracious hand of my God was on me, the king granted my requests. So I went to the governors of Trans-Euphrates and gave them the king’s letters. The king had also sent army officers and cavalry with me" [Bible].

The passport had evolved from a simple letter of safe conduct to a more personalized paper identifying the person carrying it as a member of the state, but also as an individual since around World War One. The modern states, compared to the more prejudice medieval states, had relaxed their border control and had seen the potential in foreigners and the trade that could be gained from other states. But the beginning of World War

One in August 1914 changed all that. The states became closed in and began to rigorously enforce the passport law, greatly restricting foreigners' movement in the state. This in turn also posed a problem to inhabitants of the state, should they not have the correct papers also known as passports. As the war raged on, Germany enforced an even stronger regulation to the passports so they had to include a personal description, photograph and signature of the bearer along with an official certification from the state, from which the bearer belongs, which identifies the bearer as actually the person on the picture. Even though World War One had ended, remnants of it still remained in the minds of the states and thus the rules and requirements of the passport remained [Torpey, 2001].

In 1985 two big changes were made to the passport. Firstly a machine-readable text was added to the passports so the people at the airport, or other places which use passport as identification, no longer had to write all the information from the passport into the system, but could now just scan the information from the passport into the system. The second change was the introduction of the Electronic passport. This passport is also called Biometric passport [William, 2009]. This passport came to life 5 years after the attack on World Trade Center on the 11^{th} of September 2001. This attack showed there was a need for more security regarding the documents of travel. The United States were the first nation to demand that in the future passports should have a digital photo on a chip inside the passport so it would be harder to forge a passport [Teknologi-Rådet].
This later resolved in very strict rules regarding what a passport photo should look like. It is important for the nations to be sure that the person who is trying to enter their country is the person he or she claims to be. The biometric passport was introduced in Denmark the 1^{st} of August 2006. The reason why this passport was introduced was more strict demands to security and identification especially from The United States. The new rules for the passport photo was introduced by the National Police in 2006 [Carlsen, 2009]. They are listed later in the report in section 2.3.

2.2 Biometric Passport Photo

After the attack on world trade center in 9.11.01 there has been an increased demand from the USA regarding the security surrounding passports and a new technique has been developed. Essentially it is a digital version of the passport photo stored on a microchip that is in turn placed inside the first page of the passport. It was first introduced in the fall of 2006. This new security measure is to make it harder to forge a passport. This will potentially lessen the traffic at the airport terminal, if the passports can be made completely digital [Politi, b]. The already strict requirements for a photo to be accepted as a passport photo has been increased because of this, as an example: The person has to face the camera completely, if the face is pointing just a bit to either side then the photo will be rejected. If the technique is implemented into the security protocol that is used today then the digital photo is used to identify the person on the picture by scanning picture and then scanning the real person's face to confirm that they match. The microchip will also contain a digital fingerprint as yet another measure of security. The increased security measures have been made to prevent identity theft, forgeries and if used correctly also increase effectiveness of the flow, in for instance, airports. Because the requirements to passport photos have been made stricter it also makes producing a passport more difficult and it can be frustrating for the general consumer to acquire a passport [Teknologi-rådet].

2.3 Photo Requirements

There are several requirements to a passport photo in Denmark [Politi, a]. The same rules apply to passport photos of children.
General rules

- The photo must look like you, be taken straight in front of the person and suitable for scanning.
- The photo must be 35mm x 45mm and the space between the chin and the top of the head must be between 30 and 36mm. The photo can either be in color or black and white.

Requirements to the Motive

- There may only be one person in the photo and no other objects.
- The face needs to be in the middle of the picture, and the background must be light without any motives on it.
- The eyes may not be closed or covered by hair or other objects.
- The head has to be straight and the person needs to look straight into the camera, with their mouth closed.
- The photo has to show the face and the top of the shoulders.

Requirements to the Photo Quality

- The light has to be even and no shadows may be present on the face nor the background.
- Photos, where the person has white or light hair or wear a light colored scarf, need to have a background in sharp contrast.
- The photo, when printed, needs to be in good quality without any holes, stamps or other damages.
- The photo needs to be a photograph or printed on photo-paper of good quality and with very high resolution.
- Red eyes are not allowed on the photo
- The photo is not allowed to be pixelated.

Requirements about Glasses

- The eyes need to be completely free of the spectacle frame and the glasses are not allowed to have reflections and tones.

Requirements about Headgear

- Headwear is only allowed if there is a religious reason behind it.
- Forehead, chin, and cheekbones need to be visible.

The same rules do not apply to every country. For example there are very few countries besides Denmark which accept black and white photos, and the size of the passport photo is also very different from country to country - in Canada the photo needs to be 5cm x 7cm. But most countries are using the 45mm x 35 mm standard size [Tomgreen, 2009].

2.4 Problems with Photos

Through time the rules for ID photos have changed a lot, and it can be hard getting ones photo approved. Because of this fact many people tend to get the photo done by a professional photographer and that could be expensive compared to the price by taking the photo at home. The price for 4 pictures is around 100 DKK, that means a family of 4 must pay around 400 DKK just for the photos [Hafnia Foto, 2007]. Getting the passport made costs 600 DDK for people above the age of 18, while it is only 115 DDK for people under 18 years and lastly 350 DKK for people above 65 years of age. This means, a family consisting of 2 parents, and 2 children (one above 18 years, and one below 18 years) must pay a total of 2350 DDK for passports for all of them [Politi, 2009]. But the prices for getting the passport made is the same independent of where the photo was taken.

Some people try the photo booths that can be found at train stations, airports or local malls but without luck. Numbers show that up to 8 out of 10 pictures taken in a photo booth do not make it through the check and get discarded [Carlsen, 2009]. It is a problem that photo booth pictures rarely get approved at the citizen service department and people will have to retake their photos. The reason could be that the customers using the photo booths do not know all the rules and by that get the photo discarded, all though some photo booths have posters with the rules regarding the passport photo.

An alternative is to visit a photographer that knows all the rules about passport photos but that is more expensive than a photo booth or taking the photo at home. There is a small chance that the photographer actually does not know rules and therefore does not get the photo right [Carlsen, 2009]. The source is from an earlier state of the passport rules (from July 2009), but the article describes the problems with new rules, and not the specific rules. The photos for ID can be taken at home but that means the user needs to have knowledge about the rules that applies to pictures for generally accepted ID as drivers licence, passport and international identity card.

To help users with pictures for ID some websites make a business out of offering services where the website runs a check on the photo and either approves it or discards it [ePassportPhoto]. If the picture gets discarded then the user is told to cancel or try again with another picture, but the websites that offer these services still do not draw any guiding lines, so the customer understands what the rules for passport photos are. Using this kind of service/program can also backfire if the program you are using has not updated the rules or is following a different set of rules. The scenario could also be that the program actually does not check for all the rules that go for passport photos. The problem is that the services on the Internet might not be updated to the latest rules if the service has been abandoned.

This can the be checked by researching the program's functions and eventually try it out. Through this research it is possible to gain an understanding of what functions a possible solution would need to have. On The Go Soft is an example of a program that claims to help its users in the process of making a picture for ID but actually only helps the user find and crop the pictures [On the go soft]. On their website some of the rules for passport pictures are displayed, but there are still some of the rules that are missing. This can cause the customers' pictures to not be usable even when they use this program and that is not the wanted effect.

2.5 Passport photo programs on the market

In this section a closer look will be taken on other existing programs that can help a user make a passport photo. This will tell what other programs on the market offers the user in some way. In the following section there will be a short description of each program and in the end a summary of what features the program of this project will have, what it will not have and what might be improved based on what other programs offers.

2.5.1 CNET Editors: CamToPrint

This program helps make the passport photo in the way that the users want it. They start by choosing a photo they want to make into a passport photo and the program helps center the head and offer tools to crop it to the right size. The program also offers tools to help adjusting brightness and contrast. In the menu the passport photo requirements for The United States of America or United Kingdom can be seen. When the users are finished with the photo they can choose to print it directly from the program or save the photo as a file for later use. But this program offers more functions than just tools for making a passport photo. It offers some other photo tools, such as utilities for making photo albums, calendars, greeting cards, and collages. The program works on Windows and is free [CNET Staff, 2011].

2.5.2 OnTheGoSoft

This program helps create the passport photo with the functions to select passport photo size, rotate and crop the photo. The size formats that the user can select, goes for the US or Canadian standard or custom passport photo size. Besides this, the program also helps with the printing of the photo so the user gets a lot of photos on a single page. OnTheGoSoft is free and works on Windows [On the go soft].

2.5.3 ePassportPhoto.com

This is not a program but rather a website that helps the customer with the editing of the photo by allowing access to some tools that the user can use to scale the photo into the right format and it gives a good explanation on how to edit the photo. After editing the photo the user can download the edited photo and print it from their own computer. The site is free to use, but offers more help if the user is willing to pay a fee of $9.99 dollars. For this price they get a passport photo expert to inspect the photo and check that all things required in the photo are in order. If they do not get the photo approved they will get a refund [ePassportPhoto].

2.5.4 Passport Photo Studio

This program helps by cropping the photo to the right size but the user has to help it find the head with a circle where the head needs to be in the centre of the circle. After this they have to choose what passport photo rules the photo needs to follow either the ones in USA or the ones in the UK. This program is free and operates under both Mac OS X and Windows [Grogware LLC, 2011].

2.5.5 ePhotozine

This is not a program but a set of guidelines on how to make a passport photo in a photo editing program such as Adobe Photoshop. It gives a list of the rules required in a passport photo [ePhotozine, 2003].

2.5.6 Recapitulation

Based on the requirements for passport photos and what the different programs offers, a table of functions have been made to give a better overview of the different programs. On the horizontal lines are the different programs. On the vertical lines there is a comparison between the different ways to help the user making the passport photo. The comparison can be seen on Table 2.1.

Included in product **Solutions**	CNET Editors:CamToPrint	OnTheGoSoft	ePassportPhoto.com	Passport Photo Studio	ePhotozine
Installable program	x	x		x	
Help with setting the head	x	x	x	x	
Help with setting the scale	x	x	x	x	
Guide to how use the solutions	x	x	x	x	
Guide to taking the photo			x		x
Guide to making the passport photo					x
Checking the photo			x*		
Helping printing the photo	x	x		x	
Display image	x	x	x	x	

*The photo is only checked if the fee is paid.

Table 2.1: Gives an overview of the function provided in the different software solution.

2.6 Thesis Statement

It has been discovered that there was a problem with a lot of passport photos being discarded. So a solution is needed to solve the problem. The problem is that the rules for the photo are very strict and that makes it very hard to make a photo that will be approved. A program that can help the user get his/her photo approved more easily and check for themselves if the photos are following the rules, is the preferred product of this project. The project will take the thesis statement as the starting point for finding the right solution. From the process analysis the thesis has been found:

- Is it possible to develop a program that helps the customer through the process of making a passport photo?

Sub Statement

The sub statements help to find the solution for the thesis statement. These sub statements describe some of the statements trough should be answered in order to find the solution for the thesis statement.

- **How does the user interact with a program that is command line based?** Since the project is using a language that makes it very difficult to make a graphical interface, the interaction with the user will be through command lines. The solution should work with JPEG-files because this is one of the most common image formats for personal images and this will most likely be the format the user has the photos in.

- **Which functions are needed to implement the rules of passport photos?** Rules for passport photos state that they need to have a specific height and width, and that the space between the top of the head and the chin needs to be a specific height too. The program should therefore be able to scale the photo according to the rules so the user's photo will not be discarded, because it is not following the rules. There are more rules that will need implementation before the program actually will help on the problem.

- **How can it be checked if a photo is of quality high enough ?** There are rules for the quality of a passport photo. The quality is very important when the passport photo is to be approved. The product of the solution of the thesis statement should check or secure that the photo is in a quality high enough and through this prevent further problems regarding the photos quality.

From the thesis statement and the sub statements some product specifications have been designed, see Specification requirement at section 4.2 for a detailed explanation of the product specifications.

2.7 Method

The method used in this project is the Aalborg PBL (Problem Based Learning) model. Working with this model puts the students in charge of their learning by letting them find a problem in the real world, related to the courses they are taking, that they want to try to fix or improve upon. The problem is researched and documented. By utilizing the knowledge gained from their courses, the students will then develop a product which they believe will solve or improve the conditions for the problem. Their work is still under supervision to give a broader perspective to the product and help the students improve it. In short Aalborg PBL is about teaching students some basic knowledge about their course and then let them try to use it in a real world problem [Aalborg University, 2010].

The Aalborg model is used, because problem based learning in groups makes use of a lot of great tools to solve a given problem. The participants learn how to deal with a real problem, and find a solution while working together.

To find information and gain knowledge about this project the Internet was of great use. Through the AUB (Aalborg University Library) database articles and books were found. Because of its focus on the academic aspects Google scholar has also been used. When using the Internet to find sources it is important to use sites that are trustworthy.

The target group analysis will be used to limit the project to a group of people and based on them determine how the program should be developed. For more information read the target group analysis at section 4.1.

Theory 3

This section of the study is about image editing and how it has developed through time up to now, and its impact on the credibility of photos. It describes what it is and how it can be used. The next part of the section describes how photo analysis is today and what it is. The last part of this section is a short description of the libJPEG, other possible libraries, what it is and how it is used in this project. Knowing how image editing has changed through time and how it is today make it easier to determine where the development is heading and where the project should be heading, and to not make some of the same mistakes that might have been made in the past.

3.1 Image Editing Through Time

Before the invention of the computer, image editing was not widespread as the cost and skills needed to make these images look good were very high. Because of this the art of photo editing was mostly done by professionals [Fourandsix]. With the computer invented and after some improvement, further development of the camera and the digitalization of images have ushered a new era in photo editing. With computers it became possible to develop functions that can edit certain features in a photo. Because the computers have storage, it was possible to store the complex function, call and use them without a deeper understanding of the editing functions construction.

At first the act of editing a photo still required skill, advanced knowledge of computer software, programming and commands. But as computer technology developed it also became easier for the consumer to use, as the interface to which they were to give commands improved and became more accessible. The editing software got a more developed interface so it became easier to use and therefore the consumer cluster, grew which boosted the profit. This was caused by the price of having professionals working with the software. With the more developed user interface, more and more began using the software [Coleman, 2007]. The sudden burst in photo editing software has made it easy for anyone owning a computer to edit their own, or someone else's photos, but the many options, buttons and functions can make it difficult for the inexperienced user to comprehend. In spite of these experiences, the program of this project should therefore be easy to understand, and the user should be guided through the process

3.2 Image editing today

Modern image editors, such as Adobe's Photoshop, feature very advanced and complex functions compared to earlier years. The newest version available of Adobe's Photoshop, version CS5, features functions such as Content-Aware Fill, which essentially is able to remove a detail or object from a photo, and intelligently fill in the space behind the object. Compared to command line image editing, graphical user interface(GUI) based editors are easier and more intuitive to use for the end user. The simple philosophy behind GUI is: "What you see is what you get" (wysiwyg), meaning if it has a function then you can find it in the interface and using the cursor increases precision compared to writing a set of coordinates. If you were to convert a highly advanced image editing tool such as Adobe Photoshop CS5 to a command line tool, you would have thousands of commands to remember. Also the GUI provides essential tools for precise image editing such as various brushes and selection tools. It would be very complicated to edit minimal correction in photos in a command line tool.

That being said, using command line can easily perform simple corrections to thousands of files very quickly. Imagine a program when being run in a folder with images, resizes all images in this folder. This would be faster than resizing them individually. Because it is relatively simple to program, Photoshop has implemented a feature called "actions" where you can set up an action to resize a batch of images quickly.

3.3 Photo analysis

There are many programs today, which can analyse pictures automatically and make corrections, adjustments and many other features to the image. There are programs which can analyse surveillance photos, and find a certain person [Foresti et al., 2003]. This can help the authorities find a specific person using the surveillance systems in the country. To do this manually would take too long to be effective and the possibility of spotting a specific person in a crowd is very small. This is an example of how automatic photo analysis can handle an enormous task both faster and more precisely than a human could. Another example relevant to the project is the biometric passport which has its own section in the report. In this section the procedure regarding the use of biometric passport and why it is used in modern society will be discussed.

With the help of these auto functions developers can optimize their programs by eliminating the human factor, granted the function works correctly. Photo analysis contains a lot of different procedures and purposes today, and is used in many different business. The problem of photo analysis is, that it can be difficult to analyse pictures, because they are very different. The program of this project will have a few photo analysis functions, such as a means to analyse the size and DPI of the picture, face recognition, brightness and sharpener.

3.4 Choice of library

As it is needed to work with images in the program of this project, the program must have the ability to handle images. The most common image format is JPG/JPEG, so the program should work with this format. This is something which others have done plenty of times already, and there are therefore no need to write a library specific to this project - this would also be a lot of work, for something only remotely relevant for the project. There was taken a look at existing libraries, and chosen the library that was most relevant for the program. A library is an extension to the normal functions in C. The library needed for the program of this project, is a library which can:

- Work together with C
- Read, edit and write JPEG pictures
- Be easily implemented

This section contains a description of four different libraries for working with JPEG files in C.

3.4.1 libJPEG

libJPEG is mentioned as the largest and most important contribution to the JPEG format on the Joint Photographic Experts Group (JPEG). JPEG also recommends libJPEG for applications that want to work with JPEG images [JPEG]. The main advantage in libJPEG is that it is very well documented and widely used. It is written with C as target, and can therefore be directly implemented in C code. A disadvantage is that there is no sample file to see how it works - you will have to, program the sample file through the documentation.

This library is especially designed for C, so the first requirements for the library is fulfilled. The library can read, edit and write JPEG pictures, but it is not in the standard form easy to implement in the code. There must be written a function to use the library, but this is no problem, because many other people have written these functions. The library with a support function do therefore fulfil all the requirements for a library to this project.

3.4.2 libJPEG-turbo

libJPEG-turbo is a derivative of libJPEG, which focuses on speed, mainly on x86, x86-64 and ARM based systems. The main advantage of libJPEG-turbo is, of course, the speed - as it has the ability to run on 64-bit systems. This is, however; not something that is needed in the program, given the small size and memory use.

libJPEG-turbo is working with both C and JAVA, so the first requirement is fulfilled. It is also able to read, edit and write JPEG images, which is the second requirement. But libJPEG-turbo supports ARM based systems, which necessarily means, that there is a lot of extra code in the library, that we will never use, as we do not work with ARM based systems. It can therefore not be easily implemented.

3.4.3 OpenCV

OpenCV is described as "a library of programming functions for real time computer vision" [OpenCV] which makes OpenCV the most advanced library that known today. It seems very large and very unnecessary for the program's needs. Again this is by far the advantage of OpenCV - it can properly manipulate images in any way possible.

OpenCV is a cross language library. Hence, it works together with C, but also many other programming languages. It will work in C, but is not easy to implement in C, because it is not directly written to C. OpenCV does provide many functions, and can read, edit and write JPEG pictures and much more. The many functions provided by OpenCV is too many functions for this project, and they can make it hard to find and use the right functions, when there are so many. Bottom line is, that OpenCV can work together with C and it provides all the necessary functions, but it is not easily implemented.

3.4.4 ImageMagick

ImageMagick seems like a simple image library for JPEG, but as far as the program's needs, libJPEG seemed more usable due to the fact that ImageMagick, like OpenCV has a lot of functions that the program does not need.

ImageMagick is a C library, so the first requirement is fulfilled. It is able to read, edit and write JPEG images - so the second requirement is also fulfilled. The third requirement is also fulfilled - ImageMagick is as easy to implement as libJPEG. However, the fact that ImageMagick contains a vast amount of features which we do not need, was the reason we did choose libJPEG over ImageMagick.

3.4.5 Recapitulate

In the end libJPEG was chosen because it was the library recommended by the JPEG, and it is very simple to understand how to read and alter pixels in the library. libJPEG is also written in C, where OpenCV for example is cross-language, and supports both C, C++ and Python. It also seemed as if libJPEG was the easiest, to get started with - it only took a couple of hours before a program that could open an image was created.

3.5 Theory of libJPEG

libJPEG is a library, which makes it possible to read and work with jpeg images in C. The library is made by the Independent JPEG Group [IJG]. The library contains a lot of header files, containing the functions of libJPEG. The header file jpeglib.h contains the "main library". It is possible to include this in code of own projects, and the jpeglib.h will include the other libraries. JPEG is a compressed image standard, which means that it is not possible to read the pixels directly out of the bit structure [Wallace, 1992]. Jpeglib decompresses the data from the JPEG file, and load it into a large string array. The libJPEG can also compress the data again, and write it to a JPEG file. The use of libJPEG in this project will be to read and write data to a JPEG file. The string which contains the data from the read file, is changed and while storing the new picture, only parts of the string are saved, so the image will get the right size.

3.6 Pixel to Inch

Digital images use pixels to determine the height and width of an image. The appearance of the image on the monitor is therefore relative to the resolution of that exact monitor. For example, an image with the dimensions 400x400 pixels will appear large on a monitor with a resolution of 800x600 pixels. But that same image will appear very small on a monitor with a Full-HD resolution: 1920x1080 pixels. When the image is printed, the height and width are measured in, for example, centimeters or inches. Because the height of a passport must be exactly 45 mm, and the width must be exactly 35 mm, it is necessary to convert the height and width from pixels to inches. When working with resolution of picture, inches are the standard unit instead of millimeters. The inches can easily be converted to millimeters later. When the width in pixels is known, it can be determined how many pixels are needed per inch. On the monitor this is called pixels-per-inch (PPI), but when it is printed, it is called dots-per-inch (DPI), which can also be used to determine if the image is of sufficient quality. For example with only 5 pixels per inch, the image is simply of too low quality. In the program a minimum of 400 DPI has been set, because this is the minimum requirement in the Netherlands [Ministerie van Binnenlandse Zaken en Koninkrijksrelaties] - there is no specified minimum DPI in Denmark, or in the EU, therefore the Dutch standards are used. Now that we have the measurements in inches, we can easily convert this to millimeters. This is done by simply multiplying the desired result in millimeters with 1/24.5, which is how many millimeters are needed for one inch. Now the result is in millimeters.

Design 4

With the research done the actual designing of the program can begin. This section describes how the program's initial steps were done. First of all the target group analysis. After this comes the solution suggestions, the choice of solution and the specification requirements to the program. After this comes the flowchart which gives a broad overview of the structure of the programs. From the flowchart a pseudo code is made which describes the code structure in greater detail.

4.1 Target Group Analysis

This project is using the Gallup Compass to describe the target group of the program, and in that way design the program to fit the target group. There are higher chance for success, if the product is designed for the target group from the start.

The Gallup Compass is a marketing model used to determine whether you are trying to reach traditional or modern and individual people, or society oriented kinds of people.

The Gallup Compass consists of 9 segments:

- Modern oriented
- Modern society oriented
- Modern individually oriented
- Society oriented
- Individually oriented
- Traditionally oriented
- Traditionally society oriented
- Traditional individually oriented
- Center

4.1.1 Description of the Segments

In the following section the different segments will be described, the segments are from the Gallup compass. The Gallup Compass can be seen on Figure 4.1.

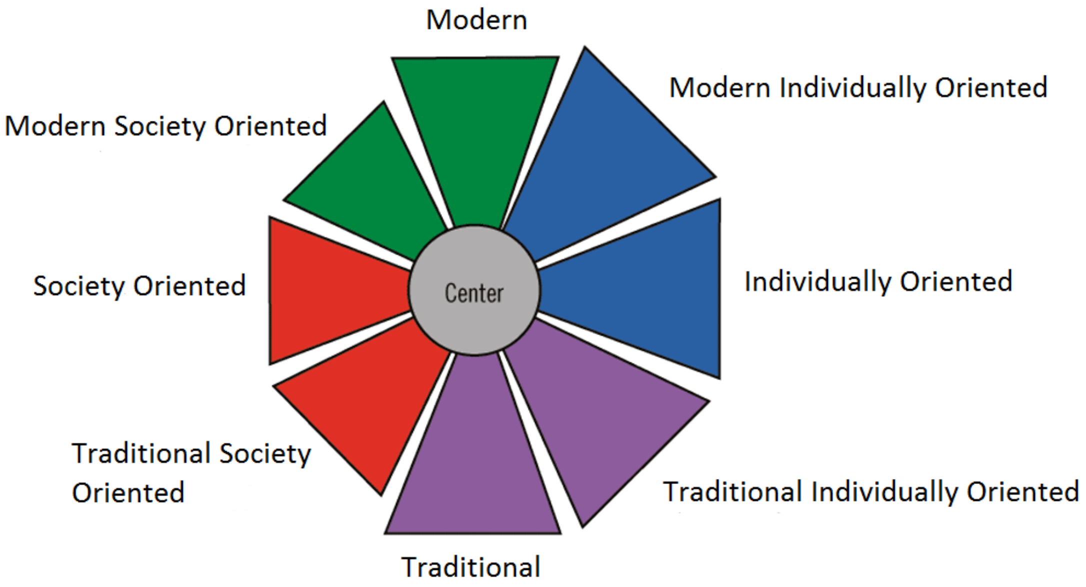
Modern
Modern Individually Oriented
Modern Society Oriented
Society Oriented
Center
Individually Oriented
Traditional Society Oriented
Traditional Individually Oriented
Traditional

Figure 4.1: The Gallup Compass

The Modern Segments (Modern society, modern individual, and modern)

The "modern segments" contain people who tend to be well educated and earn more than 300.000 DKK a year. A big amount of this segment are men, and most of them have not founded a family yet. They love new technology, especially when it comes to cell phones, tablets, and similar gadgets. The segment mostly consists of people between 20-49 years of age. Their political view depends on whether they are society- or individually oriented. The society oriented are socialists or social-liberals. They primarily vote for The Danish Social Democrats, Socialist People's Party, The Danish Social-Liberal Party or The Red-Green Alliance. The individual oriented are more right-oriented so they tend to vote for "Venstre" (A Liberal Party), The Conservative People Party or The Danish People Party.

The Traditional Segments (Traditional society, traditional individual, and traditional)

People in the traditional segments generally tend to be more conservative when it comes to technology. Some of them are afraid of it, and wish it would not develop as fast as it actually does. They praise family values very high. The wages are not very high, less than 400.000 DKK per household. The traditional segments mostly consist of the elderly. Their political view is very dependent of what background they have. If they are children of academics or other people in the higher social layer they tend to vote for The Conservative People Party or "Venstre". If their ancestors were belonging to the middle or lower social layer they will tend to vote for the left-oriented parties.

The remaining Segments (Society oriented, individual oriented, center segment)

These segments are like a combination of modern and traditional, except the center segment. The center segment is the average segment. They do not have any significant opinions that may turn them into one of the other segments.

The society oriented segment consists people who are very much concerned about other people. Ecology, nutrition, compassion, and responsibility have high values for this segment. Most of them are between 40 and 59 years of age, and especially women are represented in this segment. Their political views mostly lie on the left oriented parties.

The individual segment is mostly men, who tend to have limited interest in society in general. It has an over representation of younger men, especially from Jutland. Their political views mostly lie on the right oriented parties.

4.1.2 Choice of Target Group

This product is aimed at people within the modern segment, the modern individual segment, and the modern society oriented segment. It points to users with computers using Windows as operating system (OS), and with normal knowledge of Windows as minimum. The knowledge is needed because what the project is all about, is using a command line function in Windows to check whether the passport photo is correct or incorrect. People, who are not experienced with windows, would probably not use a command line based tool [Gallup].

The target group analysis is made to find out what kind of people the project is trying to reach. This information can now be used to design a program, which will be so optimal as possible for the user. The analysis is used when we develop the program to specify how advanced the program can be, while the user is still able to use it without too much difficulty.

It is concluded from the target group analysis that the program should be designed for the modern segments. It specifies for Windows, and the program should be a tool which makes a task easier or better performed. It is expected that they are familiar with technology and windows, so it is possible to make a program which is command line based and is normally advanced.

4.2 Specification Requirements

The program has some features, which are essential for the program to work, these are catalogued as "targeted features". The other features the program does not necessarily need to work, is catalogued as "Nice to have" features.

4.2.1 Targeted Features

These are the features that the program will have.

Written in C: The requirements of the project is, that the solution must be a program written in C. It must have a command line based user interface.

Guide the user: The program will have a little readme file, or other form of guide, that will tell the customer to take the picture they want to use as their passport photo, and rename it to a file name, that matches the program's structure. This is an easy way to simply have a preset name for the file the program has to load.

Pinpoint head: With help from the customer, the program has to place some lines on the photo to frame the head thus centering the picture on the person and giving the program a pinpoint. When the specified lines have been set, the photo will pop up, so the user can see if the lines are placed correctly, if not the user will have to rearrange the lines so they fit.

Scaling: When the pinpoint has been made the program will then crop the picture so the head, shoulders, and part of the background is all that is left. Then it will scale the picture to the size of a passport photo, if needed.

Display image: As the user opens the program and has successfully renamed the file, the image will then pop up. This will make it easier for the users to know what is going on, and what they have to look for in the picture.

Dots per inch: For the photo to be accepted as an approved passport photo, it has to meet certain quality specifications. So the program will also have a feature that determines whether or not the photo is of a good enough quality. By measuring the dot per inch (DPI) the program will tell the customer if it is not high enough quality, and then the photo will not be approved.

Checking: When the photo is done the program will show a preview to the user so he/she can go through a number of questions to help check the photo to determine whether the rules for passport photo are met.

Finish: This is the last part of the program. It is simply a little preview of how the picture looks, when it is done without all the guidelines.

4.2.2 Optional Features

The program does not take into account that an image might be turned to one side. So a "nice to have" feature would be a rotate function that will solve this problem. The program of this project is based on European requirements for photos, but some countries have different requirements, so to further increase usability a plug-in or a built-in feature to choose from different countries would be preferable. Furthermore, due to the bit-structure of black and white images they have not been included in this project.

4.2.3 Will not Have

Given the limited visual functions of C programming, the program will not have a graphical user interface, only lines which to write on. Furthermore, the program is intended as substitute for an image expert, so it will help the user check whether or not the photo follows the requirements. For the same reason the program will not print the photo.

4.3 Solution Suggestions

In this chapter the different solution suggestions to the problem with passport photos that have been defined in the thesis statement in section 2.6, will be explained and the different circumstances regarding the solutions will be clarified. Out of the three suggested solutions, one will be selected and two others will get discarded. The selection will be discussed in the sub section "Choice of solution". The three solution suggestions below are chosen from the mindmap figure B.1 in appendix B. The mindmap was made in the idea generating phase and it is a map over general ideas and their relations. They were chosen because it was the best suggestions on the mindmap.

4.3.1 Extension for Existing Programs

A solution to this problem could be an extension for another program or solution that already exists. An extension that takes care of the missing functions in the existing program. The extension will need to determine if the person is looking straight, control if the person has red eyes, control the background to see if it is the same color and other basic function that is needed to get the passport photo approved. The already existing solutions offers a program that helps crop and adjusting light contrast, but lacks the ability to recognize the head and determine, where in the photo the head is. Here would the extension come in handy and do the work finding the head and determine where in the photo it is. This will work through algorithms and input parameters and return the position of the head and if there are more than one head, discard the photo.

This solution can be possible because the program already knows the position of the head, and with that in hand, and with the help of other algorithms, it can compare the color of the iris with a color scheme. Through this comparison the extension would make the decision if the photo should be discarded or approved. The extension should also be able to determine, if the quality of the photo is good enough or the photo will need to be of better quality. There are still many functions that could be added with this extension as a solution, but there is a deadline for the project. Because of the deadline there is not enough time to understand and develop the algorithms needed in facial recognition. This should have been done and implemented in the extension, but again because of the time factor this solution must be demarcate, so it is possible to make the product within the time schedule. There is also the restriction, that the product should be based on the language C. Because of this restriction it is needed to take a look on the other program the extension should work with, and think of a solution to how the two languages will communicate together. In the understanding of the two languages there is also the dilemma with C and JPG/JPEG files. Another problem with this solution example is that the company's approval is needed, if the program is not an open source. This means that the solution idea first must be "sold", in the meaning that the company will have to agree on the idea of a extension for the product. This problem is only encountered if a test of the extension is needed, which is the case. It is possible to construct an extension that does not need another product as foundation, but then there is no guarantee that it works with other programs as the first idea was.

4.3.2 Program Based Solution for Home Users in C

This solution builds on the idea about making the whole program in the same programming language and using libraries that offer the functions needed handling JPG/JPEG. Because it is the plan to make the whole program it is possible to write the program in the language C, as one of the requirements for this project dictates. A problem with the use of languages C, is the difficulty in the handling of JPG/JPEG, but this problem can be solved by using a libJPEG. But there still lies some work in understanding this library and figure out how it is used. This means there must be set some time off to the process of learning how to use the JPG/JPEG library. The program should also have the same function as the extension solution, but again the deadline comes into consideration. The program with the deadline could be solved by making a dialogue with the users and ask them about the different requirements that goes for a passport photo. This way, the program will solve the main problem and if it gets done before the deadline, there will be time making the program check the requirements automatically by itself. This solution does not meet the same problem as the extension solution regarding testing, because there is no need for another program before it is possible to perform the tests. Using this solution means that the target group needs a basic knowledge using a computer. Therefore, when using this solution the target group must be taken into account. An extension to the solution could be making it so the program can load different files that contain rules set for different countries.

4.3.3 Step by Step as Application for Smartphones

This solution goes by the idea of making a guide for smartphones, that helps the customer through the process in making the passport photo. This means as long as the customers have their smartphone and this solution they will be able to make a passport photo and hopefully enlarge the chances for success. This application (App) solution requires learning in using the language C for programming on smartphones. Therefore, this solution requires modifying the time schedule, so there is set some time off for the learning process that can be steep. The application solution is not a complex programming solution, because its purpose only is to guide the user and instruct in what requirements there are for passport photo. Because this solution is not complex and actually is a long dialogue between the user and the program, it does not require a lot of coding to get the difficult functions working. An optimized version of this solution could involve making the application, so it could do the image editing and automatically control the photo and determine if the photo goes by the rules, or simply would not work.

4.3.4 Choice of Solution

From the understanding of the three solutions and the idea they are built on, it is possible to determine which of the solutions are the best to solve the identified problem. To determine which solution is the best for the dictated problem, it is a necessity to compare the solutions with the standards, set for a possible solution to the problem. The first criterion to the solution is that, it should be possible to solve the problem in the programming language C. The three suggestions could be made in the language C, but there will be some circumstance for the three suggestions to work. The first solution suggestion will need to work across platforms and this can be a difficult operation to do in C. The reason for cross platforms is that this solution will need to work together with another program to fulfil its purpose. The second suggestion will need to import a library for handling JPG/JPEG files. The mentioned library has already been made and there is no need to invent it again, if it is possible to use an existing library. The third suggestion requires a certain amount of knowledge regarding making applications with C, normally it is C++, C# or Java that is used for making applications. So the most optimal solution to use regarding the language C is the second solution suggestion. The reason for this is that the two other solutions need a lot more preparation rather than the second solution even when it needs a library for JPG/JPEG to work.

The fourth criteria at the specification requirements in section 4.2 is that the program should be able to scale and edit the DPI (dots per inch) setting of a JPG/JPEG image file. The DPI settings are to secure the quality of the printed photo. This cannot be done in C without the library that can handle JPG/JPEG. All three solutions will need this library to work with JPG/JPEG but solution three will cause problems regarding where the library should be placed in the smartphone's system. But through the library, libJPEG, a C program will gain the ability to scale and crop an image. Therefore solution one and two are more suitable for this project rather than solution three. The point for the product is that the chosen solution should help the user getting their photo approved. The first solution helps the user checking the rules and sees if they are fully met, same goes for solution number two. It is a slightly different case with solution number three that, at first, should only guide the user with the process. This means that the third solution actually does not do any editing on the image. Because of this factor solution one and two classify much better as a solution. A problem with solution three is that it will need a form of user interface, so the user can navigate through the program and use its functions.

From this context it is possible to see which of the three solutions that is best suited to solve the problem. The third solution is the worst of the three, while solution one and two are almost equal. But because solution one will need to work together with another existing program and what problems that brings, it has been chosen to use solution number two "Program based solution for home users in C".

4.4 Program Planning

To develop the program, which could fulfil all the requirements for the program, a flowchart was made. A flowchart can give an overview of the structure of the program, but also an outline for which procedures that have to be made for the program. The flowchart is a developer tool, which can be used in the very start of the process, to get an overview and an understanding for the flow in the program. The flowchart can be seen on this figure 4.2.

The first thing the program needs to do is to start up, hereafter it needs to get the photo into the program, so it can work with the photo and show the photo before the user starts doing something to the photo. Then the user needs to put in two points where the face needs to be enclosed by, and make a box with the two points so the user can see if the head is within the points, and if the points are right with what the user has chosen and scales the area to be in right format. After scaling, the program needs to check if the DPI (dots per inch) is right, if not it needs to say so and stop the program. DPI is the setting for the printer on how many dots per inches it is going to print the photo with, and thereby the quality of the print. Thereafter, the program needs to ask a series of questions. If the answers are right it will make the photo, and end the program. Otherwise it will tell what is wrong and end the program.

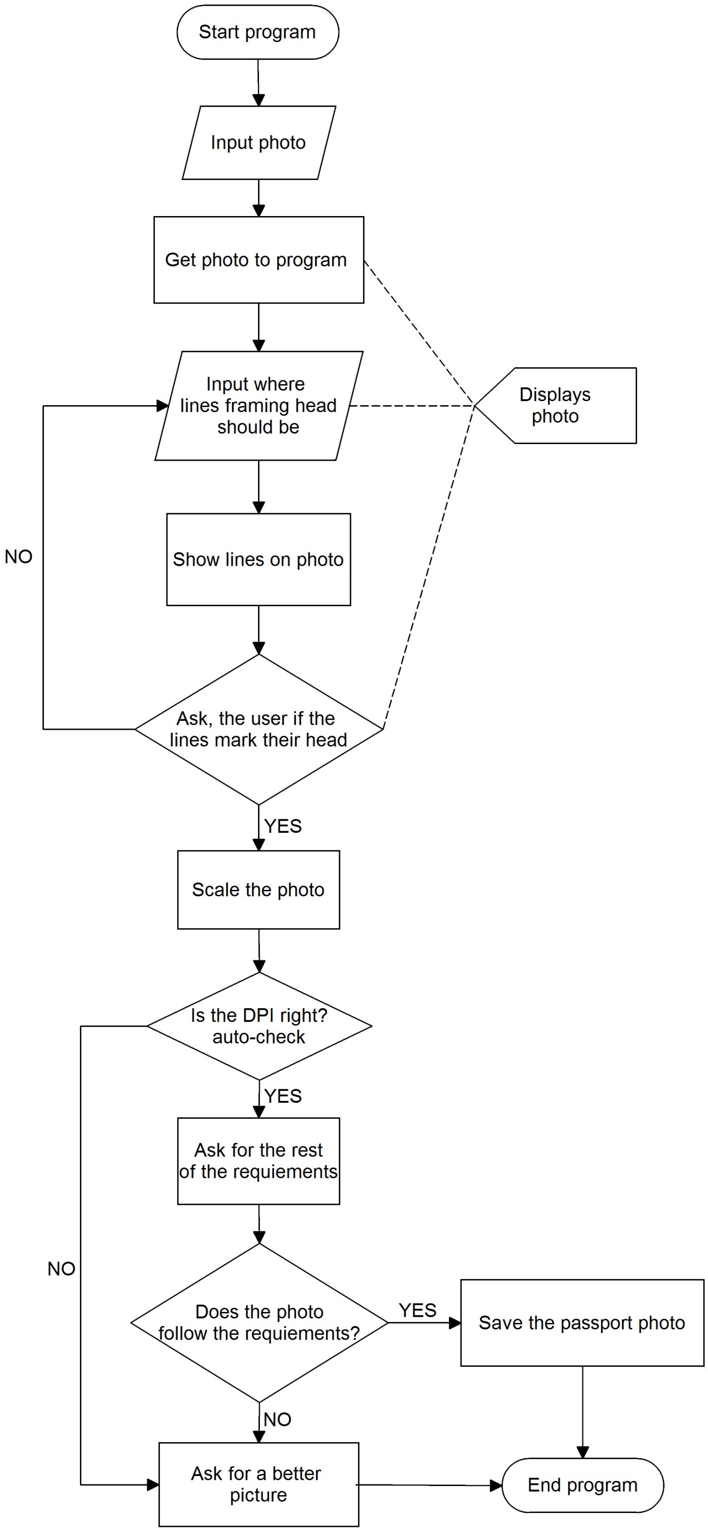
Start program
Input photo
Get photo to program
Input where lines framing head should be
Displays photo
Show lines on photo
NO
Ask, the user if the lines mark their head
YES
Scale the photo
Is the DPI right? auto-check
YES
Ask for the rest of the requiements
NO
Does the photo follow the requiements?
YES
Save the passport photo
NO
Ask for a better picture
End program

Figure 4.2: The flowchart of the program

The flowchart shows the flow of the program, where the boxes and arrows give an overview. But it does not look like program code. To help with making the code, we are making a pseudo code, which is a step further than flowchart. The program procedures are not written in boxes and explained in words. This does not give the same overview as the flowchart, but it can give an overview of, how the structure of the code in the program should be, and it is the starting point for the program. The pseudo code shows which code there has to be written, and how they should be arranged. The pseudo code of this program can be seen on Figure 4.3.

```
Program start
Prompt for photo name (photo must be in same folder)
Get photo height and width
If the photo is not in good enough resolution:
    Tell the user, to find a better photo

Else
    If the image is too bright or too dark
        Tell user, that the picture is too bright or too dark
While head not pointed correct by the photo
    Prompt the user for the heads position
    Display the photo with the face framed with lines
    Ask user if the frame is okay
Scale the image to the rules of passport photos
Save photo
Display photo

While there are more questions
    Ask user if the picture lives up to the different questions from the rules of passport photos.

If there are any rules the photo does not live up to
    Print the mistakes to the user

End program
```

Figure 4.3: The pseudocode of the program

Development 5

In this section there are a description of each part of the program. So it is easier to understand the program and how the functions work. Each part is described line by line and each line is represented by their respective line number in the listings. Not all the codes are shown, as only the more complex and important parts of the code are described.

5.1 Description of the grid marks

The grid marks are the red grid in the image. The grid is made to help the user find the points which the user needs to give to the program to crop the photo in the right size. The grid marks have been placed with a distance of 100 pixels between them. That can be seen on line 1 on Listing 5.1 where the defined *SCALING_SPACE* is set to 100 which is also explained to the user. The reason why it is a "define" is because it makes it easier to see where it is used and change should the value need to change. Then on line 2 the y (in the code described by the variable *count*) coordinate is set and it continues until it hits the other end of the photo. So the line goes through the photo. The same will be done for the y-axis.
From line 4 to 6 it makes every 5th line on the grid blue. The reason for the blue diversity is meant to help the user count the lines more quickly. The other part of the line will be red, that can be seen from line 8 to 10. Then on line 12 it will change the color on the specific pixel.

```
 1      for (x = 0; x < original_width; x += SCALING_SPACE){
 2          for (count = 0; count <= original_height; count++){
 3              if(x%500 == 0){
 4                  pixel_color_mark.R = 0U;
 5                  pixel_color_mark.G = 0U;
 6                  pixel_color_mark.B = 255U;
 7              }else{
 8                  pixel_color_mark.R = 255U;
 9                  pixel_color_mark.G = 0U;
10                  pixel_color_mark.B = 0U;
11              }
12              change_pixel (x,count,pixel_color_mark);
```

Listing 5.1: The function to scaling marks is found in ownfunc.h

5.2 Finding the head

The user must mark the head, so it is possible for the program to crop the image to the right size and dimensions. This will be done by the function *find_head*.

The function provides an output with instructions for the user on how to mark the head.
This function starts with making a char: *ans* that holds the answer with the users answer to if the selection of the head is correct. A while loop is the main part of this function and to get out of this loop the *ans* needs to be y (yes)- this happens when the user confirms that his or her selection of the head is correct. First thing that happens in the loop is that the user needs to set two points to enclose the head. If the user makes a mistake in his or her points. He or she will still be in the while loop because the user will be given a question if the selection is correct.

```
        /* Mark the head */
        mark_head(filename);
        /* Add scaling marks */
        scaling_marks();
        /* Function that writes the jpeg file */
        write_jpeg_file( filename );
        /* Open the image to display */
        show_pic(filename);
```

Listing 5.2: The code is taken from ownfunc.h

The function, as seen on listing 5.2, generates a new file containing the selection. This file is deleted when the user has determined if the selection is correct. If the selection is correct, then another new file will be generated, containing the cropped image, and the image containing the selection is deleted. If the selection is wrong, the file containing the selection will be deleted, and a new containing the new selection will be generated. The deletion of the file happens at first in the while loop, but not if it is the first time the while is being run, because there will not be a generated file yet. When the file containing the selection is generated, it is assigned a number in the filename. This is because the program not can redo the selection in the same file.

Then it uses the *mark_head* function to mark the head and then use the *scaling_marks* function to draw the marks along the edges of the image makes the image to a JPEG file. After this, it uses the *show_pic(filename)* function to open the new photo file in the users default JPEG viewer, so the user can see if the selection is correct. At the end of the while loop comes the question that asks if the selection is right. If the user answers y, the while loop ends. Now that the loop is finished it will delete the image with the selection and check if the brightness of the cropped image is at it should be.

5.3 Marking the head

The function *mark_head* uses the two points from *find_head* to make a box. So the user can see if the coordinates of the head are correct.
What is described here is related to listing 5.3. On line 4 it declares two variables that are used in the function. On line 2 to 5 these variables are part of a function seen on listing 5.5 to help the user if they misread the instructions. The variables which end with *tmp* will be equal to any leftovers, if the dimensions of the image is not in whole hundreds. The other two variables will then be equal to the amount grid marks the image contains.

For example if the user wrote 9, 9 for the first x and y and 10, 11 for the last x and y they may have thought they should count the grid marks, because each marker equals 100 pixels which is the coordinates they should write.

```c
/* Function to draw the marking of the head */
void mark_head(char *outfilename){
    /* Declare varibles */
    int i = 0, j = 0;
    int x_scales, y_scales, x_scales_tmp, y_scales_tmp;
    pixel pixel_color;

    x_scales_tmp    = original_width%SCALING_SPACE;
    x_scales        = (original_width-x_scales_tmp)/SCALING_SPACE;
    y_scales_tmp    = original_height%SCALING_SPACE;
    y_scales        = (original_height-y_scales_tmp)/SCALING_SPACE;
```

Listing 5.3: Part of the function *mark_head* in ownfunc.h

As described above, in listing 5.4 these two "if" conditions will set the variables to one hundred times more if they are lower than the amount of grid marks in the picture and if the variables times hundred are less then the original dimensions.

```c
if(upper_left.x < x_scales && upper_left.y < y_scales &&
     upper_left.x*SCALING_SPACE < original_width &&
     upper_left.y*SCALING_SPACE < original_height){
    upper_left.x = upper_left.x*SCALING_SPACE;
    upper_left.y = upper_left.y*SCALING_SPACE;
}
if(lower_right.x <= x_scales && lower_right.y <= y_scales &&
     lower_right.x*SCALING_SPACE < original_width &&
     lower_right.y*SCALING_SPACE < original_height){
    lower_right.x = lower_right.x*SCALING_SPACE;
    lower_right.y = lower_right.y*SCALING_SPACE;
}
```

Listing 5.4: Part of the function*mark_head* in ownfunc.h

In listing 5.5 the program goes into two "for" loops that determine which pixel has been reached. These loops hold an "if" condition: The "if" condition on line 4 uses the function *redline*, which can be seen in listing 5.6, to check if the pixel should be painted red. If the point is within the selection, it will then set the 4 outer pixels to red and change the pixel in the photo. This way the picture will now contain a red box surrounding the selection. The function *makedark* on line 5 will be initiated if redline returns 0 see listing 5.7 for the details.

```
1      /* For loop to change the pixels on the image (draw the marking) */
2      for (i = 0; i < original_width; i++){
3          for (j = 0; j < original_height; j++){
4              if(redline(outfilename, i, j, pixel_color) == 0){
5                  makedark(outfilename, i, j, pixel_color);
6              }
7          }
8      }
```

Listing 5.5: Part of the function *mark_head* in ownfunc.h

The function redline can be seen on listing 5.6. This function contains an "if" condition, the condition is to only include pixels that will draw a frame using the coordinates given by the user. On line 12 the given pixel's red value is set to 255 also known as the max value and the on line 13 and 14 the two other colors, green and blue are set to 0. On line 16 the pixel is changed to these values and thus is turned red. Also if the condition previously mentioned is met, the function will return 1 to indicate success, if not it will return 0.

```
1  int redline(char *outfilename, int i, int j, pixel pixel_color){
2      /* check if the pixel is inside the box frame */
3      if ((i <= upper_left.x +1 && i >= upper_left.x -1 &&
4          j <= lower_right.y +1 && j >= upper_left.y -1) ||
5          (i <= lower_right.x +1 && i >= lower_right.x -1 &&
6          j <= lower_right.y +1 && j >= upper_left.y -1) ||
7          (j <= upper_left.y +1 && j >= upper_left.y -1 &&
8          i <= lower_right.x +1 && i >= upper_left.x -1) ||
9          (j <= lower_right.y +1 && j >= lower_right.y -1 &&
10         i <= lower_right.x +1 && i >= upper_left.x -1) ){
11             /* Set the pixel to be red */
12             pixel_color.R = 255U;
13             pixel_color.G = 0U;
14             pixel_color.B = 0U;
15             /* The function that changes the pixel */
16             change_pixel(i,j, pixel_color);
17             return 1;
18     }
19     return 0;
20 }
```

Listing 5.6: Redline part of the function *mark_head* in ownfunc.h

On listing 5.7 the function that darkens the surroundings of the marking can be seen. This function contains an "if" condition on line 3. If the condition is met, it will then on line 6 get the color of the pixel and on line 8, 9 and 10 reduce the given color by a factor defined in the beginning of the program. On line 12 it will then replace the pixels' colors with the darkened ones.

```c
 1 void makedark(char *outfilename, int i, int j, pixel pixel_color){
 2     //Check if the pixel is outside the box
 3     if(i <= upper_left.x -1 || i >= lower_right.x +1 ||
 4         j <= upper_left.y -1 || j >= lower_right.y +1){
 5             /* Get the present color of the picture */
 6             pixel_color = get_pixel(i,j);
 7             /* Reduce the color of the picture (Darken the surroundings of
                   the marking)*/
 8             pixel_color.R = pixel_color.R/DARKEN_FACTOR;
 9             pixel_color.G = pixel_color.G/DARKEN_FACTOR;
10             pixel_color.B = pixel_color.B/DARKEN_FACTOR;
11             /* The function that changes the pixel */
12             change_pixel(i,j,pixel_color);
13     }
14 }
```

Listing 5.7: Makedark part of the function *mark_head* in ownfunc.h

5.4 Function to Calculate Selection into Passport Photo Format

As passport photos are required to have space around the head of the person, a function is needed which takes the current head-selection, and adds some space around it. This also makes the selection in the right format, which is required.

It is known that the height of a Danish passport photo must be 45 mm in total, where the width must be 35 mm. The height of the head in the photo must be between 30 and 36 mm. This is also where we use the function that calculates the DPI (see section 5.5), to determine how many pixels the image must be in dimensions.

A series of images have been chosen to illustrate how the function transform the selection.

Figure 5.1: Correct passport photo size marked

Figure 5.1 shows the selection after the function has transformed the user's selection. Now we imagine that the user has made a selection of the head. This user's selection is marked with red.

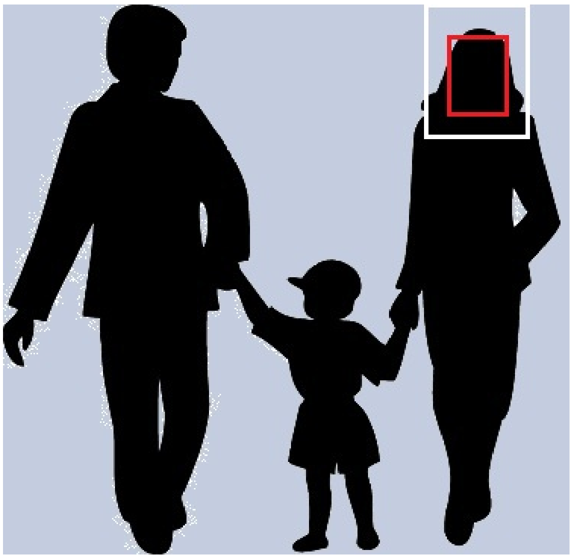

Figure 5.2: User's selection marked with red

Figure 5.2 shows how the function receives the user's selection of the head, and two pointers as input. The pointers are where the final selection points will be saved when the function is done.
First the function calculates the ratio between the height and width of the passport photo.
Next the function calculates the height and width of the current selection in pixels.
Now the function calculates what the height of the image really should be.

The difference between the head's height and width and the correct height and with of a passport photo is now calculated.
The difference of the height is divided by 3, and 1/3 is subtracted to the top of the selection of the head.

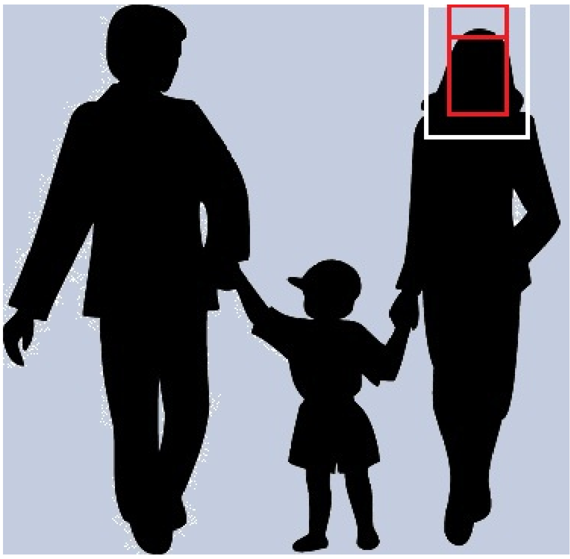

Figure 5.3: User's selection with added space on top

Figure 5.3, shows how this space is added from the head to the border of the photo.

2/3rds of the difference of the height is now added to the bottom of the photo, which adds space at the bottom of the head.

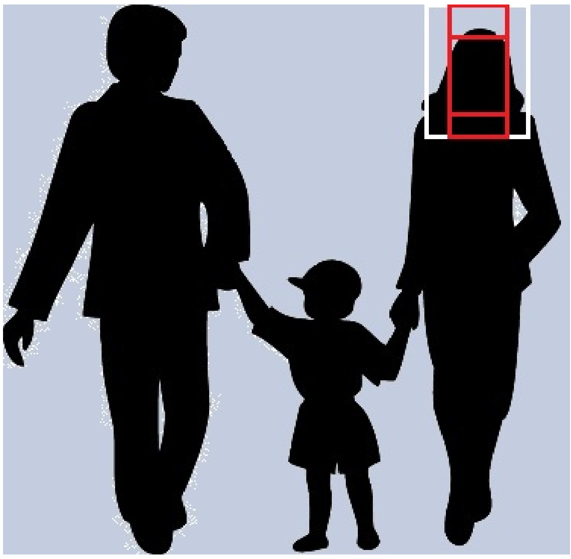

Figure 5.4: User's selection with added space on top and bottom

Figure 5.4 shows how the function subtracts half of the difference between the width of the head from the left of the selection. This adds space at the left of the image.

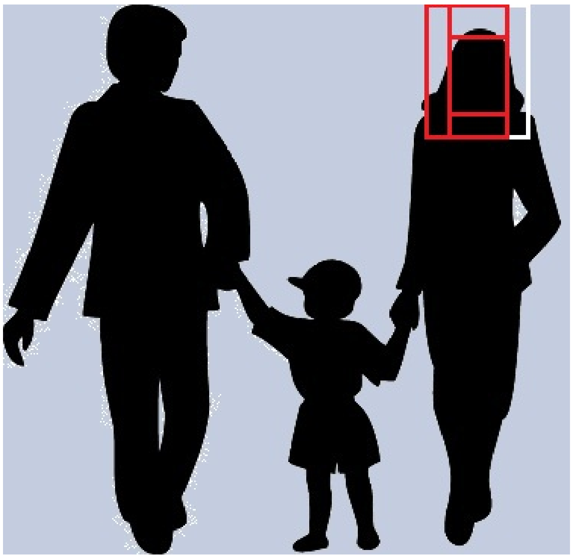

Figure 5.5: Users selection with added space on top, bottom and left

At last, figure 5.5 shows how the function adds half of the difference between the width of the head, to the right of the selection. Again, this adds space, but this time to the right of the image.

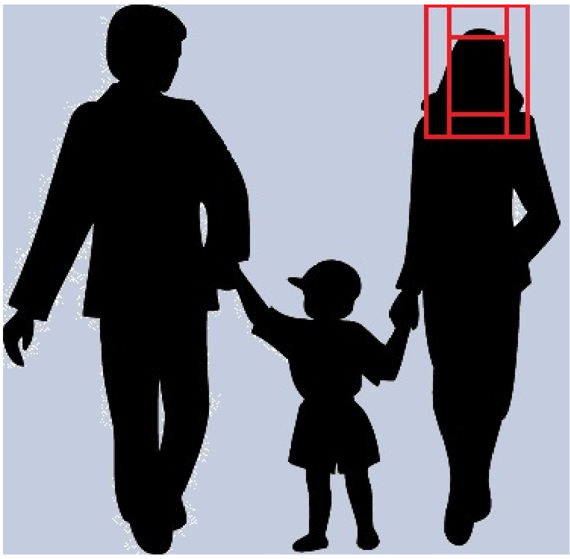

Figure 5.6: Users selection with added space on top, bottom, left and right

Figure 5.6 shows how the function now adds space around the whole selection, and this new selection is transformed into coordinates again, and saved in the pointers *final_ upper_ left* and *final_ lower_ right* as seen on listing 5.8.

```
    upper_left.x = final_upper_left_x;
    upper_left.y = final_upper_left_y;

    lower_right.x = final_lower_right_x;
    lower_right.y = final_lower_right_y;
```

Listing 5.8: Part of the function *Mark_ head*

5.5 Function to Calculate DPI

The *calc_ dpi* function receives 4 input parameters: the image width and height in pixels, and the image height and width in inches.

The DPI is calculated by dividing the pixel width(on line 3 on Listing 5.9) and height (on line 4 on Listing 5.9) with the inch width and hight. The function now has two values, the horizontal DPI and the vertical DPI. With these two values the average between the two are found(on line 6 on Listing 5.9). This is the image's DPI, which is then returned by the function. DPI is a measurement unit of how many dots that can be placed in a line within the span of 1 inch. The DPI is also used as a measurement of a printer's or scanner's resolution [Den Store Danske].

```
1 int dpi, dpi_hor, dpi_ver;
2
3     dpi_hor = pixelwidth/inchwidth;
4     dpi_ver = pixelheight/inchheight;
5
6 dpi = (dpi_hor+dpi_ver)/2;
7
8 return dpi;
```

Listing 5.9: The function to calculate DPI comes from *ownfunc.h* and it is called *calc_ dpi*

5.6 Light/Dark Check Function

The light/dark check function checks the cropped image to see if it is too light or too dark. The function starts by calling the function *brightness*. To determine whether the image is too light or too dark, the *brightness* function checks the return value of the function *find_rgb*. This function will be explained later in this section. If the *find_rgb* function's values is between 80 and 140 the *brightness* function returns 0 (from line 7 to 9 on listing 5.10), which means the image is fine. If the *find_rgb* function's value is under 80 the *brightness* function returns 1 (from line 1 to 3 on listing 5.10), which means the image is too dark, and if the *find_rgb* function's value is over 140 brightness function returns 2 which means the image is too light (from line 4 to 6 on listing 5.10). This return value is used to determine what should be printed to the user. There will either be printed a warning if the photo is too bright or too dark. If the photo is neither too bright nor too dark the program proceeds.

```
1 if (find_rgb(upper_left.x,upper_left.y,lower_right.x,lower_right.y) < 80){
2     return 1;
3 }
4 else if(find_rgb(upper_left.x,upper_left.y,lower_right.x,lower_right.y) >
      140){
5     return 2;
6 }
7 else{
8     return 0;
9 }
```

Listing 5.10: Checks if the selected area is too light or too dark. The code is taken from the function *brightness* in Ownfunc.h

The *find_rgb* function finds the average of the three colors in every pixel and has four input parameters. The average of one color is found by finding the color for one pixel and then add it to the sum multiplied with *count* and then divide with *count*+1 (see line 5 to 11 on listing 5.11). This is done through two for-loops that do it for the three colors and place them in three separate variables. One for-loop for horizontal movement in the image and another for-loop for vertical movement in the image (see line 2 and 3 on listing 5.11). *Count* acts as a counter and is counted by one for every pixel it runs through, and is used in calculations to find the average of each color (see line 13 on listing 5.11).

```
 1 /* This for loop go through the image and measures the RGB values in each
       pixel */
 2  for (i = ulx; i <= lrx; i++){
 3      for (j = uly; j <= lry; j++){
 4 
 5      pixel_color = get_pixel(i,j);
 6 
 7     /* Because unsigned ints only have a range around 4 billions
 8     the average of each color is calculated on the go */
 9     red = (red*count + pixel_color.R)/(count+1);
10     green = (green*count + pixel_color.G)/(count+1);
11     blue = (blue*count + pixel_color.B)/(count+1);
12 
13     count ++;
14     }
15    }
```

Listing 5.11: This code finds the average of the three colors from all pixels in the marked area. The code is taken from the function *find_rgb* in Ownfunc.h

After the two for-loops the 3 variable values are added together and then divided by three because every pixel together of three colors and stored in a variable *rgb* (see line 2 on Listing 5.12). The *find_rgb* then returns the value of *rgb* (see line 4 on Listing 5.12). The value is used to determine if the photo is too dark or bright.

```
1 /* The average value of the 3 colours together are caculated and stored*/
2 rgb = (red+blue+green)/3;
3 
4 return (int)rgb;
```

Listing 5.12: Calculate the average value for the three colors. The code is taken from the function *find_rgb* in Ownfunc.h

5.7 Function to Crop Photo

The purpose of the function *crop_jpeg_file* is to crop the image. It is used in the program when the user has pinpointed the head on the picture, and the program has calculated the right size of the output image. The function has its base in the function *write_jpeg_file*, written by Junaed Sattar [Sattar]. The functions are in the headerfile "jpegfunc.h". The input parameters of the function are the filename of the *outputfile*, two variables of the type point, and an integer value of DPI. The two points, mark the area of the crop. The point *crop_upper_left* marks the upper left corner of the crop area, while the points *crop_lower_right* mark the lower right corner of the crop area. The two points together mark the rectangle which should be cropped (see Figure 5.7). The variable *DPI* determines the DPI of the output picture which is a setting for the printing of the picture. This can be seen at Listing 5.13.

```
jpeg_create_compress(&cinfo);
jpeg_stdio_dest(&cinfo, outfile);
/* Setting the parameters of the output file here */
/* B205: the width and heigth is set to the differens of the two points*/
cinfo.image_width = crop_lower_right.x - crop_upper_left.x ;
cinfo.image_height = crop_lower_right.y - crop_upper_left.y;

cinfo.input_components = bytes_per_pixel;
cinfo.in_color_space = color_space;
```

Listing 5.13: Setting the compression parameters and the points where the picture should be cropped too. Code is from crop_jpeg_file in jpegfunc.h

The output of the function is an integer, which indicates if the process was successful. An output of 1 indicates that the function has ended with success. An output of -1 will indicate that there was an error. This integer value is a return output, but there is also another output - the output file of the cropped image. The file will be saved as the name of the input variable *filename* in the format jpg.

The function is a customized version of the function *write_jpeg_file*, because it contains many of the same codelines. It starts by opening the file for writing, and starts the compression which is a function in libJPEG. The width and height of the image are set to the width and height of the cropped area, and not of the whole original picture.

Unlike the function *write_jpeg_file*, the while-loop which writes data to the program, does not run through all the lines (as seen on listing 5.14), only the lines containing the information for the cropped area.

```
/*B205: This whileloop will not run the whole picture, but only between the
    cropping markers*/
while( cinfo.next_scanline + crop_upper_left.y < crop_lower_right.y )
{
```

Listing 5.14: While loop that writes the new cropped picture. Code is from crop_jpeg_file in jpegfunc.h

Inside the loop, the pointer is placed in the image buffer, at the modified *crop_ upper_ left* coordinates as seen on listing 5.15.

```
    row_pointer[0] = &raw_image[ ((i + crop_upper_left.y) *
        cinfo.image_width *
        cinfo.input_components)+crop_upper_left.x*    cinfo.input_components];
```

Listing 5.15: Place pointer in image buffer. Code is from crop_jpeg_file in jpegfunc.h

The program then finishes the compress of the image and closes the image file (see line 1 to 3 at listing 5.16). If the program succeeds in this process it will return 1. This snipped of code can be seen at listing 5.16.

```
1 jpeg_finish_compress( &cinfo );
2 jpeg_destroy_compress( &cinfo );
3 fclose( outfile );
4 /* success code is 1! */
5
6 return 1;
```

Listing 5.16: Close file and return 1 of success. Code is from crop_jpeg_file in jpegfunc.h

5.8 Function to Check Photo for Passport Rules

This part of the program asks the user a number of questions to determine if the photo requirements (decribed by the police as seen on section 2.3) are fulfilled. In the program 3 arrays of the type char have been made that respectively contains answers, questions and errors.

Every time an answer is given. a control is initiated to see if that answer corresponds with the answer list. If the answer is not correct the number of the question will be saved into the array *errors_ number*, which will be printed at the end of the questioning so the user can see which of the requirements the photo lacks.

```
                printf("%d of %d: %s",(i+1),MAX_QUESTIONS, questions[i]);
                scanf(" %c", &temp_answer);
                printf("\n");

                // put the number of the questions into a array
                 if(temp_answer != answer[i])
                {
                  errors_number[k] = i;
                  k++;
                }
```

Listing 5.17: A part of the code asking the user for questions. Part of the function *ask_ questions* in the headerfile que.h

5.9 Instruction Manual

This is a short manual on how to use the program. The program is made by university students on their first semester to become software engineers. Please read EVERYTHING in this manual and in the program. The program will explain what you have to do to check your photo.
Before you start the program, you need a photo of yourself, or the person you want to make a passport photo for, standing in front of a wall that is either white, off white, cream or light grey or is in contrast with your hair or a coloured scarf.
To start this program you need to put a photo into the folder where the program is under the name "input_photo.jpg". Notice that the photo needs to be in natural colors, therefore not allowed to be in black and white! When you have the photo in the folder, you start the program called "Passportmagic.exe". When the program is done, you can take your new photo that now is called "passport.jpg". After asking some questions about the photo the program will tell you if the photo is fine. If there are any problems with the photo the program will tell you what needs to be corrected in the end.
Before the questions you need to tell the program where your head is, so it can crop and scale it to the right size for you. No worries if your selection is not correct, the program will ask you whether the selection is correct or not, if not, then you can make a new selection. The marking of the head, should be by typing the upper left and lower right corner of the head on the photo as shown on Figure 5.7. There are 100 pixels between each line.

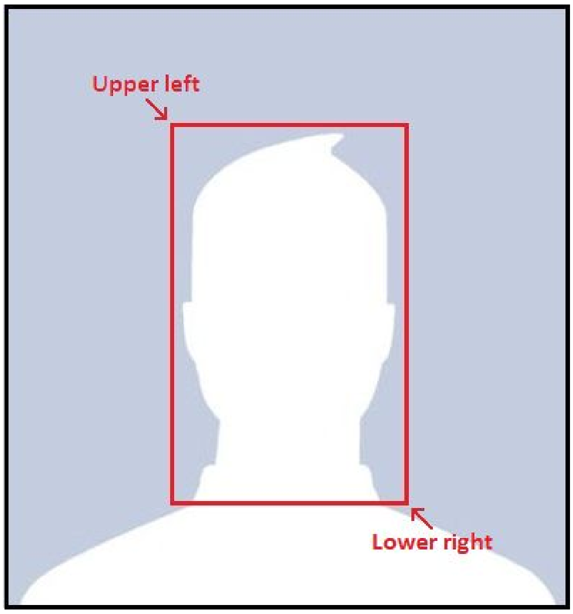

Figure 5.7: Shows the upper left and lower right corner of a head

The program is command line based, so everything the user wants to do needs to be written in the program.

Testing 6

When a piece of software is developed, the testing is very important. The developers know the functions and how to use the program. Input from the user can be needed, which the developers take as implicit, and users might not understand. The possibilities of letting a third party test the program, is to find out if the instructions of the program are enough for the user, so they can understand how to use the program. This testing will give valuable information for technical improvements and improvements of description when an outsider is testing the program.

6.1 Choice of test person

The test persons that were chosen, were categorised as belonging to the modern segment, the modern individual segment, and the modern society oriented segment that we have defined as our target group. The test persons chosen were because they were in this segments and therefore relevant for the product. The test persons also had insight into how to use a modern computer and use it in their daily routine.

6.2 Choice of Survey Method

To test the product and see how the customer would interact with the program, a test was scheduled where a test subject was set to try out the program. For the analysis and further work regarding the test result a qualitative survey method has been chosen.

For the interview there has been made a questionnaire (see appendix A) that will be filled out during the interview. The questionnaire should work as a tool for the interview to get the right information from the test subject [Creative Research Systems]. The qualitative survey is best suited for the project, because it gives a concrete picture of how the user sees and uses the program and which problems the user meets in the use of the product [Ospina, 2003]. The problem with the interview method is that it uses more resources than the questionnaire, because the group will have to set time off to execute and conduct the survey, process the results gathered from the survey and that will take more time than to process results from a questionnaire (quantitative method). The interview must be very precise and well thought to get a valid result, same goes for the questionnaire. From the answer, new improvements should emerge and be implemented in the program. With the test results it is also possible to draw a conclusion from test results regarding the solution of the problem [Creative Research Systems].

It is important when making a questionnaire that it is structured correctly and that the questions are well formulated so the test subject does not miss the point of the question or misunderstands its meaning. The formulation of the questions is also important regarding the result from the research, because of the fact that a poorly formulated question maybe leads to useless answers and thereby the questionnaire has been in vain [Creative Research Systems].

So the reason for the survey is based on the idea that the survey will help the improvement of the product and show potential problems that could emerge from the use of the program.

6.3 First Test of the Program

A test of the program was made with help from other students at the university. The main idea behind the test was to see if any part of the program needed to be changed/bug fixed. The test persons were first asked to read the instruction manual. After having read this, the program opened so the test person could test it, while one of the group members of this project documented what happened.
An interview was made with the test person after the test was done, so the test person's opinion about the program and what could be better was documented.
As a result of the test the instruction manual was changed so that it ended up containing some new info, like that the program only supports photos in natural colors. Different parts of the program code were changed so it was easier for the user to use the program and easier to get an overview of what is happening in the program. For example the scaling marks were changed so it is now a grid and not just marks – this makes it easier to find the right coordinates to supply the program with. Numbers on the questions were also added so the user has an idea of how far in the program he is. Furthermore, more bugs have been fixed.
Therefore the biggest change after the test was making the program easier to use and help the user understand what to do, and get an easy overview of the program.

6.4 Second Test of the Program

The second test was also made with help from other students of the university. The main idea about testing again was to see if we had corrected all the bugs from our last test, and to see if any new bug could be found. Like the last test, the test person was first asked to read the instruction manual, and after this, test the program while a group member documented what happened during the test.
After the test the group members made an interview with the test person to get his opinion about the program and what could be better.
After the second test only minor bugs or mistakes were corrected. Some of the minor mistakes were questions which were too difficult to understand and spelling errors. The test person suggested painting every fifth line in the grid blue, and that the program should have a function correcting the user's input, if the user, by mistake, counted the lines in the grid instead of pixels. These two suggestion have been successfully added to the program.
So all the small bugs and mistakes has been fixed and the two suggestions from the test person has been added to the program after the test.

Discussion 7

This chapter will discuss and compare the product to the thesis statement in section 2.6, whether the product actually helps the user and gives a solution to the thesis statement. In this chapter there are also a perspectivation regarding the further work with the project and some ideas to what problems the users might experience while using the program.

7.1 Discussion

The focus in this discussion is to see if the product is in conjunction with the problem analysis and to see the different requirements integrated in the product.

In the problem analysis we stated the following requirements for the program:

- Integrate picture
- Display image
- Pixels per inch
- Pinpoint head
- Scaling
- Checking
- Finish

The program has been designed based on the requirements and the target group. The targeted group is used in the design phase because it is important to form the product so it helps the desired group. The target group would also highlight functions that the program will need to have. The target group is found by using the target group analysis (see section 4.1).

In section 3.4 it was described why libJPEG was chosen as the used image library, and why a library was needed. The ability to work with an image is the core-function in the program because the program both crops and scales the image to the right size of a passport photo. This also includes the ability to actually display an image. Although the program cannot display an image itself, it can call the default JPEG viewer in Windows, and display the image that way. The program needs to display the image to the user when the user has entered the marking of the head. This way the user can, by viewing his or her selection, determine if this was correct, or redo it. This is also the feature called pinpoint head. These are important features that the program has been made after to help the user with the editing of the photo and try to solve the problem that has been stated.

A printed image, and an image on a screen has different measurements. Therefore the program needs to convert from pixels to inches, in order for the passport photo to become the right size when scaling it. When the program converts the measurements, it also checks dots per inch, which basically is the quality of the image. The quality is important regarding the passport photo and therefore it is important that this function is implemented in the program.

The program needs to scale the image based on the selection of the head. A passport photo is not simply a cropped out head - it is a photo of a head with shoulders visible and so on. Therefore the program needs to scale the photo correctly as described in section 5.4. This is an important function regarding editing a photo and therefore essential in this program.

When the generation of the photo has been completed the program will display the result, and the user will be asked a series of questions, which will help the user determine if the photo is actually usable as a passport photo according to the rules of passport photos. A later extension of this feature would be to have the user define which country the passport photo is to be issued in. This way the program could load different questions according to the rules in that particular country.

It can be seen that all the requirements to the program has been fulfilled. The solution to the program is therefore satisfying. The thesis statement was: "Is it possible to develop a program that helps the customer through the process of making a passport photo?" (see section 2.6). From the review of the requirements to solve this thesis statement, it can be seen that it is possible. The test persons who tested the program all concluded that the program was very useful in the process of making a passport photo, and both the requirements for the program and the thesis statement is fulfilled.

7.2 Perspectivation

When working with problem based learning it is important to relate the product to the user environment and discuss how it would work in the user environment. This will unravel possible problems and effects the product can have. This capture will describe the considerations regarding the further development of the program and potential problems regarding the use of the program.

The product is made in the language C which means that the program will have to run through a UNIX prompt. Therefore, the user must have a certain knowledge regarding the use of computers. This can be a problem because many people are not familiar with the term prompt and the UNIX environment. If the product should have been on the market it would be important to find a solution to the requirements for the program to run. This could be solved by including the requested files regarding using the execution file. Through this there should not be a problem running the program without the prompt installed.

This software does not support the use of mouse and therefore the user will have to use input commands to use and control the program. This may cause the customer to feel insecure using the product and that the product may be too difficult for them to handle. This should be prevented by making a good instruction manual and a help section so the customer can always seek help and will then be capable of using the program. This leads to the idea that most of the regular users would rather use a program with a user interface than a program without an interface. The user interface is an important part of the product in the modern society and big companies invest a lot of time and money developing their interfaces.

Eventually more problems may emerge when the customer starts using the program. The problem may be that the program enters a loop and repeats itself over and over. To prevent this and make a better product, more tests phases might be needed. The later test will show problems that the customer may encounter. After this test stage, the major problems the customer may have experienced should be solved and refined the product more.

All these problems should be taken into account when further development of the product takes place. The major part of the development lies in making and designing an effective interface with mouse support.

Conclusion 8

In this final chapter a conclusion on how well the final product works and which of the requirements. The user will have to use commands to navigate through inputs given to the program. The program then reads the input and uses the input for its purpose and decides what actions should be taken via the inputs. The program will write text to the user that explains what the user needs to know with the different steps through the program.

Then there is the problem that the program is a console application and not a windows application. Therefore one of the main problems is to make the program as user friendly as possible. This problem has been managed at a reasonable point, with text that helps and guides the user and if the user is failing in the process, the program can handle it in most cases.

The functions that are used in the program in order to handle the jpg photo come from a library named "libJPEG". This library helps the program handle images and edit them to the right size in terms of height and width or passport photos. Then the program asks a series of questions to check that the other rules are covered. To check if the photo is of high enough quality there has been made a function that can check the quality of the photos.

The program can open an image, and are able to draw a grid to help the user with the input coordinates. The program can then crop the image with the input from the user and show the crop image, so the user can see if the image is cropped right. The program then checks the DPI and brightness of the image and will inform the user if any of these two the do not meet the restrictions. The user is then to answer a list of questions regarding the photo to ensure that all rules for passport photo are followed. After the questions the program saves the edited image which then should be possible to use as a passport photo. The program informs the user what to do all the way through the process and guides the user through the rules.

So the program covers the core requirements that have been developed for the product and thereby the program should help solve the problem set in the thesis statement. It can be conclude that the program gives a product that should be more reliable as passport photo rather than just a photo taken by the user themselves. Maybe there are small problems with the interface but that does not play any role regarding the product that the program gives the user. Therefore it is a good program that helps the user with the process but it can be improved by using another language that makes it easier to make use of an interface.

Bibliography

Aalborg University, **2010**. Aalborg University. *Principles of Problem and Project Based Learning: The Aalborg PBL Model.* AAU, 2010.

Bible. Hebrew Bible. *Hebrew Bible.* http://www.biblegateway.com/passage/?search=nehemiah

Carlsen, **2009**. Lene Torp Carlsen. *Pasfoto bliver afvist på stribe.* Politiken, page 1, 2009.

CNET Staff, **july 2011**. CNET Staff. *Free passport photo software.* http://download.cnet.com/Free-passport-photo-software/3000-18494_4-75322099.html, 2011. download: 02-11-2011.

Coleman, **2007**. Stephen Edwin Coleman. *Digital photo manipulation: A descriptive analysis of codes of ethics and ethical decisions of photo editors.* 2007.

Creative Research Systems. Creative Research Systems. *Survey Design.* Creative Research Systems, http://www.surveysystem.com/sdesign.htm.

Den Store Danske. Den Store Danske. *Den Store Danske - dpi.* Gyldendals åbne encyklopædi, http://www.denstoredanske.dk/It,_teknik_og_naturvidenskab/M%C3%A5l_og_v%C3%A6gt/dpi.

ePassportPhoto. ePassportPhoto. *ePassportPhoto.com.* http://www.epassportphoto.com/. download: 02-11-2011.

ePhotozine, **1st January 2003**. ePhotozine. *ePhotozine manual.* http://www.ephotozine.com/article/make-your-own-passport-photos-4797, 2003. download: 02-11-2011.

Foresti, Micheloni, Snidaro, and Marchiol, **sept. 2003**. G.L. Foresti, C. Micheloni, L. Snidaro, and C. Marchiol. *Face detection for visual surveillance.* pages 115 – 120, 2003. doi: 10.1109/ICIAP.2003.1234036.

Fourandsix. Fourandsix. *Photo Tampering throughout History.* URL `http://www.fourandsix.com/photo-tampering-history/`.

Gallup. Gallup. *Gallup Kompas.* TNS Gallup, http://www.gallup.dk/vores-markedsfokus/medier/printmedier/gallupkompas/kompas-segmenter.aspx.

Grogware LLC, **2011**. Grogware LLC. *Passport Photo Studio.* http://www.passportphotostudio.com/win/index.html, 2011. download: 02-11-2011.

Hafnia Foto, **2007**. Hafnia Foto. *Hafnia Foto - Pasfoto.* http://www.hafniafoto.dk/produkter/pasfoto.html, 2007.

IJG. IJG. *Independent JPEG Group.* http://www.ijg.org/.

JPEG. JPEG. *JPEG homepage.* URL `http://jpeg.org/jpeg/index.html`.

Ministerie van Binnenlandse Zaken en Koninkrijksrelaties. *Ministerie van Binnenlandse Zaken en Koninkrijksrelaties.*

On the go soft. On the go soft. *On the go soft: Passport photo.* http://www.onthegosoft.com/passport_photo.htm. download: 02-11-2011.

OpenCV. OpenCV. *OpenCV Wiki.* URL `http://opencv.willowgarage.com/wiki/`.

Ospina, **August 2003**. Sonia Ospina. *Qualitative Research.* SAGE Publications london, 2003. URL `http://wagner.nyu.edu/leadership/publications/files/Qualitative_Research.pdf`.

Politi, **a**. Politi. *Foto: Pas og kørekort.* http://www.politi.dk/da/borgerservice/pas/pasbilleder/pasbilleder.htm.

Politi, **29 december 2009**. Politi. *Borgerservice passport prices by politi.dk.* http://www.politi.dk/da/borgerservice/pas/paspriser/, 2009.

Politi, **b**. Politi. *Det biometriske pas.* http://www.politi.dk/da/borgerservice/pas/biometrisk_pas.

Sattar. Junaed Sattar. *Libjpeg sample code howto.* McGill University, http://www.cim.mcgill.ca/ junaed/libjpeg.php.

Teknologi-rådet. Teknologi-rådet. *Biometriske pas – et redskab mod terror og illegal indvandring.* http://www.biometri.info/7-biometriske-pas-%E2%80%93-et-redskab-mod-terror-og-illegal-indvandring.html.

Teknologi-Rådet. Teknologi-Rådet. *Biometriske pas – et redskab mod terror og illegal indvandring.* URL `http://www.biometri.info/7-biometriske-pas-%E2%80%93-et-redskab-mod-terror-og-illegal-indvandring.html`.

Tomgreen, **Februar 6 2009**. Tomgreen. *The passport photo blog by ePassportPhoto.com.* http://www.epassportphoto.com/Blog/, 2009.

Torpey, **2001**. John Torpey. *Documenting individual identity.* Princeton University Press, 2001.

Torpey, **2000**. John Torpey. *The invention of the passport: surveillance, citizenship, and the state.* The Press Syndicate of the University of Cambridge, 2000.

Wallace, **feb 1992**. G.K. Wallace. *The JPEG still picture compression standard.* Consumer Electronics, IEEE Transactions on, 38(1), xviii –xxxiv, 1992. ISSN 0098-3063. doi: 10.1109/30.125072.

William, **2009**. Floyd William. *An Empirical Analysis of the State's Monopolization of the Legitimate Means of Movement: Evaluating the Effects of Required Passport use on International Travel.* Texas State University, 2009.

Questionnaire A

1: Did you have any technical problems?
Yes _
No _
If yes, which problems did you encounter?

2: Did you find the program useful?
Yes _
No _

3: Did you have any problems understanding what to do in the program?
Yes _
No _
If yes, please explain what problems you had.

4: What grade will you give this program on a scale from 1-10? 1 point is given if you think the program is not helpful/useful in any way. 10 points are given if you find this program very useful/easy to use.

5: Please tell us your opinion in your own words.

6: If you have suggestions for improvement for the program, please let us know.

Mindmap B

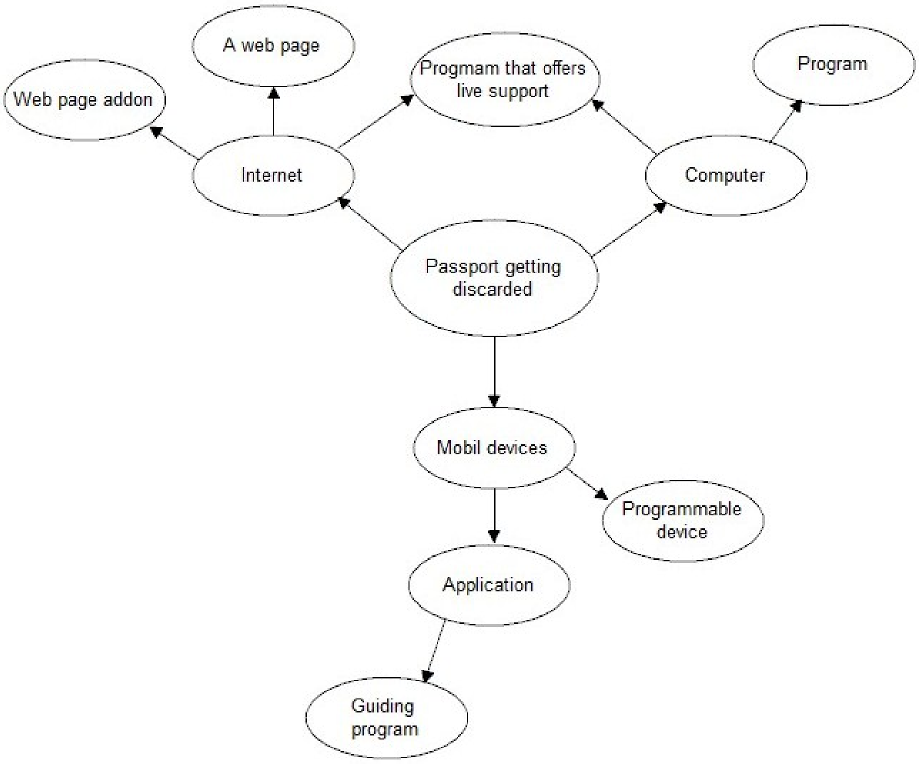

Figure B.1: Mindmap over ideas to different products

Suggestion for new P1 project proposal C

Photo Editing of Passport Photos

This is our proposal for another project.

Initiating Problem

Are you able to edit a passport photo by writing and running a program, instead of using an interactive photo editing program, so the photo will end up following all the rules about passports photo?

The Problem

It is very difficult to take a passport photo by yourself which follows the rules in Denmark. Many people get their photos rejected at the civil service department due to the strict rules. Some times it would be practical if you could solve this problem through a program that would automate the progress of scaling and checking if all the rules are being followed. For example checking if the user has red eyes or reflections in the glasses, and in that case, remove it.

Purpose

The purpose of the project is to create a program which will help getting a photo approved as a passport photo.

Object

The object is to analyze image editing for the purpose of making a model that allows us to work with photos through simple C programs.

Engineering and Science Disciplines

Digital photo editing, algorithms and datastuctures, programlibraries.

Examples of Contextual Fields

Everyone uses passports today. If you are going anywhere your passport is an important document you need to bring, if you want to go to other countries. In airports as ID the passport is needed. Therefore everyone needs passport photos. And because of the strict rules the photo is hard to take yourself without any photo editing.

Proposer

1st Semester
Group B205
Paricipants:
Dag Toft Børresen Pedersen
Christian Jødal O'Keeffe
Niels Brøndum Pedersen
Aleksander Sørensen Nilsson
Mette Thomsen Pedersen
Rasmus Fischer Gadensgaard
Kasper Plejdrup