Senior project defense  
Photo castle 1.0

Brian Aguilar Diaz – Bachelor’s in Computer Science

Project Advisor: Dr. Sean Hayes

Graduation Date: May – Spring 2022

# Statement of purpose

Users today are all about their photos, they take photos wherever they go but sometimes the photo is not right. It might just need a spark or a bit of a touch up but where can you go to easily provide this? Fotosizer exists, yes, but the problem is that it is a bit outdated to what users currently are used to. Users want a user-friendly program adapted to newer features that are used now a days. Users today want a program that can easily manipulate many photos effortlessly without having to waste hours learning a new program such as Fotosizer or even Photoshop. Most importantly users will not be spending an exorbitant amount of money on editing programs so therefore an affordable solution must be provided.

The users today admit that they do not have much time to learn complex programs, the user also admits that they would not be willing to spend a lot of money on editing a simple photo. That is why I propose to create a free and user-friendly photo manipulating program that will be known as Photo Castle 1.0.

Users will be able to edit photos in bulk and create masterpieces with a user-friendly menu and with no need to become photoshop experts. In only a matter of seconds a user can become an expert in Photo Castle without being limited to any restrictions. This effortless, fast, and unlimited photo editing program is a solution to the many issues faced by the newest and youngest generation coming forward into the world. This is my purpose in creating Photo Castle 1.0 and the solution to many photo editing blunders that have been bothering our new generation.

# Research & BackGROUnd

## Research Efforts

* Step #1: Figure out how the main interface will look, feel, features and accessibilities added.
* Step #2: Make sure I know the libraries and packages that need to be included for my code to work.
* Step #3: Begin coding in C++ for the base program and make sure the base cases/functions work.
* Step #4: Research and learn how Qt works, watch tutorials, and view example programs.
* Step #5: Learn QT and Research how to use QT in my own code.
* Step #6: IDE Qt creator, figure out and see if using this is better than visual basic.
* Step #7: Integrate C++ code with Qt to have a nice-looking GUI/interface.
* Step #8: Update the interface, clean the code up, and add any more features that were not added before
* Step #9: Run the program and run test cases to make sure no bugs appear.

## Personal Motivation

* Motivation #1: Improve my C++ language skills and further my knowledge in all the different areas of C++. Out of the languages I have learned, C++ is my favorite, and I was upset I did not get to code much more in C++. However, this project will provide me the experience I so much desired.
* Motivation #2: Learn and understand how to use GUI’s and interfaces that are a bit more advanced. I loved the idea of using Qt as the design interface program for my project simply because it was used to create the same interface in my Mercedes-Benz, and I think it is one of the best interfaces used on a vehicle by far.

# Project language, software, & hardware

## Implemented Languages

* Language #1: C++
* Language #2: Qmake

## Libraries, Packages, and Development Kits

* Library #1: Qt
* Library #2: IDE Qt Creator
* Library #3: Qt Image libraries

## Additional Software/Equipment

* Need #1: Qt program for the design interface (https://www.qt.io/)
* Need #2: Qt Creator
* Need #3: Visual Basic
* Need #4: System requirements of at least …

# Project requirements

## Functional

The new system **must** include the following:

**i. The Main Program**

**ID:** 1.1

**Type:** Functional

**Description:** The program must show the policies and rules to which the user must agree to before using the program. The program must allow the user to customize the interface size, and tool pop up location and size. The user must be given the option to stick to the free Photo Castle or upgrade to the pro edition. If any issues occur the system must have a help section that guides them to the forum/support team. The program must show a minimal version of all tools. The program will open a pop up of a tool section they choose. The program must allow the user to apply any number of effects/filters on a photo. The program must allow photos to be edited in bulk. The program must allow the user to save photos in bulk or individually.

**Rationale:** The user will open the program and edit photos. The user will use tools to edit photos. The user will add filters to change the entire photo. The user will crop/customize photos. The user will edit photos in bulk. The user will save photos and download photos.

**Fit Criterion:** 99% of the main program will run smoothly for 99% of the users on the first 24 hours of usage. The program will display every tool clearly and have sections for each different type of help. 89% of the users will agree Photo Castle can manage the essential editing process quickly and efficiently.

**ii. Functions**

**ID:** 1.2

**Type:** Functional

**Description:** The program must have tools on the left to choose from (pop up menu) that allows editing of photos. Each tool must have a symbol and when chosen on the right a pop up menu will appear to customize the tool to the users’ liking. The program must have a top bar where files can be added, saved, and filtered. The program must also have a help section at the top of the program to access the forum and manuals. The program will have a huge black block in the middle where the photos will be placed to edit. If user chooses to go Pro the tools will have extra pop up screens on customizability.

**Rationale:** The user will want a menu to choose the tools they want to edit with. The user will need a section to choose the picture they want to upload and edit/save. The user needs a main menu to find tutorials and the forum section. The user will need to change the sizes of their tools and colors in which a pop up menu for each tool should appear.

**Fit Criterion:** 99% of the tools on the right of the program should work and be visible to the user after opening the program. 75% of the users will agree the tool menu is simple and gets just what they need done. The program will display the tool and customizability after being chosen.

**iii. Tools/Use**

**ID:** 1.3

**Type:** Functional

**Description:** The program must have a blur, erase and match tool. The program must have a drawing tool to which many different styles can be chosen. The program must have a cut and crop tool. The program must have a heal feature and magnifying one as well. The program must have an undo button and redo. The program must have a text box insert and move tool. The program must have a copy and paste feature. The program must have a fill/paint tool. The program must have a pop up menu for each tool to further edit the tool. The program must have a save/upload and exit button. The program must have a filter selection. The program must have a tool to edit photos in bulk.

**Rationale:** The user will want to erase and blur facial features. The user will want to draw or paint their photos. The user will crop images that were taken in large portions. The user will want to heal a surface that turned out edgy. The user will want to paint and fill an entire section they choose. The user will want to save and edit photos in bulk. The user will want to add filters to make colors look more vivid. The user will make mistakes and needs an undo button

**Fit Criterion:** Within seconds the user will see the tools on the left of the screen with simple symbols to identify them. The user can click and apply each tool without any limitations. 89% of users will agree that every feature they need to edit is in Photo Castle. After a week of use the user will be given a survey to see if any more tools are needed. 10% of the users will recommend more tools.

## Look and Feel (Functional - how the program needs to be simple)

The new system **must** include the following:

1. **Appearance**

**ID:** 2.1

**Type:** Appearance (Look)

**Description:** This program must be suited for a younger teenage audience. The program must be user friendly and be given a minimal design. A dark and light mode will be incorporated for different times of the day. The interface must have a small menu pop up for tools and colors. Photo Castle branding must be standard on the program as well as minimal.

**Rationale:** The user’s who love to edit photos are a younger teenage audience. Young audience like updated modern apps that function easier than photoshop but can still do the same job. Photoshop type apps use dark and light modes on their programs. Photoshop contains their branding on the program which helps people remember the brand and their work.

**Fit Criterion:** Within a couple of minutes of downloading and opening the app the user will be given the interface and can instantly interact with it. The user can test out each tool through the pop ups and pick whether they want a dark or light mode. Photo Castle branding will be inspected to look and match the Photo Castle logo whether it be in dark or light mode (small or large).

1. **Style**

**ID:** 2.2

**Type:** Style (Look)

**Description:** The program must be minimal and attractive. Similar to Mac OS and their minimal and detailed look, Photo Castle’s look will be inspired by the same style.

**Rationale:** Younger users tend to like updated modern and minimal UI like Apple’s programs. The users want a program that can do it all without looking complicated. The users do not want a program that looks like a 90’s program but instead need a clean and simple UI.

**Fit Criterion:** Once the program is used, 80% of the potential users will agree that the design element is clean and minimal (however with room to improve). Users will be given a survey after 12 days of use and decide if the UI needs improvement or should stay the same.

## Usability

The new system **shall** include the following:

1. **Ease of Use**

**ID:** 3.1

**Type:** Ease of Use (Usability)

**Description:** The program was created to be an easier photoshop that is just as efficient/accurate. The program shall be easy to use for a young 12-year-old teen to use. The program is meant to help and avoid the user from making mistakes. The program’s tool pop up shall make the user want to use it more as its very convenient. The program can be used by anyone without need of proper English as symbols and pictures on tools will be incorporated (however more languages shall late be incorporated).

**Rationale:** Users want photoshop however do not want to pay money for it. The user also finds photoshop very complicated yet needs to edit their photos professionally. Users do not understand photoshop symbols.

**Fit Criterion:** 90% of younger teenagers shall be able to complete basic editing on their selfies and photos within a couple of minutes. After usage of about one week the user shall have an error rate of less than .5%. Surveys shall be given to identify how long users use the program and specific functionalities to know if symbols were used/understood.

1. **Personalization and Internationalization**

**ID:** 3.2

**Type:** Internationalization (Usability)

**Description:** The program shall allow users to choose from a variety of languages, starting alphabetically. For text boxes a spell checker will be incorporated. The pro-option will come with a purchase option in U.S. dollars and allow for more personal configurations. Currently only purchasable in U.S. dollars however there is an intention to have it purchasable in pesos and other currencies.

**Rationale:** The user may not know English. User’s who prefer photoshops complicatedness and do not mind paying more money. The user may not have the currency in U.S. dollars.

**Fit Criterion:** Purchases will be logged to see which currency is used the most. The program will store the language that is most used on it and display it on a bar chart. 50% of users who use Photo Castle after 2 weeks will purchase the pro-option to utilize more features.

1. **Learning**

**ID:** 3.3

**Type:** Learning (Usability)

**Description:** This program shall be easy to learn for a middle school student. The program was meant for ease of use therefore no training is needed to use Photo Castle. For the older generation, a tutorial and manual is provided in case they need help (Video Tutorials in the future).

**Rationale:** Users who edit photos are all very young teenagers. The user does not wish to waste too much time learning another version of photoshop. The user needs to find a way to learn in case they do choose to put in time.

**Fit Criterion:** 99% of teenagers shall be successful in using all the features photo castle has in less than 3 days. After receiving video tutorials and some training, older people will be able to use the app with less than 10% errors in 1 week. Without using manuals or tutorials the average person can learn Photo castle in 3-6 days.

1. **Understandability and Politeness**

**ID:** 3.4

**Type:** Understandability (Usability)

**Description:** The program shall use very basic verbiage and symbols that most people understand and have seen before in their daily lives. The program shall also hide its complexity and construction so the user can see a very minimal and clean interface.

**Rationale:** Users need to easily understand and quickly see what they are doing on the program. The user is stressed whenever they have to learn new complex programs.

**Fit Criterion:** A survey is administrated to find that 80% of the users are still using the program after 3 weeks. A test is given that shows 99% of the users stayed on the program for more than 15 minutes after the first use.

1. **Accessibility**

**ID:** 3.5

**Type:** Accessibility (Usability)

**Description:** The program shall be usable for those who are color blind with a color-blind help mode. The program shall also have an easy mode with less tools and bigger sizes to adapt to those with disabilities and also following the Americans with disabilities act.

**Rationale:** User is color-blind. The user can have disabilities and needs an easier way to manage a program (accessibility).

**Fit Criterion:** 99% of color-blind users will agree that the program had features to help their disabilities. 90% of users over the age of 55+ will agree that the program had options that would help them see the tools better, and manage the few essential tools they needed.

1. **Convenience**

**ID:** 3.6

**Type:** Convenience (Usability)

**Description:** The program shall give them a small pop up of new changes and features added to Photo Castle. The program will also notify the user of their last work and if they want to continue where they left off.

**Rationale:** User wants their tools to pop up just how they customized them to be when they first opened the program. The user needs to open up a photo editor and pick up where they last left off.

**Fit Criterion:** 89% of users will agree the features Photo Castle contains are helpful and unique, there are no rivals to compete with. Users will always see auto pop up of the tools they last used. The weekly news feed that opens up on the main display will show updates/fixes when the program is opened up.

## Performance

The new system **shall** include the following:

1. **Speed and Latency**

**ID:** 4.1

**Type:** Speed (Performance)

**Description:** The interface the user is given shall have a response time of 1 second for every operation that can be chosen. The response time needs to be fast as the user’s thought process cannot be forgotten or interrupted. The photo manipulating program must download the photo the user wants to manipulate in 10 seconds. The program shall save all the progress and store the photo away within 15 seconds. The program could be delayed in downloading and saving if a bulk of photos are chosen to edit all at once (10 more seconds per photo). Response time to applying effects on photos cannot take more than .5 seconds as applying whitening and waiting for it to show up can disorient the user.

**Rationale:** The user wants the effects applied immediately so they can see what they have so far. The user needs a filter added in bulk without having to wait more than 5-10 minutes.

**Fit Criterion:** The program’s response time shall have no more than 1 second for 70 percent of the actions. The response should only take longer in downloading and saving if a bulk amount of photos are edited at once.

1. **Safety-Critical**

**ID:** 4.2

**Type:** Safety (Performance)

**Description:** The program shall not purposely cause heating or memory issues to the system. The program will not cause malware or contain third party ads that could potentially steal the users privacy/data.

**Rationale:** The user does not want a power soaking, heat causing program. The user is afraid of what other software on the program could cause in terms of viruses/ads.

**Fit Criterion:** The program shall be certified by the AEE’s and other organizations policies and rules. It is to be certified by qualified computer science majors. No third party or other unauthorized users shall manage or edit the program after its testing qualifications and the program shall be given to download once it is safe to administrate. The program will comply with all safety standards such as ethical and mandatory.

1. **Precision or Accuracy**

**ID:** 4.3

**Type:** Precision (Performance)

**Description:** The programs tools all contain accurate measurements in opacity and size of the tools in decimal places (0.00-0.99) and percentages (0-100%). All copies are saved with the accurate changes and a copy is created as well with the same precise changes in case the user chooses to go back.

**Rationale:** The user needs accurate measurements of effects/filters on their photos. The user could change their mind and cannot undo their saved photo.

**Fit Criterion:** The program will give users a bar to change the opacity and size of their tools or a space to enter the amount in decimals. One click filters will have a percentage bar of how much of the effect they desire. 99% of the users will agree that the decimal and percentages is the most accurate way to measure the effects in the span of 1 day.

1. **Reliability and Availability**

**ID:** 4.4

**Type:** Reliability (Performance)

**Description:** The program shall run and be available at all times the user needs it (24 hours a day/365 days per year). If the program were to close, the program will contain autosave and produce a copy for every major change. It is vital to have autosave, so the user does not have to start over again and produce frustration and uninstallation.

**Rationale:** The user wishes to use the program during the day and during the night (emergency needs). The program may crash and the user could lose everything. They need a way to not lose all the work they did.

**Fit Criterion:** The program will have a 90% uptime as its downloadable and usable offline. If a beta online version were to be released, then that would vary between 60-90% uptime. The autosave feature will be added on the online version and can be selected from the file menu.

1. **Robustness or Fault-Tolerance**

**ID:** 4.5

**Type:** Robustness (Performance)

**Description:** The program does contain autosave features so if the program abruptly closes or internet connection is lost, the users work is not lost with it. The program can run offline, so an internet connection is not necessary. If power is lost and the program does not manage to autosave it will recover the last version before a save.

**Rationale:** The user will be using the program and if it fails, they will lose all their work. The user could lose internet and not be able to use the program. The user could lose power and internet as a combination and lose everything they have done.

**Fit Criterion:** When a program closes abruptly, it will prompt a crash panel, after that the user may reopen the program and find a menu to choose from the last copy it saved. 80% of the programs will contain a copy if both power and internet is lost for the user.

1. **Capacity**

**ID:** 4.6

**Type:** Capacity (Performance)

**Description:** The program shall be able to handle 100 images or 400 megabytes at one time (when it comes to bulk editing). Maximum picture size can be up to 100 megabytes. Unlimited editing is allowed and is an important part of the program’s functionality.

**Rationale:** The user wants to edit more than one photo/the photo is more than 100 megabytes. The user wants to edit a million photos individually and save them one by one.

**Fit Criterion:** There is no limit to how many photos can be edited individually. A survey will be issued to discuss the capacity amount. 70% of the users will agree that the capacity is at the perfect range.

1. **Scalability or Extensibility**

**ID:** 4.7

**Type:** Scalability (Performance)

**Description:** The program shall later be capable of 1,000 images or 4 gigabytes of space at one time (from the original 100 images/400 mbs). This program supports about 20 different editing tools which will be extended to 100 tools on the Pro edition.

**Rationale:** The user is currently editing personal photos but needs to edit bulk photos for professional work. The user needs more tools in their professional work apart from the essential tools.

**Fit Criterion:** The pro edition will provide a log on how much its purchased and how much the tools are being used. 80% of the program will be scaled into a much higher capacity total.

1. **Longevity**

**ID:** 4.8

**Type:** Longevity (Performance)

**Description:** The program will continuously receive updates and improve functionality over the span of 5 years.

**Rationale:** The user wants a program that will last forever and have an investment towards their future.

**Fit Criterion:** 99% of the updates will be given on a monthly basis over the span of 5 years. User’s 99% of the time will agree that the updates keep everything running and working for them.

## Maintainability and Support

The new system **shall** include the following:

1. **Maintenance**

**ID:** 5.1

**Type:** Maintenance (Maintainability)

**Description:** A new program update can be seen issued around the first week of launch. A report shall be given thereafter and a weekly feed on improvements and changes coming over the next couple of updates will be given. Users should not have to worry about maintenance work as the program will continue to function offline. Monthly updates at the end of each month will be adding the majority of new tools/functions and bug fixes.

**Rationale:** The user wants updates weekly. The user wants to be notified of updates and improvements. The user wants to keep editing even if the program is being worked on. The user wants maintenance done on programs that are not working.

**Fit Criterion:** Maintenance will be done 50% of the time during the 12pm eastern time on the last week of the month. A weekly feed is going to appear when the program is first opened.

1. **Supportability**

**ID:** 5.2

**Type:** Supportability (Maintainability)

**Description:** The program will contain pop ups and tutorials on how to use each feature to help the user understand what each tool does. A more dedicated and detailed tutorial will be located in the help section of the program. If any issues are perceived or questions need to be asked there will be a forum dedicated to questions and bugs/issues. A forum support team will communicate these issues to the programmers and a fix shall be incorporated and a reply in as little as one week shall be issued.

**Rationale:** The user does not understand the new tools to edit photos. The user needs a manual. The user finds an error or is lost and needs to contact someone.

**Fit Criterion:** Forum support teams will be contacted through the help section. The forum support team will provide the issues/bugs to the programmers every week. 99% of the issues will be resolved through the support team after a week.

1. **Adaptability**

**ID:** 5.3

**Type:** Adaptability (Maintainability)

**Description:** The program is expected to run under Windows 10 and Linux systems. The program currently is installable in the United States however international downloads for other countries such as Mexico and Japan will be integrated in their respective languages. The program currently is dedicated for computers however a mobile and tablet version is intended to be developed.

**Rationale:** The user has a Windows system and a Linux system. The user is outside of the U.S. and does not speak English. The user wants to transition their work from a pc to a mobile device.

**Fit Criterion:** A download will be available in more than one language for other countries outside of the U.S. The mobile version of Photo Castle will begin testing on beta after the main program on pc is cleaned up from major bug issues.

## Security – FUNCTIONAL REQUIREMENT

The new system **must** include the following:

1. **Access**

**ID:** 6.1

**Type:** Access (Security)

**Description:** Only high-level cyber security personnel will access and manage accounts.

Forum support team can only access public data from accounts and therefore must contact cyber security personnel for more detailed info such as password resets and changes/deletions.

**Rationale:** The user wants to know who has access to their account. The user does not want anyone else who is not certified to have their information.

**Fit Criterion:** 99.9% of the accounts will be accessible to and only to the cyber security personnel.

1. **Integrity**

**ID:** 6.2

**Type:** Integrity (Security)

**Description:** The program does not store and does not upload any photos the user is editing. The program will include a layer of security to access accounts such as a two-factor authentication to prevent external attacks. High-level cyber security personnel will be certified and held accountable to weekly logs.

**Rationale:** The user wants to keep all their photos private. The user does not want to lose their identity. The user wants security on their accounts. The user wants accountability in the program and the users running the program.

**Fit Criterion:** Weekly logs will be issued to everyone working for Photo Castle. A two-factor authentication must be implemented before using Photo Castle.

1. **Privacy**

**ID:** 6.3

**Type:** Privacy (Security)

**Description:** If any data must be collected from the user, they will be aware and it shall be known what shall be conducted on their personal information. If any changes occur in our policy the user shall be sent a pop up upon opening the program. According to the relevant laws on privacy and private information, all practices shall be followed according to policy.

**Rationale:** The user wants to know what is going to be done with the information they provide. The user wants to know if anything changes from their original policy they accepted. The user wants the program to follow the policies every other program follows.

**Fit Criterion:** The most up to date policy will be posted on the info section of the program. 99% of the users will be informed on policy changes/data collection.

1. **Audit (what information must be recorded to allow security checks. E.g., logs)**

**ID:** 6.4

**Type:** Audit (Security)

**Description:** If a user opens the program a log is created, and their login time and logout time is recorded to measure stats and obtain accurate information on usage. Security checks will be running monthly with the updates and weekly logs will be assessed to higher-level personnel. These actions support security for the user on where and when the program was accessed which can be sent to the user on request.

**Rationale:** The user wants to know who logs-on their account and log on times. The user wants weekly logs on security checks and assessment of higher personnel.

**Fit Criterion:** On the account section of the program the user will get to see when they logged on and logged off and what computer they did it from. 89% of the weekly logs will contain security checks and assessment of the higher personnel.

1. **Immunity**

**ID:** 6.5

**Type:** Immunity (Security)

**Description:** A security program is to be implemented on the program to prevent malicious software on existing programs that run onto photo castle. Ads and third party programs are not incorporated into this program which helps prevent them from outside attacks. This program is to be built as secure as it can be so the user can edit without any malicious attacks or disturbances.

**Rationale:** The user wants a security protocol. The user does not want ads nor a third party program running on their system. The user wants a secure photo editing app to help them without causing malicious attacks.

**Fit Criterion:** 99% of the programs that run through Photo Castle will be scanned and assessed. No ads will be implemented in the program.

## Cultural

The new system **shall** include the following:

1. **Cultural**

**ID:** 7.1

**Type:** Cultural

**Description:** The program shall not be offensive to any religion or ethnic group. The program shall provide options for different languages and provide the appropriate spell checks. The program shall keep in mind all holidays for all countries and their system of measurements. Due to its minimalistic look, it will not have any issues with cultural diversity. Every symbol and color used will be checked to see if there are different meanings in another culture to avoid these culturally diverse issues.

**Rationale:** The user has a religion that does not allow certain symbols. The user speaks many languages and wants to change between them. The user can use a program that is not offensive to their culture.

**Fit Criterion:** A test panel will be shown to every user who downloads Photo castle and uses the program for 1+ day. The panel will display a result of 90% cultural diversity and provide each symbol a future in the toolbar.

# project implementation description & explanation

## Users Main Screen

Graphical user interface, text, application

Description automatically generated

* When you open Photo Castle this is the screen you will always load up to. As you can see it has a very similar format to photoshop.

## Main Screen Tools Tutorial

Graphical user interface, text, application

Description automatically generated

## Red Box Tools (from left to right)

* Tool #1: New Image
* Tool #2: Open Image
* Tool #3: Save Image
* Tool #4: Print Image
* Tool #5 & 6: Undo & Redo
* Tool #7: Image size
* Tool #8: Canvas size
* Tool #9: Add text box
* Tool #10: Batch files
* Tool #11: Zoom In/Out

## Green Box Tools (from top to bottom)

* Tool #1 & 2: Lower & Higher brightness
* Tool #3 & 4: Lower & Higher contrast
* Tool #5 & 6: Lower & Higher gamma correction
* Tool #7 & 8: Lower & Higher saturation
* Tool #9 & 10: Lower & Higher brightness
* Tool #11: Soften
* Tool #12: Blur
* Tool #13: Sharpen
* Tool #14: Reinforce
* Tool #15 & 16: Flip vertical & Flip horizontal

## Blue Box Tools (from left to right & top to bottom)

* Tool #1: Selector
* Tool #2: Color Picker
* Tool #3: Magic Wand
* Tool #4: Lines
* Tool #5: Paint Bucket
* Tool #6: Spray Can
* Tool #7 & 8: Brush & Advanced Brush
* Tool #9: Stamp
* Tool #10: Blur
* Tool #11: Erase
* Tool #12: Smudge

## Example 1: Flower Case

Graphical user interface

Description automatically generated

* When you open Photo Castle and click on the open image button. You will have your files open, and you will be able to choose an image from your files. In this example we will use the flower.jpeg. Lets say we want to make this photo darker, and we want a sharp and black style image. We apply the sharpen tool once, the grayscale tool once, and the lower brightness three times. This gives us our result which we then hit save image since we are satisfied.

Graphical user interface

Description automatically generated

* This is the result image of the flowers edited and the tools did in fact apply the correct effects.

## Example 2: House Case

A screenshot of a home

Description automatically generated with medium confidence

* When you open Photo Castle and click on the open image button. You will have your files open, and you will be able to choose an image from your files. In this example we will use the house.jpeg. Let’s say we want to make this photo more saturated, flipped horizontally, and add some touch ups. We apply the saturation tool twice, the contrast tool once, flip it horizontally, and reinforce it once more. This gives us our result which we then hit save image since we are satisfied.

Graphical user interface

Description automatically generated

* This is the result image of the house edited and the tools did in fact apply the correct effects. You can see the house is now flipped and it looks sharper with a much higher saturation.

## Example 3: Family Case

Graphical user interface, application

Description automatically generated

* When you open Photo Castle and click on the open image button. You will have your files open, and you will be able to choose an image from your files. In this example we will use the family.jpeg. Let’s say we want to make this photo have random effects given by our tools in the right-hand side. We apply brush tool and create red squares and create green circles, we also add a line pointing to the family using the lines tool. We want to mess with the advanced brush, so we add some yellow scribbles and add a tiny text box as well and erase the bottom half of the image. This gives us our result which we then hit save image since we are satisfied.

Graphical user interface, application

Description automatically generated

* This is the result image of the family edited and the tools did in fact apply the correct effects. You can all of our random effects were applied to the image.

# Test plan

**Introduction of Test Plan**

* The objective of Photo Castle 1.0 is to be the new competitor in the photo editing software market. Photo Castle wants to offer everything Photoshop offers with a neat and simple user interface at no cost for the customer. The project goal is then to offer a photo editing software free of cost that can offer photoshop tools with a much simpler UI. The first release of this software will be named and known as Photo Castle 1.0
* The audience that is targeted are the people who want to edit their photos professionally however they do not want to pay the premium prices photoshop currently offers. The audience is also not looking find a new learning curve in a program that is like photoshop. The program that is going to be produced will be guided to the ease of use for the customer and will provide many great tools that photoshop currently offers.
* The issue is that the Photo Castle 1.0 software needs to be tested in a proper manner for it to work for its customers. The testing should be organized into levels to deeply find any errors and to correct them before any release is sent out. These levels are listed as following:
  + Code Quality and Dependency
  + Main System Interface
  + System Quality
  + Tools Implementation and Use

In these levels they are all going to be tested for the same aspects. The Code quality is to make sure that testing on a small set of code will work and integrate correctly. The main system will mostly provide performance, ease of use, reliability, and functionality. The system quality is then the acceptance of the systems incorporations. The final test is the use of all tools and its functionality and performance on the photos.

**Test Items**

* The Items that will be tested are:
  + Photo Castle 1.0 Software and Infrastructure
  + Other supporting software

**Features to be Tested**

* The testing will include many levels as stated in the introduction, in every level there will be testing of the following for the Photo Castle software.
  + Accessibility
  + IEEE Coding Standards
  + Compatibility
  + Functional
  + Good Content
  + Navigation
  + Reliability
  + Performance
  + Security
  + Usability
  + Organization

**Features Not to be Tested**

* All features will be tested at every level following the aspects mentioned before however if the level is not of high importance and is of low importance than its possible it may be skipped.

**Approach**

* The testing on Photo Castle will be of priority testing. Each level will be tested however each level will have its importance such as high, medium, and low testing. This rule shall be followed unless the low testing is a pre-requisite of a higher level. The testing will be of all manual testing done by the tester or professor. The tester will go to each tool and test its accessibility and functional-use first. If the tool works well the reliability is marked off and its performance level as well. The tester will then look for any vulnerabilities in the tool and test for the IEEE coding standards. If all aspects are checked off then the level is complete and the tester can continue to another art tool. The main tools are the high priority while the subcategories of the tool will be of medium priority. The low priority are the tools that will be added as an extra and not a requirement. The end approach will be the time it took to test Photo Castle tools, the amount of test cases produced and incidents. The performance will be timed and the organization of the tool will be criticized.

**Item Pass/Fail Criteria**

* The Item Pass Criteria for each level of testing will need to be passed in order to continue to the next level. The approval of this level being passed will be of the Professor when he is checking off every aspect. The Item Fail Criteria will also be of the Professors decision. If the level cannot meet all of the aspects to be tested then the project Photo Castle 1.0 will be delayed or restarted.

**Suspension Criteria and Resumption Requirements**

* If the testing is being done on Photo Castle 1.0 and the software crashes, then the testing must be stopped. If the testing is stopped because of a crash, then the software must be restarted. Once the program starts up again then the testing may resume. Testing can also be skipped for a subcategory tool if the main tool previously failed.

**Test Deliverables**

* The Test Deliverables include the following:
  + Master Test Plan (this document itself)
  + Test Cases
  + Test Scripts
  + Defect/Enhancement logs
  + Test Reports

**Test Environment**

* Photo Castle 1.0 will have the test environment hardware of their own personal computer to test. At first release the Photo Castle program will only be an executable and will not have any network connections needed. The hardware requirements for a laptop/desktop are i5, 8gb ram, colored screen, running windows 10, and an HD display is preferred but not needed. The tools needed to test Photo Castle 1.0 are simple manual calculations and automated tools will not be needed.

**Schedule**

* The following schedule will be met:

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Start Date | End Date | Duration |
| Project Proposal and Requirements Document | 08/20/2021 | 08/30/2021 | 10 days |
| Draft Test Plan & Project Update | 08/20/2021 | 10/08/2021 | 18 days |
| Updating the Test Plan by its feedback | 10/09/2021 | 11/21/2021 | 44 days |
| Continuing the Project Construction | 10/09/2021 | 11/21/2021 | 44 days |
| Final Test Plan & Project Repository | 11/21/2021 | 11/22/2021 | 1 day |

**Staffing and Training Needs**

* Summary: The Photo Castle Lead manager will make sure that staff that are going to work on the project have experience with:
  + General testing techniques
  + Knowledge of other photo editing programs
  + Knowledge of testing tools that will be used
  + Photo Castles requirements document and project proposal

**Responsibilities**

* The following individuals will be responsible for:
  + Brian Diaz (Creator of Photo Castle) – Project Lead and Tester
  + Dr. Sean Hayes (Director/Project Overseer) – Senior Project Approval and Tester

**Risks**

* The following identifies all the likely risks the project may undergo:
  + The software becomes unusable after a crash. The testing is then delayed until the program can be fixed by the lead programmer.
  + The main tools to edit photos are not working. The testing is delayed again until the programmer can fix the tools.
  + The tools of the program cause errors in other categories to occur. The testing is delayed once again and the program may need to be restarted.
  + Not enough time, the project may undergo many errors and test cases and may not meet the deadline to turn in the project.
  + Communication between tester and programmer may not understand each other. This causes the program to run even more time wasted without being tested.

# challenges overcome

* Challenge #1: Figure out how the main interface will look, feel, and features to be added. The brainstorming process took a while as many decisions were made in how the entire UI would look and operate.
* Challenge #2: Make sure I know the libraries and packages that need to be included for my code to work using qmake and qt creator.
* Challenge #3: Begin coding in C++ for the base program and make sure the base cases/functions work, there are over 20 tools to customize so it took a lot of time to make sure they work, the best option was to create individual files for each tool.
* Challenge #4: Learn how Qt operates, functions, and feels while creating a photo editing software.

# future enhancements

## Photo Castle 2.0

* Enhancement #1: Include a Dark/Light mode.
* Enhancement #2: Enhance the UI into something even more eye pleasing that is simple and unique.
* Enhancement #3: Introduce the purchase version that has the advanced tools for the pros that need it.
* Enhancement #4: Become a bigger competitor to Photoshop.