SE 3XA3: Development Plan
Image Processing Application

Team 1, Team NAR
Rahul Bablani - bablanr
Abeed Alibhai - alibhaa
Nezar Dimitri - dimitn

December 8, 2016

Contents

1 Project Drivers **1**

1.1 The Purpose of the Project 1
1.2 The Stakeholders 1
1.2.1 The Client 1
1.2.2 The Customers 1
1.2.3 Other Stakeholders 1
1.3 Mandated Constraints 2
1.3.1 Solution Constraints 2
1.3.2 Implementation Environment of the Current System . 2
1.3.3 Partner of Collaborative Applications 3
1.3.4 Off-the-Shelf Software 3
1.3.5 Anticipated Workplace Environment 3
1.3.6 Schedule Constraints 3
1.3.7 Budget Constraints 3
1.4 Naming Conventions and Terminology 3
1.5 Relevant Facts and Assumptions 4

2 Functional Requirements **4**

2.1 The Scope of the Work and the Product 4
2.1.1 The Context of the Work 5
2.1.2 Work Partitioning 5
2.1.3 Individual Product Use Cases 7
2.2 Functional Requirements 7

3 Non-functional Requirements **10**

3.1 Look and Feel Requirements 10
3.2 Usability and Humanity Requirements 10
3.3 Performance Requirements 10
3.4 Operational and Environmental Requirements 11
3.5 Maintainability and Support Requirements 12
3.6 Security Requirements 12
3.7 Cultural Requirements 12
3.8 Legal Requirements 12
3.9 Health and Safety Requirements 13

4 Project Issues **14**
4.1 Open Issues . . . 14
4.2 Off-the-Shelf Solutions . . . 14
4.3 New Problems . . . 14
4.4 Tasks . . . 14
4.5 Migration to the New Product . . . 15
4.6 Risks . . . 15
4.7 Costs . . . 15
4.8 User Documentation and Training . . . 15
4.9 Waiting Room . . . 15
4.10 Ideas for Solutions . . . 16

5 Appendix **16**
5.1 Symbolic Parameters . . . 16

List of Tables

1 **Revision History** . . . iii
2 **Work Partitioning** . . . 6

List of Figures

1 Context of the Work . . . 5
2 Uses Case Diagram . . . 7

Table 1: **Revision History**

Date	**Version**	**Notes**
Oct 6, 2016	0	Rev0 - Split up parts and started documentation.
Oct 11, 2016	0.1	Rev0 - Finished separate parts and compiled document.
Dec 7, 2016	1	Rev1 - • Added professor to stakeholders • "Context of the Work" explained • Eye strain added to safety • Added upcoming features to waiting room

1 Project Drivers

This section provides information about the project drivers within our project.

1.1 The Purpose of the Project

The purpose of this project, is to re-implement the open source image processing project *Marvin Frameworks*. This Java framework allows clients to upload and process normal photos, into elegant images by applying one of the various filters available in our application, much like Instagram does. Once properly implemented, the processed image will then be available to export as an image file for use in social media, advertisements, etc.

1.2 The Stakeholders

Stakeholders include: social media users (Instagram, Snapchat), photo/video editors, Team NAR (us), Spencer Smith (Professor) and future developers who may take over the project if it requires more man-power.

1.2.1 The Client

Everyday photo takers wishing to transform their images into something more elegant by applying photo filters.

1.2.2 The Customers

Users of social media outlets such as Instagram or Snapchat. This can be a platform to further build on those applications by adding new filters and implementing unique video processing filters that neither application currently supports.

1.2.3 Other Stakeholders

Contributors: Future and current developers interested in the development process of this project and are willing to share their knowledge in order to further improve the image processing framework.

Testers: This may include developers of our application, or general users of the program who will test the projects functionality and usability.

Image Processing Specialist: Developing new in-depth image processing algorithms and possibly video processing algorithms as well. Our team will require an expert in the field in order to make future upgrades possible. The addition of a specialist will really allow our application to include some unique processing powers that will take us to the next level of image computing and allow us to compete with the likes of Instagram and Snapchat.

1.3 Mandated Constraints

This section provides information about the mandated constraints we face in our project.

1.3.1 Solution Constraints

- **Description:** Project must be completed by the week of November 14th, 2016
  Rationale: Date that revision 0 demonstration is due
  Fit Criterion: N/A

- **Description:** The Java application must be functional on all operating systems
  Rationale: Users should not be limited to only using Windows or just 1 operating
  system but include as many as possible allowing the product to potentially reach as many users as possible
  Fit Criterion: Testers and users will ensure that it works for all operating systems

1.3.2 Implementation Environment of the Current System

This project will be coded in Java, using the Marvin Image Processing Framework.

1.3.3 Partner of Collaborative Applications

N/A

1.3.4 Off-the-Shelf Software

There are many implementations of Marvin frameworks in Java, which we plan on using as a reference to ensure we are following the standard coding style.

1.3.5 Anticipated Workplace Environment

We plan to make our application available on any computer system (Window, Mac, Linux).

1.3.6 Schedule Constraints

- Test Plan Revision 0: October 28th, 2016
- Proof of Concept Demonstration: Week of October 17th, 2016
- Design Document Revision 0: November 11th, 2016
- Revision 0 Demonstration: Week of November 14th, 2016
- Final Demonstration: Week of November 28th, 2016
- Final Documentation: December 7th, 2016

1.3.7 Budget Constraints

Currently there is no budget constraint for this project, it will be open source and free for use when created. Later into the product lifecycle we may look at monetization.

1.4 Naming Conventions and Terminology

Image Filter: A software routine that changes the appearance of an image or part of an image by altering the shades and colours of the pixels in some manner. Filters are used to increase brightness and contrast as

well as to add a wide variety of textures, tones and special effects to a picture.

Image Processing: The analysis and manipulation of a digitized image, especially in order to improve its quality.

Framework: In computer programming, a framework is an abstraction in which common code providing generic functionality can be selectively overridden or specialized by user code providing specific functionality.

1.5 Relevant Facts and Assumptions

- Users will have a basic understanding of how to work a computer
- The user's desktop can support a Java application

2 Functional Requirements

This section provides information about functional requirements implemented in our project.

2.1 The Scope of the Work and the Product

Our image processing app is designed to provide a reliable, user-friendly, and simplistic approach to editing ones photos. Through the use of a vast library of Marvin Frameworks, we are able to offer the user with many options on how they want to edit their photos. The app enables the user to quickly access common tasks through an interactive interface. By using graphical utilities such as buttons to access image processing functions, users can efficiently explore and converge on a solution for a particular image processing problem.

2.1.1 The Context of the Work

refer to *Figure 1*

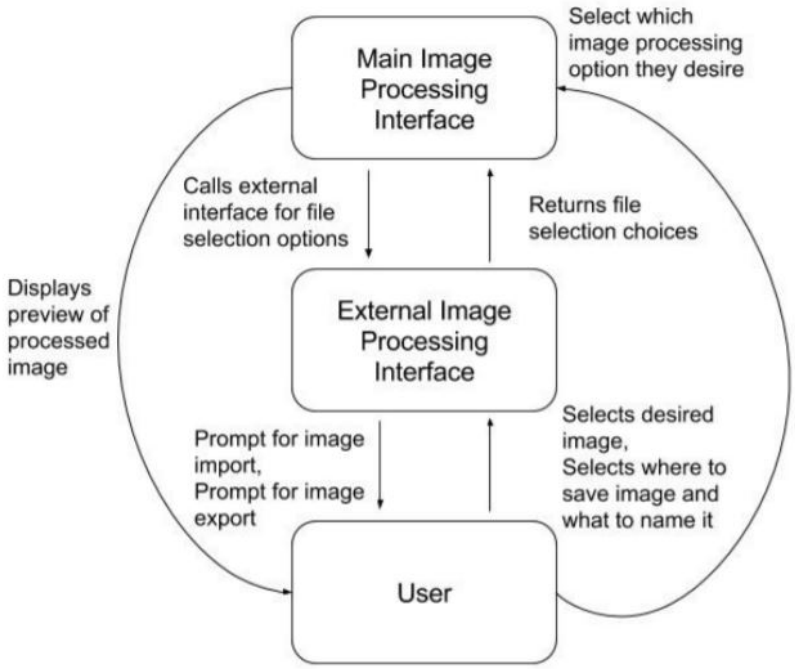

Figure 1: Context of the Work

Explanation:

- The external entity is the user who initially selects the image they would like to process. The image then goes through the external image processing interface which calls the Marvin framework to apply the users inputted desired filter. The interface then displays a preview of the image and this cycle continues until the user exports the file into a raster image file they see suitable.

2.1.2 Work Partitioning

refer to *Table 2*

Table 2: **Work Partitioning**

Event	Input/Output	Summary
1. Application prompts user to select file	none	Application asks user to select an image of file type png/jpg from their local drive.
2. User selects image	File from local drive of type png/jpg	User chooses desired image to edit through interface.
3. Application uploads selected image to main interface	none	Upon selection, users desired file is uploaded onto main interface.
4. User selects desired processing option	JButton selected, MarvinPluginLoader. loadImagePlugin(desired option)	User chooses one of the options on the main interface and their choice is processed.
5. Application processes image	Preview of the processed image shows up on main interface	Based on the user's choice, the application selects the appropriate plugin and processes the image, then displaying it on the main interface.
6. Application prompts user to export image	none	Through an external interface the user is prompted to save their processed image to their local drive.
7. User selects destination and name of file	Processed image stored to local drive	The user may choose where to store the processed image as well as what to name it. Once completed file is saved to local drive.

2.1.3 Individual Product Use Cases

refer to *Figure 2*

. . .

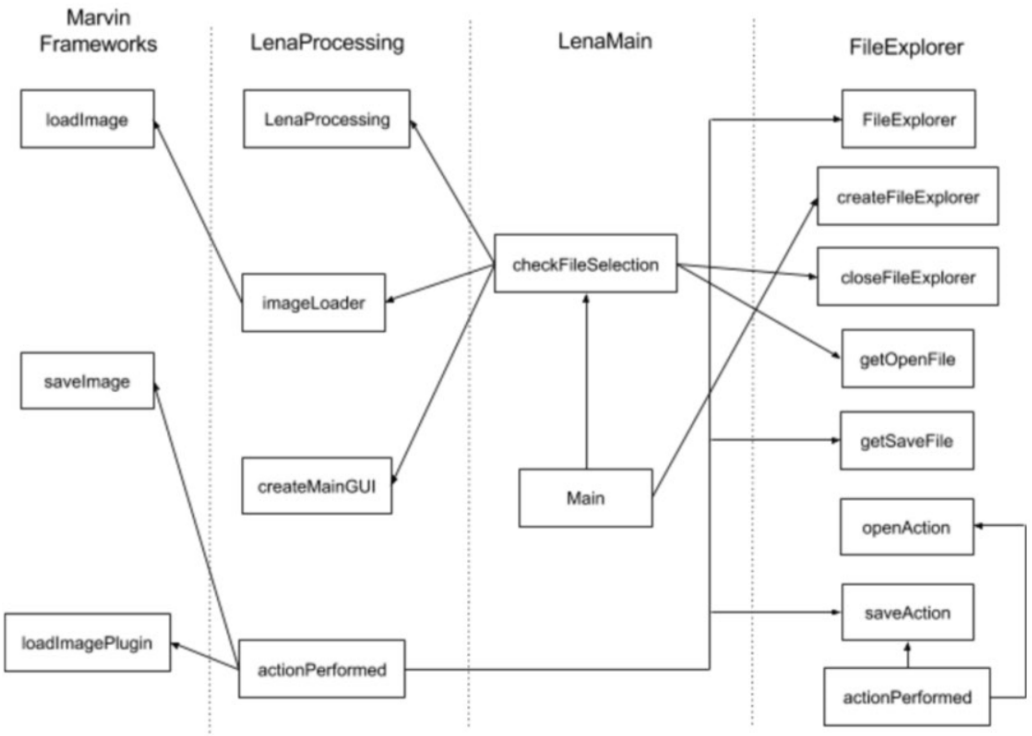

Figure 2: Uses Case Diagram

2.2 Functional Requirements

Requirement # 1
Requirement Type: 9
Event/use case: 1
Description: The application must be able to accept an image of file type png or jpg.
Rationale: Images come in different formats and sizes, and to truly be use friendly more than one file type should be accepted.
Fit Criterion: Application can accept and process pngs and jpgs.

Requirement # 2
Requirement Type: 9
Event/use case: 2,3,4
Description: The application must be able to select correct plugin corresponding to selected processing option.
Rationale: There are many processing options that the user can choose from, and the program should be able to recognize the chosen one, and process the image accordingly.
Fit Criterion: The image is processed correctly.

Requirement # 3
Requirement Type: 9
Event/use case: 2,3,4
Description: The application must display a preview of the selected processing option upon selection.
Rationale: Allows users to see how the option they selected effects their chosen image.
Fit Criterion: The image is processed correctly.

Requirement # 4
Requirement Type: 9
Event/use case: 5
Description: Once the user is satisfied with their edit, the application must prompt them with an external interface, asking them to save it to their local drive.
Rationale: Most users will be editing their image with a purpose in mind, maybe they want to post it on social media or use it on their personal website. Regardless, having an option to export would fulfill this need.
Fit Criterion: User is prompted to export image.

Requirement # 5
Requirement Type: 9
Event/use case: n/a
Description: The application must respond appropriately to the closing, minimizing, maximizing, as well as resizing functions of the window.
Rationale: These are basic functions that most users are already familiar with and that the application should follow too.
Fit Criterion: Application closes when the "X" button is pressed, minimizes once "-" button is pressed, maximized when the "+" button is pressed, and resizes in the direction of the mouse drag.

3 Non-functional Requirements

This section provides information about non-functional requirements implemented in our project.

3.1 Look and Feel Requirements

- This application's user interface will be visually appealing and will operate seamlessly on any operating system.
- The application will take a new approach on the way users edit and filter their photos, they will be able to achieve new and better results to attain the perfect image.

3.2 Usability and Humanity Requirements

- The software will have a simple UI that will allow for any individual no matter their age or familiarity with image processing to be able to use the application.
- There will be no manual or guide on how to use the application, therefore The UI must be extremely intuitive to result in a plane learning curve.
- The software will use a standard coding style and will be well documented for anyone interested in examining the code.

3.3 Performance Requirements

Speed

- The app should take no longer than PROCESSING_TIME to load.
- Users will be able to upload an image in a maximum of UPLOAD_TIME.
- All interaction between a user and the UI will be instantaneous. Response times should not exceed RESPONSE_TIME to complete any action. This includes all the different filters that are available.

Reliability

- For now we plan to release on desktop computer as this is a java application. Due to the nature of not actually needing a wireless connection nor connection to any server, the user can use this app freely, and whenever they see fit.

- After taking the appropriate measures to accommodate for any risks that may arise during the use of the application, the program should not crash but instead send an error message if misused.

Quality

- Through no point in the image processing (including addition of filters and saving the new image) will the resolution be affected from its original value nor will any form of distortion occur.

Capacity

- Due to the local nature of the program and its independency from any sever there is no limit to the amount of people that may be processing there images.

- Storage is based solely on the limitations of each individual user. The application can support processing of STORAGE amount of photos.

3.4 Operational and Environmental Requirements

Physical Environment

- The application will be available to anyone wishing to process an image.

Technological Environment

- The software will run on any desktop computer and is supported on all operating systems.

- At the moment any image size will be accepted as long as the file format is either a PNG or a JPG and will be returned as a PNG or JPG.

- All app data and images will be stored within each users local hard drive or any other personal storage device.

3.5 Maintainability and Support Requirements

- The program will be continuously updated to fix any bugs and add more functionality.
- New patches will be released whenever we have made any major updates or fixes so users can experience the best software we have to offer.
- Proper documentation will be maintained as the software expands and new code is added.
- To broaden the scope of the project we plan to start accepting more file types as well as introduce video processing capabilities.
- The program will perform on any desktop but will also be portable, meaning it can be stored on a USB or external hard drive and run from there.

3.6 Security Requirements

- Using correct segregation of the design, we can achieve information hiding thus protecting us from potential intellectual property theft.

3.7 Cultural Requirements

- Filters will not resemble any offensive, profane or non-ethical ideologies.
- The application will include filters which parallel current pop culture.

3.8 Legal Requirements

- Throughout the documentation we reference the use of The Marvin Framework. Our citation legalizes the use of their plugins.
- No legal documentation is needed in terms of the images themselves as the application is only responsible for the processing of images not the supply of them.

3.9 Health and Safety Requirements

- The occurrence of an eye strain from using the application for too long could be possible issue, if this occurs take a minimum 15 minute break from using digital screens but if symptoms still persist contact your doctor.

4 Project Issues

This section provides information about the issues we face in our project.

4.1 Open Issues

At the current stage of development there are no issues to be considered.

4.2 Off-the-Shelf Solutions

There are many implementations of Marvin frameworks in Java, which we plan on using as a reference to ensure we are following the standard coding style.

4.3 New Problems

Currently we are using the Marvin framework image processing pre-set filters and implementing them for a better user experience so no new problems should arise. We are also considering adding video processing at this point but it may be out of the scope for this project due to our time constraint, it would be a lot harder to implement resulting in more natural occurring problems.

4.4 Tasks

- Requirements document revision 0
- Test plan revision 0
- Proof of concept demonstration
- Design document revision 0
- Revision 0 demonstration
- Test report revision 0
- Final Demonstration revision 1
- Write final revisions to document

4.5 Migration to the New Product

N/A

4.6 Risks

There are no legal risks our products can violate due to the fact we do not store any personal information. The only risk our application can encounter is if our code is not efficient and uses too much ram when processing images and can cause the computer to overheat but no serious risk is applicable to our application.

4.7 Costs

There are no costs associated with the development of the project because we are redeveloping an open source project. If we plan to further development and add more features we could possibly need a bigger team but for now there will be no direct costs.

4.8 User Documentation and Training

For our project we plan to optimize the usability and implement a UI that is as straight forward as possible allowing any age group to be able to use our application. For running our executable JAR file we have added a readme file to explain the steps to process your first image.

4.9 Waiting Room

- Future releases my include buttons in different languages
- Our application could possibly implement video filter processing
- In the near future we plan to add the ability to use your webcam in combination with the application
- We plan to implement a web interface similar to Instagram

4.10 Ideas for Solutions

- Users of the application will be able to download the the program in the language of their choice
- Create a functionality to allow video files to be processed. With the current time constraint this may not be plausible but with more time or more developers this can be a future addition to the application.

5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

5.1 Symbolic Parameters

- PROCESSING_TIME = 3 Seconds
- UPLOAD_TIME = 2 Seconds
- RESPONSE_TIME = 1/2 a Second
- STORAGE = Disk Space of User