*University Polytechnica of Bucharest*

*Faculty of Engineering in Foreign Languages*

*Software Engineering Master, year II, group 1272E*

Software Design Document

“Voting-Based Image Binarization”

**Team:**

**Barbu Sabina**

**Poșircaru David**

**Radu Ioan Cosmin**

Table of Contents

[1. Document purpose 3](#_Toc371248635)

[2. Objectives 3](#_Toc371248636)

[3. Document overview 3](#_Toc371248637)

[4. Data design 3](#_Toc371248638)

[4.1 Global data structures 3](#_Toc371248639)

[4.2 Linking 3](#_Toc371248640)

[4.3 Temporary 3](#_Toc371248641)

[4.4 File formats 4](#_Toc371248642)

[4.5 Database description 4](#_Toc371248643)

[4.6 Database structure diagram 4](#_Toc371248644)

[4.7 Table description 4](#_Toc371248645)

[5. Architectural design 4](#_Toc371248646)

[5.1 System architecture 4](#_Toc371248647)

[5.2 Architectural patterns 5](#_Toc371248648)

[5.3 Architecture diagram 5](#_Toc371248649)

[5.4 Description 5](#_Toc371248650)

[5.5 Implementation requirements 6](#_Toc371248651)

[5.6 Component interaction 6](#_Toc371248652)

[6. User interface design 6](#_Toc371248653)

[6.1 Flow chart 6](#_Toc371248654)

[6.2 Screen images 6](#_Toc371248655)

[7. Testing issues 6](#_Toc371248656)

[7.1 Critical components 6](#_Toc371248657)

[7.2 Alternatives 6](#_Toc371248658)

# 1. Document purpose

This software design document describes the data, architectural and user interface design of the “Voting-Based Image Binarization” project, containing the software structure and components necessary for implementation.

# 2. Objectives

The main objective of this document is to provide roadmap for the developers, offering technical details about the way the project should be implemented.

# 3. Document overview

# 4. Data design

## 4.1 Global data structures

To be completed!

## 4.2 Linking

Because the project consists in two parts: a “Binarization Algorithm Module” (BAM) and a “Voting Binarization Algorithm Module” (VBAM) we have to consider the way the two modules are linked.

The “Binarization Algorithm Module” (BAM) output is a 1bpp image, *output\_*image and an 8bpp image, *output\_image-*confidence. The first image is the actual binarization and the second a gray-scale image containing the confidence for the binarization for every pixel. 0 means that the respective pixel was randomly assigned a color (black or white); 255 means that the algorithm is absolutely certain that the respective color of the pixel is correctly assigned.

This output will be then given as input for the “Voting Binarization Algorithm Module” (VBAM).

## 4.3 Temporary

There are no temporary files because the output of the “Binarization Algorithm Module” (BAM) will be saved permanently for verifications even after it is given as input for the “Voting Binarization Algorithm Module” (VBAM).

## 4.4 File formats

The following file formats are used in the program:

* The input file for the “Binarization Algorithm Module” (BAM),which is a Bitmap image (bmp extension).
* The two images which are outputs of the BAM , also in Bitmap format
* the input of the “Voting Binarization Algorithm Module” (VBAM) is also in Bitmap format.
* The output VBAM is in Bitmap format also.

## 4.5 Database description

The project does not need a database as the images will be saved in folders and will have suggestive names.

# 5. Architectural design

## 5.1 System architecture

The project has in two parts: a “Binarization Algorithm Module” (BAM) and a “Voting Binarization Algorithm Module” (VBAM).

The “Binarization Algorithm Module” (BAM) is an executable which receives as input from the command line two file names: *input\_image* and an *output\_image*.

The BAM will return an error code: zero for no error and nonzero in case an error occurred, specifying the error type.

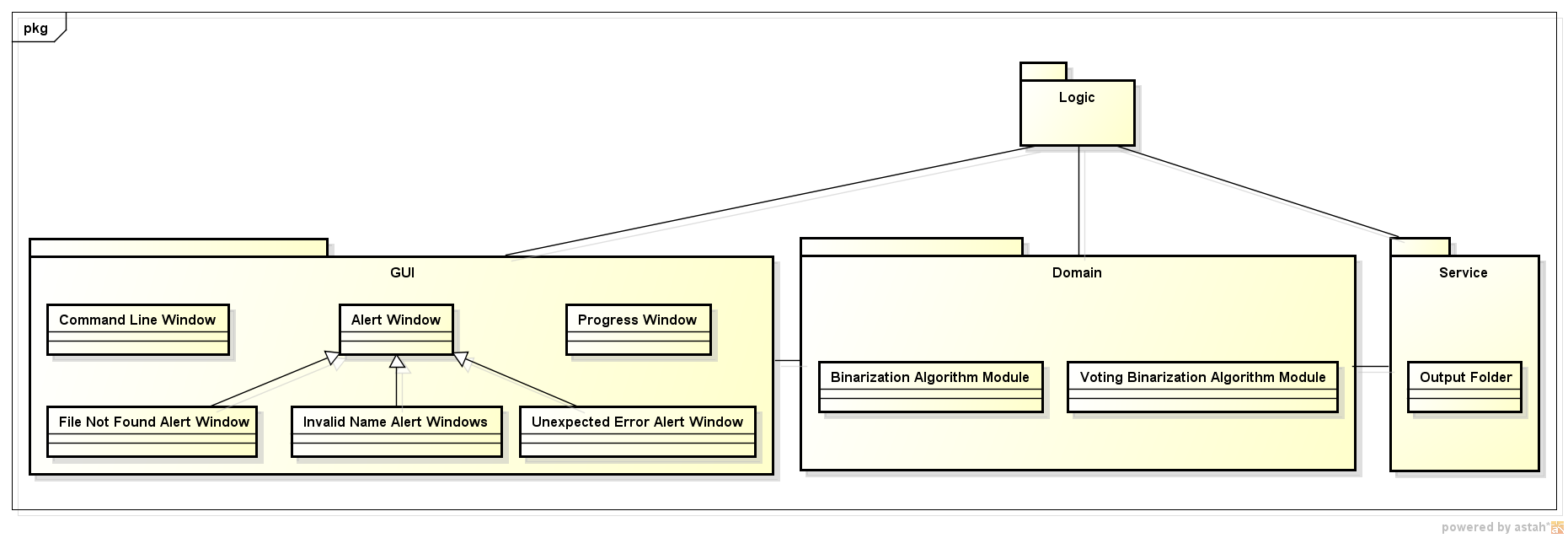
The “Binarization Algorithm Module” (BAM) output is a 1bpp image, *output\_image* and an 8bpp image, *output\_image-*confidence. The first image is the actual binarization and the second a gray-scale image containing the confidence for the binarization for every pixel. 0 means that the respective pixel was randomly assigned a color (black or white); 255 means that the algorithm is absolutely certain that the respective color of the pixel is correctly assigned.

This output will be then given as input for the “Voting Binarization Algorithm Module” (VBAM) in case the error code is zero, meaning there were no errors, otherwise the result won’t be considered. The VBAM can receive as input any number of BAM outputs and performs an “educated voting”. This means that VBAM can take into consideration things as the number of images that say a certain pixel is white or black and/ or a weighted decision based on the confidence image.

## 5.2 Architectural patterns

To be completed!

## 5.3 Architecture diagram



## 5.4 Description

The architecture consists in three main parts: a graphic user interface (GUI), the domain and the service part with the logic part supervising them.

In the GUI we support the command line input for the *input\_image* and an *output\_image* names which are introduces by the user. Also in the GUI we alert the user in case errors appear, such as file not found, invalid names given by the user and other unexpected errors.

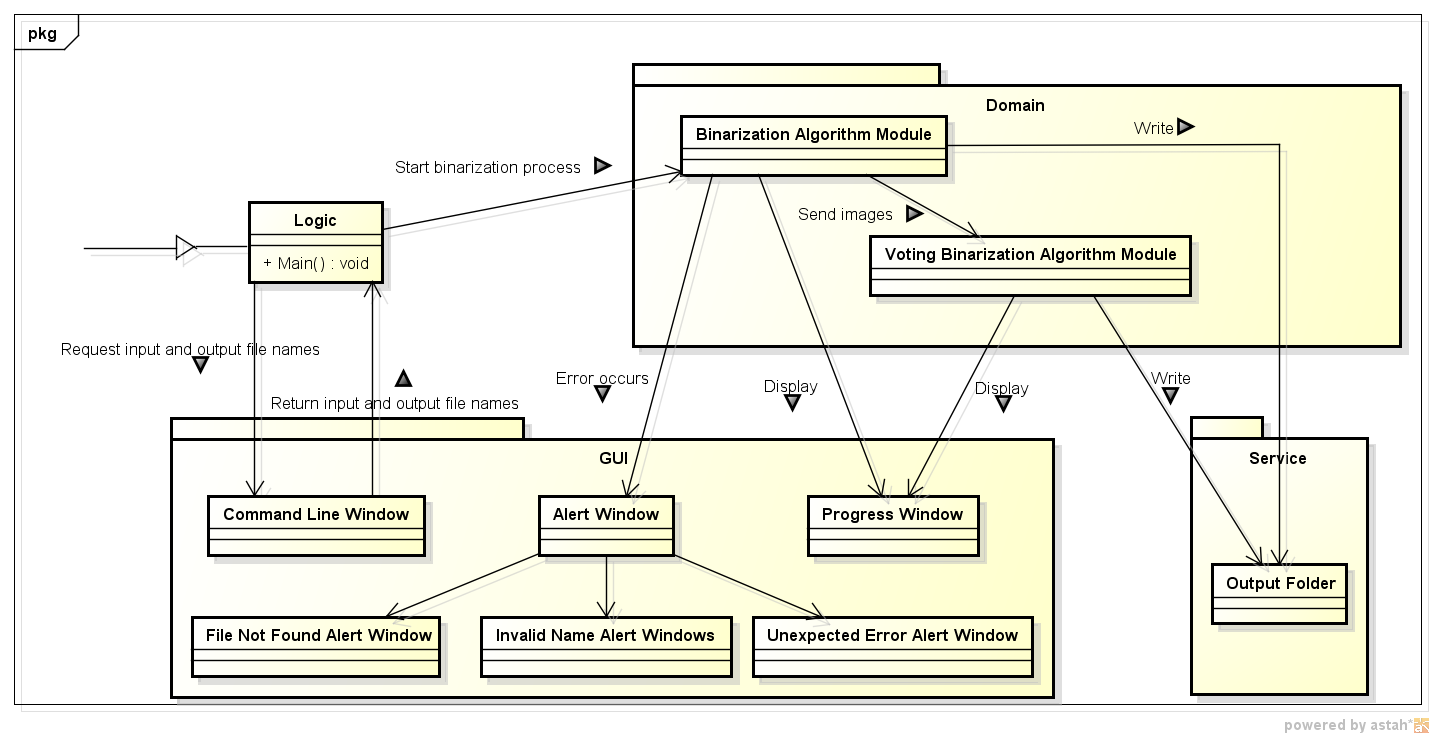
In the domain the algorithms for “Binarization Algorithm Module” (BAM) and for “Voting Binarization Algorithm Module” (VBAM) are implemented.

The service represents just the output folders in which the algorithms in the domain logic should write.

## 5.5 Implementation requirements

To be completed!

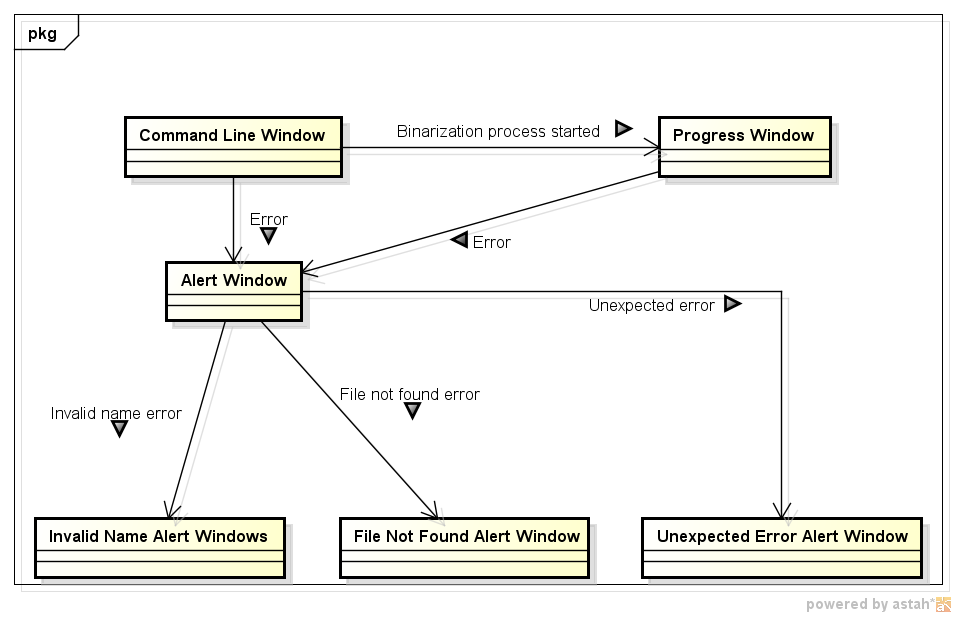
## 5.6 Component interaction



# 6. User interface design

## 6.1 Flow chart

The user interface consists in a command line where the user must input two file names: *input\_image* and an *output\_image*. Also alert windows are displayed in case of errors such as file not found, invalid names given by the user and other unexpected errors. The progress window shows the state of the binarization.



## 6.2 Screen images

To be completed!

# 7. Testing issues

## 7.1 Critical components

To be completed!

## 7.2 Alternatives

To be completed!

