Politehnica University of Bucharest

Software Project Management

Binarization Algorithm Module

Software Design Document

Coordinator:

Mihai Zaharescu

Team:

**Irina Talmaciu, gr 1272E**

**Gabriel Silaghi, gr 1272E**

**Teodora Mohai, gr 1272E**

**Ciulu Razvan, gr 1272E**

SDD

1. Purpose

This document intends to provide a detailed description of a binarization process applied in a software system which is intended to allow the interpretation of a color picture, conversion via a given algorithm and generation of an end result. The desired outcome is a black and white replica of the initial picture.

1. Objectives

This project has included in its checklist the following:

* Research: search for binarization algorithms and techniques and build a data repository for them with characteristics
* Analysis: previously gathered findings are meant to produce a strength / weakness comparison which will be used to decide on one
* Decision: pick an algorithm/method based on previous step and present a valid argumentation
* Build documentation: construct the needed documents for the requirements/ design / test and implementation phases
* Implementation: build the software product corresponding to the previous documents
* Test: internal software product testing
* Presentation or Go-Live: run the software product on given input samples and present outputs

1. Overview

A preliminary phase was completed with the realization of the Software Requirements Document which contains information around research, analysis and decision steps listed in the Objectives section.

In the following parts of this paper we will offer insight on the design phase of the software product. This will be related to data, architecture, user side or graphical interface, testing and project management tasks.

1. Data design

When speaking about data we need to keep in mind several aspects such as: types of data/information, places where it is used, relationships between various fragments/elements of data, structures in which data is organized and flows that will be the paths of particular data elements through the software product body (code).

* 1. Global data structures

In this section we will classify the data structures used throughout the code under their scope and or purpose:

* Basic: INT, FLOAT, STRING, ARRAY, LIST, etc.
* Image related: BufferedImage – image storing and handling format used in java to manipulate an image loaded into memory and perform operations on it;
* File related: BufferedReader – file storing and handling format used in java to manipulate a file loaded into memory and perform operations on it including read, write, update, generate, etc.
* Database related: DB connection – allowing the setup of a connection through sockets and other parameters to a local or network/internet shared database
  + - DB records – allowing access to, or capabilities to retrieve/delete/update information inside the DB
* GUI related: Frame, Buttons, Triggers and Events, Panels
  1. Linking

The data structures will be used in scopes corresponding to their capacity to maximize software productivity. Linking will be maintained error free and within predefined templates throughout the code.

* 1. Temporary

We can name temporary all objects created in memory for use in instance operations such as file retrieval/push, database connection, image processing in memory.

* 1. File formats

Image – we need to

* 1. Database description
  2. Database structure diagram
  3. Table description

1. Architectural design
   1. System architecture
   2. Architectural patterns
   3. Architecture diagram
   4. Description
   5. Implementation requirements
   6. Component interaction (Configuration items)
2. User interface
   1. Flow chart
   2. Screen images
3. Testing issues
   1. Critical components
   2. Alternatives
4. Workdown Break Structure
5. Gantt