Software Requirements Specification Document for project Automated Detection of White Blood Cells Cancer Diseases

Ali Essam, Nermeen Saeed, Rowan Omar, Hend Mohammed
Supervised by Dr. Ashraf abd el raouf, Dr. Taragy and Eng. Nada Ayman

June 22, 2018

1 Introduction

1.1 Purpose of this document

The main purpose of this Software Requirements Specification document is to illustrate our system (Cancer Chaser) requirements that are mainly differentiating, detecting and classifying various white blood cells cancer diseases. Cancer Chaser is considered to be a desktop application that will be built over Python Open CV. We also provide a fulfilled description about each single stage input, output and algorithms used in this stage. Along with a full illustration for each stage requirements and development process. And stating down system constraints, what difficulties have we faced during development and how should we interact with it.

1.2 Scope of this document

This document targets the hematologist experts that would use the Cancer Chaser Application to help them discriminating between different types of white blood cells cancer diseases (Myeloma, Leukemia ALL and AML) which will save much more time rather than manual inspection. It will also be beneficial and helpful for researchers and developers that may work on the Cancer Chaser Application.

1.3 Overview

In Cancer Chaser application we will implement a non-existing system with a high accuracy, speed differentiation and detection of various blood cancer diseases. Cancer Chaser goes through three essential stages. Firstly, uploading microscopic image for blood. Secondly, testing the image after applying different algorithms. Finally, classifying whether the disease is (AML, ALL, Myeloma). The application workflow is illustrated below in fig. 2. Starting with pre-processing which will be performed on the uploaded image. Then, segmentation that converts images to YCBCR color space and constructs Gaussian distribution of CB and CR values. After that is feature extraction, mainly extracting 29 features of 4 types of features Morphologic, statistical, texture and size ratio between nucleus and the cytoplasm. Final step is classification using different algorithms to train and test the dataset. The system stores medical records to save patients medical history. It also has the ability of learning from misclassified tests to enhance the future accuracy of the system.

Our proposed system contains two contributory approaches. The first approach works under the direction of the doctor who test the input image into one of two categories. The first category differentiates between M5 Acute Myeloid Leukemia(AML), L1 and L2 Acute lymphoblastic leukemia (ALL). The second category differentiates between the rest of the sub types. Recursive feature elimination (RFE) algorithm is applied for the sake of picking the crucial features that discriminate between the sub-types in the two main categories.The second approach works automatically without the doctor expectation of the input blood test image. It is a multi stage cascaded architecture of two efficient classifiers Convolutional Neural Network(CNN) and Random Forest(RF). It works based on the notion of rejection. The highest layer is a top level differentiation between the parent disease families(AML,ALL,Myeloma). It then passes the highest two diseases predicted to the lower layers to continue classification between its relative sub-types through a multi stage architecture, Until a final decision is produced automatically in the lowest layer.the performance of CNN and RF is tested to choose final optimization results.The system overview is shown in Fig. 1

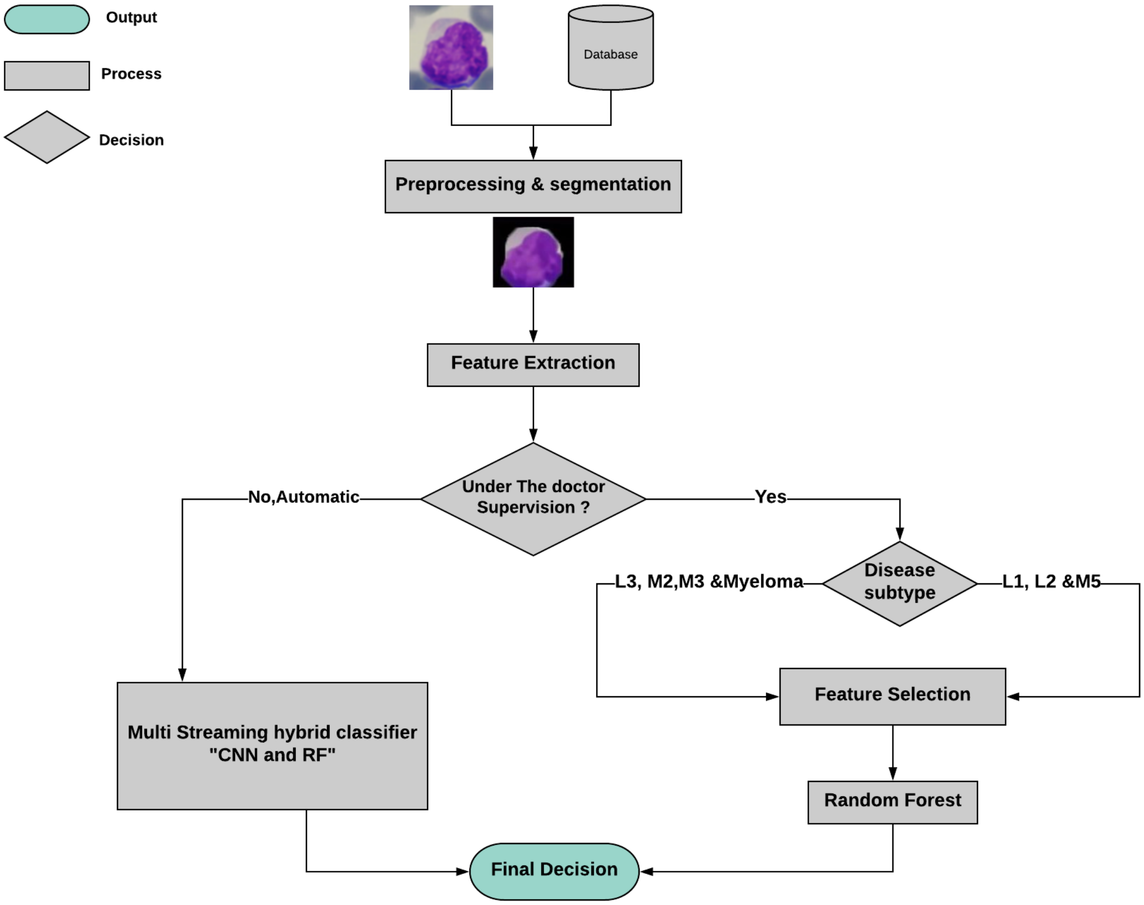


Figure 1: Flowchart of the system Overview

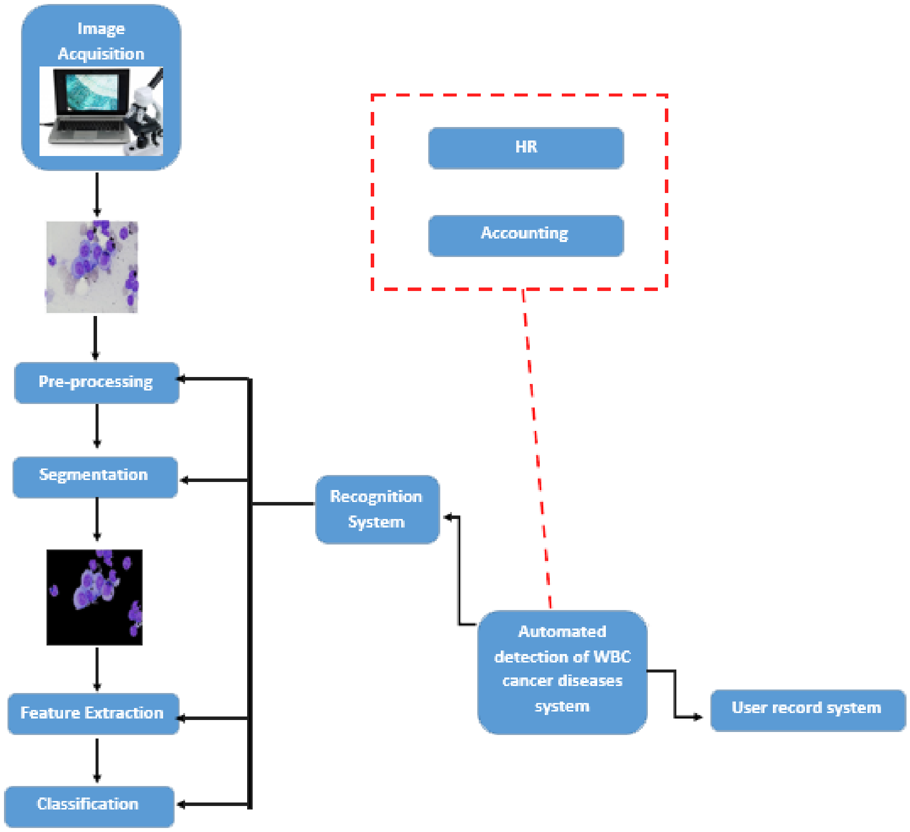


Figure 2: Block diagram of the system Overview

1.4 Business Context

As the process of manually inspecting the microscopic images is costly and time con-summing , there is an urgent need for an automated cancer detection system which saves time for experts and provides more accurate results and costs less than manual inspection when the experts inspecting the microscopic images themselves. Moreover, our proposed automated system is striking to reduce the confusion that may cause misclassification due to similar visual features between sub-types which is a main concern in all nowadays diagnosis.

2 General Description

2.1 Product Functions

Pre-Processing Module
1. Convert to YCBCR color space

2. Convert to grayscale
3. Resizing
Cropping Module
1. Get First Point
2. Get Second Point
3. Perform Cropping
Image Manipulation Module (Pathologists Module)
1. Upload image
Segmentation Module (Application Module)
1. Apply Gaussian distribution and Max Pixel Computing
2. Apply Laplacian Filter
3. Dividing Over Max Pixel
4. Apply Adaptive Threshold
5. Connected Components
6. Remove Small Particles
7. Fetching Segmented Image
Features Extraction Module (Application Module)
1. Extract features
2. Perform texture features
3. Perform statistical features
4. Perform morphological features
5. Perform size ratio features
Classification Module
1. Random forest
2. CNN
Admin Module
1. Sign in/Out.
2. Search for hematologist.
3. Delete test results.
4. Create user account.
5. Delete user account.
6. Update patient Result.
7. Update hematologist info.
8. Feed system
Hematologist Module
1. Sign in/Out.
2. Search for patients info.
3. Update patient info.
4. Delete patient info.
5. Add patient info.
6. Save results.
7. Update result.
8. Delete result.
9. search and view results.
10. Edit profile.
11. Re-Classify.

Encryption Module
1. Encrypt.
2. Decrypt.

2.2 Similar System Information

Flow cytometry is designed based on the idea of characterization and measurements of cells and cellular constituents as they travel in a fluid system. Its composed of 3 main components. First, Fluidics to introduce and focus the cells for interrogation, laser to generate and collect the light signals and Electronics to convert the laser signals to proportional electronic signals (voltage pulses) to use it in computerized analysis. And It works as follows, each cell has a unique natural marker called CD (Cluster Differentiation) on its surface that marks it. Flow cytometry is identifying this CD by putting different antibodies of the CD that is suspended to be on the surface of the cell and the proper antibody for the CD will stick to it and it will cause a fluorescence, then optical detectors will identify which CD is matched for the fluorescent antibody, then the electronics convert this emission of electrons to voltage pulses to use it in computerized analysis.

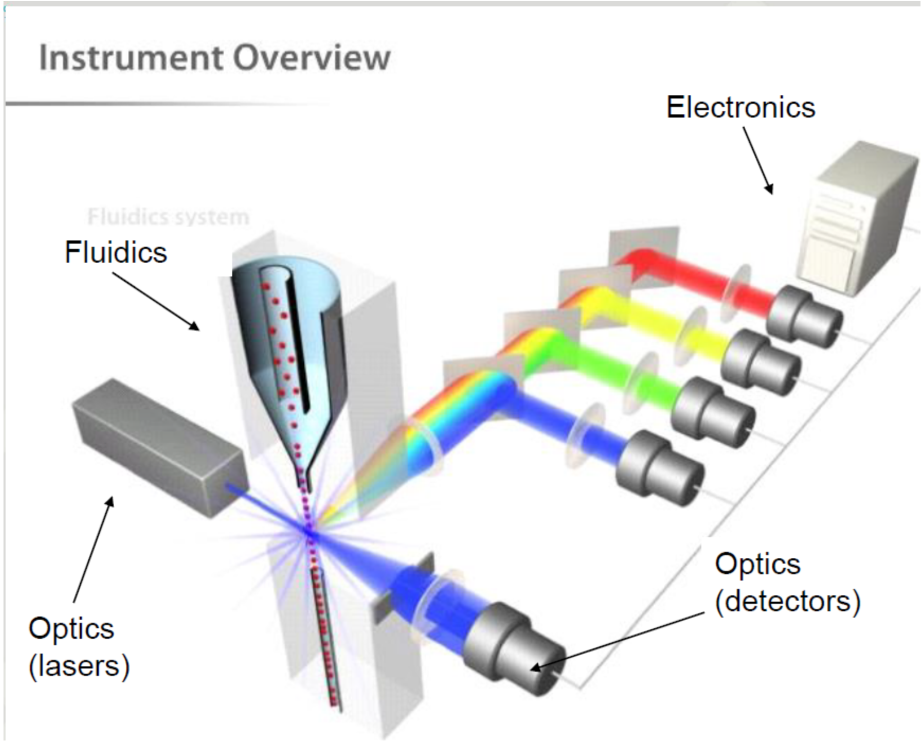


Figure 3: The Entire Functionality of Flow-Cytometry

2.3 User Characteristics

There are 2 types of users that interact with the system: Hematologists and Administrators. Both types of users have different use of the system so each of them has his own requirements.
1. Hematologist
- Must have a basic knowledge in using computers.
- Must have domain knowledge and skills (Hematology).
- Must be able to interact with the UI to add, delete, search and edit records.
2. Administrator
- Must be able to work with databases and to manage the database through UI provided in the system
- Must be able to manage and monitor the database efficiency.
- Must be able to sustain the security and integrity of data.

2.4 User Problem Statement

Hematologists consume a long time and effort in diagnosing blood diseases, moreover, most of diagnostic blood tests are very expensive. Misdiagnosis and medical errors are common nowadays, it is dangerous because it often results in unnecessary, ineffective, and dangerous treatments. The market currently lacks a system which automatically detects blood cancer diseases with high accuracy to reduce the confusion that may cause misclassification due to similar visual features between sub-types to save time, effort and money.

2.5 User Objectives

1. Automated system to accurately detect WBC diseases (Myeloma and Leukemia).
2. Hematologist can view type, sub-type and stage of disease based on the automatic detection.
3. Save the patients results and medical history.

2.6 General Constraints

- Our system is only used for medical purposes in hospitals, clinics and laboratories.
- Our system is applicable for PCs and laptops ONLY.

3 Functional Requirements

3.1 Pre-Processing
3.1.1 Function: Convert to YCBCR
Code: (1).
Input: RGB image.
Action: Check that image is in RGB color space.
Requirement: User must be Hematologist.
Pre-condition: Image is created.
Output: YCBCR image.
Description: Converts RGB image to YCBCR.
Post-condition: Image is converted into YCBCR.
Dependency: None.

3.1.2 Function: Convert to Grayscale
Code: (2).
Input: RGB or YCBCR image.
Action: Check that image is in RGB/YCBCR color space.
Requirement: User must be Hematologist.
Pre-condition: Image is created.
Output: Grayscale image.
Description: Converts RGB/YCBCR image to Grayscale.
Post-condition: Image is converted into Grayscale.

Dependency: None.

3.2 Hematologist
3.2.1 Function: Search for Patient
Code: (3).
Input: Patient ID.
Action: Check that patient exists and is in the set of this Hematologists patients.
Requirement: User must be Hematologist.
Pre-condition: Patient ID is found.
Output: Patient with all information related to him and all test results.
Description: Hematologist will search for a specific patient to get all his data and results.
Post-condition: Patient info is fetched from database and shown.
Dependency: None.

3.2.2 Function: Save test results
Code: (4).
Input: Patient ID.
Action: Check that patient exists.
Requirement: User must be Hematologist.
Pre-condition: Patient ID is found.
Output: Confirmation of successfully adding the results.
Description: Hematologist will search for a specific patient to save his test results including his blood sample image, classification of the disease and its stage.
Post-condition: Data is saved in the database.
Dependency: (3).

3.2.3 Function: View Report
Code: (5).
Input: None.
Action: None.
Requirement: User must be Hematologist.
Pre-condition: None.
Output: The selected report is displayed.
Description: Hematologist will choose one of the available reports.
Post-condition: Data is fetched from the database.
Dependency: None.

3.2.4 Function: Update patient results
Code: (6).
Input: Patient ID.
Action: Check that Patient exists.
Requirement: User must be Hematologist.
Pre-condition: Patient ID is found.
Output: Confirmation that data successfully updated.

Description: Hematologist will search for a search to update his results.
Post-condition: Data is updated in the database.
Dependency: (5).

3.2.5 Function: Sign in
Code: (7).
Input: Username and Password.
Action: Check that Username and Password exist and are correct.
Requirement: User must be Hematologist.
Pre-condition: Username and Password are found.
Output: Hematologist Profile.
Description: Hematologist will enter his username and password in order to sign in.
Post-condition: Hematologist is signed in.
Dependency: (34, 35).

3.2.6 Function: Sign out
Code: (8).
Input: None.
Action: None.
Requirement: User must be Hematologist.
Pre-condition: None.
Output: Logged out confirmation.
Description: Hematologist will sign out of his account.
Post-condition: Hematologist is signed out.
Dependency: (7).

3.2.7 Function: Upload Image
Code: (9).
Input: RGB image.
Action: Check that image is uploaded and validate that it is correct.
Requirement: User must be Hematologist.
Pre-condition: None.
Output: Image is set on the UI.
Description: Hematologist will upload the image which he wants to test.
Post-condition: Image is uploaded.
Dependency: None.

3.2.8 Function: Crop Image
Code: (10).
Input: RGB image.
Action: Check that image is uploaded.
Requirement: User must be Hematologist.
Pre-condition: None.
Output: Image is set on the UI.
Description: Hematologist will crop the cell which he wants to classify.

Post-condition: Image is cropped.
Dependency: (9).

3.2.9 Function: Classify
Code: (11).
Input: RGB image.
Action: Check that image to classify is uploaded.
Requirement: User must be Hematologist.
Pre-condition: Image is found.
Output: Image Result is set on the UI.
Description: Hematologist will click classify in order to start testing.
Post-condition: Image is classified.
Dependency: (9).

3.2.10 Function: Re-classify
Code: (12).
Input: disease label, array of features and segmented image.
Action: Check that user is an expert.
Requirement: User must be an expert Hematologist.
Pre-condition: None.
Output: Re-classification confirmation.
Description: In case of mis-classification, experts can reclassify the misdiagnosed samples manually with their correct labels.
Post-condition: Disease is re-classified and repeated the training process of the whole system after adding the the new labelled sample to database.
Dependency: none.

3.3 Admin
3.3.1 Function: Search for Patient/Hematologist
Code: (12).
Input: Patient/Hematologist ID.
Action: Check that patient/hematologist exists.
Requirement: User must be Admin.
Pre-condition: Patient/Hematologist ID is found.
Output: Patient/Hematologists data with all his data.
Description: Admin will search for a specific patient/hematologist to show his test results including his blood sample image, classification of the disease and its stage/info.
Post-condition: Data is fetched from the database and shown.
Dependency: None.

3.3.2 Function: Delete test results
Code: (13).
Input: Patient ID.

Action: Check that patient exists.
Requirement: User must be Admin.
Pre-condition: Patient ID is found.
Output: Confirmation of successfully deleting the results.
Description: Admin will search for a specific patient to delete his test results including his blood sample image, classification of the disease and its stage.
Post-condition: Data is persisted in the database.
Dependency: (12).

3.3.3 Function: Create user account
Code: (14).
Input: Username and Password.
Action: Check that username is unique.
Requirement: User must be Admin.
Pre-condition: Username doesnt previously exist.
Output: Confirmation of successfully adding the account.
Description: Admin will add a new hematologist.
Post-condition: User is added in the database with its type as hematologist.
Dependency: None.

3.3.4 Function: Delete user account
Code: (15).
Input: Patient/Hematologist ID.
Action: Check that user exists.
Requirement: User must be Admin.
Pre-condition: User ID is found.
Output: Confirmation of successfully deleting the account.
Description: Admin will delete a user whether a hematologist or a patient.
Post-condition: User is deleted from the database.
Dependency: (12).

3.3.5 Function: Update patient results
Code: (16).
Input: Patient ID.
Action: Check that user exists.
Requirement: User must be Admin.
Pre-condition: Patient ID is found.
Output: Confirmation of successfully updating the results.
Description: Admin will update patients info.
Post-condition: User is updated in the database.
Dependency: (12).

3.3.6 Function: Update Hematologist info
Code: (17).
Input: Hematologist ID.
Action: Check that hematologist exists.

Requirement: User must be Admin.
Pre-condition: Hematologist ID is found.
Output: Confirmation of successfully updating account.
Description: Admin will update hematologists info.
Post-condition: Data is updated in the database.
Dependency: (12).

3.3.7 Function: Sign in
Code: (18).
Input: Username and Password.
Action: Check that Username and Password exist and are correct.
Requirement: User must be Admin.
Pre-condition: Username and Password are found.
Output: Admin Panel.
Description: Admin will enter his username and password in order to sign in.
Post-condition: Admin is signed in.
Dependency: (34, 35).

3.3.8 Function: Sign out
Code: (19).
Input: None.
Action: None.
Requirement: User must be Admin.
Pre-condition: None.
Output: Logged out confirmation.
Description: Admin will sign out of his account.
Post-condition: Admin is signed out.
Dependency: (18).

3.3.9 Function: Feed system
Code: (20).
Input: Features operations object and authentication password.
Action: check authentication password.
Requirement: User must be Admin.
Pre-condition: admins authentication password is authenticated.
Output: Confirmation of successfully feeding the system.
.
Description: Admin users have an option to feed the system with more training labelled blood samples.
Post-condition: Data is updated in the database.
Dependency: none.

3.4 Segmentation
3.4.1 Function: Perform Gaussian Distribution and Max Pixel Computing

Code: (21).
Input: YCBCR image.
Action: None.
Requirement: Performed by system.
Pre-condition: None.
Output: Max pixel computed.
Description: Gaussian distribution and max pixel are extracted from the image.
Post-condition: Max pixel computed.
Dependency: (1).

3.4.2 Function: Perform Laplacian Filter
Code: (22).
Input: Float image.
Action: None.
Requirement: Performed by system.
Pre-condition: None.
Output: New Filtered image.
Description: Image is filtered using Laplacian.
Post-condition: New Filtered image.
Dependency: None .

3.4.3 Function: Normalize
Code: (23).
Input: Float image and Max pixel.
Action: Check that max pixel not null.
Requirement: Performed by system.
Pre-condition: Max pixel has a value.
Output: New image with new pixel values.
Description: Divide image pixels over max pixel calculated by Gaussian distribution and filtered by Laplacian.
Post-condition: Image pixels are replaced with the new ones.
Dependency: (21) and (22).

3.4.4 Function: Perform Adaptive Threshold
Code: (24).
Input: Original image and Float image.
Action: None.
Requirement: Performed by system.
Pre-condition: None.
Output: New image after passing through threshold value.
Description: Adaptive Threshold is performed on the float image and any pixels that are less than the value determined are set in the new image as the original one.
Post-condition: A new image is created with the new pixels.
Dependency: None.

3.4.5 Function: Get Connected Components
Code: (25).
Input: Threshold Binary image.
Action: None.
Requirement: Performed by system.
Pre-condition: None.
Output: Array of connected components in the image.
Description: Get connected components in the image to know the positions and sizes of the cells.
Post-condition: Array containing all the components with their info like size.
Dependency: (2).

3.4.6 Function: Apply Morphological Operations
Code: (26).
Input: Image returned from Remove Small Particles Function and the original image.
Action: Check that given images are not null.
Requirement: Performed by system.
Pre-condition: Images given are found.
Output: Cells are back to the original image with the small particles.
Description: Returns the image to its original state after segmenting the cell and removing noise.
Post-condition: New image with the segmented cells.
Dependency: (25).

3.5 Feature Extraction

3.5.1 Function: Extract Texture Features
Code: (27).
Input: Grayscale image.
Action: Check that the input image type is grayscale.
Requirement: performed by the system.
Pre-condition: Grayscale image given is not null.
Output: Returns array of 14 texture features.
Description: Calculates Haralick texture features for four types of adjacency by finding the mean of all four types of GLCM and taking the mean of it and return it.
Post-condition: Haralick textures features have been extracted from the input image.
Dependency: (2).

3.5.2 Function: Extract statistical Features
Code: (28).
Input: RGB image.
Action: Check that the input image is not null.
Requirement: performed by the system.

Pre-condition: none.
Output: Array of statistical features (mode, mean, Standard deviation, variance, sum and gradient).
Description: Extracts features that concerns cell shape information.
Post-condition: Statistical features have been extracted.
Dependency: (1, 2).

3.5.3 Function: Extract Morphological Features

Code: (29).
Input: RGB image.
Action: Check that the input image is not null.
Requirement: performed by the system.
Pre-condition: none.
Output: Array of morphological features (Area to perimeter ratio, circularity, elongation, major to minor axis length ratio, extent and solidity).
Description: Extracts features that represents the shape of the cell and its dimensions.
Post-condition: Morphological features have been extracted.
Dependency: (1).

3.5.4 Function: Extract Size ratio Features

Code: (30).
Input: RGB image.
Action: Check that the input image is not null.
Requirement: performed by the system.
Pre-condition: none.
Output: Array of size ratio features (Nucleus cytoplasm area, nucleus cell area and nucleus cell perimeter).
Description: Extracts some features such as nucleus cytoplasm area, nucleus cell area and nucleus cell perimeter.
Post-condition: size ratio features have been extracted.
Dependency: (1).

3.7 Classification

3.7.1 Function: Train data

Code: (31).
Input: none.
Action: Check that the array of features is not null.
Requirement: performed by the system.
Pre-condition: The array of features is not null.
Output: none.
Description: Trains the classifier based on the array of features.
Post-condition: Classifier has trained the data.

Dependency: none.

3.7.2 Function: Fit data
Code: (32).
Input: array of features, array of labels and classifier created object.
Action: Check that the arrays of features, array of labels and the classifier object are not null.
Requirement: Performed by the system.
Pre-condition: The arrays of features, array of labels and the classifier object are not null.
Output: Returns the classifier object.
Description: Fits the training data and labels to the classifier.
Post-condition: Training data has been fitted to the classifier.
Dependency: (31).

3.7.3 Function: Classify
Code: (33).
Input: The Features, Which Features, Model Name and Chosen Labels.
Action: Check that inputs are not null.
Requirement: Performed by the system.
Pre-condition: The inputs are not null.
Output: Pop up message to inform the hematologist the detected disease.
Description: Evaluates the model and predicts label to classify the disease.
Post-condition: The disease has been classified.
Dependency: (32).

3.8 Security
3.8.1 Function: Encrypt
Code: (34).
Input: key and plain text.
Action: Check that the plain text is not null.
Requirement: Performed by the system.
Pre-condition: Plain text is not null.
Output: Returns Cipher text.
Description: Encode sensitive information with special characters representing typed characters.
Post-condition: Plain text is encrypted and only authorized parties can access it.
Dependency: none.

3.8.2 Function: Decrypt
Code: (35).
Input: key and cipher text.
Action: Check that the cipher text is not null.
Requirement: Performed by the system.

Pre-condition: Cipher text is not null.
Output: Returns Plain text.
Description: Takes encoded or encrypted text or other data and converting it back into the original format.
Post-condition: Cipher text is decrypted.
Dependency: (34).

4 Interface Requirements

4.1 User Interfaces

Our system Cancer Chaser user interface is very usable and clear. You can login whether you are an admin or a doctor. The system navigates you to different screens depends on your type. System admins have some duties such as manipulating doctors. While doctors are responsible for dealing with their patients.

4.1.1 GUI

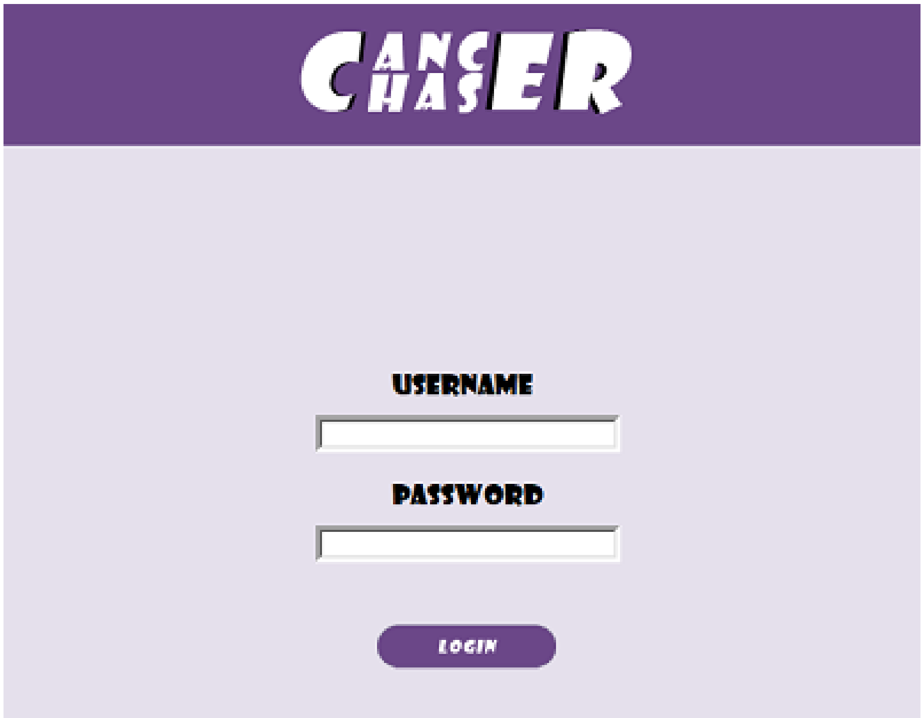
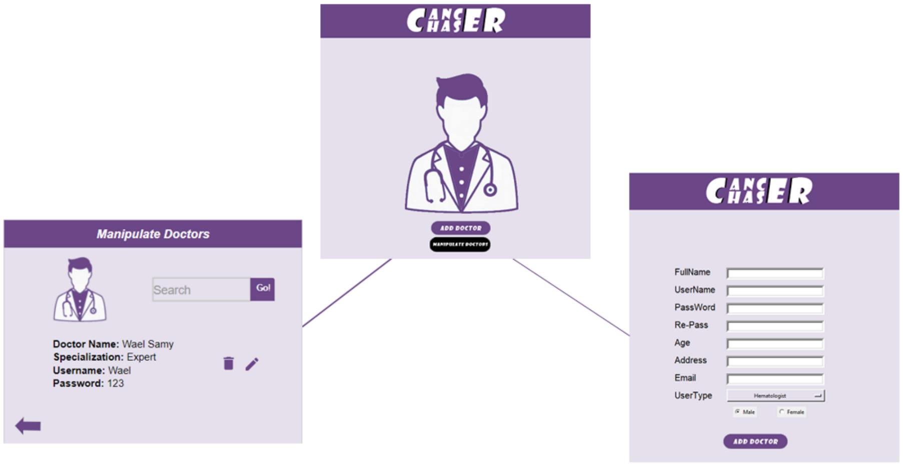


Figure 4: Login Screen

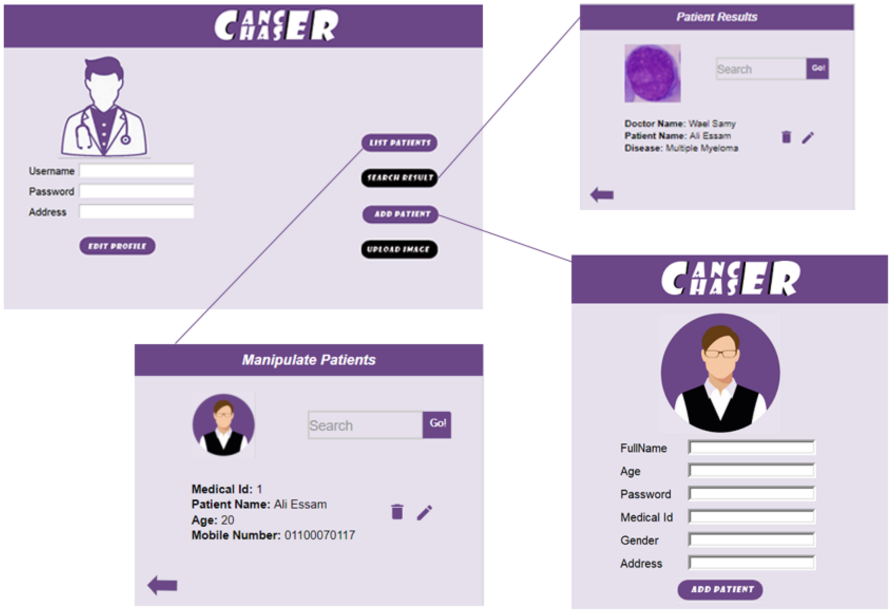


Doctor Name: Wael Samy
Specialization: Expert
Username: Wael
Password: 123

UserType Hematologist

● Male ○ Female

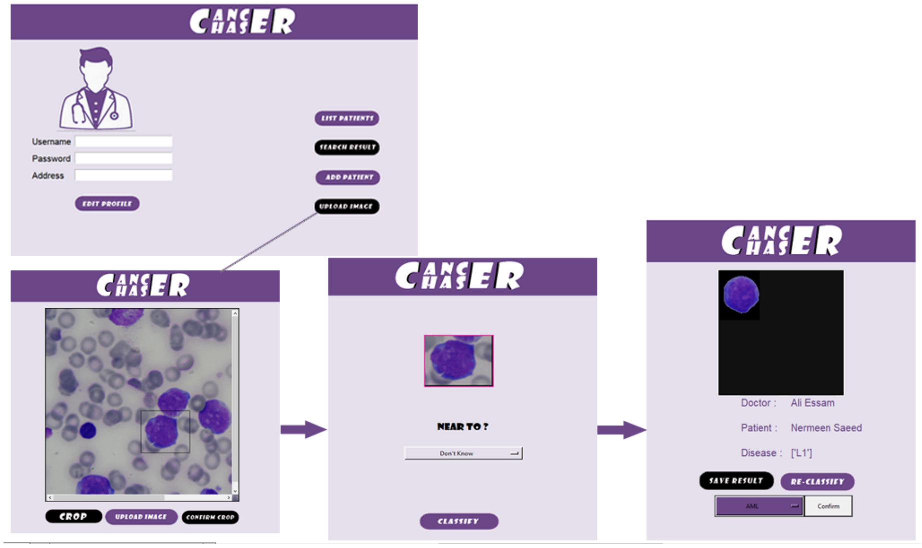
Figure 5: Admin Screen



Doctor Name: Wael Samy
Patient Name: Ali Essam
Disease: Multiple Myeloma

Medical Id: 1
Patient Name: Ali Essam
Age: 20
Mobile Number: 01100070117

Figure 6: Hematologist Screen 1



NEAR TO ? Don't Know

Doctor : Ali Essam

Patient : Nermeen Saeed

Disease : ['L1']

AML

Figure 7: Hematologist Screen 2

4.1.2 CLI

N/A

4.1.3 API

1. Mahotas
2. Scikit-learn
3. OpenCV
4. Numpy
5. Theano
6. Keras
7. Tkinter
8. Pymysql

4.1.4 Diagnostics or ROM

N/A

4.2 Hardware Interfaces

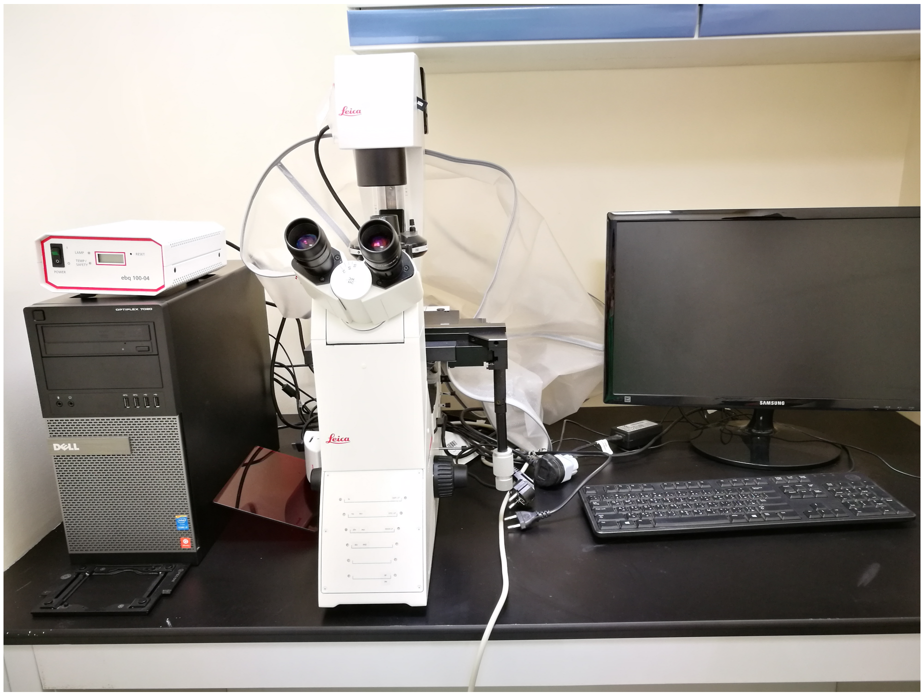


Figure 8: To watch the sample

4.3 Communications Interfaces

N/A

4.4 Software Interfaces

1. Monitor Calibration Wizard that connects the digital microscope camera to the PC.

5 Performance Requirements

High Recognition Accuracy

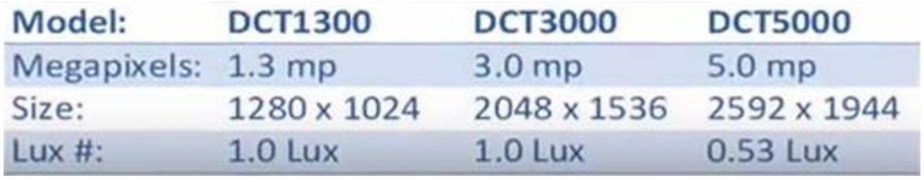
6 Design Constraints

6.1 Standards Compliance

1. 64-bit operating system, x64 based processor.
2. PC with 4.00 GB of RAM. (Minimum).

6.2 Hardware Limitations

Digital microscope with a digital camera.
Specifications :



Model:	DCT1300	DCT3000	DCT5000
Megapixels:	1.3 mp	3.0 mp	5.0 mp
Size:	1280 x 1024	2048 x 1536	2592 x 1944
Lux #:	1.0 Lux	1.0 Lux	0.53 Lux

Figure 9: Microscope Specifications

6.3 others as appropriate

7 Other non-functional attributes

7.1 Security TAG: Encryption and Decryption GIST: The security of protecting sensitive data for users of the system. SCALE: The back-end servers shall never display sensitive data such as passwords and patients data. It should always be encrypted with special characters representing typed characters. MUST: 100percent of the time. 7.2 Maintainability TAG: Ease of system maintainability GIST: The system should be easy to extend. The code should be written in a way that it favors implementation of new functions so the system ensures ease of maintainability through the implementation of MVC, strategy, decorative and observer design patterns. SCALE: Future functions will be implemented easily to the system. MUST: 100percent of the time.

7.3 Scalability TAG: Scalability of the system GIST: The system should be highly scalable as we have growing data of patients and tests results. SCALE: The system will work well on a small set of data as well as working well on a large set of data. MUST: 100percent of the time.

7.4 Extensibility TAG: Ease of system extensibility GIST: The system should be easy to extend. Extensions can be through the addition of new types, properties, attributes or through modification of existing functionality. The system ensures extensibility by applying EAV data model. SCALE: Changes are possible in the system with minimum impact to the system structure. MUST: 100percent of the time.

8 Preliminary Object-Oriented Domain Analysis

This section presents a list of the fundamental objects that must be modeled within the system to satisfy its requirements. The purpose is to provide an alternative, ”structural” view on the requirements stated above and how they might be satisfied in the system. A primitive class diagram to be delivered.

8.1 Inheritance Relationships

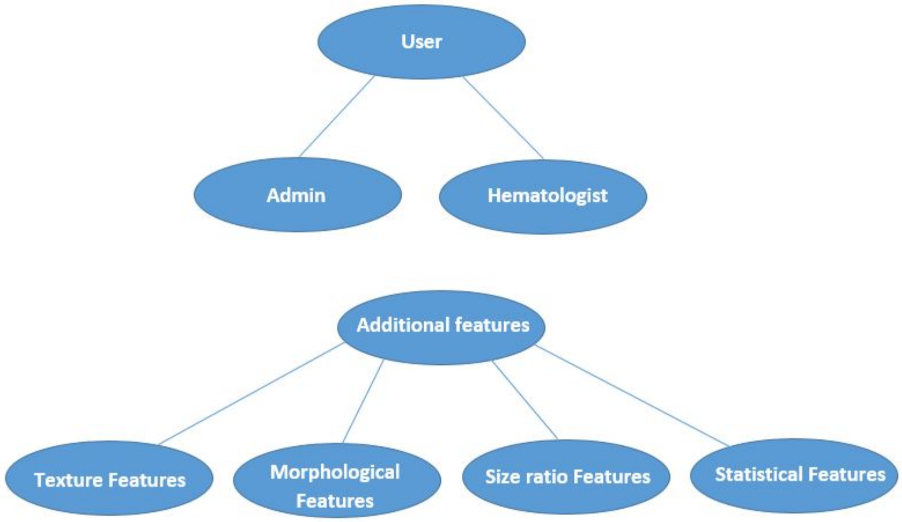


Figure 10: Inheritance Relationships

8.2 Class descriptions

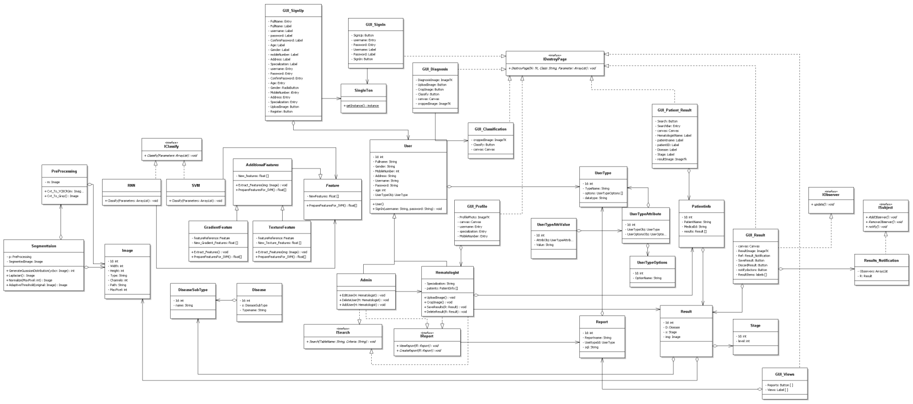


Figure 11: Class Diagram

8.2.1 Class name: User
Type: Concrete.
List of super classes: N/A.
List of sub classes: Hematologist, Admin.
Purpose: Class to encapsulate different user-types with their common attributes.
Collaboration: This class aggregates class usertype, aggregated by class GUI-SignUp, Inherited By class Admin and Hematologist.
Attributes: Id, Fullname, username, password ,Gender, Mobile Number, Address, age, usertype object.
Operations: SignIn(username, password).

8.2.2 Class name: UserType
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: To allow scalability to add new user types.
Collaboration: This class is aggregated by class User, UserTypeAttribute and Report.
Attributes: id, TypeName, options[ ].
Operations: None.

8.2.3 Class name: UserTypeOptions
Type: Concrete.
List of super classes: N/A.
List of sub classes: None.
Purpose: Class to contain possible options for all user types.
Collaboration: This class is aggregated by class UserTypeAttribute.

Attributes: Id, optionname, datatype.
Operations: None.

8.2.4 Class name: UserTypeAttribute
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: To represent what options each usertype has.
Collaboration: This class aggregates UserType, UserTypeOptions, aggregated by class UserTypeAttrValue.
Attributes: Id, UserType object, UserTypeOptions object.
Operations: None.

8.2.5 Class name: UserTypeAttrValue
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: To contains the actual values of user-type options attributes.
Collaboration: This class aggregates class UserTypeAttribute.
Attributes: id, UserTypeAttribute Object, value.
Operations: None

8.2.6 Class name: Admin
Type: concrete.
List of super classes: User.
List of sub classes: None.
Purpose: To represent the admin.
Collaboration: This class extends class User, assisted by class Hematologist, implements Interfaces (ISearch and IReport).
Attributes: Id.
Operations: AddUser(Hematologist H), DeleteUser(Hematologist H), EditUser(Hematologist H).

8.2.7 Class name: Hematologist.
Type: concrete.
List of super classes: User.
List of sub classes: None.
Purpose: To represent the Hematologist Expert.
Collaboration: This class extends class User, associates class Admin, implements Interfaces (ISearch and IReport), aggregated by classes GUI-Profile, aggregates class Patientinfo, assisted by class Result.
Attributes: Id ,Specialization, Results[ ].
Operations: UploadImage( ), CropImage( ),SaveResults(Result R), DeleteResult (Result R).

8.2.8 Class name: ISearch

Type: Interface.
List of super classes: None.
List of sub classes: None.
Purpose: To allow searching with different Criterias.
Collaboration: classes(Admin, Hematologist) implements this class.
Attributes: None.
Operations: Search (TableName, Criteria).

8.2.9 Class name: IReport
Type: Interface.
List of super classes: None.
List of sub classes: None.
Purpose: To allow creating and viewing reports.
Collaboration: classes (Admin, Hematologist) implements this class, this class is assisted by class Report.
Attributes: None.
Operations: CreateReport (Report R), ViewReport (Report R).

8.2.10 Class name: Feature
Type: Abstract.
List of super classes: None.
List of sub classes: class AdditionalFeatures.
Purpose: To contain the basic features of the Image to be wrapped with additional features layers.
Collaboration: This class is inherited by class AdditionalFeatures.
Attributes: Id ,Newfeatures [ ].
Operations: PrepareFeaturesFor-SVM().

8.2.11 Class name: AdditionalFeatures
Type: concrete.
List of super classes: Feature.
List of sub classes: class TextureFeature, GradientFeature.
Purpose: to append to the basic features.
Collaboration: This class is inherited by class TextureFeature and GradientFeature, inherits class Feature.
Attributes: Id ,Newfeatures [ ].
Operations: Exctract-Features(Image im), PrepareFeaturesFor-SVM().

8.2.12 Class name: TextureFeature.
Type: concrete.
List of super classes: AdditionalFeatures.
List of sub classes: None.
Purpose: To extract the texture features of the Image.
Collaboration: This class inherits class AdditionalFeatures.
Attributes: Id , New-Texture-Features [ ].
Operations: Exctract-Features(), PrepareFeaturesFor-SVM().

8.2.13 Class name: GradientFeature.
Type: concrete.
List of super classes: AdditionalFeatures.
List of sub classes: None.
Purpose: extract the Gradient features of the Image.
Collaboration: This class inherits class AdditionalFeatures.
Attributes: Id ,New-Gradient-Features [ ].
Operations: Exctract-Features(), PrepareFeaturesFor-SVM().

8.2.14 Class name: IClassify
Type: Interface.
List of super classes: None.
List of sub classes: None.
Purpose: To allow classification with different classifiers strategies.
Collaboration: Class SVM and RNN implements this class.
Attributes: None.
Operations: Classify(Parameters [ ]).

8.2.15 Class name: SVM
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: To allow classification with svm classifier.
Collaboration: This class implements class IClassify, assisted by class Features.
Attributes: None.
Operations: Classify (Parameters [ ]).

8.2.16 Class name: RNN
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: To allow classification with RNN classifier.
Collaboration: This class implements class IClassify, assisted by class Features.
Attributes: None.
Operations: Classify(Parameters [ ]).

8.2.17 Class name: Image
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: To contain the image.
Collaboration: This class is aggregated by classes(preprocessing ,segmentation, Result).
Attributes: Id , Width, Height, Type, Channels, Path, MaxPixel.
Operations: None.

8.2.18 Class name: Preprocessing
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: To convert the color space and do some processing on the image before segmentation.
Collaboration: This class aggregates class Image, aggregated by segmentation class.
Attributes: Id ,Image.
Operations: Cvt-To-YCBCR(), Cvt-To-GRAY().

8.2.19 Class name: Segmentation
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: To remark the parts of the doctors interest in the image (cells).
Collaboration: This class aggregates class Image, preprocessing.
Attributes: Id, SegmentedIamgeObject, PreprocessingObject.
Operations: GenerateGuassianDistribution(YcbcrImage),Laplacian(), Normalize(), AdaptiveThreshold().

8.2.20 Class name: Disease
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to identify different diseases.
Collaboration: This class aggregates class DiseaseSubType.
Attributes: Id, DiseaseSubTypeObject, Typename .
Operations: None.

8.2.21 Class name: DiseaseSubType
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to identify different subtypes of diseases.
Collaboration: This class is aggregated by class Disease, Result.
Attributes: Id, name.
Operations: None.

8.2.22 Class name: Result
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to represent the final result doctors received from the classification.
Collaboration: This class aggregates classes DiseaseSubType,Image.

Attributes: Id, DiseaseSubTypeObject, StageObject, Image.
Operations: None.

8.2.23 Class name: GUI-Views
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to represent the screen that allow users to view reports.
Collaboration: This class aggregates class Report, implements Interface IDestroy.
Attributes: Buttons[ ], Label [ ].
Operations: None.

8.2.24 Class name: GUI-Result
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to represent the screen that shows the final result of the diagnosis.
Collaboration: This class aggregates class Result, implements Interface IDestroy.
Attributes: canvas, Image, SaveResultButton, DiscardResultButton, label [].
Operations: None.
8.2.25 Class name: GUI-PatientResult
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to represent the screen that shows the Result of a specific Patient diagnosis.
Collaboration: This class aggregates class patientinfo, implements Interface IDestroy.
Attributes: SearchButton, SearchBar, canvas, label [], Image.
Operations: None.

8.2.26 Class name: GUI-Profile
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to represent the user profile screen.
Collaboration: This class aggregates class Hematologist, implements Interface IDestroy.
Attributes: canvas, Image, SaveResultButton, DiscardResultButton, label [].
Operations: None.

8.2.27 Class name: GUI-Classification
Type: concrete.
List of super classes: None.

List of sub classes: None.
Purpose: to represent the user profile screen.
Collaboration: This class associates class GUI-Diagnosis, implements Interface IDestroy.
Attributes: canvas, Image, ClassifyButton, canvas.
Operations: None.

8.2.28 Class name: GUI-Diagnosis
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to represent the user profile screen.
Collaboration: This class assisted by class GUI-Classification, implements Interface IDestroy.
Attributes: canvas, Image, ClassifyButton, canvas,CropImagebutton.
Operations: None.

8.2.29 Class name: GUI-SignIn
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to represent the sign in screen.
Collaboration: This class implements Interface IDestroy.
Attributes: username, password, SignupButton,SignupButton.
Operations: None.

8.2.30 Class name: GUI-SignUp
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to allow adding a new user.
Collaboration: This class implements Interface IDestroy, aggregates class user.
Attributes: fullname, username, password, age, Gender, mobileNumber, Address, specialization.
Operations: None.
8.2.31 Class name: IDestroy
Type: Interface.
List of super classes: None.
List of sub classes: None.
Purpose: to switch between different screens.
Collaboration: All GUI classes implement this interface.
Attributes: None.
Operations:DestroyPage(Tk, class,parameters[ ]).

8.2.32 Class name: PatientInfo
Type: concrete.

List of super classes: None.
List of sub classes: None.
Purpose: to contain personal information about patients.
Collaboration: this class aggregates class Result, aggregated by class GUI-patientResult,Hematologist.
Attributes: Id, patientname, MedicalId, Results [ ].
Operations: None.

8.2.33 Class name: Report
Type: concrete.
List of super classes: None.
List of sub classes: None.
Purpose: to allow multiple reports in the system.
Collaboration: this class aggregates class userType, aggregated by class GUI-views, associates class IReport.
Attributes: Id, reportname, usertypeiD, SQL.
Operations: None.

9 Operational Scenarios

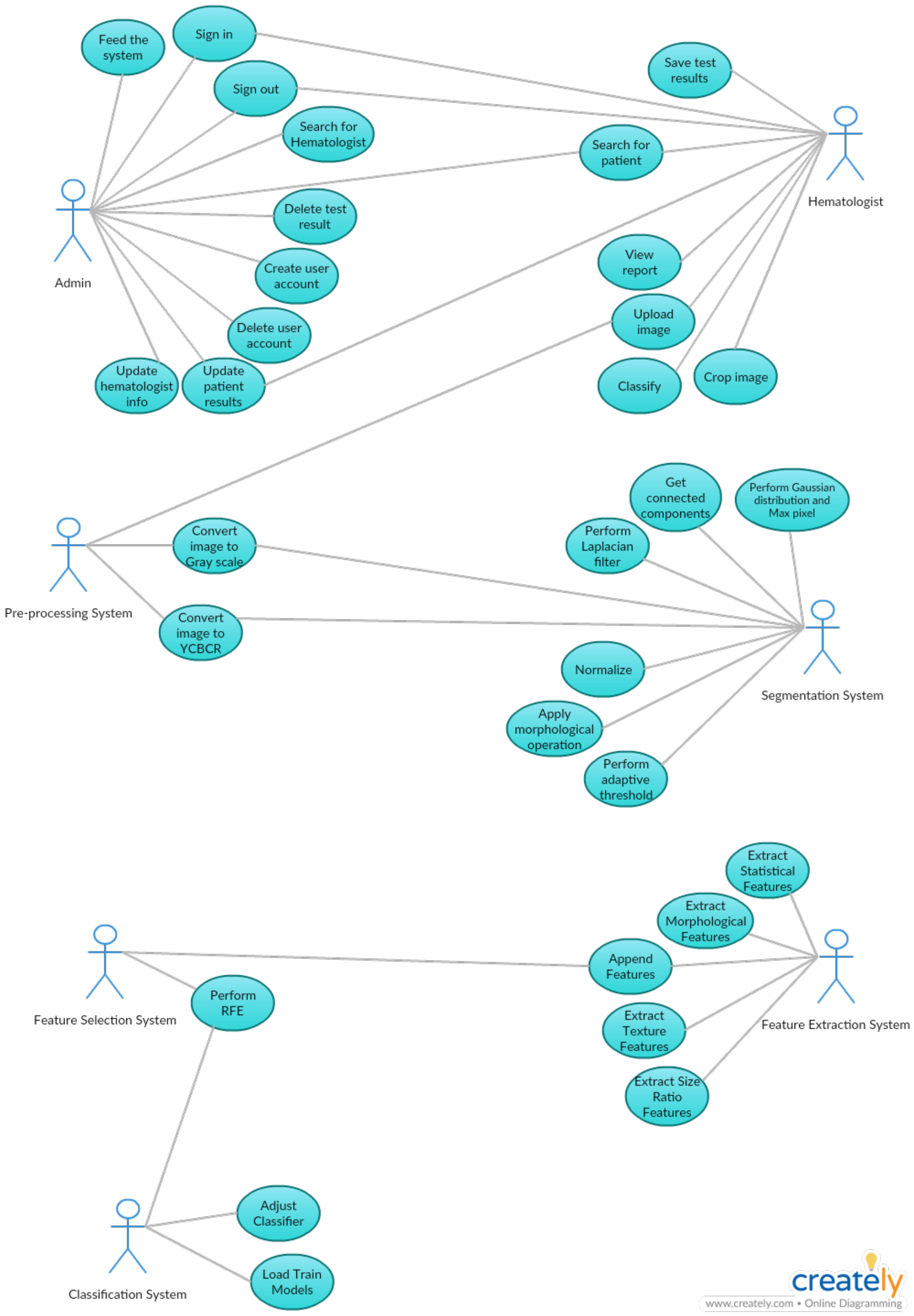


Figure 12: UseCase Diagram

Tabular description of the Upload Image use case
Actors Hematologist.
Description Hematologist will upload the test image to the system.
Data Image of the test sample.
Stimulus User command issued by Hematologist.
Response Image uploaded on the system.
Comments Hematologist must be signed in.

Tabular description of the Crop Image use case
Actors Hematologist.
Description Hematologist will be able to crop a cell in the image in order to be classified.
Data Image of the test sample.
Stimulus User command issued by Hematologist.
Response The selected cell is cropped.
Comments Hematologist must be signed in.

Tabular description of the Save test results use case
Actors Hematologist.
Description Hematologist will be able to save the test result along with the patient data. Also, experts will be able to confirm the result or edit it.
Data Patients test results.
Stimulus User command issued by Hematologist.
Response Confirmation that Data has been saved.
Comments Hematologist must be signed in.

Tabular description of the Classify use case
Actors Hematologist.
Description Hematologist will be able to choose between three different paths for the image to be classified.
Data Test Sample Image.
Stimulus User command issued by Hematologist.
Response The type of the disease.
Comments Hematologist must be signed in.

Tabular description of the Search for patient use case
Actors Hematologist/Admin.
Description Hematologist/Admin can search for test results of a patient that are saved on the system.
Data Patients name or national id.
Stimulus User command issued by Hematologist/Admin.
Response Patients test results displayed.
Comments Hematologist/Admin must be logged in.

Tabular description of the View Report use case
Actors Hematologist.

Description Hematologist can choose from a set of reports to view.
Data None.
Stimulus User command issued by Hematologist.
Response Patients test results displayed.
Comments Hematologist must be logged in.

Tabular description of the Update Patient Results use case
Actors Hematologist/Admin.
Description Hematologist/Admin can update patients test results.
Data Patient ID.
Stimulus User command issued by Hematologist/Admin.
Response Confirmation that data has been updated.
Comments Hematologist/Admin must be logged in.

Tabular description of the Search for Hematologist use case
Actors Admin.
Description Admin can search for a Hematologist.
Data Hematologist ID.
Stimulus User command issued by Admin.
Response Hematologist profile is displayed.
Comments Admin must be logged in.

Tabular description of the Delete test results use case
Actors Admin.
Description Admin will be able to delete patients test result.
Data Patients ID or national id.
Stimulus User command issued by Admin.
Response Confirmation that results have been deleted.
Comments Admin must be logged in and patient must exist on the system.

Tabular description of the Create user account use case
Actors Admin.
Description Admin will be able to create accounts for new Hematologists.
Data Hematologists info (Name, National Id, etc.
).
Stimulus User command issued by Admin.
Response Confirmation that account has been created.
Comments Admin must be signed in.

Tabular description of the Delete user account use case
Actors Admin.
Description Admin will be able to delete Hematologist accounts.
Data Hematologists name and id.
Stimulus User command issued by Admin.
Response Confirmation that account has been deleted.
Comments Admin must be signed in.

Tabular description of the Update Hematologist info use case Actors Admin.
Description Admin will be able to update hematologists data.
Data Hematologist ID.
Stimulus User command issued by Admin.
Response Hematologists Profile displayed for editing.
Comments Admin must be signed in.

Tabular description of the Convert image to grayscale use case
Actors Pre-processing System.
Description Test image is converted from RGB to Grayscale.
Data RGB Test Image.
Stimulus System command.
Response Image is converted.
Comments None.

Tabular description of the Convert image to YCBCR use case
Actors Pre-processing System.
Description Test image is converted from RGB to YCBCR.
Data RGB Test Image.
Stimulus System command.
Response Image is converted.
Comments None.

Tabular description of the Extract Statistical Features use case
Actors Feature Extraction System.
Description Statistical features like mean, mode and standard deviation are calculated per image.
Data Segmented image.
Stimulus System command.
Response Array of statistical features.
Comments Segmented image must be converted to YCBCR.

Tabular description of the Extract Morphological Features use case
Actors Feature Extraction System.
Description Morphological features like area, perimeter, extent and circularity are calculated per image.
Data Segmented image.
Stimulus System command.
Response Array of morphological features.
Comments Segmented image must be converted to YCBCR.

Tabular description of the Extract Texture Features use case
Actors Feature Extraction System.
Description Texture features like contrast, correlation, variance and entropy are calculated per image.

Data Segmented image.
Stimulus System command.
Response Array of Texture features.
Comments Segmented image must be converted to YCBCR.

Tabular description of the Extract Size Ratio Features use case
Actors Feature Extraction System.
Description Size Ratio features like Nucleus cell area, Nucleus cell perimeter and Nucleus Cytoplasm area are calculated per image.
Data Segmented image.
Stimulus System command.
Response Array of size ratio features.
Comments Segmented image must be converted to YCBCR.

Tabular description of the Append Features use case
Actors Feature Extraction System.
Description Extracted features are appended.
Data Arrays of features.
Stimulus System command.
Response Array of all extracted features.
Comments None.

Tabular description of the Perform RFE use case
Actors Feature Selection System.
Description RFE is performed on array of features to select the best features for each image.
Data Appended array of features.
Stimulus System command.
Response Array of the selected features.
Comments None.

Tabular description of the Adjust Classifier use case
Actors Classification System.
Description The classifier is created based on the selected approach.
Data Classifier Type.
Stimulus System command.
Response Classifier object created.
Comments None.

Tabular description of the Load Train Models use case
Actors Classification System.
Description Train models are loaded based on the selected approach.
Data Train models.
Stimulus System command.
Response The models are loaded.
Comments None.

Tabular description of the Perform Gaussian Distribution and Max Pixel use case
Actors Segmentation System.
Description Extract max pixel from image.
Data Image to segment.
Stimulus System command.
Response Max pixel computed.
Comments None.

Tabular description of the Perform Laplacian Filter use case
Actors Segmentation System.
Description Apply Laplacian filter.
Data Image to segment.
Stimulus System command.
Response New filtered image.
Comments None.

Tabular description of the Normalize use case
Actors Segmentation System.
Description Divide image pixels over max pixel.
Data Max pixel to divide over it.
Stimulus System command.
Response New image with new pixel values.
Comments None.

Tabular description of the Perform Adaptive Threshold use case
Actors Segmentation System.
Description Adaptive thresholding is applied on image.
Data Grayscale image.
Stimulus System command.
Response New image.
Comments None.

Tabular description of the Get Connected Components use case
Actors Segmentation System.
Description Process the image to get all connected objects in it.
Data Binary Image.
Stimulus System command.
Response Array of connected labels.
Comments None.

Tabular description of the Apply Morphological Operation use case
Actors Segmentation System.
Description Get the image after removing small particles to get it back to the original shape.

Data Image resulted from removing the small particles function.
Stimulus System command.
Response New image with the cells in original state.
Comments None.

10 Preliminary Schedule Adjusted

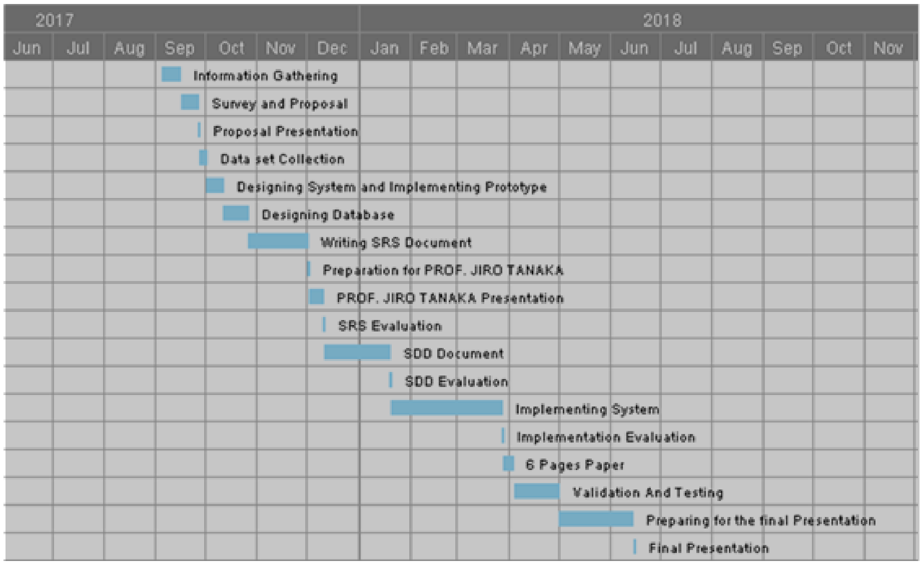


Figure 13: The place in which the sample is placed

11 Preliminary Budget Adjusted

The initial Budget is for the Hardware that Doctors will interact with.
Desktop: Average of 10000
Digital microscope with a camera: average of 250 $

12 Appendices

12.1 Definitions, Acronyms, Abbreviations

- WBC: White blood cells are the cells of the immune system that are involved in protecting the body against both infectious disease and foreign invaders.
- Hematologist: A hematologist is a specialist in hematology, the science or

study of blood, blood-forming organs and blood diseases.
- UI: User interface is the space where interactions between humans and machines occur.
- GIST: A short, simple description of the concept contained in a PLanguage statement.
- SCALE: The scale of measure used by the requirement contained in a Planguage statement.
- METER: The process or device used to establish location on a SCALE contained in a PLanguage statement.
- MUST: The minimum level required to avoid failure contained in a PLanguage statement.
- PLAN: The level at which good success can be claimed contained in a Planguage statement.
- WISH: A desirable level of achievement that may not be attainable through available means contained in a PLanguage statement.
-GLCM: A statistical method of examining texture that considers the spatial relationship of pixels is the gray-level co-occurrence matrix.
-YCBCR: is a family of color spaces used as a part of the color image pipeline in video and digital photography systems. Y is the luma component and CB and CR are the blue-difference and red-difference chroma components.

12.2 Collected material

13 References

[1] K. B. Qanita Bani Baker, Exploiting GPUs to Accelerate White Blood cells Segmentation in Microscopic Blood Images, Information and Communication Systems (ICICS), 2017 8th International Conference on, 2017
(2) Omid Sarrafzadeh1 , Hossein Rabbani1 , Alireza Mehri Dehnavi1 , Ardeshir Talebi2, DETECTING DIFFERENT SUB-TYPES OF ACUTE MYELOGENOUS LEUKEMIA USING DICTIONARY LEARNING AND SPARSE REPRESENTATION, Image Processing (ICIP), 2015 IEEE International Conference on, 2015.
(3) Sos Agaian1, Monica Madhukar 2 , Anthony T. Chronopoulos, A New Acute Leukemia Automated Classification System, Department of Electrical and Computer Engineering1, 2, Department of Computer Science 3 University of Texas at San Antonio, One UTSA Circle, San Antonio, Texas 78249, 2016.
(4) D. P. A. B.K.LAKSHMIKANTH, Acute Myelogenous Leukemia Detection in Blood Microscopic Images using Different Wavelet Family Techniques, International journal of scientific engineering and technology Research, 2016. 21
(5) T. K. I. K. Emmanouil Michail Kosmas Dimitropoulos and N. Grammalidis, Morphological and textural analysis of centroblasts in low-thickness sliced tissue biopsies of follicular lymphoma, Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE,2014.

(6) Emmanouil Michail, Evgenios N. Kornaropoulos, Kosmas Dimitropoulos, Nikos Grammalidis ,Triantafyllia Koletsa, Ioannis Kostopoulos, Detection of centroblasts in H and E stained images of follicular lymphoma, Signal Processing and Communications Applications Conference (SIU) IEEE, 2014 22nd, 2014.
(7) I. M. M. Sos Agaian Senior Member and I. Anthony T. Chronopoulos Senior Member, Automated Screening System for Acute Myelogenous Leukemia Detection in Blood Microscopic Images, IEEE Systems Journal, 2014.
(8) J. R. Chaitali Raje, Detection of Leukemia in microscopic images using image processing, Communications and Signal Processing (ICCSP), 2014 International Conference on IEEE, 2014.
(9) 1 Shailesh J. Mishra , 2 Mrs.A.P.Deshmukh, DETECTION OF LEUKEMIA USING MATLAB, in International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), 2015.
(10) K. L. P. N. and K. P. R. Saravanakumar G.Kiruthiga, Analysis Of Microscopic Blood Images for Detecting Leukemia using Nuclear Segmentation, The SIJ Transactions on Computer Science Engineering and its Application
(11) Application of Support Vector Machine and k-means Clustering Algorithms for Robust Chronic Lymphocytic Leukemia Color Cell Segmentation, International Conference on e-Health Networking, Applications and Services IEEE, 2013.
(12) D. S. H. (Biji G, White Blood Cell Segmentation Techniques in Microscopic Images for Leukemia Detection, IOSR Journal of Dental and Medical Sciences (IOSR-JDMS), 2016.
(13) L Padma Suresh, S. S. (2014). Artificial Intelligence and Evolutionary Algorithms in Engineering Systems . New Delhi, India