Software Requirement Specification Document for Classification of Alzheimer's by DNA Analysis project

Ahmed Samir, Fairuz Soufy, Omar Ehab, Sara Hassan

Lobna Shaheen, Nora El-Samanody, Omar El-Demrdash, Rawan El-Kady
Ashraf AbdelRaouf, Lamiaa Nabil

January 16, 2020

1 Introduction

1.1 Purpose of this document

The purpose of this software requirement document is to present a detailed description of the Classification of Alzheimer's (AD) by DNA Analysis project. The main purpose of this project is to be able to classify AD patients to healthy patients and people who carry AD. Early diagnosis of AD may help in slowing down the progression of the disease considerably. This document clarifies the purposes and features of the project.

1.2 Scope of this document

The system is developed to reveal which if the patient is healthy and if not, how much is the progression of the disease in his body. Either way, this classification helps the patient and the doctors diagnose the disease early on which gives them a chance to slow down the progression of his disease depending on the stage the patient is in since early diagnosis is key in these type of situations.

1.3 Overview

The project distinguishes between two stages which are healthy patients and patients who have AD. And also we can know if the patient has AD from their patient's history.

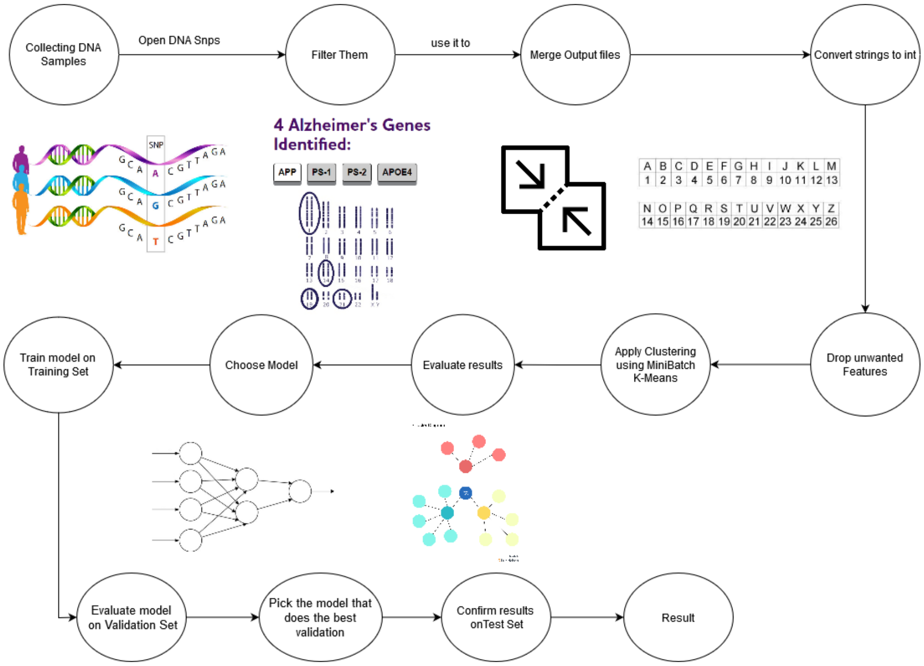

Collecting DNA Samples
Open DNA Snps
Filter Them
use it to
Merge Output files
Convert strings to int
SNP
4 Alzheimer's Genes Identified:
APP
PS-1
PS-2
APOE4
A B C D E F G H I J K L M
1 2 3 4 5 6 7 8 9 10 11 12 13
N O P Q R S T U V W X Y Z
14 15 16 17 18 19 20 21 22 23 24 25 26
Train model on Training Set
Choose Model
Evaluate results
Apply Clustering using MiniBatch K-Means
Drop unwanted Features
Evaluate model on Validation Set
Pick the model that does the best validation
Confirm results onTest Set
Result

Figure 1: System Overview

1.4 Business Context

AD specialists and the patients will be able to know if they are carrying the disease or not as early diagnosis of AD may help in slowing down the progression of the disease. AD specialists will benefit more from early diagnosis as they will have time to see how they can slow down the progression of the disease.

2 General Description

2.1 Product Functions

The system main functionality is to take the file that contains the patient's DNA and show if the patient has AD or not. If he is healthy, it will show if he is totally healthy or he has a chance to have AD. If he already have AD, the system will show the progression of the disease.

2.2 Similar System Information

1. **Predicting cancer type from tumour DNA signatures.**[1] The researchers tried to know the cancer type more accurately to give the best course of treatment to the patient. Their goal was to know the cancer type

more accurately than before. They collected sequenced tumour DNA from Cancer Genomes. Around 6640 tumor samples showing 28 cancer types and used linear support vector machines with feature selection to predict the cancer type. They found that linear support vector machine is the most accurate model to predict cancer type with accuracy 49.4%. We saw how they used machine learning techniques to predict the cancer type.

2. **Convolutional neural networks for classification of alignments of non-coding RNA sequences [2]**. The researchers wanted to prove that Convolutional neural networks (CNN) is a good way for RNA analysis. The main problem statement is to classify non-coding RNA sequences into positive and negative classes to prove it's classifying correctly. The CNN classified the pairwise alignments of sequences for accurate clustering of sequences and show the benefits of the CNN method of inputting pairwise alignments for clustering of non-coding RNA (ncRNA) sequences and for motif discovery. The researchers solved a problem very similar to ours with a method analogous with the one that we intended to use demonstrating the feasibility of making the system. The accuracy of this project is 94.5%.

3. **Convolutional Neural Networks In Classifying Cancer Through DNA Methylation [3]**. The researchers decided to pursue the topic because traditional methods of cancer identification are generally not efficient. Moreover they usually require effort and have lower accuracy. The main problem is that the regular methods of cancer detection are quite troublesome moreover the possibility of false positives is present so a method with a higher accuracy was needed. The contribution the research team accomplished was building a model that can learn the changing DMR patterns to detect 32 cancer types. The model was able to attain a training accuracy of 96.54% and a testing accuracy of 92.87% the model was based on 10000 samples.

4. **Recurrent Convolutional Neural Networks for Text Classification [4]**.Researchers wrote this paper to use Recurrent convolutional neural network (RCNN) for text classification. The key problem in text classification is feature representation. The reseachers used four separate text datasets to perform CNN and RCNN. Thus, they discovered that Neural Networks can collect more contextual feature data than conventional BoW-based approaches. This paper is important for my project as we may use the Recurrent Convolutional Neural Network (RCNN) for text classification.

5. **DNA Sequence Classification by Convolutional Neural Network [5]** The motivation of this paper is to prove that CCN has an excellent performance in many fields even dealing with A, C, T and genes of the DNA. The main problem statement of the work is that DNA sequences are sequences of successive letters without space. There is no term of word in DNA sequence. so, we made a method to translate DNA sequences to

sequence of words in order to apply the same representation technique for text data without losing position. The researches reached that the lowest improvement is nearly 1% of accuracy and the highest improvement is over 6% of accuracy. These improvements are quite high in comparison with other approaches such as finding good representations for sequences. This paper was important and valuable because it added to my knowledge in the sense that it showed how computer science and neural networks can help in diagnosing more than just one disease moreover it mentioned a plethora of guides and tips that may help in the proposed system.

2.3 User Characteristics

The expected users of the system should be either admin(Head of laboratory) or lab technicians. Therefore, the system's expected users will have knowledge or may have past experiences dealing with such applications as logging in, uploading samples and checking results. The user will consequently adapt with the system the more he uses it. Moreover, using system would be simple and straightforward.

2.4 User Problem Statement

It's only possible to classify AD patients into two stages (healthy patients and patients with sever AD). It's done by asking a set of questions to the patient to determine which level of AD he has. It's not 100% accurate because some patients can lie or forget the answers. To get the best results we combine both the questions and the DNA test to reach the most accurate diagnosis. Detecting AD in early stages could help prevent its progression.

2.5 User Objectives

The user's purpose and goal is to have a final product that takes the patient DNA and reveal if the patient is healthy or if he is suffering from AD in any level. It should also be easy to use with a straightforward design in order to reduce user friction as much as possible.

2.6 General Constraints

The uploaded file should carry the DNA not another content (CSV File). The computer that is supposed to run the system should have a minimum processor of 4GHz quad core and a minimum amount of memory of 4GB with recommended 8GB to run the system smoothly.

3 Functional Requirements

3.1 Register User

Use Case Name	Register User
Input	Name, username and password
Output	User information added successfully
Prerequisite	N/A
Priority	Must have
Risk	The user may insert wrong input about an employee
Dependency	N/A
Description	This function takes the user information and inserts him into the system's database

3.2 Login User

Use Case Name	Login
Input	username and password
Output	If successful, the system redirects the user to his page. If not successful the system asks the user to re-enter his information
Prerequisite	The user must be registered in the system
Priority	Must have
Risk	N/A
Dependency	Dependant on 3.1
Description	This function takes the credentials of the user and checks if the user is registered in the system or not

3.3 Upload DNA

Use Case Name	Upload DNA
Input	Folder that contains csv files
Output	If successful, the system starts preprocessing the data and showing the result. If not successful the system asks the user to check the input again
Prerequisite	The user must be logged in the system
Priority	Must have
Risk	The user may choose wrong directory
Dependency	Dependant on 3.2
Description	This function takes the files that has the format(.gb) in the directory and starts the preprocessing

3.4 View Result

Use Case Name	View Result
Input	N/A
Output	The system shows the user in which stage is the patient
Prerequisite	The user must be logged in the system and has uploaded DNA
Priority	Must have
Risk	N/A
Dependency	Dependant on 3.3
Description	The system takes the sample taken from the patient and starts processing the data and shows the result

3.5 Check medical history

Use Case Name	Check medical history
Input	Patient's SSN
Output	if successful the system shows the medical history of the patient. if not successful, the system asks the user to re-check the SSN entered
Prerequisite	The user must be logged in the system
Priority	Must have
Risk	The user may enter wrong SSN
Dependency	Dependant on 3.2
Description	The system takes the SSN entered and search the patients database then view the medical history if found

3.6 Print result

Use Case Name	Print result
Input	N/A
Output	the system prints the result in a pdf document
Prerequisite	The user must be logged in the system
Priority	Optional
Risk	N/A
Dependency	Dependant on 3.4 or 3.5
Description	The system takes the result and prints it out in a pdf document

3.7 Logout

Use Case Name	Logout
Input	N/A
Output	If successful, the system redirects the user to his page. If not successful the system asks the user to re-enter his information
Prerequisite	The user must be logged in the system
Priority	Must have
Risk	N/A
Dependency	Dependant on 3.2
Description	This function is used to log out the user

3.8 Filter

Use Case Name	Filter
Input	A folder that contains DNA csv files
Output	Filtered csv files
Prerequisite	A csv file must exist
Priority	Must have
Risk	the user may upload wrong files
Dependency	Dependant on 3.3
Description	This function takes the csv file and keeps only the data within the AD range

3.9 MergeCSV

Use Case Name	MergeCSV
Input	A folder that contains csv files
Output	A csv file
Prerequisite	Csv files must exist
Priority	Must have
Risk	The user may choose empty folder
Dependency	Dependant on 3.8
Description	This function takes all the csv files in a folder and merges them together in one csv file

3.10 Conversion

Use Case Name	Conversion
Input	A csv file
Output	A csv file
Prerequisite	Csv files must exist
Priority	Must have
Risk	N/A
Dependency	Dependant on 3.9
Description	This function converts all the string inside the csv file to numeric data to prepare it for clustering

3.11 Cluster

Use Case Name	Cluster
Input	A csv file
Output	A csv file contains the results of clustering
Prerequisite	A csv fils must exist
Priority	Must have
Risk	N/A
Dependency	Dependant on 3.10
Description	This function applies Kmeans and Mini batch Kmeans clustering on the giving csv file

3.12 Searcher

Use Case Name	Searcher
Input	A .txt or .gb file
Output	A .txt file
Prerequisite	must extract the four desired chromosomes out of the whole Genome and specific locations given
Priority	Must have
Risk	N/A
Dependency	Dependant on 3.3
Description	This function extracts the desired area out of the whole chromosome

3.13 Remover

Use Case Name	Remover
Input	A .txt file
Output	A .txt file
Prerequisite	must extraxt only the desired areas out of the chromosomes
Priority	Must have
Risk	N/A
Dependency	Dependant on 3.12
Description	This function removes and clears all the unwanted characters

3.14 ToCsv

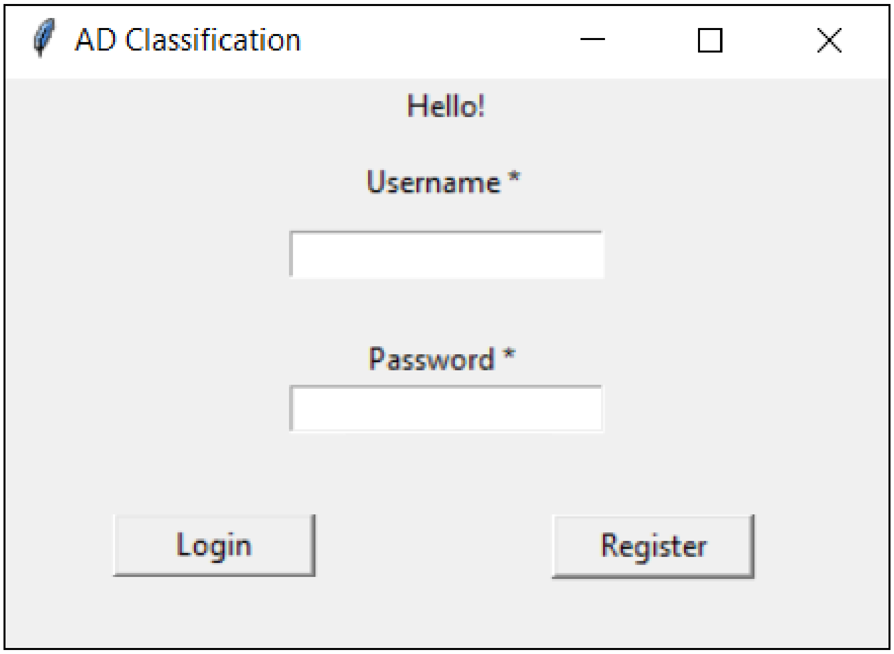
Use Case Name	ToCsv
Input	A .txt file
Output	A .csv file
Prerequisite	file must contain no other characters except our four main characters(a,c,g,t)
Priority	Must have
Risk	N/A
Dependency	Dependant on 3.13
Description	This function divides the text into three's, separates them by (',') and saves them into a csv file

4 Interface Requirements

4.1 User Interfaces

The system designed with a friendly UI to be easily used by the user. On starting the system, the user is asked to login or to register. If he logs in, another window will be shown to upload .csv file of the DNA and when he uploads it the result of determining which stage he is in will appear. If the user chooses to register he'll be asked to enter his wanted credentials and will be asked to log in to use the system.

4.1.1 GUI



AD Classification

Hello!

Username *

Password *

Figure 2: User Login

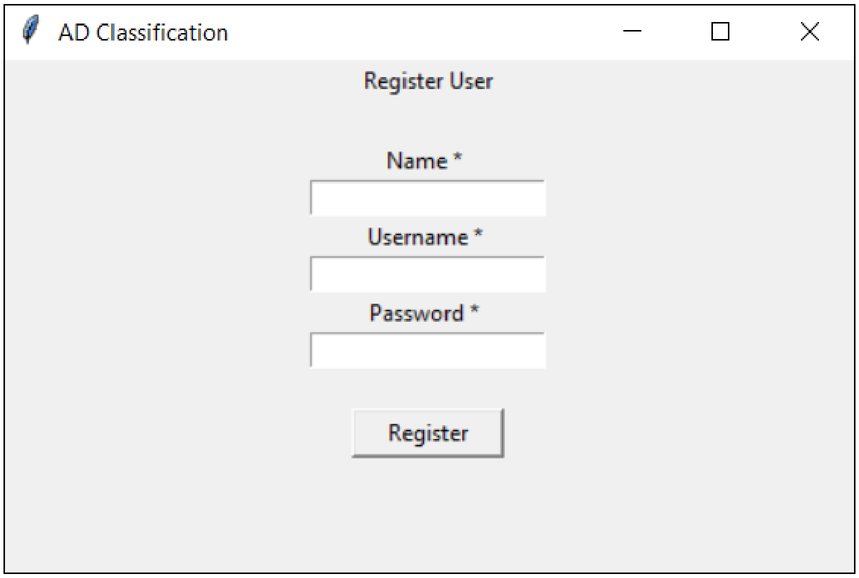


Figure 3: Register User

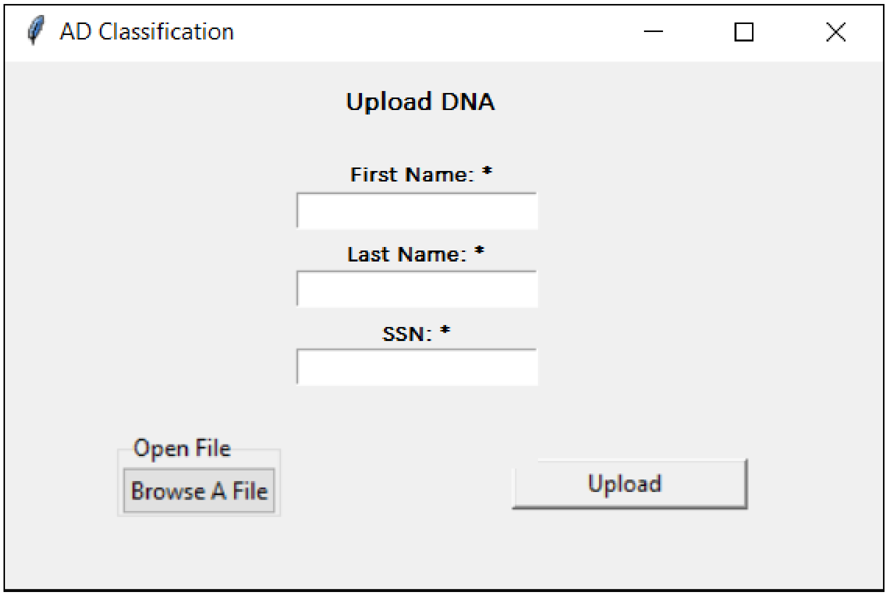
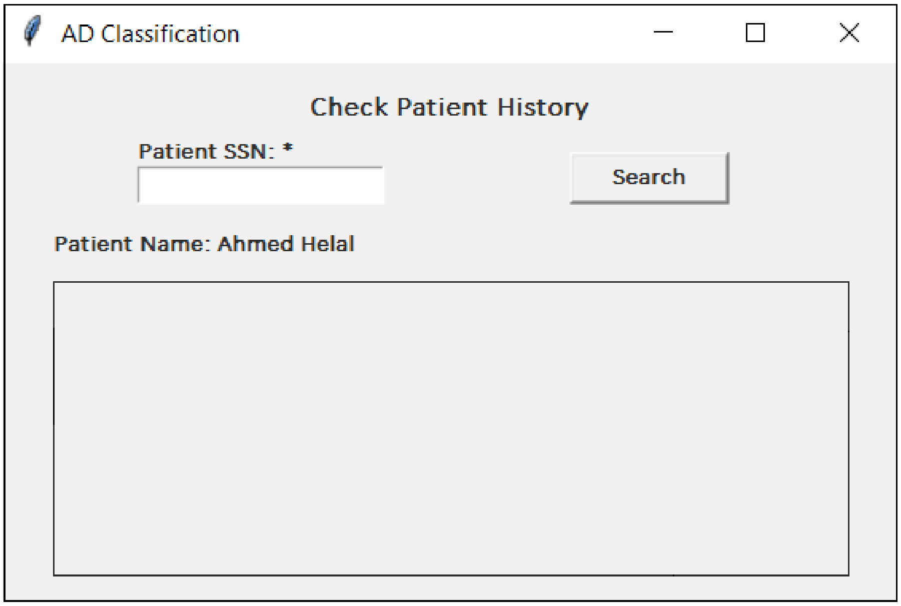
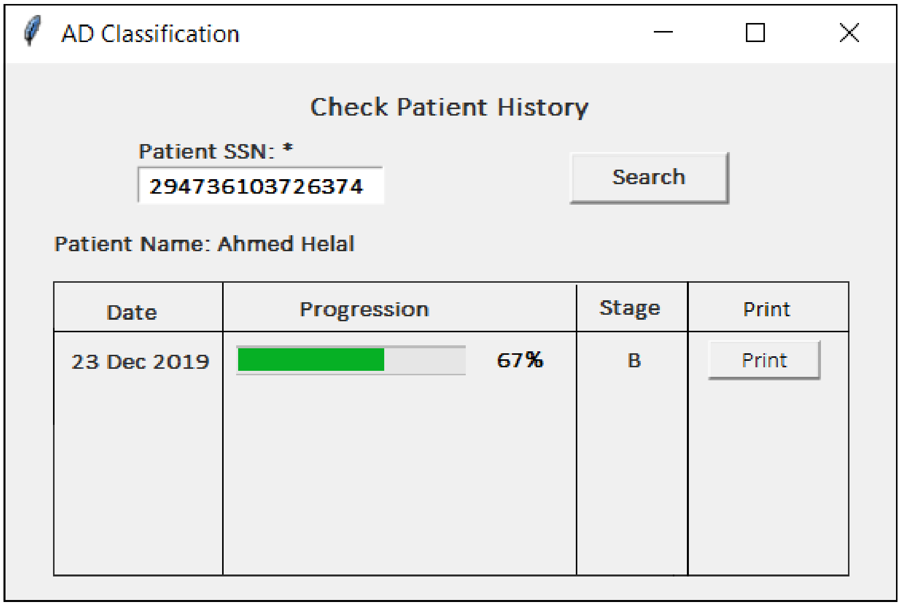


Figure 4: Upload DNA File



Patient Name: Ahmed Helal

Figure 5: Patient History



Check Patient History

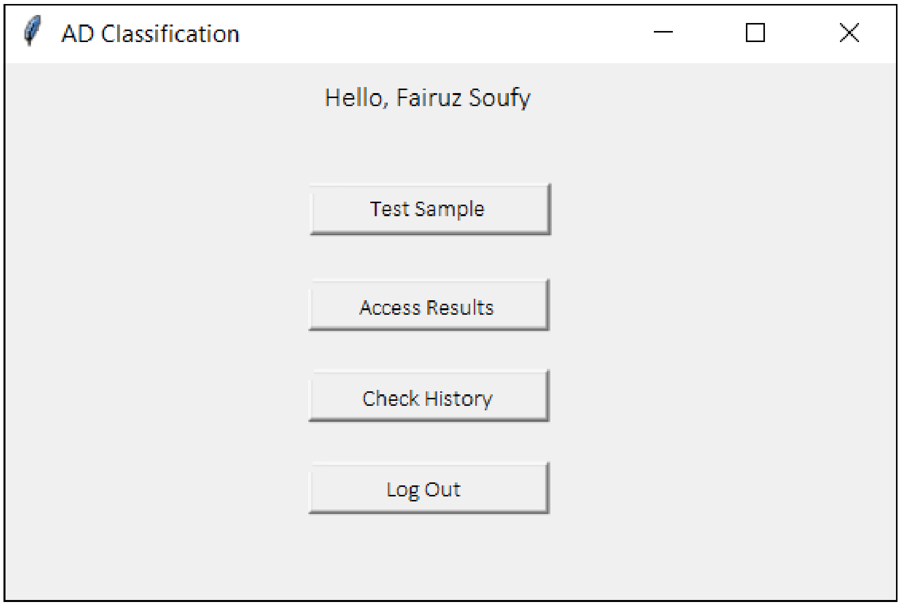
Patient SSN: *

294736103726374

Patient Name: Ahmed Helal

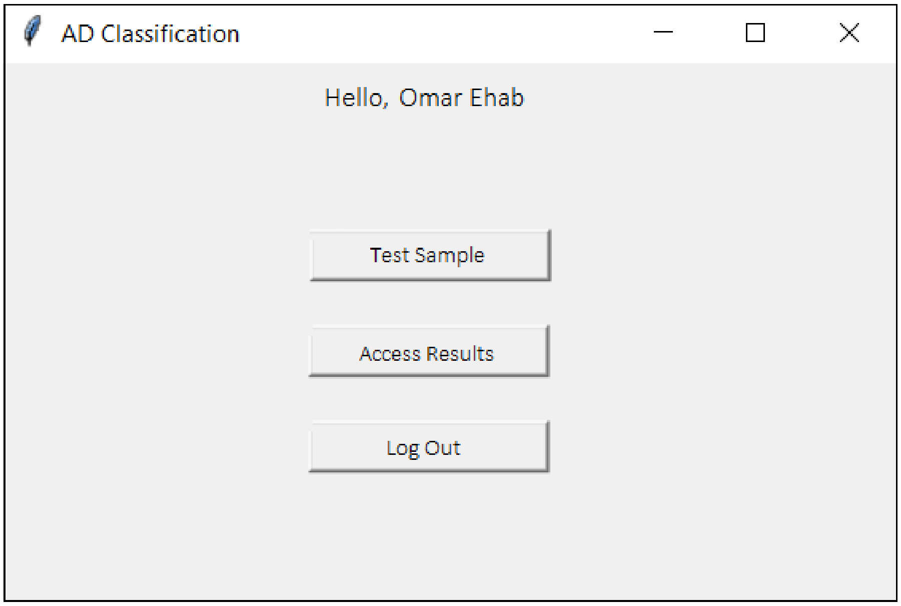
Date	Progression	Stage	Print
23 Dec 2019	67%	B	Print

Figure 6: Patient History



Hello, Fairuz Soufy

Figure 7: Admin after Login



Hello, Omar Ehab

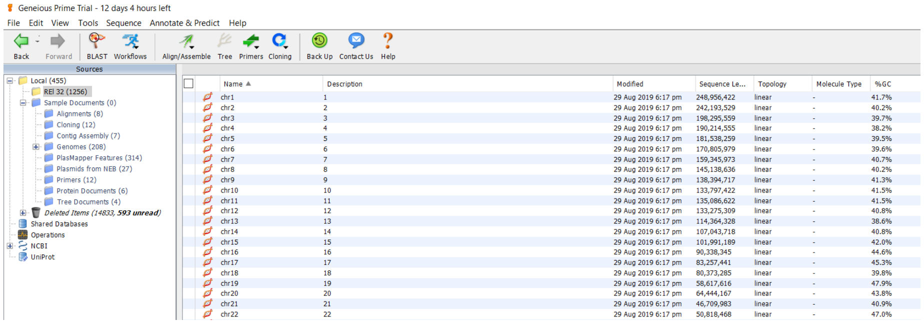
Figure 8: Lab Technician after Login

4.1.2 CLI

- Run GUI: python filename.py

4.2 Software Interfaces

Geneious Prime Software is used to open sequenced DNA and to cut required chromosomes needed.



Name ▲	Description	Modified	Sequence Le...	Topology	Molecule Type	%GC
chr1	1	29 Aug 2019 6:17 pm	248,956,422	linear	-	41.7%
chr2	2	29 Aug 2019 6:17 pm	242,193,529	linear	-	40.2%
chr3	3	29 Aug 2019 6:17 pm	198,295,559	linear	-	39.7%
chr4	4	29 Aug 2019 6:17 pm	190,214,555	linear	-	38.2%
chr5	5	29 Aug 2019 6:17 pm	181,538,259	linear	-	39.5%
chr6	6	29 Aug 2019 6:17 pm	170,805,979	linear	-	39.6%
chr7	7	29 Aug 2019 6:17 pm	159,345,973	linear	-	40.7%
chr8	8	29 Aug 2019 6:17 pm	145,138,636	linear	-	40.2%
chr9	9	29 Aug 2019 6:17 pm	138,394,717	linear	-	41.3%
chr10	10	29 Aug 2019 6:17 pm	133,797,422	linear	-	41.5%
chr11	11	29 Aug 2019 6:17 pm	135,086,622	linear	-	41.5%
chr12	12	29 Aug 2019 6:17 pm	133,275,309	linear	-	40.8%
chr13	13	29 Aug 2019 6:17 pm	114,364,328	linear	-	38.6%
chr14	14	29 Aug 2019 6:17 pm	107,043,718	linear	-	40.8%
chr15	15	29 Aug 2019 6:17 pm	101,991,189	linear	-	42.0%
chr16	16	29 Aug 2019 6:17 pm	90,338,345	linear	-	44.6%
chr17	17	29 Aug 2019 6:17 pm	83,257,441	linear	-	45.3%
chr18	18	29 Aug 2019 6:17 pm	80,373,285	linear	-	39.8%
chr19	19	29 Aug 2019 6:17 pm	58,617,616	linear	-	47.9%
chr20	20	29 Aug 2019 6:17 pm	64,444,167	linear	-	43.8%
chr21	21	29 Aug 2019 6:17 pm	46,709,983	linear	-	40.9%
chr22	22	29 Aug 2019 6:17 pm	50,818,468	linear	-	47.0%

Figure 9: Geneious Prime

5 Performance Requirements

The system should have sufficient processing power and memory that can allow the classification process to be done on the hardware locally by taking the sample and the trained model to generate a prognosis.

6 Design Constraints

6.1 Standards Compliance

Because of their lack of professional computer skills, the system needs to be user friendly to ease the process of doctors performing the required tasks.

6.2 Hardware Limitations

The system will perform poorly if not equipped with a minimum processor of 4GHz quad core and a minimum amount of memory of 4GB with recommended 8GB in order to be able to handle big files like the DNA samples files.

7 Other non-functional attributes

7.1 Security

Security is a very important factor for the project so no one has the access to the patient's data unless he has a profile and his profile is allowed to access the data.

7.2 Reliability

The system is reliable enough to handle all failure events. The time needed to diagnose a patient on the system has an average speed to check since the data is large.

7.3 Portability

The system is written by Python so it is an executable file that can be deployed on Windows operating system and Mac OS.

7.4 Efficiency

The system is very efficient with the way it handles both system memory and storage. Since the dataset is very large and many operations are done on each file in the dataset the system handles each file and moves the desired portion of the file into a new smaller sized file therefore the dataset's size is reduced significantly, moreover after processing the files we delete them in order to eliminate any wastage of the system resources

7.5 Maintainability

The code is very simple so it has the availability to be maintained later.

8 Preliminary Object-Oriented Domain Analysis

8.1 Inheritance Relationships

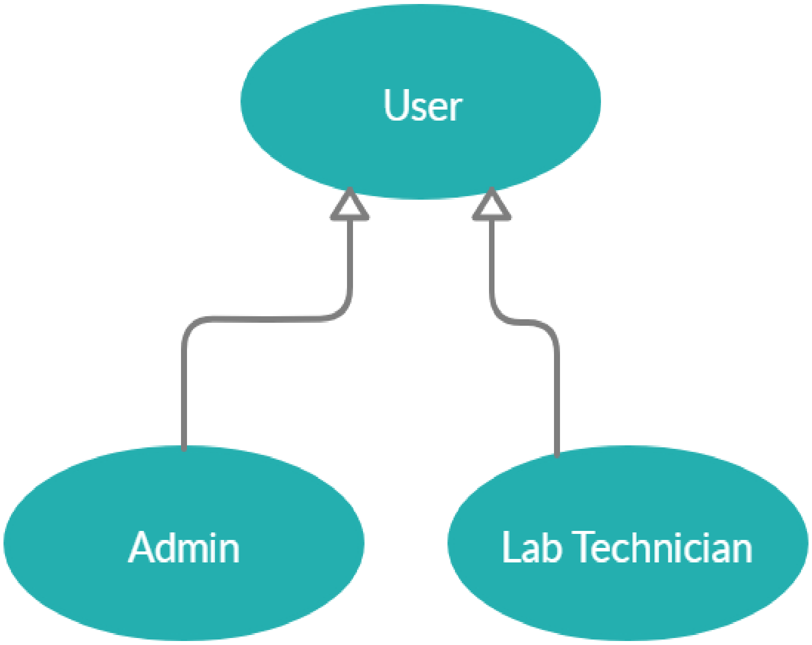


Figure 10: Inheritance Relationships

8.2 Class descriptions

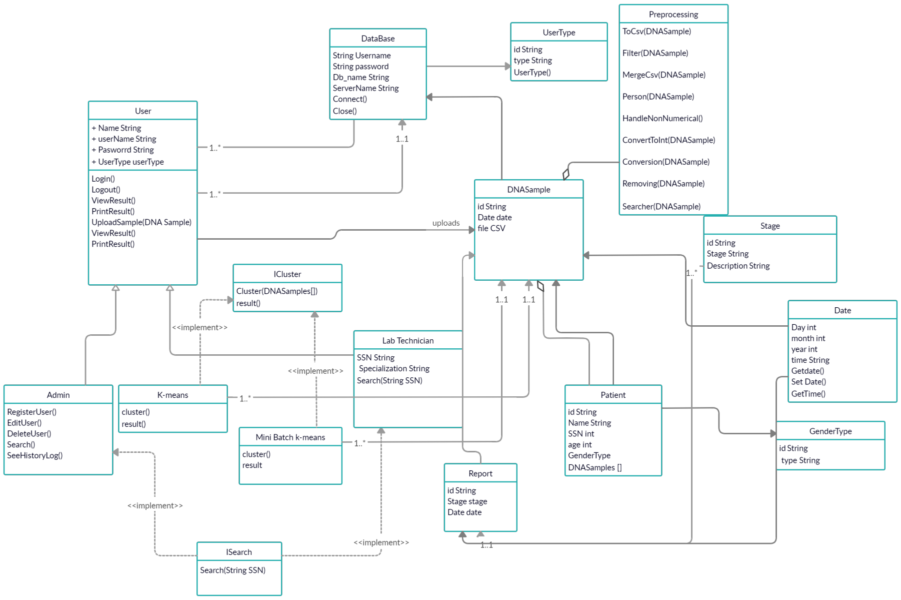


Figure 11: Class diagram

Each class description should conform to the following structure:

8.2.1 User

1. Class Name: User
2. Super Classes: N/A
3. Sub Classes:Admin,Lab Technician
4. Purpose: this class is the main class holds all functionality for other classes
5. Collaborations: userType
6. Attributes:Name,Username,password,user type.
7. Operations:Login,Log Out, Uploads sample,view results,print Results.

8.2.2 Admin

1. Class Name: Admin
2. Super Classes: User

3. Sub Classes:N/A
4. Purpose: this class is the holds all functionalities for Admin
5. Collaborations: N/A
6. Attributes:N/A
7. Operations:CRUD Lab Technician

8.2.3 Lab Technician

1. Class Name: LabTechnician
2. Super Classes: User
3. Sub Classes:N/A
4. Purpose: this class is the holds all functionalities for Lab Technician
5. Collaborations: N/A
6. Attributes: specualization, SSN,gender.
7. Operations:none.

8.2.4 DNA Sample

1. Class Name: DNA Sample
2. Super Classes: N/A
3. Sub Classes:N/A
4. Purpose: this class is the holds all information about a DNA Sample.
5. Collaborations: patient,Report.
6. Attributes: sample id , sample date ,sample File.
7. Operations:none.

8.2.5 patient

1. Class Name: patient
2. Super Classes: N/A
3. Sub Classes:N/A
4. Purpose: this class is the holds all information about a any patient.
5. Collaborations: gender Type.
6. Attributes:id ,name, SSN, age,Gender.
7. Operations:none.

8.2.6 Report

1. Class Name: Report
2. Super Classes: N/A
3. Sub Classes:N/A
4. Purpose: this class is the holds all information about DNA sample report .
5. Collaborations: DNA Sample,Stage,patient.
6. Attributes:id ,date.
7. Operations:none.

8.2.7 Preprocessing

1. Class Name: Preprocessing
2. Super Classes: N/A
3. Sub Classes:N/A
4. Purpose: this class is responsible for all the processing that will be done before clustering.
5. Collaborations: DNA Sample
6. Attributes:none.
7. Operations:Searcher, Removing,ToCsv,Filter,MergToCsv,Convert

8.2.8 ICluster

1. Class Name: ICluster
2. Super Classes: N/A
3. Sub Classes:K-means, Mini Batch
4. Purpose: This interface initiates the cluster function.
5. Collaborations: DNA Sample
6. Attributes:id, Stage .example stage A or B.
7. Operations: none.

8.2.9 Stage

1. Class Name: Stage
2. Super Classes: N/A
3. Sub Classes:N/A
4. Purpose: This class responsible for storing the stages types
5. Collaborations: report
6. Attributes: id, stage.
7. Operations: none.

9 Operational Scenarios

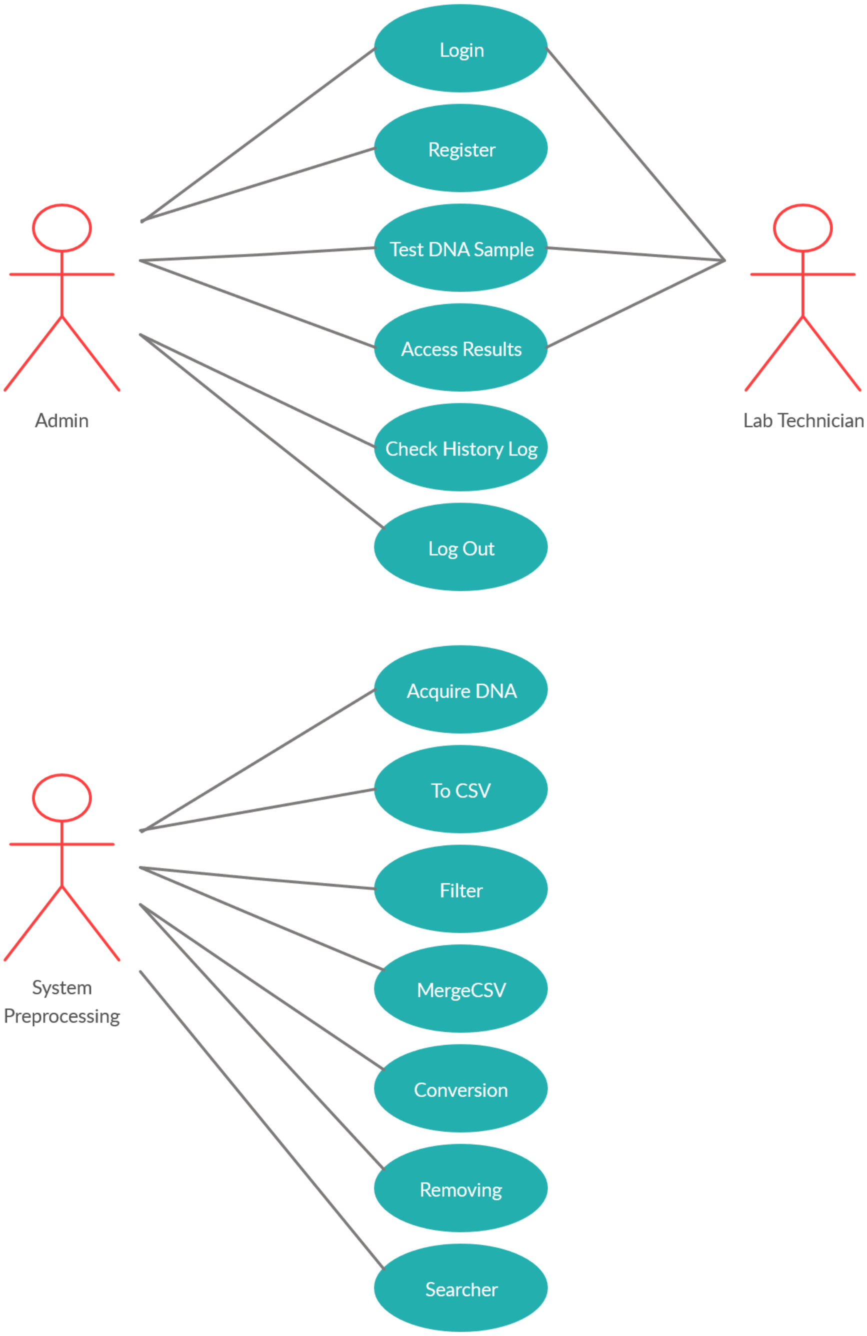
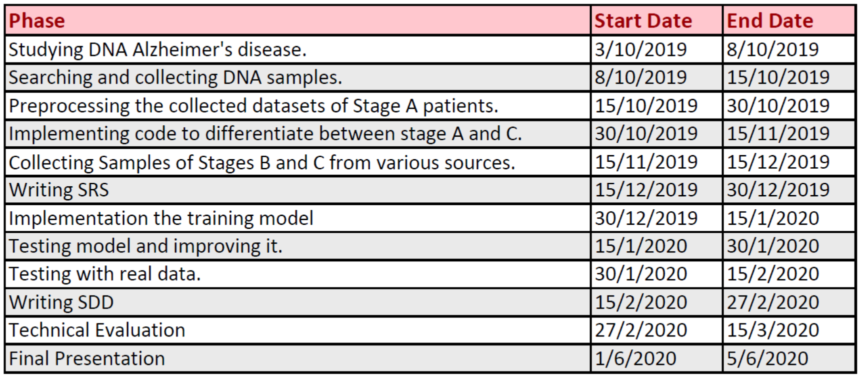


Figure 12: Use Case

There are two types of users, lab technician and admin. The user will first open the system, he is required to log in with a username and a password or to register his account. If he chooses to login and the credentials is incorrect he'll be asked to enter the correct credentials. When the user logs in successfully and the user is an admin then four option will appear to him: Test Sample, Access Results, Check Patient History and logout. And if the user is the lab technician then three options will appear to him: Test Sample, Access Results and logout. If Test Sample is chosen, then the user is asked to enter the patient's first and last name, the patient's SSN, and the file that has his DNA. If the user chooses Access Results, he is required to enter the patient's SSN to view his latest test. If the user chooses Check Patient History, he'll be asked to enter the patient's SSN to view all his test. The last option is logout and when he chooses it, his session end and he doesn't have an access to the system anymore.

10 Preliminary Schedule Adjusted



Phase	Start Date	End Date
Studying DNA Alzheimer's disease.	3/10/2019	8/10/2019
Searching and collecting DNA samples.	8/10/2019	15/10/2019
Preprocessing the collected datasets of Stage A patients.	15/10/2019	30/10/2019
Implementing code to differentiate between stage A and C.	30/10/2019	15/11/2019
Collecting Samples of Stages B and C from various sources.	15/11/2019	15/12/2019
Writing SRS	15/12/2019	30/12/2019
Implementation the training model	30/12/2019	15/1/2020
Testing model and improving it.	15/1/2020	30/1/2020
Testing with real data.	30/1/2020	15/2/2020
Writing SDD	15/2/2020	27/2/2020
Technical Evaluation	27/2/2020	15/3/2020
Final Presentation	1/6/2020	5/6/2020

Figure 13: Project Timeline

11 Preliminary Budget Adjusted

1- The system needs Genius Prime Application because it is used in our project to open DNA sequence (.fna) files and convert it into the chromosome file type (.fasta) and (.gp) in order to be processed by the CNN we're using. It is 200$ per year for student license and 450$ for government and non-profit organizations to use.
2- The system needs an average ram of 64GB, as some files require large memory to view.

12 Appendices

12.1 Abbreviations

AD : Alzheimer's Disease
SNP: single nucleotide polymorphism (pronounced "snips")

12.2 Collected material

References

[1] Kee Pang Soh, Ewa Szczurek, Thomas Sakoparnig, and Niko Beerenwinkel. Predicting cancer type from tumour dna signatures. *Genome medicine*, 9(1):104, 2017.

[2] Genta Aoki and Yasubumi Sakakibara. Convolutional neural networks for classification of alignments of non-coding rna sequences. *Bioinformatics*, 34(13):i237–i244, 2018.

[3] Soham Chatterjee, Archana Iyer, Satya Avva, Abhai Kollara, and Malaikannan Sankarasubbu. Convolutional neural networks in classifying cancer through dna methylation. *arXiv preprint arXiv:1807.09617*, 2018.

[4] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[5] Ngoc Giang Nguyen, Vu Anh Tran, Duc Luu Ngo, Dau Phan, Favorisen Rosyking Lumbanraja, Mohammad Reza Faisal, Bahriddin Abapihi, Mamoru Kubo, and Kenji Satou. Dna sequence classification by convolutional neural network. *Journal of Biomedical Science and Engineering*, 9(05):280, 2016.