

Classification of Alzheimer's by DNA analysis

by

Ahmed Samir

Fairuz Soufy

Omar Ehab

Sara Elbedeawi

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Bachelor of Computer Science

in

Department of Computer Science

in the

Faculty of Computer Science

of the

Misr International University, EGYPT

Thesis advisor:

Dr. Ashraf Abdelraouf

Eng. Lobna Shaheen

(July 2020)

Abstract

One advantage of exploring the human DNA is the revelation of its contribution to many human diseases. In result of exploring the DNA, there were specific genes in specific chromosomes in the DNA that reveal Alzheimer's Disease (AD). Early diagnosis of AD helps in slowing down the progression of the disease considerably which is our main goal. AD is a progressive brain disorder that slowly causes decline in memory affecting the patient's life. AD can be detected by screening the whole genome sequence or by finding the mutations that happens in the Single-Nucleotide Polymorphism (SNP).

Acknowledgments

We would like to thank our supervisors Dr. Ashraf Abdelraouf and Eng. Lobna Shaheen for their patient guidance, encouragement and advice that they have provided throughout the whole year. We have been extremely lucky to have a supervisor who cared so much about my work and responded to our questions and queries so promptly. We would like also to thank some people in faculty of pharmacy. Prof. Lamiaa Nabil, Dr Rawan Hossam, Dr. Omar Eldemerdash and Doctor Nora Elsamanody. They guided us since day one, stood by our side, did all the researches needed and helped us a lot. A special thanks to Dr. Mohamed Shahin who has been helping us from America and guiding all of us to the right direction. We would like to thank Misr International University for providing us the perfect atmosphere for learning from a very qualified staff.

Contents

Abstract	ii
Acknowledgments	iii
List of Tables	4
List of Figures	5
1 Introduction	6
1.1 Background	6
1.1.1 DNA scanning types	8
1.2 Motivation	9
1.3 Problem Definition	9
1.4 Project Description	9
1.5 Scope	9
1.6 Project Overview	10
1.7 Project Management and Deliverable	12
1.7.1 Tasks and Time Plan	12
1.8 Abbreviations	12
2 Literature Work	13
2.1 Related work	13
2.1.1 Whole Genome Sequence researches	13
2.1.2 SNPs researches	14
3 System Requirements Specification	15
3.1 Introduction	15
3.1.1 Purpose of this chapter	15
3.1.2 Scope of this chapter	15
3.1.3 Overview	15
3.1.4 Business Context	17
3.2 General Description	17
3.2.1 Product Functions	17
3.2.2 User Characteristics	17

3.2.3	User Problem Statement	17
3.2.4	User Objectives	17
3.2.5	General Constraints	18
3.3	Functional Requirements	18
3.3.1	Register User	18
3.3.2	Login User	18
3.3.3	Upload DNA	19
3.3.4	View Result	19
3.3.5	Check medical history	20
3.3.6	Print result	20
3.3.7	Logout	20
3.3.8	Filter	21
3.3.9	MergeCSV	21
3.3.10	Conversion	21
3.3.11	Cluster	22
3.3.12	Searcher	22
3.3.13	Remover	22
3.3.14	ToCsv	23
3.4	Interface Requirements	23
3.4.1	User Interfaces	23
3.4.2	Software Interfaces	29
3.4.3	Communication Interface	29
3.5	Performance Requirements	29
3.6	Design Constraints	29
3.6.1	Hardware Limitations	29
3.7	Other non-functional attributes	29
3.7.1	Security	29
3.7.2	Reliability	30
3.7.3	Portability	30
3.7.4	Efficiency	30
3.7.5	Maintainability	30
3.8	Preliminary Object-Oriented Domain Analysis	31
3.8.1	Inheritance Relationships	31
3.8.2	Class descriptions	32
3.9	Operational Scenarios	36
3.10	Preliminary Schedule Adjusted	37
3.11	Preliminary Budget Adjusted	37
3.12	Appendices	38
3.12.1	Definitions, Acronyms, Abbreviations	38
4	Software Design Document	39
4.1	Introduction	39
4.1.1	Purpose	39
4.1.2	Scope	39
4.1.3	Overview	39

4.1.4	Definitions and Acronyms	40
4.2	System Overview	40
4.3	System Architecture	42
4.3.1	Architectural Design	42
4.3.2	Decomposition Description	43
4.3.3	Design Rationale	52
4.4	Data Design	54
4.4.1	Data Description	54
4.4.2	Data Dictionary	54
4.5	Component Design	55
4.6	Human Interface Design	57
4.6.1	Overview of User Interface	57
4.6.2	Screen Images	57
4.7	Requirements Matrix	63
5	Evaluation	64
5.1	Introduction	64
5.2	Algorithm Assessment 1	64
5.2.1	Setup	64
5.2.2	Goals	64
5.2.3	Findings	64
5.3	Algorithm Assessment 2	65
5.3.1	Setup	65
5.3.2	Goal	65
5.3.3	Findings	65
6	Conclusion	67
6.1	Future directions	67
	References	68

List of Tables

4.1	Requirement Matrix	63
5.1	Comparing different algorithms to choose the proper classifier	65
5.2	Comparing different algorithms to choose the proper clustering algorithm .	66

List of Figures

1.1	The human DNA	7
1.2	The 4 Chromosomes responsible for AD	7
1.3	The 4 AAD genes in the 4 mentioned chromosomes	8
1.4	The whole genome sequenced DNA	8
1.5	System Overview	11
1.6	Project Timeline	12
3.1	System Overview	16
3.2	Home Screen	24
3.3	Register Screen	25
3.4	Login Screen	26
3.5	Admin Screen	27
3.6	Researcher Screen	28
3.7	Inheritance Relationships	31
3.8	Class diagram	32
3.9	Use Case	36
3.10	Project Timeline	37
4.1	System Overview	41
4.2	Architecture Diagram	42
4.3	Context Diagram	42
4.4	Class Diagram	43
4.5	Activity Diagram	48
4.6	Admin Sequence Diagram	49
4.7	Researcher Sequence Diagram	51
4.8	Database Diagram	54
4.9	SVR	56
4.10	Home Screen	58
4.11	Register Screen	59
4.12	Login Screen	60
4.13	Admin Screen	61
4.14	Researcher Screen	62
5.1	Statistical Report	66

Chapter 1

Introduction

Machine learning and Genomics are two rapidly growing fields, combining them together gives us better opportunities to improve healthcare treatments.

1.1 Background

Alzheimer's disease (AD) is a permanent brain disorder that slowly destroys memory and the ability to carry out simple tasks [1]. The DNA is a long molecule that contains our distinctive genetic code. Sort of a formula book, it holds the instructions for creating all the proteins in our bodies. DNA consists of 2 strands that wrapped around one another to create a helix form, sort of a spiral stairs. DNA exists in a cell's nucleus composed of structures called chromosomes shown in Figure 1.1 [2].

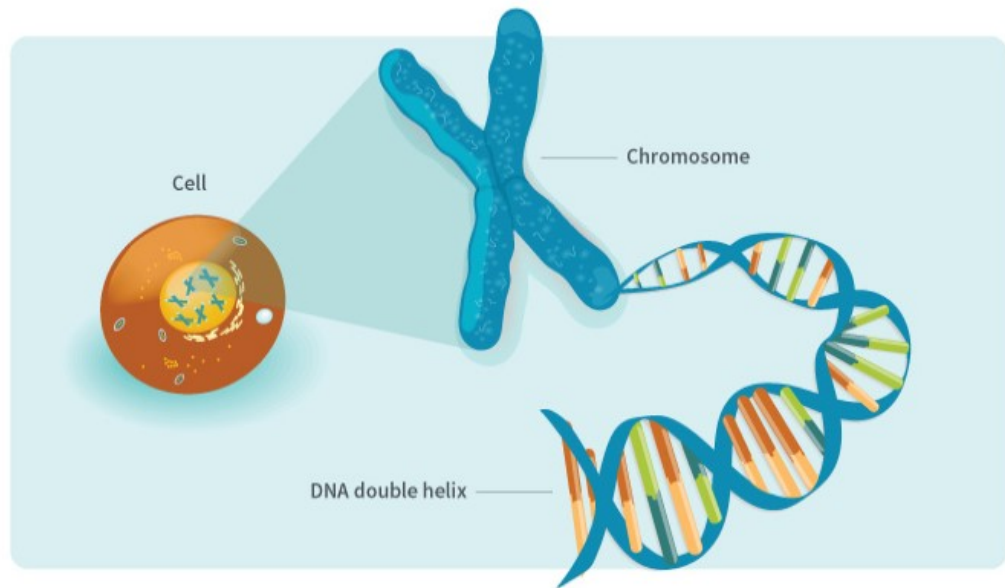


Figure 1.1: The human DNA

A chromosome is made of a very long strand of DNA and have many thousand genes. Every human cell have 23 pairs of chromosomes, for a total of 46 chromosomes in each cell [3]. To detect AD, we search inside four definite chromosomes (chromosomes number 1, 14, 19, 21) shown in Figure 1.2[4]. In these chromosomes, we search inside them for four specific genes related to the detection of AD. Each chromosome contains a huge number of segments called genes and each gene has a particular location on the chromosome.



Figure 1.2: The 4 Chromosomes responsible for AD

The four genes related to AD are: Amyloid precursor protein (APP), Presenilin-1 (PSEN-1), Presenilin-2 (PSEN-2), and Apolipoprotein (APOE4) shown in Figure 1.3 [5].

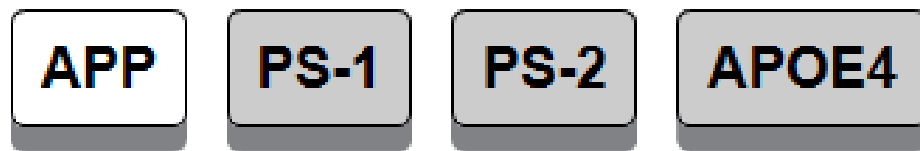


Figure 1.3: The 4 AAD genes in the 4 mentioned chromosomes

Each gene is made of a sequenced codons composed of three nucleotide represented by either one of the following four letters: Adenine(A), Thymine (T), Cytosine (C), Guanine (G). Each three characters represent an amino acid, that then linked together with peptide bonds to form a protein as shown in Figure 1.4[6]. If the sequence of the above mentioned genes is altered, then the patient will suffer from AD.

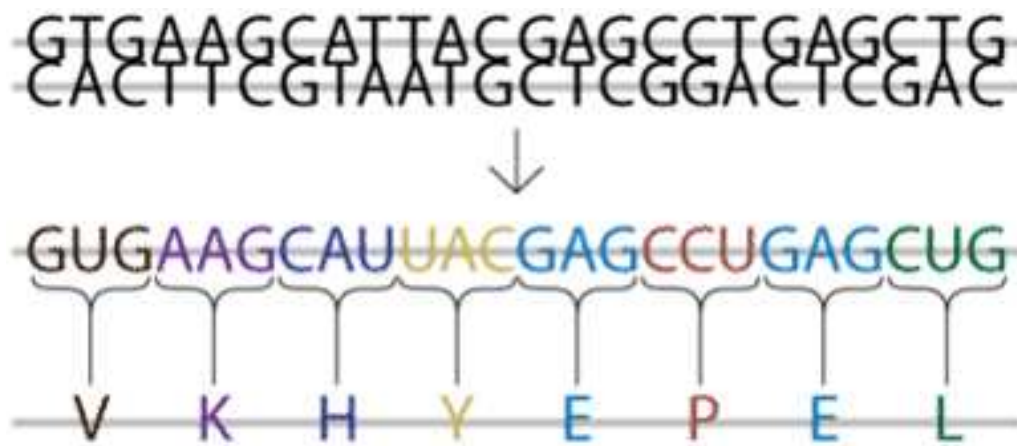


Figure 1.4: The whole genome sequenced DNA

1.1.1 DNA scanning types

There are two ways to screen the DNA, one by using WGS and the other one by analyzing the SNPs. WGS is the entire sequence of the human DNA, it retains all the information that is carried by the sequence. SNPs represents single precise positions in the DNA sequence that carry the information but not all detailed information since it is only one position in the sequence.

1.2 Motivation

The normal procedure to diagnose AD takes a period of 6 to 12 months with the patient going through multiple physical examinations , diagnostic tests, and brain scans. Moreover in order for ADNI (Alzheimer's Disease Neuroimaging Initiative) to detect the patients labeled as mild cognitive impairment (patients who are potential to have AD), they kept the patients under a clinical study for an initial study of five years which were later extended by two years [7] . However, the proposed desktop application can determine an initial diagnosis whether the patient is healthy, potential or diseased to help guide the physicians with how they will decide a treatment plan as soon as possible rather than taking multiple months.

1.3 Problem Definition

The existing automated systems of AD is either the patient is healthy, or the patient is in severe case of AD, which makes our main target is to show new phase which is a potential patient. Along with reducing the time taken by manual diagnosing to help guide the doctors with how they will decide a treatment plan as soon as possible rather than taking multiple months.

1.4 Project Description

Automated diagnosis for AD patients whether the patient is healthy, potential or diseased.

1.5 Scope

The system is developed to reveal if the patient is healthy or could have AD(potential) or diseased. Either way, this helps the patient and the doctors to diagnose the disease early in 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.

1.6 Project Overview

In the proposed system, we are implementing a system that will be able to accurately and swiftly diagnose AD patients and categorize them into three classes scrupulously: healthy patients, patients with a high risk of developing the disease or disease carrier. Moreover, Our approach starts with the collection of the patient's sample. Then they are analyzed and filtered in order to get the desired chromosomes and the particular genes locations that is needed to be able to properly diagnose the patient. The sample will then be compared with an SVR model that extracts the needed data from the datasets. So the system is able to classify the sample correctly. Then we use another model to determine patients with potential to get AD using Mini-Batch k-means.

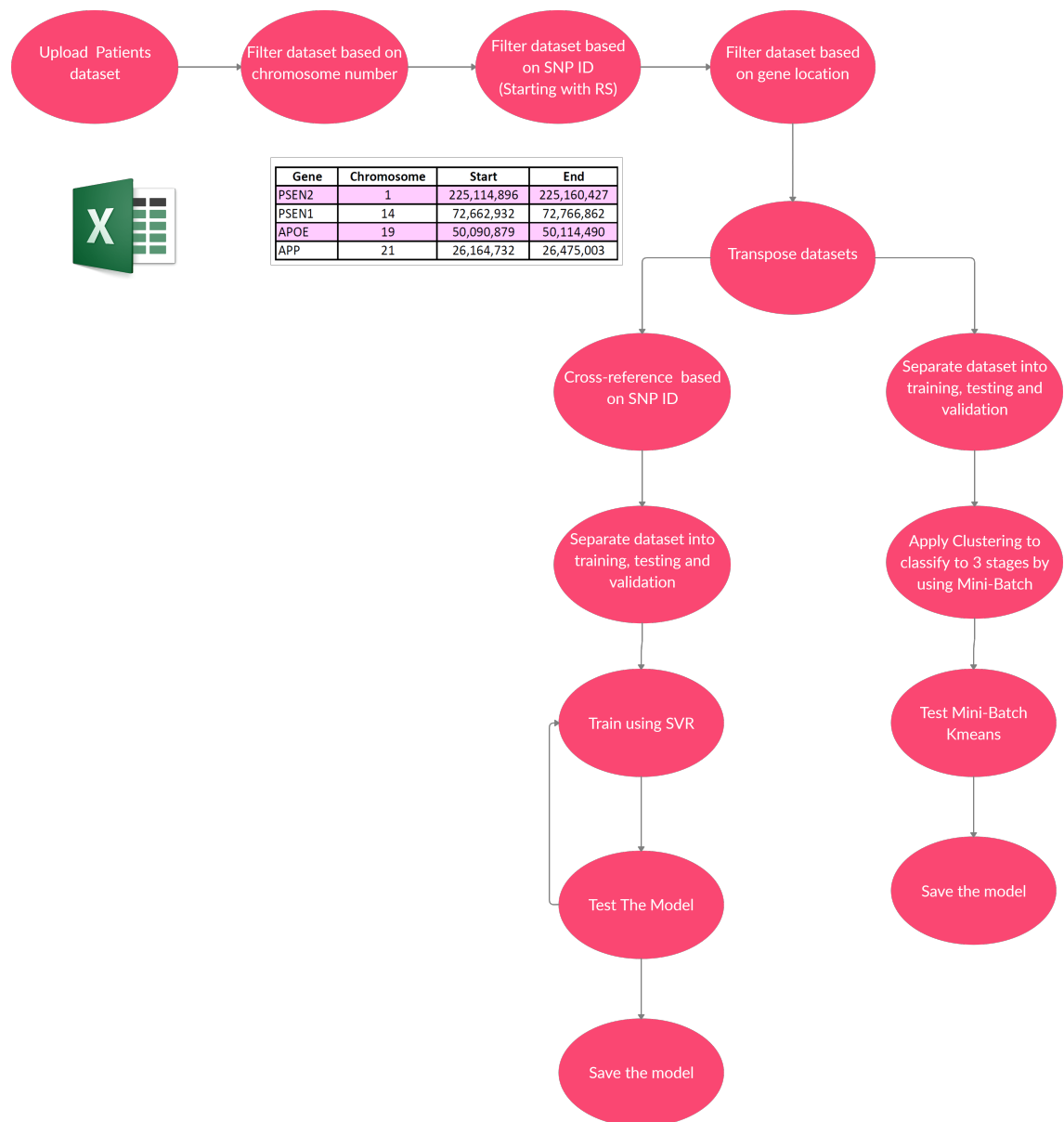


Figure 1.5: System Overview

1.7 Project Management and Deliverable

1.7.1 Tasks and Time Plan

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 1.6: Project Timeline

1.8 Abbreviations

Abbreviation	Meaning
WGS	Whole Genome Sequencing
SNP	Single Nucleotide Polymorphism
AD	Alzheimer's Disease
ADNI	Alzheimer's Disease Neuroimaging Initiative
APP	Amyloid precursor protein
PSEN-1	Presenilin-1 protein
PSEN-2	Presenilin-2 protein
APOE-4	Apolipoprotein
NB	Naive Bayes
CNN	Convolutional Neural Networks
RNA	Ribonucleic acid
DNA	Deoxyribonucleic acid
SVR	Support vector regression

Chapter 2

Literature Work

2.1 Related work

This section is divided into two parts, systems that worked with WGS datasets and systems that worked with SNPS datasets.

2.1.1 Whole Genome Sequence researches

They used the CNN as it classed the pairwise alignments of sequences into two classes. The main problem statement is to classify non-coding RNA sequences into positive and negative classes to prove it's classifying correctly. They reached accuracy of 94.5%.

Chatterjee et al[8] proposed this research to identify Cancer type through DNA Methylation. Their main issue was that the ways of detecting cancer are troublesome and the possibility of wrong positives exists. So they needed a method with higher accuracy to detect cancer types. The contribution that this research team accomplished was building a model that can learn the changing differentially methylated regions (DMRs) patterns to detect 32 cancer types. The model was able to attain an accuracy of 92.87% and it was based on 10,000 samples.

Bonvicini et al[9] proposed this research to recognize known uncommon variations in major candidate genes, related to Early Onset Dementia by sequencing the DNA. Their key problem was collecting data samples for people with uncommon and common variations in dementia-associated qualities. They examined 22 patients enrolled in Memory Clinics. In order to predict the functional consequences of non-synonymous variations, they exploited

eight different bioinformatics tools: SIFT, PolyPhen-2, FATHMM, phyloP, MutationTaster, LRT, and CADD and GERP++. This paper was beneficial on how to detect the mutations that happens inside DNA that causes early on set dementia.

Soh et al [10] presented a research to predict cancer type from tumor DNA signature. 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 6,640 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+-0.4%.

2.1.2 SNPs researches

Mostafa et al[11] presented a research to identify genetic biomarkers associated to Alzheimer's disease. This research has used the SNPS and searched in APOE gene to detect AD. They used two machine learning techniques to detect healthy people from diseased patients. They used Naive Bayes (NB) and K2 techniques which resulted with accuracy 98% and 98.40%.

Erdogan and Aydin[12] introduced this research that used a data mining method in order to help with the early diagnosis of AD. Since it's the most promising field that will help with making a diagnosis. The decision tree that they ended up using yielded an accuracy of almost 57% and had generated 26 rules from 38 SNPs. Although this model's accuracy isn't satisfying and needs improvements, It is a promising demonstration of how genome wide association studies can benefit from data mining approaches for the interpretation of the SNPs variations in disease models.

Li et al [13] presented a research where they collected 843 participants SNPs from ADNI and uploaded all of them to INTERSNP software to apply 2-marker model. They confirmed that the APOE, APOC1 (Apolipoprotein C1), and TOMM40 (Translocase of outer mitochondrial membrane 40) genes are very effective to detect AD. They also observed another additional 14 genes who also affects AD.

Chapter 3

System Requirements Specification

3.1 Introduction

3.1.1 Purpose of this chapter

The purpose of this software requirement document is to present a detailed description of 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.

3.1.2 Scope of this chapter

The system is developed to reveal 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.

3.1.3 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.

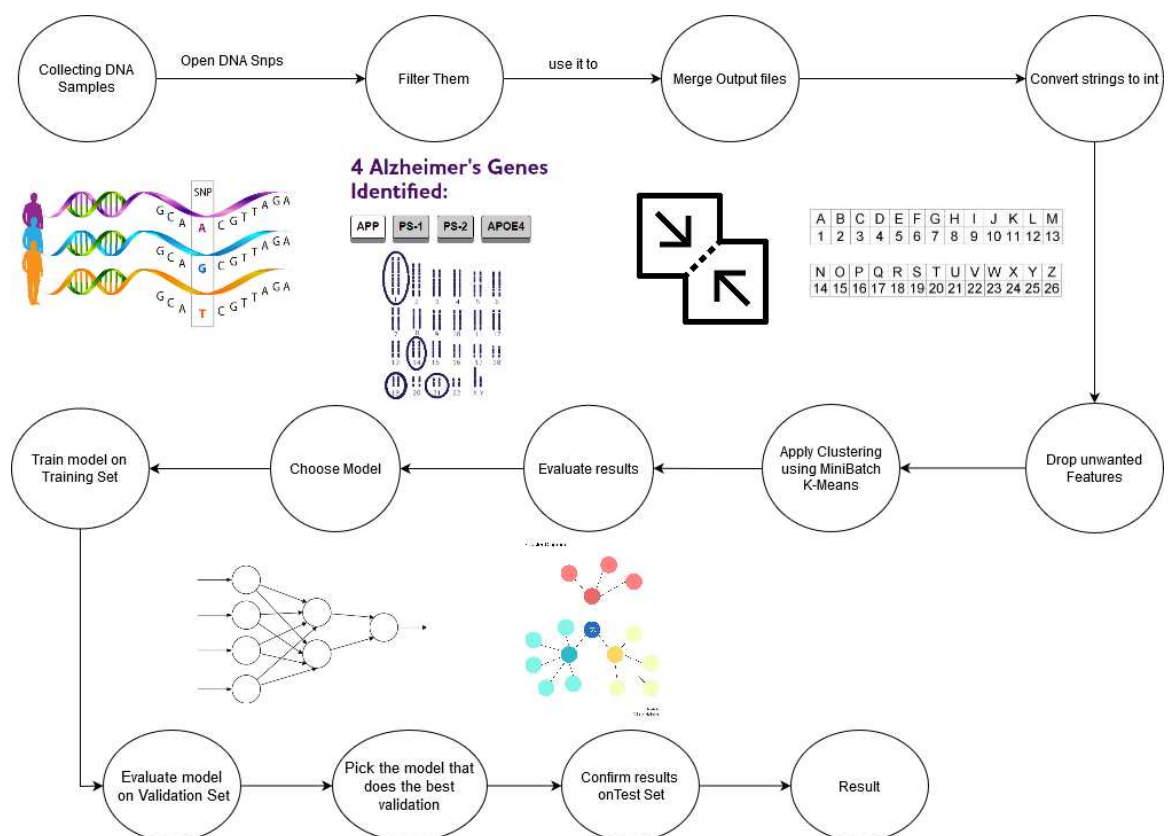


Figure 3.1: System Overview

3.1.4 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.

3.2 General Description

3.2.1 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.

3.2.2 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.

3.2.3 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.

3.2.4 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.

3.2.5 General Constraints

The uploaded file should carry the DNA not another content and it should be a 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.

3.3 Functional Requirements

3.3.1 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

3.3.2 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

3.3.3 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(.csv) in the directory and starts the preprocessing

3.3.4 View Result

Use Case Name	View Result
Input	N/A
Output	The system shows which stage the patient is in
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

3.3.5 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

3.3.6 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

3.3.7 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

3.3.8 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

3.3.9 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

3.3.10 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.

3.3.11 Cluster

Use Case Name	Cluster
Input	A csv file
Output	A csv file contains the results of clustering
Prerequisite	A csv files 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

3.3.12 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

3.3.13 Remover

Use Case Name	Remover
Input	A .txt file
Output	A .txt file
Prerequisite	Must extract 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

3.3.14 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

3.4 Interface Requirements

3.4.1 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 the 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.

3.4.1.1 GUI

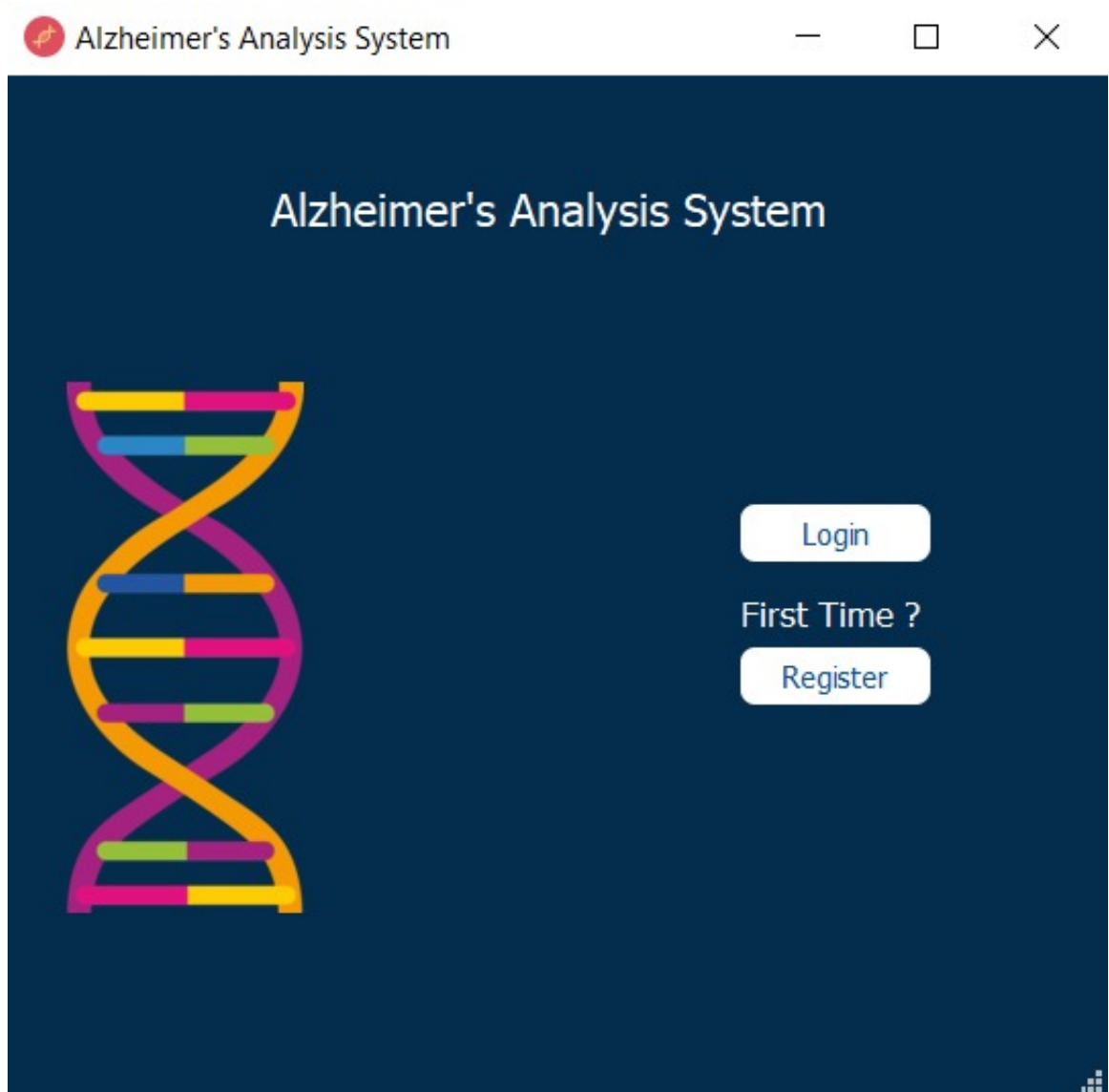
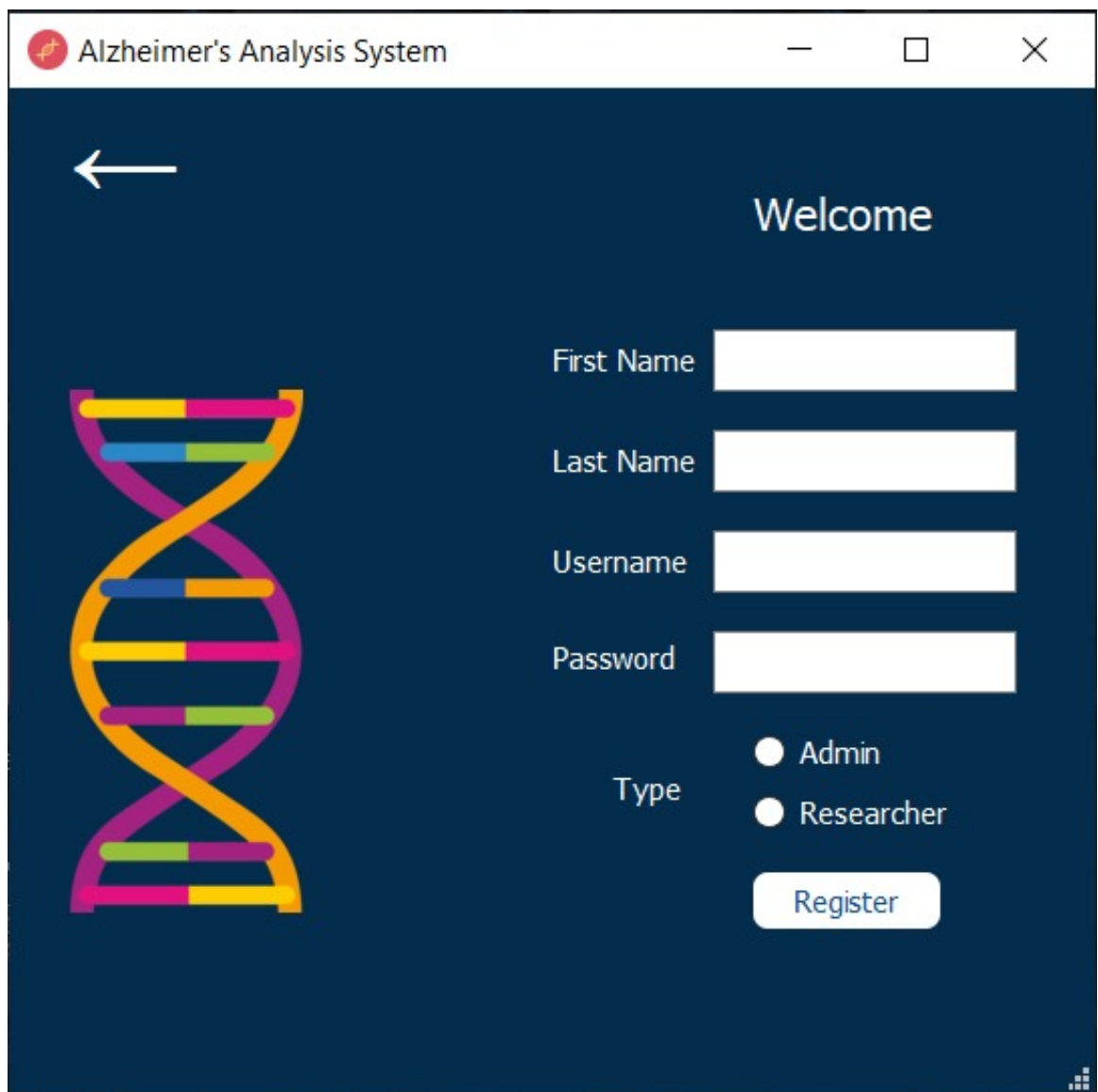


Figure 3.2: Home Screen



The screenshot shows a web application window titled "Alzheimer's Analysis System". The interface has a dark blue background. On the left, there is a large, colorful DNA double helix graphic. Above it is a white left-pointing arrow. On the right, the word "Welcome" is displayed in white. Below it are four white input fields for "First Name", "Last Name", "Username", and "Password". To the right of the "Password" field is a "Type" label with two radio button options: "Admin" and "Researcher". Below these options is a white "Register" button. In the bottom right corner, there is a small logo consisting of a grid of dots.

Alzheimer's Analysis System

←

Welcome

First Name

Last Name

Username

Password

Type

☐ Admin

☐ Researcher

Figure 3.3: Register Screen

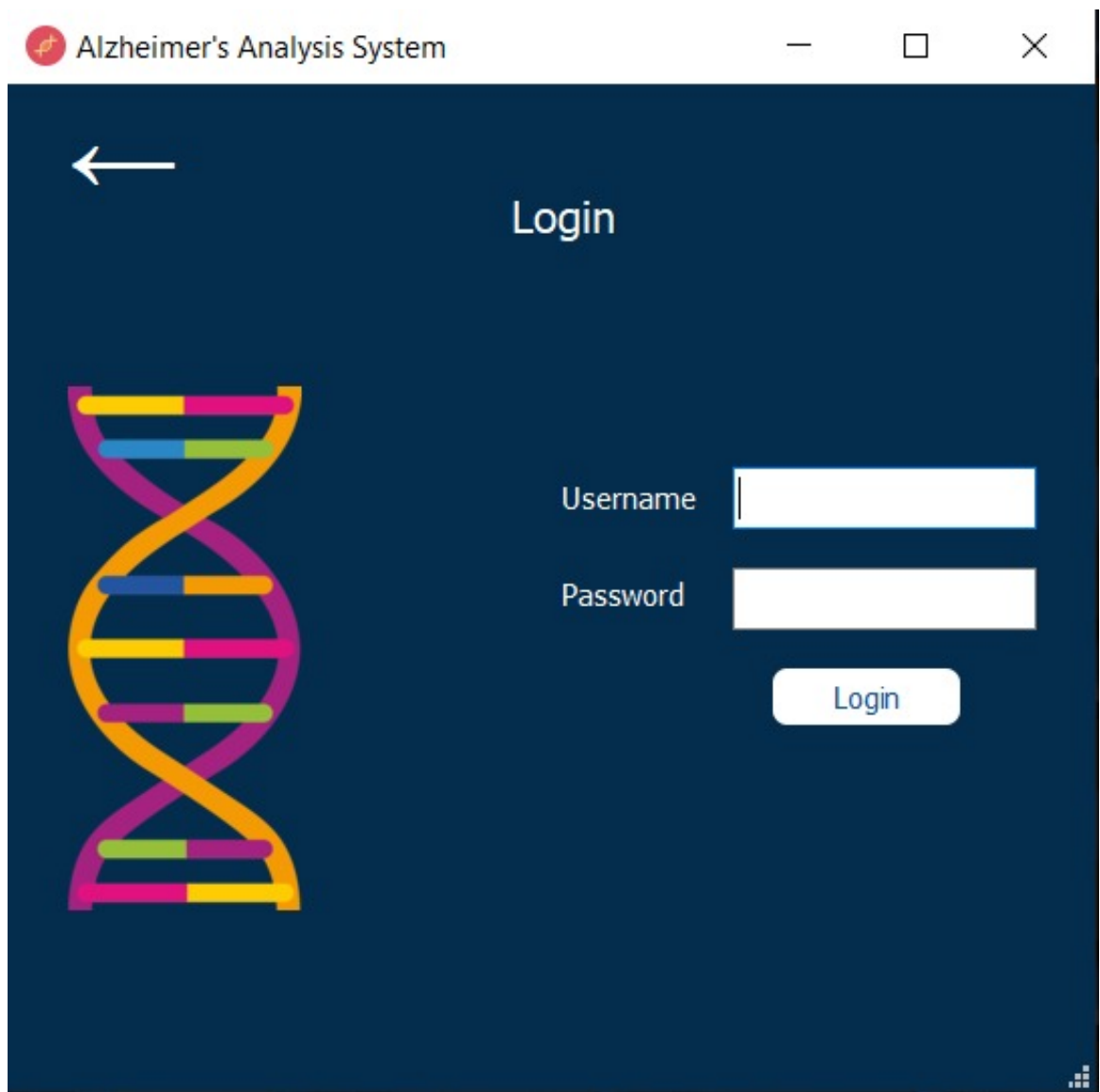


Figure 3.4: Login Screen

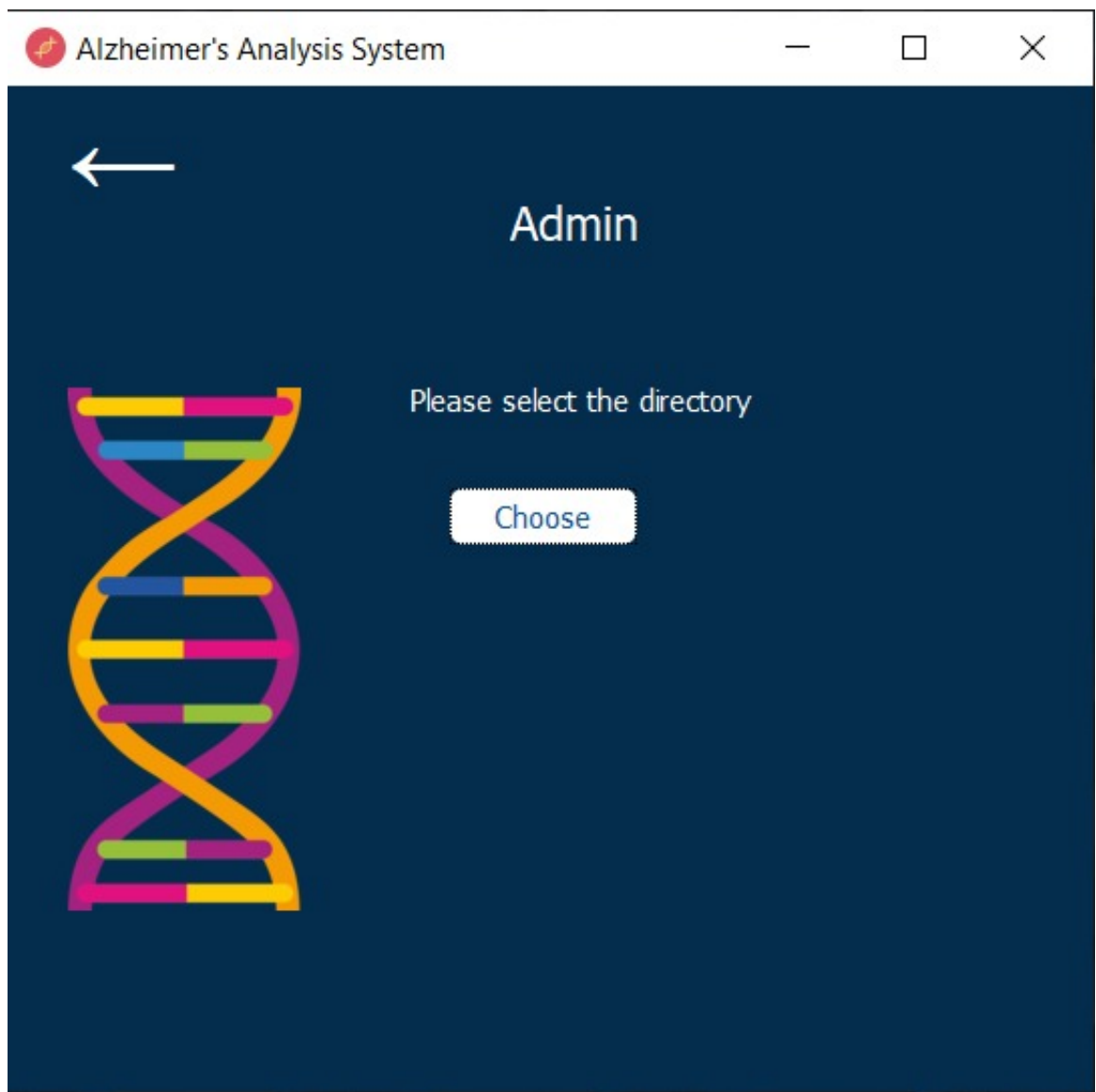


Figure 3.5: Admin Screen

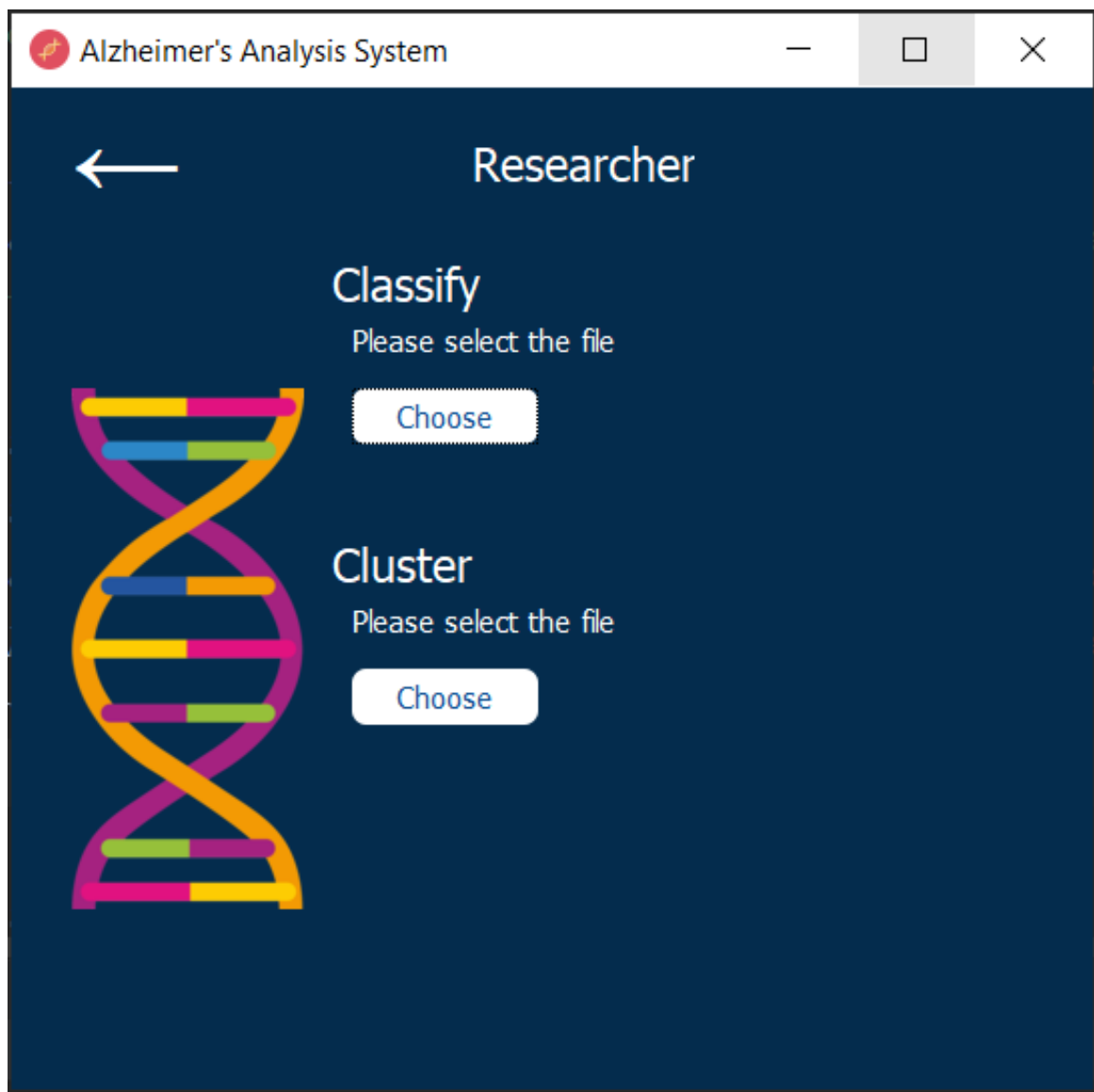


Figure 3.6: Researcher Screen

3.4.1.2 API

- Scikit-learn
- Tkinter
- Numpy
- Pyqt

- Firebase

3.4.2 Software Interfaces

N/A

3.4.3 Communication Interface

N/A

3.5 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.

3.6 Design Constraints

Due to the lack of professional computer skills for some users, the system needs to be user friendly to ease the process of doctors performing the required tasks.

3.6.1 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.

3.7 Other non-functional attributes

3.7.1 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.

3.7.2 Reliability

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

3.7.3 Portability

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

3.7.4 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

3.7.5 Maintainability

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

3.8 Preliminary Object-Oriented Domain Analysis

3.8.1 Inheritance Relationships

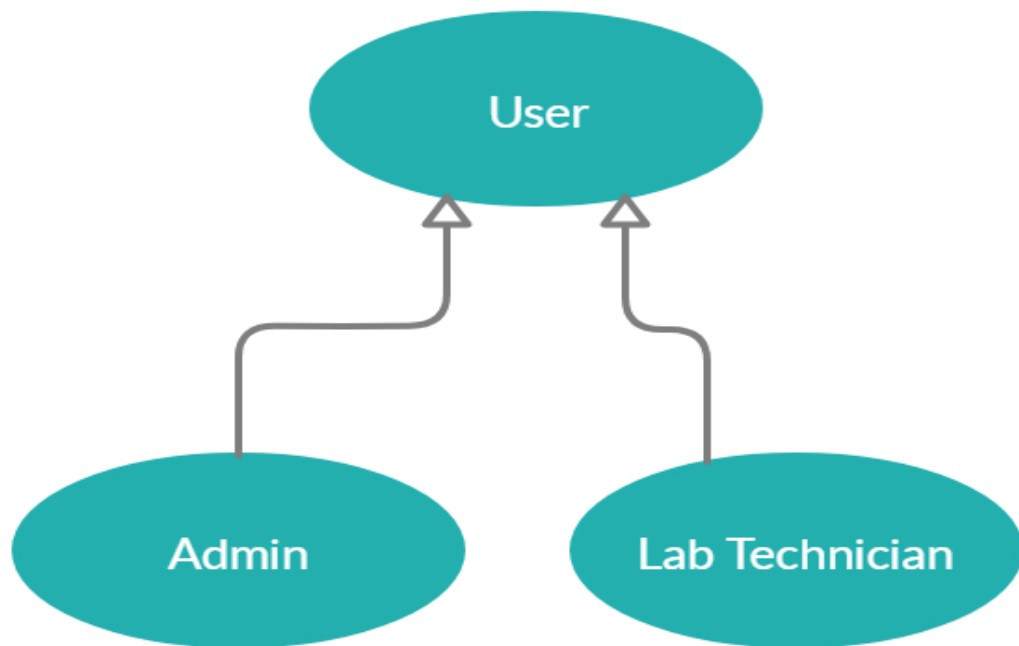


Figure 3.7: Inheritance Relationships

3.8.2 Class descriptions

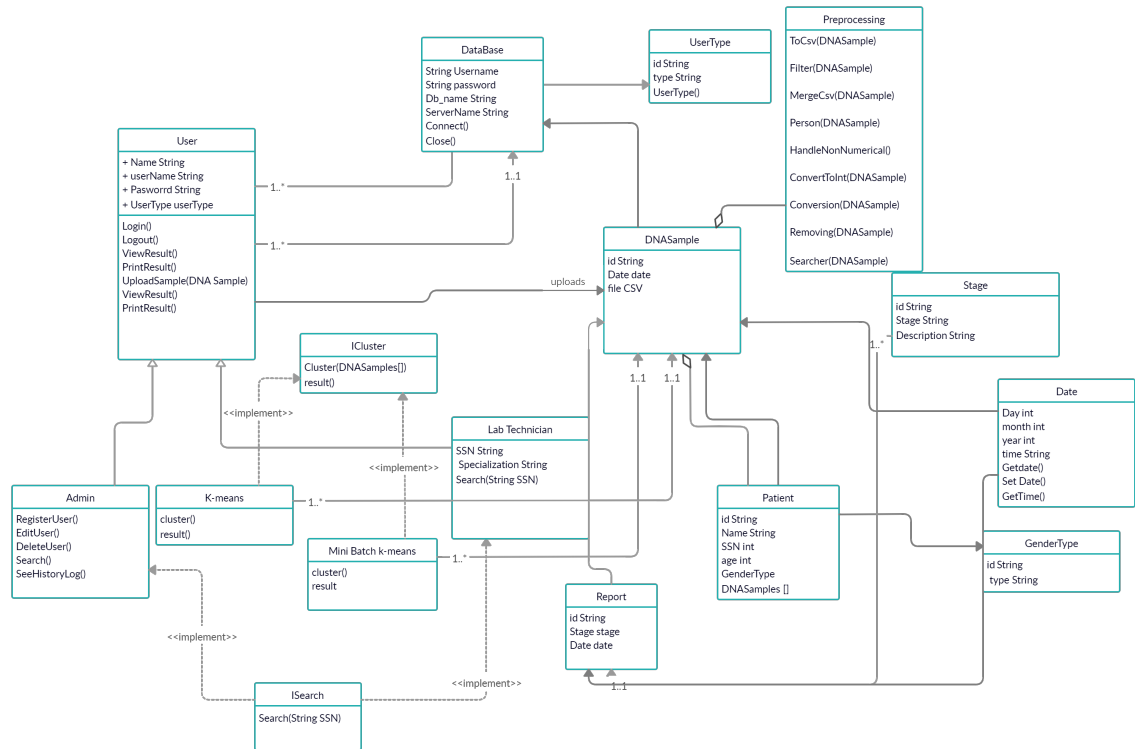


Figure 3.8: Class diagram

Each class description should conform to the following structure:

3.8.2.1 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 and user type.
7. Operations: Login, log out, upload samples, view results and print results.

3.8.2.2 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

3.8.2.3 Lab Technician

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

3.8.2.4 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.

3.8.2.5 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.

3.8.2.6 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.

3.8.2.7 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: N/A
7. Operations: Searcher, Removing, ToCsv, Filter, MergToCsv and Convert

3.8.2.8 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: N/A

3.8.2.9 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: N/A

3.9 Operational Scenarios

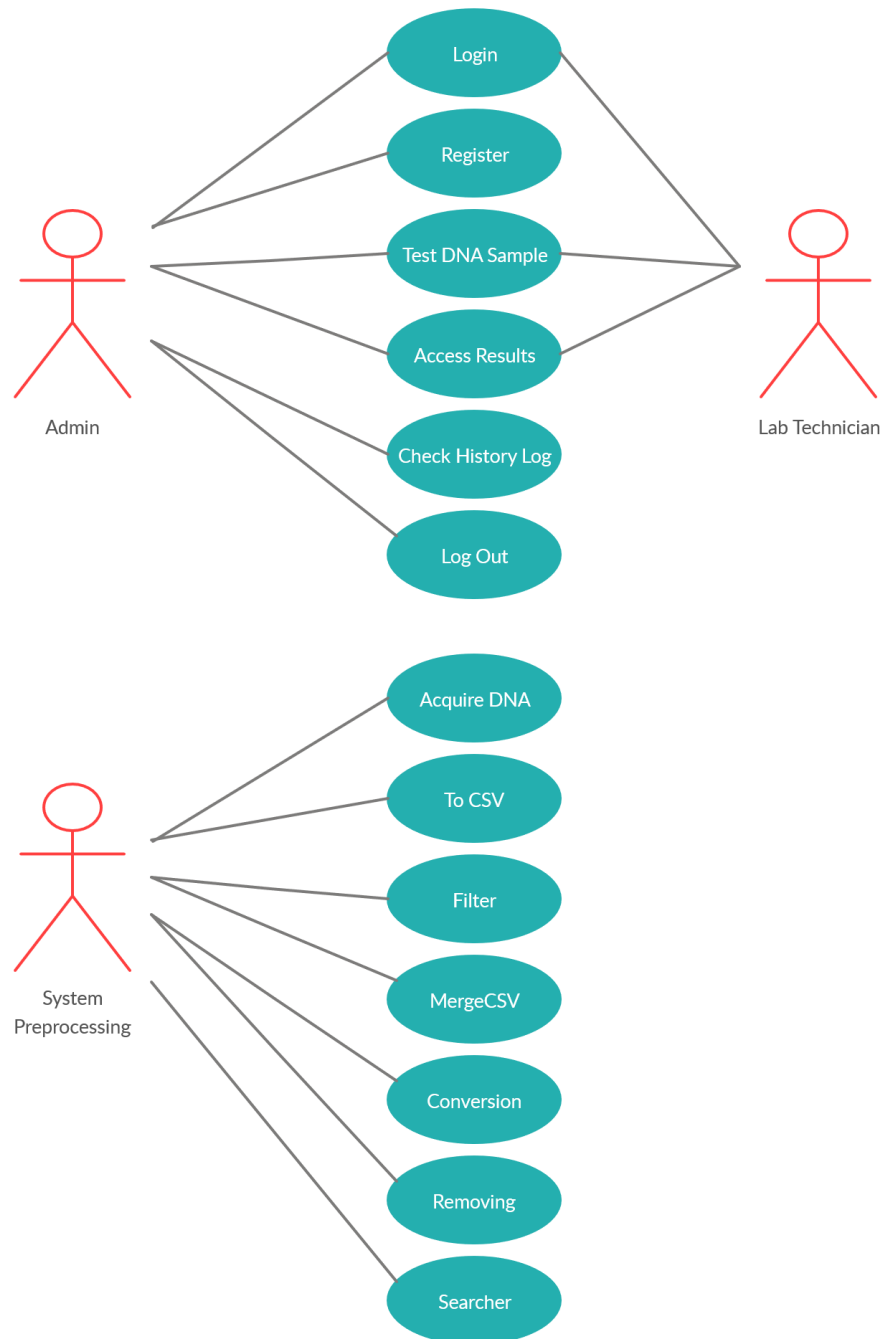


Figure 3.9: Use Case

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.

3.10 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 3.10: Project Timeline

3.11 Preliminary Budget Adjusted

N/A

3.12 Appendices

3.12.1 Definitions, Acronyms, Abbreviations

1. AD : Alzheimer's Disease.
2. SNP: Single Nucleotide Polymorphism.

Chapter 4

Software Design Document

4.1 Introduction

4.1.1 Purpose

This software design document describes and presents a detailed description of the Classification of 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.

4.1.2 Scope

The system is developed to reveal 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 to diagnose the disease early in 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.

4.1.3 Overview

This SDD document includes 8 main sections. The first section is an introduction to our system including our scope and purpose. The second section is the system overview illustrating our system work flow. The third section includes the architecture design of the system, activity diagram, sequence diagram and class diagram. The fourth section

illustrates the database design in details. The fifth section illustrates our component design including the used algorithms and techniques. The sixth section illustrates the human interface design and describes how the user will interact with our system. The seventh section is the requirement matrix that shows which components satisfy each of the functional requirements.

4.1.4 Definitions and Acronyms

Term	Definition
SDD	Software Design Document , used as the primary medium for communicating software design information.
Design Entity	An element of a design that is structurally and functionally distinct from other elements.
AD	AD Disease.
SVR	Support Vector Regression.
DNA	Deoxyribonucleic acid, a self-replicating material which is present in nearly all living organisms as the main constituent of chromosomes. It is the carrier of genetic information..

4.2 System Overview

In the application, we are implementing a system that will be able to accurately and swiftly diagnose AD patients and categorize them into two classes scrupulously: healthy patients and patients with a high risk of developing the disease or disease carrier. Moreover, if the patient turns out to have AD we group them into three classes those being severe cases, moderate and mild. Our approach starts with the collection of the patient's sample. Then they are analyzed and filtered in order to get the desired chromosomes and the particular locations that is needed to be able to properly diagnose the patient concluding our preprocessing of the sample. The sample will then be compared with a model prebuilt already using a SVR that extract the needed features from the datasets. so it can be able to classify the sample correctly after this process if indeed the patients sample is positive for AD it'll be compared with another model to be able to determine the severity of the disease in this specific sample. The system overview is shown in Figure 4.1.

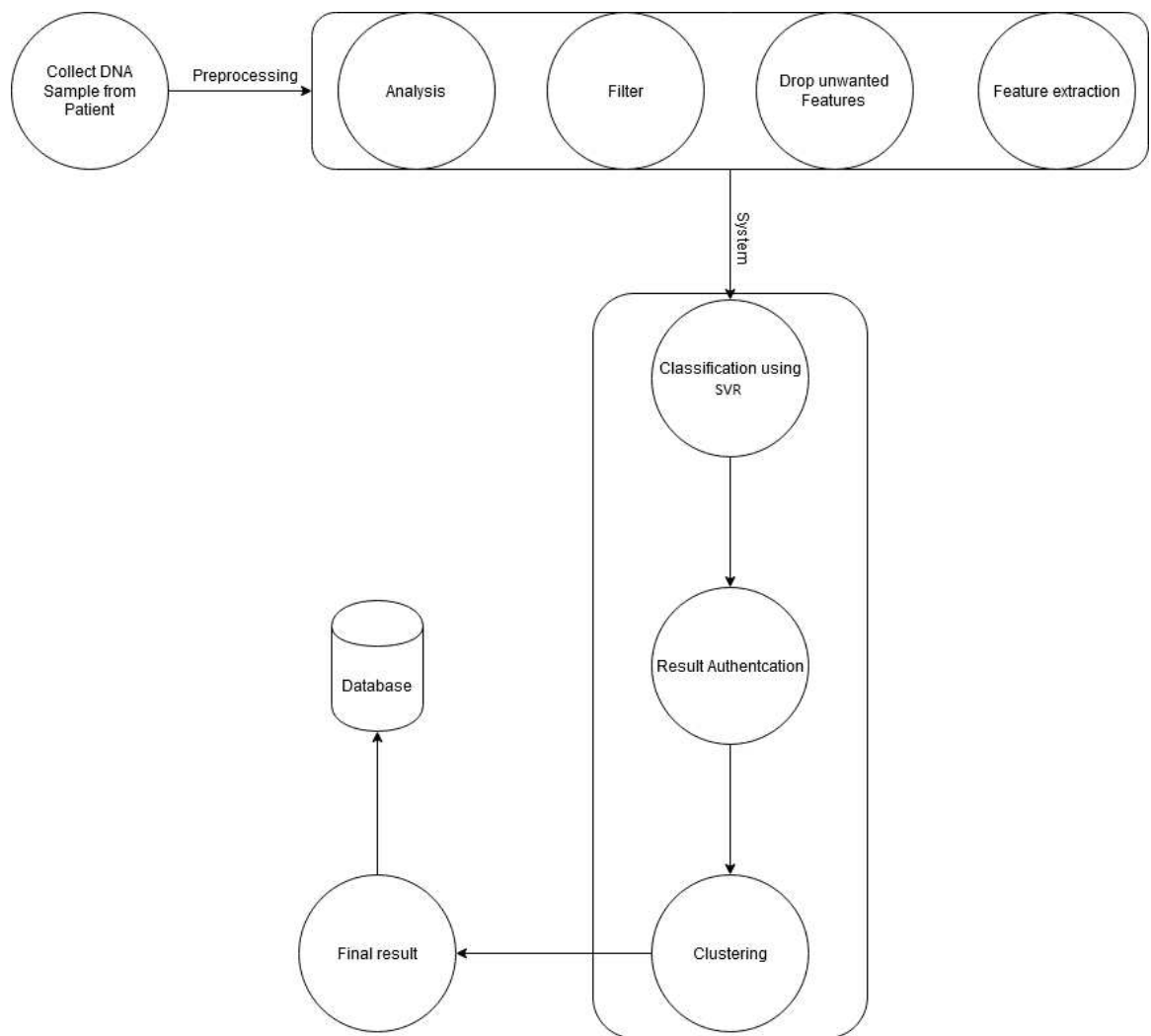


Figure 4.1: System Overview

4.3 System Architecture

4.3.1 Architectural Design

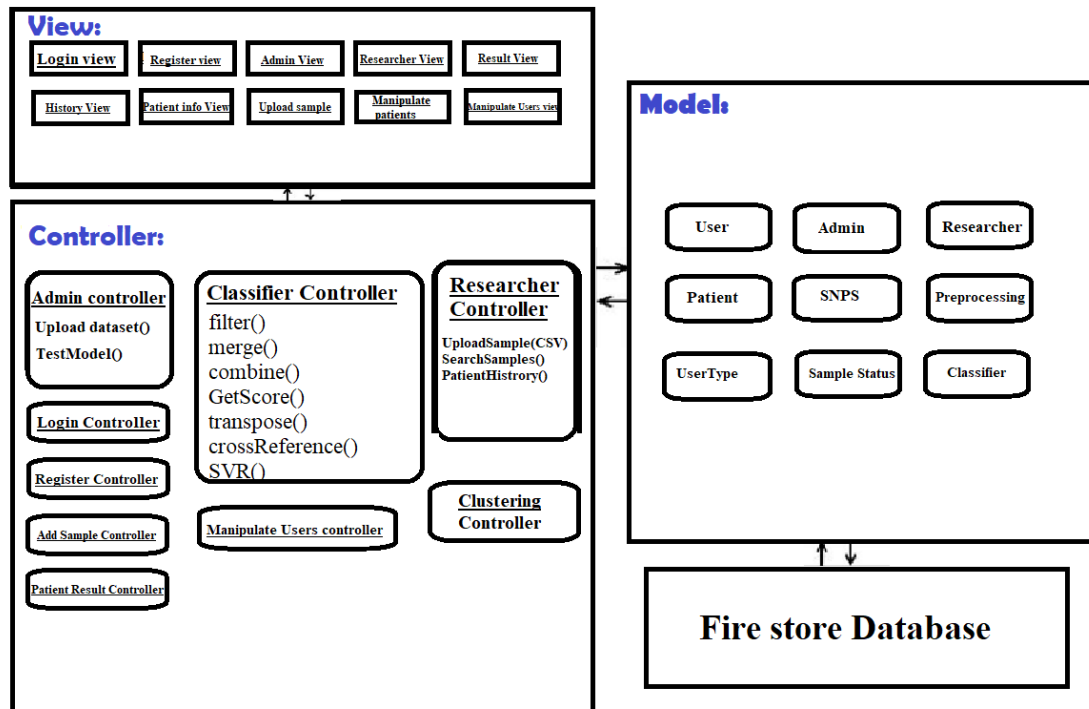


Figure 4.2: Architecture Diagram

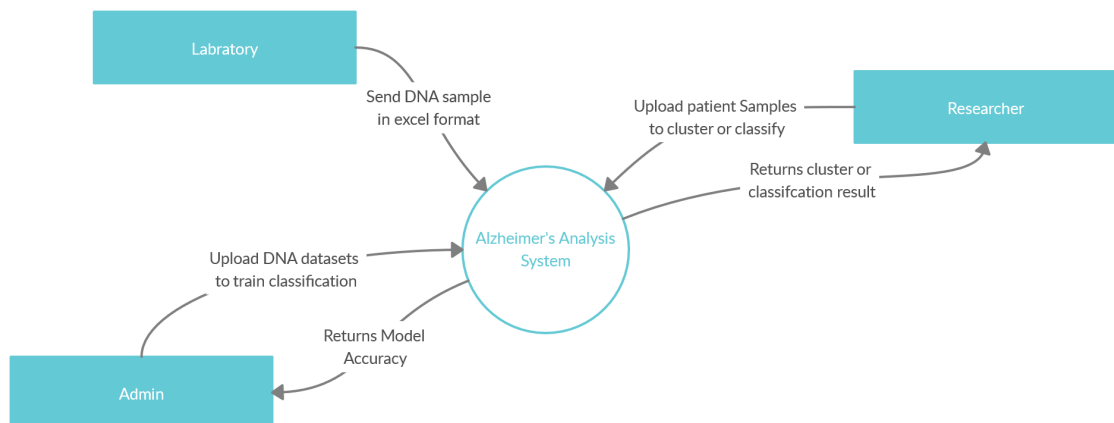


Figure 4.3: Context Diagram

4.3.2 Decomposition Description

4.3.2.1 Class Diagram

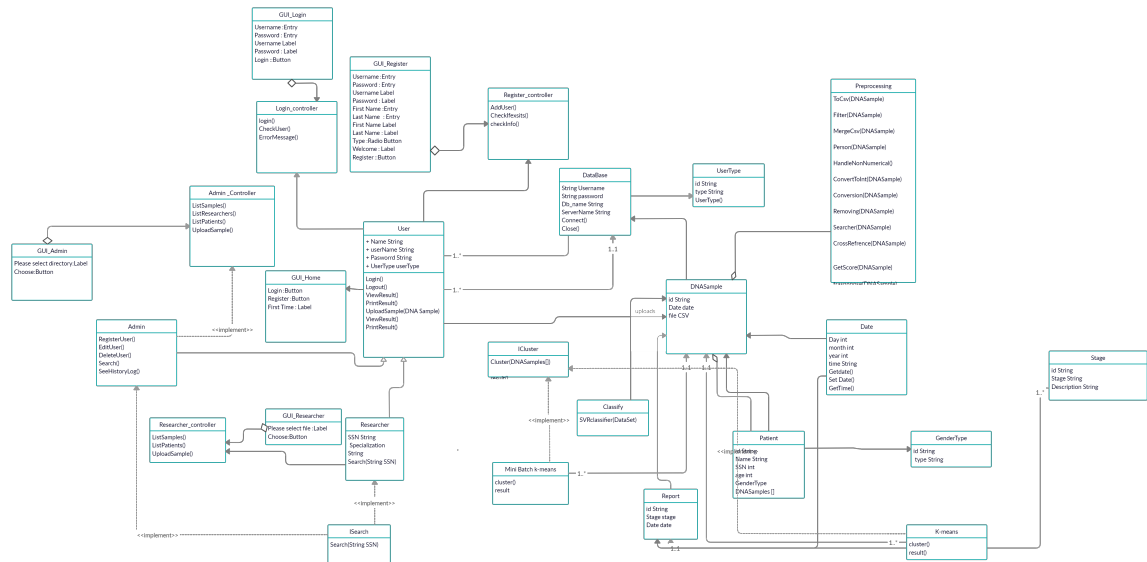


Figure 4.4: Class Diagram

4.3.2.2 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.

4.3.2.3 UserType

1. Type: concrete.
2. List of super classes: None.

3. List of sub classes: None.
4. Purpose: To allow scalability to add new user types.
5. Collaboration: This class is aggregated by class User, UserTypeAttribute and Report.
6. Attributes: id, TypeName, options[].
7. Operations: None.

4.3.2.4 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

4.3.2.5 Reseracher

1. Class Name: Reseracher
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.

4.3.2.6 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.

4.3.2.7 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.

4.3.2.8 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.

4.3.2.9 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

4.3.2.10 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: N/A

4.3.2.11 ISearch

1. Class Name: ISearch

2. Super Classes: N/A

3. Sub Classes:none
4. Purpose:To allow searching with different Criteria.
5. Collaborations:classes (Admin, Researcher) implements this class
6. Attributes:name,id
7. Operations:Search (String userName).

4.3.2.12 Classify

1. Class Name: Classify
2. Super Classes: N/A
3. Sub Classes:N/A
4. Purpose: To allow classification with different classifiers strategies
5. Collaborations: Class SVR
6. Attributes: none.
7. Operations: Classify(Parameters []).

4.3.2.13 LoginController

1. Class Name: LoginController
2. Super Classes: N/A
3. Sub Classes:N/A
4. Purpose: to control the Login view.
5. Collaborations:this class aggregated by LoginView
6. Attributes: none.
7. Operations:ControlUserLogin (String username, String pass)

4.3.2.14 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.

4.3.2.15 Activity Diagram

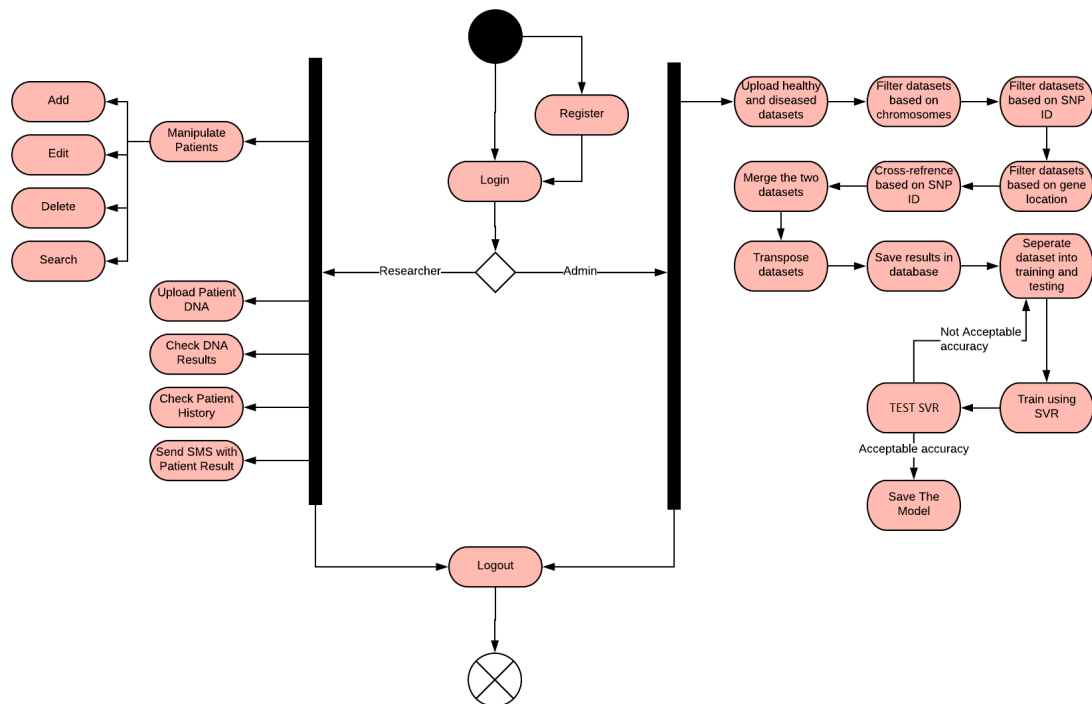


Figure 4.5: Activity Diagram

4.3.2.16 Admin Sequence Diagram

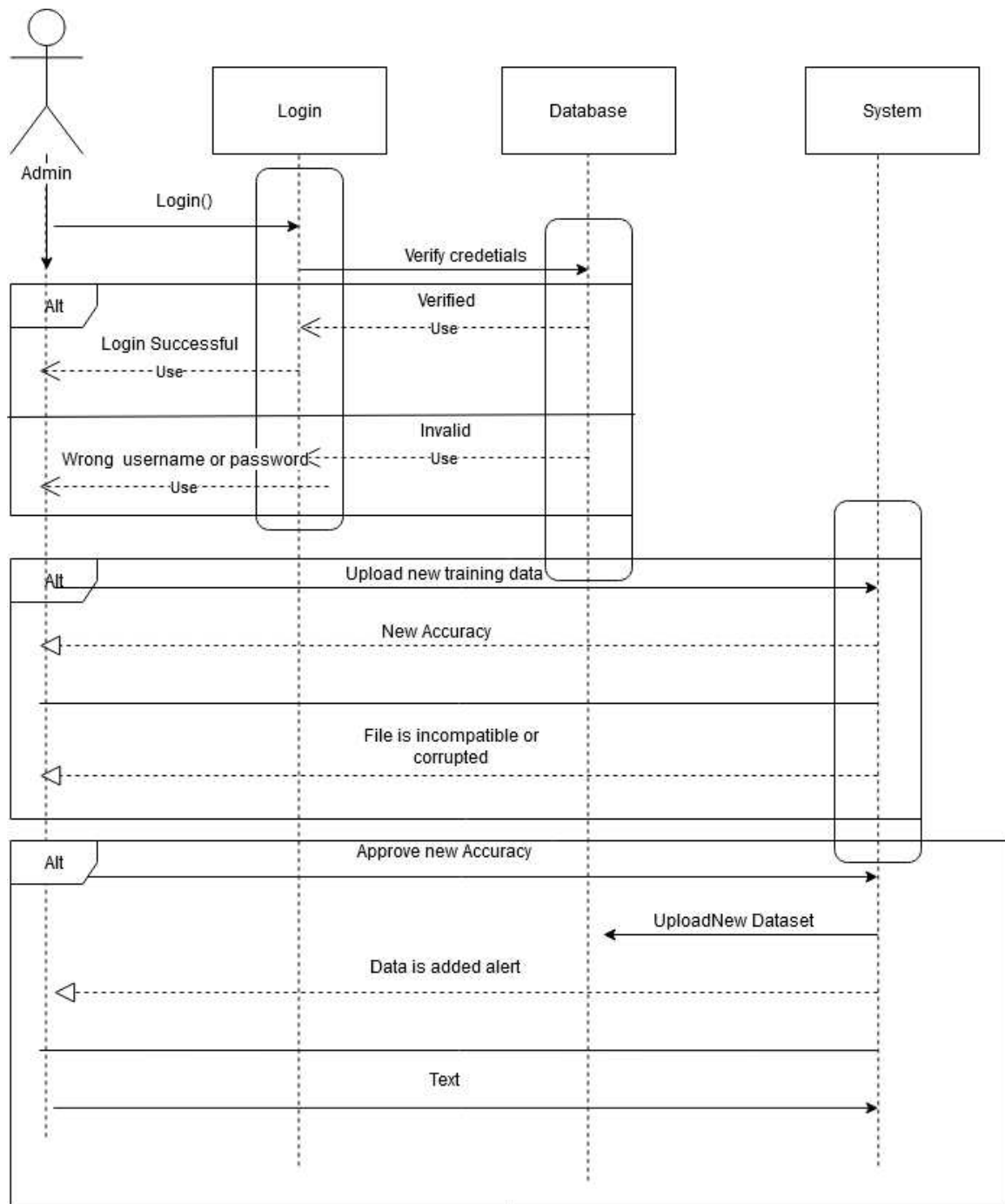


Figure 4.6: Admin Sequence Diagram

Figure 4.6 displays the sequence of the process of how the system admin can retrain the model in order to further improve the accuracy of the existing model the process start with the admin logging in then the admin will upload the new data that will be appended to the existing dataset so the system will check first if the file is in the same format before appending it then it'll retrain the model and compare its accuracy with the existing one if the accuracy is indeed higher a model will be created to replace the current one if not the new model and dataset will be discarded.

4.3.2.17 Researcher Sequence Diagram

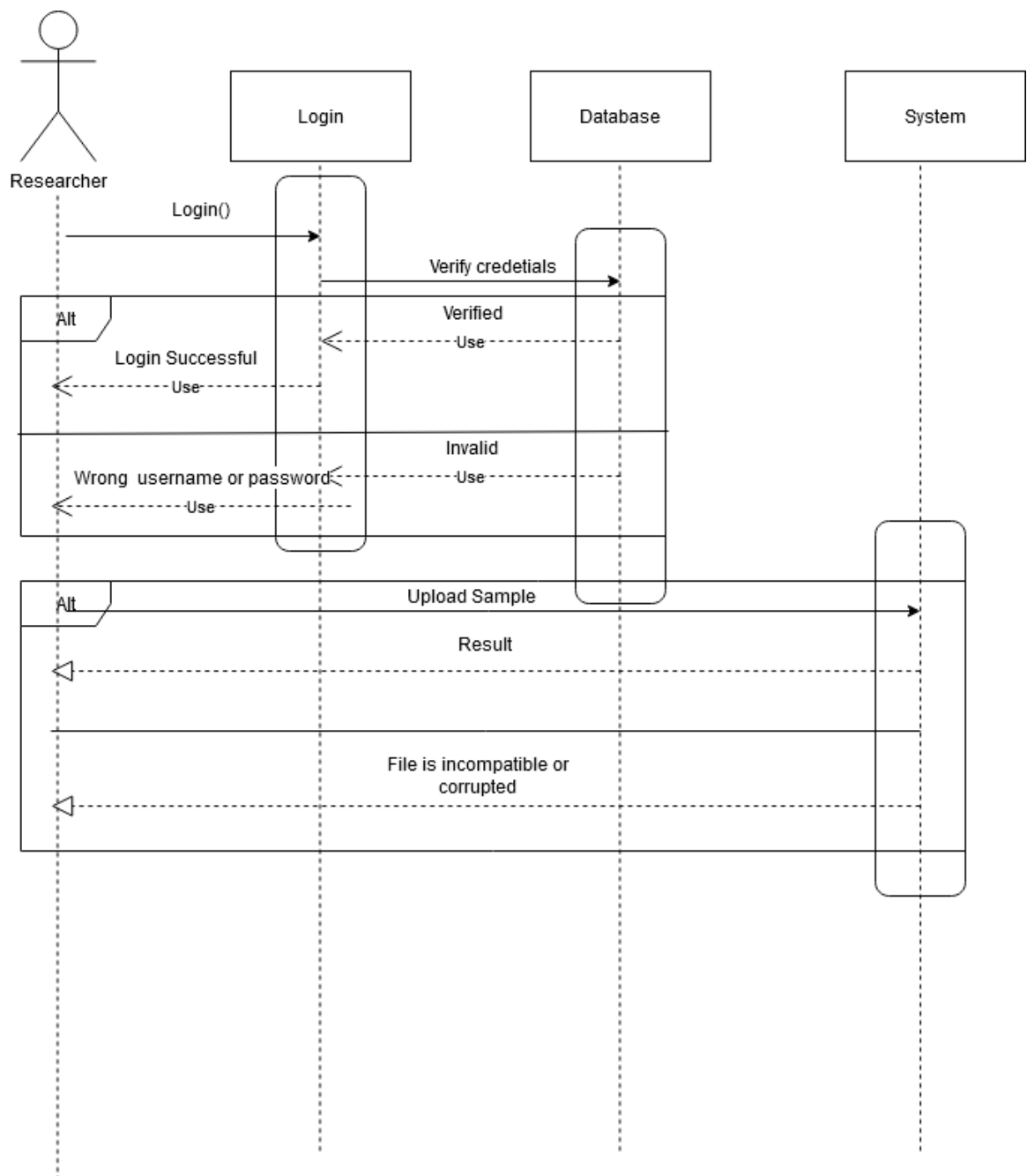


Figure 4.7: Researcher Sequence Diagram

In figure 4.7 the processes of how a researcher can check if the patient has AD or not is clarified. It is clarified by uploading the DNA file to the system after logging in the

researcher and then use the existing model to get a result that is after the system checks the file to make sure its compatible.

4.3.3 Design Rationale

Concerning the architecture of the system, the model view controller architecture will be the best due to its many advantages over other architectures. For example, the development of the application becomes faster and simple , it also simplifies for multiple developers to collaborate and work together on the same project since the system is divided into separate parts. Furthermore it makes it easier to apply updates in the application since only the component or the part that requires updating is accessed.

In order to classify the patients, support vector regression (SVR) was used [14]. This algorithm works by parsing through the given dataset and mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. This algorithm was chosen after trying out numerous algorithms that may fit the needs of our project however SVR turned out to be the best one overall regarding its complexity and its accuracy. There were a handful of other algorithms that were experimented with like Decision Tree, Random Forest , Grid search (Linear , RBF), Random Search (Linear , RBF) , Naïve Bayes , Logistic Regression , Deep Learning , Generalized Linear Model, Fast Large Margin , and Gradient Boosted Trees.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision.

Random forest is similar to the Decision tree model since the random forest is comprised of a number of Decision trees instead of just one When training, each tree in a random forest learns from a random sample of the data points. With each tree training slightly different on different observation therefore producing a well-balanced prediction derived from the average of the all the trees that were trained however the abundance of trained trees may

produce noise.

Grid Search and Random Search are algorithms that are given models and data and they each parse through the parameters that can be passed to the models in order to produce the best possible results that can be produced given that particular model however each algorithm parse through the data in a different way. Grid searching works by trying all the different combinations of parameters by iterating through them thus producing great results however can be computationally expensive while random search works by trying out sets of random combinations of parameters in order to find the best possible model while optimizing the parameters by evaluating the model at random configuration points. Random search can produce better results because of its random method of trying out random parameters and can have a lower complexity when compared to Grid Search.

Logistic regression does not try to predict the value of a numeric variable given a set of inputs. Instead, the output is a probability that the given input point belongs to a certain class. It is primarily used in binary classification.

Deep learning works by training an AI that if given a set of inputs can predict the output the AI consists of 3 types of layers (input, hidden, output) with each layer having a number of neurons inside it the connections between neurons represents the weight of each input and based on this weight the neuron decides what to do with the given output through an activation function inside each neuron.

Generalized Linear Models are an extension of traditional linear models. This algorithm fits generalized linear models to the data by maximizing the log-likelihood. The elastic net penalty can be used for parameter regularization. The model fitting computation is parallel, extremely fast, and scales extremely well for models with a limited number of predictors with non-zero coefficients.

Fast Large Margin (FLM) is similar to normal linear SVM algorithms in the sense that it tries to map the data and inputs onto a linear plane in order to properly classify the inputs however FLM is designed to work on huge datasets with millions of records.

Gradient Boosted Trees is a tree-based learning model that trains many models in a gradual, additive and sequential manner. Predictions of the final ensemble model is therefore the weighted sum of the predictions made by the previous tree models that were previously built therefore producing a model based on the best fitted trees trained during the run time.

However due to the nature of the dataset the models built with the previously mentioned classifiers ended up overfitting and produced unrealistic accuracy in almost all attempts with results ranging from 98-100 % therefore the SVR model was chosen since it had the most realistic accuracy of almost 92% and it was also one of the most computationally efficient algorithms that were tested.

4.4 Data Design

4.4.1 Data Description

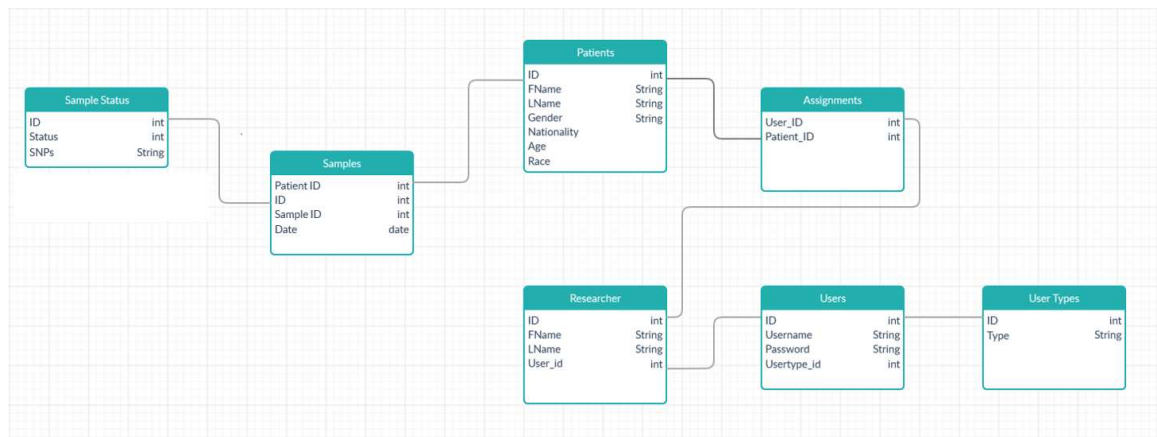


Figure 4.8: Database Diagram

4.4.2 Data Dictionary

4.4.2.1 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. The password of every system user is hashed by using sha324 function.

4.4.2.2 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.

4.4.2.3 Portability

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

4.4.2.4 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

4.4.2.5 Maintainability

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

4.5 Component Design

In this phase we prepare the DNA sample for the classifier, first we check if the format of the file is csv or not, then we filter the data by locating only the desired locations based on the chromosome numbers and locations. Furthermore, the system selects only the SNP names that starts with "rs", drops any rows with null values, drops any other columns that are not important and finally we check if the data is all numeric or not if not. We calculate a score by comparing the ref column with Allele1- forward and Allele2-forward. If both of them are equal to ref column, we represent it by 0 if only one of them matches with the ref column their score will be represented by 1 and if both of them don't match, the ref column their score will be represented by 2.

4.5.0.1 SVR

In details, SVR works by using a hyper-plane and two boundary lines also known as thresholds. In order to properly build an accurate model that fits the data properly, the hyper-plane is the line that separates the different classes in the data while the boundaries are the error threshold that is allowed the goal is to create a model that has a hyper-plane and boundaries that definitively covers most of our data points. Furthermore, the boundary lines should be equidistant from the hyper-plane meaning that the model should not favor a class more than the other in other words if represented by linear equation that lines for the boundaries and the hyper-plane should be $wx+b=0$ for the hyper-plane and $Wx+b=+e$, $Wx+b=-e$ for the boundary lines so that the e is the same for both lines creating the margin that'll become the model that fits the data.

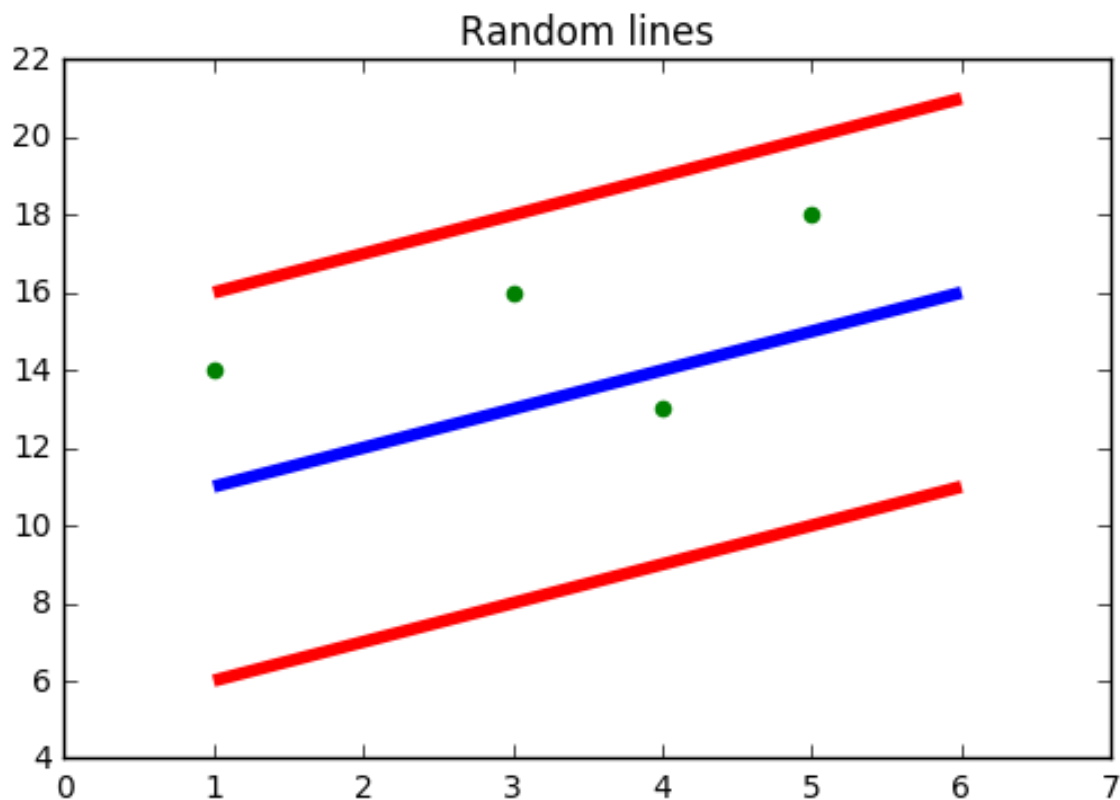


Figure 4.9: SVR

Where the red lines represent the boundaries and the red line represent the hyper-plane.

4.6 Human Interface Design

4.6.1 Overview of User Interface

Our system is a desktop application. It's user interface is very applicable and usable. You can login whether you are an admin or a researcher. The login screen directs you to different screens depends on the user type. System admins have some functions such as uploading data and testing the model. While researchers are responsible for dealing with patients. Illustration for the whole system is shown in Section 6.2.

4.6.2 Screen Images

First we have the home Screen with two buttons login and register shown in Figure 4.10. If login button is pressed, it will direct the user to the login screen in Figure 4.12. The first field is to enter username and second is for the password. If you logged in as an admin you will access the screen in Figure 4.13 and if you logged in as a researcher you will access screens in Figure 4.14 Both the admin and the researcher can upload a DNA sample to be tested and both can view the patients history. The admin can also upload a new dataset that can be appended to the existing one in order to retrain the model and get a higher accuracy than before. If the register button is pressed the user will be directed to the register screen shown in Figure 4.11. The user will have to enter their first name, last name, unique username, password and at last choose a user type either admin or researcher and after the registration will be directed to the login screen shown in Figure 4.12.

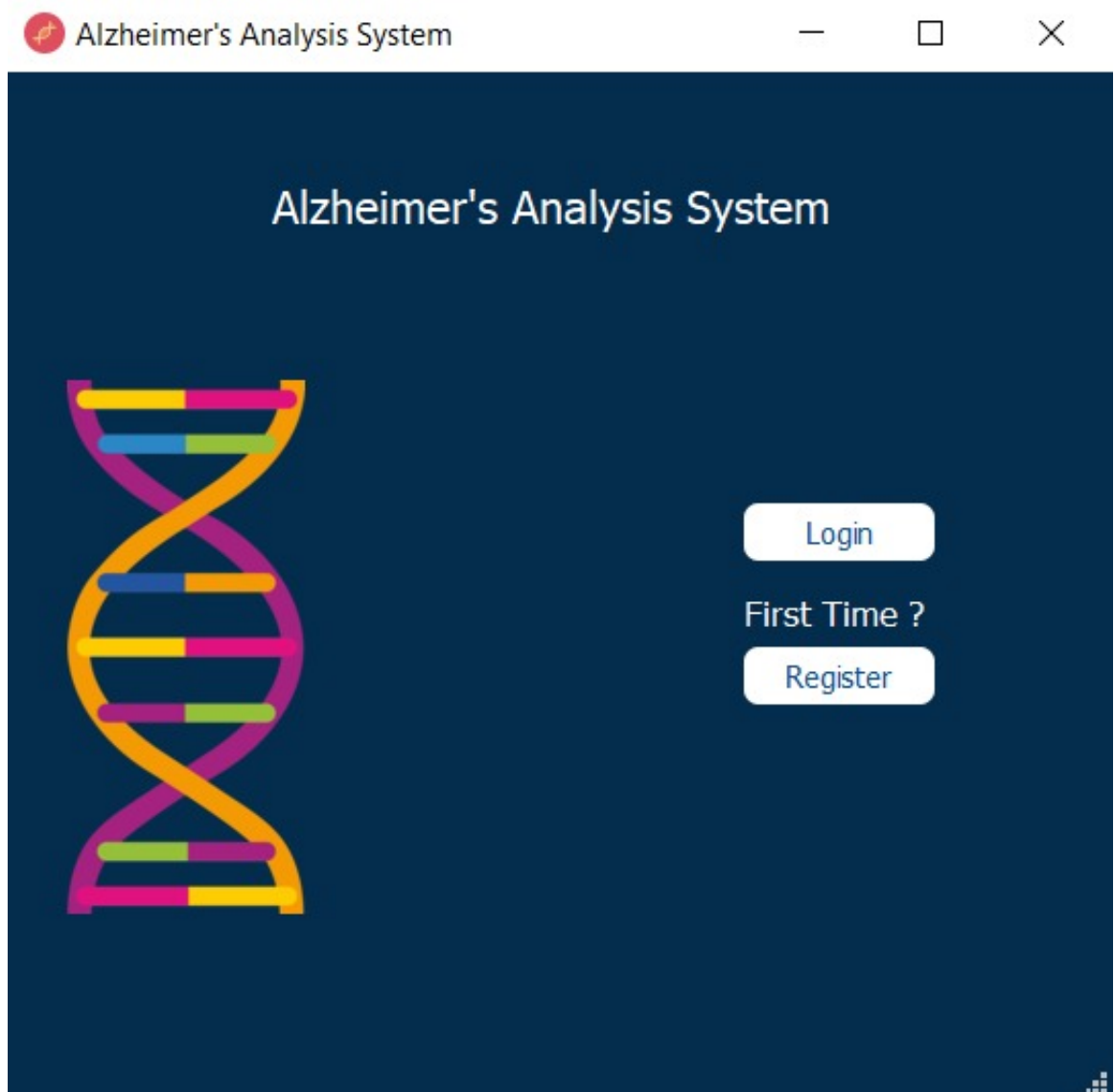
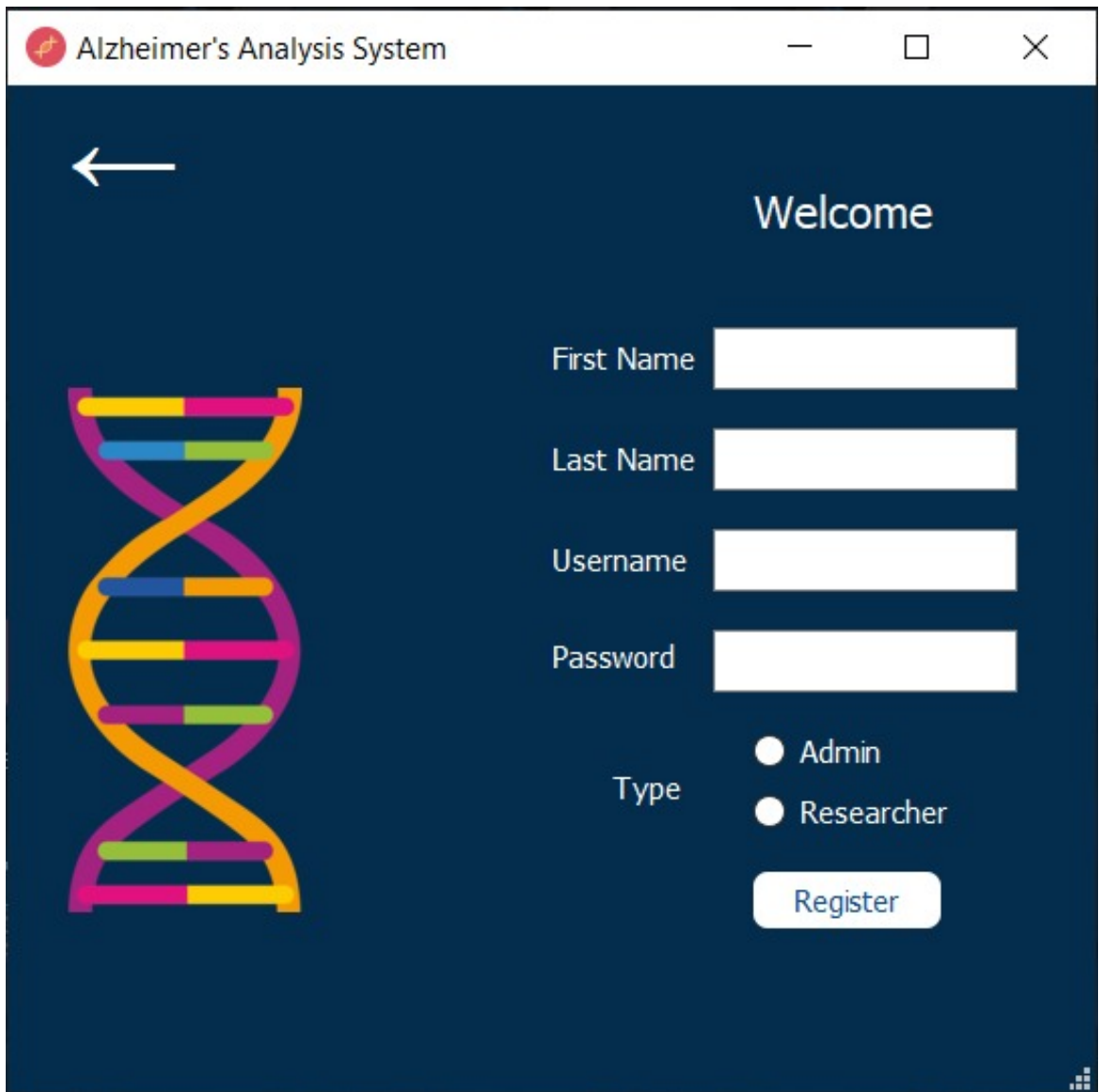


Figure 4.10: Home Screen



The image shows a web application window titled "Alzheimer's Analysis System". The window has a dark blue background. On the left side, there is a large, colorful DNA double helix graphic. Above the DNA graphic is a white left-pointing arrow. On the right side, the word "Welcome" is displayed in white. Below "Welcome", there are four white input fields for "First Name", "Last Name", "Username", and "Password". Below these fields, there is a "Type" label followed by two radio button options: "Admin" and "Researcher". At the bottom right, there is a white "Register" button. The window also features standard window controls (minimize, maximize, close) in the top right corner.

Alzheimer's Analysis System

←

Welcome

First Name

Last Name

Username

Password

Type

☐ Admin

☐ Researcher

Figure 4.11: Register Screen

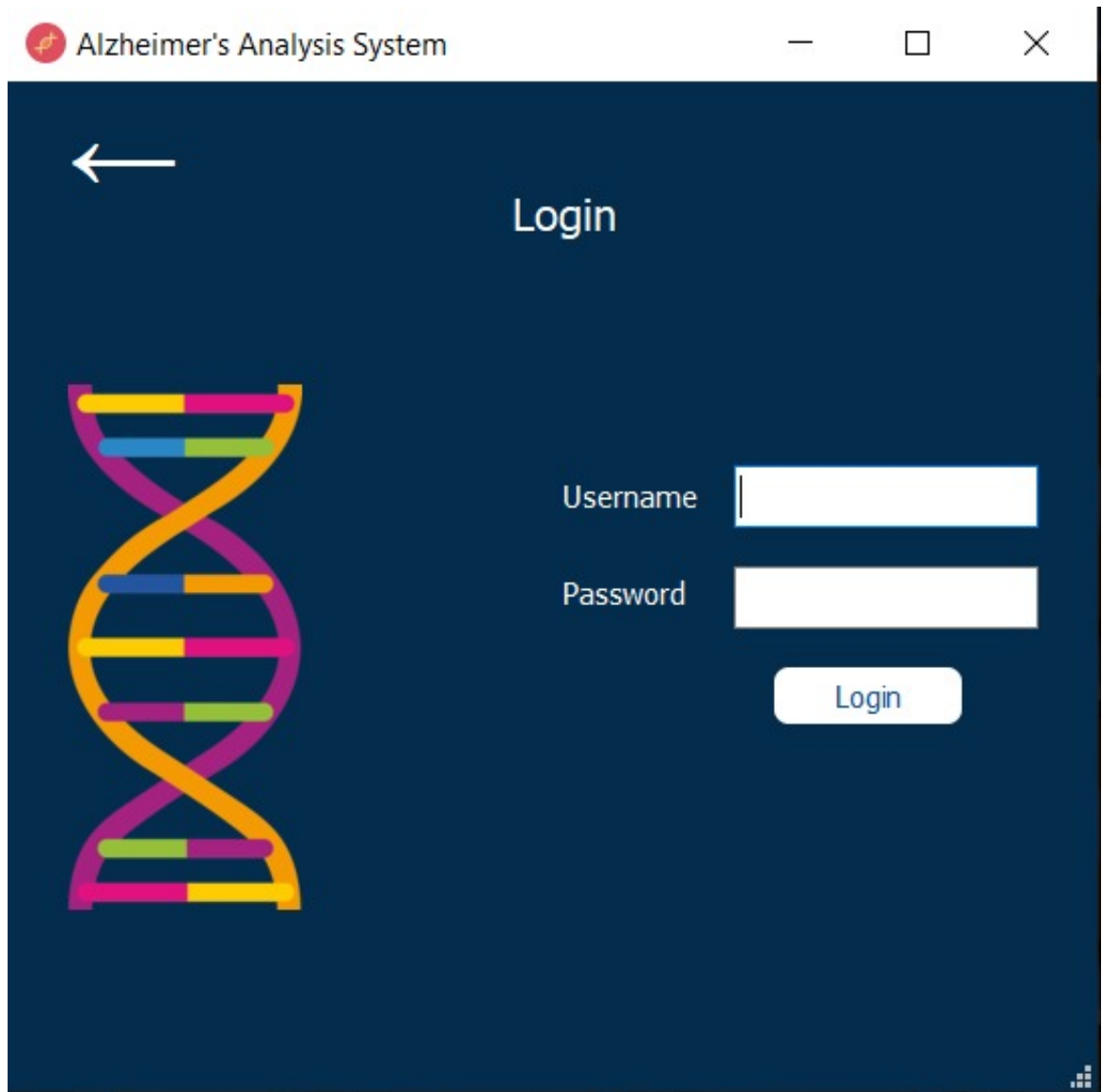


Figure 4.12: Login Screen

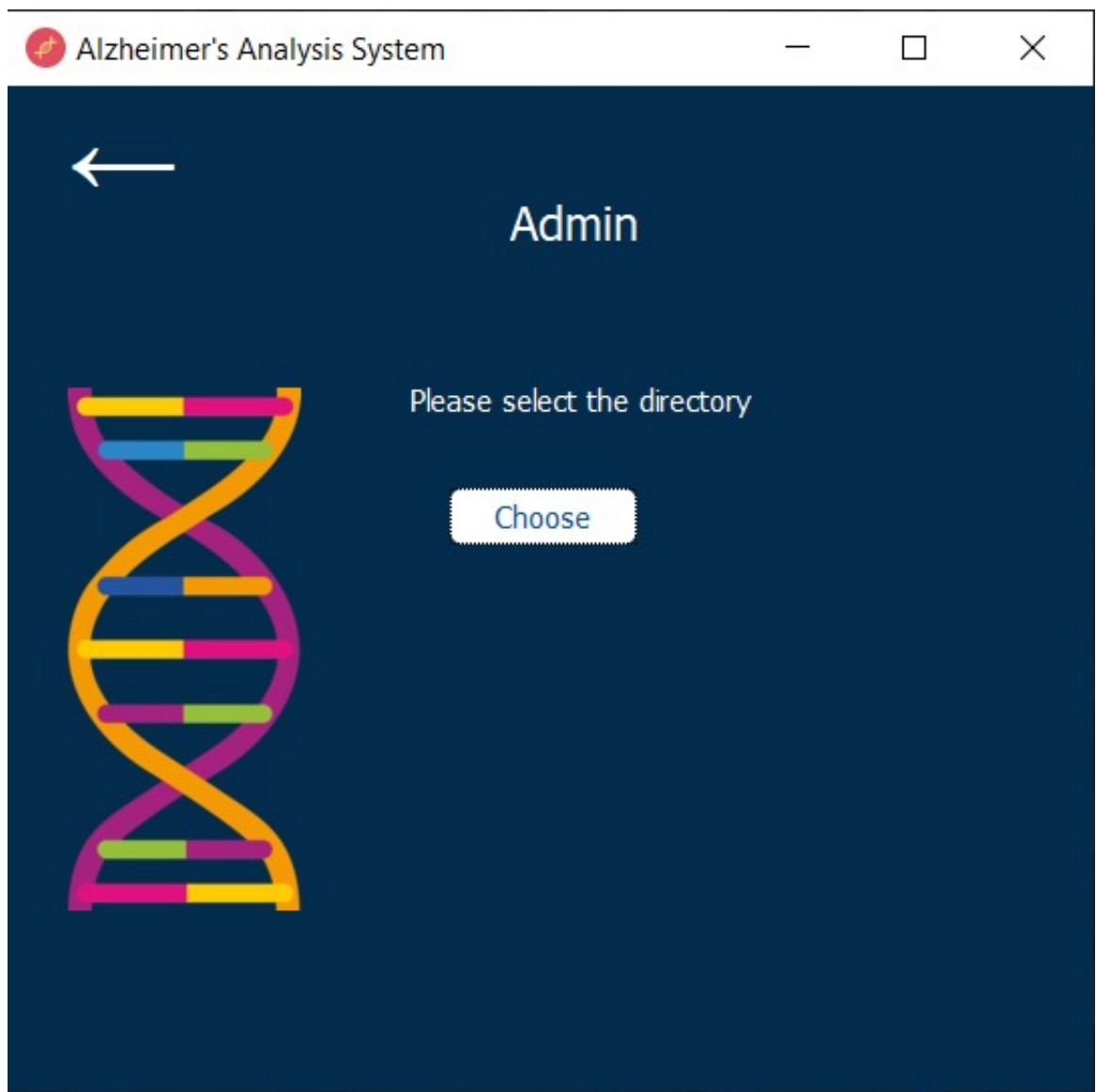


Figure 4.13: Admin Screen

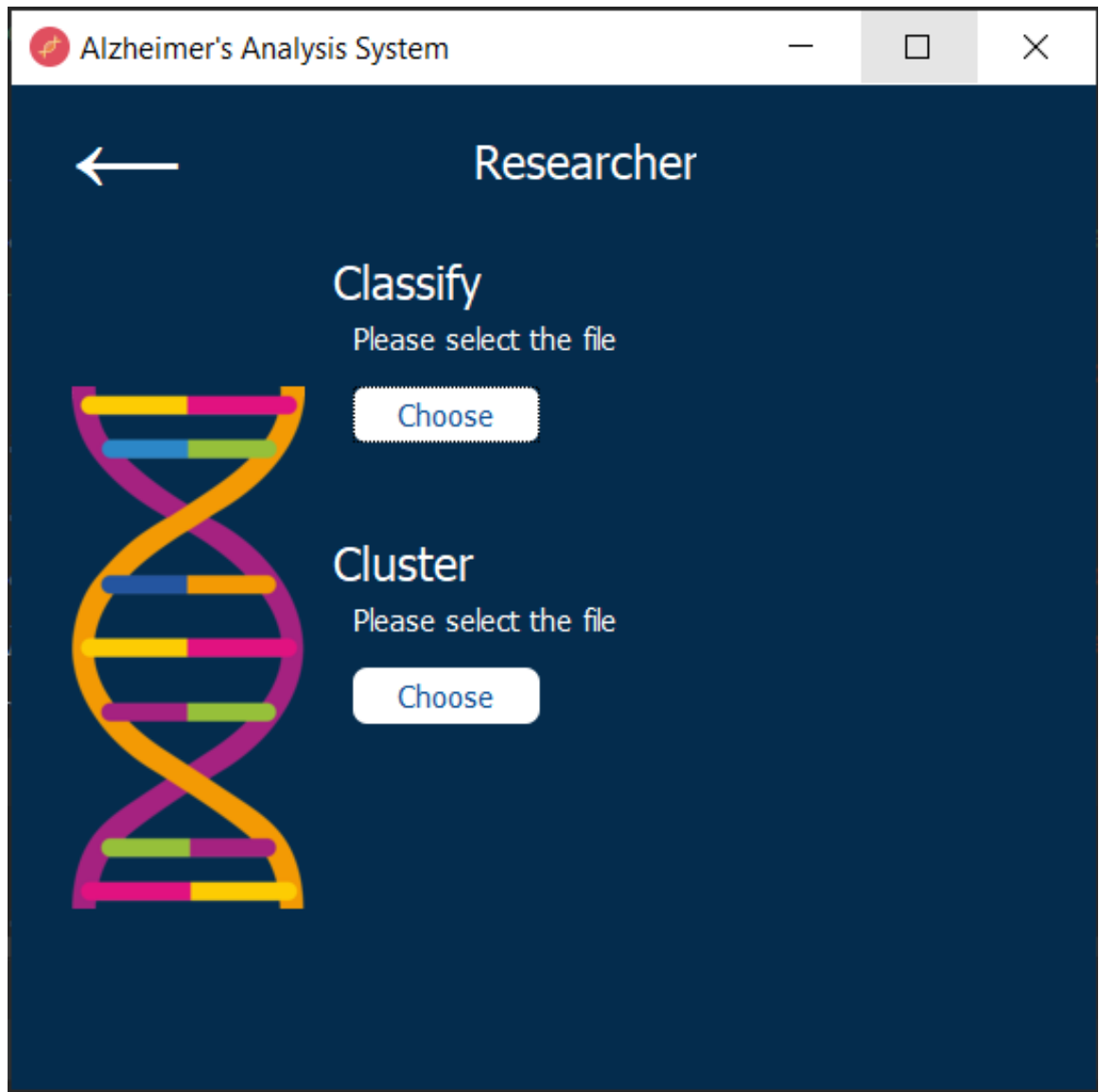


Figure 4.14: Researcher Screen

4.7 Requirements Matrix

Table 4.1: Requirement Matrix

Name	Requirement Id	Type	Description	Module	Status
Upload DNA	3.3	Required	Lab technicians will be able to upload the sample DNA.	Lab technicians /Researcher	Completed
View Result	3.4	Required	Outputs the result of the analysis to the user.	Lab technicians /Researcher	Completed
Check medical history	3.6	Required	By using the patients SSN it will retrieve the medical history of the patient.	Lab technicians /Researcher	Pending
Filter	3.8	Required	Filters all the unnecessary data that may alter the process	Preprocessing	Completed
MergeCsv	3.9	Required	Merges all the .csv files in a given directory into one	Preprocessing	Completed
Conversion	3.10	Required	Converts data into numerals in order to be handled by an algorithm	Preprocessing	Completed
Cluster	3.11	Required	Clusters a given dataset by using kmeans and minibatch kmeans to cluster the diseased samples into 3	Lab technicians /Researcher	Completed
Searcher	3.12	Required	Extracts the four desired chromosomes out of the WGS	N/A	Completed
Remover	3.13	Required	Cuts the unnecessary parts from the WGS chromosomes	N/A	Completed
ToCsv	3.14	Required	Converts WGS files into .csv and splits the sequence into threes each in a cell	Preprocessing	Completed
CrossReference	3.15	New	Crosss references the snps in the datasets and outputs a new dataset with common snps	Preprocessing	Completed
GetScore	3.16	New	Gets the score of the patient in each of the SNPs	Preprocessing	Completed
Transpose	3.17	New	Reshapes the dataset into the required shape	Preprocessing	Completed
SVRClassifier	3.18	New	Classifies the given dataset using SVR	Classification	Completed

Chapter 5

Evaluation

5.1 Introduction

Succeeding the deployment of the desktop application where all the application's functionalities are finalized, the framework ought to pass through the assessment phase in order to ensure that the all of the functionalities are performing as they should be and also ensure that the algorithm chosen is the best for the cases in question. Furthermore, we will include a user study with real users to test the system.

5.2 Algorithm Assessment 1

5.2.1 Setup

Our dataset was downloaded from ADNI. The data was split into three groups training, testing and validation. Afterwards the data was tested with several classifiers.

5.2.2 Goals

To determine the best classifier and implement it into the application.

5.2.3 Findings

Succeeding the end of the comparison SVR was chosen as the algorithm that will be in the system. Since most of the other models achieved 100% training and testing accuracy and due to the nature of the dataset not being large enough to build an accurate model

or diverse. The other classifiers likely over-fitted the data therefore they would not do well in real-world scenarios. As to why SVR was chosen that can be attributed to how the algorithm works, its because it by parsing through the given dataset and mapping the data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable.

Table 5.1: Comparing different algorithms to choose the proper classifier

Number	Classifier	Training Accuracy	Testing Accuracy	Precision	Recall	F1-Score	Support
1	SVR	97%	94%	1.00/0.88	0.92/1.00	0.96/0.93	222/129
2	Random Forest	100%	100%	1.00/1.00	1.00/1.00	1.00/1.00	222/129
3	Logistic Regression	100%	100%	1.00/1.00	1.00/1.00	1.00/1.00	218/133
4	SGD	100%	100%	1.00/1.00	1.00/0.99	1.00/1.00	222/129
5	SVM(SVC)	100%	100%	1.00/1.00	1.00/1.00	1.00/1.00	222/129
6	Decision Tee	100%	100%	1.00/1.00	1.00/1.00	1.00/1.00	222/129
7	MLP Classifier	100%	100%	1.00/1.00	1.00/1.00	1.00/1.00	222/129
8	Ada Boost Classifier	100%	100%	1.00/1.00	1.00/1.00	1.00/1.00	233/118

5.3 Algorithm Assessment 2

5.3.1 Setup

Our data set was downloaded from ADNI. This data consisted of three classes: the healthy patients, the mild patients(potential), and the diseased patients. The data was then split into three groups training testing and validation.

5.3.2 Goal

The goal is to find the best clustering algorithm that fits the current data because even though the data is labeled each class contains only a small amount of patients which made it incredibly hard for a classifier to accurately fit the data.

5.3.3 Findings

After conducting multiple experiments the best approach found was to change the amount of records in each class. Before changing the number was 360 in class one , 214 in class two , and 180 in class three. After the change each class was about 200 records approximately. MiniBatch kmeans was chosen since it yielded the highest accuracy score.

Table 5.2: Comparing different algorithms to choose the proper clustering algorithm

Number	Clustering Algorithm	Accuracy	Miscellaneous
1	K-means	42%	All data was used
2	Birch	45%	All data was used
3	Birch	53%	Not all data was used
4	Agglomerative Clustering	44%	All data was Used
5	MiniBatchKmeans	56%	All data was used
6	MiniBatchKmeans	65%	Not All data was used

Confusion Matrix		
72	65	13
1	190	28
35	60	125

#	Precision	Recall	F1-score	Support
1	0.67	0.48	0.56	150
2	0.60	0.87	0.71	219
3	0.75	0.57	0.65	220

Figure 5.1: Statistical Report

Chapter 6

Conclusion

In this thesis, We have introduced the structure and assessment of a desktop application that uses a DNA sample from the patient in order to be able to correctly diagnose the patient with AD. And also to be able to associate the patient with one of three groups either being healthy, potential, or diseased with AD. Our process starts with the DNA sample being uploaded to the system, afterwards the application preprocesses it to ensure that it is in the same format and size that is needed to amply diagnose the patient. Subsequently, the data along with the patient name and his DNA are loaded into our database and the lab technician can either classify the sample using a model made with SVR-C or cluster the sample with a model made with MiniBatchKMeans. Our models achieved 94% and 85% respectively. Therefore, the choice of which one or both is left to the patient or the lab technician conducting the test.

6.1 Future directions

The future work for the application can include adding more samples to the dataset to improve the accuracy of the model. Furthermore we have added a method to train and deploy new models for even more chronic neurodegenerative diseases.

Bibliography

- [1] H. Förstl, “What is alzheimer’s disease,” *Dementia*, vol. 2, pp. 371–382, 2010.
- [2] “Where is dna found?” 2020 (accessed April 20, 2020). [Online]. Available: <https://www.ancestry.com/lp/where-is-dna-found>
- [3] S. Saranya, V. Loganathan, and P. RamaPraba, “Efficient feature extraction and classification of chromosomes,” in *International Conference on Innovation Information in Computing Technologies*. IEEE, 2015, pp. 1–7.
- [4] C. Jack, “Genetics of alzheimer’s disease,” in *Alzheimer’s Turning Point*. Springer, 2016, pp. 75–83.
- [5] N. I. on Aging, N. Relkin, A. A. W. Group *et al.*, “Apolipoprotein e genotyping in alzheimer’s disease,” *The Lancet*, vol. 347, no. 9008, pp. 1091–1095, 1996.
- [6] E. Suberbielle, B. Djukic, M. Evans, D. H. Kim, P. Taneja, X. Wang, M. Finucane, J. Knox, K. Ho, N. Devidze *et al.*, “Dna repair factor brca1 depletion occurs in alzheimer brains and impairs cognitive function in mice,” *Nature communications*, vol. 6, p. 8897, 2015.
- [7]
- [8] S. Chatterjee, A. Iyer, S. Avva, A. Kollara, and M. Sankarasubbu, “Convolutional neural networks in classifying cancer through dna methylation,” *arXiv preprint arXiv:1807.09617*, 2018.
- [9] C. Bonvicini, C. Scassellati, L. Benussi, E. Di Maria, C. Maj, M. Ciani, S. Fostinelli, A. Mega, M. Bocchetta, G. Lanzi *et al.*, “Next generation sequencing analysis in early onset dementia patients,” *Journal of Alzheimer’s Disease*, no. Preprint, pp. 1–14, 2019.

-
- [10] K. P. Soh, E. Szczurek, T. Sakoparnig, and N. Beerenwinkel, "Predicting cancer type from tumour dna signatures," *Genome medicine*, vol. 9, no. 1, p. 104, 2017.
 - [11] M. M. Abd El Hamid, M. S. Mabrouk, and Y. M. Omar, "Developing an early predictive system for identifying genetic biomarkers associated to alzheimer's disease using machine learning techniques," *Biomedical Engineering: Applications, Basis and Communications*, vol. 31, no. 05, p. 1950040, 2019.
 - [12] O. Erdogan and Y. A. Son, "Predicting the disease of alzheimer with snp biomarkers and clinical data using data mining classification approach: decision tree." in *MIE*, 2014, pp. 511–515.
 - [13] J. Li, Q. Zhang, F. Chen, X. Meng, W. Liu, D. Chen, J. Yan, S. Kim, L. Wang, W. Feng *et al.*, "Genome-wide association and interaction studies of csf t-tau/ $\alpha\beta$ 42 ratio in adni cohort," *Neurobiology of aging*, vol. 57, pp. 247–e1, 2017.
 - [14] M. Awad and R. Khanna, "Support vector regression," in *Efficient Learning Machines*. Springer, 2015, pp. 67–80.