**Software Design Document**

For

**Recona**

Analysis of cell composition from tissue expression profiles and identification of infection with lynch syndrome

Version 1.0

Prepared by:

* **Mariam Adel Abdelrazik**
* **Marwa Gamal Salman**
* **Marwa Kamal Badri**
* **Mai Ramadan Mohamed**
* **Maha Mamdouh Adly**
* **Amira Emad Abdelhamid**
* **Ayat Saber Mohamed**

**Faculty of computers and Information Science, Assiut University**

**11** / **1 / 2023**

**TABLE OF CONTENTS**

1.0 INTRODUCTION 3

1.1 Purpose 3

1.2 Scope 3

1.3 Overview 3

2.0 SYSTEM OVERVIEW 4

3.0 SYSTEM ARCHITECTURE 5

3.1 Architectural Design 5

3.2 Decomposition Description 6

3.3 Design Rationale 6

4.0 DATA DESIGN 7

4.1 Data Description 7

4.2 Data Dictionary 9

5.0 COMPONENT DESIGN 10

6.0 HUMAN INTERFACE DESIGN 12

6.1 Overview of User Interface 12

6.2 Screen Images 13

6.3 Screen Objects and Actions 19

7.0 REQUIREMENTS MATRIX 20

## INTRODUCTION

## Purpose

This Software design document describes the Recona website system and describes the design decisions, Architectural design, and detailed design needed to implement the system, provide visibility in the design, and provide the information needed to implement new reliable software. The expected audience is end users of the website Doctors, Bioinformaticians, and bioinformatics students and will be used by all the development teams including developers, testers, quality assurance, and maintenance.

## Scope

## This document contains a complete description of the design of the Recona website. The basic architecture is a web server from a client-server paradigm. The basic pages will be in HTML, bootstrap, and, JSP. Recona mainly is a website that identifies the cellular composition of tissues and chromosomes for cells to help doctors detect Lynch syndrome. People with this syndrome are at risk of developing various types of cancers, they are advised that early identification will significantly speed up treatment and reduce risk. Users of Recona can access the platform anytime, anywhere, and receive fast and meticulous service. The website will serve all doctors, bioinformaticians, and bioinformatics students. The main scope is to identify the cellular composition of tissues and chromosomes and detect lynch syndrome mutation.

## Overview

## The remaining sections and their contents are listed below.

## Introduction: that describes the purpose and scope of the Recona project and basic information about it

## System Overview: that gives a general description of the functionality, context, and design of the Recona project.

## Architectural Design: develops a modular program structure and explains the relationships between the modules to achieve the complete functionality of the system. This is a high-level overview of how the responsibilities of the system were partitioned and then assigned to subsystems. Identify each high-level subsystem and the roles or responsibilities assigned to it.

## Data Design: that concerns the data structure design

## Component Design: that provides a closer look at what each component does in a more systematic way you gave a functional description in section 3 and provide a summary of our algorithm for each function listed in 3

## Human Interface Design: that discusses the user interface design and how it can be created with maximum user efficiency and ease of use.

## Requirements Matrix: that provides us a cross reference that traces components and data structures to the requirements in our SRS document.

## SYSTEM OVERVIEW

Recona mainly is a website that identifies the cellular composition of tissues and chromosomes for cells to help doctors detect Lynch syndrome. People with this syndrome are at risk of developing various types of cancers, they are advised that early identification will significantly speed up treatment and reduce risk. The website will provide better options for users to search and manipulate their sequence so that they can know the type of tissue they are working on, know the chromosome and discover if the owner of the genetic sequence has Lynch syndrome. This genetic disorder increases the risk of many types of cancer, especially colon cancer and endometrial cancer. Thus, predicting a patient's risk of developing these cancers is possible.

## SYSTEM ARCHITECTURE

## Graphical user interface, diagram Description automatically generatedArchitectural Design

## Decomposition Description

1. Upload DNA Sequence: This component is responsible for receiving and storing the DNA sequence uploaded by the user. It should validate the sequence to ensure that it is a valid DNA sequence.
2. Validate DNA Sequence: This component is responsible for checking the validity of the DNA sequence and ensuring that it meets certain criteria, such as having the correct number of base pairs and not containing any invalid characters.
3. Analyze DNA Sequence: This component analyzes the DNA sequence and determines the corresponding body tissue. This may involve comparing the sequence to a reference database of DNA sequences and identifying matches.
4. Determine Body Tissue: This component uses the results of the analysis to determine the body tissue corresponding to the DNA sequence.
5. Check for Lynch Syndrome: This component checks the DNA sequence for the presence of mutations associated with Lynch syndrome.
6. Determine Type of Cancer: If Lynch syndrome is detected, this component uses the DNA sequence to determine the type of cancer the person may have.
7. Display Results: This component displays the results of the analysis to the user, indicating whether Lynch syndrome is present and, if so, what type of cancer the person may have.

## Design Rationale

1. Upload DNA Sequence: We chose to include a component for uploading the DNA sequence because it allows the user to easily input the sequence and avoid manual data entry errors. The sequence is validated to ensure that it is a valid DNA sequence and meets certain criteria, such as having the correct number of base pairs and not containing any invalid characters.
2. Validate DNA Sequence: We included a validation step to ensure the accuracy of the analysis results. By checking the validity of the DNA sequence, we can reduce the risk of errors in the analysis and ensure that the results are reliable.
3. Analyze DNA Sequence: We chose to use a reference database of DNA sequences to determine the body tissue corresponding to the DNA sequence. This approach allows us to make use of existing knowledge and data and more accurate predictions based on large sample size.
4. Determine Body Tissue: We used a comparison method to match the DNA sequence to those in the reference database to determine the corresponding body tissue. This allows us to identify the body tissue quickly and accurately, based on the best available information.
5. Check for Lynch Syndrome: We chose to check for Lynch syndrome because it is a well-known genetic disorder that can lead to cancer, and early detection is important for effective treatment.
6. Determine Type of Cancer: If Lynch syndrome is detected, we used the DNA sequence to determine the type of cancer the person may have. This allows us to provide more specific and targeted information to the user.
7. Display Results: We chose to display the results of the analysis to the user clearly and concisely. This allows the user to understand the results quickly and easily, and to take appropriate action based on the information provided.

.

## DATA DESIGN

## Data Description

## The data will be stored in databases will be used to store and manage the DNA sequence data that users upload. A high-level overview of the components of a database for the system

## Tables: The database will likely include several tables, each designed to store specific types of data. For example, there may be a table for storing user Information, a table for storing DNA sequences, and a table for storing information about Lynch syndrome and other types of cancer.

## Fields: Each table will have a set of fields that define the data stored in the table. For example, the user table might include fields for user name, password, and email address. The DNA sequence table might include fields for the sequence itself, the date it was uploaded, and any associated information such as the tissue type.

## Relationships: The tables in the database will be related to one another in specific ways. For example, each DNA sequence will be related to a specific user, and each sequence will be associated with information about a specific type of cancer or Lynch syndrome.

## Data Validation: The database should include data validation rules to ensure that only valid data is stored in the tables. For example, the system might validate that the DNA sequence data is in the correct format and that the tissue type and cancer information is consistent with the sequence data.

## Security: The database should include security measures to ensure that sensitive data, such as user information and DNA sequences, is protected from unauthorized access. This might include password protection, encryption of sensitive data, and access controls to limit who can view and modify data.

## Backups: The database should include a backup and recovery system to ensure that data is not lost in the event of a system failure or other problems. Regular backups should be taken, and the system should be designed to allow for easy recovery of data in the event of a disaster.

## These are the basic components of a database for the system you described. The exact design of the database will depend on the specific requirements of the system, including the types of data that need to be stored, the relationships between data, and the security and performance requirements.

## Data Dictionary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Filed name** | **Data type** | **Filed size** | **description** | **Example** |
| First name | text | 20 | User first name | Ali |
| Last name | text | 20 | User last name | Salah |
| E-mail | text | 15 | User E-mail to sign up | Ali7409@gamil.com |
| Password | numbers | 15-20 | User password and confirm it | 14526523 |
| history | Files | --- | User projects | Project1,Project2,…etc |
| Cancer type | text | 30 | Patient cancer | Lung cancer |
| RNA sequence | text | --- | RNA sequence for the patient to detect the cancer | ACGACGACGACG |
| Tissue type | text | 30 | Tissue type | Muscle tissue |

## COMPONENT DESIGN

1. **Upload DNA Sequence:**

def upload\_dna\_sequence(sequence):

    # Store the DNA sequence

    dna\_sequence = sequence

    # Validate the DNA sequence

    if not validate\_dna\_sequence(dna\_sequence):

        raise ValueError("Invalid DNA sequence")

    return dna\_sequence

1. **Validate DNA Sequence**:

def validate\_dna\_sequence(dna\_sequence):

    # Check the number of base pairs

    if len(dna\_sequence) != NUM\_BASE\_PAIRS:

        return False

    # Check for invalid characters

    for char in dna\_sequence:

        if char not in VALID\_CHARS:

            return False

    # If all checks pass, the DNA sequence is valid

    return True

1. **Analyze DNA Sequence:**

def analyze\_dna\_sequence(dna\_sequence):

    # Compare the DNA sequence to the reference database

    for reference\_sequence in REFERENCE\_DATABASE:

        if dna\_sequence == reference\_sequence:

            return REFERENCE\_DATABASE[reference\_sequence]

    # If no match is found, return None

    return None

1. **Determine Body Tissue:**

def determine\_body\_tissue(dna\_sequence):

    analysis\_result = analyze\_dna\_sequence(dna\_sequence)

    if analysis\_result is not None:

        return analysis\_result

    else:

        raise ValueError("Unable to determine body tissue")

1. **Check for Lynch Syndrome:**

def check\_for\_lynch\_syndrome(dna\_sequence):

    for mutation in LYNCH\_SYNDROME\_MUTATIONS:

        if mutation in dna\_sequence:

            return True

    return False

1. **Determine the Type of Cancer:**

def determine\_cancer\_type(dna\_sequence):

    if check\_for\_lynch\_syndrome(dna\_sequence):

        for cancer\_type in CANCER\_TYPES:

            if cancer\_type in dna\_sequence:

                return cancer\_type

        return "Lynch syndrome-associated cancer"

    else:

        return None

1. **Display Results:**

def display\_results(dna\_sequence):

    body\_tissue = determine\_body\_tissue(dna\_sequence)

    cancer = determine\_cancer\_type(dna\_sequence)

    if cancer is not None:

        print("DNA sequence corresponds to body tissue:", body\_tissue)

        print("Lynch syndrome detected. Possible cancer type:", cancer)

    else:

        print("DNA sequence corresponds to body tissue:", body\_tissue)

        print("Lynch syndrome not detected.")

## HUMAN INTERFACE DESIGN

## Overview of User Interface

# The user will be able to access this application through the web. The upload page will be the main focus of the application with a navigation bar at the top for other tabs from the home page, log-in, signup, researchers, about, and profile page, to access the upload page the user will have to login or create an account if he does not have an account Then he will be able to upload the file containing his DNA sequence and wait for the result.

## Screen Images

## 6.2.1 Home page:

## the home page consists of four sections header section, the about us section, our team, and the contact us section.

## 

# **6.2.2** **About page :**

# The about Page works to clarify a set of main points about the project from the main function of the project, the project abstract, objectives, approach and methodology, and references.

# 

# 

# **6.2.3 Researcher page :**

# The researcher's page provides links through which the researcher can download the test file and the database through which he can obtain other files.

# 

# **6.2.4 Log in & Sign in page :**

# You log in to officially record your presence as a returning user, whereas sign-up simply registers your intent to be present officially in the system. Sign-up means creating an account as a new user so that you can log in later with your credentials.

# 

# 

# **6.2.5 Profile page :**

# The profile page consists of your profile picture and a history of the previous uploads you made.

# **6.2.6 Upload page :**

# On the upload page, there is a button that, when pressed, will open a box in which you can write the name of the case and the attached file, then press start.

# 

## Screen Objects and Actions

## REQUIREMENTS MATRIX

## A picture containing table Description automatically generated