DOCUMENT OF THE SYSTEM

1. Introduction

- **Purpose of the Document:**
  This document aims to provide readers with a deeper understanding of the design of our entire program. After presenting the system, readers will want to explore the internal structure of our database for detecting allergies in detail.
- **Project Scope:**
  This project is intended solely for doctors and patients. The main objective is to make the doctor's work as easy as possible by efficiently conveying necessary information to patients and maintaining a clear and comprehensive control of all patients. On the other hand, patients will have all the necessary information readily available whenever they need it.

2. System Overview

- **Problem Description:**
  The primary problem our database aims to improve, as previously mentioned, is enhancing the interaction between doctor and patient to make it as effective as possible, thereby improving the quality of life for both.
- **Proposed Solution:**
  Our system addresses this problem by quickly analyzing a patient's conditions and associating them with an allergy. Additionally, the doctor can securely manage all patients by adding, deleting, or modifying their information. Prescriptions and treatments will be stored in the database, allowing patients to access them anytime, preventing issues like lost paper prescriptions.

3.System Requirements

- **Functional Requirements:**
  Our database requires the creation of an account to access it, using a username and password. Once the account is created, the user must choose a role, either as a patient or a doctor. Each role allows different actions. Doctors can add, delete, and update patients, add new symptoms or allergies, diagnose treatments, and provide prescriptions. Patients can view their medical scores and the prescriptions given by the doctor.
- **Non-Functional Requirements:**
  If a patient is registered twice, the system will not be able to perform certain actions. Additionally, if the patient has no registered symptoms or allergies, some actions (like creating a prescription) cannot be performed.

4. Database Design

- **Data Model:**
  We have created a data model that relates six entities (patient, doctor, allergy, symptom, treatment, and prescription).
- **Entity-Relationship Diagrams (ERD):**
  Includes ER diagrams showing the main entities (Patient, Doctor, Symptom, Allergy, Treatment, Prescription) with their 1-n and n-n relationships (e.g., between treatment and allergy, patient and allergy, and patient and symptom).

5. Functionality Description

- **User Management:**
  Create and validate accounts by choosing your role (Patient or Doctor) and entering required details. Log in with your credentials to access specific menus.
- **Doctor's Functionality:**
  Add, delete, and update patient information, add symptoms and allergies, diagnose and assign treatments, and create prescriptions.
- **Patient's Functionality:**
  Check medical records and view prescriptions.
- **Why We Did That:**
  To ensure each user fulfills a specific role and to maintain an organized and detailed database. Additionally, signing up securely stores all user information in the database.

6. Design Decisions Justification

- **Database Platform Choice:**
  We chose DBBrowser because we learned in class how to integrate it with Eclipse, making it easy to ensure the project works correctly, particularly in relating symptoms and allergies to patients.
- **Data Structure:**
  We brainstormed in the initial days, considering a typical doctor's visit and identifying the main entities and actions involved.

7. Implementation and Testing

- **Implementation Plan:**
  Using Eclipse, we will demonstrate a console menu showing how a patient or doctor can create an account, enter their information, and how the doctor can relate symptoms to the patient's allergy.

8. Conclusions

- **Design Summary:**
  The project starts with a menu where, depending on your role, you can access the database in different ways. The project's objectives were perfectly met, as our main idea was to associate a patient's symptoms (e.g., respiratory allergy symptoms like mucus, digestive allergy symptoms like stomach pain, skin allergy symptoms like redness and itching) with treatments stored in the database.
- **Future Improvements:**
  A significant improvement could be including a Hospital as an "actor" to manage all this information from specific hospitals. Additionally, allowing patients to choose their visit dates could help both patients and doctors coordinate appointments more effectively.