**Software Design Specification (SDS)**

Revision History: (The server and client documents should be combined into one for a single project)

|  |  |  |
| --- | --- | --- |
| Date | Author | Description |
| 5.2 | Jarvis Liu |  |
| 5.2 | Egbert Ding |  |
| 5.5 | Jarvis Liu |  |
| 5.5 | Egbert Ding |  |
|  |  |  |
|  |  |  |

Contents

1. Introduction

1.1. Intended Audience and Purpose

1.2. How to use the document

2. System Design

2.1. Context

2.2. Design Pattern

3. Module Interface Design

4. Detailed Design

A.    Appendices

A.1    Definitions and acronyms

A.1.1    Definitions

A.1.2    Acronyms and abbreviations

A.2    References

## **Introduction**

## **Intended Audience and Purpose**

The purpose of this document is to describe the database structure and design in detail. The readers of this document can be the developers of the database , the testers of the system and the whole server team.By reading this document, the developers can know names and types for each attribute in each table, and they can implement all the interfaces. The testers can come up with different testing methods based on the design and the server team may know what functions to call to satisfy the needs of each team.

## **How to use the document**

For the developers: we give detailed design and the required interface design in the document as a reference.

For the testers:we give detailed interface design so they understand what operations the server team would do so they can design some oriented testings.

For the server team:By reading this document,the server team may know how to satisfy the needs of each team with the interfaces we give.

## **System Design**

we give a detailed description of the system contexts from an architect's point of view and clarify the context of the software in the part.

## **Context**

the programming languages to develop the software with：SQL

the operating system your software runs on：Windows

the database management system your data will be stored：MYSQL

the internet protocol for the component communication：TCP/IP

## **Design Pattern**

We develop the relational database via MySQL.There are five tables in the database, including User Table, Message Table, Patient Record Table, News Table and Chatbot Table. User Table describes the information of each patient and doctor, such as name, birthday, account type and so on. The primary key of this table is User ID.

Message Table describes the information of each message between patients and doctors, such as ID of sender and receiver. The primary key of this table is Message ID.

Patient Record Table describes the information of each diagnose record, such as result, diagnose doctor and so on. The primary key of this table is Record ID.

News Table describes the information of each news, such as the content of the news, time and so on. The primary key of this table is News ID.

Chatbot Table describes the information of each answer of the chatbot, such as the content of the answer, keyword and so on. The primary key of this table is Answer ID.

There is possession relation between User Table and Patient Record Table, and one patient table corresponds with more than one patient record tables.

There is possession relation between User Table and Meaasge Table, and one patient table corresponds with more than one meaasge tables.

## **Module Interface Design**

**User Profile APIs**

User Profile APIs provides basic functions for users to register/login/logout/manage their accounts.

**Algorithm APIs**

Algorithm APIs mainly receive images, which need to be detected, then send back automatic detection result through ML algorithm. Algorithm APIs also provides Algorithm management API to choose which algorithm to be used.

**Treatment APIs**

Treatment APIs mainly serve for user to find their upload history and provide records for clients to generate line chart. For admin, Treatment APIs allow administrators to search records of all users.

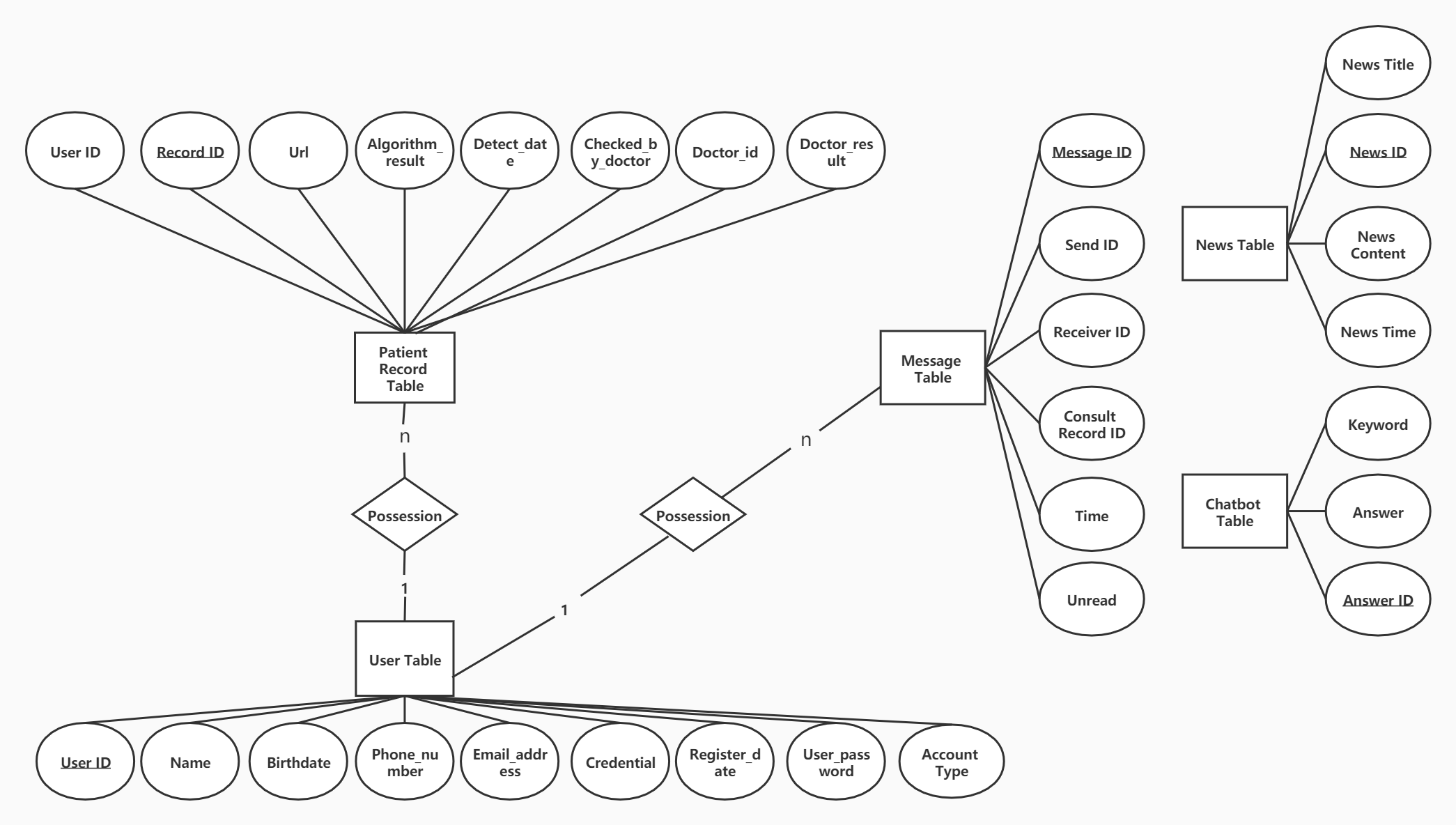
**Message APIs**

Message APIs mainly serve for patients to send messages to doctor and for doctors to reply to patients’ message.

**Backup APIs**

Backup APIs are designed for administrators to back up all data in the system including user profiles, upload records and messages.

## **Database Detailed Design**



## **A.    Appendices**

## **A.1    Definitions and acronyms**

## 

## **A.1.1    Definitions**

|  |  |
| --- | --- |
| **Keyword** | **Definitions** |
|  |  |
|  |  |
|  |  |
|  |  |

## **A.1.2    Acronyms and abbreviations**

|  |  |
| --- | --- |
| **Acronym or**  **Abbreviation** | **Definitions** |
|  |  |
|  |  |
|  |  |

## **A.2    References**