|  |
| --- |
| DISEASE PREDICTION SYSTEM |
| Software document Specification |
| Project 1 |

|  |
| --- |
| Windows User  [Pick the date] |



NAME: SANTOS OKELLO

REGISTRATION NUMBER: 19/05036

Course: BSC.IT

UNIT: PROJECT 1

INTRODUCTION

Purpose

This software design specification is made with the purpose of outlining the software architecture and design of the car repair registration system in detail. The document will provide developers an insight in meeting client’s needs efficiently and effectively. Moreover the document facilitates communication and understanding of the system by providing several views of the system design.

Scope

The software Design document would demonstrate how the design will accomplish the functional and non- functional requirement captured in the software requirement specification (SRS) .The document will provide a framework to the programmers through describing the high level components and architecture, sub systems, interface, database design and algorithm design. This is achieved through the use of architectural patterns, design patterns, sequence diagrams, class diagrams, relational models and user interface

Intended audience

This document is mainly for the developers and technical and garage administrative staff of Jericho garage and customers of Jericho garage

Glossary

|  |  |
| --- | --- |
| Algorithm design | Specific method to create a mathematics process in solving problems |
| Architectural design | Establishing the overall structure of software system |
| Compatible | Capable of ordering efficient intergration and operation with other elements in a system with no modification |
| Database | A collection of stored related data |
| Encapsulate | To express or show the most important facts about something |
| ER diagram | Entity Relationship Diagram, Data model for describing data base in an abstract way |
| SDS | System design specification |
| Sequence Diagram | An interaction diagram that show how process interact with one another and in what order. |
| SRS | Software Requirement Specification |

Document overview

The next chapter of the document has described architectural design of the garage repair registration system. The high level components and their interactions, suitable architectural patterns, physical arrangement of components and design applied to the whole system.

The other chapter of this SDS is on components and detailed design, includes design patterns , sequence diagrams, class diagrams, database design in detail and user- interface design with screenshots of the interface

Username

password

Login

This the screenshot of the user interface that account already existing account

Firstname

Lastname

Email

Residence

Phone number

Sign up

Password

car

This the sign up format of a car owner with no existing account in hand

HIGH LEVEL COMPONENT AND INTERFACE

Components

Car components

This is one of the key components of the car repair registration system. This is where the car repair parts subject selection option is implemented. This also include previous damage, payment and discount given.

Authentication and user management components

This is the major sub system that is responsible for the security of the security of the car repair registration. It authenticate users and also handles the user management activities such as creating new user account from the system etc. Furthermore this component implements the control access privilege matrix

Subject component

This is the key component that implements the functions related to the subject operations of administrator such as adding a new spare parts of the car repaired, editing credits of an existing spare and removing spare etc.

Publish components

Publish components is the component responsible for publishing notice create by the administrator on when the repair will be done

Public components

This is a relatively small subsystem compared to the other component of the car repair system. This is the component which is rensponsible for the guest viewer (public) to view progress of the car’s repair being done. It also entails a comment section where all the witten comments written by a customer can be viewed by anybody

Car owner components

Car Owner Info: This interface will provide the available car info that is being repaired.

Car Owner profile: This interface is used for the car owner to view and edit personnel information.

Authentication and user management component

Authenticate user: This is the interface that allows the users to login to the system. This will guide the user to the relevant home page

Publish component

Get notice: This is the interface where notifications are published. It is connected with the user component

Public component

View result: This is the interface which shows the result of the car to a guest user, therefore connected with the user component

Architectural styles/Patterns

The garage management system will be developed under two main architectural styles/patterns. Development of the project will be done in MVC architectural style and also 3 tier Client/Server architecture. Client can browser the internet and access the Student system provided within the local area network of the garage

MVC Architecture (Model – View – Controller)

MVC Style separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other

The model component –Manages the system data and associated operations on that data.

The View component – Defines and manages how the data is presented to the user.

The Controller components – Manages user interaction and passes these interaction to the View and the model

We will use this MVC style for the garage management system because, there are multiple ways to view and interact with data. Also used when the future requirements for interaction and presentation of data are unknown. In some software systems the code between the process logic and interface are mixed. This will reduce the modularity of application and make the make the system more difficult to maintain. To avoid this problem we have decided to use MVC architectural style to separate the application logic with the interface. The main advantage of this style allows the data to change independently of its representation and vice versa. Support presentation of the same data in different ways with changes made in one representation shown all of them

Three-tier Client/Server Architecture

In a client server architecture, the functionality of the system is organized into services, with each service delivered from separate server. Client are users of these services and access servers to make use of them. We will use this 3- Tier Client are Architecture because, When data in a shared database has to be accessed from a range of locations. Because server can be replicated, may also be used when the load on a system is a variable.

Data tire

The data tire maintains the applications data such as User’ data, car details and comments and the SQL queries. It stores these data in a relational database management system (RDBMS). All the connections with the RDBMS are managed in the tier.

Middle Tire

The middle tier (web/ application server) implements the business logic, controller logic and presentation logic to control the interaction between the application’ client and data. Business rules enforced by the business logic dictate how client and cannot access application data and how application process data.

Client Tire

The client tire is the application user interface connecting data entry forms and client side applications. It displays data to the user server .User interact directly with the application through user interface. The client tier interacts with the web/application server to make requests and to retrieve data from database. IT the displays to the user the data retrieved from the server.

Different process and Their communication

In the garage management system, there are number of different processes, such as database server process, web server process, connections between above server likewise. When sending mails there should run a mail server. HTTP protocol is using to communicate with web server, SMTP protocol is using to communicate with mail servers. They should communicate each other well to perform the functions of whole application.

Arrangement of devices and serves

Jericho garage management system needs some specific set of servers and devices: Such as:

Server to host web applications and web services service applications.

Personal computer, note book, smart phone etc… to access the website

Mode /router/switch/hub/WI-FI network/ cable network etc ……. And also need an internet Service Provider to have the Internet connectivity.

Communication among components

Below devices are communicating with each other. Personal computer communicates with web server and the database through HTTP protocol. It communicates with mail server through SMTP protocol. Cable network or wi-Fi network is also a communication method using in connecting different network components

Three-Tier Client Server Architecture

Reasons:

* As more user access the system a three tier solution is more scalable than the other solutions because you can add as many middle tiers as needed to ensure good performance.
* Security is also the best in three-tier architecture because the middle layer protects the database tier
* Ease to coding and provide well defined interfaces within each logic

Techniques Used

Prototyping

In designing the garage management system prototyping will be used to demonstrate underpinning concept of the designing and for user interface. This techniques will provide the opportunity for the system users to experiment the software to a certain extent during the development process.

Architecture

The architecture provides the top level design view of a system and provides a basis for more detailed design work

Strategy 1

Database schema

Tables, Fields and Relationships

Provide a description of any new tables, fields and relationships that need to be created for the design

Databases

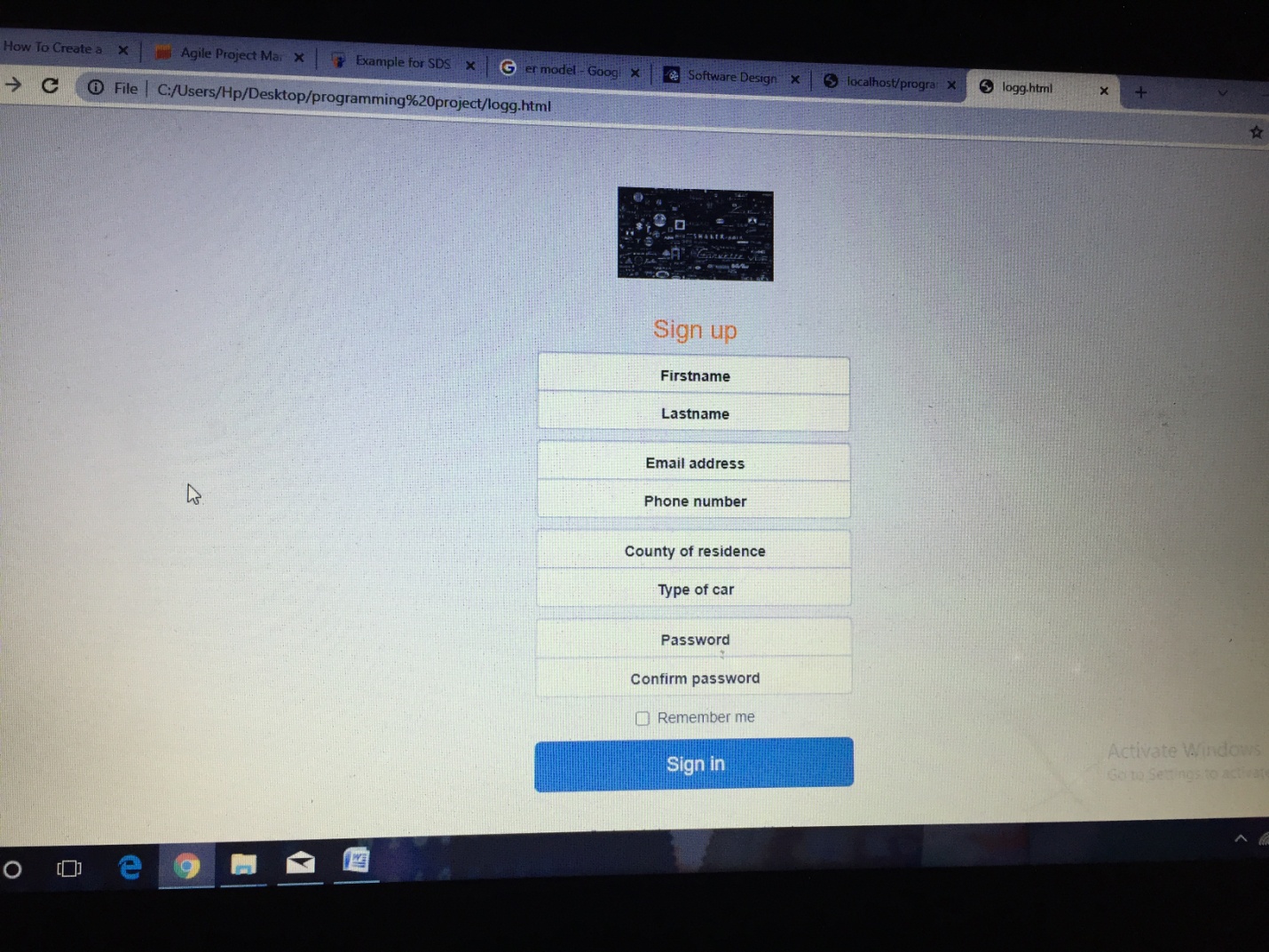
Wampserver to be used

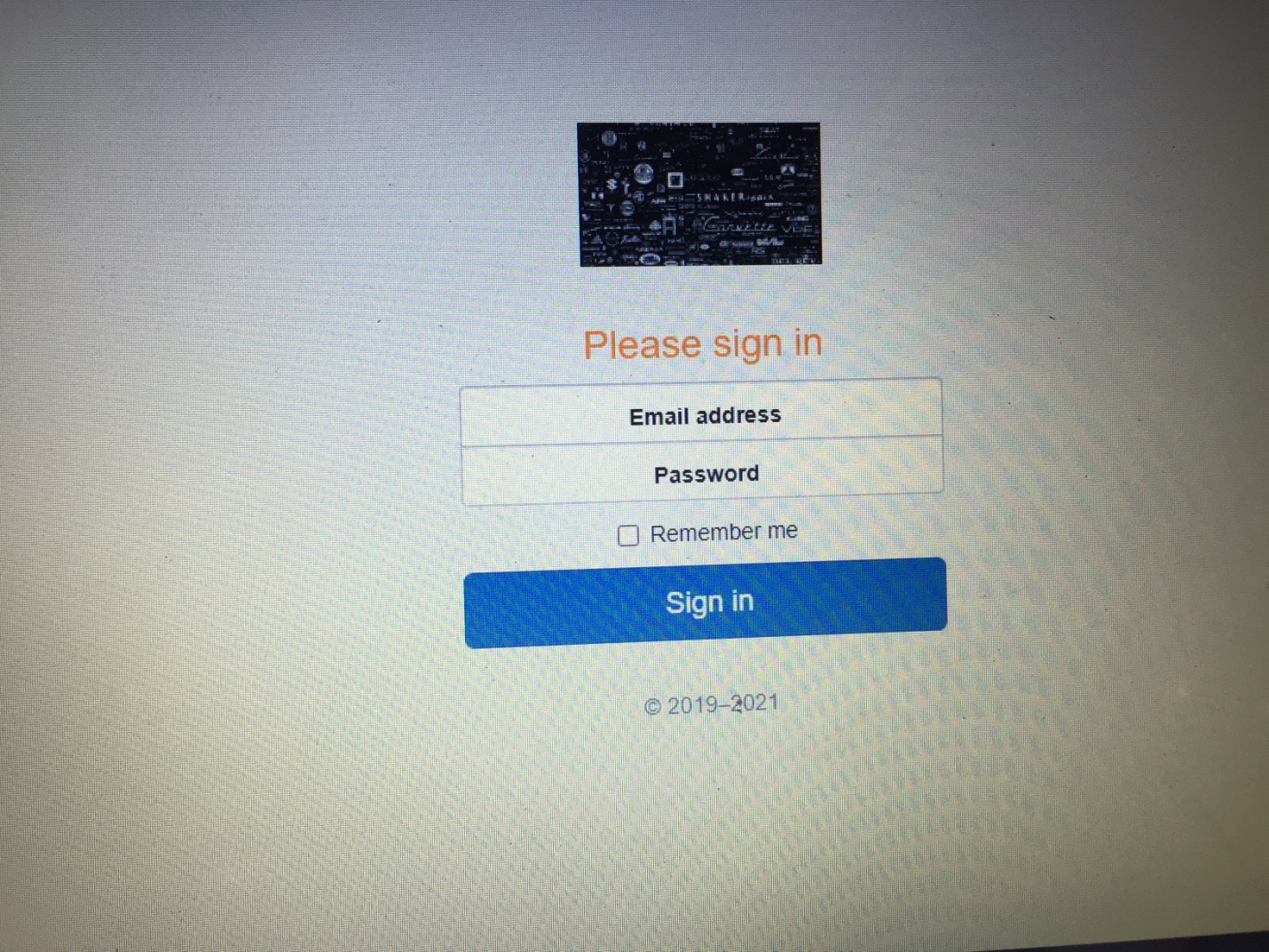
New Tables

List of tables that will be needed one including the table name table description and related table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Field Name | Data Type | Allows Nulls | Fields Description |
| user | First name | Varchar(250) | Not null | Get firstname |
|  | Last name | Varchar(250) | Not null | Get the fields lastname |
|  | Email | Varchar(250) | Not null | Get email address |
|  | Phone number | Int(250) | Not null | Phone number of the customer |
|  | Residence | Varchar(250) | Not null | Get residence address |
|  | Car | Varchar(250) | Not null | Type of car to be repaired |

User Interface

This how the login and sign up form is going to look like



The two pictures shows how the user will be able to sign up for a new account to get in touch with the services offered by the system

Testing

The administrative will test the system expecting pros and cons after the test list the rating performance influence that it will have to the society

Expected software response

Expected software response is required unless otherwise

Identification of critical components

The key error and wrong placed and written components during testing are exposed to open and rectified