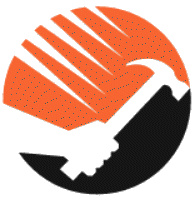
Anwarzada, Akbar

Asaad, Omar

Ewaz Zadeh, Mersad

Nemat, Sepehr



Extrahands Documentation

Table of Contents

[1 OVERVIEW 3](#_Toc15674686)

[2 Design Goals 4](#_Toc15674687)

[3 Branding Guidelines 5](#_Toc15674688)

[3.1 Master logo 5](#_Toc15674689)

[3.2 Typography 5](#_Toc15674690)

[3.3 Primary Color 6](#_Toc15674691)

[3.4 Secondary Color 6](#_Toc15674692)

[4 Tools 7](#_Toc15674693)

[5 Backend 8](#_Toc15674694)

[5.1 Server 8](#_Toc15674695)

[5.2 Languages 10](#_Toc15674696)

[5.3 Technologies 11](#_Toc15674697)

[6 Frontend 14](#_Toc15674698)

[6.1 Languages 14](#_Toc15674699)

[6.2 Bootstrap 15](#_Toc15674700)

[7 Middleware 16](#_Toc15674701)

[8 Use-Case Realizations 16](#_Toc15674702)

[9 Database 18](#_Toc15674703)

[9.1 Data tables 18](#_Toc15674704)

[9.1.1 dbo.question 18](#_Toc15674705)

[9.1.2 dbo.service\_providers\_ref 19](#_Toc15674706)

[9.1.3 dbo.service\_providers 20](#_Toc15674707)

[9.1.4 dbo.services 21](#_Toc15674708)

[9.1.5 dbo.user\_address 22](#_Toc15674709)

[9.1.6 dbo.users 22](#_Toc15674710)

[9.1.7 dbo.verbal\_answer 24](#_Toc15674711)

[9.1.8 dbo.visual\_answer 24](#_Toc15674712)

[9.2 Relation of Tables: 25](#_Toc15674713)

[9.3 VIEWS 26](#_Toc15674714)

[9.3.1 dbo.user\_serviceprovider 26](#_Toc15674715)

[9.4 STORE PROCEDURES 27](#_Toc15674716)

[9.4.1 uspAddAddress: 27](#_Toc15674717)

[9.4.2 uspAddCategories: 28](#_Toc15674718)

[9.4.3 uspAddUser: 28](#_Toc15674719)

[9.4.4 uspGetAllServicesOfProviders: 29](#_Toc15674720)

[9.4.5 uspLogin: 29](#_Toc15674721)

[9.4.6 uspSearchServiceProvider: 30](#_Toc15674722)

[9.4.7 uspService\_Provider: 30](#_Toc15674723)

[9.4.8 UspServices\_ref: 31](#_Toc15674724)

[9.4.9 uspUserInfo: 32](#_Toc15674725)

[10 Main functionalities and final Results: 33](#_Toc15674726)

[11 Difficulties during the project: 36](#_Toc15674727)

[12 Cannot be achieved: 36](#_Toc15674728)

# OVERVIEW

This document describes the architecture and design for the **ExtraHands** Platform being developed by students from **DHBW** (Duale Hochschule Baden-Württemberg). **ExtraHands** is a website built to gather both service providers and client.

**ExtraHands** platform can gather both customers and service providers and enables customers from one side to see different offers, chose the suitable one and get the service they want and from another side, it helps the service providers to understand customers´ problems faster, save operating time, and market their business as well.

This platform should ensure that:

**Customer**

* Will be able to identify their problems easily without going inside any technical details.
* See different choices and offers from one platform.

**Service Provider**

* Will reach more customers and clients.
* Save time and effort to understand customers problem.
* Digitalize their process.

The purpose of this document is to describe the architecture and design of **ExtraHands** platform in a way that addresses the interests and concerns of all major stakeholders. For this application the major stakeholders are:

* **Users and the customer** – they want assurances that the architecture will provide for system functionality and exhibit desirable non-functional quality requirements such as usability, reliability, etc.
* **Developers** – they want an architecture that will minimize complexity and development effort.
* **Project Manager** – the project manager is responsible for assigning tasks and coordinating development work. He wants an architecture that divides the system into components of roughly equal size and complexity that can be developed simultaneously with minimal dependencies. For this to happen, the modules need well-defined interfaces. Also, because most individuals specialize in a skill or technology, modules should be designed around specific expertise. For example, all UI logic might be encapsulated in one module.
* **Maintenance Programmers** –want assurance that the system will be easy to evolve and maintain on into the future.

The architecture and design for a software system are complex and individual stakeholders often have specialized interests.

# Design Goals

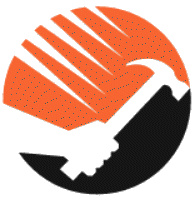
There is no absolute measure for distinguishing between good and bad design. The value of a design depends on stakeholder priorities. For example, depending on the circumstances, an efficient design might be better than a maintainable one, or vise versa. Therefore, before presenting a design it is good practice to state the design priorities. The design that is offered will be judged according to how well it satisfies the stated priorities.

The priorities for the design that follows are:

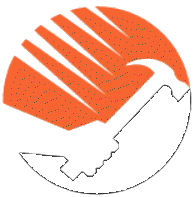
* The design should minimize complexity and development effort for service providers and customers.
* The design should consider the development environment which is one small team with complementary skills. If the components have well-defined interfaces each team can work independently coding to the interfaces of the other components. The concerns of each component should be narrow so that each team member can specialize on a technology or skill.
* The design shouldn’t inhibit reusability. The two previous design goals are more important, but the ability to reuse components is also desirable.

# Branding Guidelines

## Master logo



**Master Logo**



**Transparent Logo**

## Typography

**Font:** Arial

**Paragraph font**

ABCDEFGHIJKLMNOPQRSTUVW

abcdefghijklmnopqrstuvw

1234567890! @#$%^&\*()\_+=”:?><

Regular

Semiotics aesthetic freegan pour-over

jianbing. Artisan blog retro neutra.

*Oblique*

*Semiotics aesthetic freegan pour-over*

*jianbing. Artisan blog retro neutra.*

**Bold**

**Semiotics aesthetic freegan pour-over**

**jianbing. Artisan blog retro neutra.**

## Primary Color

HEX FFFFFF HEX 212121 HEX FF5722

R255 G255 B255 R33 G33 B33 R255 G87 B34

## Secondary Color

HEX 5D5D5D HEX 737373

R93 G93 B93 R115 G115 B115

# Tools

**Visual studio code:**

visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

it is half-way between a text editor and an IDE. There are main reasons that lead us to use it:

* It comes with a built-in support for JavaScript, TypeScript, NodeJS (auto-completion, syntax check, debug, …).
* It has a great ecosystem of plugins for supporting other languages (C, C++, C#, Python, …).
* Visual Studio Code supports many features for JavaScript and Node.js development. The features that ship with the downloaded product are the core features: debugging, IntelliSense, code navigation, etc.

In addition, to these core features, we can install many quality extensions to add features to VS Code for JavaScript development.

**Azure data studio:**

used to manage SQL Server databases and cloud-based Azure SQL Database and Azure SQL Data Warehouse systems. The lightweight software is designed to make routine database development, querying and administration work easier than it is with Microsoft's more-functional SQL Server Management Studio (SSMS) software.

**GitHub desktop:**

to simplify all process and workflow in our GitHub <https://github.com/mersadwm/SrvcWeb>.

GitHub Desktop is an open source Electron-based GitHub app.

It is written in TypeScript and uses React.

It has many awesome features like:

* Attributing commits with collaborators easily
* Checkout branches with pull requests and view CI statuses
* Syntax highlighted diffs
* Expanded image diff support
* Extensive editor & shell integrations
* It’s open source

# Backend

## Server

**Azure**

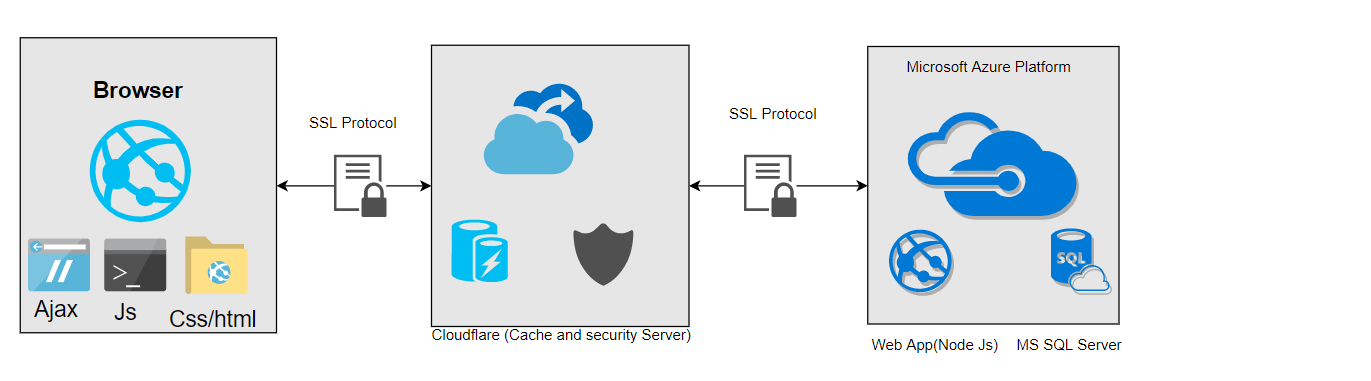
a cloud platform for building, deploying, and managing services and applications.

Azure SQL Database is a general-purpose relational database managed service that enables users to create highly available and high-performance data storage layer for the applications and solutions in Microsoft Azure cloud. SQL Database can be the right choice for a variety of modern cloud applications because it enables users to use powerful functionalities for processing both relational data and non-relational structures such as graphs, JSON, spatial, and XML.

SQL Database enables us to easily define and scale performance. SQL Database is fully-managed service that has built-in high-availability, backups, and other common maintenance operations. Microsoft handles all patching and updating of the SQL and OS code seamlessly and abstracts away all management of the underlying infrastructure.

SQL Database provides built-in business continuity and global scalability features, including:

**Automatic backups:** SQL Database automatically performs full, differential, and transaction log backups of Azure SQL databases to enable us to restore to any point in time



overview architecture ExtraHands

**Point-in-time restores:**

All SQL Database deployment options support recovery to any point in time within the automatic backup retention period for any Azure SQL database.

**Auto-failover groups:**

All SQL Database deployment options allow to use failover groups to enable high availability and load balancing at global scale.

**Zone-redundant databases:**

SQL Database allows to provision premium or business critical databases or elastic pools across multiple availability zones.

**DB WebApp**

**SQL Server**

SQL Server is a relational database management system, or RDBMS.

**Blob storage**

It is a Microsoft's object storage solution for the cloud Blob storage is optimized for storing massive amounts of unstructured data which is data that does not adhere to a data model or definition, such as text or binary data.

Blob storage is designed for Serving images or documents directly to a browser which is used in our platform to make enable users to add their profile pictures.

## Languages

SQL

Java Script

with the advent of Node.js, we can also run JavaScript on the server.

**EJS**

Is a JavaScript templating library used to render html code along with JSON data in an easy and clean way. By using EJS we could directly render pages dynamically using Express and there’s a lot of flexibility you get that way which you wouldn’t have got otherwise with static HTML.

**Json**

a way of storing information in an organized and easy manner. The data must be in the form of a text when exchanging between a browser and a server. You can convert any JavaScript object into JSON and send JSON to the server.

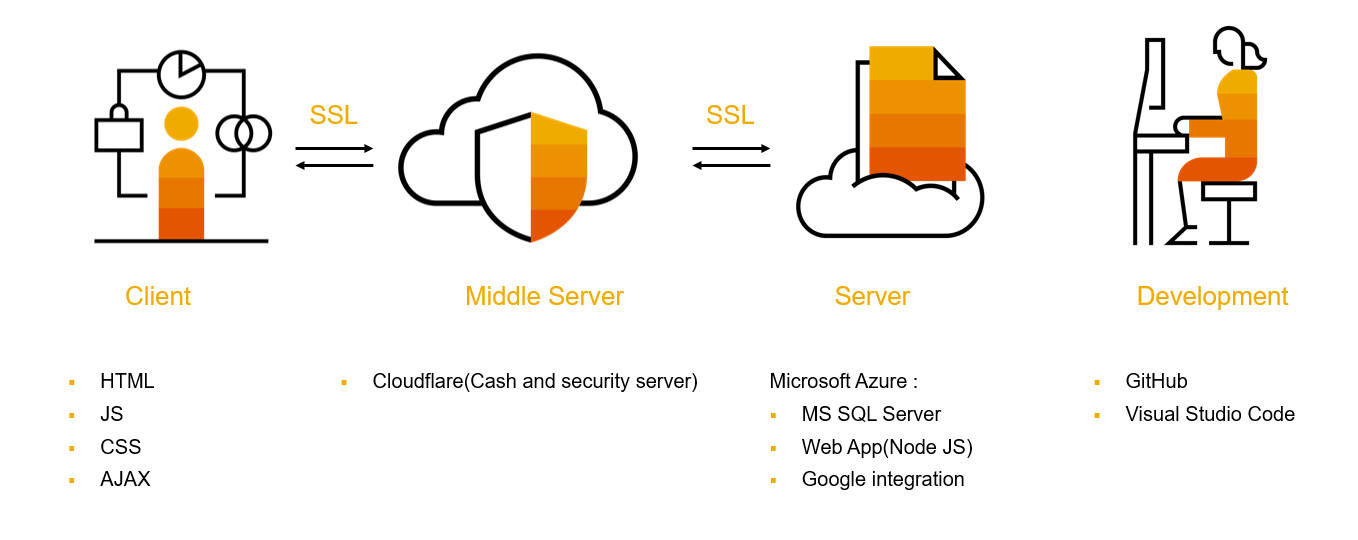
is lightweight and easy-to-use when compared to other open data interchange options.

Faster – The XML software parsing process is slower than JSON. We used JSON because:

* JSON is simple to read and write.
* It also supports array.
* JSON files are more human-readable than XML.

## Technologies

Using ‘state of art technologies’ was one of the main requirements, that’s why we tended to us the most modern and comprehensive technologies and tools in our website because we believe that good website design is based on the combination of design and technology, and it’s important to consider all aspects of the site throughout the development process.



Tools and technology overview

In terms of complexity ExtraHands platform is a simple to med web application which not only contains simple functions but also some challenging aspects which require a lot of search and effort to find and learn so we could connect all the process and fill business requirements we have.

Here are some technologies we used:

**Node.js**

Node.js is a server-side programming framework, built on Chrome's JavaScript engine, which provides event-driven and non-blocking I/O. The coding is done in JavaScript.

The most important advantages of Node include:

* It makes coding in JavaScript for both the client and server side possible.
* It increases the efficiency of the development process as it fills the gap between frontend and backend developers.
* The ever-growing NPM (Node Package Manager) gives us multiple tools and modules to use, thus further boosting our productivity,
* code executes faster than in any other language,

The Node Package Manager, known as an **npm**, allows us to install, update, and use smaller open-source software packages (modules), which means we do not have to write common features from scratch and can avoid new layers of complexity that often come with that territory.

**Express.js**

Express is a minimal and flexible Node.js web application framework that provides a robust set of features to develop web and mobile applications. It facilitates the rapid development of Node based Web applications. Following are some of the core features of Express framework:

* Allows to set up middleware’s to respond to HTTP Requests.
* Defines a routing table which is used to perform different actions based on HTTP Method and URL.
* Allows to dynamically render HTML Pages based on passing arguments to the templates we use.

**Session.js**

The local Storage and session Storage properties allow us to save key/value pairs in a web browser.

The session Storage object stores data for only one session (the data is deleted when the browser tab is closed).

We have used sessions to communicate data to middleware that’s executed later, or to retrieve it, on subsequent requests.

**Cookie parser**

Cookies used for tracking visitors to a website including node.js projects made with express.js. In express the usual choice for parsing cookies is the cookie-parser module. Cookie parser used in our website to check if the cookie is expired and if is it works, and it will be saved in user browser cache.

**Body parser**

It takes the page/form and send it to server used for post request.

Body-parser extract the entire body portion of an incoming request stream and exposes it on req.body.

**Mssql.js**

Driver connector to SQL server database, we use SQL server in an enterprise level database.

**Passport.js**

Passport is authentication middleware for Node.js. As it’s extremely flexible and modular, Passport can be unobtrusively dropped into any Express-based web application. A comprehensive set of strategies supports authentication using a username and password, Facebook…

**Google auth**

Module of the passport.js to connect with google account using google API.

**Local auth**

Module of the passport.js to connect**.**

**Debug.js**

Module for debugging**.**

**Logger (Morgan)**

Logging middleware for node.js http apps.

**Path.js**

Module of node.js it connects different texts and create URLs.

**Defined**

Error handling framework**.**

**Favicon.js**

Module for generating favicons and their associated files.

**Formidable**

Focused on uploading and encoding images and videos. It has been battle-tested against hundreds of GB of file uploads from a large variety of clients and is considered production-ready.

**Crypto**

Random values generator

# Frontend

## Languages

**Java Script**

used to add the interactivity and dynamic features to our website .

**jQuery**

a JavaScript library, jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is easier to use compared to JavaScript and its other JavaScript libraries.

**Ajax**

technique for web applications to send and receive data from server asynchronously, without impacting the other contents or reloading the page. AJAX uses XMLHttpRequest object to communicate with the server, JavaScript/DOM to make the requests and XML as a format to transfer data.

**Html**

Hypertext Markup Language (HTML) is used to describe and define the content of a webpage.

**CSS**

while HTML is used to structure a web document (defining things like headlines and paragraphs, and allowing you to embed images, video, and other media), CSS comes through and specifies your document’s style—page layouts, colors, and fonts are all determined with CSS.

CSS Provides Efficiency in Design and Updates with CSS, we can create rules, and apply those rules to many elements within the website. This approach offers many advantages when site-wide changes are required by a client. Since the content is completely separated from the design, we can make those changes in our Style Sheet and have it effect every applicable instance.CSS Use Leads to Faster Page Downloads.

## Bootstrap

a collection of tools for creating a websites and web applications

It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions.

We preferred Bootstrap Framework because it is:

* Easy to get started.
* Great grid system.
* Base styling for most HTML elements (Typography, Code, Tables, Forms, Buttons, Images, Icons).
* Extensive list of components.
* Bundled JavaScript plugins.

# Middleware

**cache server**

We used Cloudflare platform to make our site faster and more secure.

Thanks to the multiple caching setting available in Cloudflare, the performance of the will be better because of eh automatic caching that Cloudflare offers.

**SSL**Secure Sockets Layer (SSL) is a standard security technology for establishing an encrypted link between a server and a client—typically a web server (website) and a browser, we used edge SSL between user and cache server and origin SSL between cache server and application server.

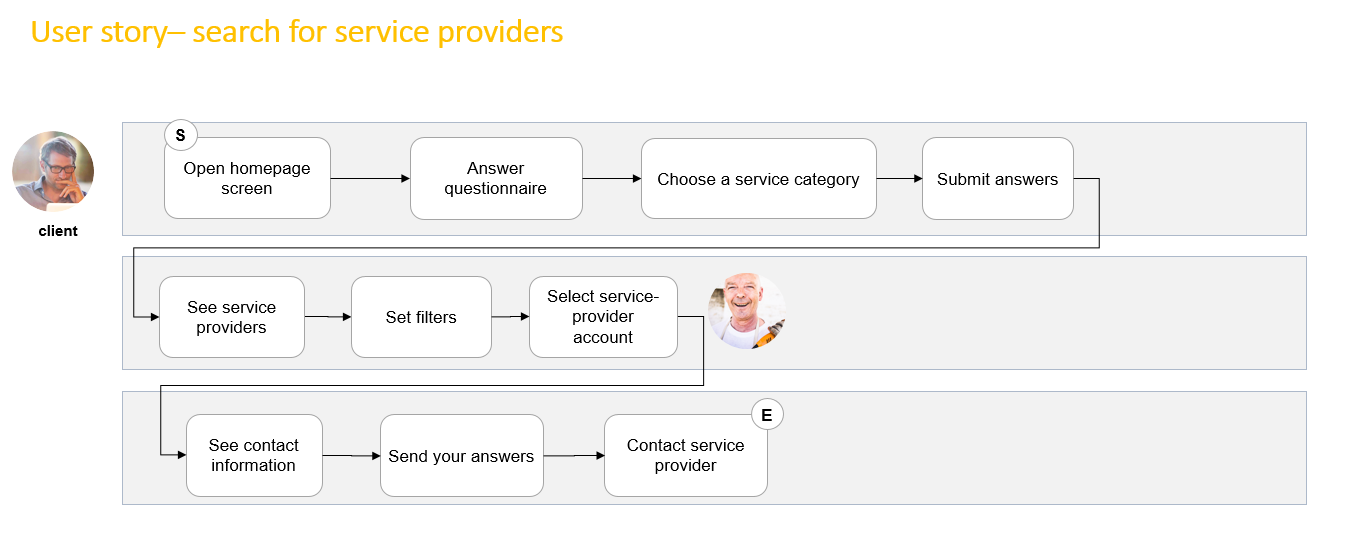
# Use-Case Realizations

**Login**

User credentials are authenticated, and user is redirected to application home page.



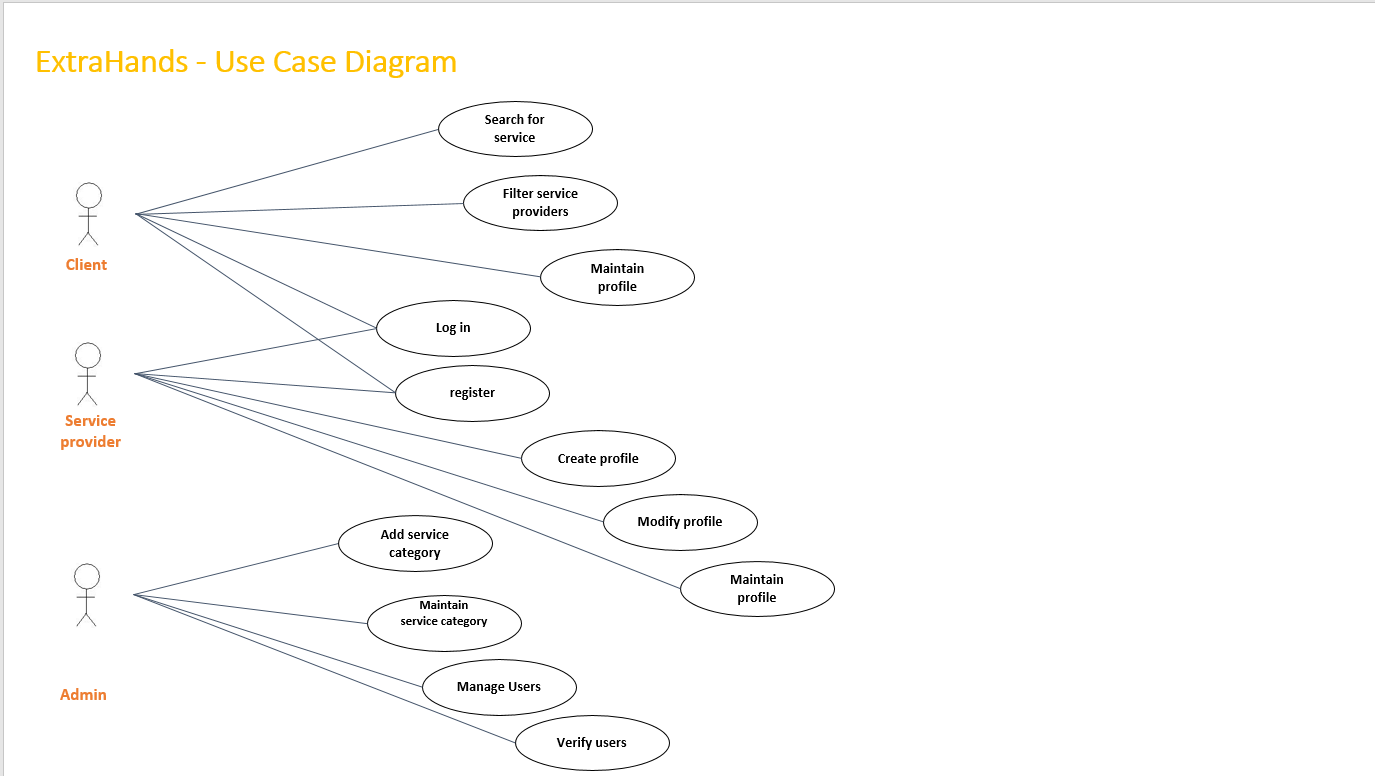
**User story – search for service provider**



**User Story – Registration**

**Use Case – diagram**

**Use Case – diagram**



# Database

as mentioned before we use **MySQL** as a database technology, and before designing databases and tables, our team analyze what data would be collected and expected uses of the data in the system.

## Data tables

Below tables were identified throw several processes and were created and used in the system respectively.

### dbo.question

**Description:**

In the Question table we record questions which our clients may have and can use these data to find related needs.

Clients can write extra information or question for the needed services in the user interface of the system.

**Columns:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| questions | | | NVARCHAR | 255 |
| Parent\_key | | | NVARCHAR | 255 |
| Moreinfo | | | NVARCHAR | 255 |
| Question\_key | | | NVARCHAR | 255 |
| invisualized | | | BIT |  |

**Depending Objects:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.verbal\_answer | | | Table | 1 |
| dbo.visual\_answer | | | Table | 1 |

### dbo.service\_providers\_ref

**Description:**

This table holds the relation between users and services, and it was identified during normalization of the hole database, which is acts as the bases of our services.

**Columns:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| sp\_id | | | INT |  |
| user\_id | | | INT |  |
| service\_id | | | INT |  |
| more\_info | | | NVARCHAR | 255 |

**Depending Objects:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.services | | | Table | 1 |
| dbo.users | | | Table | 1 |
| dbo.user\_serviceprovider | | | View | 1 |

### dbo.service\_providers

**Description:**

Service providers (sps) are one of the main parts of the system. In the system sps are extended from users table with more information in service\_providers table which are the fundamental and essential information of the sps.

**Columns:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| sp\_id | | | INT |  |
| user\_id | | | INT |  |
| service\_id | | | INT |  |
| company\_name | | | NVARCHAR | 40 |
| address\_sp | | | NVARCHAR | 40 |
| telephone | | | NVARCHAR | 40 |
| website\_link | | | NVARCHAR | 40 |
| contact\_email | | | NVARCHAR | 50 |
| Zip | | | INT |  |
| city | | | NVARCHAR | 50 |
| about\_me | | | NVARCHAR | 255 |
| Verified | | | BIT |  |
| login\_user | | | NVARCHAR | 50 |

**Depending Objects:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.services | | | Table | 2 |
| dbo.users | | | Table | 1 |
| dbo.user\_serviceprovider | | | View | 1 |

### dbo.services

**Description:**

The core of a business is the type of services which it provides, and all our business-related services are stored in services table

**Columns:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| sp\_id | | | INT |  |
| title | | | NVARCHAR | 255 |
| category | | | NVARCHAR | 255 |
| super\_cat | | | NVARCHAR | 50 |
| name | | | NVARCHAR | 50 |
| pic | | | NVARCHAR | 50 |
| info | | |  | 50 |

**Depending Objects:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.service\_prividers\_ref | | | Table | 1 |
| dbo.user\_serviceprovider | | | View | 1 |

### dbo.user\_address

**Description**:

User\_address table was found as an essential table for our system, which we extend the information of clients to be able to serve them in better manner.

**Columns:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| address\_id | | | INT |  |
| login\_name | | | NVARCHAR | 40 |
| address\_1 | | | NVARCHAR | 120 |
| address\_2 | | | NVARCHAR | 120 |
| address\_3 | | | NVARCHAR | 120 |
| city\_name | | | NVARCHAR | 50 |
| state\_name | | | NVARCHAR | 20 |
| Country | | | NVARCHAR | 20 |
| PLZ | | | INT |  |

**Depending Objects:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.users | | | Table | 1 |

### dbo.users

**Description:**

All basic information of our platform users is stored in the users table, which is later widened into other tables based on the different types of the user which we have.

In order that our customers feel secure, passwords are stored encrypted using hash bytes and salt.

Combination of hash bytes (type SHA\_512) and salt makes it harder to decrypt the data, salt is a randomly generated unique string using (UNIQUEIDENTIFIER) which is used in enterprises for password encryption as standard.

**Columns:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| user\_id | | | INT |  |
| login\_name | | | NVARCHAR | 40 |
| password\_hash | | | HASHBYTES | 64 |
| first\_name | | | NVARCHAR | 40 |
| last\_name | | | NVARCHAR | 40 |
| Salt | | | UNIQUEIDENTIFIER |  |
| email | | | NVARCHAR | 50 |
| admin\_rights | | | BIT |  |
| profile\_pic\_url | | | NVARCHAR | 255 |

**Depending Objects:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.user\_address | | | Table | 1 |
| dbo.service\_providers | | | Table | 1 |
| dbo.service\_prividers\_ref | | | Table | 1 |
| dbo.user\_services | | | Table | 1 |
| dbo.user\_serviceprovider | | | View | 1 |

### dbo.verbal\_answer

Description:

For user friendliness of our platform user’s needs are generated in two types of Descriptive and Visual demonstration, which the descriptives are stored in verbal\_answer table.

**Columns:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| question\_key | | | VARCHAR | 255 |
| txt | | | VARCHAR | 255 |
| id | | | INT |  |
| next\_slide\_key | | | VARCHAR | 255 |

**Depending Objects:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.question | | | Table | 1 |
| dbo.visual\_answer | | | Table | 1 |

### dbo.visual\_answer

**Description:**

For user friendliness of our platform user’s needs are generated in two types of Descriptive and Visual demonstration, which the visuals are stored in visual\_answer table.

**Columns:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| question\_key | | | VARCHAR | 255 |
| image\_description | | | VARCHAR | 255 |
| id | | | INT |  |
| next\_slide\_key | | | VARCHAR | 255 |
| image\_caption | | | VARCHAR | 255 |
| image\_url | | | VARCHAR | 255 |

**Depending Objects:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.question | | | Table | 1 |
| dbo.verbal\_answer | | | Table | 1 |

## Relation of Tables:

## VIEWS

### dbo.user\_serviceprovider

**Description:**

Our aim of the project is not only creating a functional one, but one that is compatible with current standards.

Despite that our project isn’t faced much traffic yet, but we still wanted to design it in the best manner and considered the performance of our database as one of our priorities.

One method to tune a database for better performance is to use materialized views, this view is used for joining different information related users and services and service providers.

**Columns:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Column Name | Description | Datatype | Length | Allow Nulls | Default | Formula |
|  | **user\_id** | Summary | int | 4 |  |  |  |
|  | **first\_name** | Summary | nvarchar | 40 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **last\_name** | Summary | nvarchar | 40 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **email** | Summary | nvarchar | 50 |  |  |  |
|  | **company\_name** | Summary | nvarchar | 40 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **address\_sp** | Summary | nvarchar | 40 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **telephone** | Summary | nvarchar | 40 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **website\_link** | Summary | nvarchar | 40 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **contact\_email** | Summary | nvarchar | 50 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **zip** | Summary | int | 4 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **city** | Summary | nvarchar | 50 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |
|  | **about\_me** | Summary | nvarchar | 255 | C:\Users\akbar\OneDrive\Documents\Document! X\Extrahands4.me\template\packages\core-db\images\tick.gif |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Name | | | **Data Type** | **Size** |
| User\_id | | | INT |  |
| first\_name | | | NVARCHAR | 40 |
| last\_name | | | NVARCHAR | 40 |
| email | | | NVARCHAR | 50 |
| company\_name | | | NVARCHAR | 40 |
| address\_sp | | | NVARCHAR | 40 |
| telephone | | | NVARCHAR | 40 |
| website\_link | | | NVARCHAR | 40 |
| contact\_email | | | NVARCHAR | 50 |
| Zip | | | INT |  |
| city | | | NVARCHAR | 50 |
| about\_me | | | NVARCHAR | 255 |

**Objects that depends on dbo.user\_serviceprovider**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.uspSearchServiceProvider | | | Stored Procedure | 1 |

**Objects dbo.user\_serviceprovider depends on**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction | Description | | | | Data Type | Size |
| @pLogin | In | Summary | | | | nvarchar | 50 |
| @pAddress1 | In | Summary | | | | nvarchar | 120 |
| @pAddress2 | In | Summary | | | | nvarchar | 120 |
| @pAddress3 | In | Summary | | | | nvarchar | 120 |
| @pPLZ | In | Summary | | | | int | 4 |
| @pCity\_name | In | Summary | | | | nvarchar | 50 |
| @pState\_name | In | Summary | | | | nvarchar | 20 |
| @pCountry | In | Summary | | | | nvarchar | 20 |
| @responseMessage | Out | Summary | | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | | | | int | 4 |
| Database Object | | | **Object Type** | **Dep Level** |
| dbo.service\_providers | | | Table | 1 |
| dbo.users | | | Table | 1 |

## STORE PROCEDURES

### uspAddAddress:

**Description:**

This procedure is used in the system to insert below mentioned parameters into user\_address table.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter | Direction |  | | | Description | | | Data Type | Size |
| @pLogin | In |  | | | Summary | | | nvarchar | 50 |
| @pAddress1 | In |  | | | Summary | | | nvarchar | 120 |
| @pAddress2 | In |  | | | Summary | | | nvarchar | 120 |
| @pAddress3 | In |  | | | Summary | | | nvarchar | 120 |
| @pPLZ | In |  | | | Summary | | | int | 4 |
| @pCity\_name | In |  | | | Summary | | | nvarchar | 50 |
| @pState\_name | In |  | | | Summary | | | nvarchar | 20 |
| @pCountry | In |  | | | Summary | | | nvarchar | 20 |
| @responseMessage | Out |  | | | Summary | | | nvarchar | 250 |
| @RETURN\_VALUE | Return Value |  | | | Summary | | | int | 4 |
| Parameter | | | **Direction** | **Direction** | | **Size** |
| @pLogin | | | In | NVARCHAR | | 50 |
| @pAddress1 | | | In | NVARCHAR | | 120 |
| @pAddress2 | | | In | NVARCHAR | | 120 |
| @pAddress3 | | | In | NVARCHAR | | 120 |
| @pPLZ | | | In | INT | |  |
| @pCity\_name | | | In | NVARCHAR | | 50 |
| @pState\_name | | | In | NVARCHAR | | 20 |
| @pCountry | | | In | NVARCHAR | | 20 |
| @responseMessage | | | Out | NVARCHAR | | 250 |
| @RETURN\_VALUE | | | Return Value | INT | |  |

**Depending Objects:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |
| --- | --- | --- |
| Database Object | **Object Type** | **Dep Level** |
| dbo.user\_address | Table | 1 |
| dbo.users | Table | 2 |

### uspAddCategories:

**Description:**

This procedure is used in the system to insert below mentioned parameters into categories table.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | **Direction** | **Direction** | **Size** |
| @pCategorie | In | NVARCHAR | 50 |
| @pDescription | In | NVARCHAR | 50 |
| @pParentID | In | INT |  |
| @RETURN\_VALUE | Return Value | INT |  |

### uspAddUser:

**Description:**

This procedure is used in the system to add new users into uses table.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | **Direction** | **Direction** | **Size** |
| @pLogin | In | NVARCHAR | 50 |
| @pPassword | In | NVARCHAR | 50 |
| @pEmail | In | NVARCHAR | 50 |
| @responseMessage | Out | NVARCHAR | 250 |
| @RETURN\_VALUE | Return Value | INT |  |

**Depending Objects:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |
| --- | --- | --- |
| Database Object | **Object Type** | **Dep Level** |
| dbo.users | Table | 1 |

### uspGetAllServicesOfProviders:

**Description:**

This procedure is used in the system to show all service providers and services which they provide.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | **Direction** | **Direction** | **Size** |
| @RETURN\_VALUE | Return Value | INT |  |

**Depending Objects:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |
| --- | --- | --- |
| Database Object | **Object Type** | **Dep Level** |
| dbo.service\_prividers\_ref | Table | 1 |
| dbo.service\_providers | Table | 2 |
| dbo.services | Table | 2 |
| dbo.user\_serviceprovider | View | 1 |
| dbo.users | Table | 2 |

### uspLogin:

**Description:**

This procedure receive login\_name and password of the user and check in database if the user and password match with stored information, if the information provided is correct then user can login in and if password is wrong user receive (incorrect password) message and if login and or password both are wrong user receive (Invalid login) message.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | **Direction** | **Direction** | **Size** |
| @pLoginName | In | NVARCHAR | 254 |
| @pPassword | In | NVARCHAR | 50 |
| @responseMessage | Out | NVARCHAR | 250 |
| @RETURN\_VALUE | Return Value | INT |  |

**Depending Objects:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |
| --- | --- | --- |
| Database Object | **Object Type** | **Dep Level** |
| dbo.users | Table | 1 |

### uspSearchServiceProvider:

**Description:**

This procedure searches all service providers list which can provide a certain service.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | **Direction** | **Direction** | **Size** |
| @pServices | In | NVARCHAR | 50 |
| @RETURN\_VALUE | Return Value | INT |  |

**Depending Objects:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |
| --- | --- | --- |
| Database Object | **Object Type** | **Dep Level** |
| dbo.service\_prividers\_ref | Table | 1 |
| dbo.service\_providers | Table | 2 |
| dbo.services | Table | 2 |
| dbo.user\_serviceprovider | View | 1 |
| dbo.users | Table | 2 |

### uspService\_Provider:

**Description:**

This procedure is used to insert information of the service provider in service\_providers table.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | **Direction** | **Direction** | **Size** |
| @plogin\_user | In | NVARCHAR | 50 |
| @pcompany\_name | In | NVARCHAR | 40 |
| @paddress\_sp | In | NVARCHAR | 40 |
| @ptelephone | In | NVARCHAR | 40 |
| @pwebsite\_link | In | NVARCHAR | 40 |
| @pcontact\_email | In | NVARCHAR | 50 |
| @pzip | In | INT |  |
| @pcity | In | NVARCHAR | 50 |
| @pabout\_me | In | NVARCHAR | 255 |
| @pverified | In | BIT | 1 |
| @RETURN\_VALUE | Return Value | INT |  |

**Depending Objects:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |
| --- | --- | --- |
| Database Object | **Object Type** | **Dep Level** |
| dbo.service\_providers | Table | 1 |
| dbo.users | Table | 2 |

### UspServices\_ref:

**Description:**

This procedure gets login name of service provider and service name which a service provider can provide and convert the names into user\_id and service\_id and inserts them into service\_prividers\_ref table which is relation between service providers and services tables.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | **Direction** | **Direction** | **Size** |
| @pLogin | In | NVARCHAR | 50 |
| @pservice | In | NVARCHAR | 50 |
| @pmore\_info | In | NVARCHAR | 50 |
| @RETURN\_VALUE | Return Value | INT |  |

**Depending Objects:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |
| --- | --- | --- |
| Database Object | **Object Type** | **Dep Level** |
| dbo.service\_providers\_ref | Table | 1 |
| dbo.services | Table | 1 |
| dbo.users | Table | 1 |

### uspUserInfo:

**Description:**

This procedure gets login name and password of a user and shows all related information.

**Parameters:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| Parameter | **Direction** | **Direction** | **Size** |
| @pLogin | In | NVARCHAR | 50 |
| @pPassword | In | NVARCHAR | 50 |
| @RETURN\_VALUE | Return Value | INT |  |

**Depending Objects:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Direction | Description | Data Type | Size |
| @pLogin | In | Summary | nvarchar | 50 |
| @pAddress1 | In | Summary | nvarchar | 120 |
| @pAddress2 | In | Summary | nvarchar | 120 |
| @pAddress3 | In | Summary | nvarchar | 120 |
| @pPLZ | In | Summary | int | 4 |
| @pCity\_name | In | Summary | nvarchar | 50 |
| @pState\_name | In | Summary | nvarchar | 20 |
| @pCountry | In | Summary | nvarchar | 20 |
| @responseMessage | Out | Summary | nvarchar | 250 |
| @RETURN\_VALUE | Return Value | Summary | int | 4 |

|  |  |  |
| --- | --- | --- |
| Database Object | **Object Type** | **Dep Level** |
| dbo.user\_address | Table | 1 |
| dbo.users | Table | 1 |

# Main functionalities and final Results:

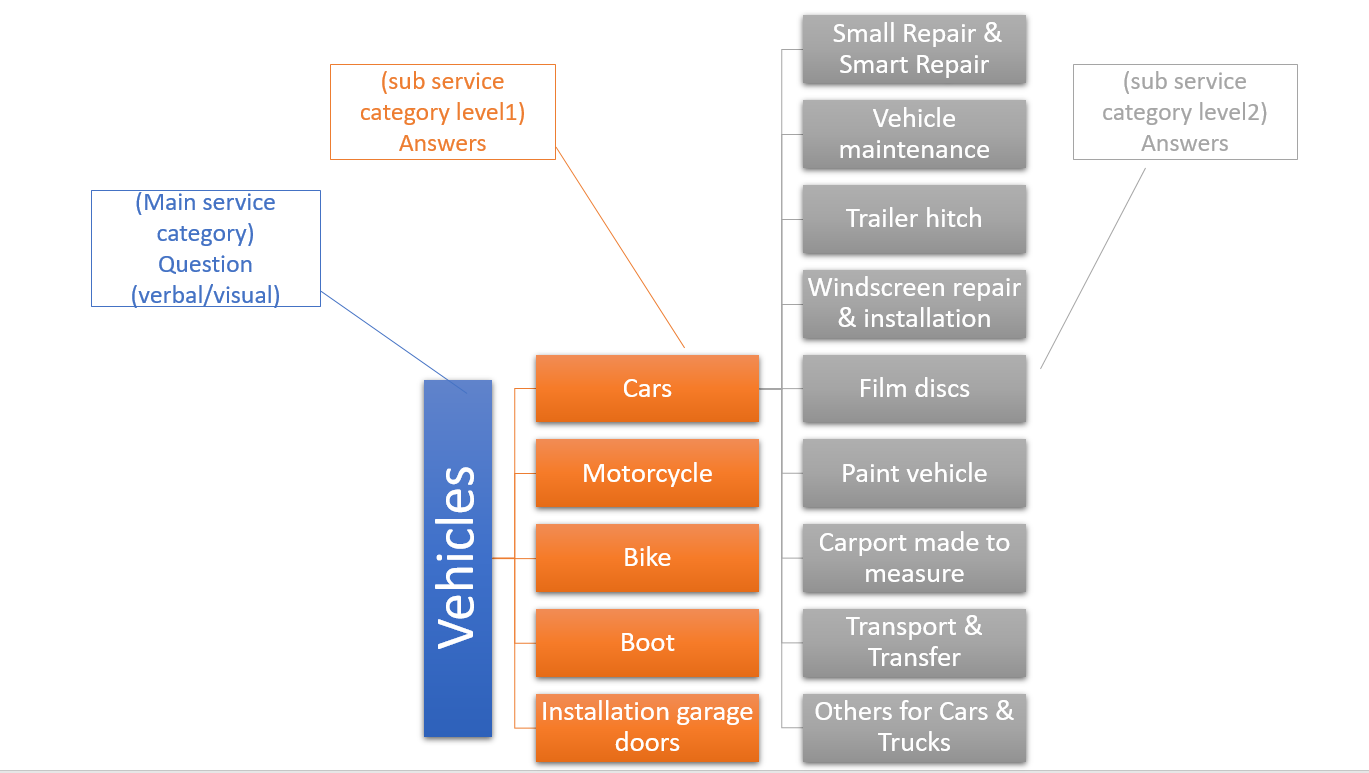
We could achieve our scope and fulfill our requirements when it came to both business users (service providers and clients), we also make sure to use the best technologies, a defined architecture and to enhance functionalities not only for business users but also for administrators and maintain developers.

**The Questionnaire:**  
one of the main functionalities on our website, which enable the clients to define their problem and easily find their appropriate service provider.

the questionnaire contains both visual and verbal questions and answers, admins can maintain the questionnaire (adding new services, answers, question).

The questionnaire is connected to the search page where the user can see the available service providers for his specific problem.

all the data inside the questionnaire coming from JSON file using API and questionnaire reads those data and show it to us.

Users can also see the available service providers by clicking on a specific service category or to see all the service providers, the search page is connected to both services list and the questionnaire.  
  
tow concepts need to be clarified in the questionnaire:   
questions are the main service categories which have service categories for example Vehicles is our main service category which contains sub-service categories like (Cars, Motors ….) ‘the answers in our case’.  
  
**Users management system and registration**  
 ExtraHands platform contains three main users:

**questionnaire structure**

**Client**: they can register to the website using their email or using Google account, after registration the user can add his data to his profile the website will show the user an error in case he missed or forgot to fill his information. Uploading a profile picture is also possible.

**Service** **provider**: after registration the user can upgrade his status to be a service provider where he needs to enter more data about his company and the services he provides, service providers can add many services in their profile.

Service providers need to contact with ExtraHands administrators to be verified. Uploading profile pictures, a code will be generated for each profile picture to save the picture so other users will not have access to the file

**Administrators:** administrators are the one responsible for maintaining the website, users and services. The main functionalities for admins are:

* **Add questions:** add question (main services) to the questionnaire.
* **Update questions**: update/change question(main services) in the questionnaire.
* **View all questions:** see all available questions and answers.
* **Add visual answers:** Add a picture to the question/answer.
* **Add verbal answers:** add a question (sub category).
* **Update answers:** update/change answers (sub services).
* **Upgrade users:** upgrade users to admins or clients to service providers.

In our user management system, we managed to have all the data in normalized tables**.**

**Contact service providers:**

As mentioned the user can send an email to the service providers or call them, when the user clicks on ‘’send an email’’, this will generate an email with all the service provider and client information inside.

**Search page:**shows the service providers available on the website, register users can see the contact information for each service provider and send an email to the service provider or call him.  
The service providers profiles will be shown on the search page from the most to the less relevant depending in your input.

If the user is not registered, he will be able to see the service provider picture and name, but he needs to register to see the email and the phone number.

**For developers**

* All the data could be a json file, using API for example <https://extrahands4.me/services/questionnaire/raw> . So, they can easily use the data for different purposes.
* We use a well-structured architecture, the website organized in routs to pass all the functionalities, we organized the router and we created controllers where we have the functionalities.
* Adding more authentication methods to the website is easy.
* Hight security and cookies on our website are encrypted.
* The connection to the database is stable.
* SQL code and the database, we have user backups, so the admin can easily restore the user data in another database. Complete database setup.
* All videos and media are saved in one folder.
* Html files are in views we have partial views.

# Difficulties during the project:

During our project, we had many difficulties, one of the main difficulties was our time since we are a student and we need to attend classes and at the same time do other projects. Our business requirements were challenging and with our basic knowledge in web programming, we struggled to find the right and the best technologies to achieve our scope and requirements.  since many things need to be created from scratch, most of the functionalities need to be written many times to make sure we have a better code quality and to enhance the functionalities.

questionnaire needed to be developed from scratch, we did not find any similar technology which we can reuse.

# Cannot be achieved:

As mentioned we were able to realize our scope and the main functionalities, many other functionalities we could not accomplish which we think could enhance the process on the future are:

* add filter criteria to search field (ZIP, City, name …)
* Add a star rating system to service providers (out of our scope, but we believe it will be a good feature to have).

# User Guide

In GitHub there are videos explaining the process and the functionalities in the website for all users.