VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

B. Tech., Fall Semester 2021-2022

Online Service Center

Course Faculty

Prof. Braveen M

Batch Members

1. Lalith Kanakamedala 20BCE1491
2. Chalasani Yogeshwar Sai 20BCE1531
3. Ambati Sesha Sai Sahithya 20BCE1605

Table of Contents

Abstract

Due to the continuous lockdowns and COVID situations in the outside world most of our works has to be done online using different electrical devices as they are being used continuously for a much longer time there is a high chance there may be some problem after some time. So to solve these problems people cannot move out to showroom like before due to many COVID restrictions and health issues. So our application will help such people by providing the solution for various kinds of problems in a variety of electronic devices. People not only solve their problems but they can even buy and sell their products in this application. We are planning to design an android app which consist of features mentioned in the above statement. Along with the solution to the problems and buy or sell products. Our application also provides the GPS location of the nearby service center to the customer.

Introduction

In today's fast-growing world, people cannot waste a lot of time on resolving the service based issues on electronic devices. They must be able to resolve these issues accurately and effectively. Online Service Center App is a comprehensive and fully integrated service which helps people to solve their problems. The main purpose or the motive of this app is to save the time of the people. The project is on Online Service Center. There are service centers outside but an online service center is something new which is being habituated to people from the past 5 to 8 years. People are not showing up interest to go to the service centers or showrooms to get their device repaired. Even the people now a days are not even going to the showroom to but new devices. This led to the development of many companies such as Amazon, Flipkart etc. Mainly in these times where COVID hit the whole world and people are afraid to come in contact with other people or go to the service centers located nearby to solve the problems facing with their electronic devices. This product helps the people to solve their problems just by sitting in their home or office. They should initially sign in to the app give their problem and follow the steps provided in the solution. If they are not happy with the given solution, then we also provide with the nearby location of the service centers so that they can resolve their device issue there and they can even get the discount by using the code displayed in the settings of their account. There are some new features in our system where people can exchange their old devices and get them replaced with the used ones or new ones by paying a valid amount after deducting the price for their damaged device. The app makes sure the customer information and details are stored safe and there will be no inconvenience.

Software Requirements Specification

Functional Requirements:

- **Login and Sign-up**
  - ★ Professional and easily accessible login page is required.
  - ★ Valid login credentials are to be provided by user/customer to use the app. Incase invalid credentials are given the user/customer is asked to re-enter the login password.
  - ★ The user should give correct details like phone number so that it would be easy to communicate.
  - ★ The customer is requested to provide his locality during signup so that the app can directly show the nearest stores.

- **Selection of the type of device and problem the customer is facing**
  - ★ Customer should provide with correct location for later usage.
  - ★ Customer should specify the device model correctly to get a correct solution.
  - ★ Customer should know clearly about the problem so that he can search easily without any confusion because the system has solutions for hundreds of problems.

- **If not satisfied with the solution check nearby service centers**
  - ★ Customer should provide his location correctly to find the nearby service centers.
  - ★ Customer should choose one of the nearby service center and go for the repair of his device.
  - ★ The customer should not forget about the promo code given in his settings for additional discount at the service center.

- **Exchange of devices for reasonable prices.**

  ★ The customer's device should meet the exchange options.

  ★ The customer should have enough money to buy the new device or a used

  one after exchanging the old piece.

Non-Functional Requirements:

Performance Requirements:

The software should have high performance and low failure rates. The software should be user friendly and basic so that all the variety of customers who use it should feel the service good and should be able to use the app frequently. The hardware and software should be able to transmit/receive data from databases with high baud rates, ranging from Mbps to Gbps. Machines should have all recent Android updates installed, and have their security not compromised by viruses. It should be made sure that the Androids firewall is always on. Machines must have firewalls installed and active virus scanning software in usage.

Safety Requirements:

- The database should be managed properly.
- The data of the customers should be stored safely.
- The customer phone numbers should not be accessible to all the management staff.
- The customers should not disclose their credentials to anybody as the app may involve the latest transactions which include the bank or card details of the customer.
- The devices provided in exchange or new devices bought by the customer should be packed and sealed properly and they should be delivered in one piece.
- The service centers provided in the system should be checked properly so that the customers will be treated properly in the service centers also.

Security Requirements:

All data receiving and transmissions should be done using FOSS TLS or higher encryption, in order to keep the patient's private medical and social security information out of the wrong hands. The FOSS TLS software must be Software Requirements Specification for Online Service Center Page 12 inspected, in order to verify if recent security exploits/hacks of the system are patched. In addition, all computers must have firewalls, and be operating on a LAN internet connection, not a WiFi connection. Moreover, all computers must have all recent Windows updates installed, and must have solid anti-virus software. Also, the user-permissions system mentioned in §2.3 will be implemented. Before any user can access the system, they shall be required to input a company username, an ID number, and a password. Each password shall be required to be between 8-12 characters in length and shall be required to contain at least one capital letter, one number, and one special character. Passwords will need to be changed every half-year, with a unique password.

Software Quality Attributes:

Flexibility, reusability, robustness, and maintainability of the Supermarket management system should be maximized. The customers should be easily accessible to the site for their needs.

Structure of the Project

- First, the user creates an account for himself and logs in to the website using valid login credentials. He also provides his place and city during registration.
- Then he selects his device. He can choose either a mobile phone or a laptop.
- He will choose whether he wants to search for a solution or to replace/buy a new phone.
- Now, he will choose the model of the mobile/laptop which he is using currently.
- Then he will choose the problem that he is facing with his current mobile. If that problem is already known to us, then the website will display the solution to that problem so that he can follow the steps as shown.
- If he is not satisfied with the solution that is provided by the website, then he can view all the service centers that are available around him. Our website displays the exact locations of the service centers that are located in the place that he provided during the registration.
- All the latitudes and longitudes of these locations are stored in the database. So according the place and city selected by the user during the registration, the locations are displayed. By clicking on the location the user will get to know more about the store and the directions from his place to the service center.
- Also the user can logout from his account when he is done using our website. The user information is kept secured in the database.

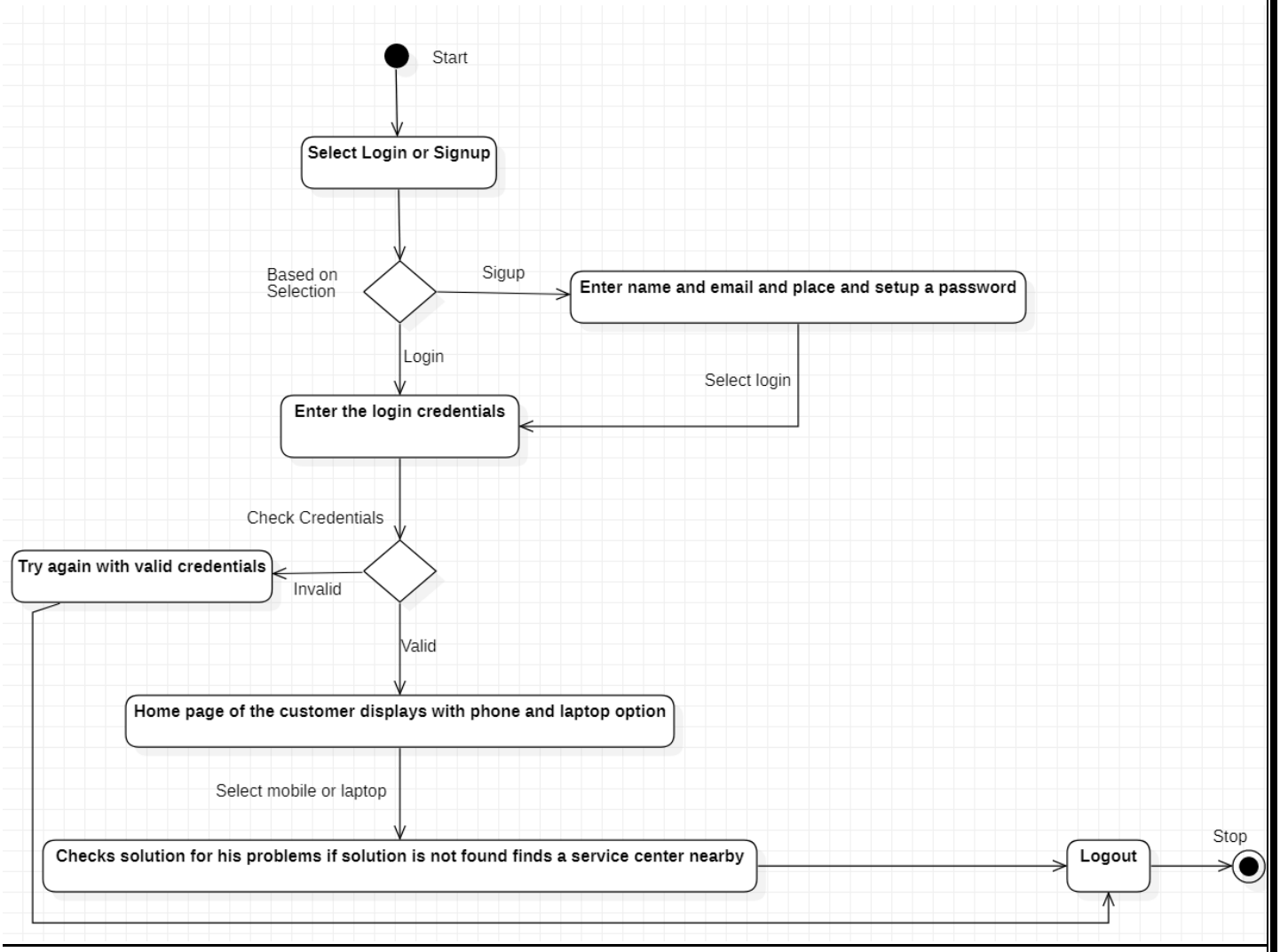
Planning and Scheduling

WBS Number	Task Name	WBS Description	Must Start by Date	Must End by Date	*Task Level of Effort (in hours)*	Task Duration (in days)	Predecessor (Finish-to-Start Dependency) (comma separated)	Resource Name(s)	Author (of Item for follow-on questions)
1	**Initiation Phase**								
1.1	Requirements analysis	All the customers requirements will be explained to the software Developer. According to the given specifications the project team initiates their Project.	**04-08-2021**	**07-08-2021**	**1**	**3**		**Customer**	Sai Sahithya
1.2	Template Design	According to the customer's requirements the developer designs a basic template which briefs about all the steps that will be taken by the project developer.	**04-08-2021**	**08-08-2021**	**2**	**4**			Sai Sahithya
1.3	Designing SRS	Based on the template designed by them, the in-depth descriptions of the software that will be developed will be incuded in the Software Requirements Specifications(SRS).	**11-08-2021**	**15-08-2021**	**3**	**4**	**1.1, 1.2**		Sai Sahithya
2	**Planning Phase**								
2.1	Project outcome	This system provides an efficient way of providing the solutions to different regular problems faced by the users with their electronic devices.	**11-08-2021**	**15-08-2021**	**1**	**4**			All
2.2	Appropriate Model Selection	The team selects an appropriate model that is most suitable for their website. They have various options including Waterfall Model, Spiral Model, V-Model, Agile Model etc.	**18-08-2021**	**20-08-2021**	**2**	**2**			All
2.3	Work Breakdown Structure	WBS is prepared by the Project manager in order to simplify the complex project into smaller chunks of work and divide the work among his employees.	**18-08-2021**	**20-08-2021**	**4**	**2**			All
2.4	Project Approval	The Project manager approves the project after going through all the customers requirements.	**25-08-2021**	**25-08-2021**	**1**	**0**			
3	**Development Phase**								
3.1	Design the Database	Create a database using MySQL in order to store the required information like user details, problems, solutions, device details and locations of service centers.						**MySQL**	Yogeshwar Sai
3.1.1	Insert appropriate details	The back end developer updates the information such as new problems and solutions, new locations of service centers into the database. The user details are automatically updated into the database when he registers.	**26-08-2021**	**29-08-2021**	**6**	**3**		**MySQL**	Yogeshwar Sai
3.1.2	Link Database to a Web Page	In order to display the information stored in the database to the users we link the database to the website.	**30-08-2021**	**31-08-2021**	**1**	**1**		**PHP, MySQL**	Yogeshwar Sai

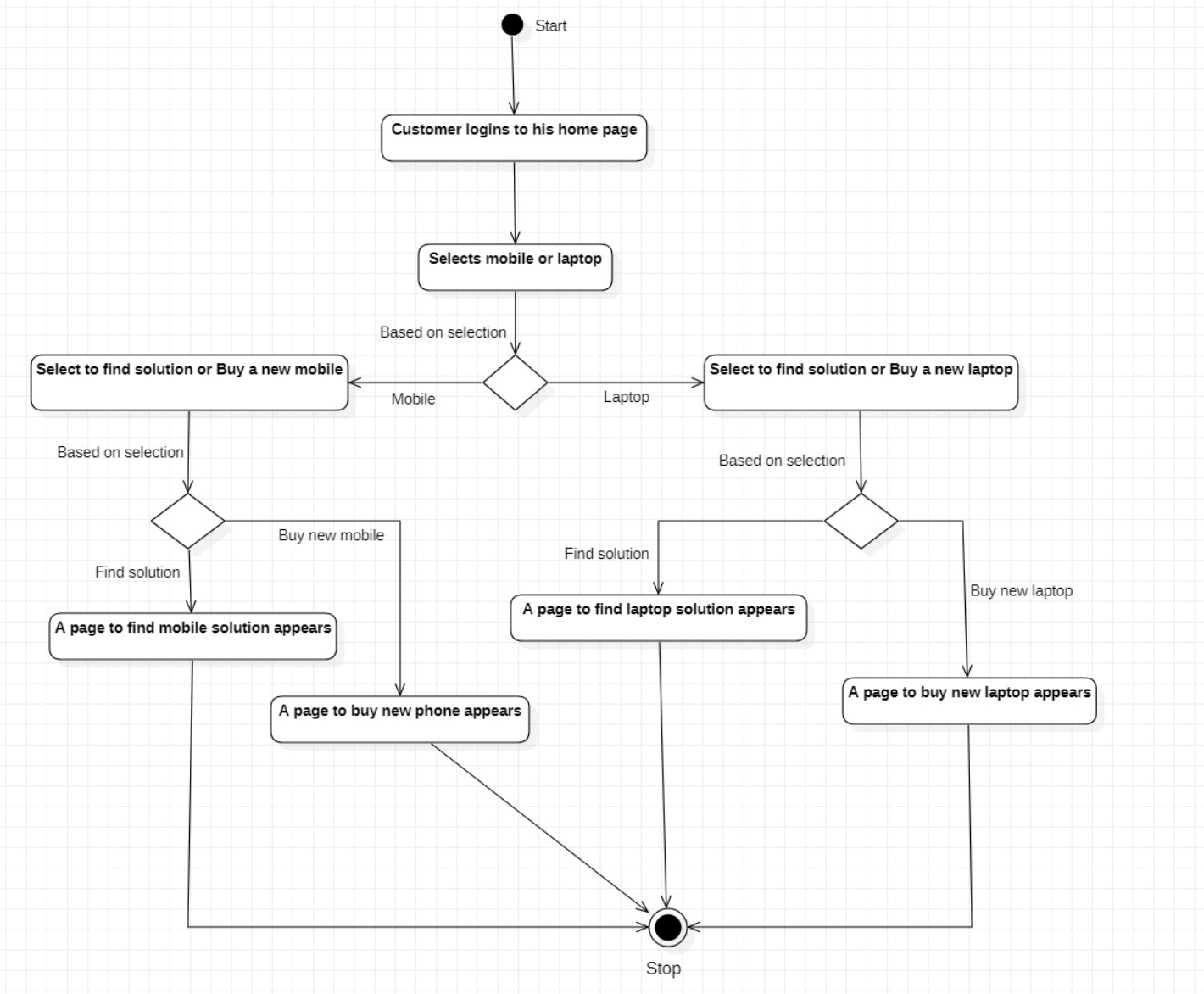
3.2	Developing Web Site	Creating a website for the users to view the solutions to their problems and view the phones available to purchase..							Lalith Kanakamedala
3.2.1	User Interface	To provide a convinient and a professional look to the website.	**01-09-2021**	**04-09-2021**	**5**	**3**		**HTML, CSS, Javascript**	Yogeshwar Sai, Lalith Kanakamedala
3.2.2	Login Page	This page allows the user to access the website using valid login credentials in order to view the solutions or to locate the nearest service centers.	**04-09-2021**	**05-09-2021**	**2**	**1**		**HTML, CSS, Javascript**	Yogeshwar Sai
3.2.3	Products Management							**AJAX**	
3.2.3.1	Contact Vendors	The manager can contact the vendor to send the new products available in the market so that the details can be updated in the database to show the users.	**08-09-2021**	**10-09-2021**	**3**	**2**		**Phone**	Lalith Kanakamedala
3.2.3.2	Updating Database through website	Once the vendor delivers the products, the manager is responsible to update the database by adding the new device details or increasing the quantity.	**10-09-2021**	**11-09-2021**	**2**	**1**		**AJAX**	Lalith Kanakamedala
3.2.4	Problems and Solutions	The user after selecting the device model is asked to select the problem that he is facing. The solution if available in the database is displayed on the user's screen.	**11-09-2021**	**12-09-2021**	**1**	**1**		**Javascript**	Lalith Kanakamedala
3.2.4	Problems and Solutions	The user after selecting the device model is asked to select the problem that he is facing. The solution if available in the database is displayed on the user's screen.	**11-09-2021**	**12-09-2021**	**1**	**1**		**Javascript**	Lalith Kanakamedala
3.2.5	Locations	The user if not able to find solution in the website he can search for the exact locations of the nearest service centers.	**12-09-2021**	**12-09-2021**	**1**	**1**		**Javascript, HTML**	Lalith Kanakamedala, Yogeshwar Sai
3.2.4	Payment Management	It helps to generate a QR code for UPI payments. So that whenever the user is buying a phone he can scan the QR code for payment.						**AJAX**	Lalith Kanakamedala
4	**Website Deployment**								
4.1	Testing and Launching	Several trials are conducted before making the website available to the supermarket. Any changes to be done should be made during the trials itself. Finally, the website is launched.	**15-09-2021**	**19-09-2021**	**6**	**4**	**3.2**	**GitHub**	
4.2	Managing Website Versions	Any changes or updates according to the manager/customer can be done anytime and the website can be relaunched.	**15-10-2021**	**16-10-2021**	**3**	**1**		**GitHub**	

UML Diagrams

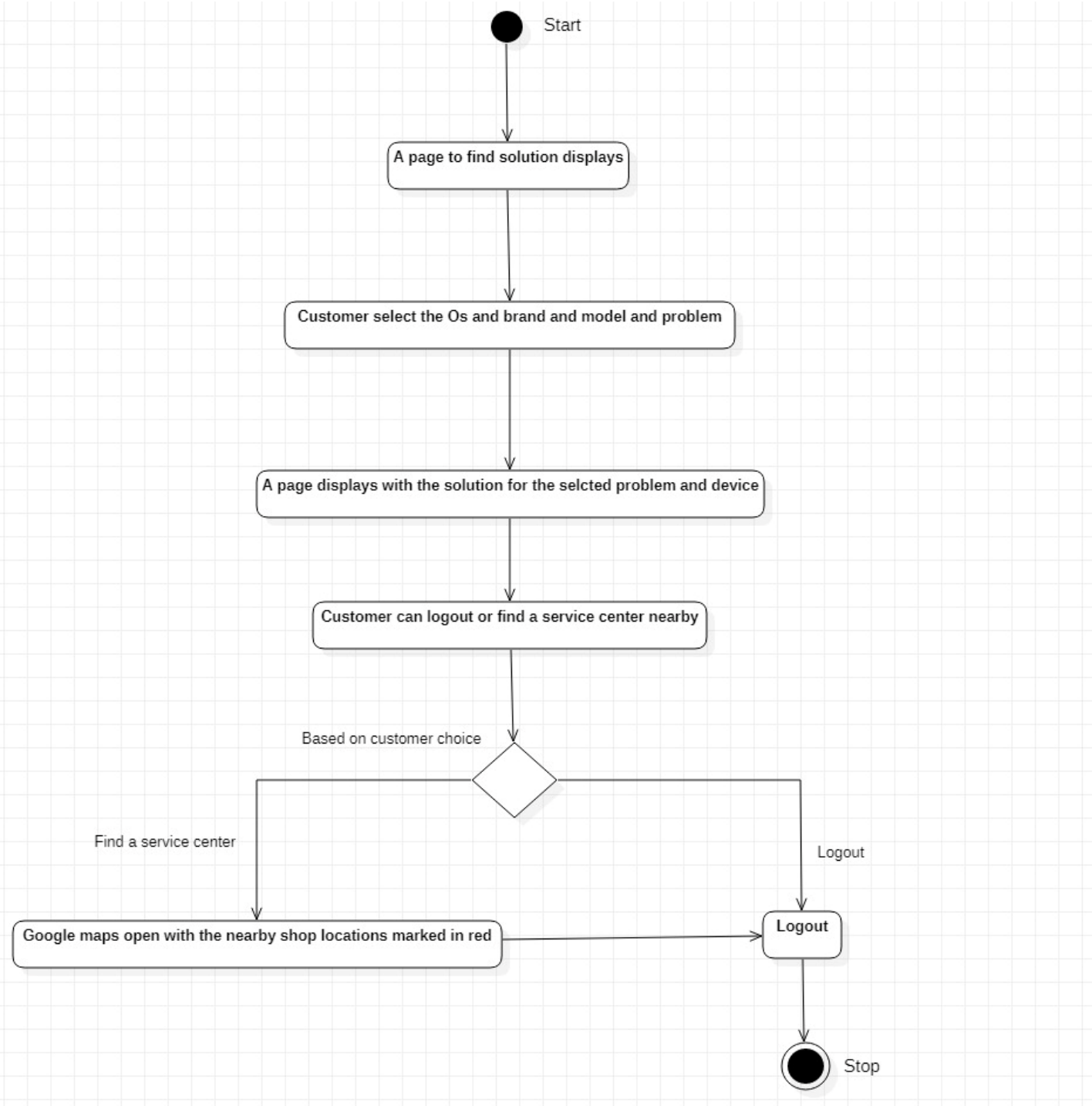
Activity Diagram for Login and Sign-up module:



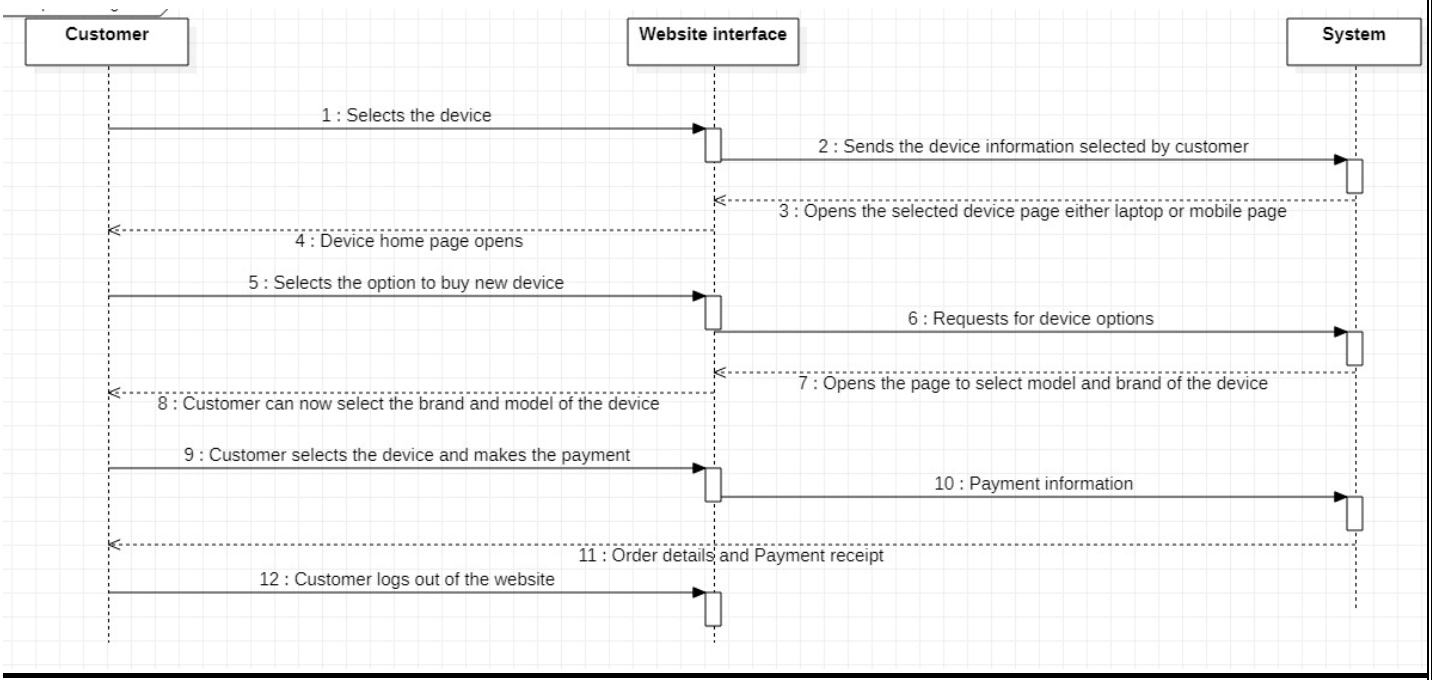
Activity Diagram for Model Selection module:



Activity Diagram for Solution finding and Store Locating module:



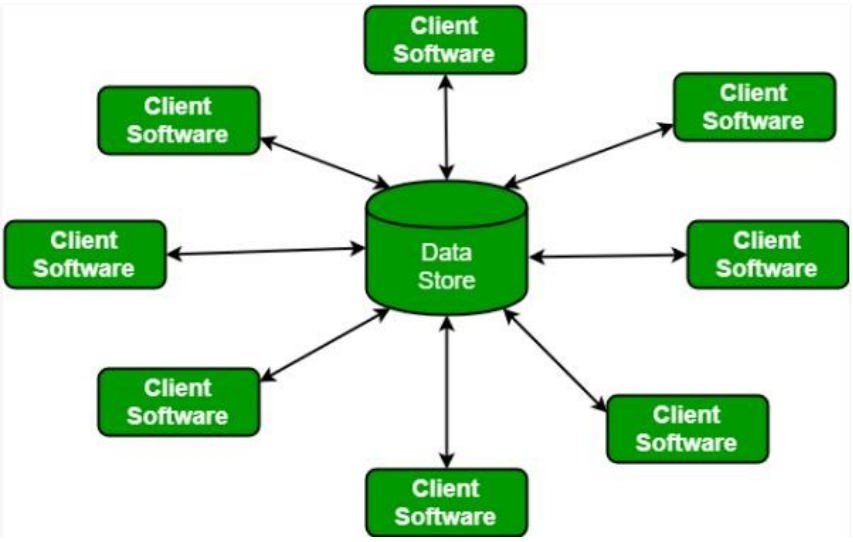
Sequence Diagram for Buying New Phone module:



Architecture

Our website is totally based on the client-server communication, so we decided to proceed with the data-centric architecture also known as client-server architecture.

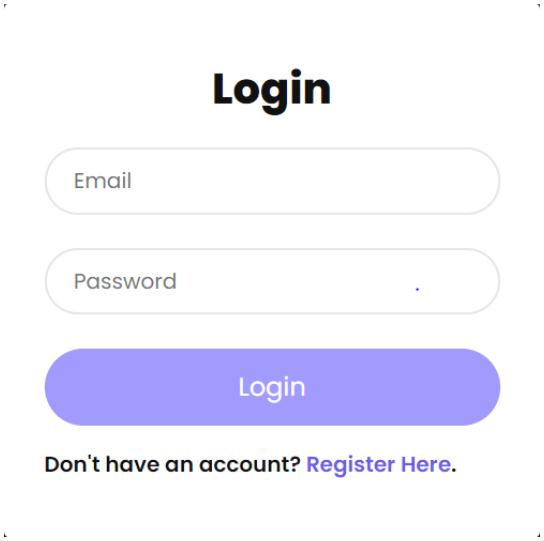
A data store will reside at the center of this architecture and is accessed frequently by the other components that update, add, delete or modify the data present within the store. This data-centered architecture will promote integrability. This means that the existing components can be changed and new client components can be added to the architecture without the permission or concern of other clients.

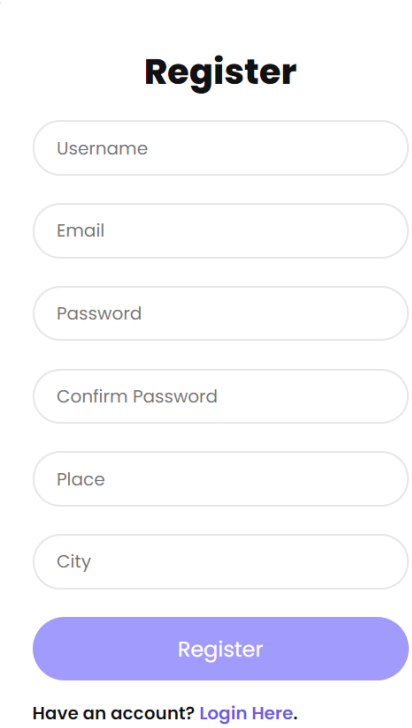


In our project all the data is stored in a central database and the required information is displayed on the user's screen according the input given by him/her. Therefore, we decided to go with the data-centric architecture. Whenever the user registers himself the data is stored in the central database. Even during log in the entered credentials will be checked with the data in the database. During the selection of the problem all the common problems that are stored in the database are displayed from which the user can select the most appropriate one. Also the solutions to these problems are stored and whenever the user selects the problem, the corresponding solution is fetched from the database and is displayed on the screen. If the user is not able to find a satisfactory solution, then the user can view the exact locations of the nearest service centers. All the latitudes and longitudes of these locations are stored in the database as well. So a central database and a data-centric architecture would be more efficient and easy to manage the website.

Graphical User Interface (Screenshots)

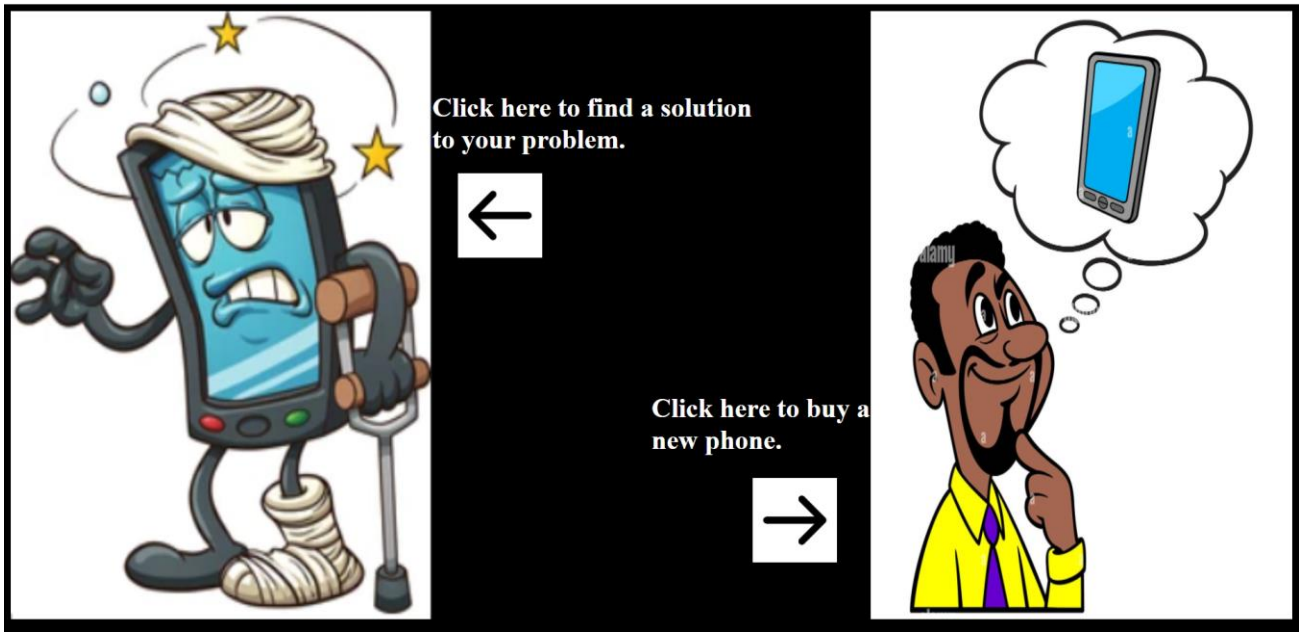
Login and Register





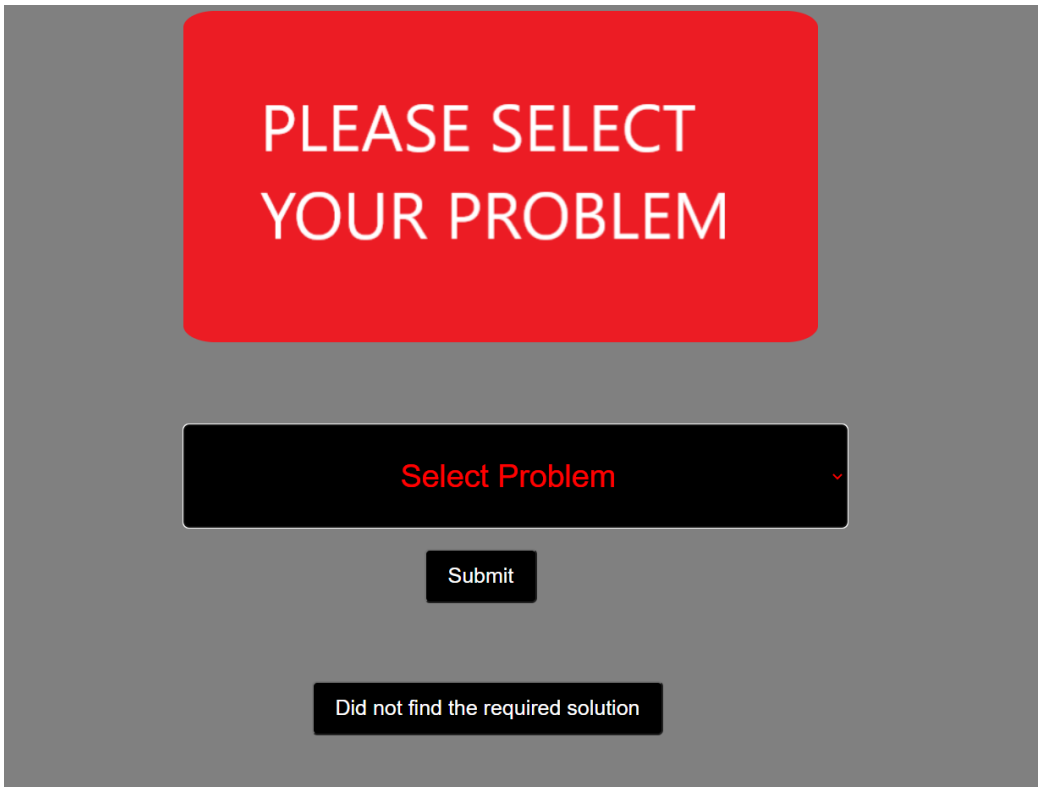
Select Mobile or Laptop

Select whether you want to search for a solution or to buy a new phone

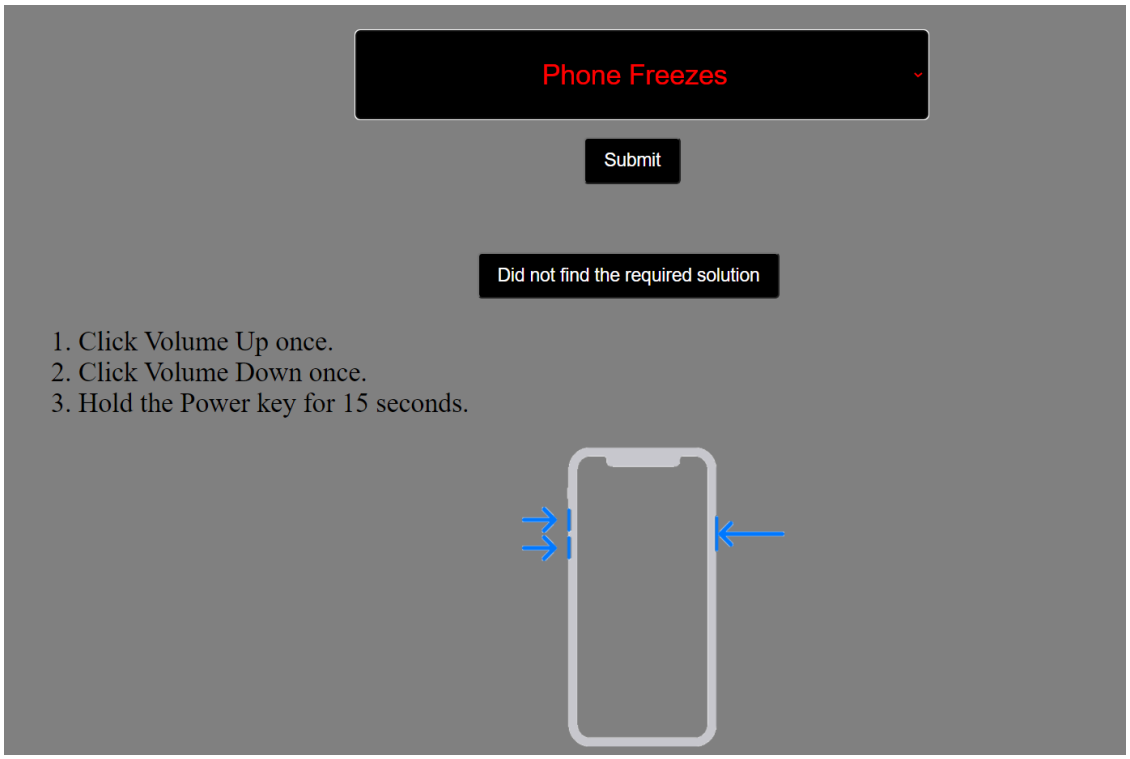


Select Model

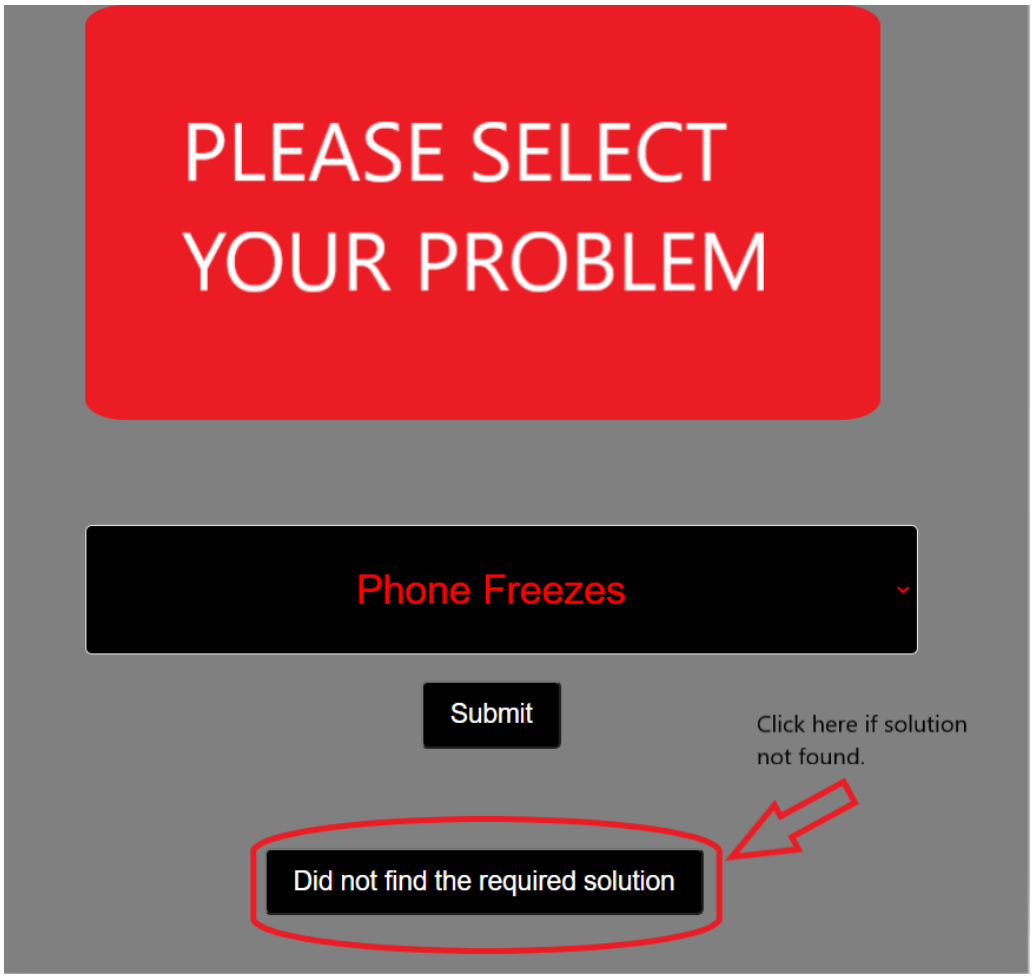
Select Problem



Available Solution shown



Click here to search for nearest service centers



All locations of service centers marked in red around the user are shown in the maps.

Source Code (Sample)

Code for Register page:

```php
<?php
include 'config.php';
error_reporting(0);
session_start();
if (isset($_SESSION['username'])) {
    header("Location: index.php");
}
if (isset($_POST['submit'])) {
    $username = $_POST['username'];
    $store_user = $username;
    $email = $_POST['email'];
    $password = md5($_POST['password']);
    $cpassword = md5($_POST['cpassword']);
    $place = $_POST['place'];
    $city = $_POST['city'];

    if ($password == $cpassword) {
        $sql = "SELECT * FROM users WHERE email='$email'";
        $result = mysqli_query($conn, $sql);
        if (!$result->num_rows > 0) {
            $sql = "INSERT INTO users (username, email, password, place, city)
                    VALUES ('$username', '$email', '$password', '$place', '$city')";
            $result = mysqli_query($conn, $sql);
            if ($result) {
                echo "<script>alert('Wow! User Registration Completed.')</script>";
                $username = "";
                $email = "";
                $_POST['password'] = "";
                $_POST['cpassword'] = "";
            } else {
                echo "<script>alert('Woops! Something Wrong Went.')</script>";
            }
        } else {
            echo "<script>alert('Woops! Email Already Exists.')</script>";
        }

    } else {
        echo "<script>alert('Password Not Matched.')</script>";
    }
}
?>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

    <link rel="stylesheet" type="text/css" href="style.css">

    <title>Register</title>
</head>
<body>
    <div class="container">
        <form action="" method="POST" class="login-email">
            <p class="login-text" style="font-size: 2rem; font-weight: 800;">Register</p>
            <div class="input-group">
                <input type="text" placeholder="Username" name="username" value="<?php echo $username; ?>" required>
            </div>
            <div class="input-group">
                <input type="email" placeholder="Email" name="email" value="<?php echo $email; ?>" required>
            </div>
            <div class="input-group">
                <input type="password" placeholder="Password" name="password" value="<?php echo $_POST['password']; ?>" required>
            </div>
            <div class="input-group">
                <input type="password" placeholder="Confirm Password" name="cpassword" value="<?php echo $_POST['cpassword']; ?>" required>
            </div>
            <div class="input-group">
                <input type="text" placeholder="Place" name="place" value="<?php echo $place; ?>" required>
            </div>
            <div class="input-group">
                <input type="text" placeholder="City" name="city" value="<?php echo $city; ?>" required>
            </div>
            <div class="input-group">
                <button name="submit" class="btn">Register</button>
            </div>

            <p class="login-register-text">Have an account? <a href="index.php">Login Here</a>.</p>
        </form>
    </div>
</body>
</html>
```

Code for Select Model:

```
<!DOCTYPE html>
<html>
    <head>
    <script>
    if ( window.history.replaceState ) {
        window.history.replaceState( null, null, window.location.href );
    }
    </script>

        <style>
            body {
                background-image: url("brands.png");
            }
            img {
                margin-left: 30%;
                border-radius: 5%;
                width: 600px;
            }

            .select {
                font-size: 25px;
                padding: 20px 40px;
                color: red;
                background-color: black;
                border: 1px solid #dddddd;
                cursor: pointer;
                border-radius: 5px;

            }
            .select: focus,
            .select: hover {
                outline: none;
                border: 1px solid #bbbbbb;
            }

        </style>
        <title>Solution</title>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    </head>
    <body>
        <img src="mob_select_model.png">
        <br><br><br><br><br>
<form name="mobiles" action="<?=$_SERVER['PHP_SELF'] ?>" method="post">
<label class="form-group select-boxes col-md-6" class="form-control" style="margin-left: 25%">
    <?php
        $OS = '';
        $query = "SELECT OS FROM mobiles GROUP BY OS ORDER BY OS ASC";
        $result = mysqli_query($conn, $query);
        while($row = mysqli_fetch_array($result))
        {
        $OS .= '<option value="'.$row["OS"].'">'.$row["OS"].'</option>';
        }
    ?>
    <select name="OS" id="OS" class="select action">
    <option value="">Select OS</option>
    <?php echo $OS; ?>
```

```
  </select>
</label>
<label class="form-group col-md-6">
    <select name="Brand" id="Brand" class="select action">
    <option value="">Select Brand</option>
  </select>
</label>
<label class="form-group col-md-6">
    <select name="Model" id="Model" class="select">
    <option value="">Select Model</option>
    </select>
        </label>

  <label class="form-group col-md-6">

        </label>

</div>
  </form>
  <br>
    <form action=select_problem.php id="sub">
        <input type="submit" name="submit" value="Submit" form="sub" class="btn float-right btn-block"
    </form>


    </body>
</html>

<script>
    $(document).ready(function() {
        $('.action').change(function() {
            if ($(this).val() != '')
            {
                var action = $(this).attr("id");
                var query = $(this).val();
                var result = '';
                if (action == "OS")
                {
                    result = 'Brand';
                }
                else
                {
                    result = 'Model';
                }
                $.ajax({
                    url:"fetchdata.php",
                    method: "POST",
                    data:{action:action, query:query},
                    success: function(data) {
                        $('#'+result).html(data);
                    }
                })
            }
        });
    });
</script>
```

Code for selecting problem and fetching solution:

```php
<?php
    $query = "SELECT Problem FROM prosol";
    $result = mysqli_query($conn, $query);
?>
<!DOCTYPE html>
<html>
    <head>
        <title>Select Problem</title>
    </head>
    <body>
        <img src="problem_select.png">
        <br><br><br><br><br>
    <form action="" method="post">
        <label class="form-group select-boxes col-md-6" class="form-control" style="margin-left: 30%">
            <select name="Problem" id="Problem" class="select">
            <option disabled selected value="">Select Problem</option>
            <?php
                while($row = mysqli_fetch_array($result))
                { ?>
                    <option value="<?php echo $row['Problem']; ?> " > <?php echo $row['Problem']; ?> </option>;
                <?php
                } ?>
            </select>
        </label>
        <input type="submit" name="submit" value="Submit" class="btn float-right btn-block" style="background-co

    </form>

    <form action=maps1.php id="sub">
        <input type="submit" name="submit" value="Did not find the required solution" form="sub" class="btn floa
    </form>

    <?php
        if(isset($_POST['submit'])){
            if(!empty($_POST['Problem'])) {
                $selected = $_POST['Problem'];
                $query1 = "SELECT Solution FROM prosol WHERE Problem = '$selected' ";
                $result1 = mysqli_query($conn, $query1);
                while($row1 = mysqli_fetch_array($result1))
                {
                    ?> <p> <?php echo nl2br($row1['Solution']); ?> </p>
                <?php
                }
                if (strncmp($selected,'Phone Freezes',13)==0) { ?>
                    <img src="iphone.gif" style="width:400px; margin-left: 40%">
        <?php   }
            }
        }
    ?>


    </body>

</html>
```

Code for fetching and displaying locations of service centers:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Maps</title>
        <style>
            #mapCanvas {
                width: 100%;
                height: 750px;
            }
        </style>
        <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyAov2sc2rGZWjYStf6G30YxgGN_qHvTwYg"></script>
        <script>
            function initMap() {
                var map;
                var bounds = new google.maps.LatLngBounds();
                var mapOptions = {
                    mapTypeId: 'roadmap'
                };
                map = new google.maps.Map(document.getElementById("mapCanvas"));
                map.setTilt(100);
                var markers = [
                    <?php
                        if ($result->num_rows > 0) {
                            while ($row = $result->fetch_assoc()) {
                                echo '["'.$row['name'].'", '.$row['latitude'].', '.$row['longitude'].', "'.$row['icon'].'"],';
                            }
                        }
                        ?>
                ];
                var infoWindowContent = [
                    <?php
                        if ($result2->num_rows > 0) {
                            while ($row = $result2->fetch_assoc()) {
                                ?>
                                ['<div class="info_content">' + '<h3><?php echo $row['name']; ?></h3>' + '</div'],
                                <?php
                            }
                        }
                        ?>
                ];
                var infoWindw = new google.maps.InfoWindow(), marker, i;
                for (i=0;i<markers.length;i++) {
                    var position = new google.maps.LatLng(markers[i][1], markers[i][2]);
                    bounds.extend(position);
                    marker = new google.maps.Marker({
                        position: position,
                        map: map,
                        icon: markers[i][3],
                        title: markers[i][0]
                    });
                    google.maps.event.addListener(marker, 'click', (function(marker,i) {
                        return function() {
                            infoWindow.setContent(infoWindowContent[i][0]);
                            infoWindow.open(map,marker);
                        }
                    })(marker,i));
                    map.fitBounds(bounds);
                }
                var boundsListener = google.maos.event.addListener((map), 'bounds_changed', function(event) {
                    this.setZoom(14);
                    google.maps.event.removeListener(boundsListener);
                });
            }
            google.maps.event.addDomListener(window, 'load', initMap);
        </script>
    </head>
    <body>
        <div id="mapContainer">
            <div id="mapCanvas"></div>
        </div>
    </body>

</html>
```

Testing Results

Login and Sign-up:

Test Case#	Test Title	Test Summary	Test Steps	Test Data	Expected Result	Post-condition	Actual Result	Status	Notes
TestCase #L001	Signup	The Signup page was tested.	Open the website	Username = ******	The System saves the information and the given credentials	Login using the credentials	Perfectly loggedin using the credentials	Passed	Only new customers should Signin
			Select Signup	Email = *****@gmail.com					
			Enter required details	Password = ******					
				City = *****					
TestCase #L002	Login	The login page was tested.	Open the website	Email = ******	Home page opens with the customer name.	Selects the device as per customers need	Home page opened successfully	Passed	Customer who already have credentials can only login. New customers should Signup
			Select Login	Password = ******					
			Enter Login Credentials						
TestCase #L003	Login (invalid Credentials)	The login page was tested.	Open the website	Email = ******	Expecting an error	Will revert back to the login page.	Error Occurred	Passed	Check the credentials and only enter valid ones.
			Select Login	Password = ******					
			Enter Login Credentials						

Device model selection and Problem Selection:

Test Case#	Test Title	Test Steps	Test Data	Expected Result	Post-condition	Actual Result	Status
1	Customers selects a device	Customer Logs in using correct credentials	Customer home page pops up	Home page of selected device opens	Customer should be ready with his mobile brand and name to search for problem or to buy a device	Home page of phone or laptop opened	Successful
		Customer clicks on device he requires to proceed with	Customer clciks on phone or laptop				
2	Customer selects to find a solution	Customer should click on Find a solution on the left for the selected device	Android = *****	Solution for the given problem is displayed	Customer can find a location on map if not satisfied with solution given	Solution is displayed on the screen	Successful
		Customer should enter the detals of the device and the problem	Brand = ******				
		If not satisfied with the given solution cu	Model = ******				
			Problem = ****				

Service Center Locator and Device purchase:

3	Customer not satisfied with provided solution	Customer selects the option of not satidfied displayed on problem page	Google maps with Service centers highlighted on it	Google maps to pop up after customer selects not satisfied option.	Customer can logout if he has no other queries	Google maps pop up as soon as customer selects the option not satisfied on problem page.	Successful
		A maps pops up with red dots marked on it showing the Service center locations					
		Customer can move the cursor on to the places marked in red to fund the exact address					
4	Customer selects to buy a new device	Customer clicks on buy a new device	Android = *****	Customer finds the devices as per his requirement and buys the device if he is satisfied	Customer recieves the order details to his mail and he can log out after the payment	Customer finds the devices as per his search and buys the device which he wants	Successful
		Customer enters all the details of the device	Brand = ******				
		Customer makes the payment and buys the device if he wants	Model = *****				
			Price = ****				

Conclusion

We were able to design a web application which consist of features like fetching solutions for the user's problems faced in their respective devices along with the solution to the problems we were also able to enable buy or sell products. Our application also provides the GPS locations of the nearest service centres to the user.

So our application will help the people during the pandemic so that they would not have to go out for small problems they are facing with their devices by providing the solution for various kinds of problems in a variety of electronic devices. People not only solve their problems but they can even buy and sell their products in this application.

References

https://www.samsung.com/in/support/service-center/