

## Vehicle Emergency Service System

Rohit Mahajan<sup>1</sup>, Shruti Shende<sup>2</sup>, Prateek Meshram<sup>3</sup>, S.Hajare<sup>4</sup>

<sup>1</sup>(Student of CSE, JIT, India)

<sup>2</sup>(Student of CSE, JIT, India)

<sup>3</sup>(Prof at CSE, JIT, India)

<sup>4</sup>(Prof at CSE, JIT, India)

**Abstract:** The system of vehicle repair provider is a trouble that needs to analyze on the area of repair center consistent with the provider wishes and to be had useful resource based totally at the maximal overlaying location and precedence queuing concept. Considering the effect of ready time because of rush jobs, this challenge proposes a version that maximizes the service overlaying, and restrains. Service level of uncovered zones. As we see human beings going through many issues related to vehicles and Most of the humans use community services and presenting on line software carrier on the way to create extra advantage for users as well as provider company. So we can build the application of Management gadget with notification the usage of app could be evolved to solve all of the modern-day problem related to the automobile. Using which the person that is searching out automobile restore carrier will get all the facilities of the car of their own location. The scope of this venture will cognizance on the user and service company who will use this application through online service. This venture may also be applied in small shops. The proposed system will store the efforts and time of a person in addition to improve the boom of employment.

**Keywords :** Online service, Service Provider, Vehicle service.

### I. Introduction

Emergency system (ES) is a system of several components including organizations, transportation and communication networks, trained professionals, and administrators aiming to work in coordination for success[1]. Having the primary concern as emergencies, planning of this system requires significant work to ensure serving the public at its best Other than administrative decisions, planning of physical infrastructure, primarily locating emergency vehicles constitutes a major part in the performance of the system.

Various criteria could be important in deciding the locations of emergency vehicle server locations[3]. In our study, the problem of vehicle Emergency system such as air leaks in tire tube, petrol finished during travelling, heating of car's engine and battery drain as a server-to-customer service is discussed based on online processes The region where vehicle emergency service system would be located is divided into demand regions and some of those regions are considered as candidate vehicle locations[5]. As the studies toward vehicle service related system have been growing. We are focusing on the issues like vehicle accident management system, vehicle location tracking system, a system for a fastest route for vehicle etc. It is good to focus on such issues but we should not forget to issues of as our daily life vehicle problem. That is small and as well as an important issue to focus In our daily life, we either used 2 wheeler or 4 wheelers to go on different places like college, office or even if we want to hangout we use our vehicle. The vehicle has become our need for travelling. But every time when we out with our vehicle we forgot to keep emergency vehicle tools with us. Sometimes we stuck in places where there is no vehicle emergency shop nearby and this happens with us such as.

1. Air leaks in a tire tube.
2. Petrol finished travelling.
3. Heating of car's engine and battery drain.

You will definitely go to stuck there. So keeping such problems in mind we are proposing a vehicle emergency service system which arrives within the specific time designated place and can cover as much of the potential demand. It is critical that vehicles be all-time located so as to ensure an adequate coverage and a quick response time.

We provide many services similar to that of ola and uber:

I. Peer to peer ride sharing services

Provision of ride booking for individual or can be shared among individuals.

II. Car and bikes renting services

Provision for renting vehicle on the bases of time and distance traveled.

III. 24\*7 support

Support for emergency and uncertain condition.

IV. Vehicle tracking

Vehicle can be tracked at real time.

V. Women safety

All measures for women safety are provided such as provision of lady driver ,special alerts on vehicle.

**II. Related Work**

Our work is closely related to the study of vehicle emergency methods.

**Paresh Rathod | Pasi kampp**(2013)[10]user requirements for emergency response vehicles. A research review of existing emergency response vehicles (ERVs) shows the implementation of technological services and solutions across various spectrums in normal conditions. However, a close encounter with real users of ERVs reveals shortcomings in various aspects, especially during varied conditions [1]. MOBI (Mobile Object Bus Interaction), a research project led by Laurea University of Applied Sciences, Finland. The project aims to explore the standardization process designing an emergency response vehicle and further develop the product concept with industrial partners. The real users and industry partners are focal points of the project. A further project also explores the potential of standardization process of the fields. Generated research data has been in use to create a demo vehicle with working integrated ICT systems.

**Mingang Zeng | Gaohui Yu** (2011) [11] Online rescue vehicle scheduling problem of emergency logistics Emergency logistics of natural disaster plays an important role in effectively controlling the casualties and reducing various disaster losses. The paper proposes k-rescue vehicles scheduling problem and considers this problem with a two-stage algorithm based on the online method. The first stage, the paper proposes online rescue vehicle allocation strategies which include the repositioning strategy, the lie over strategy and the mixed strategy, and then analyses their competitive ratio. The second stage, this paper designs selection strategies of the online path which include the reposition strategy and the greedy strategy, and then analyses their competitive ratio. Finally, based on the analysis and calculation of an example, the results indicate that the above models and methods are effective.

**Li - si Cao | Zi-xian-Liu** (2011) [12] An emergency service centre location model for vehicle repair. Vehicle repair is an item of automobile warranty service. As an important effect Of warranty quality, how to provide an effective service in a constrain time is attracting more and more attention. It is a solution to this problem by establishing an automobile emergency system.

Published On	Input	output	Key Features
2013	User requirements of emergency response vehicles	Paresh Rathod   Pasi kamp	Mobile Object Bus Integration (MOBI), Emergency Response Vehicles (ERVs)
2011	Online rescue vehicle scheduling problem of emergency logistics	Mingang Zeng   Gaohui Yu	Proposes k-rescue vehicles scheduling problem, online rescue vehicle allocation strategies.
2019	Vehicle emergency service system.	Rohit Mahajan   Anjali Kungwani   Shruti Shende	Emergency service of vehicle on user location, Real time location tracking of service provider.

**III. Problem Statement**

Vehicle Emergency Service System TO Provide Emergency Vehicle Services TO The User On Their Own Location.

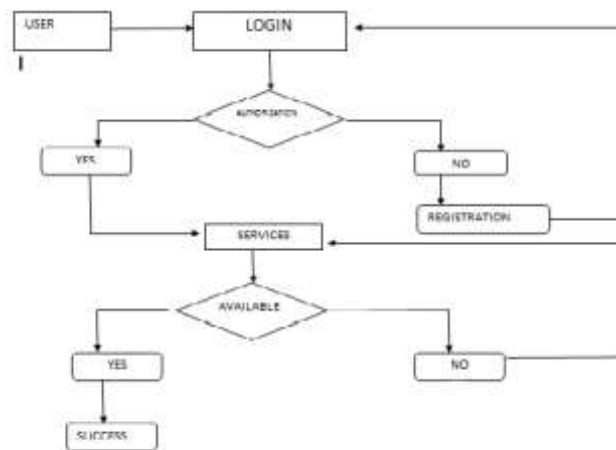
**Proposed Plan**

In order to overcome the existing problem the vehicle emergency service system is employed. In the proposed system of this project is to monitor the emergency problem related to the vehicle by using application. The user needs to register first on the vess system. After completing the registration now user can log in to the system and get benefits of the services. Same goes for the service provider, the service provider will also register on vess. After that, they will be able to login into the system. After login They can add the services they will be providing.

When a user needs any services they will log in and directed to the services page from which they will choose the service for their vehicle and get the information of the entire service provider who is providing their requested service. As soon as the user selects any of the service provider users will get their contact information. After successful communication user will able to get the services and will be able to track the service provider and estimated time to reach to the user.

**Algorithm:**

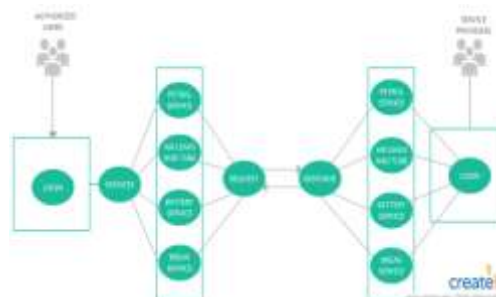
- Step 1: Open the application and log in.
- Step 2: Select the location
- Step 3: Select the appropriate services according to your vehicle condition.
- Step 4: Send the request to the service provider.



**Figure 2:** Data flow Diagram

- Step 5: if (Service provider will accept the request and response to it.) goto step 7
- Step 6: else goto step 2
- Step 7: Service provider will solve the problem of the vehicle at user’s location.
- Step 8: Logout the application.

**IV. Flow Diagram**



**Figure :** Complete Reference Diagram of VESS System

**V. Applications**

Vehicle emergency system will provide you on spot services on your location so you would not have to suffer from such problems. This will definitely save our time and we would not have to run for services.

Saving of energy that we waste on pulling vehicle, Get the exact time of the service depending on our location and able to track the service provider.

## VI. Implementation

Implementation of this project is being done on android platform. Android Studio is the official Android Intergrated Development Environment (IDE) from Google. Succeeding Eclipse in 2014, Android Studio is an all around IDE that allows developers to do everything they need without leaving the environment.

Android Studio is based on IntelliJ IDEA, which is a powerful Java code editor. In addition to that, Android Studio also offers several features that a developer would need when building an application, such as:

- Testing tools and frameworks.
- Integrated version control system.
- Lint tools to measure performance, usability and version compatibility.

Overall, Android Studio is a very powerful all around IDE to build and test Android apps. It has the best code editor the author has used and very effective tools to test developing apps live on a virtual device or plugged-in device. Android Studio includes everything that is needed to build a fully functional Android application.

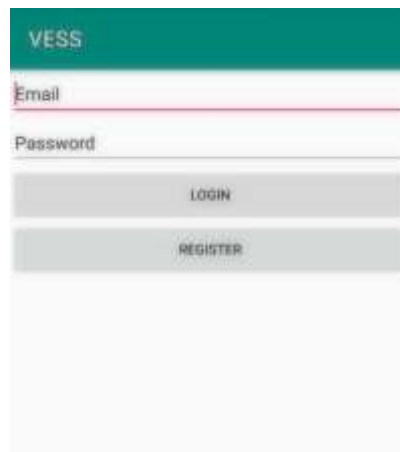


FIG : 7.1 Registration Module



FIG 7.2 : Login activity

### Step 1:

This is the first activity that the user will use when they open the app for the first time. To use the services of the system user and service provider both need to register themselves on the system. After successful registration the data of user will go to the database. After successful login user and service provider can use the features of system.

**Firestore :** A realtime database

The Firestore Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all of your clients share one Realtime Database instance and automatically receive updates with the newest data.

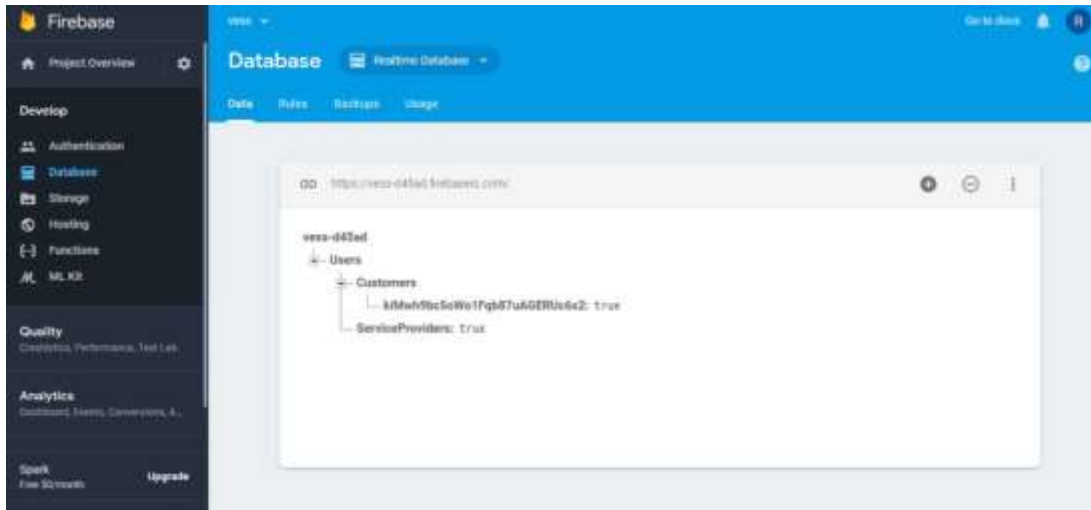


Fig 7.3 : Firestore

Key capabilities

- Realtime
- Offline
- Accessible from client devices
- Scale across multiple databases

How does it work ?

The Firestore Realtime Database lets you build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firestore Realtime Database Security Rules, to define how your data should be structured and when data can be read from or written to. When integrated with Firestore Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables you to build a great realtime experience that can serve millions of users without compromising on responsiveness. Because of this, it is important to think about how users need to access your data and then structure accordingly.

**Step 2 :**

After Login user get to see the map activity in which there are two services “Call puncture service” and “Call mechanic”. As soon as user select any of the service the system will search for nearby service provider. As soon as the service provider found then user will enter contact information and communication between user and service provider will take place.

For map services we are using some APIs

- Google Map API
- Geofire API

#### **Google Map API:**

- With the Google Maps Android API, you can add maps based on Google Maps data to your application. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures. You can also use API calls to add markers, polygons, and overlays to a basic map, and to change the user's view of a particular map area. These objects provide additional information for map locations, and allow user interaction with the map [9]. The API allows you to add these graphics to a map: Icons anchored to specific positions on the map (Markers).
- Sets of line segments (Polylines).
- Enclosed segments (Polygons).
- Bitmap graphics anchored to specific positions on the map (Ground Overlays)
- Sets of images which are displayed on top of the base map tiles (Tile Overlays)

Creating a new Android application that uses the Google Maps Android API v2 requires several steps.

The overall process of adding a map to an Android application is as follows:

1. Install the Android SDK.
2. Download and configure the Google Play services SDK, which includes the Google Maps Android API. If the Google Maps Mobile SDK for Business is used, then download and configure the Google Maps Mobile SDK for Business static library.
3. Obtain an API key. To do this, one needs to register a project in the Google APIs Console, and get a signing certificate for your app.
4. Add the required settings in your application's manifest.
5. Add a map to your application.
6. Publish the application. Detailed information on each of the above steps can be found at the following web location – <https://developers.google.com/maps/documentation/android/start>.

#### **Geofire API:**

GeoFire is an open-source library for Android/Java that allows you to store and query a set of keys based on their geographic location. At its heart, GeoFire simply stores locations with string keys. Its main benefit however, is the possibility of querying keys within a given geographic area - all in realtime.

GeoFire uses the Firebase database for data storage, allowing query results to be updated in realtime as they change. GeoFire selectively loads only the data near certain locations, keeping your applications light and responsive, even with extremely large datasets.

A compatible GeoFire client is also available for Objective-C and JavaScript. For a full example of an application using GeoFire to display realtime transit data, see the SFVehicles example in Android app in this repo.

#### **Integrating GeoFire with your data**

GeoFire is designed as a lightweight add-on to the Firebase Realtime Database. However, to keep things simple, GeoFire stores data in its own format and its own location within your Firebase database. This allows your existing data format and security rules to remain unchanged and for you to add GeoFire as an easy solution for geo queries without modifying your existing data.

##### **A. Including GeoFire in your project Android/Java:**

In order to use GeoFire in your project, you need to add the Firebase Android SDK. After that you can include GeoFire with one of the choices below.

Note that after version 1.1.1 the artifact `com.firebase:geofire` is no longer updated and has been replaced by the separate Android and Java artifacts as described below.

## 1) Gradle

Add a dependency for GeoFire to your gradle.build file.

For Android applications:

```
dependencies {
    compile 'com.firebase:geofire-
    android:2.3.1'
}
```



**Fig 7.4:** Map activity where user can search for service provider.

Here user can search for service provider using A\* algorithm. A\* is a shortest Path Algorithm based on Data structure to hold the cost of map or neural network generated while using google map interfaces to calculate the best path as, while using depth first search we find the best solution or good solution because back tracking is not possible but using breadth first search we can back track in the network. Thus by combining both with A\* and using greedy search we could perform a best path between two nodes in a network. Thus this approach can be used in real time to connect two nodes in map interfaces. However this solution utilizes time and memory but future implementations could reveal a more optimized solution.

A\* is an informed search algorithm, or a best-first search, meaning that it is formulated in terms of weighted graphs: starting from a specific starting node of a graph, it aims to find a path to the given goal node having the smallest cost (least distance travelled, shortest time, etc.). It does this by maintaining a tree of paths originating at the start node and extending those paths one edge at a time until its termination criterion is satisfied.

At each iteration of its main loop, A\* needs to determine which of its paths to extend. It does so based on the cost of the path and an estimate of the cost required to extend the path all the way to the goal. Specifically, A\* selects the path that minimizes

where  $n$  is the next node on the path,  $g(n)$  is the cost of the path from the start node to  $n$ , and  $h(n)$  is a heuristic function that estimates the cost of the cheapest path from  $n$  to the goal. A\* terminates when the path it chooses to extend is a path from start to goal or if there are no paths eligible to be extended. The heuristic

function is problem-specific. If the heuristic function is admissible, meaning that it never overestimates the actual cost to get to the goal, A\* is guaranteed to return a least-cost path from start to goal.

Typical implementations of A\* use a priority queue to perform the repeated selection of minimum (estimated) cost nodes to expand. This priority queue is known as the open set or fringe. At each step of the algorithm, the node with the lowest  $f(x)$  value is removed from the queue, the  $f$  and  $g$  values of its neighbors are updated accordingly, and these neighbors are added to the queue. The algorithm continues until a goal node has a lower  $f$  value than any node in the queue (or until the queue is empty).<sup>[a]</sup> The  $f$  value of the goal is then the cost of the shortest path, since  $h$  at the goal is zero in an admissible heuristic.

The algorithm described so far gives us only the length of the shortest path. To find the actual sequence of steps, the algorithm can be easily revised so that each node on the path keeps track of its predecessor. After this algorithm is run, the ending node will point to its predecessor, and so on, until some node's predecessor is the start node.

As an example, when searching for the shortest route on a map,  $h(x)$  might represent the straight-line distance to the goal, since that is physically the smallest possible distance between any two points.

If the heuristic  $h$  satisfies the additional condition  $h(x) \leq d(x, y) + h(y)$  for every edge  $(x, y)$  of the graph (where  $d$  denotes the length of that edge), then  $h$  is called monotone, or consistent. In such a case, A\* can be implemented more efficiently—roughly speaking, no node needs to be processed more than once (see closed set below)—and A\* is equivalent to running Dijkstra's algorithm with the reduced cost  $d'(x, y) = d(x, y) + h(y) - h(x)$ .

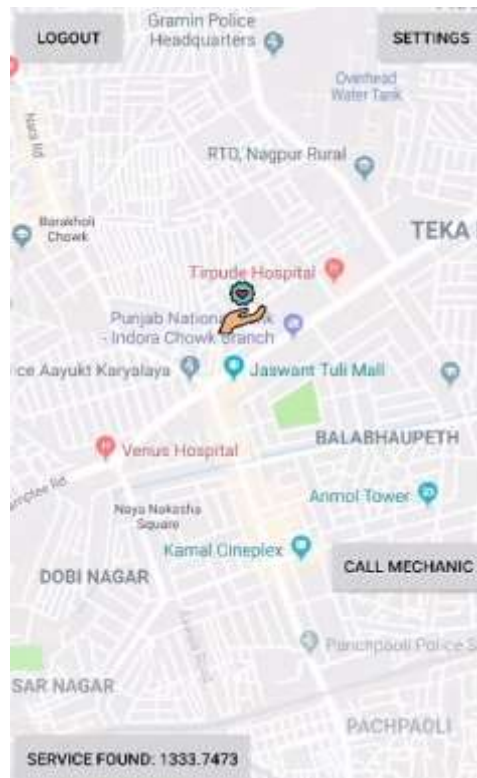


Fig 7.5 : Map activity when service found successfully.

**Step 3:**

After successful search of service provider user need to share contact information. Then service provider will get information about user and his location and now service provider can provide services to the user.





Fig 7.6 : User information activity

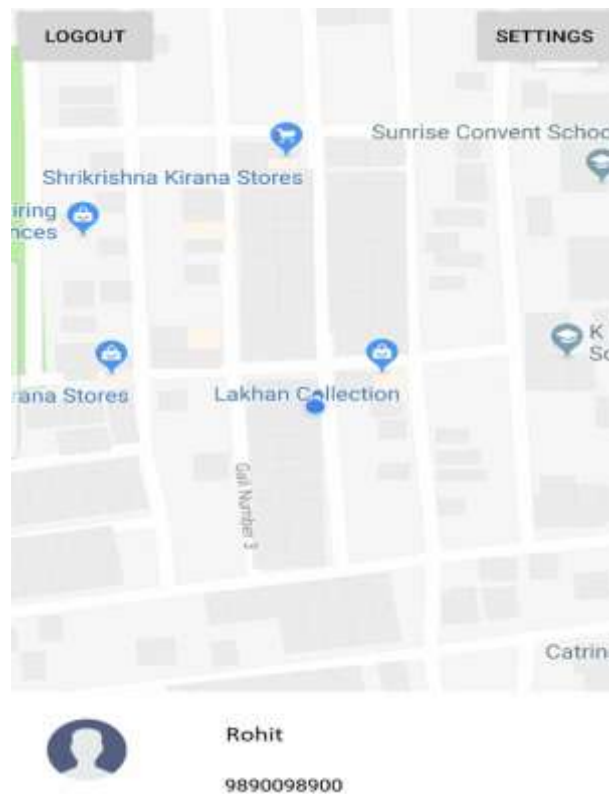


Fig 7.7: Service provider receiving user information.

### **VII. Result analysis and discussion**

Application	Ambulance emergency ,time base.	Vehicle emergency ,location based.
Real time location tracking	Vehicle cannot be tracked at real time	Vehicle can be tracked at real time
Availability	24*7	24*7
Fast Response	Searching of ambulance takes time.	Fast searching due to efficient use of algorithm.

### **VIII. Conclusion**

We have proposed an approach for the daily life problems of the vehicle that will definitely be more effective than our traditional approach. The future Vehicle renting service would be a great service as we sometimes can't afford the time to be taken on vehicle repair. In such cases, we can give our failed vehicle to the service provider and take another vehicle for rent from them and can continue our related work.

### **References**

- [1]. FACILITATING EMERGENCY RESPONSE VEHICLES' MOVEMENT THROUGH A ROAD SEGMENT IN A CONNECTED VEHICLE ENVIRONMENT BY GABY JOE HANNOUN | PAMELA MURRAY-TUITE YEAR: 2018 PAGE S: 1 - 12
- [2]. A STOCHASTIC EMERGENCY VEHICLE REDEPLOYMENT MODEL FOR AN EFFECTIVE RESPONSE TO TRAFFIC INCIDENTS BY CHAO LEI | WEI-HUA LIN YEAR: 2015 , VOLUME: 16 , ISSUE: 2 PAGE S: 898 - 909
- [3]. A METHOD OF SECURING PRIORITY-USE ROUTES FOR EMERGENCY VEHICLES USING INTER-VEHICLE AND VEHICLE-ROAD COMMUNICATION BY YUMA MOROI | KAZUMASA TAKAMI YEAR: 2015 PAGE S: 1 - 5
- [4]. Approximating Vehicle Dispatch Probabilities for Emergency Service Systems with Location-Specific Service Times and Multiple Units per Location by Susan Budge, ArmannIngolfsson and ErhanErkut Vol. 57, No. 1 (Jan. - Feb., 2009), pp. 251-255
- [5]. A parallel matheuristic for the technician routing and scheduling problem by V. Pillac, C. Guéret,A. L. Medaglia. October 2013
- [6]. User requirements of emergency response vehicles by Paresh Rathod | Pasi 17 April 2014, Electronic ISBN: 978-1-4799-2491 2, Las Vegas, NV, USA.
- [7]. Online rescue vehicle scheduling problem of emergency logistics by Mingang Zeng | Gaohui Yu(2011),DOI: [10.1109/ICEMMS.2011.6015686](https://doi.org/10.1109/ICEMMS.2011.6015686).
- [8]. Li - si Cao | Zi-xian-Liu (2011) [12] An emergency service centre location model for vehicle repair