SPARE FIX – CAR SPARE PARTS AND SERVICE PROVIDER APP

A Project report Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

By

BHAWIN A B

(Reg No: 211IT009)

Under the Guidance of

Mrs. A. SATHIYA PRIYA, M.C.A.,(Ph.D.,)

Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY

Dr. N.G.P. ARTS AND SCIENCE COLLEGE

(An Autonomous Institution, Affiliated to Bharathiar University, Coimbatore)
Approved by Government of Tamil Nadu & Accredited by NAAC with A^{++} Grade (3rd Cycle -3.64 CGPA)
Dr. N.G.P.-Kalapatti Road, Coimbatore-641 048, Tamil Nadu, India.
Website: www.drngpasc.ac.in | Email: info@drngpasc.ac.in. | Phone: +91-422-2369100

April - 2025

DECLARATION

I, **BHAWIN. A. B.(Reg.No:221IT009)**hereby declare that the project report entitled **“SPARE FIX-CAR SPARE PARTS AND SERVICE PROVIDER APP”** submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Science in Information Technology at the Bharathiar University** is a record of original project work done during the period of study under the supervision and guidance of **Mrs.A.SathiyaPriya, M.C.A.,(Ph.D.,) Assistant Professor, Department of Information Technology, Dr. N.G.P. Arts and Science College, Coimbatore - 48**, and it has not formed on the basis of award of any Degree/ Diploma/ Associateship/ Fellowship or other similar title to any candidate of any university.

Place : Coimbatore **BHAWIN A B**
Date: **(221IT009)**

CERTIFICATE

This is to certify that the project entitled **“SPARE FIX-CAR SPARE PARTS AND SERVICE PROVIDER APP”** submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Science in Information Technology at the Bharathiar University** is a record of original project work done by **BHAWIN A B (Reg No:221IT009)** during the period (2022-2025) of his study in **Department of Information Technology, Dr. N.G.P. Arts and Science College, Coimbatore-48** under my supervision and guidance, and the project has not formed the basis for the award of any Degree/ Diploma/ Associateship/ Fellowship or other similar title to any candidate of any university.

(Mrs. A. Sathiya Priya)	**(Dr. A. Adhiselvam)**	**(Prof. Dr. S. Saravanan)**
Guide	Head of the Department	Principal

Place: Coimbatore
Date:

Viva-voce Examination held on…………………

Internal Examiner **External Examiner**

Firmware
Solutions

Place: Coimbatore

TO WHOMSOEVER IT MAY BE OF CONCERN

This is to certify **Mr. BHAWIN A B(RegNo: 221IT009) III-year B.Sc. Information Technology** student from **"Dr. N.G.P. ARTS AND SCIENCE COLLEGE"** Coimbatore has successfully completed his project entitled **"SPARE FIX-CAR SPARE PARTS AND SERVICE PROVIDER APP"** from our esteem organization.

His performance and conduct were found to be very good.

During this period, he was sincere and regular in attending all the faces of the project.

Selva

Firmware Solutions CBE-09

Authorized Signature

ACKNOWLEDGEMENT

This project was the most significant accomplishment in my life and it would not have been possible without the blessing of God almighty and those who supported and believed in my caliber.

I record my deep sense of gratitude to **Dr. NALLA. G. PALANISWAMI, M.D, AB (USA) , Chairman , and Dr. THAVAMANI. D. PALANISWAMI , M.D , AB (USA),Secretary, Dr. N.G.P. Research and Educational Trust (Formerly Kovai Medical Center Research and Educational Trust)** for providing me an infrastructure facility to carry out project work successfully.

I record my sincere thanks to Prof**. Dr. S. Saravanan, M.Com., M.Phil., M.B.A., M.B.A., PGDCA., Ph.D., NET., SET, Principal, Dr. N.G.P. Arts and Science College, Coimbatore,** for every help he rendered before and during the project.

I express my sincere thanks to **Dr. A.Adhiselvam, M.Sc., P.B.D.C.A., M.C.A., M.Phil., Ph.D., SET., Professor and Head, Department of Information Technology, Dr. N.G.P. Arts and Science College, Coimbatore** for showing sustained interest and providing help throughout the period of this work.

I would like to extend the sincere thanks to my Guide **Mrs. Sathiya Priya A M.C.A., (Ph.D.,) Assistant Professor, Department of Information Technology, Dr. N.G.P. Arts and Science College, Coimbatore** for the exemplary guidance and encouragement.

I take this opportunity to acknowledge my sincere thanks to all the staff members of the Department of Information Technology for their inspiration, assistance and resourceful guidance for the completion of this project successfully.

I express my sincere thanks to our family and friends for their encouragement, love, prayer, moral support, advice and sacrifice without which we would not have been able to pursue the course of this study.

(BHAWIN A B)

Table of Contents

S.NO	PARTICULARS	PAGE NO.
	DECLARATION	
	CERTIFICATE	
	ACKNOWLEDGEMENT	
	ABSTRACT	**1**
1	**INTRODUCTION**	**2**
	1.1 OVERVIEW OF THE PROJECT	3
	1.2 ABOUT THE ORGANIZATION	4
2	**SYSTEM ANALYSIS**	
	2.1 EXISTING SYSTEM	6
	2.2 PROPOSED SYSTEM	6
3	**SYSTEM REQUIREMENTS**	
	3.1 HARDWARE REQUIREMENTS	8
	3.2 SOFTWARE REQUIREMENTS	8
4	**SYSTEM OVERVIEW**	
	4.1 MODULE LIST	9
	4.2 SOFTWARE DESCRIPTION	11
5	**SYSTEM DESIGN**	
	5.1 SYSTEM FLOW DIAGRAM	20
	5.2 ER DIAGRAM	21
	5.3 FORM DESIGN	22
6	**SYSTEM TESTING**	
	6.1 INTEGRATION TESTING	24
	6.2 FUNCTIONAL TESTING	24
7	**IMPLEMENTATION**	30
8	**CONCLUSION**	31
9	**FUTURE ENHANCEMENT**	32
10	**BIBLIOGRAPHY**	33
11	**APPENDIX**	34
	11.1 SCREENSHOTS	34
	11.2 SAMPLE CODING	43

ABSTRACT

The project entitled “Spare Fix-Car Spare Parts Shopping Application” enables customer to buy car spare parts using mobile application. This application advertises a collection of branded and high-quality car spare parts available for shopping. For shopping, the intended customers must have an authentication. The customers with authentication will avail the entire benefits offer on the application. Customers who do not have the authentication credentials, such customers can access the application and view the products details and services for the customers. Necessary, a user must have an account to become a customer, meanwhile the intended customer can shop the items and place in the available shopping cart. Importantly, the application facilitates the users or customers with billing details and tracks the customer orders with successful shipment acknowledgement. This system which is specialized in trading automobile spare parts claims to provide sufficient spare parts throughout the life cycle of products, to the clients in need, so as to achieve sustainability. This work discusses how such a system can be designed and implemented to assist clients in search of the best car parts that fits their purpose.

1. INTRODUCTION

E-commerce is poised to become one of the most important channels in the automotive market and spare-parts industry. In fact, android-based online shopping applications are growing by double digits while brick and mortar channels are relatively flat. Generally, the vehicle owners keep their vehicle for longer duration depending upon their liking and preferences.

A study reports that the car owners maintain their car averagely more than 11 years. This encourages spare-parts business and opens several opportunities to all the stakeholders involve in the spare-part industries. Precisely, with the growing of online shopping, all the stakeholders find an emerging technology-based platform to facilitate their spare-parts related activities. It enables an easy to access platform for both automotive parts owners and vehicle owners.

With the rapid growth of e-commerce in the automotive industry and aftermarket, it experiences a sizeable increase in customer shopping, behaviors and their preferences. Importantly, the business stakeholders get the customer particulars, behavior, preferences by monitoring shoppers' online navigational activities. The tracking of customers' activities helps retailers to understand customers' preferences and priorities. Based on the customer navigation history, the business stakeholders make strategies to retain the customers and improve the business performance.

1.1 OVERVIEW OF THE PROJECT

The Spare Fix mobile application is designed to revolutionize the way customers purchase car spare parts by providing a seamless, efficient, and digital shopping experience. With the rapid advancement of e-commerce and mobile technologies, the automotive industry has shifted toward online platforms, enabling users to browse and purchase products without visiting physical stores. The Spare Fix application is a response to this digital transformation, offering customers, dealers, and service providers a single platform to buy, sell, and manage car spare parts efficiently.

The automobile industry has witnessed a significant increase in vehicle ownership over the past few decades, leading to a higher demand for spare parts and maintenance services. As vehicles age, their parts wear out, necessitating replacements and repairs. Traditional methods of acquiring spare parts often involve visiting multiple stores, comparing prices manually, and searching for authentic and compatible components, which can be a time-consuming and costly process. Spare Fix eliminates these inefficiencies by providing a centralized online marketplace where users can search for authentic spare parts, make secure payments, and track their orders in real-time. One of the key challenges in the automotive spare parts industry is ensuring the availability and compatibility of parts for various vehicle models. Spare Fix addresses this issue by categorizing spare parts based on brand, model, and specifications, making it easier for users to find the exact components they need. The app is designed with an intuitive interface, allowing users to filter and search for spare parts effortlessly. Additionally, customers can read product descriptions, compare prices, and view customer reviews before making a purchase, ensuring a more informed decision-making process.

Another critical aspect of spare parts shopping is the installation process. Many vehicle owners, especially those without technical knowledge, struggle to install spare parts on their own. Spare Fix bridges this gap by integrating service providers into the platform. After purchasing a spare part, customers can choose to connect with certified mechanics and service centres for installation. The app provides a list of nearby mechanics based on the user's location, allowing them to book an appointment or request doorstep service, making car maintenance more convenient than ever.

The Spare Fix platform is built using cutting-edge mobile application technologies to ensure a seamless and secure shopping experience. The backend system is powered by a robust MySQL database, which manages product inventories, user transactions, and order tracking. The app also incorporates secure payment gateways to facilitate transactions, protecting user data and financial information from potential cyber threats. With a strong focus on user authentication, the platform ensures that only verified customers and dealers can access full functionalities, reducing the risks of fraudulent transactions. Additionally, the application benefits spare parts dealers and suppliers by providing them with an online marketplace to reach a wider audience. Dealers can list their spare parts, update stock availability, and manage customer inquiries through a dedicated dashboard. This feature enhances their business opportunities and enables them to compete in the digital marketplace.

The Spare Fix project represents a significant step forward in the digital transformation of the automotive aftermarket industry. By leveraging the power of mobile technology and e-commerce, the application eliminates traditional challenges associated with spare parts shopping, offering greater convenience, efficiency, and cost-effectiveness. It serves as a one-stop solution for vehicle owners, spare parts dealers, and service providers, ensuring that customers receive not only the right spare part but also the necessary support for installation and maintenance. With its focus on innovation, security, and user-friendliness, Spare Fix is poised to reshape the automotive spare parts industry, making spare parts shopping smarter, faster, and more reliable than ever before.

1.2 ABOUT THE ORGANIZATION

Name of the Company	Firmware Solutions 64/176, Gokale Street, Ram Nagar, (Near Senthil Kumaran Theater) Coimbatore-641009 E-Mail: info@firmwaresolutions.in Mobile: +91 9894938971
Person to contact	Mr. Selva
Products	Website Design and Software
Experience in Trade	15 years
Major Customers	All type of Industries
Working Pattern	9.AM to 6 PM
Bankers	Central Bank of India State Bank of India

Firmware Solutions is a dynamic consulting firm dedicated to delivering comprehensive end-to-end enterprise solutions to our valued customers and partners. The team comprises seasoned professionals who have successfully spearheaded and executed intricate commercial projects worldwide. At the heart of Firmware Solution's ethos lies in commitment to being people-centric, which has earned the trust of their clients as their preferred technology solution provider. Firmware Solutions establish a clear roadmap for the development of effective solutions. Company commitment extends beyond solution deployment, as they provide continued support to ensure stability and enduring results. As one of the most sought-after web development service companies in India, Firmware Solutions boasts a highly skilled team of web developers with decades of expertise. Specializing in the creation of high-end web solutions, Windows/Mobile applications, and more, company focus is on delivering robust user experiences tailored to the unique requirements of various industries. The client relationships are guided by core values, placing clients at the centre of all project activities and striving to maximize the value they provide. Embracing a commitment to continuous improvement, company constantly enhance their services and methodologies to impress clients globally.

2. SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

The existing system for purchasing car spare parts is manual and inefficient, requiring customers to physically visit stores, making the process time-consuming and inconvenient. Finding the right product is difficult due to limited information and lack of search options. Customers must rely on sales staff, visit multiple stores for availability, and face delays due to poor inventory tracking. Payments are mostly cash-based, with no order tracking or after-sales support. The system also fails to personalize recommendations based on customer preferences. A digital, user-friendly solution is necessary to improve accessibility, streamline inventory management, and enhance the overall shopping experience.

2.1.1 DISADVANTAGE

- Customers must physically visit stores, browse through available products, and complete purchases, which can be frustrating, especially for those in remote areas.
- Customers rely on sales staff or product labels, making it difficult to compare different brands, specifications, and prices, leading to poor decision-making.
- This lack of interpretability can hinder the ability of system administrators to make informed decisions based on the output of the NID system.

2.2 PROPOSED SYSTEM

The project entitled “Spare Fix - Car Spare Parts Shopping Application” enables customer to buy car spare parts using mobile application. This application advertises a collection of branded and high-quality car spare parts available for shopping. For shopping, the intended customers must have an authentication. The customers with authentication will avail the entire benefits offer on the application.

Customers who do not have the authentication credentials, such customers can access the application and view the products details and services for the customers. Necessary, a user must have an account to become a customer, meanwhile the intended customer can shop the items and place in the available shopping cart. Importantly, the application facilitates the users or customers with billing details and tracks the customers' orders with successful shipment acknowledgement.

2.2.1 ADVANTAGE

- Develop and design a mobile application for branded car spare parts, ensuring a user-friendly interface that allows customers to browse, select, and purchase genuine spare parts effortlessly.
- Provide better customer service by offering seamless navigation, real-time support, and quick responses to inquiries, ensuring a smooth and satisfactory shopping experience for every user.
- Help customers easily find spare parts for different car brands by integrating a well-organized catalog, advanced search filters, and brand-specific product listings for effortless selection.
- Enable customers to view spare parts updates online from any location, ensuring they stay informed about new arrivals, pricing changes, and availability through real-time notifications.
- Offer quick and easy product comparisons by allowing customers to analyze different spare parts based on specifications, price, and user reviews, ensuring informed purchasing decisions.

3. SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS

- processor : Dual core
- RAM : 2 GB
- Hard Disk : 20 GB

3.2 SOFTWARE REQUIREMENTS

- Language : java, Android
- Tool kit : Android 2.3.3, Android SDK Manager
- IDE : Eclipse / Android studio
- Backend : MySQL 5.0

4.SYSTEM OVERVIEW

4.1 MODULE LIST

- User Interface Module
- Dealer Module
- User Module
- Mechanic Module

4.2 MODULE DESCRIPTION

4.2.1 User Interface Module

Registration - In User registration, fill all the registration details like username, password, address, mobile number and email id. After entering, these details click register button to register user with entered details. Then all the inputs stored in user details table in database.

Login - In user login, enter user mobile no and password for login to this application. The mobile no and password already given while registering user details. This login details verified from database, if it is entered correctly then user will allow to enter our application otherwise shows pop message like check your credentials.

4.2.2 Dealer Module

Add Spare parts - Dealer Add Spare parts, like spare name, model, price.

View Spare parts - Dealer can view already added Spare parts details like spare name, model, price.

4.2.3 User Module

View Spare Parts - User can view already added Spare parts details in like spare name, model, price and Click required spare and give quantity amount of spare after that spare part added to cart.

User Cart - User spare already added in cart with total amount. User can view cart details, and also delete cart items. After adding spare parts to the cart, the user can view cart details, delete items if need, and proceed to confirm the cart, which leads to the payment page. Upon entering card details, if the payment is successful, the system prompts the user to choose whether they need a service provider. If the user selects "Yes," their details are added to the service request, and a list of nearby service providers, filtered by the mechanic's area, is displayed for selection before proceeding to the order success page. If the user selects "No," they are directly redirected to the order success page.

4.2.4 Mechanic Module

Registration - In Mechanic registration, fill all the registration details like username, password, address, mobile number and email id. After entering these details click register button to register user with entered details. Then all the inputs stored in user details table in database.

Login - In Mechanic login, enter user mobile no and password for login to this application. The mobile no and password already given while registering user details. This login details verified from database, if it is entered correctly then Mechanic will allow to enter our application otherwise shows pop message like check your credentials.

View User List - If once user selects service provider, then user's details will have stored in mechanic's view tables. Mechanic can view user list, like user name and area. Then click one user it shows full details of user, and contact user for services.

4.3 SOFTWARE DESCRIPTION

4.3.1 Front End

Java (programming language)

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the java compiler. A .class file does not contain code that is native to your processor; it instead contains bytecodes — the machine language of the Java Virtual Machine1 (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.

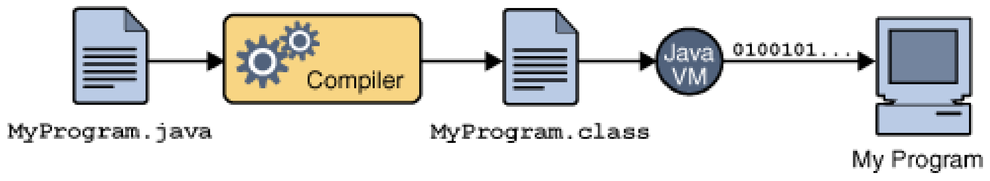


Fig 1: An overview of the software development process.

Because the Java VM is available on many different operating systems, the same `.class` files are capable of running on Microsoft Windows, the Solaris TM Operating System (Solaris OS), Linux, or Mac OS. Some virtual machines, such as the Java Hot Spot virtual machine, perform additional steps at runtime to give your application a performance boost. This includes various tasks such as finding performance bottlenecks and recompiling (to native code) frequently used sections of code. Through the Java VM, the same application is capable of running on multiple platforms.

The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Microsoft Windows, Linux, Solaris OS, and Mac OS. Most platforms can be described as a combination of the operating system and underlying hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

- The Java Virtual Machine
- The Java Application Programming Interface (API)

You've already been introduced to the Java Virtual Machine; it's the base for the Java platform and is ported onto various hardware-based platforms.

The API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, highlights some of the functionality provided by the API.

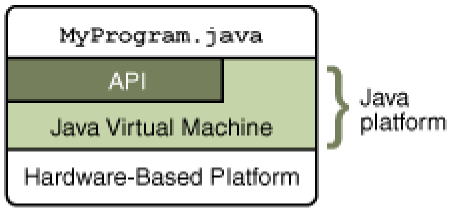


Fig 2: The API and Java Virtual Machine insulate the program from the underlying hardware.

As a platform-independent environment, the Java platform can be a bit slower than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability.

Android

A free, open-source mobile platform. A Linux-based, multiprocessing, Multithreaded OS. Android is not a device or a product It's not even limited to phones You could build a DVR, a handheld GPS, an MP3 player, etc. Android is a software stack for mobile devices that includes an operating system, middleware and key applications.

The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

- Makes mobile development easy.
- Full phone software stack including applications
- Designed as a platform for software development
- Android is open
- Android is free
- Community support

Features

- Application framework enabling reuse and replacement of components
- Dalvik virtual machine optimized for mobile devices
- Integrated browser based on the open-source WebKit engine
- Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- SQLite for structured data storage
- Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
- GSM Telephony (hardware dependent)
- Bluetooth, EDGE, 3G, and WiFi (hardware dependent)
- Camera, GPS, compass, and accelerometer (hardware dependent)
- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

Linux Kernel

Android relies on Linux version 2.6 for core system services such as

- security
- memory management
- process management
- network stack

- driver model
- The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

Android Runtime

- Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.
- Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently.
- The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint.
- The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Libraries

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework.

- System C Library
- Media Library
- Surface Manager
- SGL
- 3D libraries
- Free Type
- SQLite

Application Framework

- Being a open development platform, Android offers developers the ability to build extremely rich and innovative applications.
- Developers have full access to the same framework APIs used by the core applications.

- Views – used to build applications (lists, grid, buttons, text boxes and even embeddable web browser).
- Content providers – enable applications to access data from other applications or share their own data.
- Resource manager – provides access to non-code resources such as localized strings, graphic and layout files.
- Notification manager – enables applications to display custom alerts in status bar.
- Activity manager – manages lifecycle of applications and provides navigation back stack.
- All applications are written using the Java programming language.

Architecture

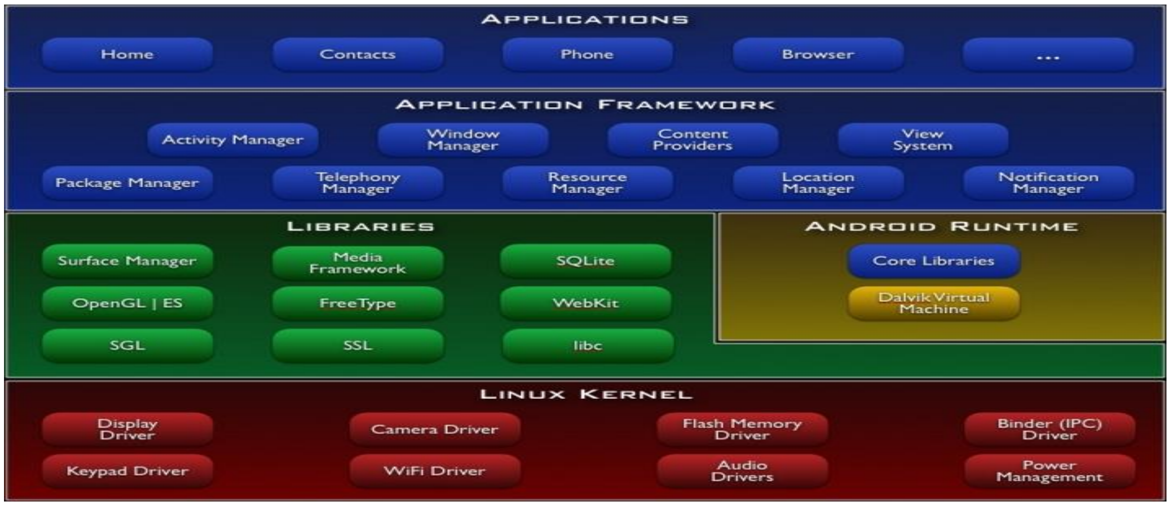


Fig 3: Architecture

Anatomy of an Android Application

There are four building blocks for an Android application:

- Activity - a single screen
- Broadcast Receiver- to execute in reaction to an external event (Phone Ring)
- Service - code that is long-lived and runs without a UI (Media Player)
- Content Provider - an application's data to be shared with other applications

Android Building Blocks

These are the most important parts of the Android APIs:

- AndroidManifest.xml - the control file-tells the system what to do with the top-level components
- Activities - An object that has a life cycle-is a chunk of code that does some work
- Views - An object that knows how to draw itself to the screen
- Intents - A simple message object that represents an "intention" to do something
- Notifications – Provider a small icon that appears in the status bar (SMS messages) for alerting the user
- Services - Is a body of code that runs in the background

Development Tools

The Android SDK includes a variety of custom tools that help you develop mobile applications on the Android platform. These are the most significant tools are:

- Android Emulator -A virtual mobile device that runs on our computer -use to design, debug, and test our applications in an actual Android run-time environment
- Android Development Tools Plugin -for the Eclipse IDE - adds powerful extensions to the Eclipse integrated environment
- Dalvik Debug Monitor Service (DDMS) -Integrated with Dalvik -this tool let us manage processes on an emulator and assists in debugging
- Android Asset Packaging Tool (AAPT) – Constructs the distributable Android package files (.apk)
- Android Debug Bridge (ADB) – provides link to a running emulator. Can copy files to emulator, install .apk files and run commands.

4.3.2 Back End

The MySQL Reference Manual covers most areas of MySQL use. This manual is for both MySQL Community Server and MySQL Enterprise Server. If you cannot find the answer(s) from the manual, you can get support by purchasing MySQL Enterprise, which provides comprehensive support and services. MySQL Enterprise also provides a comprehensive knowledge base library that includes hundreds of technical articles resolving difficult problems on popular database topics such as performance, replication, and migration.

MySQL AB develops and supports a family of high-performance, affordable database products. The company's flagship offering is 'MySQL Enterprise', a comprehensive set of production-tested software, proactive monitoring tools, and premium support services. MySQL is the world's most popular open-source database software. Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, business-critical systems and packaged software -- including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube and Booking.com. With headquarters in the United States and Sweden -- and operations around the world -- MySQL AB supports both open-source values and corporate customers' needs.

JDBC

Java Database Connectivity (JDBC) is a programming framework for Java developers writing programs that access information stored in databases, spreadsheets, and flat files. JDBC is commonly used to connect a user program to a "behind the scenes" database, regardless of what database management software is used to control the database. In this way, JDBC is cross-platform This article will provide an introduction and sample code that demonstrates database access from Java programs that use the classes of the JDBC API, which is available for free download from Sun's site.

A database that another program links to is called a data source. Many data sources, including products produced by Microsoft and Oracle, already use a standard called Open Database Connectivity (ODBC). Many legacy C and Perl programs use ODBC to connect to data sources. ODBC consolidated much of the commonality between database management systems. JDBC builds on this feature, and increases the level of abstraction. JDBC-ODBC bridges have been created to allow Java programs to connect to ODBC-enabled database software.

Application Areas of JDBC

JDBC has been designed and implemented for use in connecting to databases. Fortunately, JDBC has made no restrictions, over and above the standard Java security mechanisms, for complete systems. To this end, a number of overall system configurations are feasible for accessing databases.

- Java application which accesses local database
- Java applet accesses server-based database
- Database access from an applet via a stepping stone

Using a JDBC driver

Java Soft has defined the following driver categorization system:

Type 1

These drivers use a bridging technology to access a database. The JDBC-ODBC bridge that comes with the JDK 1.1 is a good example of this kind of driver. It provides a gateway to the ODBC API. Implementations of that API in turn do the actual database access. Bridge solutions generally require software to be installed on client systems, meaning that they are not good solutions for applications that do not allow you to install software on the client.

Type 2

The type 2 drivers are native API drivers. This means that the driver contains Java code that calls native C or C++ methods provided by the individual database vendors that perform the database access. Again, this solution requires software on the client system.

Type 3

Type 3 drivers provide a client with a generic network API that is then translated into database specific access at the server level. In other words, the JDBC driver on the client uses sockets to call a middleware application on the server that translates the client requests into an API specific to the desired driver. As it turns out, this kind of driver is extremely flexible since it requires no code installed on the client and a single driver can actually provide access to multiple databases.

Type 4

Using network protocols built into the database engine, type 4 drivers talk directly to the database using Java sockets. This is the most direct pure Java solution. In nearly every case, this type of driver will come only from the database vendor.

Regardless of data source location, platform, or driver (Oracle, Microsoft, etc.), JDBC makes connecting to a data source less difficult by providing a collection of classes that abstract details of the database interaction. Software engineering with JDBC is also conducive to module reuse. Programs can easily be ported to a different infrastructure for which you have data stored (whatever platform you choose to use in the future) with only a driver substitution. To begin connecting to a data source, you first need to instantiate an object of your JDBC driver. This essentially requires only one line of code, a command to the Driver Manager, telling the Java Virtual Machine to load the bytecode of your driver into memory, where its methods will be available to your program. The String parameter below is the fully qualified class name of the driver you are using for your platform combination:

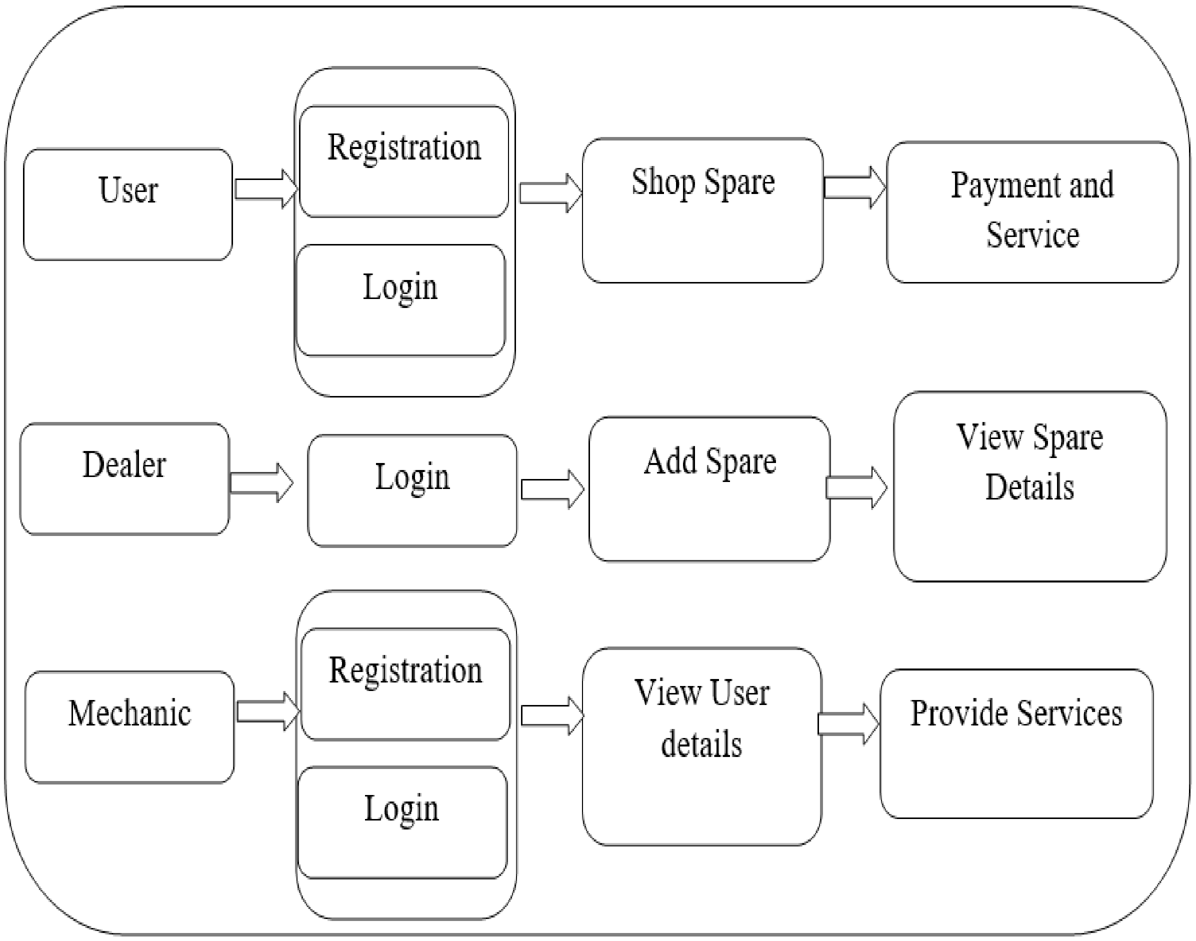
```
Class.forName("org.gjt.mm.mysql.Driver").newInstance();
```

Connecting to a Database

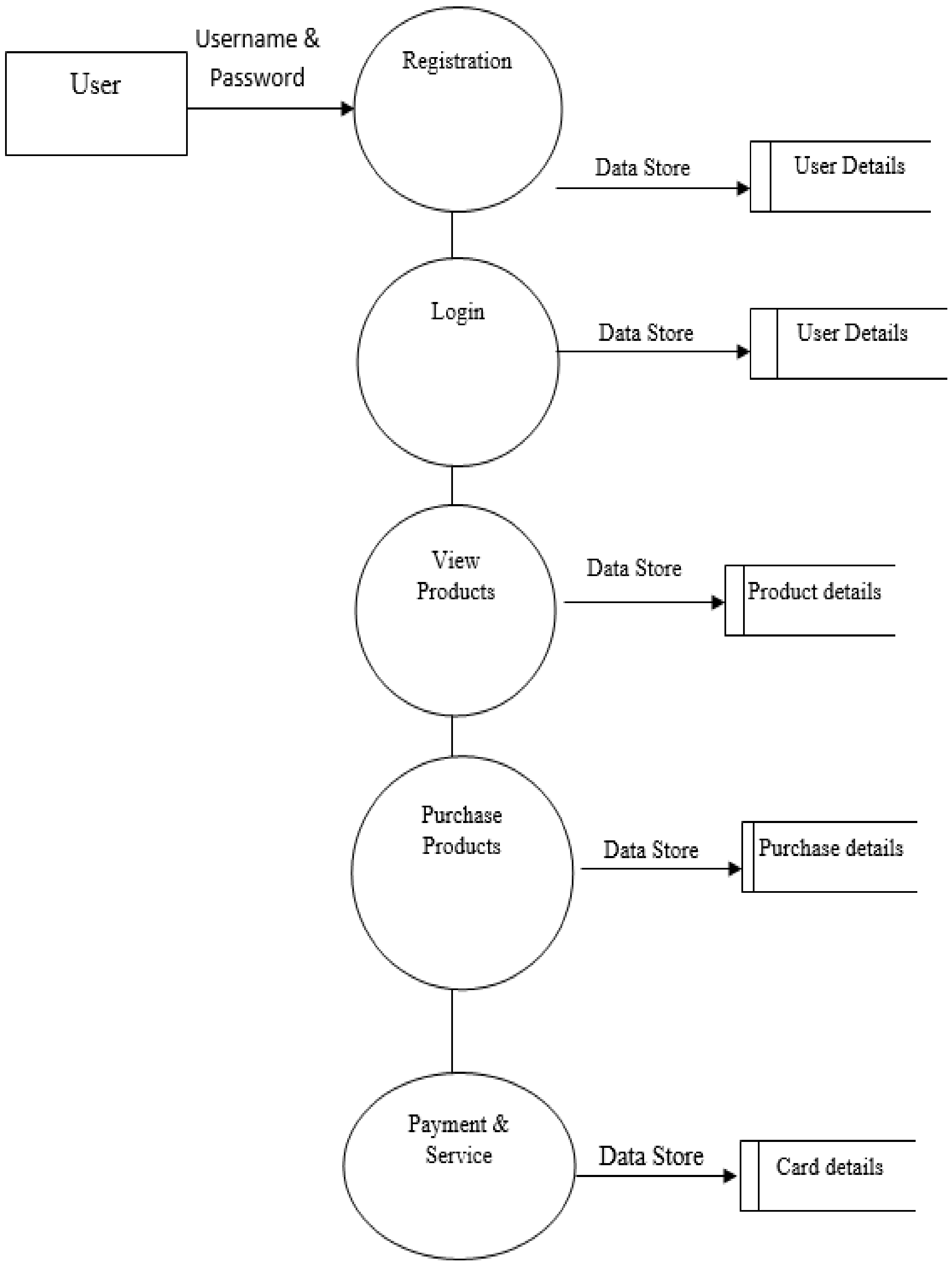
In order to connect to a database, you need to perform some initialization first. Your JDBC driver has to be loaded by the Java Virtual Machine class loader, and your application needs to check to see that the driver was successfully loaded. We'll be using the ODBC bridge driver, but if your database vendor supplies a JDBC driver, feel free to use it instead. To load the JdbcOdbcDriver class, and then catch the ClassNotFoundException, if it is thrown.

5. SYSTEM DESIGN

5.1 SYSTEM FLOW DIAGRAM



5.2 DATA FLOW DIAGRAM



DATABASE DESIGN

A robust database design is crucial for efficiently managing the Spare Fix application, ensuring smooth operations, seamless transactions, and secure data storage. The database must efficiently handle user accounts, spare part listings, inventory, orders, payments, and service requests while maintaining data integrity, security, and performance.

The database schema is structured to support various functionalities, including customer interactions, dealer inventory management, order processing, and mechanic service requests. It consists of multiple tables, each designed to capture specific aspects of the system. For instance, a Users table stores essential customer details, including authentication credentials, contact information, and addresses. The Dealers table maintains records of spare parts suppliers and their respective inventories. The Spare Parts table contains product listings, categorized by brand, model, price, and availability. A Cart table holds temporary selections made by customers before checkout.

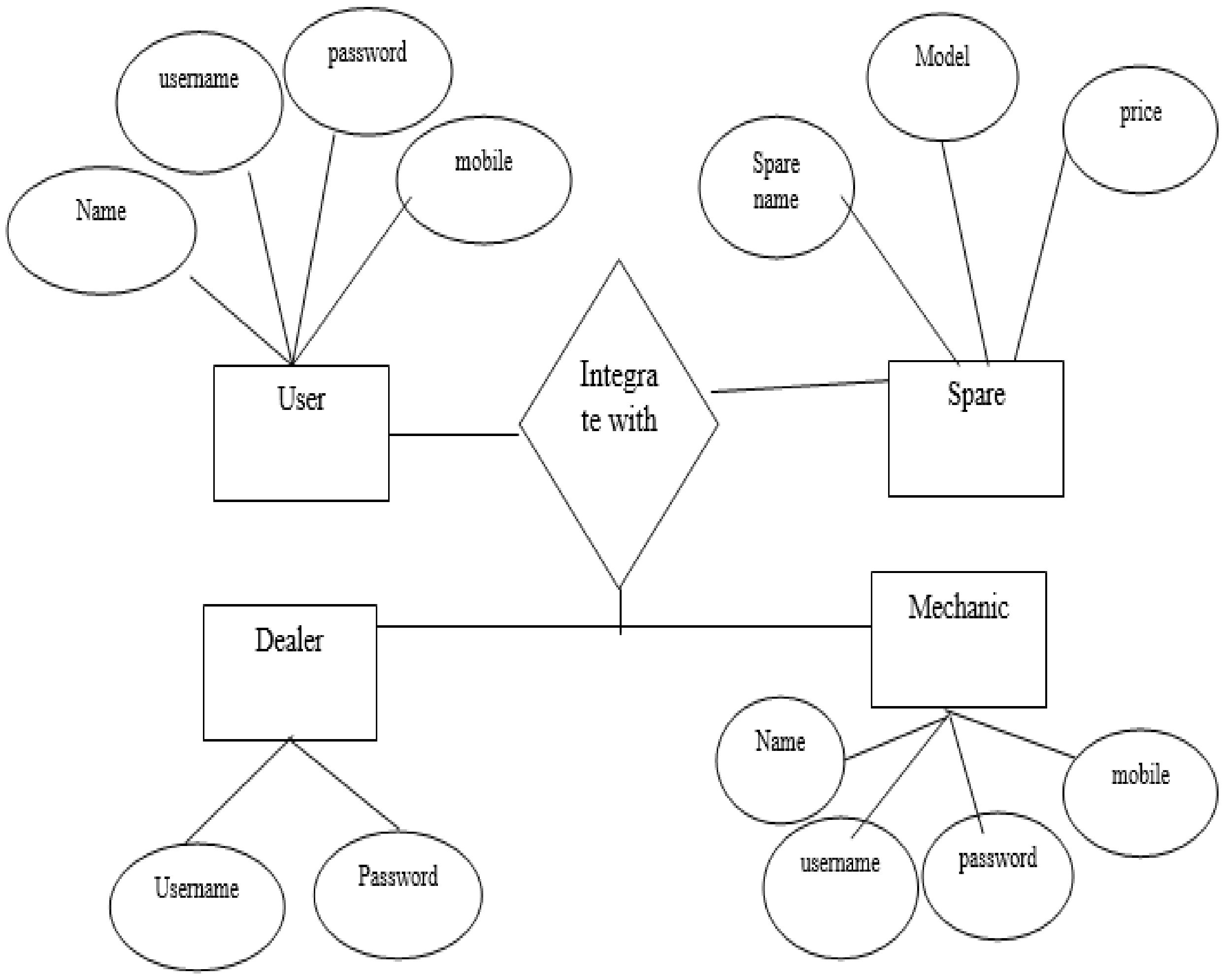
To track purchases, the Orders table manages all transactions, linking customers to their placed orders, while the Order Items table records the specific items within each order. Payments are securely processed and logged in the Payments table, ensuring a transparent record of transactions.

Additionally, the system includes a Mechanics table for registered service providers who assist with part installation and maintenance. The Service Requests table tracks customer requests for mechanic services, ensuring proper coordination between users and service providers.

To optimize performance and security, proper indexing and normalization techniques should be applied, enabling fast querying and efficient data retrieval. Given the sensitivity of transaction data, implementing encryption, secure access controls, and role-based authentication will be essential for safeguarding customer and business information.

The Spare Fix database structure is tailored to support the scalability, reliability, and security of the application. By implementing this well-defined relational database, the system will be capable of handling large volumes of transactions while providing a seamless and efficient shopping experience for customers, dealers, and service providers

5.3 ER DIAGRAM



6. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

Types Of Tests

Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

- Valid Input : Identified classes of valid input must be accepted.
- Invalid Input : Identified classes of invalid input must be rejected.
- Functions : Identified functions must be exercised.
- Output : Identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Feasibility Analysis

Whatever we think need not be feasible .It is wise to think about the feasibility of any problem we undertake. Feasibility is the study of impact, which happens in the organization by the development of a system. The impact can be either positive or negative. When the positives nominate the negatives, then the system is considered feasible. Here the feasibility study can be performed in two ways such as technical feasibility and Economical Feasibility.

Technical Feasibility:

We can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development and maintaining the system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization here we are utilizing the resources which are available already.

Economic Feasibility

Development of this application is highly economically feasible because, the organization needed not spend much money for the development of the system already available. The only thing is to be done is making an environment for the development with an effective supervision. I f we are doing so, we can attain the maximum usability of the corresponding resources Even after the development, the organization will not be in condition to invest more in the organization. Therefore, the system is economically feasible.

File Design

File design of an information system shows the major features and also how they are related to one another. The first step of the system design is to design logical design elements. This is the most creative and challenging phase and important too. Design of proposed system produces the details of the state how the system will meet the requirements identified during the system analysis that is, in the design phase we have to find how to solve the difficulties faced by the existing system. The File design of the proposed system should include the details that contain how the solutions can be implemented. It also specifies how the database is to be built for storing and retrieving data, what kind of reports are to be created and what are the inputs to be given to the system.

Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data into a usable form for processing data entry. The activity of putting data into the computer for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system.

The design of input focuses on controlling the amount of input required, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple. The system needs the data regarding the asset items, depreciation rates, asset transfer, and physical verification for various validation, checking, calculation and report generation. The error raising method is also included in the software, which helps to raise error message while wrong entry of input is done. So, input design the following things are considered.

Output Design

Computer output is the most important and direct information source to the user. Output design is a process that involves designing necessary outputs in the form of reports that should be given to the users according to the requirements. Efficient, intelligible output design should improve the system's relationship with the user and help in decision making. Since the reports are directing referred by the management for taking decisions and to draw conclusions they must be designed with almost care and the details in the reports must be simple, descriptive and clear to the user. So,While designing output the following things are to be considered.

Determine what information to present Arrange the presentation of information in an acceptable format Decide how to distribute the output to intended receipts Depending on the nature and future use of output required, they can be displayed on the monitor for immediate need and for obtaining the hardcopy. The options for the output reports are given in the appendix.

7.IMPLEMENTATION

The Car Spare Parts and Service Provider System is implemented as an Android-based mobile application, ensuring a seamless shopping experience for users. The application allows users to browse and purchase spare parts, manage their cart by adding or removing items, and proceed with the checkout process. The cart system is integrated with an interactive UI, enabling users to review their selected products before confirming their order. Once confirmed, the total amount is calculated, and the system redirects the user to a secure payment gateway for transaction processing. The app validates the card details before proceeding with the payment, ensuring security and reliability.

After the payment is successfully completed, the system provides an option to avail of a service provider for installation or maintenance. If the user opts for a service, the application fetches and displays a list of nearby mechanics based on their location. The user can then select a suitable service provider, and the details are recorded in the system for further processing. If the user declines the service provider option, they are directly redirected to the order success page, confirming their purchase and completing the transaction smoothly.

To enhance system efficiency, the application is backed by a server-side architecture that manages user authentication, order history, and payment transactions. The database securely stores product details, user preferences, and service provider information. Future enhancements may include real-time order tracking, integration with third-party logistics, and additional payment options such as digital wallets and UPI, further improving user experience and system scalability

8.CONCLUSION

Spare Fix revolutionizes the automotive spare parts industry by seamlessly connecting customers, dealers, and service providers on a single digital platform. By addressing inefficiencies in traditional spare parts shopping, it enables users to search, compare, and purchase parts conveniently while also accessing reliable installation and repair services. Built on a robust and secure infrastructure, Spare Fix ensures smooth performance, real-time order tracking, and secure transactions. Beyond benefiting customers, it empowers spare parts dealers by expanding their reach, streamlining inventory management, and enhancing customer engagement. As a pioneer in digital transformation within the automotive aftermarket sector, Spare Fix enhances accessibility, fosters a connected ecosystem, and simplifies vehicle maintenance. With its commitment to quality service, technological innovation, and customer satisfaction, Spare Fix is not just an app—it is the future of car spare parts shopping.

9.FUTURE ENHANCEMENT

- Real-Time Inventory and Order Tracking: The application will integrate real-time stock updates to reflect product availability instantly. Additionally, live order tracking will allow customers to monitor their purchases, providing accurate estimated delivery times and improving shopping transparency
- Augmented Reality (AR) for Spare Part Visualization: The app will incorporate AR technology, enabling customers to visualize spare parts in 3D. This feature will help users ensure compatibility before purchasing, reducing product returns and enhancing the overall shopping experience.
- Voice-Activated Search and AI Chatbots: The app will introduce voice-enabled search features to allow users to find spare parts hands-free. Additionally, AI-powered chatbots will assist customers by providing instant support, troubleshooting guides, and order tracking updates
- Subscription-Based Maintenance Plans: The app will offer subscription-based services where customers can receive scheduled maintenance kits and spare parts tailored to their vehicle's requirements. Subscribers will also benefit from exclusive discounts and priority support
- Subscription-Based Maintenance Plans: The app will offer subscription-based services where customers can receive scheduled maintenance kits and spare parts tailored to their vehicle's requirements. Subscribers will also benefit from exclusive discounts and priority support.
- Sustainability and Eco-Friendly Initiatives: The platform will support environmental sustainability by promoting recycled and eco-friendly spare parts. Additionally, it will encourage proper disposal of used components and collaborate with green automotive businesses to reduce waste.
- Blockchain-Based Transaction Security: Implementing blockchain technology will enhance transaction security, ensuring a fraud-proof, transparent, and tamper-resistant payment system. It will also help verify the authenticity of spare parts, preventing the sale.

10.BIBLIOGRAPHY

[1] Charan, P. (2012), "Supply chain performance issues in an automobile company: a SAP-LAP analysis", Measuring Business Excellence, Vol. 16 No. 1, pp. 67-86. https://doi.org/10.1108/13683041211204680

[2] Kapoor, V. and Ellinger, A.E. (2004), ‘‘Transforming supply chain operations in response to economic reform: the case of a motorcycle manufacturer in India’’, Supply Chain Management: An International Journal, Vol. 9 No. 1, pp. 16-22.

[3] Buyukkaramikli, N. C., Van Ooijen, H. P., & Bertrand, J. W. M. (2015). Integrating inventory control and capacity management at a maintenance service provider. Annals of Operations Research, 231(1), 185-206.

[4] Mohammaditabar, D., Ghodsypour, S. H., & O'Brien, C. (2012). Inventory control system design by integrating inventory classification and policy selection. International Journal of Production Economics, 140(2), 655-659.

[5] Persson, F., & Saccani, N. (2007). Managing the After Sales Logistic Network–A Simulation Study of a Spare Parts Supply Chain. Advances in Production Management Systems, Springer.

[6] Pourakbar, M., Frenk, J. B. G., & Dekker, R. (2012). End-of-Life Inventory Decisions for Consumer Electronics Service Parts. Production and Operations Management, 21(5), 889-906.

[7] Oumaima Bounou, Abdellah El Barkany, Ahmed El Biyaali, "Performance Indicators for Spare Parts and Maintenance Management: An Analytical Study", Journal of Engineering, vol. 2020, Article ID

[8] P. J. Siddique, H. T. Luong, and M. Shafiq, “An optimal joint maintenance and spare parts inventory model,” International Journal of Industrial and Systems Engineering, vol. 29, no. 2, pp. 177–192, 2018.

[9] O. Bounou, A. El Barkany, and A. El Biyaali, “Inventory models for spare parts management: a review,” International Journal of Engineering Research in Africa, vol. 28, pp. 182–198, 2017.

11.APPENDIX

11.1 SCREENSHOTS

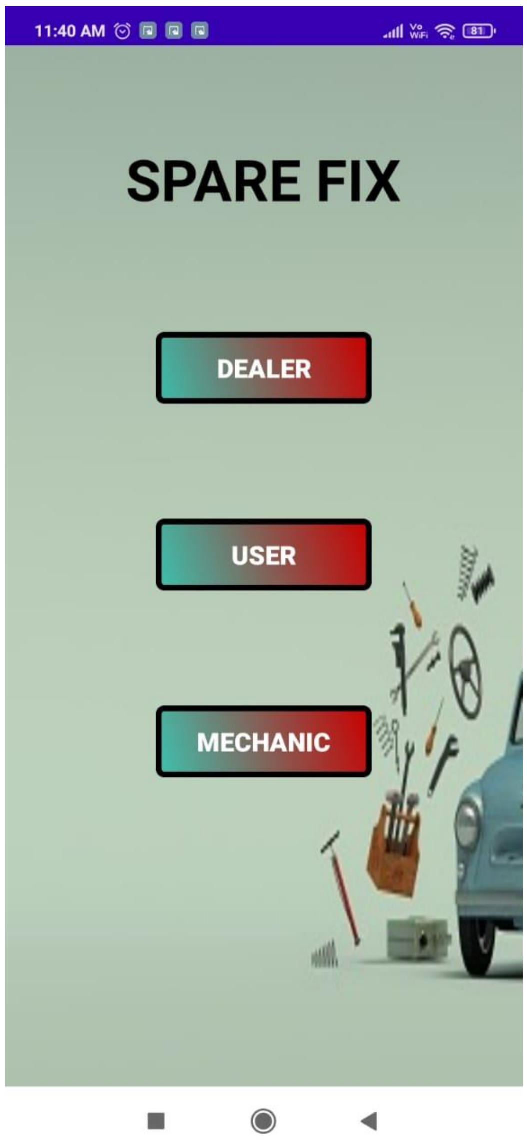


Fig.4: Login Module

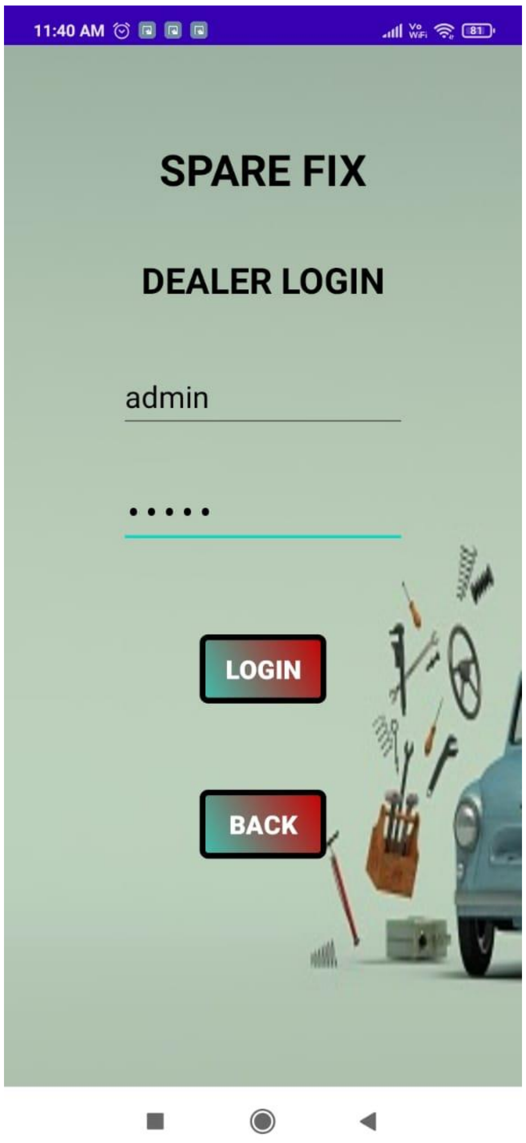


Fig 5: Dealer Login Module

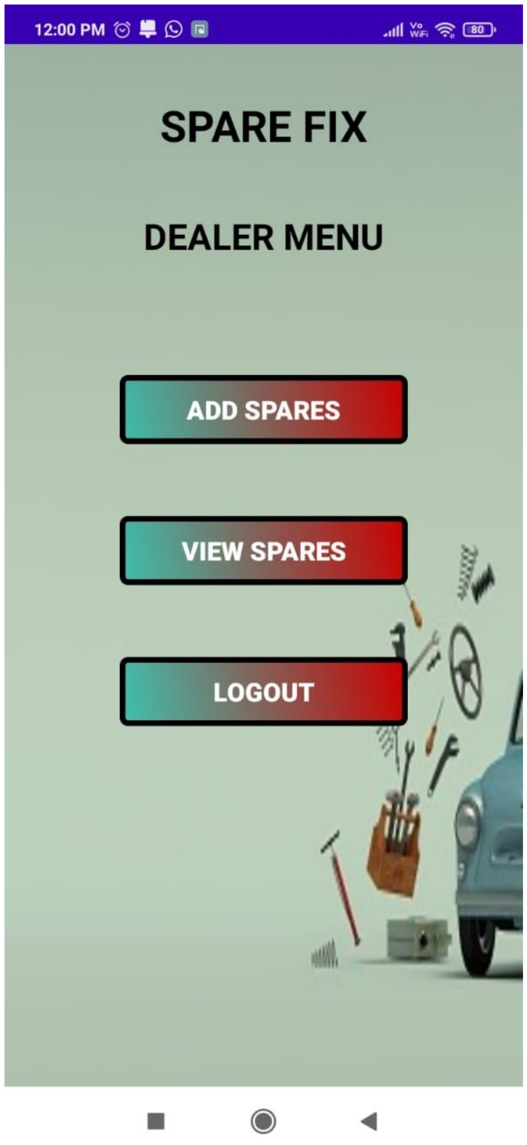
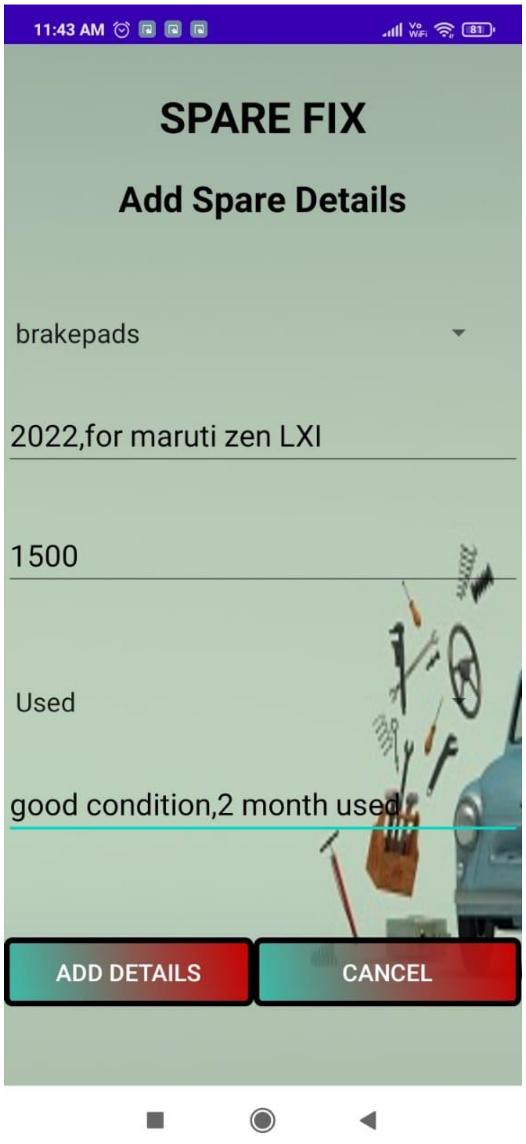


Fig.6: Dealer Menu



brakepads

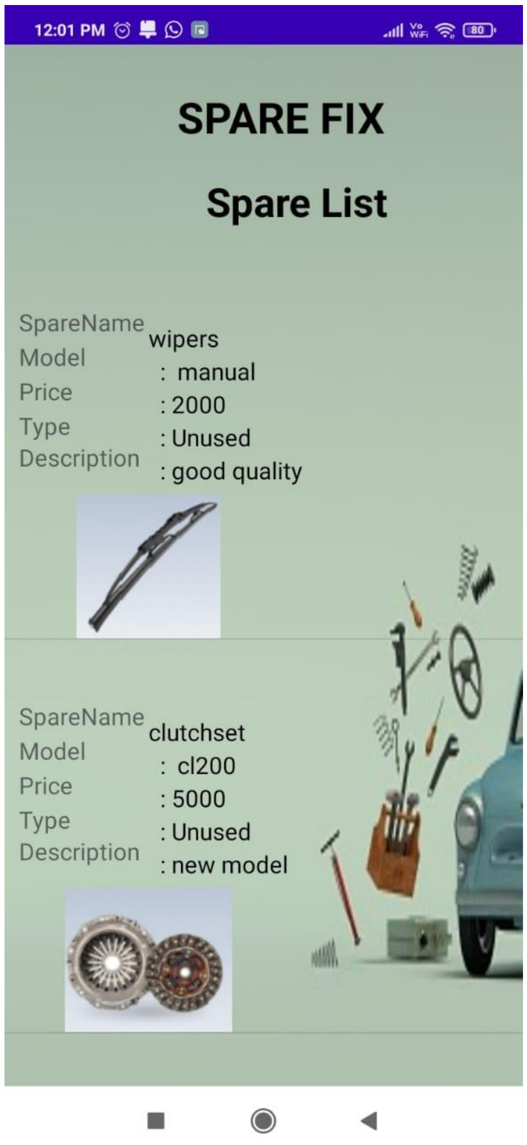
2022,for maruti zen LXI

1500

Used

good condition,2 month used

Fig.7: Adding Of Spare Parts Module



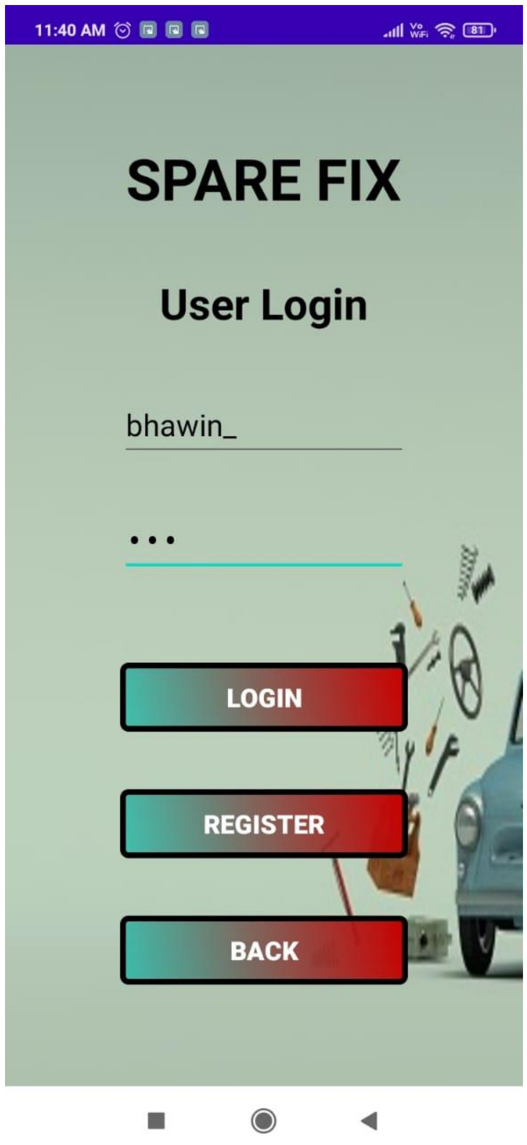
SPARE FIX

Spare List

SpareName	wipers
Model	: manual
Price	: 2000
Type	: Unused
Description	: good quality

SpareName	clutchset
Model	: cl200
Price	: 5000
Type	: Unused
Description	: new model

Fig.8: Spare Listing



SPARE FIX

User Login

bhawin_

•••

Fig.9: Customer Login

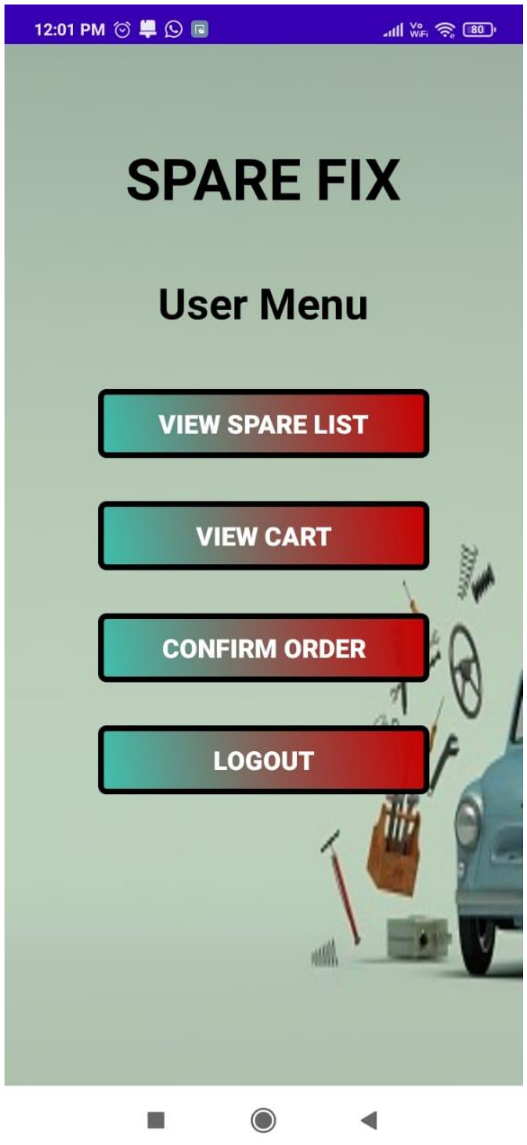


Fig.10: Customer Menu

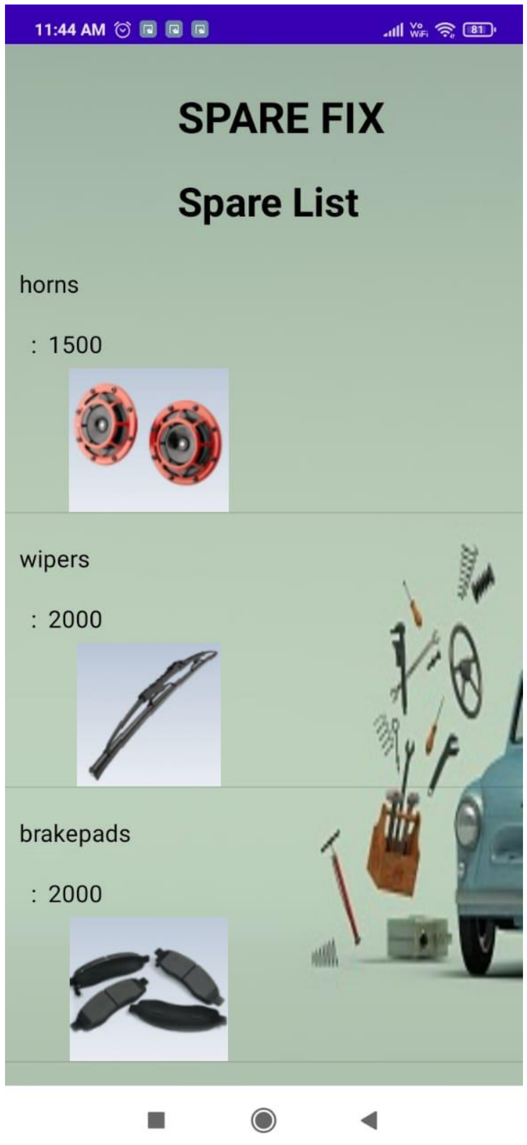
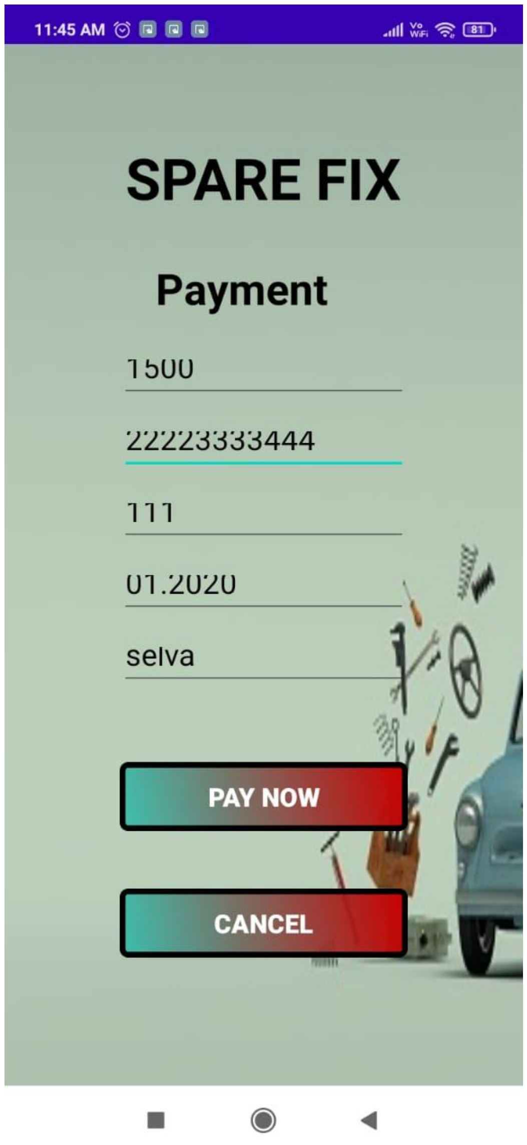


Fig.11: Spare Parts Listing


11:45 AM

SPARE FIX

Payment

1500

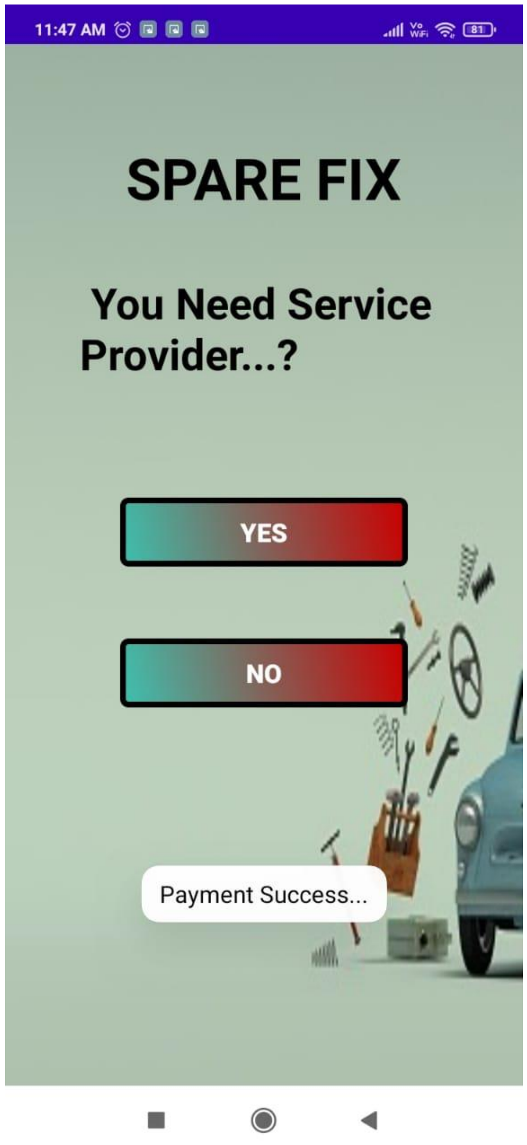
22223333444

111

01.2020

selva

Fig.12: Payment Module

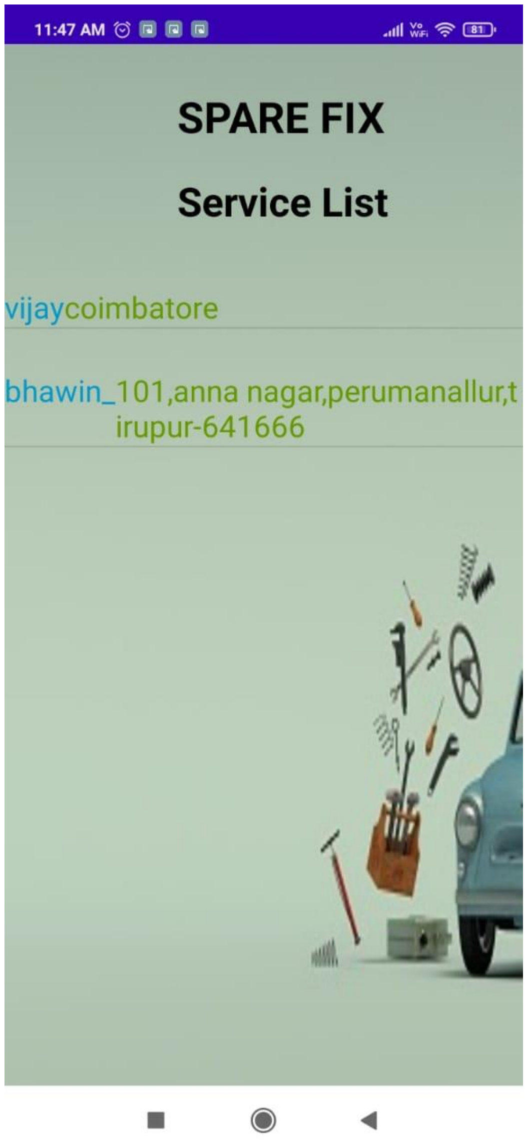

11:47 AM

SPARE FIX

You Need Service Provider...?

Payment Success...

Fig.13: Service Provider Module



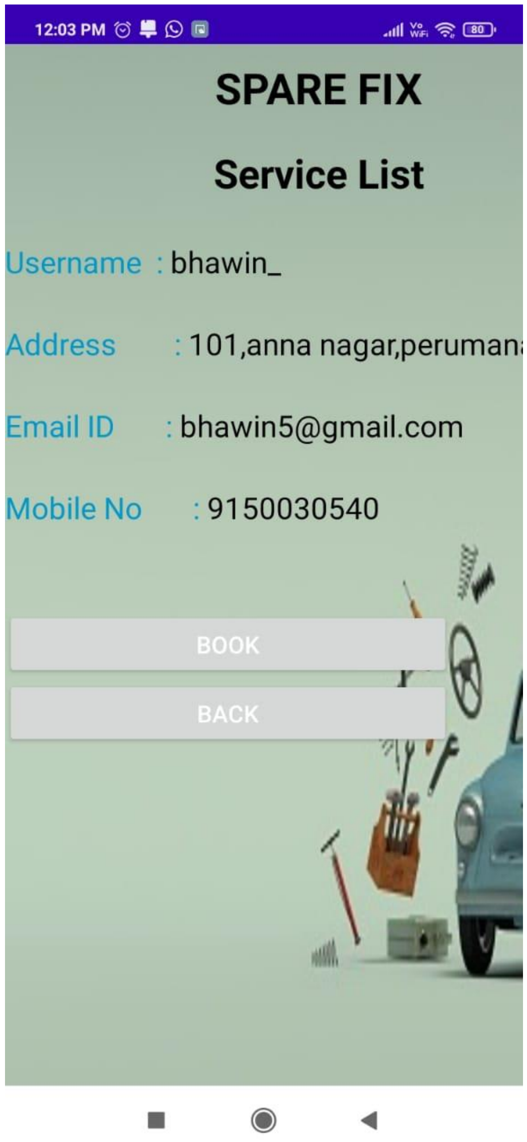
SPARE FIX

Service List

vijaycoimbatore

bhawin_101,anna nagar,perumanallur,tirupur-641666

Fig.14: Service Provider List



SPARE FIX

Service List

Username : bhawin_

Address : 101,anna nagar,perumana

Email ID : bhawin5@gmail.com

Mobile No : 9150030540

Fig.15: Service Provider Details

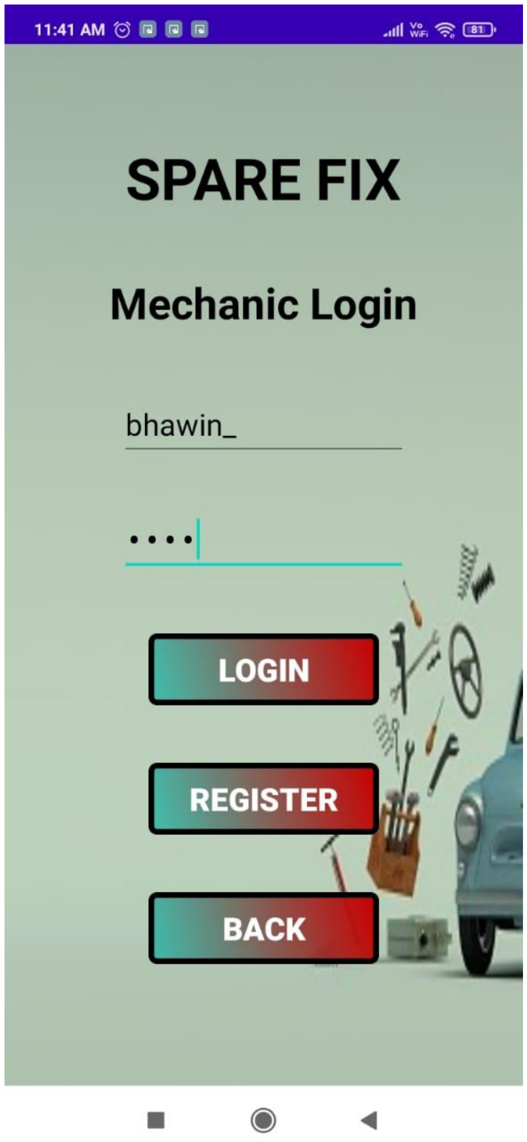


Fig.16: Mechanic Login

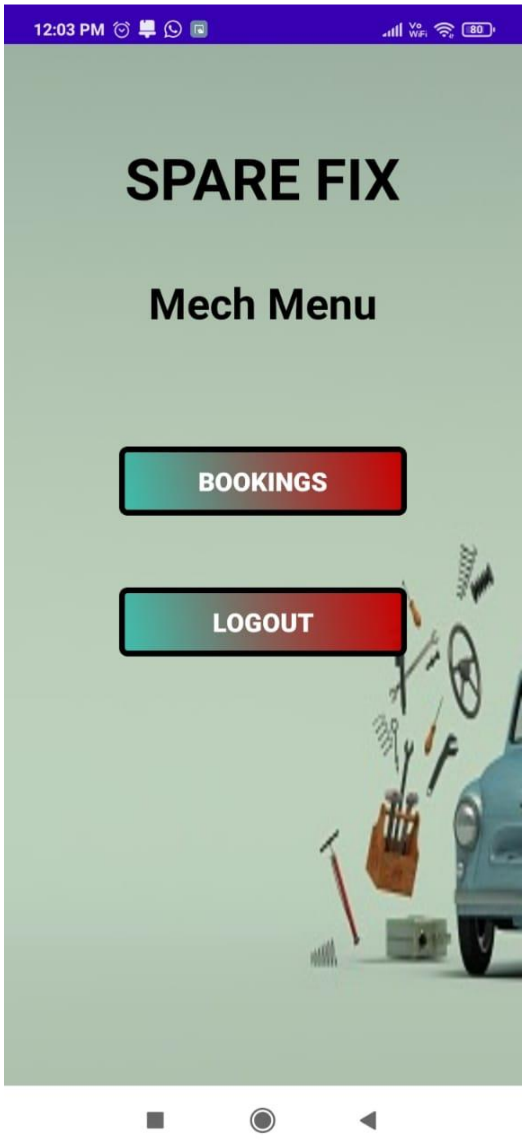
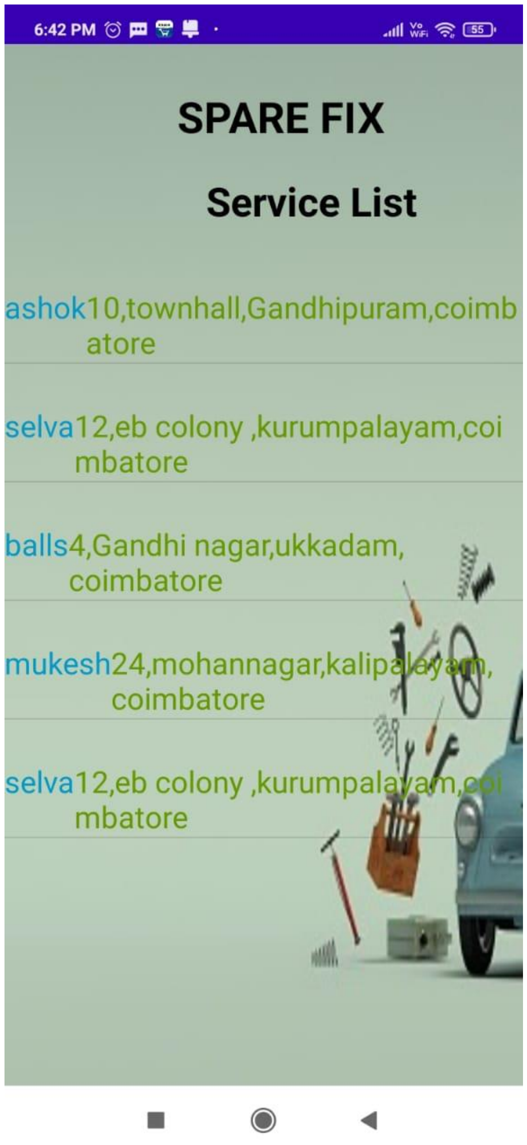


Fig.17: Mechanic Menu



6:42 PM

SPARE FIX

Service List

ashok10,townhall,Gandhipuram,coimbatore

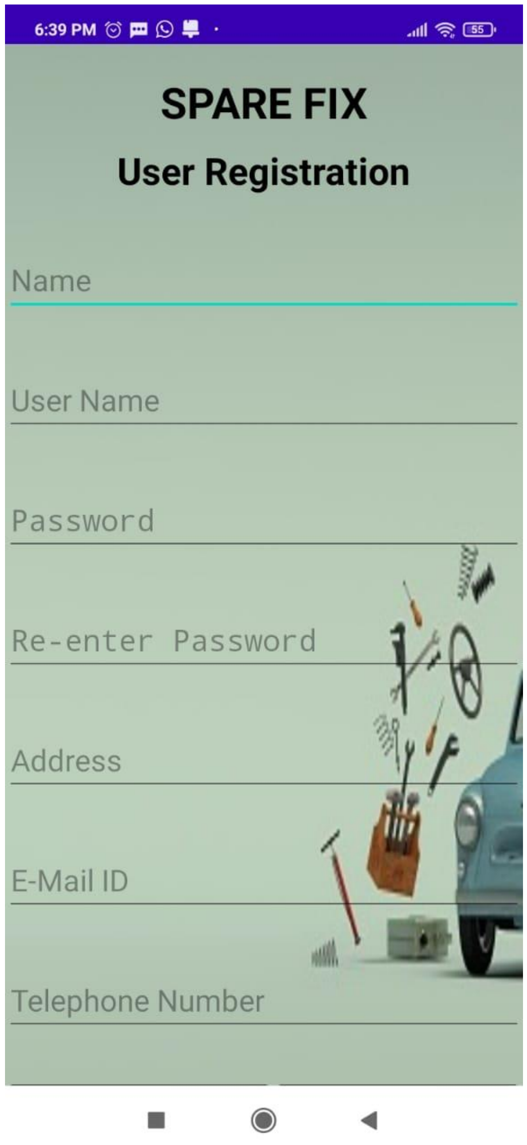
selva12,eb colony ,kurumpalayam,coimbatore

balls4,Gandhi nagar,ukkadam, coimbatore

mukesh24,mohannagar,kalipalayam, coimbatore

selva12,eb colony ,kurumpalayam,coimbatore

Fig.18: Mechanic Booking List



6:39 PM

SPARE FIX

User Registration

Name

User Name

Password

Re-enter Password

Address

E-Mail ID

Telephone Number

Fig.19: User Registration

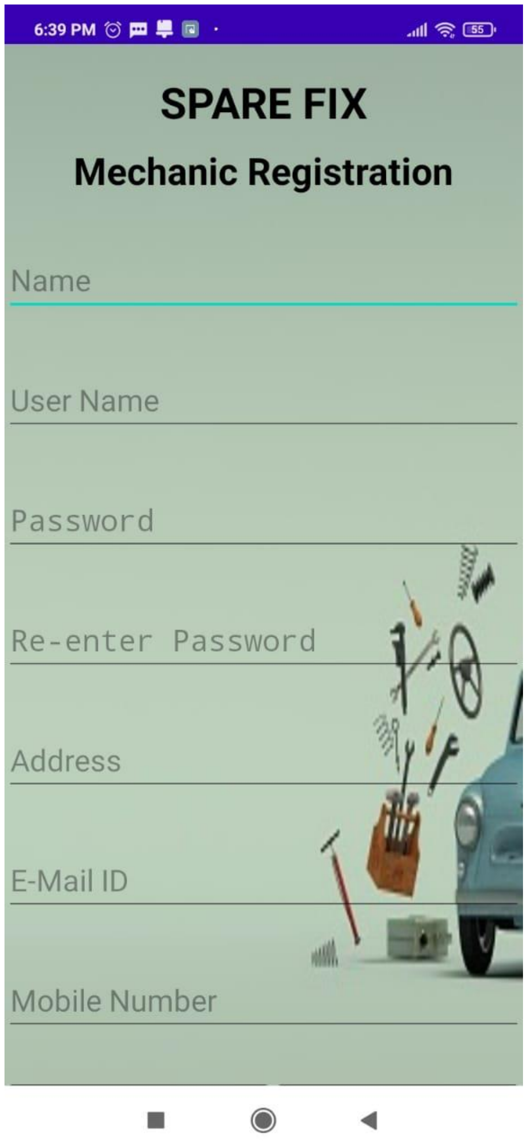


Fig.20: Mechanic Registration

11.2 SAMPLE CODING

```
 package com.prop.carspare;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import org.json.JSONException;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class UserLoginActivity extends MainActivity {
 Connection conn;
 EditText username,password,hostIP;
 Button signin,signup,backbtn;
 String user,pass,user1,pass1;
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.user_login);
 signin=(Button)findViewById(R.id.userloginbtn);
 signup=(Button)findViewById(R.id.regbtn);
 backbtn=(Button)findViewById(R.id.backbtn1);
 username=(EditText)findViewById(R.id.edtusername);
 password=(EditText)findViewById(R.id.edtpassword);
// conn=CONN();
 signin.setOnClickListener(new OnClickListener(){
 @Override
 public void onClick(View v) {
 // TODO Auto-generated method stub
 user=username.getText().toString();
 pass=password.getText().toString();
```

```
        SharedPreferences.Editor editor
=getSharedPreferences("username",Context.MODE_PRIVATE).edit();
        editor.putString("username",user);
        editor.commit();
        editor.apply();
        new QuerySQL().execute(user,pass);
        }
        });
        signup.setOnClickListener(new OnClickListener(){
        @Override
        public void onClick(View v) {
        // TODO Auto-generated method stub
        startActivity(new Intent(UserLoginActivity.this,RegisterActivity.class));
        }
        });
        backbtn.setOnClickListener(new OnClickListener(){
        @Override
        public void onClick(View v) {
        // TODO Auto-generated method stub
        startActivity(new Intent(UserLoginActivity.this,MainActivity.class));
        }
        });
        }
        public class QuerySQL extends AsyncTask<String, Void, Boolean> {
        ProgressDialog pDialog ;
        Exception error;
        ResultSet rs;
        @Override
        protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new ProgressDialog(UserLoginActivity.this);
        pDialog.setTitle("Authentication");
        pDialog.setMessage("Verifying your credentials...");
        pDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
        }
        @Override
        protected Boolean doInBackground(String... args) {
        user1 = new String(args[0]);
        pass1 = new String(args[1]);
        try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection("jdbc:mysql://mysql-75344-
0.cloudclusters.net:18880/carspare","admin","cU5zYktH");
        } catch (SQLException se) {
        Log.e("ERRO1",se.getMessage());
        } catch (ClassNotFoundException e) {
        Log.e("ERRO2",e.getMessage());
        } catch (Exception e) {
        Log.e("ERRO3",e.getMessage());
        }
        try {
        String COMANDOSQL="select * from userdetails where username='"+user1+"' &&
password='"+pass1+"'";
        Statement statement = conn.createStatement();
        rs = statement.executeQuery(COMANDOSQL);
        if(rs.next()){
        String addr = rs.getString(4);
        String mobilenumber = rs.getString(6);
        SharedPreferences.Editor editor
=getSharedPreferences("address",Context.MODE_PRIVATE).edit();
        editor.putString("address",addr);
        editor.commit();
        editor.apply();
        SharedPreferences.Editor editor1
=getSharedPreferences("mobile",Context.MODE_PRIVATE).edit();
        editor1.putString("mobile",mobilenumber);
        editor1.commit();
        editor1.apply();
        String query = "delete from temp where username = ?";
        PreparedStatement preparedStmt = conn.prepareStatement(query);
        preparedStmt.setString(1, user1);
        // execute the preparedstatement
        preparedStmt.execute();
        return true;
        }
        return false;
        // Toast.makeText(getBaseContext(),
        // "Successfully Inserted.", Toast.LENGTH_LONG).show();
        } catch (Exception e) {
        error = e;
        return false;
       // Toast.makeText(getBaseContext(),"Successfully Registered...",
Toast.LENGTH_LONG).show();
        }
        }
        @SuppressLint("NewApi")
        @Override
        protected void onPostExecute(Boolean result1) {
        pDialog.dismiss ( ) ;
        if(result1)
        {
        // System.out.println("ELSE(JSON) LOOP EXE");
        try {// try3 open
        Intent intent=new Intent(UserLoginActivity.this,UserActivity.class);
        //intent.putExtra("latitude", lati);
        //intent.putExtra("longitude", longi);
        //intent.putExtra("loginuser", user1);
        startActivity(intent);
        } catch (Exception e1) {
        Toast.makeText(getBaseContext(), e1.toString(),
        Toast.LENGTH_LONG).show();
        }
        }else
        {
        if(error!=null)
        {
        Toast.makeText(getBaseContext(),error.getMessage().toString()
,Toast.LENGTH_LONG).show();
        }
        else
        {
        Toast.makeText(getBaseContext(),"Check your credentials!!!"
,Toast.LENGTH_LONG).show();
        }
        }
        super.onPostExecute(result1);
        }
        }
       }
```