A PROJECT REPORT
ON

Mend Your Thing

A project report Submitted to
Gujarat Technological University in Partial Fulfillment of the Requirement
BACHELOR OF ENGINEERING
IN
[COMPUTER SCIENCE & ENGINEERING]

Developed By:

PATEL HIMANI JIGNESHKUMAR (120340131018)
PATEL HANI SAMIRKUMAR (120340131028)

Guided By:

Mrs. Vatika Tayal

[Assistant Prof., Dept. Of Computer Science & Engineering, NSIT]

Department of Computer Science & Engineering NARNARAYAN SHASTRI INSTITUTE OF TECHNOLOGY, JETALPUR. GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD. [April-2016]

CERTIFICATE

This is to certify that project work embodied in this report entitled **"Mend Your Thing"** was carried out by **"HIMANI PATEL"** [Enrolment No:120340131018] of Narnarayan Shastri Institute of Technology for practical fulfillment of the award of degree of bachelor of engineering from Gujarat Technological University during the academic session 2015-2016.

Assistant Prof. Mrs. Vatika Tayal

[Internal guide]
NSIT, Jetalpur.

Mr. PIYUSH PATEL

[HeadofDepartment]
Department of Computer Science & Engineering
NSIT, Jetalpur.

Dr. M. SARDA DEVI

[Principal]
NSIT, Jetalpur.

CERTIFICATE

This is to certify that project work embodied in this report entitled **"Mend Your Thing"** was carried out by **"HANI PATEL"** [Enrolment No:120340131028] of Narnarayan Shastri Institute of Technology for practical fulfillment of the award of degree of bachelor of engineering from Gujarat Technological University during the academic session 2015-2016.

Assistant Prof. Mrs. Vatika Tayal

[Internal guide]
NSIT, Jetalpur.

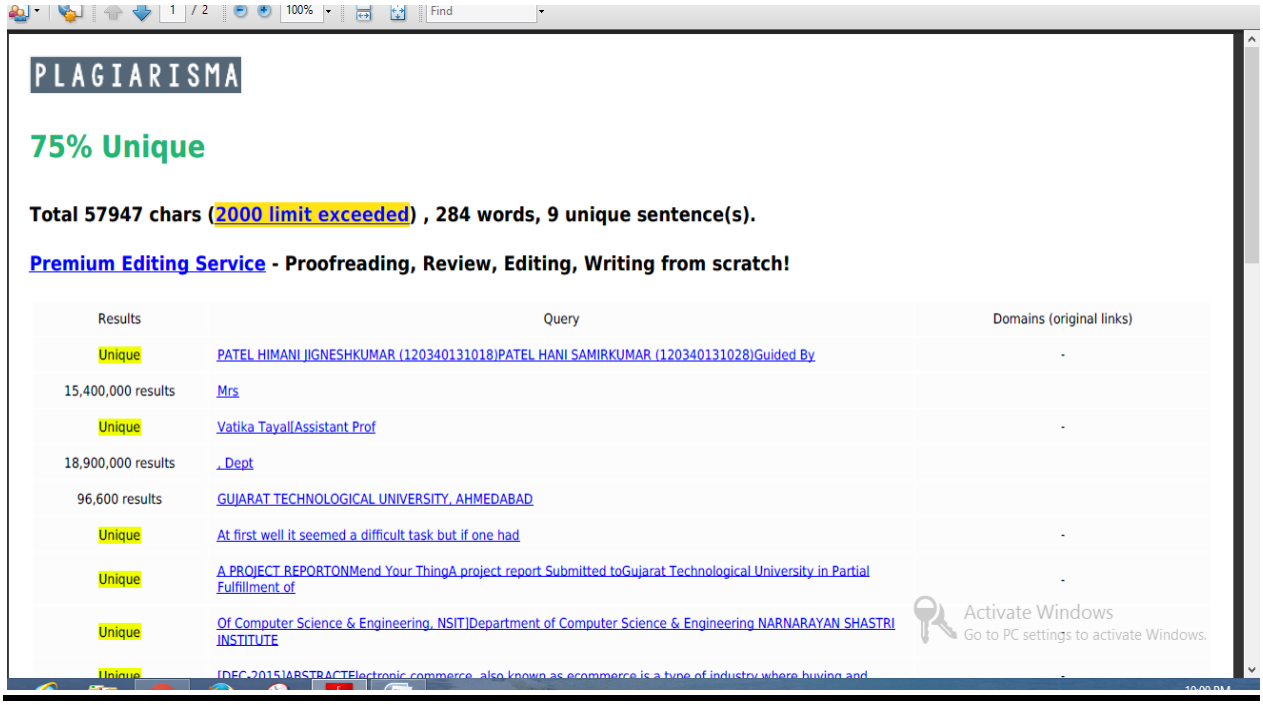
Mr. PIYUSH PATEL

[HeadofDepartment]
Department of Computer Science & Engineering
NSIT, Jetalpur.

Dr. M. SARDA DEVI

[Principal]
NSIT, Jetalpur.

Plagiarism Report:=



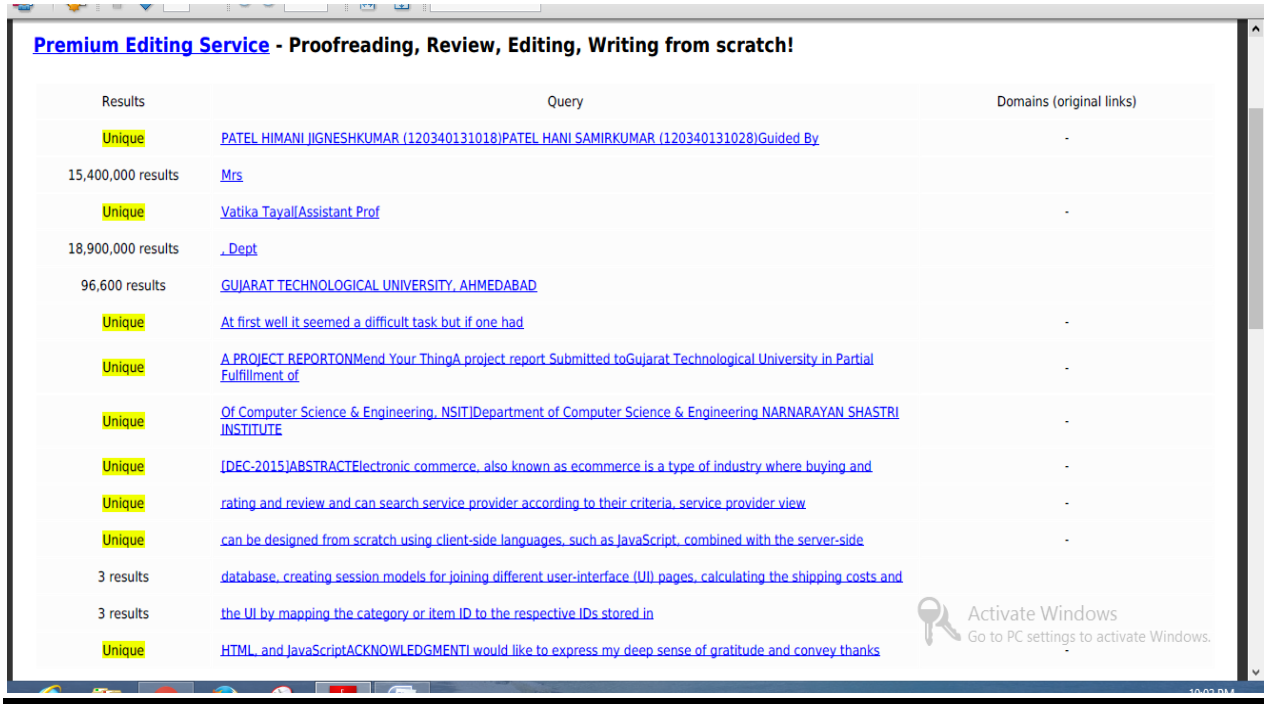
PLAGIARISMA

75% Unique

Total 57947 chars (2000 limit exceeded) , 284 words, 9 unique sentence(s).

Premium Editing Service - Proofreading, Review, Editing, Writing from scratch!

Results	Query	Domains (original links)
Unique	PATEL HIMANI JIGNESHKUMAR (120340131018)PATEL HANI SAMIRKUMAR (120340131028)Guided By	-
15,400,000 results	Mrs	
Unique	Vatika Tayal[Assistant Prof	-
18,900,000 results	, Dept	
96,600 results	GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD	
Unique	At first well it seemed a difficult task but if one had	-
Unique	A PROJECT REPORTONMend Your ThingA project report Submitted toGujarat Technological University in Partial Fulfillment of	-
Unique	Of Computer Science & Engineering, NSIT]Department of Computer Science & Engineering NARNARAYAN SHASTRI INSTITUTE	-
Unique	[DEC-2015]ABSTRACTElectronic commerce, also known as ecommerce is a type of industry where buying and	-



Premium Editing Service - Proofreading, Review, Editing, Writing from scratch!

Results	Query	Domains (original links)
Unique	PATEL HIMANI JIGNESHKUMAR (120340131018)PATEL HANI SAMIRKUMAR (120340131028)Guided By	-
15,400,000 results	Mrs	
Unique	Vatika Tayal[Assistant Prof	-
18,900,000 results	, Dept	
96,600 results	GUJARAT TECHNOLOGICAL UNIVERSITY, AHMEDABAD	
Unique	At first well it seemed a difficult task but if one had	-
Unique	A PROJECT REPORTONMend Your ThingA project report Submitted toGujarat Technological University in Partial Fulfillment of	-
Unique	Of Computer Science & Engineering, NSIT]Department of Computer Science & Engineering NARNARAYAN SHASTRI INSTITUTE	-
Unique	[DEC-2015]ABSTRACTElectronic commerce, also known as ecommerce is a type of industry where buying and	-
Unique	rating and review and can search service provider according to their criteria, service provider view	-
Unique	can be designed from scratch using client-side languages, such as JavaScript, combined with the server-side	-
3 results	database, creating session models for joining different user-interface (UI) pages, calculating the shipping costs and	
3 results	the UI by mapping the category or item ID to the respective IDs stored in	
Unique	HTML, and JavaScriptACKNOWLEDGMENTI would like to express my deep sense of gratitude and convey thanks	-

ABSTRACT

Electronic commerce, also known as ecommerce is a type of industry where buying and selling of a product is conducted over electronic systems such as the internet.

The purpose of this application is to provide online platform to service provider and customer for repairing where customer post their reparable item with description and also give feedback as rating and review and can search service provider according to their criteria, service provider view post of customer and deal with them.

And purpose is also to bring knowledge about ecommerce and how an interactive ecommerce application can be designed from scratch using client-side languages, such as JavaScript, combined with the server-side Java language through Java Server Faces. The server side, mostly Java, contains all the implementation related to setting up the database, creating session models for joining different user-interface (UI) pages, calculating the shipping costs and sales tax, etc. It is responsible for taking information from the database and making it available to the UI by mapping the category to the respective IDs stored in the database. The client side is responsible for showing the entire user interface, containing the CSS, HTML, and JavaScript.

ACKNOWLEDGMENT

I would like to express my deep sense of gratitude and convey thanks to everyone who helped me and supported during the completion of this project.

At first well it seemed a difficult task but if one had a support like what I had, it seems nothing is impossible. I consider my fortunate to be served this grand feast of knowledge and experience for last precious months I spent at my college Narnarayan Shastri Institute of Technology, Jetalpur.

First, I would like to express a deep sense of gratitude to my HOD Mr. Piyush Patel for helping, guiding, and supporting me throughout my project work. I also convey thanks to my all faculty for helping me from time to time. I acknowledge my department for providing the courses and a great atmosphere that helped complete different chapters of this project.

I especially thank my Internal Guide, Mrs. Vatika Tayal for supporting me and my concepts and for allowing me to do something the way I liked, and for helping me develop the skills necessary to design this application.

Last but not least, I would like to thank my family members for their constant and unrelenting support towards my education and for their impartial love for me. I would also like to thank my friends, without whom this project would have been impossible.

List of Table

List of Figure

INDEX

CHAPTER-1: INTRODUCTION

It is known globally that, in today’s market, it is extremely difficult to start a new small-scale business of repairing and its sustenance with competition from the well-established and settled/brand owners shop. Most often, even if the service of the product is really good, due to a lack of advertisement or business at the small scale, it just becomes another face in the sea, and at the small scale repairing service of the product does not reach a larger group of customers. In fast paced life of today when everyone is squeezed for time, the majority of people are finicky when it comes to doing physical.

Logistically, a consumer finds a product’s repairing service more interesting and attractive when they find it on the web application directly and are able to see item’s details online. The customers of today are not only attracted because online repairing insentient , but also because they have broader selections, highly competitive prices, better information about the service provider’s service (including people’s reviews) and extremely simplified navigation for searching regarding the product and service provider. Moreover, service provider give offer online repairing options at low rates because of highly competition. Further, with online repairing, their repairing service have access to a worldwide market, which increases the number of customers from different ethnic groups, adds customer value, and overall sustainable in the marketing.

1.1 Motivation

The motivation for designing this "Mend Your Thing" repairing application came from searching people's common problem or need. And because repairing is basic need of every people. In day to day life, people face lots of trouble for repair there items like….

If we want to repair our things, we must go to market or call technician at home to fix it. And In that case for repair items, we have limited choice or sometimes no choice. We have to pay money as technician want. So fix those problems, motivated to build this application.

Moreover, I value recent learning about the Java and JavaScript programming languages as well as seeing how powerful and dynamic they are when it comes to web designing and applications. Apart from helping computer science students understand the concepts of web-application designing, it would be very easy to incorporate the idea of using programming techniques from the available visuals to understand how a piece of code appears on a user interface. The languages used to build this application are JavaScript, HTML, and Java because I found them to be extremely useful while working on the technologies at my workplace.

1.2. Project Purpose

This software is developed to help me as a computer science student to learn about application designing using JavaScript and HTML from my basic capabilities. This application allows me to understand the basics about the appearance of a first web page and how a complete working application can be built from scratch. It allows me to understand the concept of user-integrated graphics and how JavaScript can be embedded into HTML. Further, it gives insight about how the client-side language interacts with the server-side language, Java, and finally with the database.

This "Mend Your Thing" application is designed, primarily, for me to learn and understand the concept of application development, and can also be used to teach ecommerce and web-application topics.

The application can be downloaded and installed on different machines. And propose of this project for user of this application is, by using this application customer can mend their any item just on one Application.

It works on both side, customer as well as service provider side, so both get benefited and both get profit. Give best service to the customer. By providing online platform, save the precious time and money of the customers.

1.3 Project Scope

In order to develop a reasonable project plan, it is required to functionally bind characteristics of the software so that by following the software engineering principles. The project management activity is the determination of software scopes. By considering the following objectives of software scopes, we derived the software scope statements.

- Software must be ambiguous and understandable at management and technical levels.
- There must be quantitative data products.

Software scope is defined by Activity context, information objectives, function and performance.

1.3.1 CONTEXT

Our Web Application provides online repairing system. For easy perspective & use to upload picture, give description, searching, dealing of reparable items and giving offers, best service from service provider according to requirements.

1.3.2 INFORMATION OBJECTIVES

- Customer Information
- Service provider Information
- Admin Information

- Date Wise Updated Information
- Item Information
- Deal Information
- Payment information

1.3.3 FUNCTION & PERFORMANCE

Major functions of our system are providing online platform to customer And Service provider for repairing.

- Admin can manage and update database.
- Customer has freedom to find service provider for items at minimum Cost.
- Service provider gets more customers then at shop.
- Customer give review and rating so other viewer can find best service.

1.4 Technology Overview

JAVA

- Java is important and exciting advance in software technology. It enables you to build platform independent programs that executes on wide variety of hardware and software environments. Developers around the world are now using Java to build software for consumer devices; personal computers, minicomputers, and main frames .They are constructing mission-critical applications for many different industries.

- One of the important environments in which java is found is the internet.web pages can include references to small java programs known as applet that are dynamically retrieved from Web server and download to a user's machine Applets execute within the environment provided by web browser. They provide powerful ways for a user to interact with a webpage.

- The language includes several significant features. It is object-

oriented and. therefore enables substantial code reuse. Exceptional handling is provided. This allows you to handle run time problems in an organized manner. Garbage collection

- is used to automatic reclaim the memory resources of objects that are no longer used. The java syntax is very straight forward. Some of the more troublesome features of the other language were deliberately omitted. Support for multithreaded program is available. These and other features make it much easier to write robust programming in Java.

- The java class library provides a wealth of functionality. Java is totally object-oriented programming.
- In Java, whatever we can do using the object.

- is used to automatic reclaim the memory resources of objects that are no longer used. The java syntax is very straight forward. Some of the more troublesome features of the other language were deliberately omitted. Support for multithreaded program is available. These and other features make it much easier to write robust programming in Java.

- The java class library provides a wealth of functionality. Java is totally object-oriented programming.
- In Java, whatever we can do using the object.

Java is Platform Independent

- Java was designed to not only be cross-platform in source form like C, but also in compiled binary form. Since this is frankly impossible across processor architectures, Java is compiled to an intermediate form called byte-code.

- A Java program never really executes natively on the host machine. Rather a special native program called the Java interpreter reads the byte code and executes the corresponding native machine instructions. Thus to port Java programs to a new platform, all you need to do is run it with an interpreter written for the new platform. You don't even need to recompile. Even the compiler is written in Java. The byte codes are precisely defined, and remain the same on all platforms.

- The second important part of Java's cross-platform savvy is

  the elimination of undefined and architecture dependent constructs. Integers are always four bytes long, and floating

- point variables follow the IEEE 754 standard for computer arithmetic exactly. You don't have to worry that the meaning of an integer is going to change if you move from a Pentium to a PowerPC. In Java everything is guaranteed.

- However the virtual machine itself and some parts of the class library must be written in native code. These are not always as easy or as quick to port as pure Java programs. This is why for example, there's not yet a version of Java 1.2 for the Mac.

- point variables follow the IEEE 754 standard for computer arithmetic exactly. You don't have to worry that the meaning of an integer is going to change if you move from a Pentium to a PowerPC. In Java everything is guaranteed.

- point variables follow the IEEE 754 standard for computer arithmetic exactly. You don't have to worry that the meaning of an integer is going to change if you move from a Pentium to a PowerPC. In Java everything is guaranteed.

- However the virtual machine itself and some parts of the class library must be written in native code. These are not always as easy or as quick to port as pure Java programs. This is why for example, there's not yet a version of Java 1.2 for the Mac.

Java is Safe

- Java was designed from the ground up to allow for secure execution of code across a network, even when the source of that code was un-trusted and possibly malicious.

- This required the elimination of many features of C and C++. Most notably there are no pointers in Java. Java programs cannot access arbitrary addresses in memory. All memory access is handled behind the scenes by the (presumably) trusted runtime environment. Furthermore Java has strong typing. Variables must be declared, and variables do not change types when you aren't looking. Casts are strictly limited to casts between types that make sense. Thus you can cast an int to a long or a byte to a short but not a long to a Boolean or an int to a String.

- Java implements a robust exception handling mechanism to deal with both expected and unexpected errors. The worst that

- a Java program can do to a host system is bringing down the runtime environment. It cannot bring down the entire system.

- Most importantly Java applets can be executed in an environment that prohibits them from introducing viruses, deleting or modifying files, or otherwise destroying data and crashing the host computer. A Java enabled web browser checks the byte codes of an applet to verify that it doesn't do anything nasty before it will run the applet.

- However the biggest security problem is not hackers. It's not viruses. It's not Visual Basic worms transmitted by Outlook Express. It's not even insiders erasing their hard drives and quitting your company to go to work for your competitors. No, the biggest security issue in computing today is bugs. Regular, ordinary, non-malicious, unintended bugs are responsible for more data loss and lost productivity than all other factors combined. Java, by making it easier to write bug-free code, substantially improves the security of all kinds of programs.

Java is High Performance

- Java byte codes can be compiled on the fly to code that rivals C++ in speed using a "just-in-time compiler." Several companies are also working on native-machine-architecture compilers for Java. These will produce executable code that does not require a separate interpreter, and that is indistinguishable in speed from C++. While you'll never get that last ounce of speed out of a Java program that you might be able to wring from C or FORTRAN, the results will be suitable for all but the most demanding applications.

- As of May, 1999, the fastest VM, IBM's Java 1.1 VM for Windows, is very close to C++ on CPU-intensive operations that don't involve a lot of disk I/O or GUI work; C++ is itself only a few percent slower than C or FORTRAN on CPU intensive operations.

- It is certainly possible to write large programs in Java. The Hot Java web browser, the JBuilder integrated development environment and the java compiler are large programs that are written entirely in Java.

Java is Multi-Threaded

- Java is inherently multi-threaded. A single Java program can have many different processes executing independently and continuously. Three Java applets on the same page can run simultaneously with each getting equal time from the CPU with very little extra effort on the part of the programmer. This makes Java incredibly responsive to user input. It also helps to contribute to

Java's robustness and provides a mechanism whereby the Java environment can ensure that a malicious applet doesn't steal all of the host's CPU cycles.

- Unfortunately multithreading is so tightly integrated with Java, that it makes Java rather difficult to port to architectures like Windows 3.1 or the PowerMac that don't natively support pre-emptive multi-threading.

- There is another cost associated with multi-threading. Multi-threading is to Java what pointer arithmetic is to C; that is, a source of devilishly hard to find bugs. Nonetheless, in simple programs it's possible to leave multi-threading alone and normally be OK.

Java is dynamically linked

- Java does not have an explicit link phase. Java source code is divided into .java files, roughly one per each class in your program. The compiler compiles these into .class files containing byte code. Each

  .java file generally produces exactly one .class file. (There are a few exceptions we'll discuss later, non-public classes and inner classes).

- The compiler searches the current directory and a few other well specified places to find other classes explicitly referenced by name in each source code file. If the file you're compiling depends on other, non-compiled files, then the compiler will try to find them and compile them as well. The Java compiler is quite smart, and can handle circular dependencies as well as methods that are used before they're declared. It also can determine whether a source code file has changed since the last time it was compiled.

- More importantly, classes that were unknown to a program when it was compiled can still be loaded into it at runtime. For example, a web browser can load applets of differing classes that it's never seen before without recompilation.

- Furthermore, Java .class files tend to be quite small, a few kilobytes at most. It is not necessary to link in large runtime libraries to produce an executable. Instead the necessary classes are loaded from the user's local system.

Java is Garbage Collected

- You do not need to explicitly allocate or deallocate memory in Java. Memory is allocated as needed, both on the stack and the heap, and reclaimed by the garbage collector when it is no longer needed. There are no malloc (), free (), or destructor methods. There are constructors and these do allocate memory on the heap, but this is transparent to the programmer.

- Most Java virtual machines use an inefficient, mark and sweep garbage collector. Some more recent virtual machines have improved matters quite a Bit by using generational garbage collection.

JSP (Java Server Pages)

- A Java Server Page (JSP) is a template for a web page that uses Java code to generate an HTML document dynamically. JSPs are run in a server-side component known as a JSP container, which translates them into equivalent Java Servlets.

- For this reason, Servlets and JSP pages are intimately related. What's possible in one is, in large part, also possible in another, although each technology has its individual strengths. Because they are Servlets, JSP pages have all the advantages of Servlets.

Java Servlet

- A Servlet is an object that receives a request and generates a response based on that request. The basic Servlet defines Java objects to represent Servlet requests and responses, as well as objects to reflect the servlet's configuration parameters and execution environments.

- The package javax.servlet.http defines HTTP-specific subclasses of the generic Servlet elements, including session management objects t

- hat track multiple requests and responses between the web server and client.

- Servlets may be packaged in a WAR file as a web application.

- The Servlet API, contained in the Java package hierarchy javax.servlet.http, defines the expected interactions of a web container and a Servlet. A web

container is essentially the component of a web server that interacts with the Servlets.

- The web container is responsible for managing the lifecycle of Servlets, mapping a URL to a particular Servlet and ensuring that the URL requester has the correct access rights.

Framework: HIBERNATE

Hibernate is a pure Java object-relational mapping (ORM) and persistence framework that allows you to map plain old Java objects to relational database tables.The main goal of hibernate is to relieve the developer from the common data persistence related tasks.It maps the objects in the java with the tables in the database very efficiently and also you can get maximum using its data query and retrieval facilities.Mainly by using Hibernate in your projects you can save incredible time and effort.

Advantages of Hibernate

Hibernate is an ORM implementation like many other implementations, it has a numerous number of pros and cons. This section is mainly intended for listing down the most advantages of using Hibernate beside for all these advantages listed for ORM.

- **Database Independent:** Hibernate is independent of the database engine at the backend. List of Hibernate Dialect are provided for connecting whatever database we prefer.
- **JPA Provider:** Java Persistence API (JPA) is a specification. A lot of implementations are available for JPA; Like EclipseLink, OpenJPA and much more. Hibernate is a standard ORM solution and it has a JPA capability. Hence, using of hibernate would help you leverage the all capabilities of ORM and a JPA in a JPA-specific projects.
- **Built-In Connection Pool Implementation:** Hibernate has integrated automatically with the most reliable connection pool implementation (C3P0).
- **Layered Architecture:** Hibernate is a layered architecture, so that we don’t be obligated to use everything provided by Hibernate. We just use those features that you may think they’re lightly enough for the project.

PostgreSQL

- **PostgreSQL**, often simply **Postgres**, is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards-compliance. As a database server, its primary function is to store data securely, supporting best practices, and to allow for retrieval at the request of other software applications. It can handle workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.
- PostgreSQL implements the majority of the core SQL:2011 standard is ACID-compliant and transactional(including most DDL statements) avoiding locking issues using multiversion concurrency control (MVCC), provides immunity to dirty reads and full serializability; handles complex SQL queries using many indexing methods that are not available in other databases; has updateable views and materialized views, triggers, foreign keys; supports functions and stored procedures, and other expandability, and has a large number of extensions written by third parties. In addition to the possibility of working with the major proprietary and open source databases, PostgreSQL supports migration from them, by its extensive standard SQL support and available migration tools. Proprietary extensions in databases such as Oracle can be emulated by built-in and third-party open source compatibility extensions. Recent versions also provide replication of the database itself for availability and scalability.
- PostgreSQL, beginning with version 9.0, includes built-in binary replication, based on shipping the changes (write-ahead logs) to replica nodes asynchronously.
- Version 9.0 also introduced the ability to run read-only queries against these replicated nodes, where earlier versions would only allow that after promoting them to be a new master. This allows splitting read traffic among multiple nodes efficiently. Earlier replication software that allowed similar read scaling normally relied on adding replication triggers to the master, introducing additional load onto it.

Spring - MVC Framework

- The Spring web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

- The **Model** encapsulates the application data and in general they will consist of POJO.

- The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.

- The **Controller** is responsible for processing user requests and building appropriate model and passes it to the view for rendering.

- DispatcherServlet delegates the request to the controllers to execute the functionality specific to it. The **@Controller** annotation indicates that a particular class serves the role of a controller. The **@RequestMapping**annotation is used to map a URL to either an entire class or a particular handler method.

- The **@Controller** annotation defines the class as a Spring MVC controller. Here, the first usage of **@RequestMapping** indicates that all handling methods on this controller are relative to the **/hello** path. Next annotation**@RequestMapping(method = RequestMethod.GET)** is used to declare the*printHello()* method as the controller's default service method to handle HTTP GET request. You can define another method to handle any POST request at the same URL.

Chapter-2 Project Management

2.1 Project Planning

2.1.1 Project Development Approach & Justification

- The **incremental build model** is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished.
- It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements.
- This model combines the elements of the waterfall model with the iterative philosophy of prototyping.
- The product is decomposed into a number of components, each of which are designed and built separately (termed as builds).
- Each component is delivered to the client when it is complete. This allows partial utilization of product and avoids a long development time.
- It also creates a large initial capital outlay with the subsequent long wait avoided. This model of development also helps ease the traumatic effect of introducing completely new system all at once.

Justification

- As incremental model is used for project whose requirements and functionalities are already known and it can be debugged step by step, for working out on all the steps of implementation. So we have chosen incremental model as an approach to develop our project.

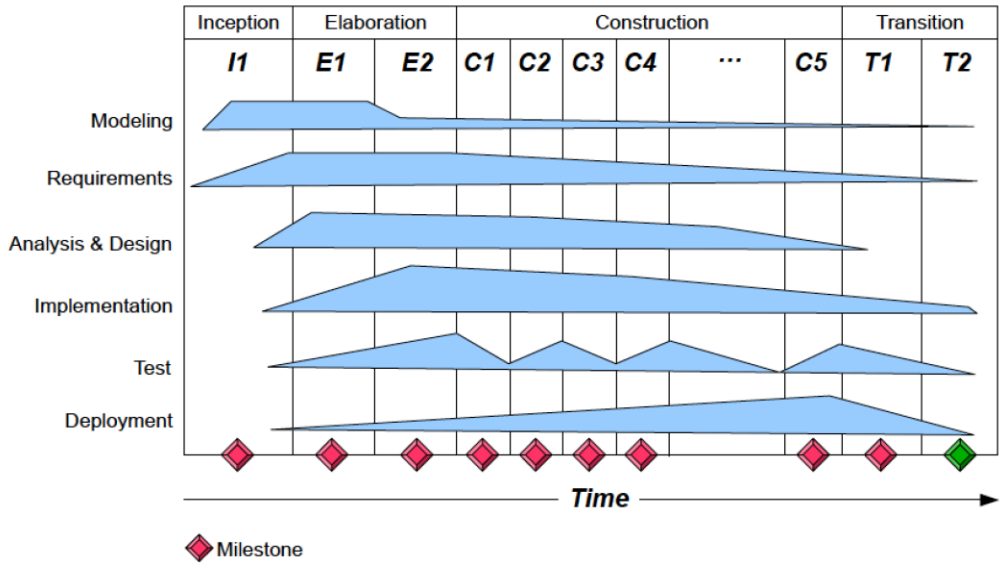


Fig - 2.1 (Phases of Software Development)
"Courtesy of www.google.com"

- The software life-cycle and development process of the Mend Your Thing followed the Rational Unified Process (RUP), which is an industry standard for iterative and incremental software development. The RUP is an adaptable framework in which iterative cycles are combined into four primary time-segments (inception, elaboration, construction, and transition) across a single development goal. Each iteration consists of a series of operational steps that vary in effort according to the current time-segment (see figure). These steps, and their associated activities, are:
    - **Component Modeling** – Identification and refinement of component goals, including defining the business goal of the component (i.e., the benefits and/or high-level achievement of the component); identification of current iteration goals,
    - **Requirements** – Identification of use-case scenarios that define and drive functional and non-functional requirements of the component (use-case scenarios are critical to the initial software development process for identifying system functionality; utilizing use-case scenarios in planning

often results in more precise requirements, which generally translates to more accurate and timely deliverables),

- **Analysis & Design** – Analysis of use-case scenarios and refinement of requirements; generation of component design (using the Unified Modeling Language); identify relevant technology; identify component unit tests to meet requirements,
- **Implementation** – Implementation of unit tests; implementation of component software and evaluation against unit tests,
- **Test** – Review and evaluation of completed component software for iteration or overall component; confirm completeness and correctness against unit tests; sign-off of iteration milestone, and
- **Deployment** – Place software in functioning environment.

2.1.2 Project Plan

- Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross- discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.
- Inception identifies project scope, risks, and requirements (functional and non-functional) at a high level but in enough detail that work can be estimated.
- Elaboration delivers a working architecture that mitigates the top risks and fulfills the non-functional requirements.
- Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements.
- Transition delivers the system into the production operating environment.

2.1.3 Milestones and Deliverables

Management needs information. As software is intangible, this information can only be provided as document that describes the state of the software being developed. Without this information it is impossible to judge progress and cost estimate and the schedules cannot be updated.

Milestones:

- Milestone is an end-point of the software process activity.
- At each milestone there should be formal output such as report.
- Milestone report need not be large document; this should be short report of achievement in software project activity.
- Milestone represents end of the distinct, logical state in the project.

Deliverable:

- Deliverable is a project report that delivers to the customer.
- Deliverables are delivered to the customer at the end of some major project Phase such as specification, design, etc.

2.1.4 Roles and Responsibilities

- We are two persons involved in this project. For any software development there is always human organization structures. Our team structure is Democratic Decentralized, so there is no permanent leader in our team.
- We take decisions regarding our project's problems discussing with each other. It has been found that this project would require a large amount of communication sociability.
- There is no particular role has been defined between us. We have taken responsibilities according to situation occur during the development of this project. System Design, coding, testing and documentation these all task have been completed by us.

2.1.5 Group Dependency

For better development of a project it is better to partition it in modules and each team member should work on different task. Here in our project we have used "***divide & conquer"***. There are **two** members in this group. Each one had taken a particular tasks and performed equal duty in each phase like Analysis, Configuring the cloud services, Database design, and various API study. Division of responsibility is very important while working up on the project.

2.2 Project Schedule

"Software project scheduling is an activity that distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks."

- **Proper scheduling requires:**
  - All tasks appear in network and dependent on some of other.
  - Effort and timing are intelligently allocated to each task.
  - Interdependencies between tasks are properly indicated.
  - Resources are allocated for the work to be done.

Project scheduling is one of the main key aspects of any project. Any project must be schedule before developing it. When project developer works on scheduled project it is more advantageous for him/her to compare to unscheduled project. It gives us timeline for finishing the particular activity.

Scheduling gives us idea about project length, its cost, its normal duration of completion and we can also find out the shortest way to complete the project with less overall cost of project. Project schedule describes dependency between

activities. The estimated time required to reach each milestones and allocation of people to activities.

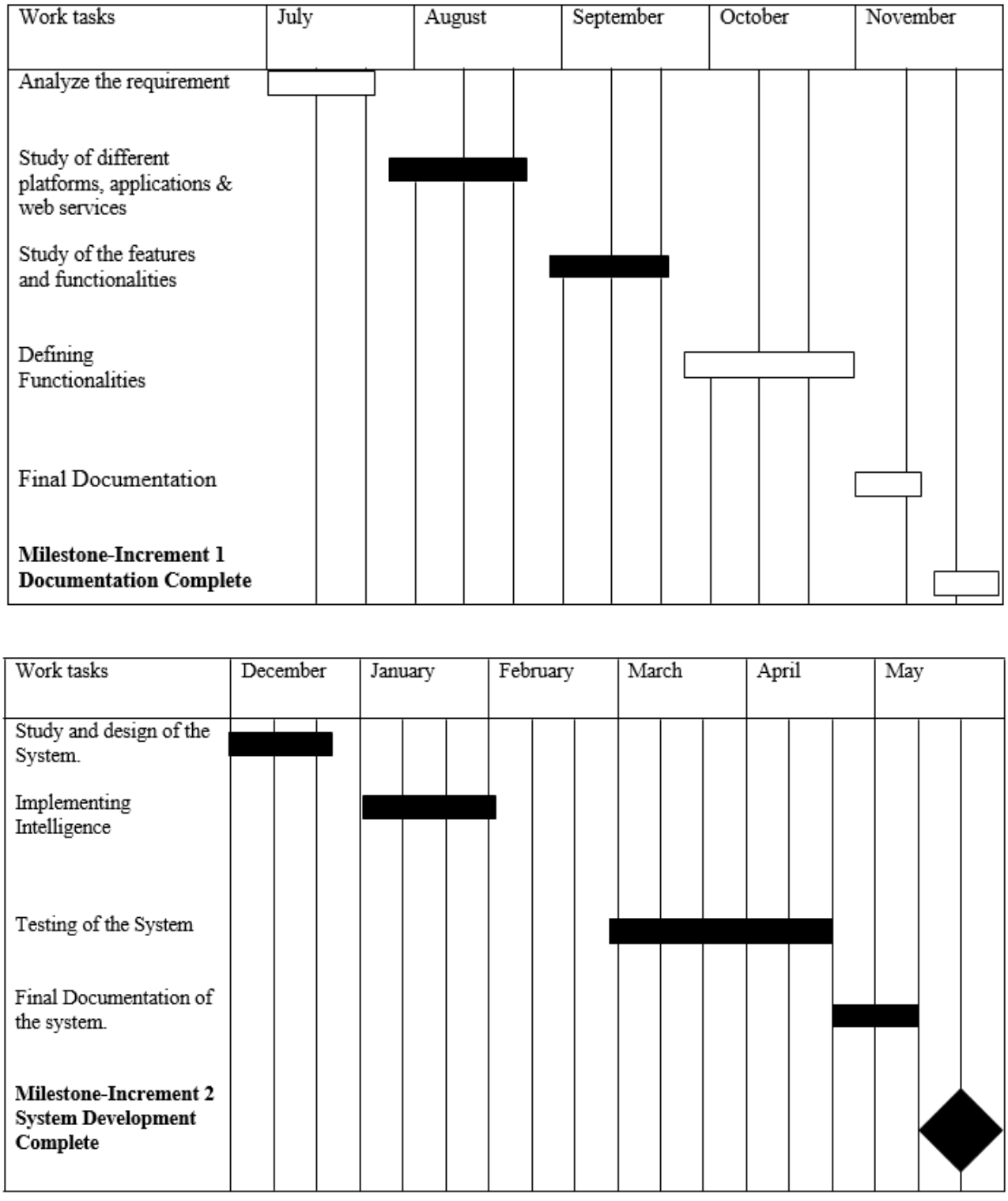


Figure – 2.2 (Pert Chart)

2.3 Risk Management

The Following is the Risk Management study report, which is under taken for the requirement traceability tools.

2.3.1 Risk Identification :

Risk identification is the first stage of risk management. It is concerned with discovering possible risks to the project. Risk identification may be carried out as a team process using a brainstorming approach or may simply be based on experience. To help the process, a checklist of different types of risk may be used. There are at least six types of risk that can arise.

- **Technology risks:** Risks that derive from the software or hardware technologies that are used to develop the system.
- **People risks:** Risks those are associated with the people in the development team.
- **Organizational risks:** Risks that derive from the organizational environment where the software is being developed.
- **Tools risks:** Risks that derive from CASE tools and other support software used to develop the system.
- **Requirements risks:** Risks that derive from changes to the customer requirements and the process of managing the requirements change.
- **Estimation risks:** Risks that derive from the management estimates of the system characteristics and the resources required to build the system.

When we finished the risk identification process, following list of risks could occur which could affect the product, the process and the business.

Risk Type	Possible Risks
Technology	The database cannot process as many transactions per second as expected. Software components which should be reused contain defects which limit their functionality.
People	Required training of users is not done.
Organizational	The organization is reconstructed so that different management is responsible for the project. Organizational financial problems force reductions in the project budget.
Requirements	Changes to requirements which require major design rework are proposed. Customers fail to understand the impact of requirements changes.
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

Table: 2.1 Risk Type – Possible Risks

2.3.2 Risk Analysis :

During risk analysis process, we considered each identified risk and made a judgment about the probability and the seriousness of it. The risk estimates are not generally precise numeric assessments but based around a number of bands.

- The effects of the risk are assessed as catastrophic, serious, and tolerable or insignificant. We have tabulated the results of this analysis process using a table ordered according to the seriousness of the risk. Once the risks were analyzed

and ranked, we assessed which were the most significant. Our judgment was dependent on a combination of the probability of the risk arising and the effects of that risk.

Risk	Probability	Effects
Organizational financial problems force reductions in the project budget.	Low	Catastrophic
Key staffs are ill at critical times in the project.	Moderate	Serious
Software components which should be reused contain defects which limit their functionality.	Moderate	Serious
Changes to requirements which require major design rework are proposed	Moderate	Serious
The organization is reconstructed so that different management are responsible for the project	High	Serious
The database cannot process as many transactions per second as expected	Moderate	Serious
The time required to develop the software is underestimated.	High	Serious
Customers fail to understand the impact of requirements changes.	Moderate	Tolerable
Required training of staff is not available	Moderate	Tolerable
The rate of defect repair is underestimated	Moderate	Tolerable
The size of the software is underestimated	High	Tolerable

Table: 2.2 Risk Vulnerability

2.3.3 Risk planning :

The risk planning process considers each of the key risks that have been identified and identifies strategies to manage the risk. The different strategies fall into three categories:

- **Avoidance strategies:** Following these strategies means that the probability that the risk will arise will be reduced. Risk avoidance strategy is the strategy with defective components.

- **Minimization strategies:** Following these strategies means that the impact of the risk will be reduced.
- **Consistency plans:** Following these strategies means that we were prepared for the worst and had a strategy in place to deal with it. An example of a contingency strategy is the strategy for organizational problems.

Risk	Strategy
Organizational financial problems	Prepare a brief document for senior management showing how the project is making a very important contribution to the goals of the business.
Staff Illness	Reorganize team so that there is no more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.
Requirements changes	Derive traceability information to access requirements change impact; maximize information hiding in the design.
Organizational restructuring	Prepare a Briefing document for senior management showing how the project is making a very important contribution to the goals of the business.

Table: 2.3 Risks – Strategy

2.4 Estimation

There are two types of models that have been used to estimate cost: cost models and constraint models. Cost models provide direct estimates of effort. These models typically have a primary cost factor such as size and the number of secondary adjustment factors or cost drivers. Cost drivers are characteristics of the project, process, products, or resources that influence effort. Cost drivers are used to adjust the preliminary estimate provided by the primary cost factor.

COCOMO (Constructive Cost estimation Model) is a Heuristic Technique which can be used for our project.

Software project can be classified in one of the three categories based on the project complexity.

- **ORGANIC**
- **SEMI DETACHED**
- **EMBEDDED**

Software project	*A*	*b*	*c*	*d*
Organic	24	2	25	1.3
Semi-detached	25	2.1	25	1.31
Embedded	26	2.12	25	1.32

Table: 2.4 Estimation Cost

Since we are a small team and working in coordination with our seniors on a simple project without rigid requirements, our project comes under Organic type.

Effort = a*(KLOC) ^b PM

Tdev = c*(effort) ^d

Chapter-3: System Requirement Study

3.1 User Characteristics

The users of the online repairing application, based on their roles, are customers (users) and the administrator (owner). These users are identified based on their experience and technical expertise.

1. Admin: The administrator is the owner of this online repairing application. One must have a basic understanding of computers and the internet as well as prior knowledge for operating the eclipse and Java programming languages. The administrator is responsible for maintaining all the training documents required for the system. The administrator View the history of the customers who repair the items.

2. Users: The users of this online repairing application are all customers and service provider who would shop to test the application. These users are anyone with repairing experience and the know-how to browse through a repairing application. They must have basic understandings about computers and the internet. The users should be able to perform the following functions using this system:

- View, browse, and select a category on the home page.
- View, add, and update items in the cart.
- Delete items from the cart.
- Sign-on/login using a username and password.
- Place the Deal.

3.2 User Constraints:

The system will work with Windows 7 and up, Linux and other devices / operating systems using web browser.

The system will only work with current releases of web browsers (for web portal access).

Chapter 4 : System analysis

4.1 Study of current system

- There are many existing system for repairing like home joy etc.
- They provided service related to the repairing to the customer.
- In that system when customers request for services they provide them service provider related that services.
- By this way that sites work.

4.2 Problem in current system

- There are many application related repairing but they provide individual services like car repairing, mobile repairing etc.
- But all are present in single module is not anywhere.
- Some applications provide more then one services but they chose service provider according to them.
- So Customer must paid money according to them.

4.3 Requirement of new System

4.3.1 Requirement Engineering Process:

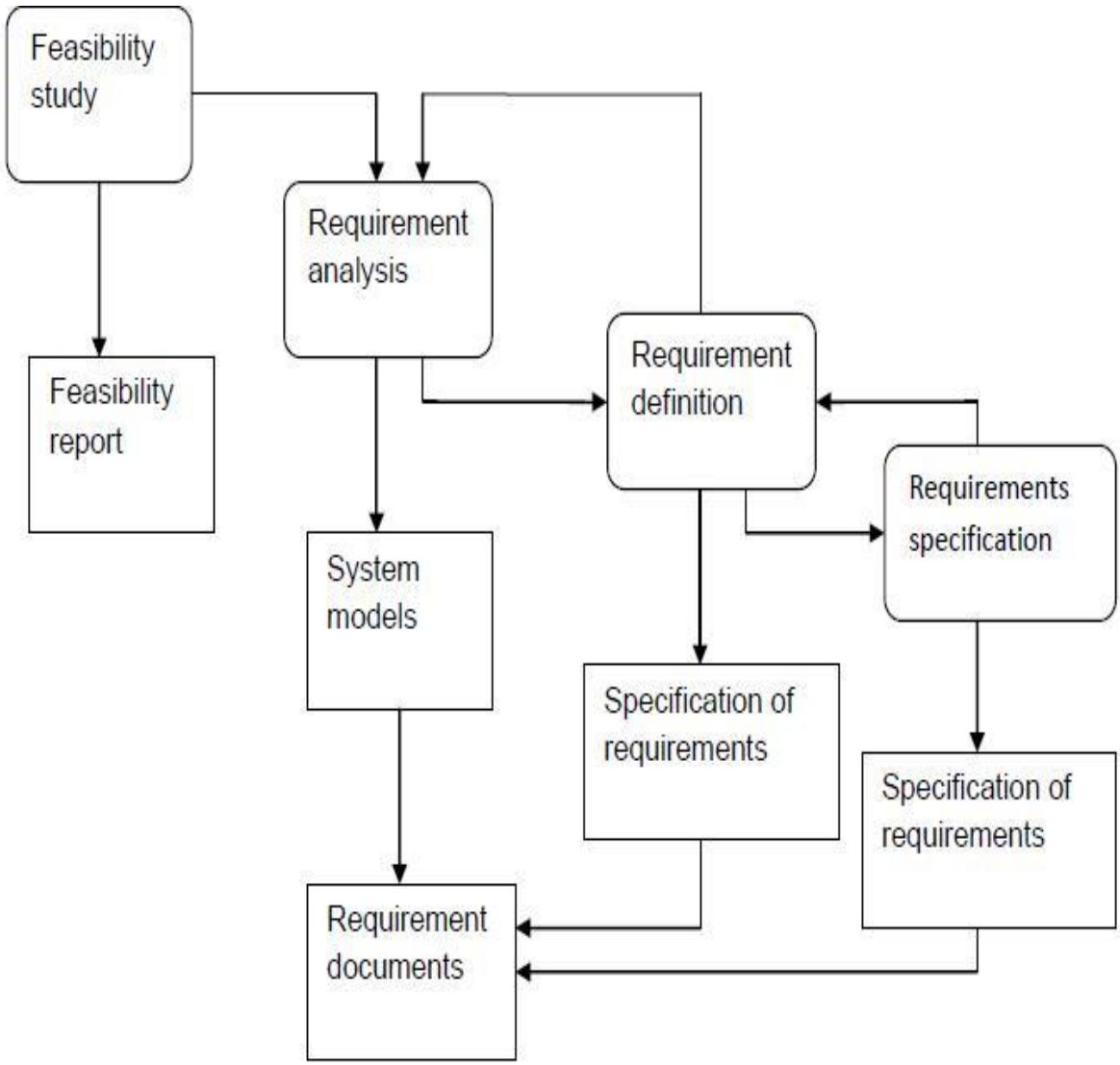


Fig 4.1 Requirement Engg process

4.3.2 Requirement Analysis Process:

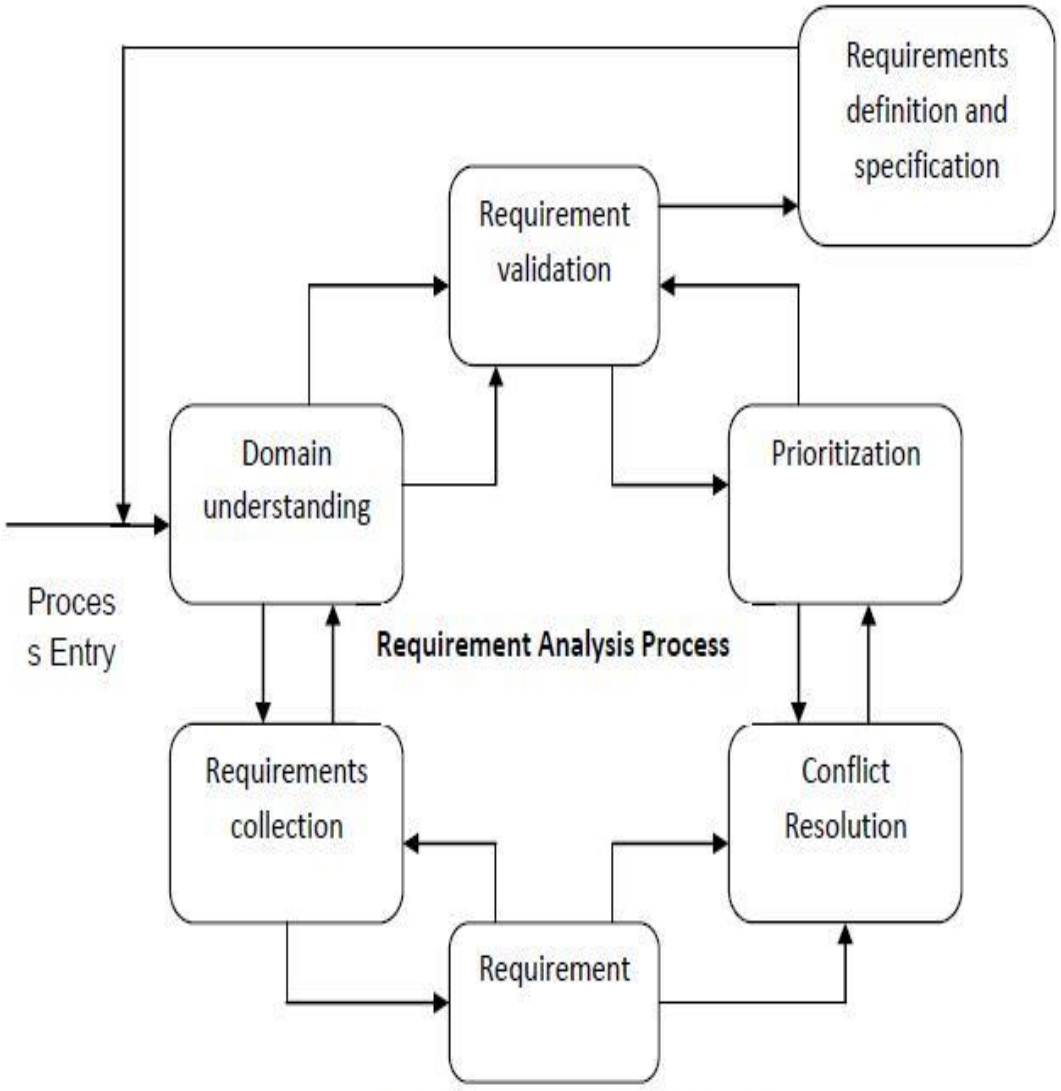


Fig 4.2 Requirement Analysis Process

4.4 Process Model

- The basic idea behind iterative enhancement is to develop a software system incrementally, allowing the developer to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the system. Learning comes from both the development and use of the system, where possible. Key steps in the process were to start with a simple implementation of a subset of the software requirements and iteratively enhance

the evolving sequence of versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added.

- The Procedure itself consists of the Initialization step, the Iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily

- The iteration involves the redesign and implementation of a task from project control list, and the analysis of the current version of the system. The goal for the design and implementation of any iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project control list. The code can, in some cases, represent the major source of documentation of the system. The project control list is modified in light of the analysis result'.

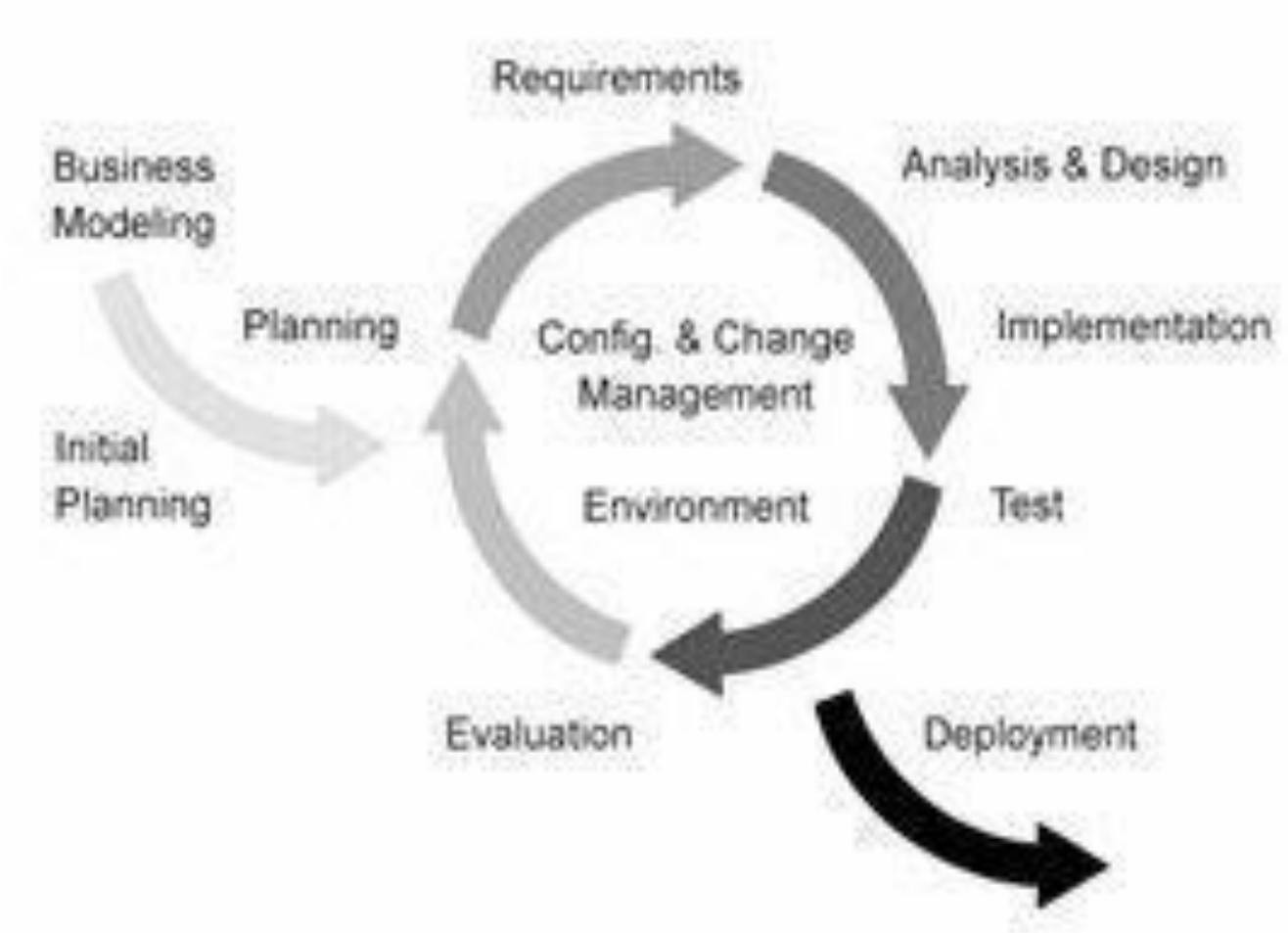


Fig 4.3 Incremental process

4.5 Feasibility Study

Objective of Feasibility Study :

- An important outcome of the preliminary investigation is the determination that the system requested is feasible. The feasibility study is carried out to examine the likelihood that the system will be useful to the organization.
- There are four aspects in the feasibility study namely.
  - Operational Feasibility
  - Technical Feasibility
  - Economic Feasibility
  - Schedule Feasibility

4.5.1 Operational Feasibility

The main purpose of checking Operational Feasibility is to find out whether the system will be functional after its development and installation or not.

4.5.2 Technical Feasibility

The main purpose of checking Technical Feasibility is to examine whether the current technology is sufficient for the development of the system.

4.5.3 Economical Feasibility

The main purpose of checking Economical Feasibility is to examine whether the financial investment in the system will meet the organization's requirements or not.

4.5.4 Schedule Feasibility

This type of the feasibility includes a measure of how reasonable the projected with respect to time aspect. When developing software it is difficult to measure such things as software complexity, quality and to estimate the amount of effort it will take to complete the project.

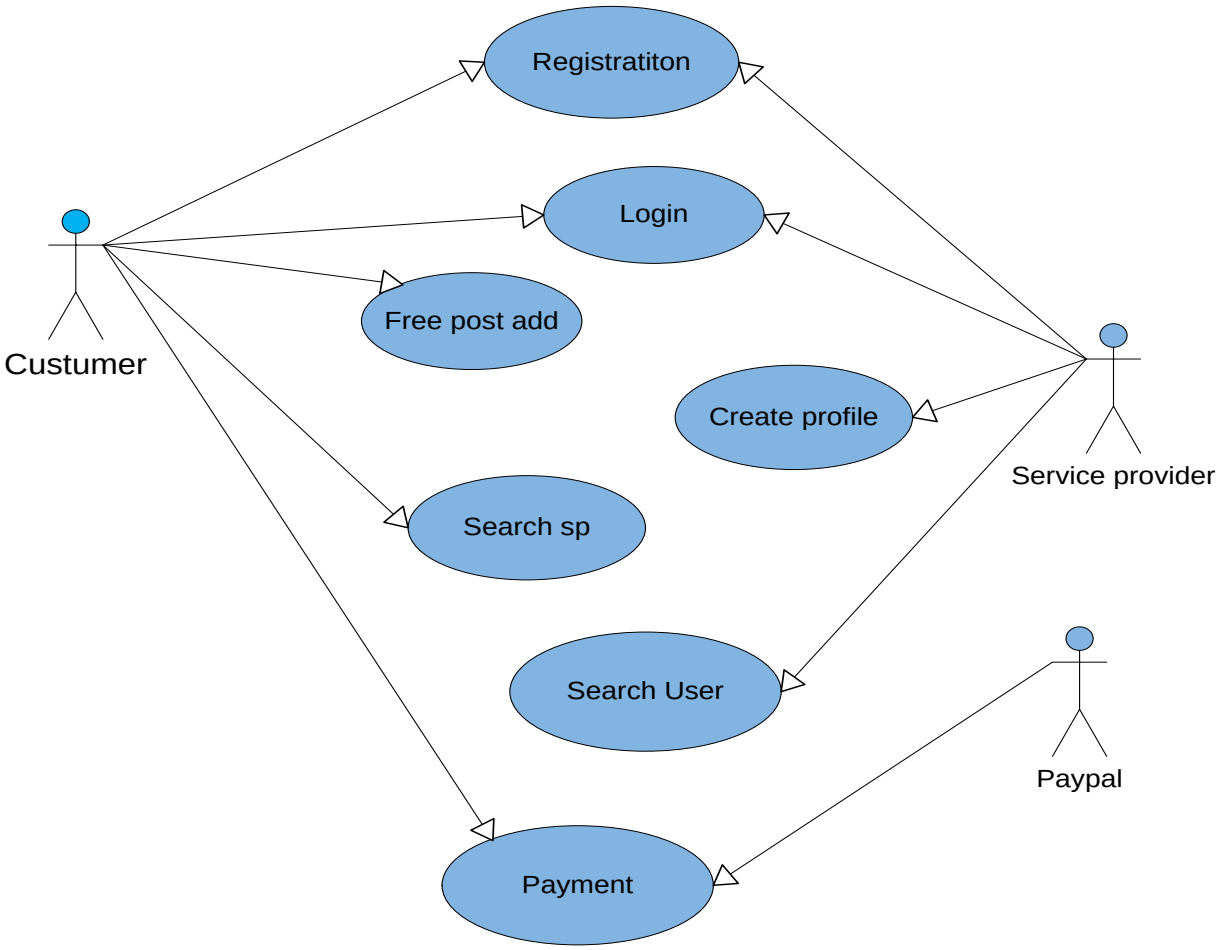
4.6 Feature of New System

- We provide online platform to customer & service provider for repairing.
- Customer can find more then one service provide for one item at minimum cost.
- Service provider gets more customers then at shop.
- Customer give review and rating so other viewer can find best service.
- Here, we can also give the security to the customer from service provider by upload their legal documents and certificate of them specialization.
- Customer and service provider has also there dealing
- Detail online so no possibility of online fraud.
- Data is secured and data is stored in Database.

Chapter-5 System Design

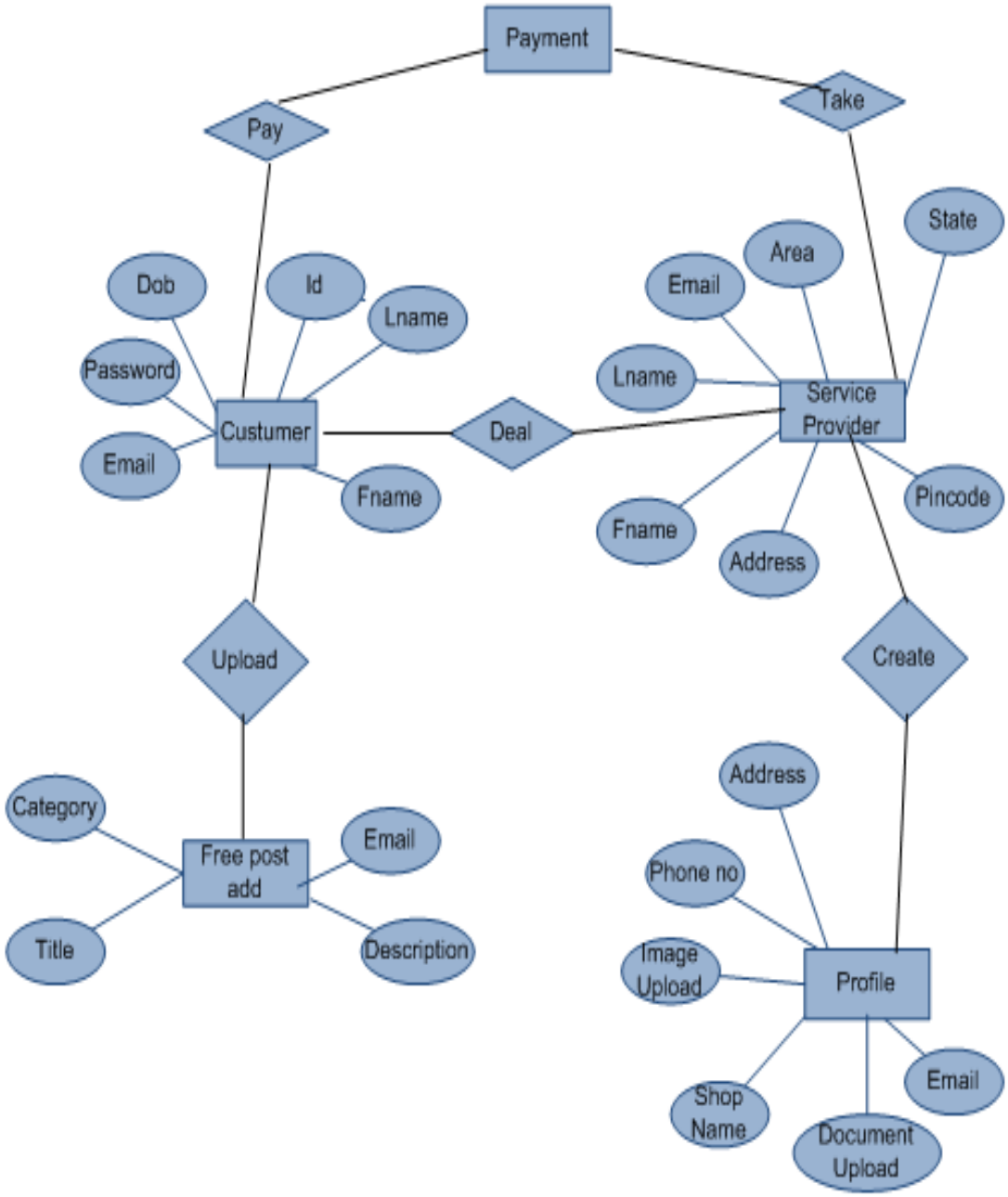
5.1 UseCase Diagram:

In software and system engineering, a use case is a list of action or event steps, typically define the interaction between a role and system, to achieve a goal. The actor can be a human, an external system, or time. In systems engineering, use case are Used at a higher level than within software engineering, often representing missions or stakeholders goal.



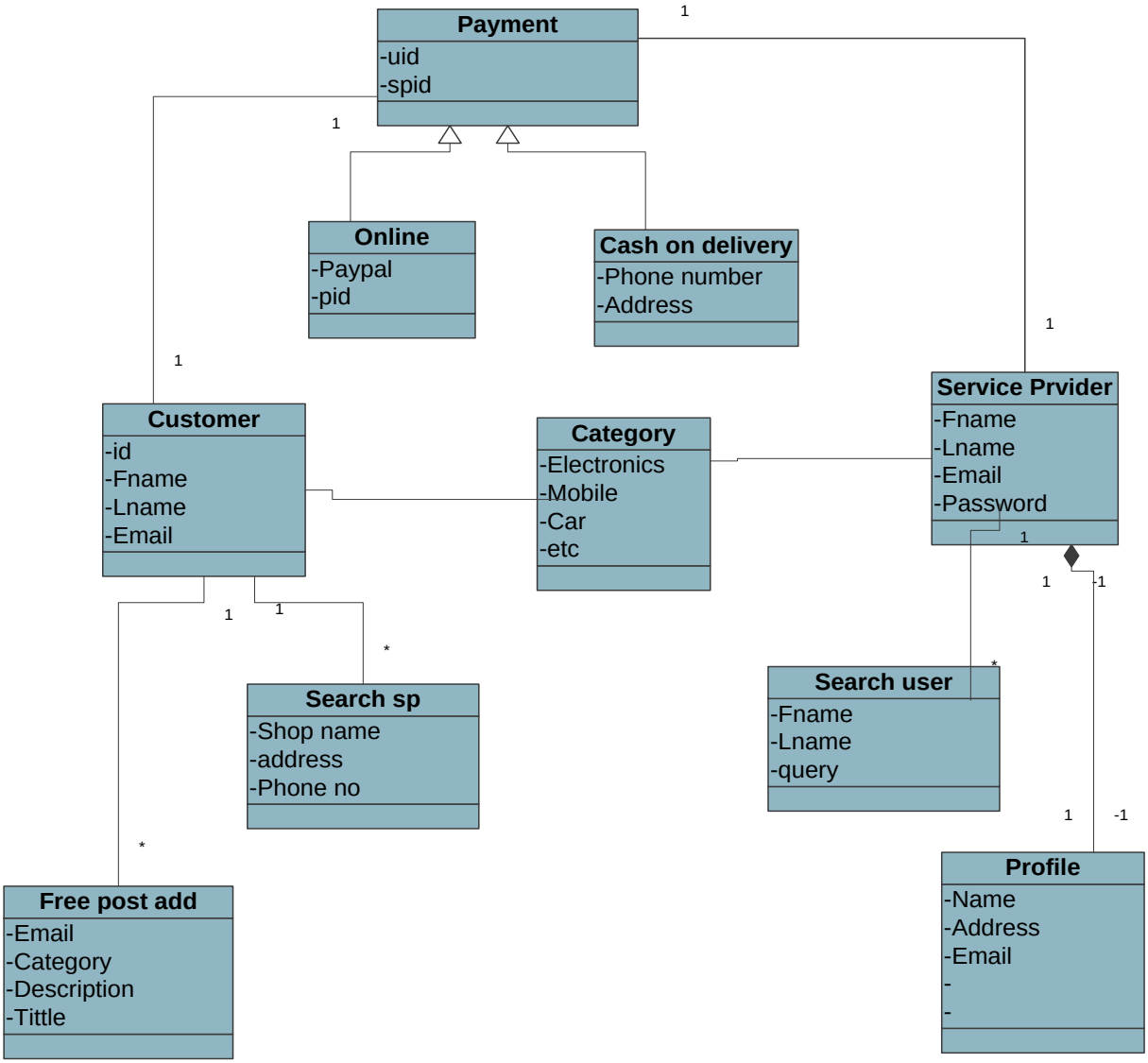
5.2 E-R Diagram:

An entity-relationship model is a systematic way of describing and defining a business process. The process is modeled as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one build.



5.3 Class Diagram:

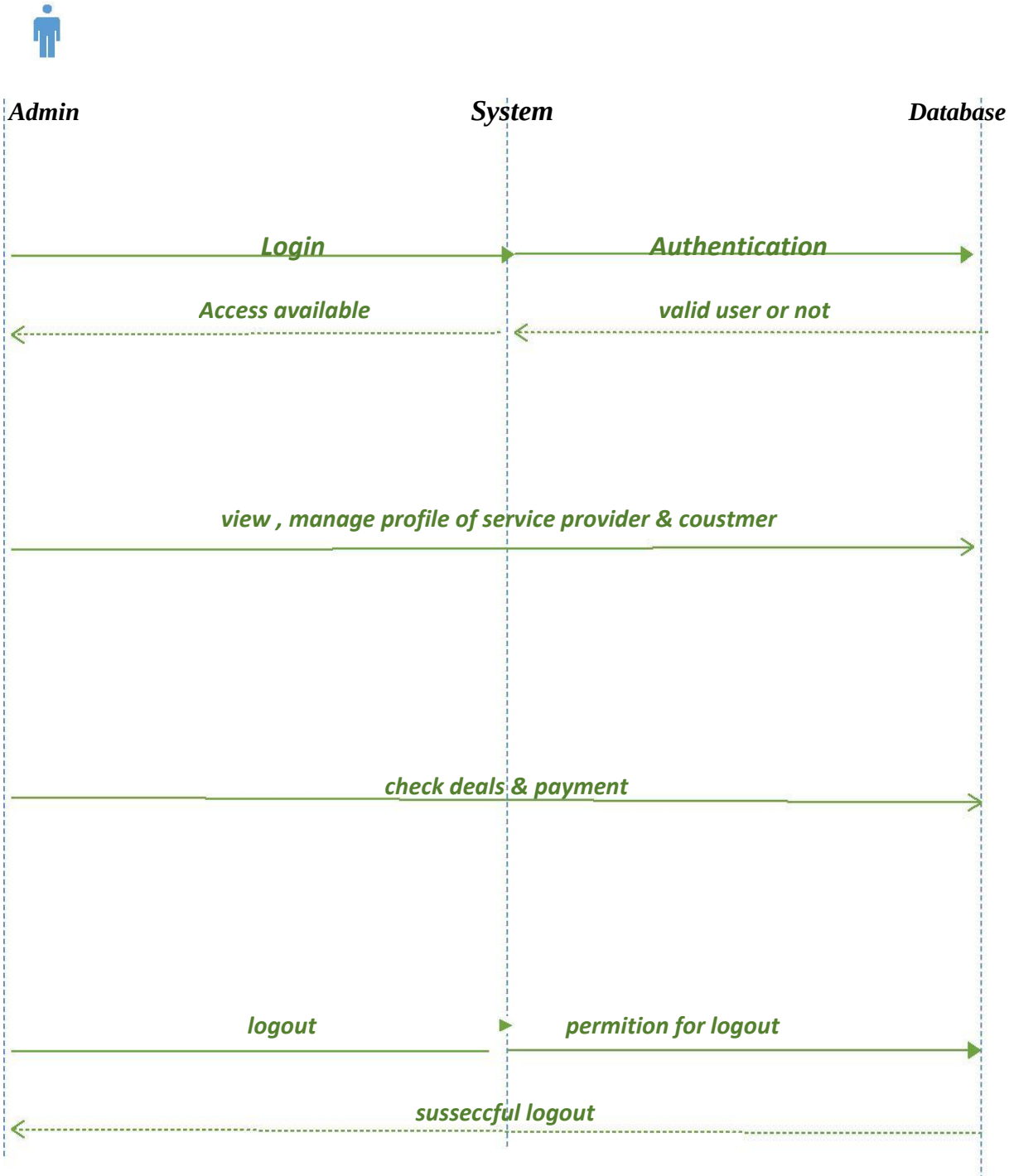
Class diagram is main building block of object oriented modeling. it is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the model into programming code. class diagram can also be used for data modeling.



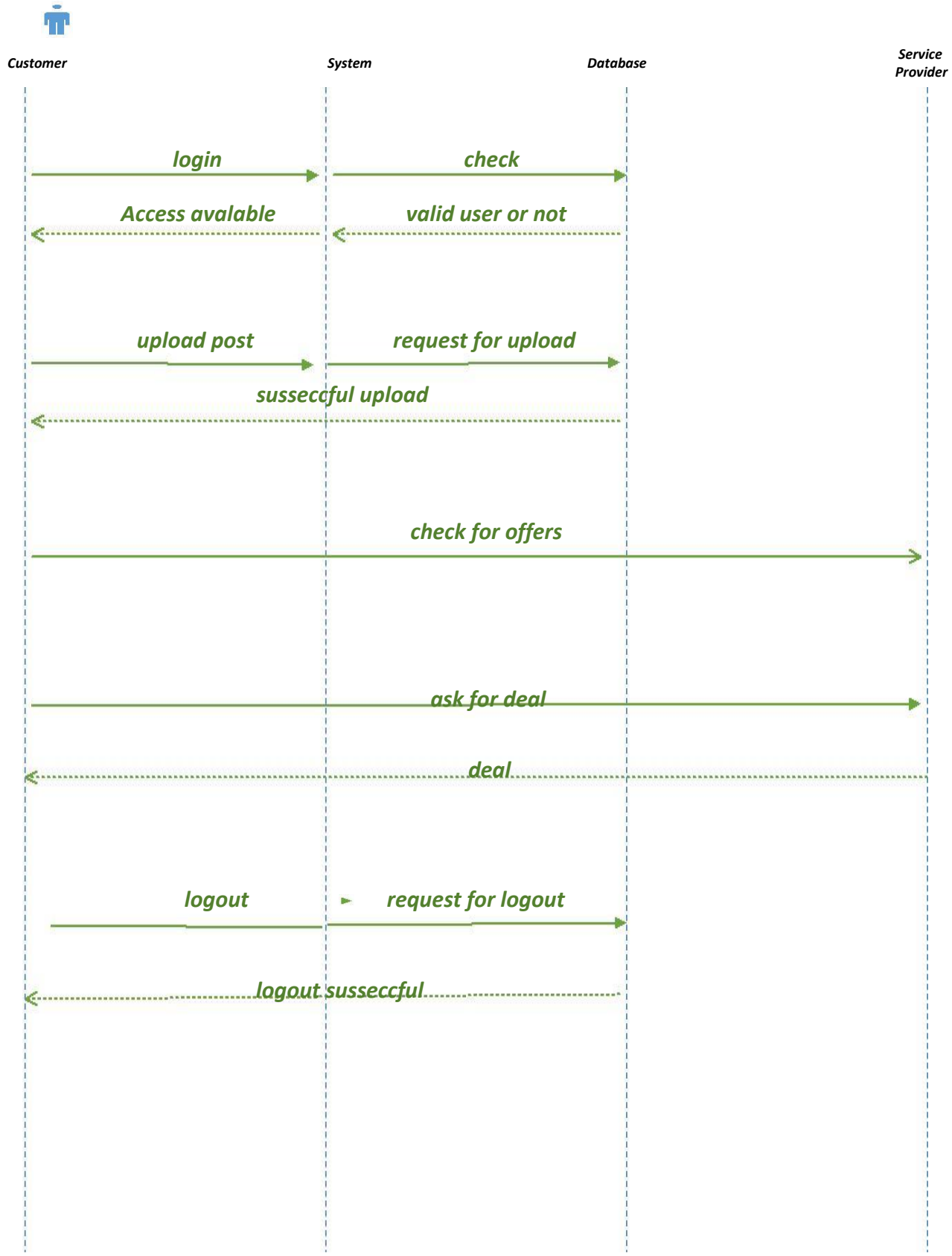
5.4 Sequence diagram

5.4.1 Sequence diagram for admin:

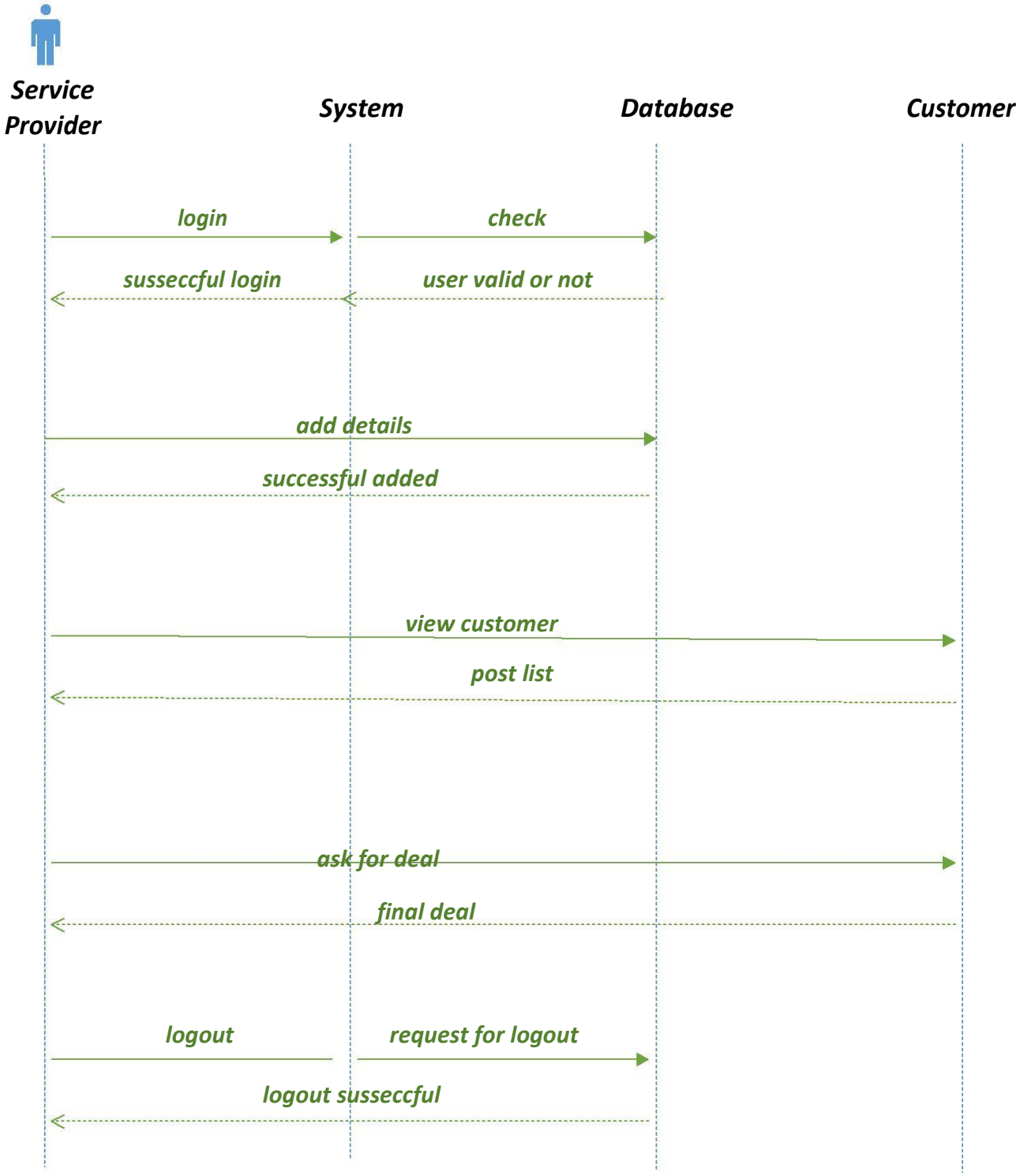
A sequence diagram is an interaction diagram that shows how processes operate with one another and what in order. It is a construct of a message sequence chart. A sequence diagram shows object interaction arranged in a time sequence.



5.4.2 Sequence diagram for customer:



5.4.3 Sequence diagram for service provider:



5.5 Data Dictionary

Admin:

Field	Data type	Size	Constraints
Id	Long	10	Primary Key
Email_id	Varchar	20	Not Null
Password	Varchar	15	Not Null

Table 5.1: Admin table

Service provider:

Field	Data type	Size	Constraints
Id	Long	10	Primary Key
Area	Varchar	20	Not Null
Citizenship	Varchar	20	Not Null
City	Varchar	20	Not Null
Fname	Varchar	20	Not Null
Lname	Varchar	20	Not Null
Email_id	Varchar	20	Not Null
Passward	Varchar	15	Not Null
State	Varchar	80	Not Null
Pincode	INT	6	Not Null
Phone	INT	6	Not Null

Table 5.2: Service provider table

Customer:

Field	Data type	Size	Constraints
Id	Long	10	Primary Key
Dob	Long	10	Not Null
Citizenship	Varchar	20	Not Null
Fname	Varchar	20	Not Null
Lname	Varchar	20	Not Null
Email_id	Varchar	20	Not Null
Passward	Varchar	15	Not Null
Gender	Varchar	15	Not Null
Phone	INT	6	Not Null

Table 5.3: Customer table

Profiles:

Field	Data type	Size	Constraints
Id	Long	10	Primary Key
Phone	Long	10	Not Null
Adreess	Varchar	20	Not Null
Category	Varchar	20	Not Null
Shop_name	Varchar	20	Not Null
Image_id	Long	10	Foreign Key
Transcript_id	Long	10	Foreign Key
User_id	Long	10	Foreign Key

Table 5.4: Profiles table

Query:

Field	Data type	Size	Constraints
id	Long	10	Primary Key
Category	Varchar	30	Not Null
Discription	Varchar	30	Not Null
Email_id	Long	10	Foreign Key
Tittle	Varchar	30	Not Null
Customer_id	Long	10	Foreign Key

Table 5.5: Query table

InterestProvider:

Field	Data type	Size	Constraints
Id	Long	10	Primary Key
Spid	Long	10	Foreign Key
Qid	Long	10	Foreign Key

Table 5.6: InterestProvider table

Files:

Field	Data type	Size	Constraints
Id	Long	10	Primary Key
Date	Long	10	Not Null
Name	Varchar	20	Not Null
Profile_id	Long	10	Foreign Key

Table 5.7: Files table

Payment:

Field	Data type	Size	Constraints
id	Long	10	Primary Key
Spid	Long	10	Foreign Key
Customer_id	Long	10	Foreign Key
Paypal AC_no.	Long	30	Not Null
Email_id	Varchar	20	Not Null

Table 5.8: Payment table

Chapter-6 Implementation Planning

6.1 Implementation Planning

The IDE(s) used for implementation is

- Microsoft Visual Studio 2013, Eclipse version LUNA, Web Server Apache Tomcat 7.0, Java EE development platform, Hibernet Framework, Spring MVC Structure.

- Postgres SQL

Business Model Canvas.

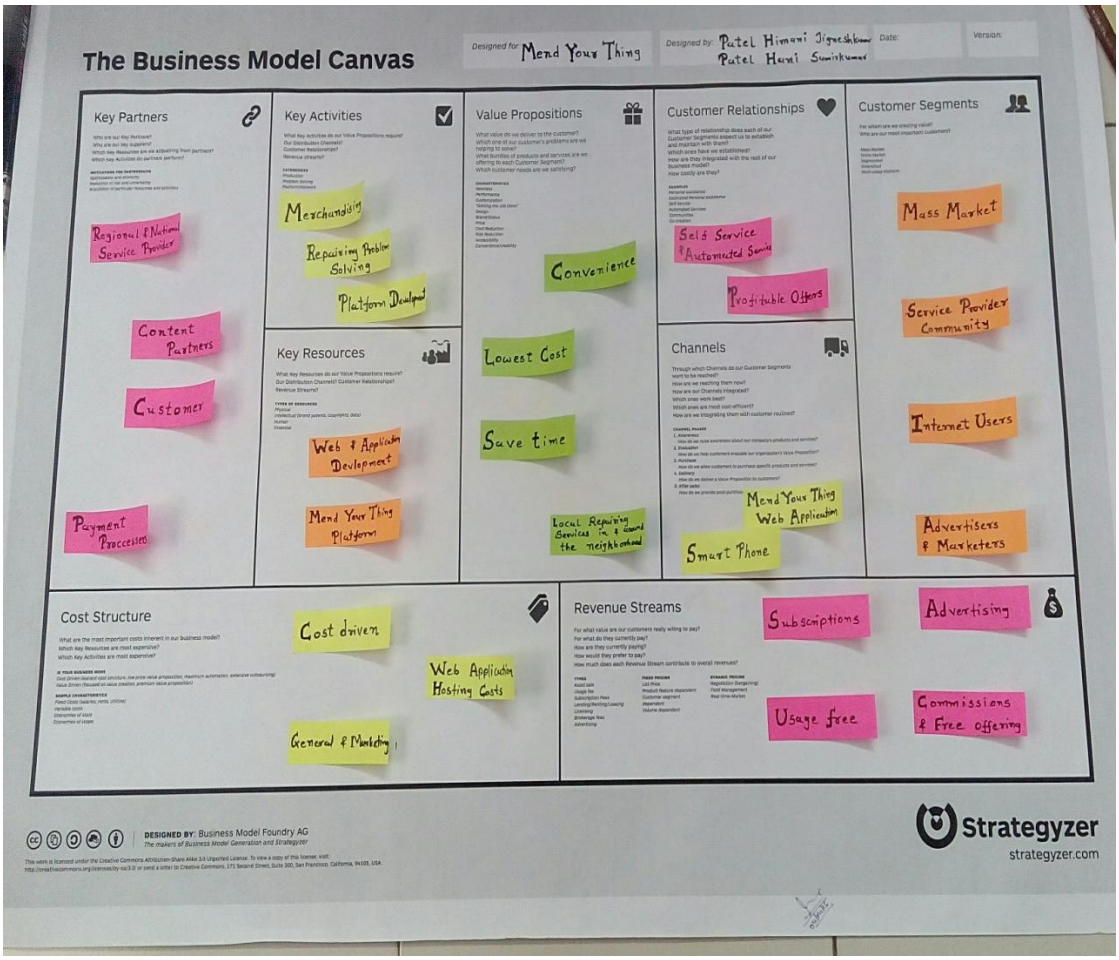


Fig 6.1: Business Model Canvas.

6.2 Program/Modules Specification

The main modules of the system are:

1.Admin:

The administrator is the owner of this online repairing application. One must have a basic understanding of computers and the internet as well as prior knowledge for operating the eclipse and Java programming languages. The administrator is responsible for maintaining all the training documents required for the system. The administrator View the history of the customers who repair the items.

2.Users:

The users of this online repairing application are all customers and service provider who would shop to test the application. These users are anyone with repairing experience and the know-how to browse through a repairing application. They must have basic understandings about computers and the internet. The users should be able to perform the following functions using this system:

- View, browse, and select a category on the home page.
- View, add, and update items in the cart.
- Delete items from the cart.
- Sign-on/login using a username and password.
- Place the Deal.

6.3 Coding Standards

UserController File:

```
package mendyourthing.controllers;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;
import javax.servlet.ServletContext;
import javax.servlet.http.HttpSession;
```

```
import javax.validation.Valid;
import mendyourthing.models.Admin;
import mendyourthing.models.InterestProvider;
import mendyourthing.models.Profile;
import mendyourthing.models.Provider;
import mendyourthing.models.Query;
import mendyourthing.models.User;
import mendyourthing.models.dao.AdminDao;
import mendyourthing.models.dao.InterestProviderDao;
import mendyourthing.models.dao.ProfileDao;
import mendyourthing.models.dao.ProviderDao;
import mendyourthing.models.dao.QueryDao;
import mendyourthing.models.dao.UserDao;
import mendyourthing.util.FileIO;
import mendyourthing.util.MailMail;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.propertyeditors.CustomDateEditor;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.validation.ObjectError;
import org.springframework.web.bind.WebDataBinder;
import org.springframework.web.bind.annotation.InitBinder;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.support.SessionStatus;
import org.springframework.web.multipart.MultipartFile;


@Controller
public class UserController {

	@Autowired
	private AdminDao adminDao;

	@Autowired
	private UserDao userDao;
```

```
@Autowired
private ProviderDao providerDao;

@Autowired
private QueryDao queryDao;

@Autowired
private ProfileDao profileDao;

@Autowired
private FileIO fileIO;

@Autowired
private ServletContext servletContext;

@Autowired
private HttpSession session;

@Autowired
private InterestProviderDao interestProviderDao;

@Autowired
private MailMail mailMail;

@InitBinder
public void initBinder(WebDataBinder binder) {
	binder.registerCustomEditor(Date.class, new CustomDateEditor(
			new SimpleDateFormat("M/d/yyyy"), true));
}


@RequestMapping(value = "/admin.html", method = RequestMethod.GET)
public String admin(ModelMap models) {
	models.put("admin", new Admin());
	return "admin";

}

@RequestMapping(value = "/admin-login.html", method = RequestMethod.GET)
public String adminlogin(ModelMap models) {
	models.put("admin", new Admin());
	return "admin-login";
}

@RequestMapping(value = "/admin-register.html", method = RequestMethod.GET)
public String adminregister(ModelMap models) {
	models.put("admin", new Admin());
	return "admin-register";
}

@RequestMapping(value = "/admin-login.html", method = RequestMethod.POST)
public String adminlogin(@RequestParam String email,
		@RequestParam String password, SessionStatus sessionStatus,
		ModelMap models) {

	if (email != null) {
		email = email.toLowerCase();
	}
```

```
		Admin adminObj = adminDao.getAdminByEmail(email);
		if (adminObj != null) {
			if (adminObj.getPassword().equals(password)) {

				session.setAttribute("admin", adminObj);
				session.setAttribute("type", "admin");
				return "redirect:/admin.html";

			} else {
				return "redirect:/admin-login.html";
			}
		} else {
			return "redirect:/admin-login.html";
		}
	}


	@RequestMapping(value = "/admin-register.html", method = RequestMethod.POST)
	public String adminregister(@Valid @ModelAttribute Admin admin,
			BindingResult bindingResult, SessionStatus sessionStatus) {
		System.out.println(bindingResult);

		Admin adminObj = adminDao.getAdminByEmail(admin.getEmail());
		if (adminObj != null) {
			ObjectError error = new ObjectError("email",
					"User already exists !");
			bindingResult.addError(error);
		}
		if (bindingResult.hasErrors()){
			return "admin-register";}
		admin.setEmail(admin.getEmail().toLowerCase());
		admin = adminDao.create(admin);


		return "redirect:/admin-login.html";
	}


	@RequestMapping(value = "/register.html", method = RequestMethod.GET)
	public String register(ModelMap models) {
		models.put("user", new User());
		return "register";
	}

	@RequestMapping(value = "/provider_register.html", method = RequestMethod.GET)
	public String provider_register(ModelMap models) {
		models.put("provider", new Provider());
		return "provider_register";
	}

	@RequestMapping(value = "/register.html", method = RequestMethod.POST)
	public String register(@Valid @ModelAttribute User user,
			BindingResult bindingResult, SessionStatus sessionStatus) {
		System.out.println(bindingResult);

		User userObj = userDao.getUserByEmail(user.getEmail());
		if (userObj != null) {
			ObjectError error = new ObjectError("email",
					"User already exists !");
```

```
			bindingResult.addError(error);
		}
		if (bindingResult.hasErrors()){
			return "register";}
		user.setEmail(user.getEmail().toLowerCase());
		user = userDao.create(user);

		mailMail.sendMail("myt61135@gmail.com", user.getEmail(),
				"Registration Succeessful",
				"Hi ! You account has been successfully created !");

		return "redirect:/login.html";
	}

	@RequestMapping(value = "/provider_register.html", method = RequestMethod.POST)
	public String provider_register(@Valid @ModelAttribute Provider provider,
			BindingResult bindingResult, SessionStatus sessionStatus) {
		Provider providerObj = providerDao.getProviderByEmail(provider
				.getEmail());
		if (providerObj != null) {
			ObjectError error = new ObjectError("email",
					"Provider already exists !");
			bindingResult.addError(error);
		}
		if (bindingResult.hasErrors())
			return "provider_register";
		provider.setEmail(provider.getEmail().toLowerCase());
		provider = providerDao.create(provider);

		mailMail.sendMail("myt61135@gmail.com", provider.getEmail(),
				"Registration Succeessful",
				"Hi ! You account has ben successfully created !");

		return "redirect:/provider_login.html";
	}

	@RequestMapping(value = "/login.html", method = RequestMethod.GET)
	public String login(ModelMap models) {
		models.put("user", new User());
		return "login";
	}

	@RequestMapping(value = "/login.html", method = RequestMethod.POST)
	public String login(@RequestParam String email,
			@RequestParam String password, SessionStatus sessionStatus,
			ModelMap models) {

		if (email != null) {
			email = email.toLowerCase();
		}

		User userObj = userDao.getUserByEmail(email);
		if (userObj != null) {
			if (userObj.getPassword().equals(password)) {

				session.setAttribute("user", userObj);
				session.setAttribute("type", "user");
				return "redirect:/user.html";
```

```
			} else {
				return "redirect:/login.html";
			}
		} else {
			return "redirect:/login.html";
		}
	}

	@RequestMapping(value = "/provider_login.html", method = RequestMethod.GET)
	public String provider_login(ModelMap models) {
		models.put("Provider", new Provider());
		return "provider_login";
	}

	@RequestMapping(value = "/provider_login.html", method = RequestMethod.POST)
	public String provider_login(@RequestParam String email,
			@RequestParam String password, SessionStatus sessionStatus,
			ModelMap models) {

		if (email != null) {
			email = email.toLowerCase();
		}

		Provider providerObj = providerDao.getProviderByEmail(email);
		if (providerObj != null) {
			if (providerObj.getPassword().equals(password)) {
				session.setAttribute("Provider", providerObj);
				session.setAttribute("type", "Provider");
				return "redirect:/provider.html";

			} else {
				return "redirect:/provider_login.html";
			}
		} else {
			return "redirect:/provider_login.html";
		}
	}

	@RequestMapping(value = "/query.html", method = RequestMethod.GET)
	public String query(ModelMap models) {
		models.put("query", new Query());
		return "query";
	}

	@RequestMapping(value = "/query.html", method = RequestMethod.POST)
	public String query(@ModelAttribute Query query,
			SessionStatus sessionStatus,
		return "redirect:/";
	}

	@RequestMapping(value = "/search.html", method = RequestMethod.GET)
	public String search(ModelMap models) {
		// models.put("query", new Query());
		return "search";
	}

	@RequestMapping(value = "/search.html", method = RequestMethod.POST)
	public String search(@RequestParam String category,
			SessionStatus sessionStatus, ModelMap models) {
```

```
		System.out.println(category);

		List<Query> queries = queryDao.getQueryByCategory(category);
		models.put("queries", queries);

		return "query_list";

	}

	@RequestMapping(value = "/interest.html", method = RequestMethod.GET)
	public String interest(@RequestParam long qid, @RequestParam long spid,
			SessionStatus sessionStatus, ModelMap models) {
		System.out.println(qid);

		Query queryObj = queryDao.findById(qid);
		Provider providerObj = providerDao.findById(spid);

		InterestProvider InterestProviderObj = new InterestProvider();
		InterestProviderObj.setQuery(queryObj);
		InterestProviderObj.setProvider(providerObj);

		interestProviderDao.create(InterestProviderObj);

		return "sp_interest";

	}

	@RequestMapping(value = "/search_user.html", method = RequestMethod.GET)
	public String search_user(ModelMap models) {
		// models.put("query", new Query());
		return "search_user";
	}

	@RequestMapping(value = "/search_user.html", method = RequestMethod.POST)
	public String search_user(@RequestParam String category,
			SessionStatus sessionStatus, ModelMap models) {
		System.out.println(category);

		List<Profile> profiles = profileDao.getProfileByCategory(category);

		if (profiles != null) {
			for (Profile pro : profiles) {
				System.out.println(pro);
			}
		}

		models.put("profiles", profiles);

		return "profile_list";

	}

	@RequestMapping(value = "/profile.html", method = RequestMethod.GET)
	public String profile(ModelMap models) {
		models.put("profile", new Profile());
		return "profile";
	}

	@RequestMapping(value = "/profile.html", method = RequestMethod.POST)
```

```
	public String profile(
			@ModelAttribute Profile profile,
			BindingResult bindingResult,
			@RequestParam(value = "transcriptFile", required = false)
MultipartFile transcriptFile,
			@RequestParam(value = "imageFile", required = false) MultipartFile
imageFile,
			ModelMap models) {

		Provider provider = (Provider) session.getAttribute("provider");
		String fileDir = servletContext.getRealPath("/profiles");
	  Profile profileObj = profileDao.create(profile);


		if (transcriptFile != null && !transcriptFile.isEmpty()){
			System.out.println("before saving file");
			profileObj.setTranscript(fileIO.save(fileDir, transcriptFile,
profileObj));
			System.out.println("after
saving::"+profileObj.getTranscript().getId());
		}

		fileDir = servletContext.getRealPath("/profiles");
		if (imageFile != null && !imageFile.isEmpty()){
			System.out.println("before saving file");
			profileObj.setImage(fileIO.save(fileDir, imageFile, profileObj));
			System.out.println("after saving::"+profileObj.getImage().getId());
		}


		profileDao.update(profileObj);

		return "redirect:/";


	}

	@RequestMapping(value = "/payment.html", method = RequestMethod.GET)
	public String payment(ModelMap models) {
		return "payment";
	}

	@RequestMapping(value = "/paymentcancle.html", method = RequestMethod.GET)
	public String paymentcancle(ModelMap models) {
		return "paymentcancle";
	}

	@RequestMapping(value = "/savepayment.html", method = RequestMethod.GET)
	public String savepayment(ModelMap models) {
		return "savepayment";
	}



	@RequestMapping(value = "/about.html", method = RequestMethod.GET)
	public String about(ModelMap models) {
		return "about";
	}
```

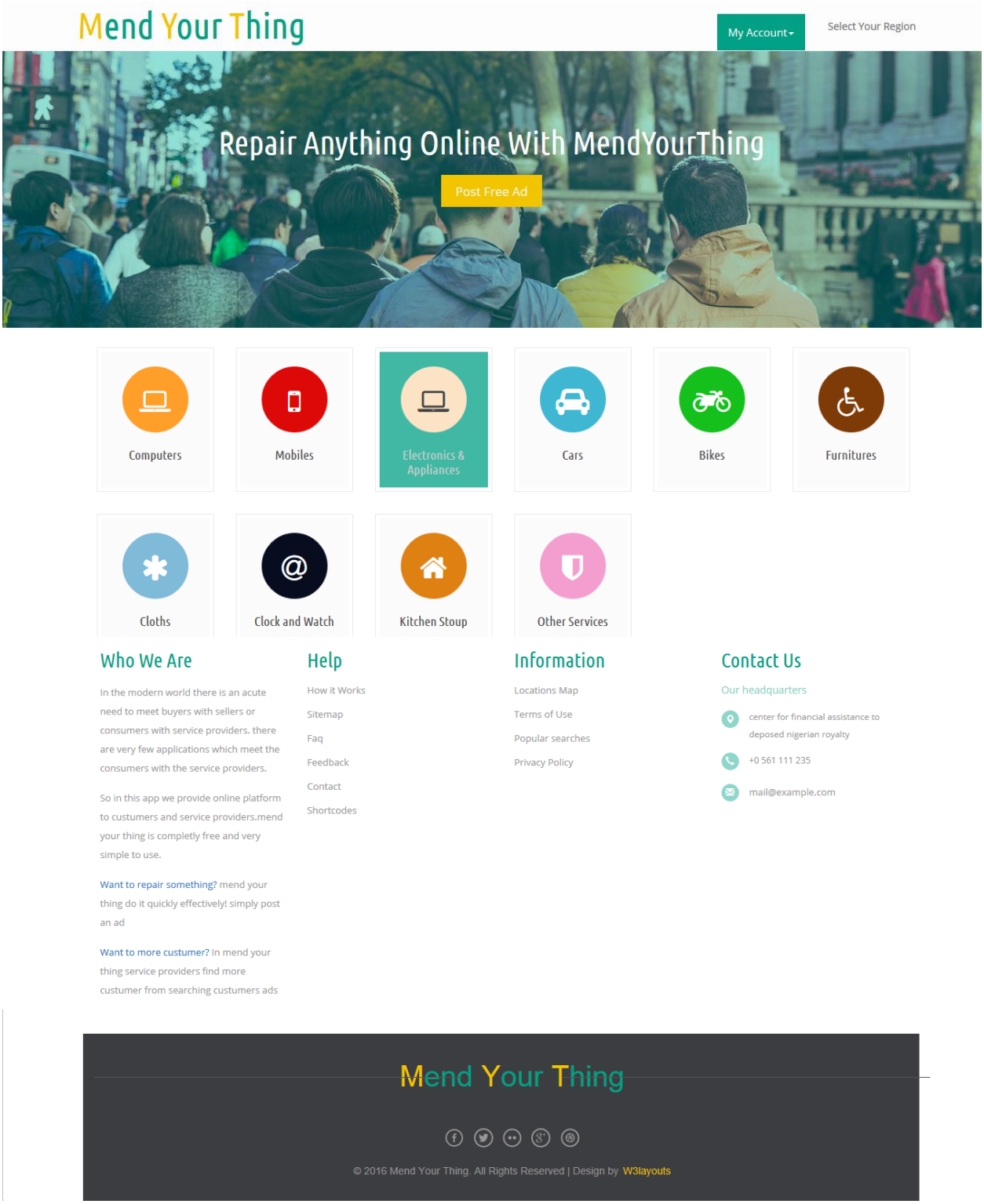
```
	@RequestMapping(value = "/logout.html", method = RequestMethod.GET)
	public String logout(ModelMap models) {
		session.invalidate();
		return "redirect:/index.html";
	}

	@RequestMapping(value = "/user.html", method = RequestMethod.GET)
	public String user(ModelMap models) {
		return "user";
	}

	@RequestMapping(value = "/provider.html", method = RequestMethod.GET)
	public String provider(ModelMap models) {

		return "provider";
	}



	@RequestMapping(value = "/edit_profile.html", method = RequestMethod.GET)
	public String edit_profile(@RequestParam String email, ModelMap models) {
		models.put("user", userDao.getUserByEmail(email));
		return "edit_profile";
	}

	@RequestMapping(value = "/edit_profile.html", method = RequestMethod.POST)
	public String edit_profile(@Valid @ModelAttribute User user,
			BindingResult bindingResult, SessionStatus sessionStatus) {
		System.out.println(bindingResult);
		if (bindingResult.hasErrors())
			return "edit_profile";

		user = userDao.update(user);
		return "redirect:/";
	}

	@RequestMapping(value = "/categories.html", method = RequestMethod.GET)
	public String categories(ModelMap models) {
		return "categories";
	}

	@RequestMapping(value = "/computers.html", method = RequestMethod.GET)
	public String computers(ModelMap models) {
		return "computers";
	}

	@RequestMapping(value = "/mobiles.html", method = RequestMethod.GET)
	public String mobiles(ModelMap models) {
		return "mobiles";
	}

	@RequestMapping(value = "/electronics-appliances.html", method =
RequestMethod.GET)
	public String electronicsappliances(ModelMap models) {
		return "electronics-appliances";
	}

	@RequestMapping(value = "/cars.html", method = RequestMethod.GET)
	public String cars(ModelMap models) {
```

```
	return "cars";
}

@RequestMapping(value = "/bikes.html", method = RequestMethod.GET)
public String bikes(ModelMap models) {
	return "bikes";
}

@RequestMapping(value = "/furnitures.html", method = RequestMethod.GET)
public String furnitures(ModelMap models) {
	return "furnitures";
}

@RequestMapping(value = "/cloths.html", method = RequestMethod.GET)
public String cloths(ModelMap models) {
	return "cloths";
}

@RequestMapping(value = "/clock-watch.html", method = RequestMethod.GET)
public String clockwatch(ModelMap models) {
	return "clock-watch";
}

@RequestMapping(value = "/kitchen.html", method = RequestMethod.GET)
public String kitchen(ModelMap models) {
	return "kitchen";
}

@RequestMapping(value = "/services.html", method = RequestMethod.GET)
public String services(ModelMap models) {
	return "services";
}

@RequestMapping(value = "/howitworks.html", method = RequestMethod.GET)
public String howitwork(ModelMap models) {
	return "howitworks";
}

@RequestMapping(value = "/sitemap.html", method = RequestMethod.GET)
public String sitemap(ModelMap models) {
	return "sitemap";
}

@RequestMapping(value = "/faq.html", method = RequestMethod.GET)
public String faq(ModelMap models) {
	return "faq";
}

@RequestMapping(value = "/feedback.html", method = RequestMethod.GET)
public String feedback(ModelMap models) {
	return "feedback";
}

@RequestMapping(value = "/contact.html", method = RequestMethod.GET)
public String contact(ModelMap models) {
	return "contact";
}

@RequestMapping(value = "/typography.html", method = RequestMethod.GET)
```

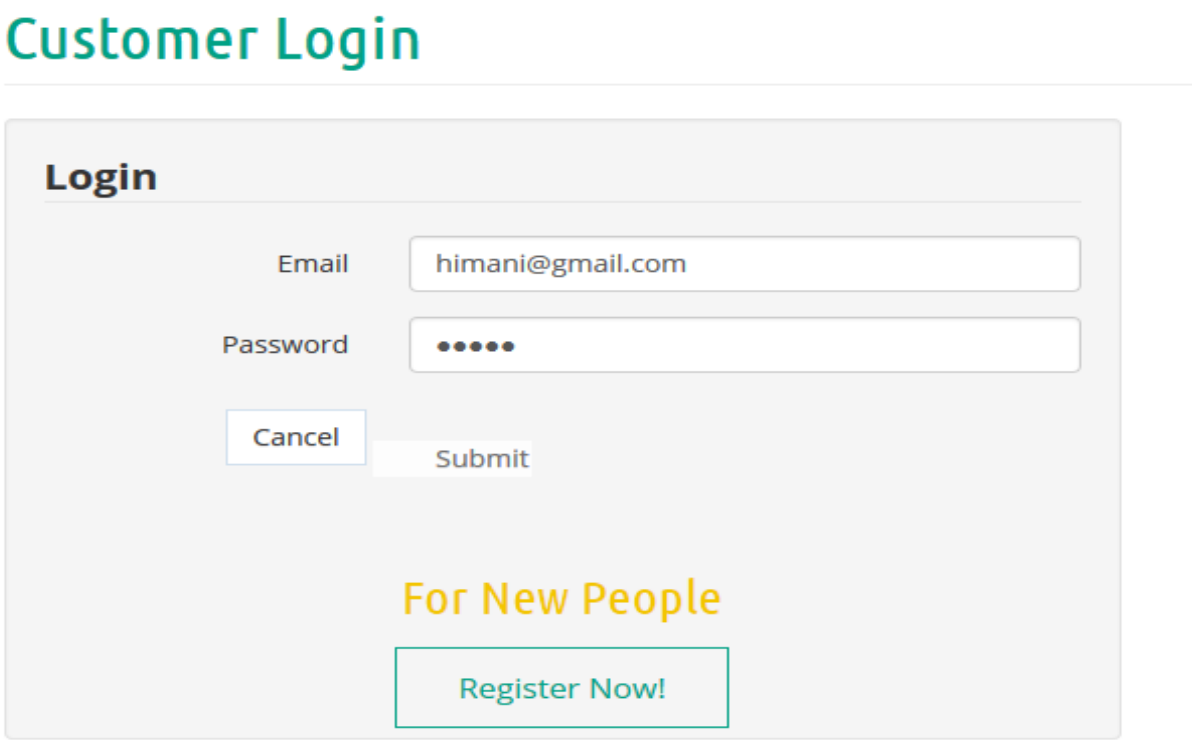
```
public String typography(ModelMap models) {
	return "typography";
}

@RequestMapping(value = "/regions.html", method = RequestMethod.GET)
public String regions(ModelMap models) {
	return "regions";
}

@RequestMapping(value = "/terms.html", method = RequestMethod.GET)
public String terms(ModelMap models) {
	return "terms";
}

@RequestMapping(value = "/popular-search.html", method = RequestMethod.GET)
public String popularsearch(ModelMap models) {
	return "popular-search";
}

@RequestMapping(value = "/privacy.html", method = RequestMethod.GET)
public String privacy(ModelMap models) {
	return "privacy";
}

@RequestMapping(value = "/single.html", method = RequestMethod.GET)
public String single(ModelMap models) {
	return "single";
}

@RequestMapping(value = "/all-classifieds.html", method = RequestMethod.GET)
public String allclassifieds(ModelMap models) {
	return "all-classifieds";
}
```

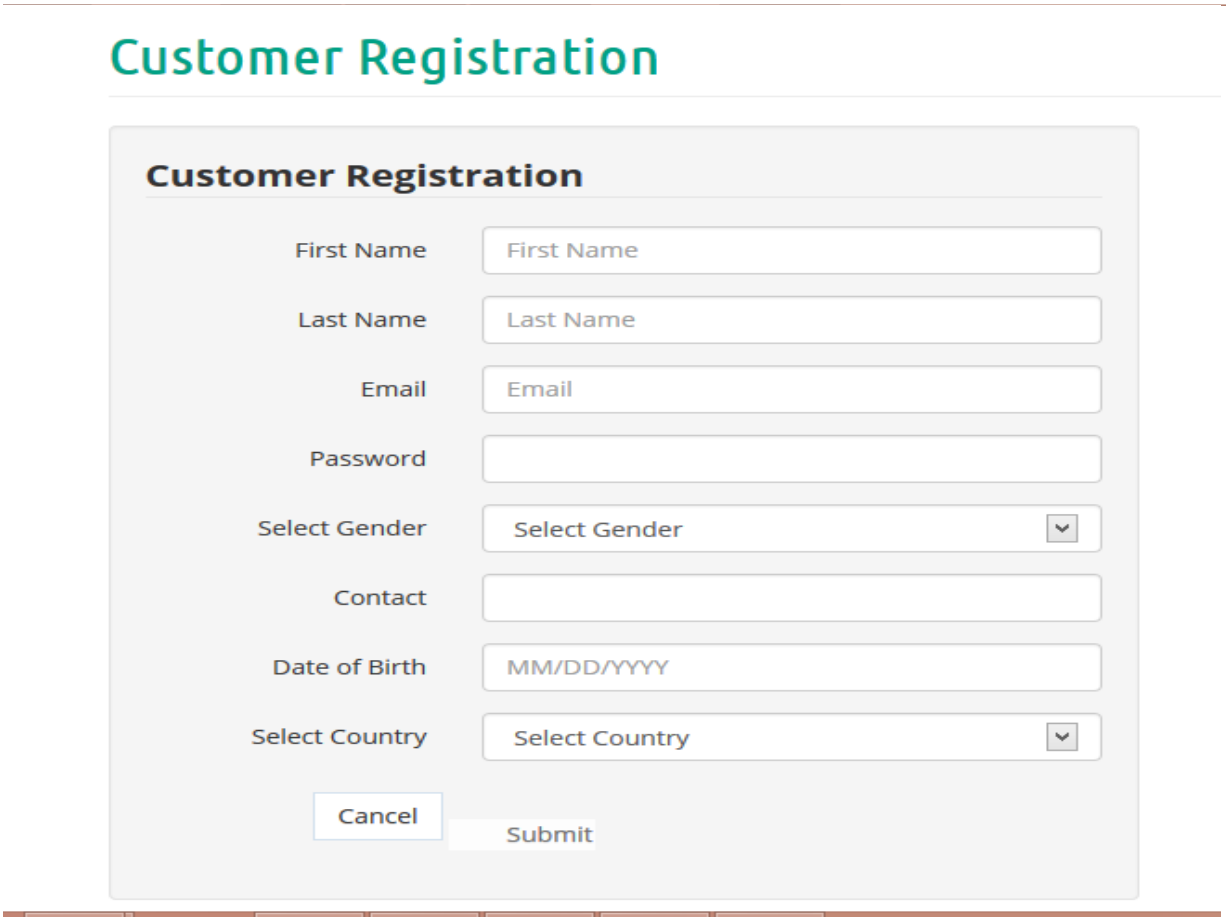
6.4 SCREEN LAYOUT

Home Page

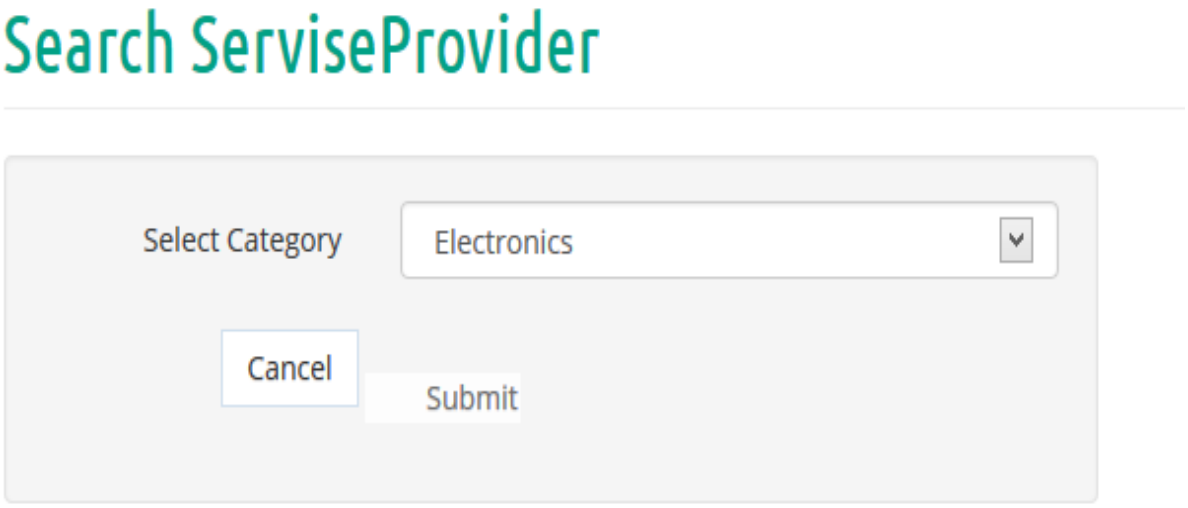


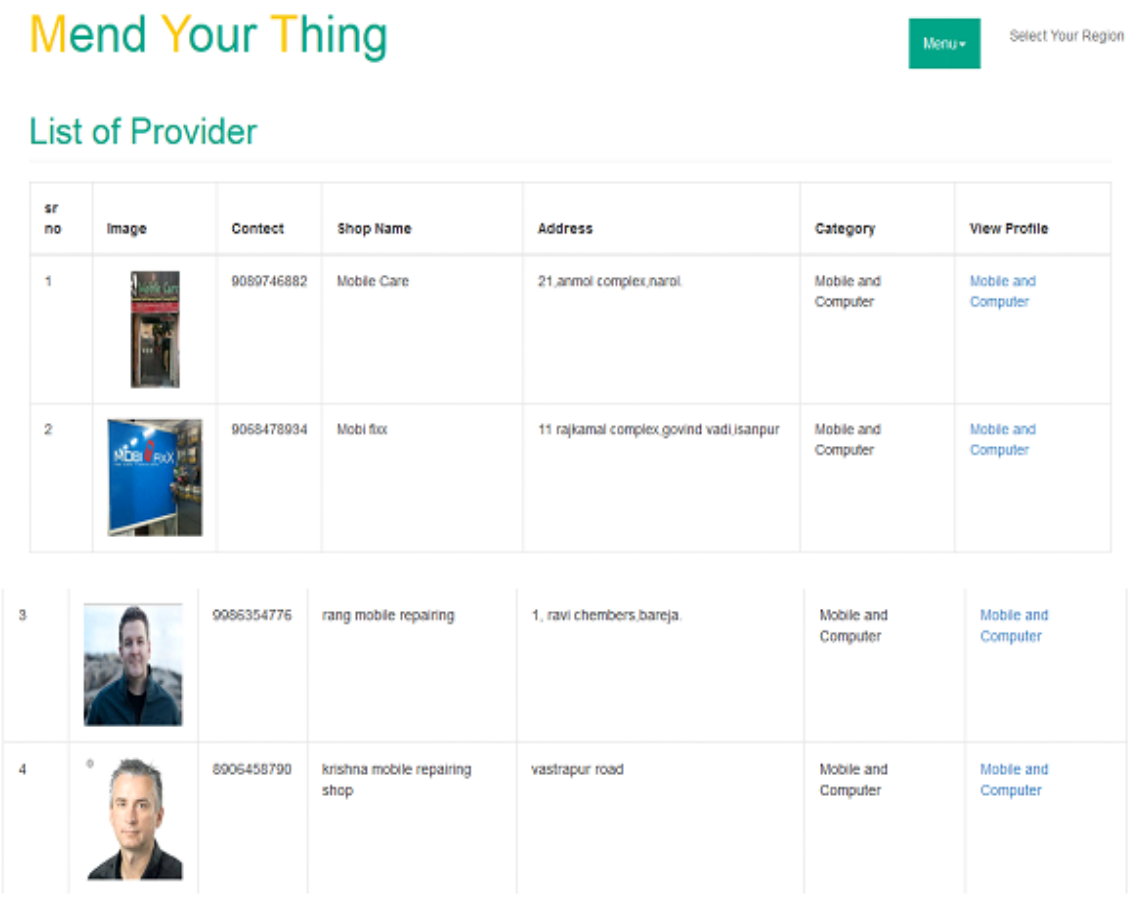
Customer Login & Registration



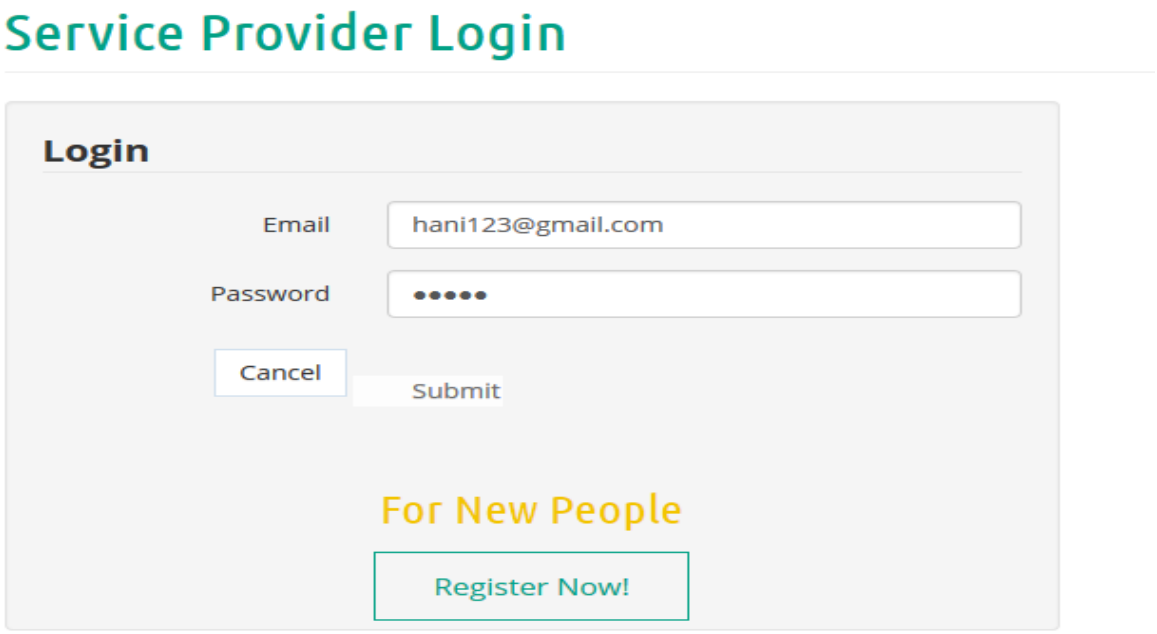


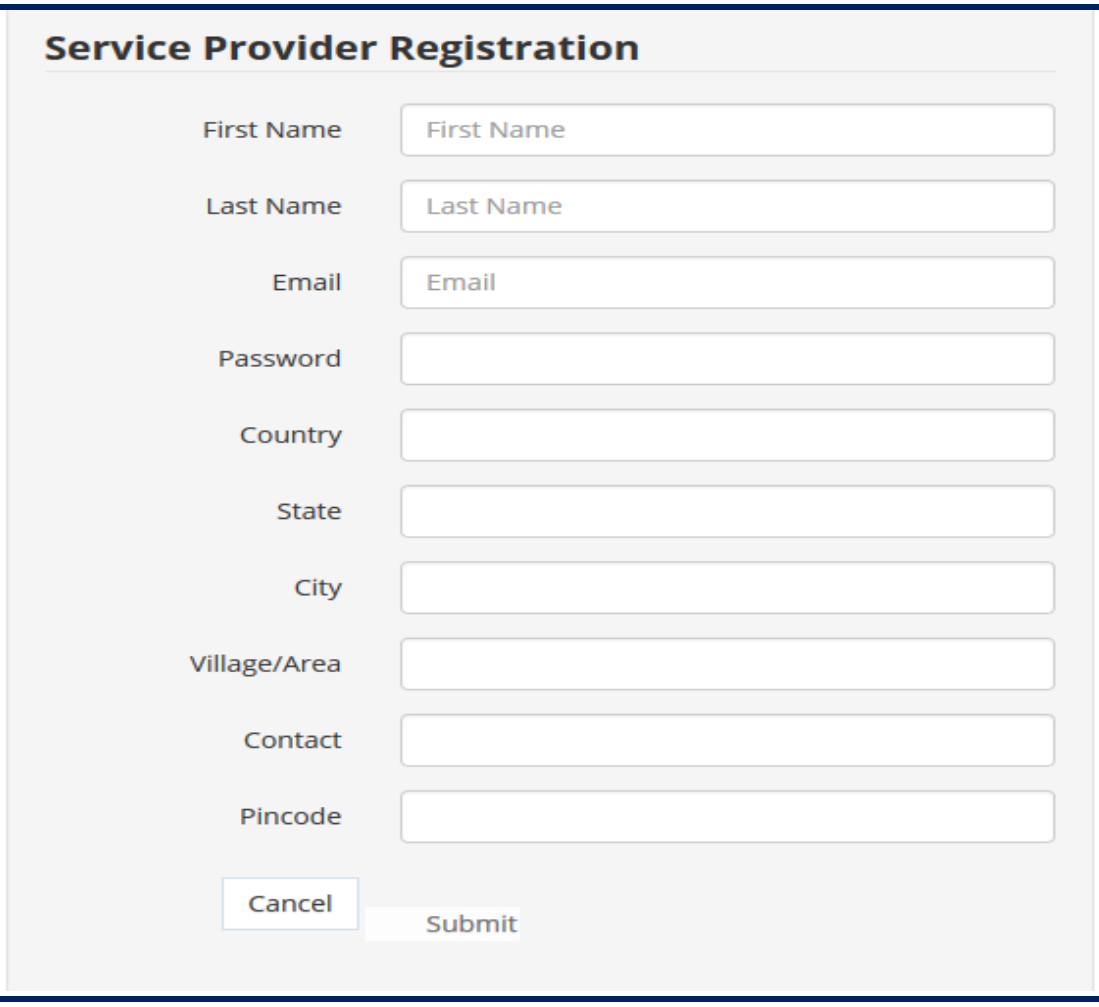
Customers can search Service Providers category vise



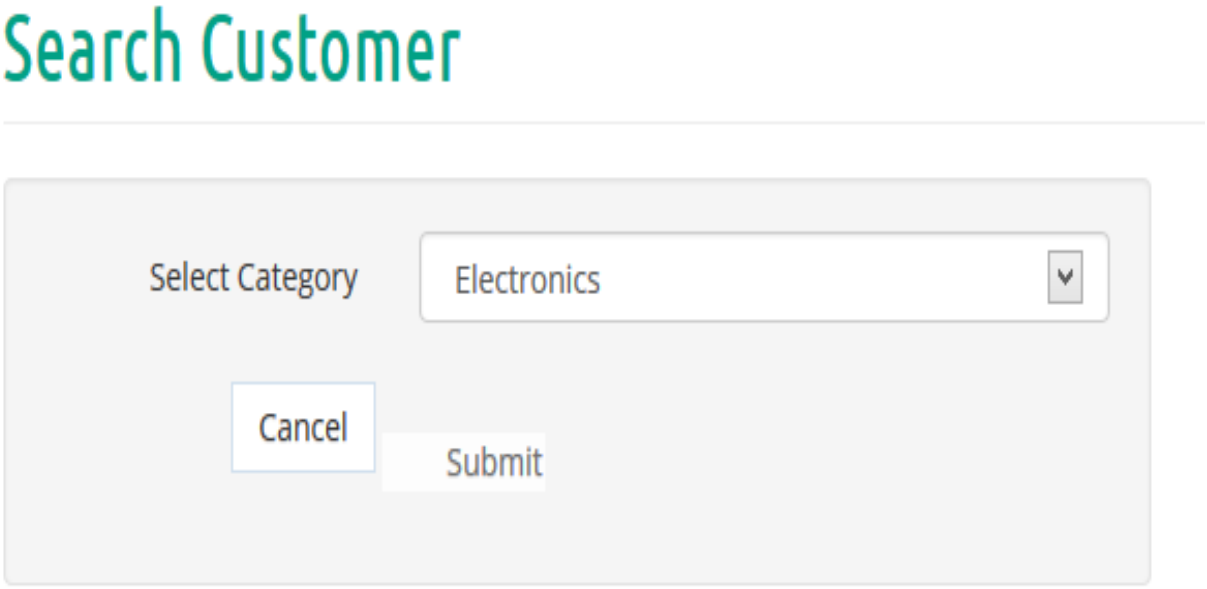


ServiceProvider Login & Registration





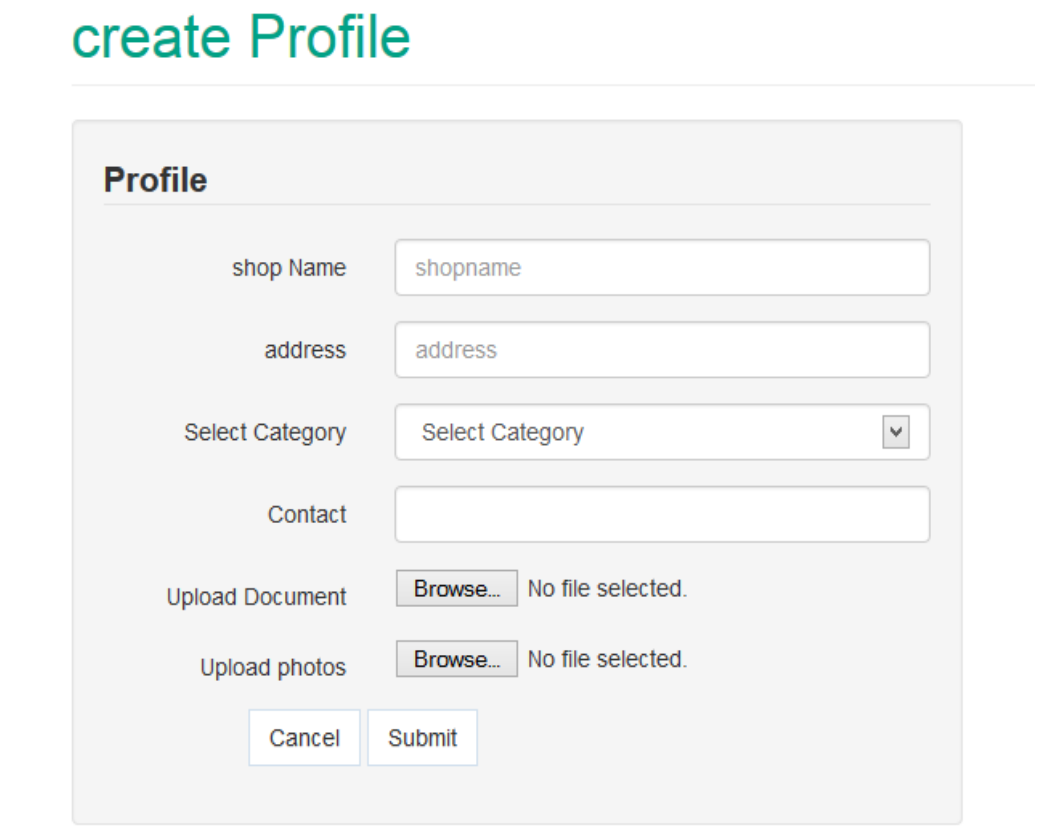
Service Providers can search Customers category vise



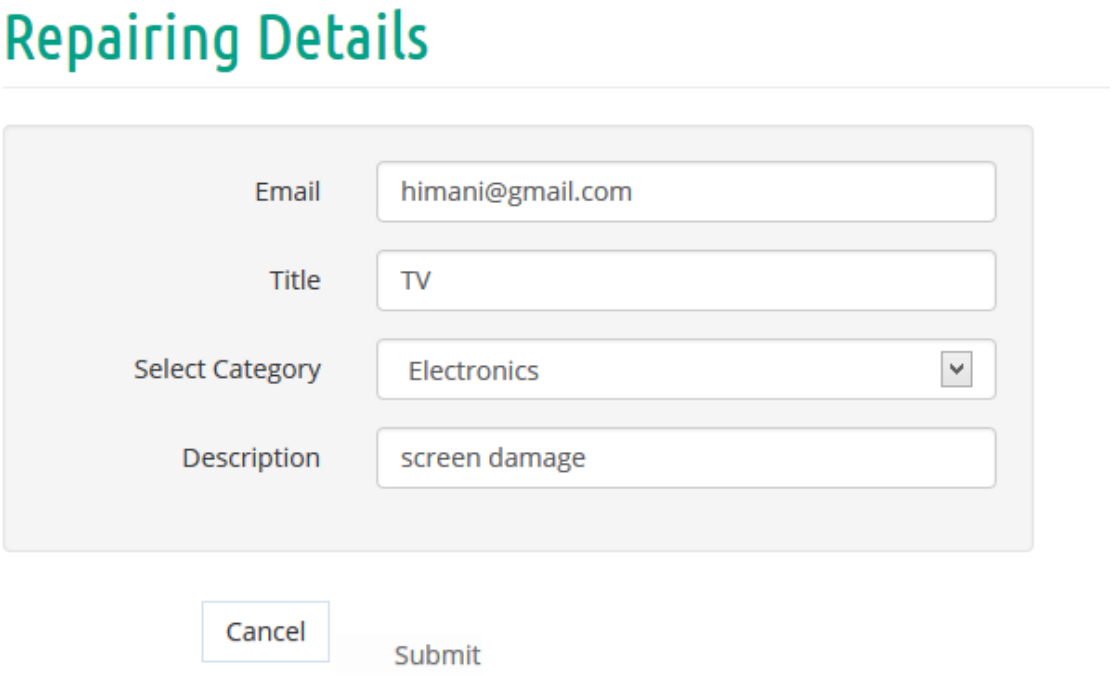
List of Categories

sr no	Email	Title	Category	Description	Interested
1	u@user	title	Electronic	repair	Interested
2	hani@gmail.com	fan	Electronic	fan	Interested
3	himani@gmail.com	ac	Electronic	ac repaiing	Interested
4	priya@gmail.com	refrizretor	Electronic	repair	Interested
5	nidhithakkar@gmail.com	tv	Electronic	tv repairing	Interested
6	kinu@gmail.com	tv	Electronic	repair	Interested
7	urvi@gmail.com	fan	Electronic	repairing	Interested
8	himani@gmail.com	TV	Electronic	screen damage	Interested

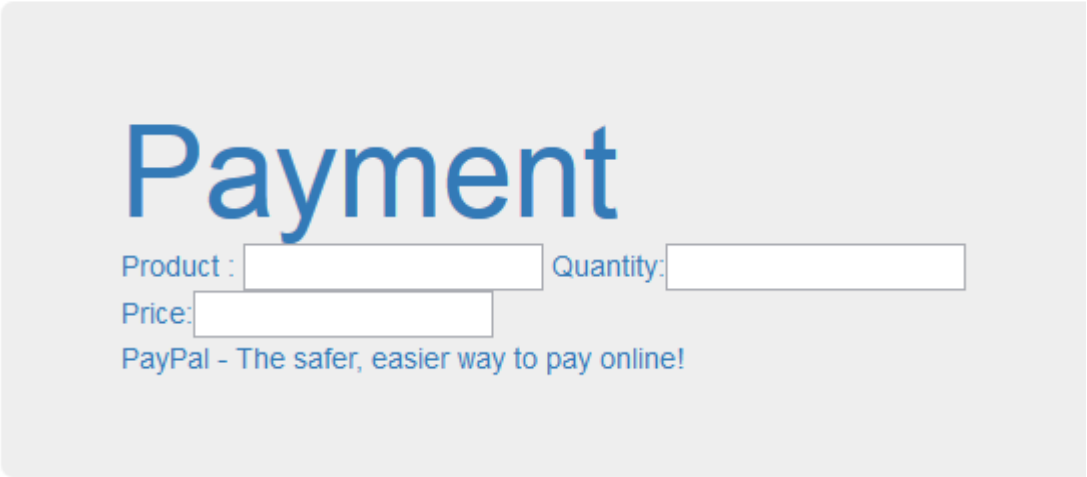
ServiceProvider can built their Profile



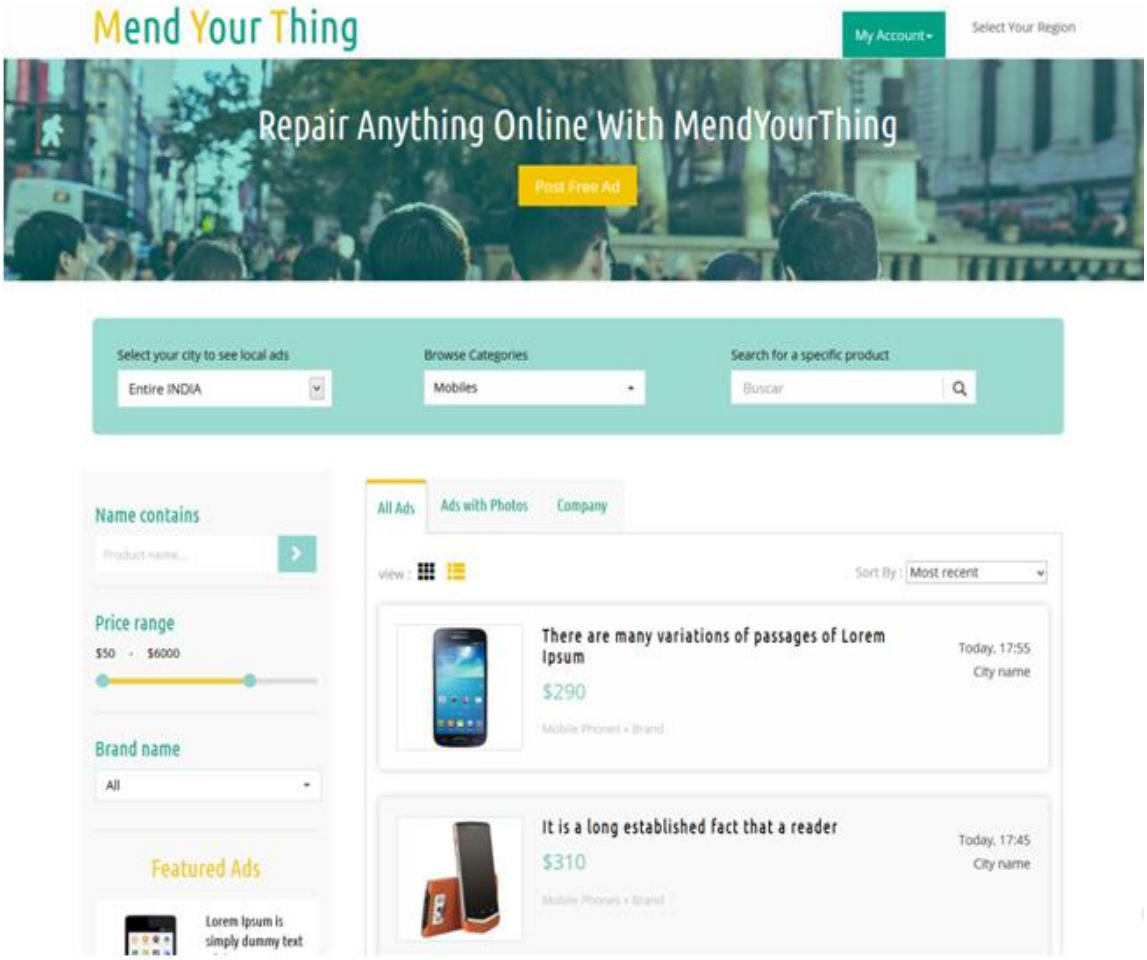
Customers & Visitors can Post Free Aid



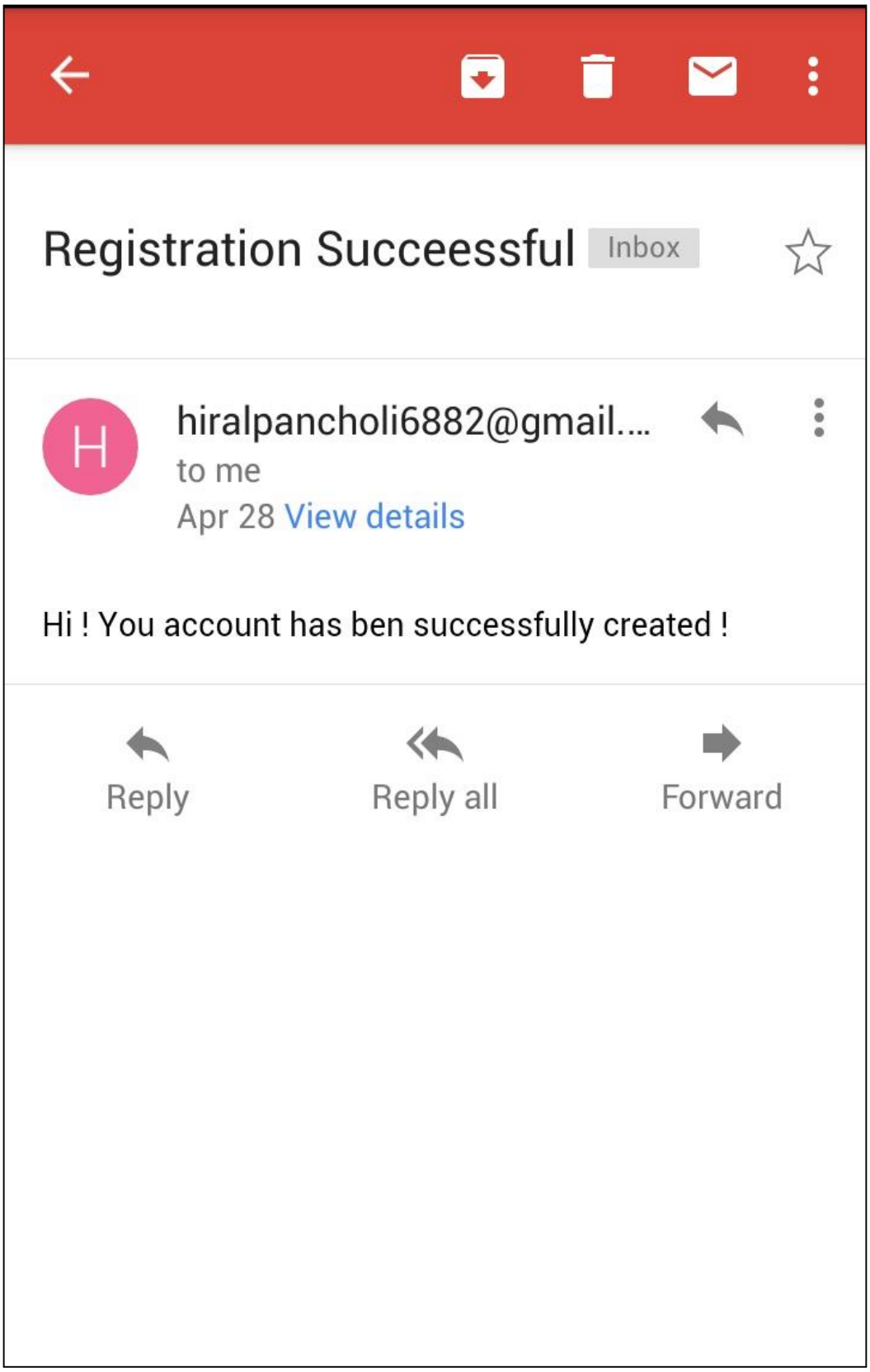
Payment



Category



Mail Functionality



Chapter-7 Testing

7.1 Testing Plan

Due to the confidentiality issues, we cannot provide details of the test plan to the development team. We will still add the core components that make up our testing plan.

7.2 Testing Strategy

A typical screen in various development technologies is tested at four levels before it goes for production.

Level 1 is generally the work to be tested by other developers or other interns (this is typical first level of testing where focus is not on requirement but end user testing) Ratio: 0% end user: 100 Technical.

Level 2 is level where a senior programmer comes into the testing cycle of the screen that was unit tested by the developer in this phase the goal is to test software for technical requirements specified. Ratio: 80% Technical: 20% end user.

Level 3 is where a tester will come into picture. The tester will test the software for both end user as well as technical point of view. Ratio: 50% Technical: 50% end user.

Level 4 is where we make the code at Release-Ready. Here screen is tested to the core and each and every standard must be followed and verified. Ratio: 80% User Testing – 20% Technical

This allows us to test a screen at four levels and at the end of four weeks when the screen goes to production, it is generally bug free because more people have looked at this screen from different viewpoints.

Unit Testing

- Black box and white box testing.
- Module interfaces are tested for proper information flow.
- Local data are examined to ensure that integrity is maintained.
- Boundary conditions are tested.
- Basis path testing should be used.
- All error handling paths should be tested.
- Drivers and/or stubs need to be developed to test incomplete software.

Integration Testing

Top-down integration testing

- Main control module used as a test driver and stubs are substitutes for components directly subordinate to it.
- Subordinate stubs are replaced one at a time with real components (following the depth-first or breadth-first approach).
- Tests are conducted as each component is integrated.
- On completion of each set of tests and other stub is replaced with a real component.
- Regression testing may be used to ensure that new errors not introduced.

Bottom-up integration testing

- Low level components are combined in clusters that perform a specific software function.
- A driver (control program) is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure.

7.3 Testing Methods

Models of Testing:-

There are different Models of testing. On the basis of testing methods there are two types of testing:

1. Black-box testing
2. White-box testing

White-box testing

There exist several popular white-box testing methodologies:

- Statement coverage
- Branch coverage
- Path coverage
- Condition coverage
- Mutation testing
- Data flow-based testing

Statement coverage

Statement coverage methodology focuses on, designing test cases so that,

- Every statement in a program is executed at least once.
- No statement in the program should remain unreachable.

Branch coverage

Test cases are designed such that, different branch conditions

- Give true and false values in some execution.
- All branches are traversed.

Branch testing guarantees statement coverage. It is a stronger testing compared to the statement coverage-based testing.

Path coverage

Design test cases such that, all linearly independent paths (LIP) in the program are executed at least once. To understand the path coverage-based testing we need to learn how to draw control flow graph of a program.

A control flow graph (CFG) describes,

- The sequence in which different instructions or statements of a program get executed.
- The way control flows through the program.

Condition coverage

Test cases are designed such that, each component of a composite conditional expression It help us to,

- Gives both true and false values.
- To check for all combination of conditions.

Black Box Testing:

Equivalence Partitioning

Black-box technique that divides the input domain into classes of data from which test cases can be derived an ideal test case uncovers a class of errors that might require many arbitrary test cases to be executed before a general error is observed

Equivalence class guidelines:

- If input condition specifies a range, one valid and two invalid equivalence classes are defined
- If an input condition requires a specific value, one valid and two invalid equivalence classes are defined

- If an input condition specifies a member of a set, one valid and one invalid equivalence class is defined
- If an input condition is Boolean, one valid and one invalid equivalence class is defined.

Boundary Value Analysis

Black-box technique that focuses on the boundaries of the input domain rather than its center

BVA guidelines:

- If input condition specifies a range bounded by values a and b, test cases should include a and b, values just above and just below a and b
- If an input condition specifies and number of values, test cases should be exercise the minimum and maximum numbers, as well as values just above and just below the minimum and maximum values
- Apply guidelines 1 and 2 to output conditions, test cases should be designed to produce the minimum and maxim output reports if internal program data structures have boundaries (e.g. size limitations), be certain to test the boundaries.

7.4 Test Cases

Case 1 - Admin Login

Condition	Action
Correct user and correct Password	Accepted
Correct user and incorrect Password	Rejected
Incorrect user and incorrect Password	Rejected
Incorrect user and correct Password	Rejected

Table 7.1 (Admin Login Test Case)

Case 2 - E-mail id Validation

Condition	Action
Correct e-mail id	Accepted
Correct email id & incorrect length	Rejected
Incorrect email id & correct length	Rejected

Table 7.2 (E-mail id validation Test Case)

Case 3 - User Validation

Condition	Action
Unique username, password length more than 6 character and unique and valid Email Id.	Accepted
Conflicted username, password length more than 6 character and unique and valid Email Id.	Rejected
Unique Username, password length less than 6 character and unique and valid Email Id	Rejected
Unique Username, password length more than 6 character and conflicted or invalid Email Id.	Rejected

Table 7.3 (Create User Test Case)

Chapter 8: Limitations & Future Enhancement

8.1 Limitation

- Limitation of this system is users must require internet connection with them. Without internet connection they can't use this system.
- Sometimes service provider cannot define exact rate of product because of improper description of product.
- User should have mobile phone with camera.

8.2 Future Enhancements

In future enhancements of this system is as below.

- This system basically design for one country in future shall be modify to more then one country..
- Some modifications must require like including google map.

Chapter-9 Conclusion And Discussion

9.1 Self Analysis of Projects Viability

- Every project work, does not matter software project or any other project, could not be the result of sole effort; even though, developer team consists of two members. We think success of any project doesn't depend only on better software development skills, but also, zeal to listen and help the users. Only user interaction at development phase can give you an idea to fulfill their requirements. Because, software product is of developer's up to the development or installation phase, but then after it is of user's. We experienced that do make the software for users not for fame of being better programmer or software developer. Ultimately, users appraise your efforts not your peers.

9.2 Summary of Project Work

- We have completed our project work, based on using software engineering and system analysis and design approach. We have done our work with preplanning scheduling related with time constrains and result oriented progress in project development.

- Our project was started in the second week of July - 2015 and we completed our project documentation and partial design of our projection the month of November - 2015. Initially, there were some problem regarding some phases but we have planned to resolve those problems. We have divided our work in different phases and solved problems and difficulties. We completed our project in the month of May – 2016.

References

- http://www.headfirstlabs.com/books/hfsvlt/
- https://www.mysql.com/
- http://www.writing.engr.psu.edu/workbooks/format.html
- http://www.w3schools.com/
- http://www.tutorialspoint.com/uml/
- http://www.eclipse.org
- http://www.codecademy.com
- Head First Servelets and Jsp-2nd Edition.
- Web application development

Pattern Drafting & Completion Certificate

Pattern Drafting

GIC Patent Drafting Exercise Team ID: 47323

GTU Innovation Council

Patent Drafting Exercise (PDE)

FORM 1 THE PATENTS ACT 1970 (39 OF 1970) & THE PATENTS RULES, 2003 APPLICATION FOR GRANT OF PATENT	(FOR OFFICE USE ONLY) Application No: Filing Date: Amount of Fee paid: CBR No:_____

1. Applicant(s) :

ID	Name	Nationality	Address	Mobile No.	Email
1	Patel Hani Samirkumar	Indian	Computer Science & Engineering , Narnarayan Shastri Institute Of Technology, Jetalpur , Gujarat Technologycal University.	8128773437	hanipatel258@gmail.com
2	Patel Himani Jigneshkumar	Indian	Computer Science & Engineering , Narnarayan Shastri Institute Of Technology, Jetalpur , Gujarat Technologycal University.	9998032191	himani.patel1295@gmail.com

2. Inventor(s):

ID	Name	Nationality	Address	Mobile No.	Email
1	Patel Hani Samirkumar	Indian	Computer Science & Engineering , Narnarayan Shastri Institute Of Technology, Jetalpur , Gujarat Technologycal University.	8128773437	hanipatel258@gmail.com
2	Patel Himani Jigneshkumar	Indian	Computer Science & Engineering , Narnarayan Shastri Institute Of Technology, Jetalpur , Gujarat Technologycal University.	9998032191	himani.patel1295@gmail.com

3. Title of invention/Project:
Mend Your Things

4. Address for correspondence of applicant/authorized patent agent in India

Name: Patel Hani Samirkumar
Address: Computer Science & Engineering , Narnarayan Shastri Institute Of Technology, Jetalpur , Gujarat Technological University.
Mobile: 8128773437

Note : This is just a mock Patent Drafting Exercise (PDE) for semester 8, BE students of GTU.These documents are not to be submitted with any patent office. Page 1

Email ID: hanipatel258@gmail.com

6. Priority particulars of the application(s) field in convention country

Country	Application No.	Filing Date	Name of the Applicant	Title of the Invention
N/A	N/A	N/A	N/A	N/A

8. Particulars for filing patent co-operation treaty (pct) national phase Application

International application number	International filing date as alloted by the receiving office
N/A	N/A

7. Particulars for filing divisional application

Original(First) Application Number	Date of filing of Original (first) application
N/A	N/A

8. Particulars for filing patent of addition

Original(First) Application Number	Date of filing of Original (first) application
N/A	N/A

9. DECLARATIONS:

(I) Declaration by the Inventor(s)

I/We, the above named inventor(s) is/are true & first inventor(s) for this invention and declare that the applicant(s) herein is/are my/our assignee or legal representative.

Date : 11 - April - 2016

	Name	Signature & Date
1	Patel Hani Samirkumar	_____
2	Patel Himani Jigneshkumar	_____

(II) Declaration by the applicant(s) in the convention country

I/We, the applicant (s) in the convention country declare that the applicant(s) herein is/are my/our assignee or legal representative.applicant(s)

(III) Declaration by the applicant(s)

I/We, the applicant(s) hereby declare(s) that:-

- ✓ I am/We in possession of the above mentioned invention.
- ✓ The provisional/complete specification relating to the invention is filed with this apication.
- ✗ The invention as disclosed in the spcification uses the biological material from India and the necessary permission from the competent authority shall be submitted by me/us before the grant of patent to me/us.
- ✗ There is no lawful ground of objection to the grant of the patent to me/us.

☑ I am/we are the assignee or the legal representative of true & first inventors.

☒ The application or each of the application,particulars of each are given in the para 5 was the first applicatin in the convention country/countries in respect of my/our invention.

☒ I/we claim the priority from the above mentioned applications(s) filed in the convention country/countries & state that no application for protection in respect of invention had been made in a convention country before that date by me/us or by any person

☒ My/Our application in India is based on international application under Patent Cooperation Treaty (PCT) as mentioned in para 6

☒ The application is divided out of my/our application(s) particulars of which are given in para 7 and pray that this application may be treated as deemed to have been filed on _____under section 16 of the Act.

☒ The said invention is an improvement in or modification of the invention particulars of ehivh are given in para 8.

10. Following are the attachments with the application:

☑ (a) Provisional specification/Complete specification

☒ (b) Complete specification(in confirmation with the international application) / as amended before the international Preliminary Examination Authority (IPEA),as applicable(2 copies),No.of pages.....No.of claims.....

☑ (c) Drawings (in confirmation with the international application)/as amended before the international Preliminary Examination Authority(IPEA),as applicable(2 copies),No.of sheets....

☒ (d) Priority documents

☒ (e) Translations of priority documents/specification/international search reports

☑ (f) Statement and undertaking on Form 3

☒ (g) Power of Authority

☒ (h) Declaration of inventorship on Form 5

☒ (i) Sequence listing in electronic Form

☒ (j) ..... Fees Rs.XXX in Cash /Cheque/Bank Draft bearin No.XXX Date: XXX on XXX Bank.

I/We hereby declare that to the best of my /our knowledge, information and belief the fact and mtters stated herein are correct and I/We request that a patent may be granted to me/us for the said invention.

Dated this 11 day of April , 2016

	Name	Signature & Date
1	Patel Hani Samirkumar	_____
2	Patel Himani Jigneshkumar	_____

FORM 3
THE PATENTS ACT, 1970
(39 OF 1970)
&
THE PATENTS RULES, 2003
STATEMENT AND UNDERTAKING UNDER SECTION 8

1. Declaration :

I/We, Patel Hani Samirkumar ,
Patel Himani Jigneshkumar

2. Name, Address and Nationality of the joint Applicant :

Patel Hani Samirkumar (Indian)
Address :Computer Science & Engineering , Narnarayan Shastri Institute Of Technology, Jetalpur , Gujarat Technologyoal University.

Patel Himani Jigneshkumar (Indian)
Address :Computer Science & Engineering , Narnarayan Shastri Institute Of Technology, Jetalpur , Gujarat Technologyoal University.

Here by declare :

(I) that I/We have not made any application for the same/substantially the same invention outside India.

(II) that the right in the application(s) has/have been assigned to,

Name of the Country	*Date of Application*	*Application Number*	*Status of the Application*	*Date of Publication*	*Date of Grant*
N/A	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*

(III) that I/We undertake that up to the date of grant of patent by the Controller , I/We would keep him inform in writing the details regarding corresponding application(s) for patents filed outside India within 3 months from the date of filing of such application.

Dated this 11 day of April , 2018

3. Signature of Applicants :

Sign and Date
Patel Hani Samirkumar

Sign and Date
Patel Himani Jigneshkumar

To
The Controller of Patent
The Patent Office, at **Mumbai.**

GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VIII, ACADEMIC YEAR 2015-2016

Date of certificate generation : 12 April 2016 (16:28:12)

This is to certify that, ***Patel Hani Samirkumar*** (Enrolment Number - 120340131028) working on project entitled with ***Mend Your Things*** from ***Computer Science & Engineering*** department of ***Narnarayan Shastri Institute Of Technology, Jetalpur*** had submitted following details at online project portal.

Periodic Progress Reports (PPR)	**Completed**
Business Model Canvas (Image)	**Completed**
Business Model Canvas (Report)	**Completed**
Patent Drafting Exercise (PDE)	**Completed**
Final Plagiarism Report	**Completed**
Final Project Report	**Completed**

Name of Student : Patel Hani Samirkumar Name of Guide : Mrs. Vatika Vikas Tayal

Signature of Student : _____ *Signature of Guide : _____

Disclaimer :
This is a computer generated copy and does not indicate that your data has been evaluated. This is the receipt that GTU has received a copy of the data that you have uploaded and submitted as your project work.

*Guide has to sign the certificate, Only if all above activities has been Completed.

GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VIII, ACADEMIC YEAR 2015-2016

Date of certificate generation : 12 April 2016 (16:25:18)

This is to certify that, ***Patel Himani Jigneshkumar*** (Enrolment Number - 120340131018) working on project entitled with ***Mend Your Things*** from ***Computer Science & Engineering*** department of ***Narnarayan Shastri Institute Of Technology, Jetalpur*** had submitted following details at online project portal.

Periodic Progress Reports (PPR)	**Completed**
Business Model Canvas (Image)	**Completed**
Business Model Canvas (Report)	**Completed**
Patent Drafting Exercise (PDE)	**Completed**
Final Plagiarism Report	**Completed**
Final Project Report	**Completed**

Name of Student : Patel Himani Jigneshkumar Name of Guide : Mrs. Vatika Vikas Tayal

Signature of Student : _____ *Signature of Guide : _____

Disclaimer :
This is a computer generated copy and does not indicate that your data has been evaluated. This is the receipt that GTU has received a copy of the data that you have uploaded and submitted as your project work.

*Guide has to sign the certificate, Only if all above activities has been Completed.