

PAPER NAME

MAIN PROJECT_ANAND_AS (1).pdf

WORD COUNT

7423 Words

CHARACTER COUNT

40224 Characters

PAGE COUNT

54 Pages

FILE SIZE

1.7MB

SUBMISSION DATE

Jul 22, 2022 3:34 PM GMT+5:30

REPORT DATE

Jul 22, 2022 3:35 PM GMT+5:30

● 19% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 19% Internet database
- 8% Publications database
- Crossref database

● Excluded from Similarity Report

- Small Matches (Less than 8 words)

23 CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

In order to link clients with service providers including mechanics, plumbers, electricians, photographers, and house cleaners, this project entails developing an online platform. People may utilize it to find offline labor to help them with daily duties. Think of a situation when your car breaks down in a strange place. You will thus require help in resolving the problem. Visit the website, decide the services you need, make an appointment for the time, day, and place that work best for you, and then reserve the service. The service staff might potentially agree to the client's request and carry out their duties on-site.

1.2 PROJECT SPECIFICATION

The proposed solution is a website where customers may book appointments with offline service providers. The system consists of two parts. the following:-

➤ Customer Module

- **Login:** - Customers can log in using their email address.
- **Browse Service:** - The app's available services and service providers may be found by using the search box.
- **Reviews and Ratings:** - Customers may share their experiences with this site's services in detail by giving ratings and reviews.
- **Service History:** - A service request can be changed or cancelled by the customer.

➤ Service Provider Module

- **Verification:** - After the admin has verified the required documentation, service providers can start their activity.
- **Receive Requests:** - Any booking requests will be communicated to the service providers.
- **Manage Requests:** - Service requests may be accepted or rejected by them. They can designate a window of time within which they will be available.
- **Service History:** - The service providers may see all of their finished service requests.

- **Customer Feedback:** - The services offered by service providers can be reviewed by users.

➤ **Admin Module**

- **Service Provider Management:** - The documentation of recently onboarded service providers may be promptly verified.
- **Service Management:** - Services and service categories may be updated, added, or removed by the administrator.
- **Regulate Reviews:** - Regularly reading customer feedback on service providers may help you develop your products.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is the procedure for gathering and analysing data, spotting problems, and suggesting system modifications. It's a duty that necessitates intensive communication between system developers and consumers. Any system's development process must include a step for system analysis or study. The system is carefully inspected and evaluated. The system analyst investigates the functioning of the present system in depth. The input from the system is seen in its entirety, and the system is acknowledged. The multiple processes that make up an organization's outputs are connected.

To thoroughly evaluate the system, a variety of tools must be employed, including surveys and interviews. The information gathered from various sources must be carefully examined in order to draw a judgement. The last stage is to comprehend how the system functions. This is the term used to describe the current system. Problem areas have been discovered after a thorough analysis of the current system. The designer now assumes the position of a problem-solver and works to find solutions to the issues the business is facing. Only suggestions serve as solutions. After doing an analytical comparison between each solution and the current system, the best option is selected. The user has the choice to accept or reject an idea when it is presented to them.

A number of approaches, including surveys and interviews, must be used to analyse the system in-depth. It is essential to thoroughly analyse the information gleaned from numerous sources before drawing a decision. As a result, you'll understand the operation of the system better. A "current system" is what is being discussed. Recently, weaknesses in the present system were discovered after a comprehensive analysis. The designer now assumes the position of a problem solver to help the company with its challenges. The answers are offered as suggestions. The best choice is made after thoroughly evaluating each potential replacement for the current system. The user is given the option to accept or reject the idea when it is presented to them.

2.2 EXISTING SYSTEM

The existing system is not entirely automated. Customers can register and schedule service. Each consumer has the option of creating their own profile. The suggested system remedies the flaws of the existing system. To accommodate extra information and make the system more efficient, versatile, and secure, the present system must be

modified. Customers may view all services based on their budget using the new system.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Lack of availability in towns and cities
- Important knowledge is challenging to keep up in books.
- The essential reports must be produced manually more often.
- Can't track service order
- Late delivery

2.4 PROPOSED SYSTEM

The suggested system is also described, along with a list of all the problems it would address. When someone requires assistance with little but important home tasks, a problem arises when service-skilled people are not available or trustworthy providers are hard to discover. Our online service system is the fastest and most convenient way to complete your tasks. We wish to assist you in identifying the most practical, practical, and, most importantly, considerate solutions to all of your domestic issues.

2.5 ²ADVANTAGES OF PROPOSED SYSTEM

The system is simple to develop and use. The system operates in virtually any environment and consumes incredibly minimal system resources. It features the following things:

➤ **Reduce Time: -**

It is really difficult to locate service providers in our area based on their service costs. Users will easily locate them by using the suggested technique based on their budget. On this platform, each service has a distinct category. Users will easily locate them by using the suggested technique based on their budget. On this platform, each service has a distinct category.

➤ **Ensure data accuracy: -**

The recommended fix eliminates errors generated by people while entering user information during registration.

➤ **Better service: -**

The device eliminates the need for hard copy storage. We can also save time and money

by carrying out the same task in a different way. Without losing any information, the data may be preserved for a longer period of time.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

To ascertain if the project will achieve the organization's objectives for the amount of time, effort, and labour put into it, a feasibility study is carried out. A feasibility study allows the project's author to forecast the usefulness and future of the project. A feasibility study is built on the workability of a system plan—the organization's influence, capability to satisfy user requirements, and efficient use of resources. As a result, before being approved for development, a new proposal is often put through a feasibility evaluation.

The project's viability is discussed in the article along with a number of carefully considered factors, such as the project's technical, economic, and operational feasibilities. The following traits apply to it:

3.1.1 Economic Feasibility

This feasibility assessment analyses if the system can be constructed with the available money. The Online Service Providing Platform's development doesn't cost a lot of money. Having a strategy in place can help you achieve this on a budget. It is therefore economically feasible.

The following significant financial concerns were brought up during the preliminary investigation:

- The expenses carry out a thorough system analysis.
- The cost of the software and hardware.
- The benefits manifest themselves as reduced costs or less expensive mistakes.

There won't be any manual expenses because the recommended method is being developed as part of a project. Furthermore, the fact that all of the resources are already available implies that the system might be constructed at a low cost. The system used, development expenses, and hosting charges made up the project's three cost categories. All estimates indicate that the project was made at a reasonable cost. because only opensource software was used in its construction.

3.1.2 Technical Feasibility

➤ Selection of the front-end and back-end

A crucial element of the proposed project is the choice of suitable front-end and back-end code. According to organizational requirements, it must be scalable and extendable, flexible, resilient, and easy to debug and maintain. It must also include a graphical user interface to help employees who do not have IT expertise. So, for front-end development, I employed React JS.

➤ Back-end Selection:

We took into account various user support, speedy data processing, built-in security measures, easy data retrieval and maintenance, stored processes, and front-end connection ease while selecting the back-end. I choose to utilize FIREBASE as the back-end due of the aforementioned features. This stage's technological viability is frequently its most difficult component. Technical feasibility research must come after the analysis and definition phase. It stresses how well the present computer system's hardware, software, and other elements can support the suggested system.

3.1.3 Behavioral Feasibility

The following inquiries are part of the suggested system:

- Is there enough user support?
- Does the suggested system constitute a risk?

Since the project is in line with the goals, it will be profitable once it is developed and completed. After a comprehensive analysis of all behavioral parameters, it is determined that the project is behaviorally feasible.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

²² Processor	- Intel core i3
RAM	- 4 GB
Hard disk	- 1 TB

3.2.2 Software Specification

Front End	- JSX, CSS
Backend	- FIREBASE
Client on PC	- Windows 7 and above.
Technologies used	- React JS, Node JS

3.3 SOFTWARE DESCRIPTION

3.3.1 REACT JS

Declarative, quick, and flexible, ReactJS is a JavaScript library for building reusable UI components. It is a front-end component library that is open-source and solely in charge of the application's display layer. Facebook developed and maintained it, and eventually used it in applications like WhatsApp and Instagram.

The creation of User Interfaces (UI) that improve app performance is ReactJS' main objective. The programme is faster because to the usage of virtual DOM (JavaScript object). JavaScript's virtual DOM is speedier than the traditional DOM. Both the client and server sides, as well as other frameworks, may use ReactJS. It uses component and data patterns to make larger programmes easier to maintain and more readable. You may divide your entire programme into several components using ReactJS. ReactJS uses the same traditional data flow but operates on a virtual DOM rather than the browser's actual Document Object Model (DOM). instead of making changes to the content in a browser.

3.3.2 FIREBASE

A cloud-hosted database is called the Firebase Real-time Database. All clients that are connected receive real-time data sync in JSON format. By utilizing our Apple, Android, and JavaScript SDKs to build cross-platform apps, all of your customers share a single Real-time Database instance and receive the most recent data instantly. modifications.

➤ **FIREBASE is a ²⁷real-time database**

A database hosted in the cloud that uses JSON to store data is called Firebase Real-time Database. Data synchronization in real time is provided to each linked client. When using our iOS and JavaScript SDKs to build cross-platform applications, all of our clients share a single Real-time Database instance and get real-time updates with the most recent data. We can store and sync data in real-time among our users thanks to the Firebase Real-time Database, a NoSQL database. It's a large JSON entity that programmers could manage instantly. Through a single API, the Firebase database provides the application with the data's current value and any changes made to it. Real-time synchronization allows our consumers to view their data from any device, including the web.

➤ **FIREBASE Authentication.**

Authenticating individuals is crucial, and if we have to develop all of this programming ourselves, it will be much more challenging. Using Firebase to achieve this is quite straightforward. We can add a complete sign-in system to our project using Firebase UI Authentication, and Firebase will take care of the user experience. For adding authentication to websites and mobile applications, Firebase UI provides a drop-in solution.

➤ **FIREBASE Fire store.**

⁹We have two choices with Firebase: Cloud Firestore and ⁹Firebase Real-time Database. Despite being more recent, Cloud Firestore does not take the place of Firebase Real-time Database. A scalable and flexible NoSQL cloud database is Cloud Fire store. Data is saved and synced using it for server and client ⁹development. It is used for mobile, web, and server development by Google Cloud Platform and Firebase. Similar to the Firebase Real-time Database, it uses real-time listeners to continually synchronize our data with the client app. By offering offline support for mobile and web, it enables us to create responsive apps that

function without regard to network latency or Internet availability.

➤ **FIREBASE Hosting**

We can easily and securely host our web application, static and dynamic content, and micro services with the aid of Firebase Hosting. It is the hosting of high-quality web content for developers. With a single command, we can start web applications fast and easily and send both static and dynamic content to a worldwide content delivery network. To develop and host microservices on Firebase, we may utilize Firebase Hosting with Cloud Function or Cloud Run.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Any developed system or product must first go through the design process. Design is a creative process. A correct design is required for a system to be effective. According to one definition, "design" is "the act of specifying a process or a system in sufficient detail to permit its actual implementation using a range of approaches and concepts." It is defined as the process of using a range of methodologies and concepts to properly specify a device, process, or system for physical implementation. The technical foundation of the software engineering process is called software design, and it is used regardless of development methodology.

The architectural detail necessary to construct a system or product is generated through system design. This programmer has followed the finest design process imaginable, fine-tuning all efficiency, performance, and accuracy levels, as with any rigorous technique. A document intended for users becomes a document for programmers or database employees during the design phase. Logical design and physical design are the two stages of system design.

4.2 UML DIAGRAM

UML is a standard language for describing, visualizing, producing, and documenting software system artefacts. UML was designed by the Object Management Group (OMG), and the OMG received a draught of the UML 1.0 definition in January 1997.

UML is an abbreviation for Unified Modeling Language. UML is distinct from other popular programming languages like as C++, Java, COBOL, and others. The graphical language UML is used to generate software designs. To design, build, and describe software systems, an all-purpose visual modelling language called UML is employed. Although software systems are a common use for UML, it is not the only one. Non-software simulation systems are also employed. As an example, consider the process flow at a manufacturing facility. Although UML is not a programming language, UML diagrams may be used to produce code in a variety of different languages. Object-oriented analysis and design are inextricably linked to UML. UML is currently an OMG standard after substantial standardization.

6 UML includes the following nine diagrams.

- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- Activity diagram
- State chart diagram
- Deployment diagram
- Component diagram

13 4.2.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the interactions between system components. A use case is a strategy for discovering, characterising, and arranging system needs in system analysis. The term "system" in this sense refers to something that is being developed or run, such as a Web site for the sale and delivery of mail-order products. 28 UML (Unified Modeling Language), a standard language for modelling actual things and systems, employs use case diagrams.

Modeling Language), a standard language for simulating real-world objects and systems. System goals may include gathering fundamental requirements, verifying a hardware design, testing and debugging a software product in development, generating an online help reference, or completing a customer service assignment.

➤ **Customer: -**

Customers can register on this platform. They may go through all of the well-organized options and book a service provider who suits their requirements.

➤ **Login: -**

By entering his user name and password, the consumer may use the cloud online purchasing programmer. The system will check to see if the login name and password match. If they do not match, the customer will receive an error notice.

➤ **Add to Service Cart: -**

When the consumer locates the services he requires, he adds them to the service carts. The information will be saved and tracked by the programme.

➤ **View Service Cart:**

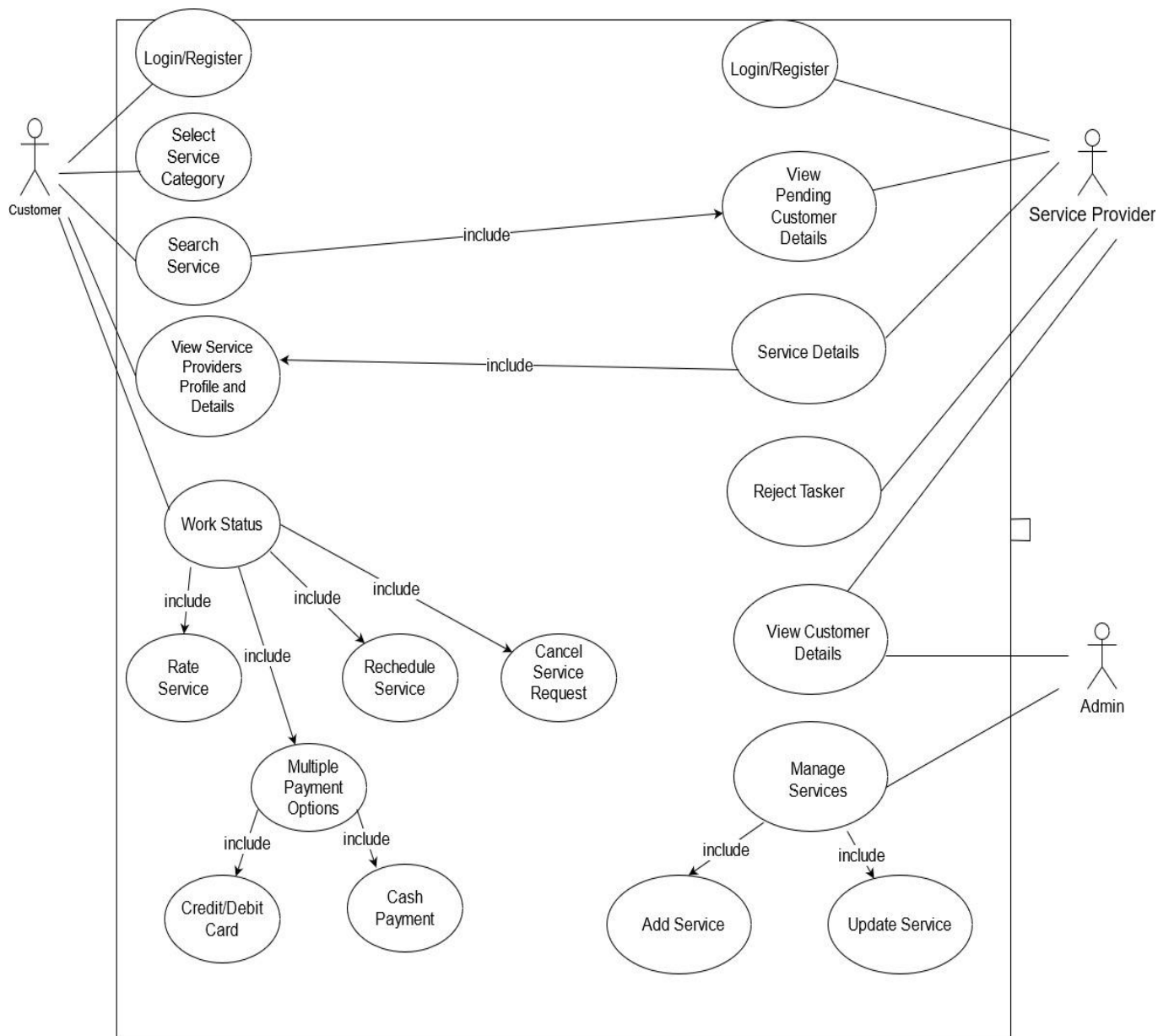
The consumer has the option of viewing the contents of the shopping cart. The programmer will return the items of the shopping cart to the consumer, together with the unit pricing and total price.

➤ **Service provider: -**

Customers can register on this platform. When a customer hires a service, they will receive a customer request. They can also choose to accept or reject the request.

➤ **Administrator: -**

Admin has the ability to add or delete services from particular categories.

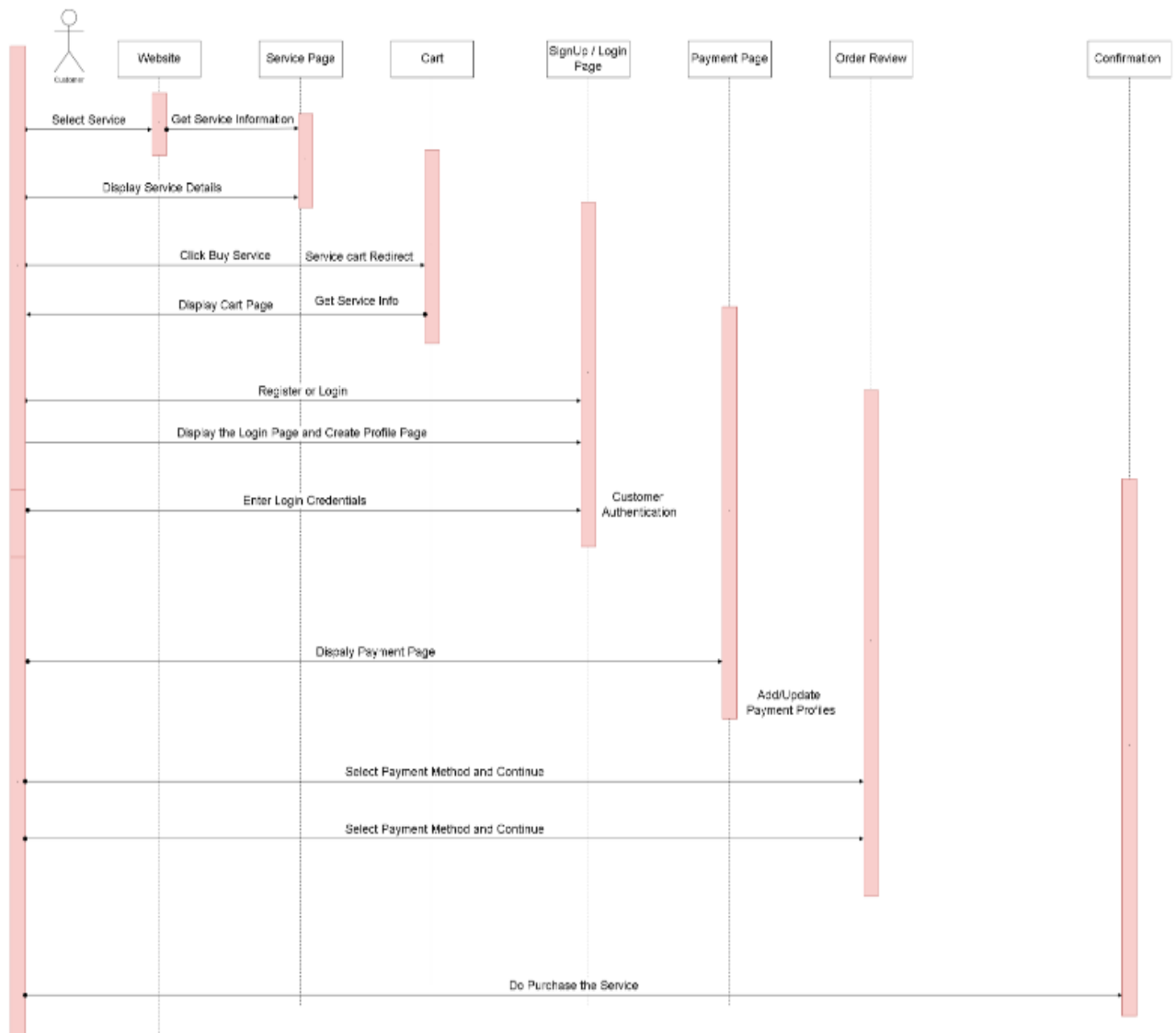


4.2.2³³ SEQUENCE DIAGRAM

A sequence diagram simply illustrates object interactions in the order in which they occur. A sequence diagram can also be described using the phrases event diagrams and event scenarios. Objects in a system are depicted as operating in a sequence diagram. Software engineers and businessmen frequently use these diagrams to explain and grasp requirements for both new and current systems.

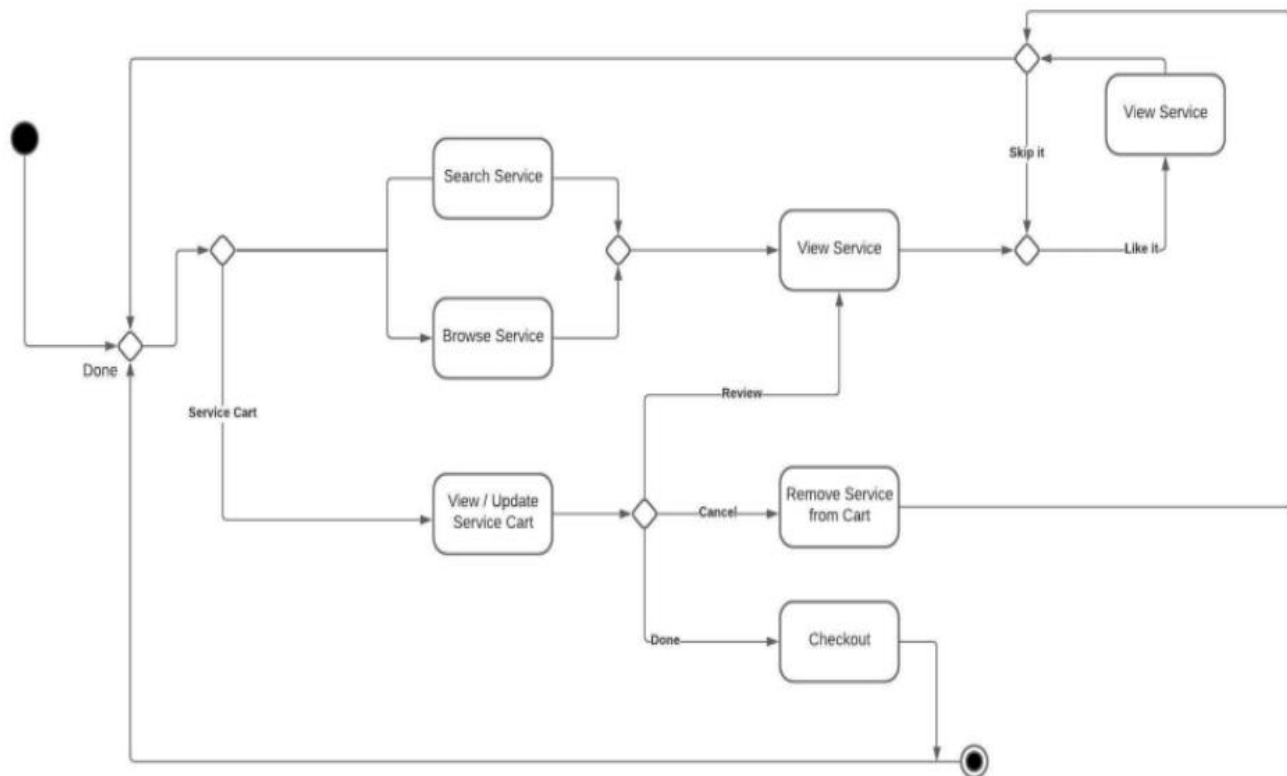
Sequence Diagram Notations –

- i. **Actors** – An actor in a UML diagram is a type of role that communicates with the system and its objects. It is important to keep in mind that an actor is never included in the UML diagram's representation of the system we are aiming to represent. Actors take on a range of roles, including those of external subjects and human users. In a UML diagram, an actor is represented by a stick person. Multiple actors can appear in a sequence diagram. understand system requirements for both new and existing systems
- ii. **Lifelines** – A identified piece in a sequence diagram that represents a specific participant is called a lifeline. Consequently, each episode is represented by a lifeline in a sequence diagram. The lifeline components of a sequence diagram are at the top.
- iii. **Guards** – Guards are used in the UML to model scenarios. We use them when we need to limit the flow of messages under the pretext of a condition being met. Guards are used by software developers to notify them of the constraints imposed by a system or a certain technique.



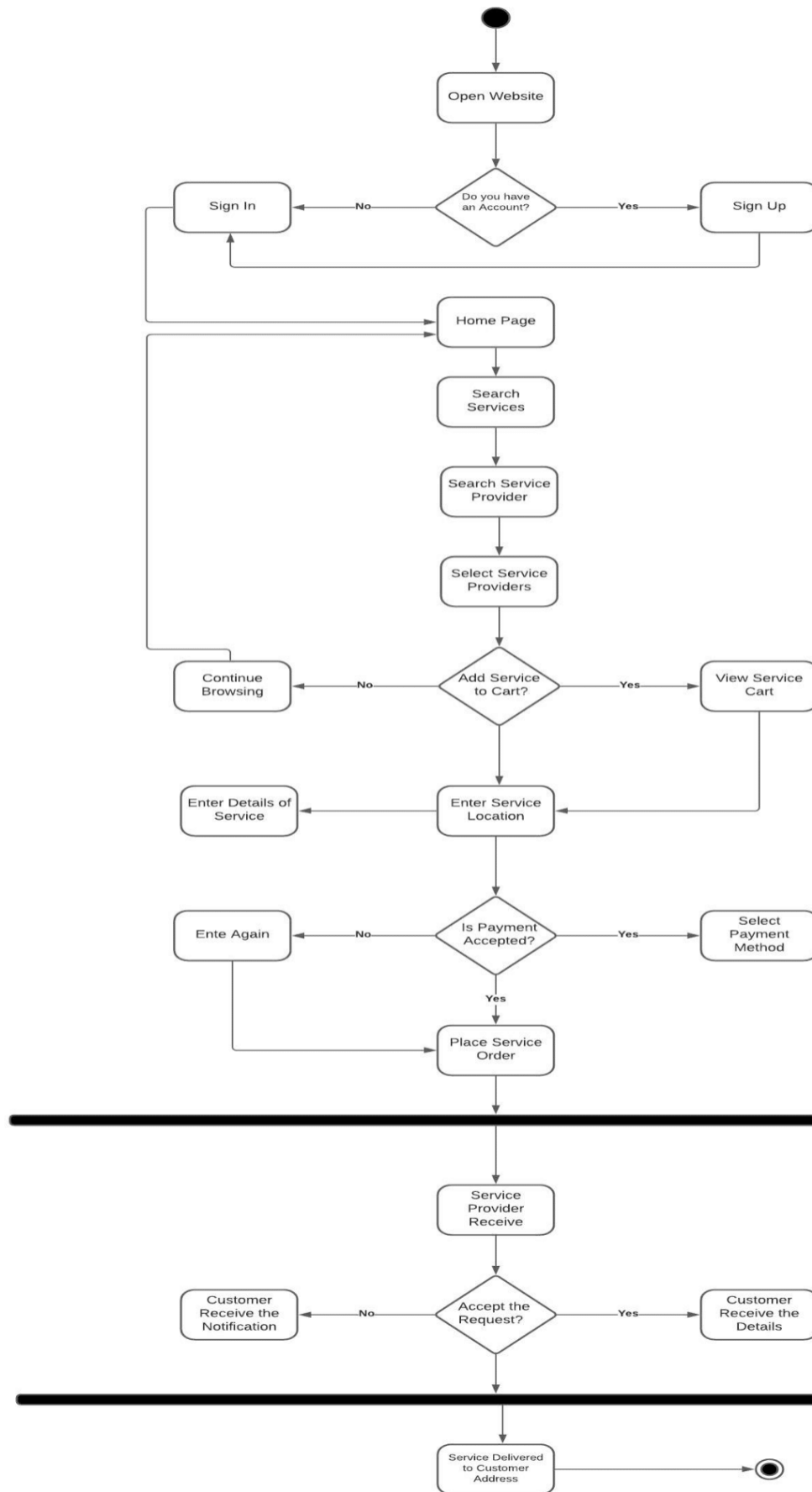
4.2.3 STATE CHART DIAGRAM

State diagrams are used to describe how software systems work. State machine diagrams in UML can be used to explain the behavior of a class, a subsystem, a package, or an entire system. A State Transition Diagram or a Statechart is another name for it. Statechart diagrams can be used to describe interactions or communication between a system and external entities. These diagrams depict the event-based system. An event manages the status of an item. State chart diagrams are used in application systems to show the many states of an entity.



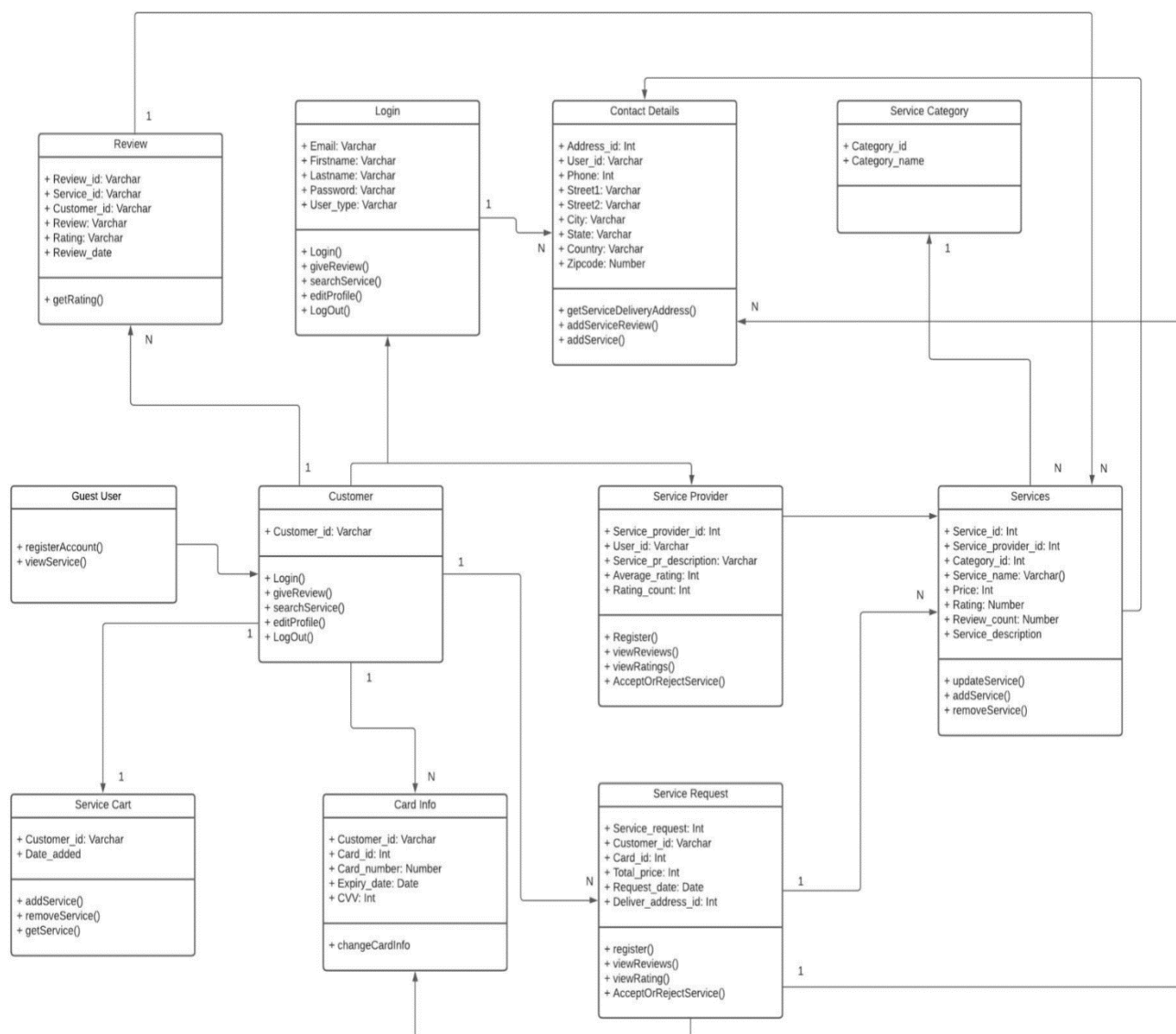
4.2.4 ACTIVITY DIAGRAM

Activity diagrams, at various levels of abstraction, explain how activities are organised to generate a service. Some operations must fulfil a typical event, especially when the operation is designed to perform numerous discrete tasks that require coordination. Another common requirement is ¹² how the events in a single use case interact with one another, particularly in use cases where activities may overlap and require coordination. It is also appropriate for simulating business processes by mimicking the coordination of a set of use cases.



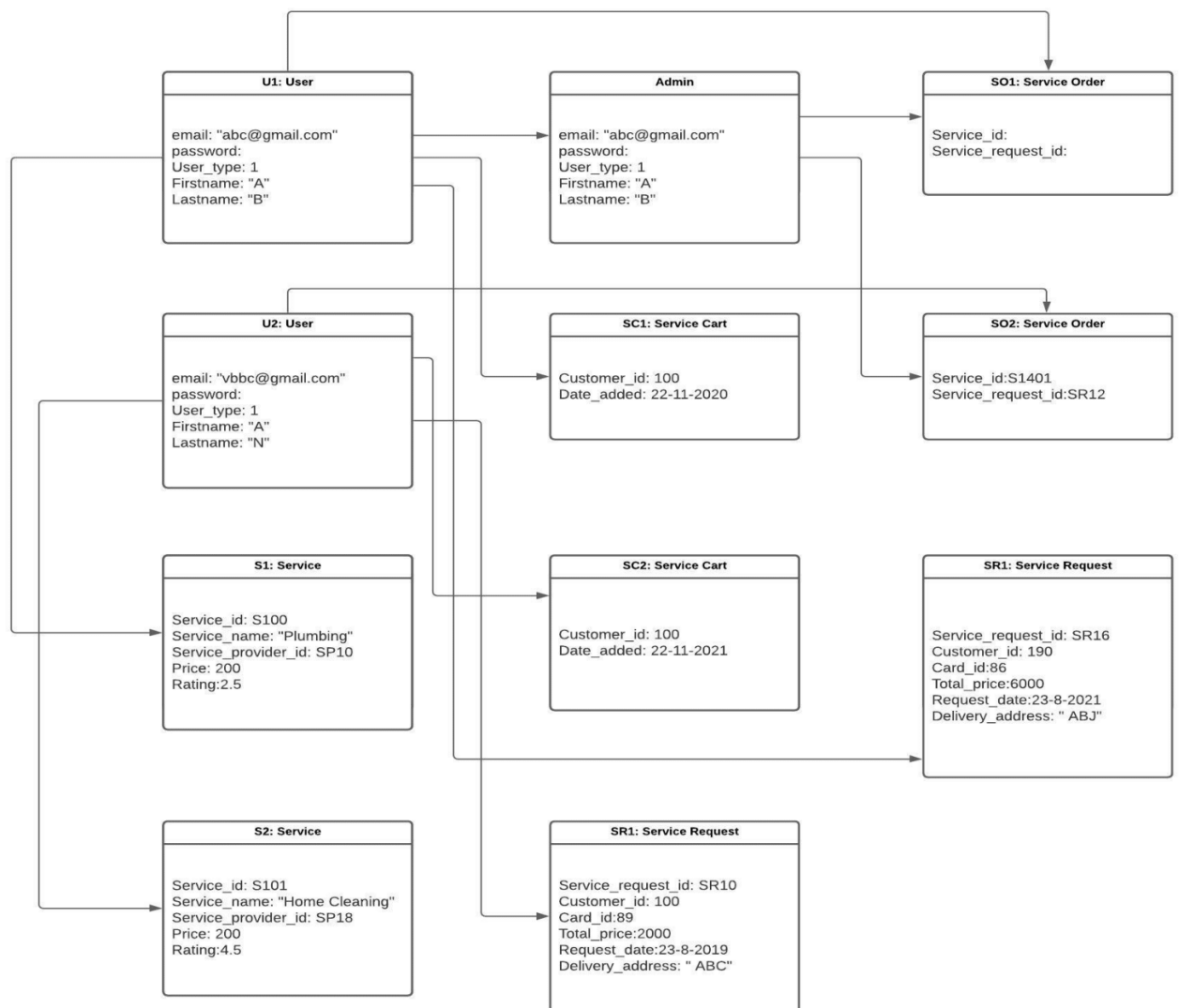
4.2.5 CLASS DIAGRAM

Static diagrams are class diagrams. It displays a static view of an application. A class diagram may be used to generate executable code for a software developer as well as to visualise, describe, and document various system components. A class diagram depicts a class's traits, behaviours, and system constraints. Because they are the only UML diagrams that can be easily transformed to object-oriented languages, class diagrams are often employed in the modelling of object-oriented systems. A class diagram graphically represents a collection of classes, interfaces, affiliations, collaborations, and limits. A structural diagram is another word for it.



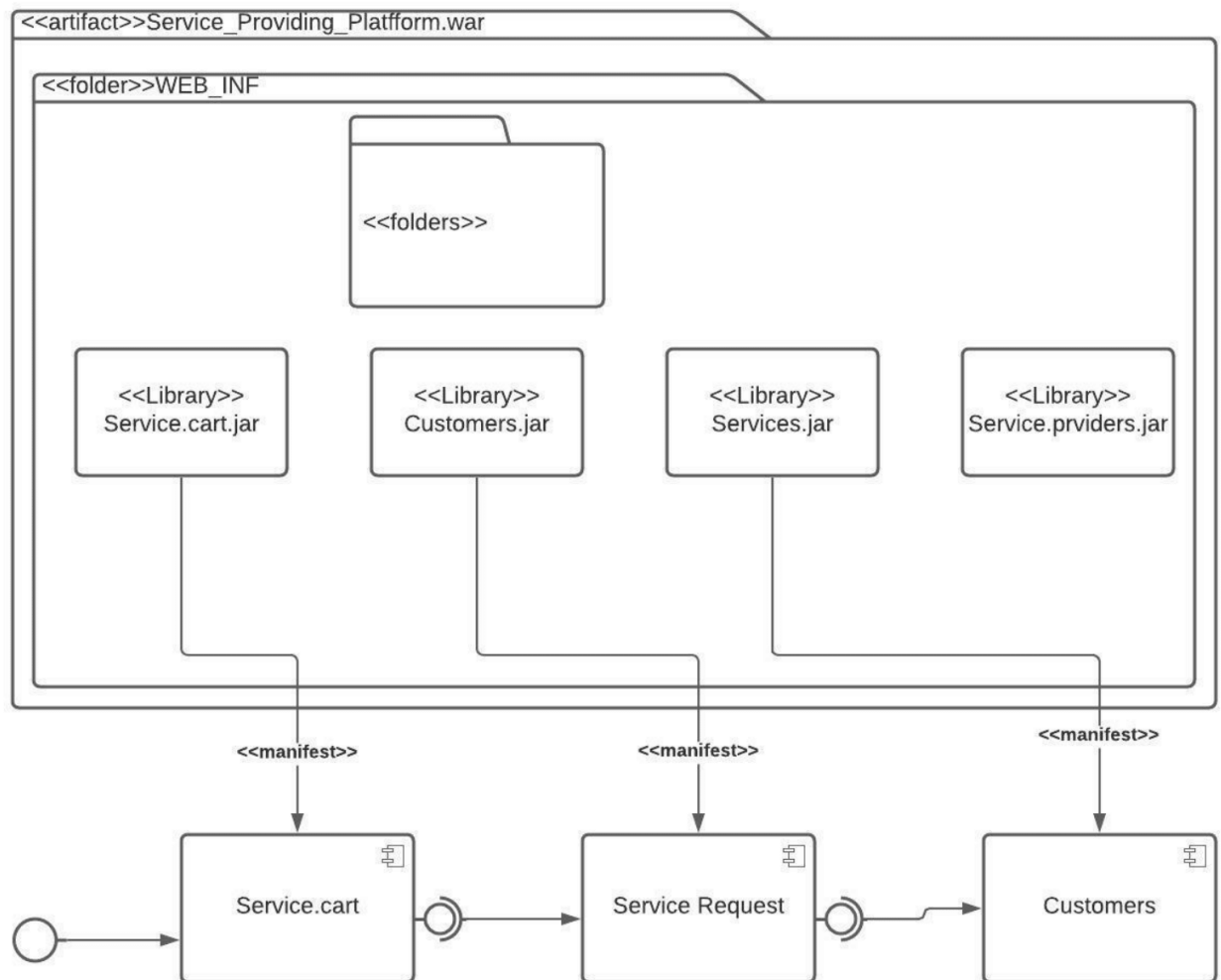
4.2.6 OBJECT DIAGRAM

Class diagrams are reliant on object diagrams since they are their relatives. A class diagram is represented visually by an object diagram. The fundamental concepts of class and object diagrams are the same. Object diagrams also reflect the static view of a system, however this static view represents a snapshot of the system at a certain time. Object diagrams are used to illustrate a set of things and their connections.



4.2.7 COMPONENT DIAGRAM

Component diagrams differ in their behavior and personality. Component diagrams are used to represent the physical components of a system. Executables, libraries, files, documents, and so on are all physically existent in a node. Component diagrams describe the organization and relationships between the components of a system. These designs are also used to construct systems that can be performed.



4.5 USER INTERFACE DESIGN

4.5.1-INPUT DESIGN

User Login

FF

Log into Fast Fix

E-mail

anandas123@gmail.com

Password

Register

Already have an account? [Sign In](#)

Activate Windows
Go to Settings to activate Windows.

User Registration

FF

Log into Fast Fix

E-mail

anandas123@gmail.com

Password

Register

Already have an account? [Sign In](#)

Activate Windows
Go to Settings to activate Windows.

4.5.2 OUTPUT DESIGN

User Login

FF

Log into Fast Fix

E-mail

anandas123@gmail.com

Password

•••••

Register

Already have an account? [Sign In](#)

Activate Windows
Go to Settings to activate Windows.

Service Provider Registration

Services Taskers **Fast Fix** Sign Out Become a Tasker 0

Share Your Skills

Be your own boss
Find local jobs that fit your skills and schedule. With FastFix, you have the freedom and support to be your own boss.

Tell us a little about yourself:
Enter your name

Enter your phone

Services you offer

Activate Windows
Go to Settings to activate Windows.

4.6. DATABASE DESIGN

A database is an organized means of storing information that allows users to quickly and readily access information. Any database's goal is to hold data, which must be maintained safe.

There are two stages to developing a database. The first stage is to gather user demands, followed by constructing a database that clearly fits those criteria. This is known as information level design, and it is done without the need of a database management system.

The second stage is modelling this data-level architecture for the particular database that will be used to create the aforementioned system. With regard to the DBMS characteristics that will be utilised, this is referred to as physical level design. System design and database design go hand in hand. The database's data configuration aims to accomplish the two main goals listed below.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A database is represented as a collection of relationships in a relational paradigm. Every connection works in the same way as a value table or record file. In formal relational model terminology, a row is referred to as a tuple, a column heading is referred to as an attribute, and the table is referred to as a relation. Each table in a relational database has its own name. In a story, a row stands for a collection of related principles.

Relations, Domains & Attributes

A connection is demonstrated via a table. Rows combine to form a table or a triple. A tuple is an ordered grouping of n objects. Columns are used to describe characteristics. Each table in the database has relationships made between them. The referential and entity links' integrity is therefore ensured. Domains are made up of atomic value collections. A common method to define a domain is to provide the data type from which the domain's data values are derived. It is also necessary to give the domain a name in order to better understand its values. A connection's values are atomic, hence it is impossible to separate them.

Relationships

- A key is used to build relationships in tables. The two most important keys are the primary key and the foreign key. These keys can be used to build Entity Integrity and Referential Integrity Relationships.
- According to entity integrity, no primary key can include null values.
- ² No Primary Key can have null values because of referential integrity.

4.6.2 Normalization

Data is organized as simply as possible to reduce the impact of any changes on data structures. The process of developing data structures to remove duplicate information and increase integrity is known as normalization. The act of breaking a large table into smaller ones and removing duplicate information is known as normalization. It also aids in the prevention of insertion, deletion, and update mistakes. In its most basic form, data modelling employs the ideas of keys and relationships. A key identifies each row in a table. The ¹⁴ two types of keys are primary keys and foreign keys. A primary key is a table element or component combination that is used to identify entries from the same table..

It refers to restoring things to their original state, as the name implies. The application developer strives to construct appropriate tables and columns with names that the user can readily relate with the data by normalizing the data. By deleting unnecessary data groups, normalization aids in the conservation of computer resources by minimizing data redundancy.

These include:

- ✓ Normalize the information.
- ✓ Give the tables and columns suitable names.
- ✓ Pick a suitable name for the data.

First Normal Form

An attribute's domain can only include atomic values, and each attribute in a tuple must have one value from that domain. The 1NF prohibits "relations inside relations" or "relations as feature values within tuples," to put it another way. 1NF only allows single atomic or indivisible values for attribute values. The data must first be converted into preliminary normal form as a first step. Data can be divided into a number of tables, each with a comparable set of data types, to accomplish this. Each table has either a Primary Key or a Foreign Key allocated to it depending on the requirements of the project. Each non-atomic property or nested relation generates a new relation. This eliminated any unnecessary data categories.

Second Normal Form

According to Second Normal Form, no non-key attribute should be functionally dependent on a specific section of the primary key in associations where the main key is made up of several qualities. This stage involves breaking down each partial key and assigning a new relation to each of its dependent characteristics. Keep the original primary key and any dependent attributes in a fully working connection. In situations where just a little amount of the key is needed, this step helps with data extraction. If and only if the connection's primary key satisfies the requirements of first normal form and all of the relation's non-primary key qualities are totally reliant on the primary key, the connection is said to be in second normal form.

Third Normal Form

According to Third Normal Form, relationships should not feature non-key attributes that are functionally determined by other non-key properties or a combination of non-key qualities. In other words, the primary key should not be transitively dependent. We analyse and link non-key qualities that operate as functional determinants of other non-key traits. At this point, everything that isn't completely dependent on the Primary Key is eliminated. If a relation is only in second normal form and its non-key qualities do not depend on one another, it is said to be in third normal form.

TABLE DESIGN

1. Table Name :- **Login**

Sl. No	Name	Type	Constraint
1	email	Varchar(255)	Primary Key
2	Firstname	Varchar(30)	Not Null
3	Lastname	Varchar(30)	Not Null
4	User_type	Varchar(255)	Not Null
5	Password	Varchar(255)	Not Null

2. Table Name :- **contact details**

Sl. No	Name	Type	Constraint
1	Address_id	Int	Primary Key
2	User_id	Varchar(255)	Foreign Key
3	Phone	Varchar(20)	Not Null
4	Street1	Varchar(255)	Not Null
5	Street2	Varchar(255)	Not Null
6	City	Varchar(50)	Not Null
7	State	Varchar(50)	Not Null
8	Zipcode	Number(5)	Not Null

3. Table Name :- **customers**

Sl. No	Name	Type	Constraint
1	Customer_id	Varchar(255)	Primary Key

4. Table Name :- **Card Info**

Sl. No	Name	Type	Constraint
1	Card_id	Integer	Primary Key
2	Customer_id	Varchar(255)	Foreign Key
3	Card_number	Number(16)	Not Null
4	Expiry_date	Date	Not Null
5	CVV	Number(3)	Not Null

5. Table Name :- **service provider**

Sl. No	Name	Type	Constraint
1	Service_provider_id	Varchar(255)	Primary Key
2	User_id	Varchar(255)	Foreign Key
3	Description	Varchar(255)	Not Null
4	Average_rating	Number(20)	Default
5	Rating_count	Number	Default 0

6. Table Name :- **Category**

Sl. No	Name	Type	Constraint
1	Category_id	Integer	Primary Key
	Category_name	Varchar(255)	Not Null

7. Table Name :- **Services**

Sl. No	Name	Type	Constraint
1	Service_id	Integer	Primary Key
2	Category_id	Integer	Foreign Key
3	Service_provider_id	Varchar(255)	Foreign Key
4	Service_name	Varchar(255)	Not Null
5	Price	Number(10, 2)	Not Null
6	Rating	Number(2, 1)	Not Null
7	Service_description	Varchar(255)	Not Null

8. Table Name :- **Service Images**

Sl. No	Name	Type	Constraint
1	Service_id	Integer	Foreign Key
2	Image_url	Varchar(255)	Not Null

9. Table Name :- **Service Cart**

Sl. No	Name	Type	Constraint
1	Customer_id	Varchar(255)	Foreign Key

10. Table Name :- Service Cart_2

Sl. No	Name	Type	Constraint
1	Service_id	Integer	Foreign Key
2	Customer_id	Varchar(255)	Foreign Key

11. Table Name :- Service Request

Sl. No	Name	Type	Constraint
1	Service_request_id	Integer	Primary Key
2	Customer_id	Varchar(255)	Foreign Key
3	Delivery_address_id	Integer	Foreign Key
4	Delivery_date	Date	Not Null
5	Total_price	Number(10, 2)	Not Null
6	Request_date	Date	Not Null

12. Table Name :- Order Service

Sl. No	Name	Type	Constraint
1	Service_request_id	Integer	Foreign Key
2	Service_id	Integer	Foreign Key

13. Table Name :- Review

Sl. No	Name	Type	Constraint
1	Review_id	Integer	Primary Key
2	Service_id	Integer	Foreign Key
3	Customer_id	Varchar(255)	Foreign Key
4	Review	Varchar(255)	Not Null
5	Rating	Number(2, 1)	Not Null
6	Review_date	Date	Not Null

14. Table Name :- Review Image

Sl. No	Name	Type	Constraint
1	Review_id	Integer	Primary Key
2	Image_url	Varchar(255)	Not Null

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

A programmer's-controlled execution to see if it works as expected. The terms software testing, verification, and validation are sometimes used interchangeably. Validation is the process of verifying items, particularly software, for consistency and conformity to an underlying specification. Other techniques of verification outside of software testing include reviews, analyses, inspections, and walkthroughs. Verifying that what was said was exactly what the user desired is the process of validation..

Static and dynamic analysis are two more methodologies that are widely used in software testing. Static analysis examines the source code for faults and errors without running the product. Dynamic analysis looks at the functionality of software in use and provides data such as execution traces, time profiles, and test coverage.

Testing is a combination of tasks that may be meticulously planned and conducted ahead of time. After the modules have been assessed, the entire computer-based system is combined. Everything requires testing, and meeting system testing objectives is critical. There are several possible exam objectives. The process of running software to hunt for faults is known as testing.

- A test case that has a strong possibility of spotting an undiscovered flaw is a good one.
- A test that finds an unidentified error is successful.

If the testing is done correctly and in accordance with the aforementioned goals, it will detect code issues. Furthermore, testing confirms that the software features appear to work as planned and that the performance benchmarks appear to have been fulfilled.

There are three methods for programmer testing.

- For accuracy
- For effective implementation
- For the difficulty of computing

5.2 TEST PLAN

The tasks that will be carried out in order to complete various testing procedures make up a test plan. The Test Plan serves as a structure for the following tasks. Computer programmers provide data structures, programmers, and documentation. The test plan has to specify each regression test. Following are the testing levels:

- ❖ ² Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

Unit testing concentrates on the software module or component, which is a crucial component of software design. The component level design description is used to facilitate the exploration of important control paths within the module's boundary. The depth and relative complexity of unit testing are evaluated. White-box unit testing enables for the testing of several components at the same time. The modular interface is evaluated to ensure that data enters and exits the software ² unit under test efficiently. The local data structure is reviewed at each step of an algorithm's execution to confirm that data that has been temporarily preserved keeps its integrity. ¹¹ Boundary conditions are tested to ensure that each statement in a module has been executed at least once. All paths for error handling are offered at the end.

Before proceeding with any further testing, data flow over a module interface must be tested. All other testing is made useless if data is not correctly input and exited. During the unit test, it is required to do systematic testing of execution pathways. When an error occurs, error circumstances should be anticipated, and error handling routes should be developed to reroute or cleanly terminate activities. The unit testing technique is followed by boundary testing. When software approaches its limits, it usually fails.

²¹ 5.2.1 Integration Testing

Integration testing is a systematic technique for establishing software architecture while also identifying interface issues. The aims are to use unit-tested components and to create a design-governed programmer structure. Throughout the programme, thorough testing is performed. Correction is difficult since the program's broad breadth makes it difficult to determine the reasons. After these defects are fixed, other ones appear, and the cycle appears to continue indefinitely. Following system unit testing, all modules were integrated to examine for interface incompatibilities. Furthermore, several programmer architectures were phased out, leaving only one programmer structure.

5.2.2 Validation Testing or System Testing

The testing has reached its conclusion. Every form, line of code, module, and class module in the system was tested. Black box testing and system testing are other names for this type

of testing.

The primary focus of the Black Box testing technique is the programmer's functional requirements. A software engineer can generate sets of input conditions that fully meet all of a program's criteria via black box testing. Black box testing searches for problems in the following areas: correct or missing functionality, interface errors, data structure or external data access errors, performance flaws, initialization errors, and termination faults.

5.2.3 Output Testing or User Acceptance Testing

The system under consideration should meet the needs of the business and has undergone user evaluation. The software should stay in touch with the upcoming system and user while developing and making necessary modifications. The following methods are used to achieve this:

- Input Screen Designs,
- Output Screen Designs,

The testing indicated above is carried out using many types of test data. In system testing, the prepared test data is essential. Utilizing the provided test data, the system under investigation is tested. Using the aforementioned testing methods, test data mistakes are found and fixed during system testing, and the fixes are also recorded for later use.

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The process of turning a theoretical design into a working system is known as implementation. It is the most crucial phase in creating a successful new system because it inspires confidence in users that the system will function, be efficient, and accurate. Documentation and user training are its main priorities. Conversion often happens either concurrently with or following user instruction. The process of converting a new, enhanced system design into an operational one is known as implementation.

The user department is currently in charge of most of the work, most of the interruption, and most of the impact on the current system. If the deployment is not well planned and managed, chaos and confusion may ensue.

Implementation includes all actions necessary to switch from the current, antiquated system to the new. The new system may be entirely different, take the place of an existing human or automated system, or make minor adjustments to the latter. In order to create a reliable system that satisfies the demands of the business, proper implementation is necessary. The process of putting the created system into practical use is referred to as system implementation. This includes all steps necessary to switch from the old to the new system. Only after thorough testing and verification that the system complies with the criteria can it be installed. The system's workers assess the system's viability.

The following tasks are included in the implementation state:

- thorough planning.
- examination of the system and its limitations.

6.2 IMPLEMENTATION PROCEDURES

Software implementation is the process of fully implementing a package in a real environment such that it serves the desired functions and operates as intended. Many organizations will hire someone to design the software but not operate it themselves. At first, people are skeptical about the programmer, but we must ensure that skepticism does not intensify.

- The new system's advantages must be known to the active user.
- Their confidence in the software is built up.

Before using the system, the user should be informed that the server programme has to be running on the server in order to access the results. If the server object is not active and

functioning on the server, the actual procedure will not happen.

2 6.2.1 User Training

User training is intended to get individuals ready for system conversion and testing. Participants must feel secure in their involvement in the new system for the goals and benefits anticipated from a computer-based system to be realised. The need for training increases as the system gets trickier. The user is given instructions on how and where to enter data, respond to error messages, query the database, and call procedures that produce reports and carry out other crucial tasks.

2 6.2.2 Training on the Application Software

The user must be trained on the new application software after getting the fundamental training in computer awareness. The basic principles of how the new system should be used, including screen flow, screen design, the kinds of help that can be found there, the kinds of mistakes that can happen while entering data, the related validation checks at each input, and methods for repairing inaccurate data, will be detailed. The information required by the particular user or group to operate the system or a component of the system should be covered while offering programmer training on the application. Depending on the user group and organisational level, this training may vary.

18

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The operational technology of the current system is antiquated, and everyday technologies like the internet and digital money are not in use. Under the proposed system, customers would be able to reserve services online and see all data.

7.2 FUTURE SCOPE

The future of the online services sector is incredibly promising. The internet services sector has possibilities that no other industrial vertical can equal when taking into account the number of dwelling units throughout the world and the continuously growing population. Finding competent individuals to complete modest jobs at home is a constant struggle for people, and this problem is not localized to any one geographic location. Although not everyone is aware of it, you may connect to these services and do a number of tasks linked to them using apps that are already accessible. And it's becoming more popular, enabling individuals to complete these tasks in a disruptive manner with remarkable ease and comfort.

You think of an agency and get in touch with them whenever you need any kind of service for your property or the area surrounding it. Alternately, you might request a recommendation from a friend. You may frequently need a range of services, such as relocating to a new house, replenishing your LPG tank, or mending water pipes. Additionally, it ought to be fundamentally altering the way we think about services given the rising relevance and popularity of on-demand home services across several utility verticals.

CHAPTER 8

BIBLIOGRAPHY

REFERENCES:

- ²⁶ Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “*Software engineering: a precise approach*”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- ³⁰ IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

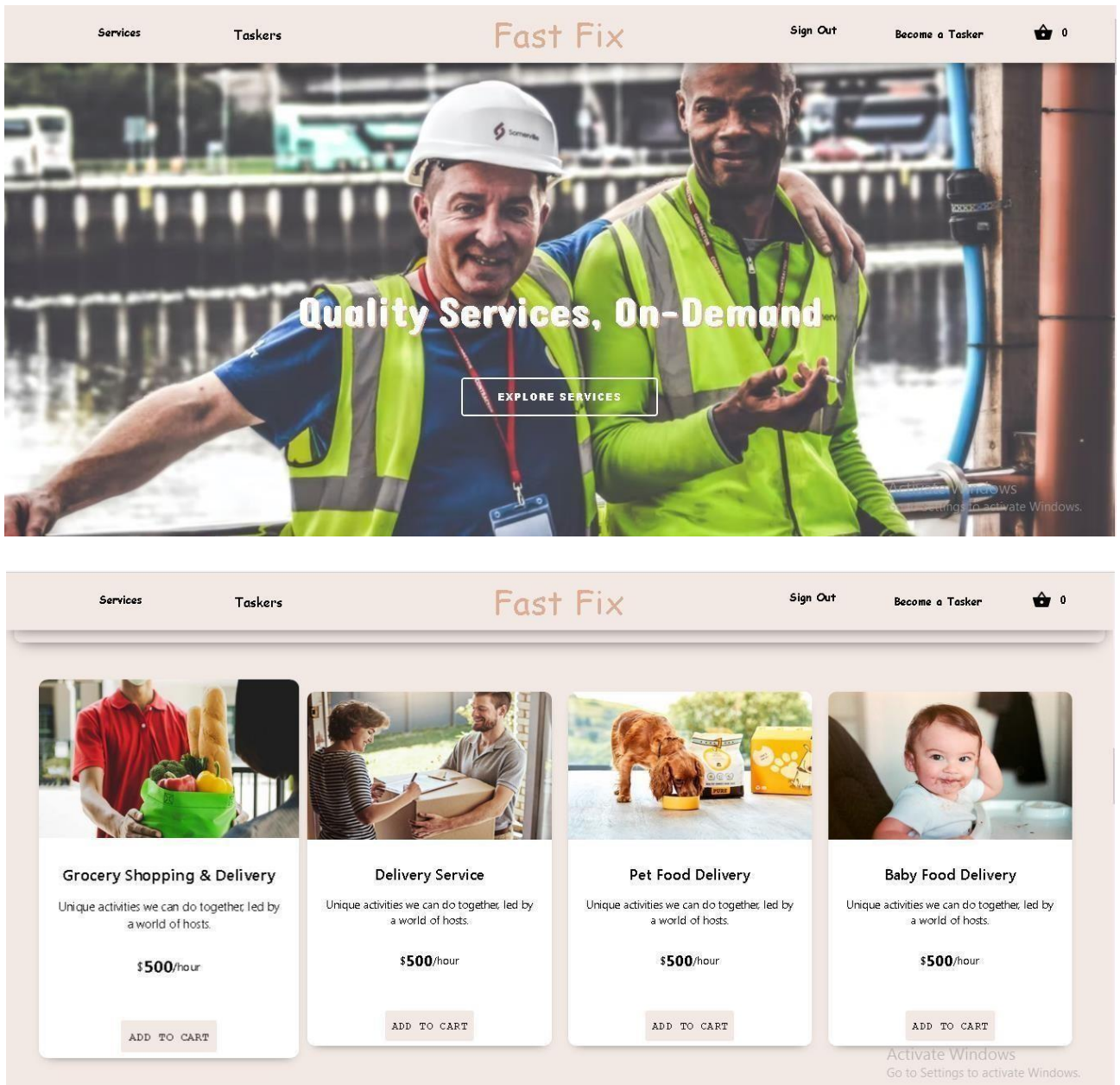
- www.w3schools.com
- www.reactjs.org
- www.javatpoint.com
-

CHAPTER 9

APPENDIX

9.1 Screen Shots

Home page



Service Page

Services

Taskers

Fast Fix

Sign Out

Become a Tasker

0

Cleaning

Furniture Assembly

Handyman

Holidays


Mounting & Installation

Moving Services

Office Services

Shopping + Delivery

Search by service!



\$50/hour

♡

Baby Food Delivery


—

Need someone to deliver baby food and toddler meals straight to your door? Hire a Tasker for same day delivery.

★ 4.73

ADD TO CART

VIEW



\$10/hour

♡

Breakfast Delivery


—

Start your day off right with a hearty breakfast. Hire a Tasker to deliver breakfast straight to your door.

★ 4.73

ADD TO CART

VIEW



\$65/hour

♡

Car Washing

—

Hire a Tasker to get your car looking like new!


★ 4.73

ADD TO CART

VIEW

Activate Windows

Go to Settings to activate Windows.



Furniture Assembly


—

Have a Tasker put together your furniture for you.

★ 4.73

ADD TO CART

VIEW



Grocery Delivery

—

Give a local Tasker your shopping list and they'll stock your pantry with the groceries you need for the week. Same day delivery available.

★ 4.73

ADD TO CART

VIEW

Service Providers

Services

Bookings


Fast Fix

Sign Out


Become a Tasker

0

Search by Location!



Ansona N Roy
—
Featured Skills
Baby Bathing ₹140/hour



Amala Mariyat
—
Featured Skills
Baby Bathing ₹120/hour

Service Requests

Services


Taskers

Fast Fix

Sign Out

Become a Tasker

0



MY TASKS
Ansona N Roy

GO ONLINE


GO OFFLINE

RECEIVED TASKS

RESCHEDULED TASKS


Anand AS
Ponkunnam
2022-07-19
Thu, 21 Jul 2022 15:16:44 GMT(requested date)
14:30
Attupurathu House


MORE DETAILS..



[Services](#)[Taskers](#)

Fast Fix

[Sign Out](#)[Become a Tasker](#) 0



MY TASKS
Ansona N Roy
Baby Bathing


Anand AS
Description
Tell us the details of your task
Date
Rescheduled
2022-08-01
Time
14:30
RESCHEDULED


Address
Attapurathu House, KVMS Hospital, Ponkunnam,
Ponkunnam, Kerala - 686506
7025942442
Do you want to proceed? YES NO

Service History

[Services](#)[Bookings](#)




Fast Fix

[Sign Out](#)[Become a Tasker](#) 0

**Anand AS**

My Bookings
ACCEPTED **REJECTED**

My Bookings

Status	Tasker	Profile	Service	Date	Action
Booking Accepted	Arya Sasi		Baby Food Delivery	2022-07-12	BOOKING STATUS
Booking Accepted	Ameena Sherin		Delivery Service	2022-07-06	BOOKING STATUS
Booking Accepted	Arya Sasi		Baby Food Delivery	2022-08-01	BOOKING STATUS

Booking Status Page


Services

Taskers

Fast Fix

Sign Out

Become a Tasker

 0

Baby Food Delivery

Arya Sasi

✓ **Booking Accepted**


ORDER CANCEL

Let us know your cancellation reason

Describe here!

CANCEL

RESCHEDULE

07 / 12 / 2022 

SUBMIT

Your order successfully rescheduled!

Contact Service Provider

Booking Date & Time

2022-07-12


Order rescheduled

Pricing

Baby Bathing

180

Hours Worked

5 

Total Amount

₹ 900

FINISH TASK

GO TO CHECKOUT >

● 19% Overall Similarity

Top sources found in the following databases:

- 19% Internet database
- 8% Publications database
- Crossref database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	G.S. Ajith, M.G. Girija, Jinson Devis. "Poly House Environment Monitori...	6%
	Crossref	
2	issuu.com	3%
	Internet	
3	repository.dinamika.ac.id	1%
	Internet	
4	coursehero.com	1%
	Internet	
5	dspace.cusat.ac.in	1%
	Internet	
6	researchpublish.com	<1%
	Internet	
7	geeksforgeeks.org	<1%
	Internet	
8	seminarprojects.com	<1%
	Internet	
9	javatpoint.com	<1%
	Internet	

10	Frederick E. Petry. "Fuzzy Databases", Springer Science and Business ...	<1%
	Crossref	
11	engineeringminiprojects.com	<1%
	Internet	
12	Ilona Veitaite, Audrius Lopata. "Chapter 12 Knowledge-Based UML Dyn...	<1%
	Crossref	
13	repository.president.ac.id	<1%
	Internet	
14	windowsitlibrary.com	<1%
	Internet	
15	dartmouth.edu	<1%
	Internet	
16	George E. Westlake. "Establishing and maintaining the data-base requi...	<1%
	Crossref	
17	definehierarchy.blogspot.com	<1%
	Internet	
18	idr.mnit.ac.in	<1%
	Internet	
19	essentialsql.com	<1%
	Internet	
20	kamalnit.blogspot.com	<1%
	Internet	
21	mafiadoc.com	<1%
	Internet	

22	trickideas.com	Internet	<1%
23	eprints.qut.edu.au	Internet	<1%
24	seminarprojects.com	Internet	<1%
25	scribd.com	Internet	<1%
26	tce.edu	Internet	<1%
27	Carlos A. Silva, Renato Toasa, Juan Guevara, H. David Martinez, Javier ...	Crossref	<1%
28	bipin0007.blogspot.com	Internet	<1%
29	ce.sharif.edu	Internet	<1%
30	idus.us.es	Internet	<1%
31	K. Anand, Vaishak Nambiathody. "Shape memory polymers (SMP's) in ...	Crossref	<1%
32	id.123dok.com	Internet	<1%
33	cours-gratuit.com	Internet	<1%

34

herc.research.va.gov

Internet

<1%