PAPER NAME	AUTHOR
Main_Project_Report___(FitZone)1.pdf	Swarna Mol K K
WORD COUNT	CHARACTER COUNT
9203 Words	48132 Characters
PAGE COUNT	FILE SIZE
51 Pages	1.2MB
SUBMISSION DATE	REPORT DATE
Jul 22, 2022 2:29 PM GMT+5:30	Jul 22, 2022 2:30 PM GMT+5:30

● **20% Overall Similarity**

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

- 15% Internet database
- 6% Publications database
- Crossref database

● **Excluded from Similarity Report**

- Small Matches (Less then 15 words)

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Health is wealth, as we all know. A spacious apartment, a luxury automobile, or a doctorate are not necessary without health. The first thing we need to remember is to stay healthy. Because our feelings often determine our attitude. Having energy to do anything comes from being healthy and fit. A healthy, stress-free life depends heavily on physical fitness. Diet, exercise, and sleep all contribute to physical fitness. Every person should be intelligent with regard to these three fundamentals for a healthy life because they each have their own significance in their daily lives. For those who own a gym business, our proposed "FitZone" is solution. An adequate study of the project's initial phase's primary challenges for gym owners is looked at. An in-depth analysis was carefully done on how to create a massive registration system without error as well as various functions for various user types depending on their privilege. A system that can easily handle all the necessary and minute details and provide sufficient database security in accordance with the user is required by the gym management.

1.2 PROJECT SPECIFICATION

The proposed system is made it easier to manage a fitness center or gym. It helps the users to get a quick access to the overall events in a gym. The main objective of this project for the better performance of gym and to manage the services through computerized system.

There are 4 modules in the system. As follows:

1. Admin Module

Admin have a login into this system. He/she has the overall control of the system. Admin can add and manage trainer, physician. And can send their username and password to their E-Mail. Admin is able to view all registered user details and their profiles, also view & download the physician’s license proof. And he/she can add, view & update the schedules of trainers and physicians. Admin can view(can choose search option) member’s profiles, their trainers, and their fee status. And can view & download active users list.

2. User Module

User can register, view & edit their profiles, select suitable schedule and join under the available trainers through viewing their profile. View physician schedule and book appointments based on the available slots. View their own joined schedules. Do secure online payment for paying monthly fees based on the number of schedules they joined, view the payment history and can download the receipt.

3. Trainer Module

Trainers get the username & password through the email & can login in this site. He/she can also see their own training schedules, can view & update their profile. And can view the users under his/her training schedules.

4. Physician Module

Physician get the username & password through the email & can login in this site. They can also see their own schedules. And can see the appointments from users, view & download total appointments.

CHAPTER 2

SYSTEM STUDY

2.1 INTRODUCTION

System analysis is the process of acquiring and analysing data, diagnosing issues, and using the data to suggest system changes. It is a problem-solving process that necessitates intensive collaboration between system developers and users. Any system development process should include a phase of system analysis or study. The system is examined and investigated in minute detail. The system analyst assumes the role of the interrogator and focuses intently on how the current system functions.The input to the system is identified and the system as a whole is viewed. The various processes are linked to the organisations' outputs. Being aware of the issue, identifying the pertinent and important variables, analyzing and synthesizing the different elements, and selecting the best or, at the very least, adequate course of action are all part of system analysis.

The procedure needs to be thoroughly studied using a variety of methodologies, including questionnaires and interviews. To draw a conclusion, the information gathered from these sources needs to be carefully examined. An comprehension of how the system works is the result. The existing system is what we refer to as. Now, the current system is carefully examined, and issue areas are found. The designer now acts as a problem-solver and works to resolve the issues the business is having. The solutions are offered as suggestions. The proposal is then analytically compared to the current system, and the best one is chosen. The user is shown the suggestion in order to get their approval. On user request, the proposal is examined, and necessary adjustments are made. As soon as the user is satisfied with the proposal, this loop breaks.

The process of doing a preliminary investigation include acquiring and analysing data in order to use it for future system research. Preliminary research is a problem-solving process that necessitates close coordination between system developers and users. It conducts a range of feasibility studies. These investigations provide an approximate estimate of the system activities, which can be used to determine the tactics to be used for an efficient system research and analysis.

2.2 EXISTING SYSTEM

Existing system is not a fully automated system. The system currently in use is referred to as an existing system. The gym is manually operated. The existing method requires a lot of paperwork, which makes it time-consuming and expensive. Handling the system manually was an extremely challenging task. However, modern computerization has made job easier. It is necessary to modify the existing system in order to include additional information and make the system efficient, flexible and secure.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Less convenient in managing project details including plan approval, less transparency, no standardized packages.
- Project often delayed with no progress visibility.
- The procedure is time-consuming and involved a lot of paperwork.
- Everything is done on paper, which is very prone to damage and needs a lot of storage space and security.
- Required purchasing products more regularly than through an internet system, such as paper and pens.
- likely to be mistaken.
- Insufficient room for keeping handwritten documents.
- Demand additional labour force and physical labour.
- Because information is not accessible to both clients and staff internationally, there is a location restriction.
- Human effort is needed.

2.4 PROPOSED SYSTEM

The suggested system is intended to address every drawback of the current system.
After the planning and analysis stage of the gym management system is complete. Following that, it was necessary to turn the data needed to build the system into a structural plan that would be used as a guide to build the actual operating system. It is a phase where the majority of dangers and faults are revealed, therefore it is best to take care of this from the beginning.

Ignoring the risk or error is not an option because later it could take a greater shape of itself as this is a fully-fledged system that will serve as the backbone of the entire management of the gym. Therefore, it is preferable to reduce the issues that the organization's staff and administrative staff encounter.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- ✓ The suggested system is highly secure because each user's unique username and password are required for login, giving each user a unique perspective on the customer information.
- ✓ It offers a variety of specific criteria in each window where the client is looking for a better and more expedient resolution.
- ✓ It keeps track of all criteria and transactions in a report.
- ✓ Manages member information independently from all bill information while taking the needs of the gym into account.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

This stage involves assessing the project's potential and presenting a business proposal with a very basic project plan and some cost forecasts. The usefulness of the proposed system must be examined during system analysis. This will ensure that the offered remedy won't put a strain on the company. The feasibility analysis requires a thorough understanding of the primary system requirements.

The following three factors are crucial to the feasibility analysis:

3.1.1 Economical Feasibility

This study is being conducted to determine the system's potential financial impact on the organization. The company's resources for research and development of the technology are limited. There must be proof to back up the costs. Because most of the technology was in the public domain, the developed system was able to finish within budget. Only the things that can be customized must be purchased.

3.1.2 Technical Feasibility

This study is being conducted to assess the system's technical feasibility or requirements in terms of technology. Any system developed must not significantly tax the existing technical resources. The client will therefore have high expectations. The system must be easy to install, needing either no changes at all or just a small number.

3.1.3 Behavioral Feasibility

The study's objective is to ascertain the degree of user acceptance of the system. This covers the guidance required for the user to use the system efficiently. The user should not feel intimidated by the system; rather, they should see it as a need. The methods utilised to inform and familiarize users with the system are the sole elements that have an impact on how accepting they are of it. His confidence needs to grow because he is the system's primary user, and constructive criticism is welcomed.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

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

3.2.2 Software Specification

Front End	-	HTML, CSS
Backend	-	MYSQL
Client on PC	-	Windows 7 and above.
Technologies used	-	JS, HTML5, AJAX, J Query, PHP, CSS

3.3 SOFTWARE DESCRIPTION

3.3.1 PHP

PHP is a server-side scripting language used for both web development and general-purpose programming. PHP is now used by more than 244 million websites and 2.1 million web servers. The PHP group presently creates the reference implementation of PHP, which was first developed by Rasmus Ledorf in 1995. PHP, a recursive acronym that formerly stood for personal home page, now stands for PHP: Hypertext Preprocessor. The PHP processor module on a web server translates PHP code to create the final web page. Instead of accessing an external file to handle data, PHP commands can be directly put into an HTML source file. The GNU General Public License is incompatible with PHP because it has evolved to incorporate a command-line interface feature and can be used independently due to restrictions on the usage of the term PHP (GPL). The majority of web servers permit the free installation of PHP, which is also accessible as a standalone shell on almost all platforms and operating systems.

3.3.2 MySQL

The most well-known Open Source SQL database management system, MySQL, was developed, distributed, and supported by Oracle Corporation. The MySQL website has the most latest information regarding its software.

- **MySQL is a database management system.**

  A database is a regular collection of data. Anything might be it, including a simple grocery list, a photo gallery, or the massive amount of data in a company network. Data saved in a computer database can only be added to, accessed, and processed via a database management system like MySQL Server. Because computers are so good at processing massive amounts of data, database management systems—whether employed as standalone programmes or as a component of other applications—are crucial to computing.

- **MySQL databases are relational.**

  A relational database stores the data in distinct tables rather than consolidating it into one enormous warehouse. The database structures are stored physically in fast files. A flexible programming environment is provided by the logical model, which includes objects like databases, tables, views, rows, and columns. You might create rules to control the relationships between different data fields, such as one-to-one, one-to-many, unique, compulsory or optional, and "pointers" between other tables. Since a well-designed database upholds these constraints, your application won't ever run into inconsistent, duplicate, orphan, out-of-date, or missing data. Structured Query Language (SQL) is what MySQL stands for. SQL is the most widely used standard language for database access. You might explicitly enter SQL (for instance, to generate reports), embed SQL statements into other languages' code, or use a language-specific API that hides the SQL syntax depending on your programming environment. SQL is specified by the ANSI/ISO SQL Standard. The SQL standard has gone through several modifications since its creation in 1986. In this document, the terms "SQL92," "SQL: 1999," and "SQL: 2003" all refer to the 1992 version of the standard, respectively. "The SQL standard" refers to the SQL Standard as it is at any given time.

- **MySQL software is Open Source.**

  Considering that the software is open source, anyone can use and alter it. Anyone can access and use the MySQL software for free online. You are allowed to look over the source code and make any necessary changes. The MySQL software uses the GPL (GNU General Public License) to define what you are and are not permitted to do with the software in specific situations. If the GPL makes you uncomfortable or if you need to incorporate MySQL code into a for-profit application, you can buy a commercially licenced version from us. Check out the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

  If that's what you want, you ought to try it. Additionally to your other applications, web servers, and software, MySQL Server can run without a hitch on a desktop or laptop with little to no upkeep. If you dedicate an entire workstation to MySQL, you can change the settings to make full use of the RAM, CPU, and I/O capabilities.

- **MySQL Server works in client/server or embedded systems.**

  A client/server system, the MySQL Database Software features a multi-threaded SQL server that supports a number of client programmes and libraries, management tools, and a wide range of application programming interfaces (APIs). Additionally, we offer MySQL Server as a built-in multi-threaded library that you can incorporate into your software to produce a standalone offering that is more compact, quick, and easy to use.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step in the development of any engineered system or product. Design is a creative process. A good design is the key to a system that works effectively. The process of specifying a system or a process in sufficient depth to enable its physical implementation using various approaches and concepts is known as "design." The process of applying several approaches and concepts to specify a device, a process, or a system in sufficient detail to enable its physical reality is one way to put it. Software design is the technical centre of the software engineering process, regardless of the development paradigm used. The system design process develops the architectural detail required to build a system or product. As with any systematic technique, this software went through the best design phase possible, fine-tuning all efficiency, performance, and accuracy levels. During the design stage, a user-oriented document is transformed into a document for programmers or database employees. Logical design and physical design are the two phases of system design development.

4.2 UML DIAGRAM

The software system artefacts are specified, visualised, constructed, and documented using a standard language called UML. A draught of the UML 1.0 specification was provided to the Object Management Group (OMG), which was in charge of creating UML, in January 1997.

The abbreviation for this language is UML. UML is different from other widely used programming languages like C++, Java, COBOL, etc. Software designs are created using the visual language UML. UML is a general-purpose visual modelling language for the specification, design, construction, and documentation of software systems. Although representing software systems is the most popular use of UML, it is not the only one. Additionally, it is utilised to simulate non-software systems. the process flow in the manufacturing facility, etc. Despite the fact that UML is not a programming language, there are tools available to create code from UML diagrams in a number of different languages. UML is directly related to the analysis and design of objects-oriented systems. UML is directly related to the analysis and design of objects-oriented systems. All the components and relationships are included in a thorough UML diagram that represents a system. The visual impact of the UML diagram is the most important factor in the entire process. Utilizing all the other elements completes it. UML includes the following nine diagrams.

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

4.2.1 USE CASE DIAGRAM

An illustration of the interactions between system components is a use case diagram. A use case is a strategy for locating, defining, and organizing system needs. The word "system" in this context refers to a thing that is being built or operated, such as a website for mail-order service and product sales. Use case diagrams are used in UML (Unified Modeling Language), a standard language for modelling real-world objects and systems. Creating an online assistance resource, validating a hardware design, testing and debugging a software product that is still in development, validating overall requirements, and performing customer support tasks are just a few examples of system objectives. For instance, use cases in the context of product sales can include payment processing, item ordering, customer assistance, and catalogue updating. A use case diagram consists of four components.

- The actors, often people involved in the system who are defined according to their roles; the boundary, which establishes the system of interest in relation to its environment.

- The relationships between and among the actors and the use cases, which are the precise roles played by the players within and around the system.

- A use case's naming is very significant. The name should be chosen in a way that makes it clear what functions are being carried out.
- Give performers a name that fits them.
- Diagram relationships and dependencies clearly.

- As the primary function of the diagram is to establish the needs, avoid attempting to incorporate all kinds of relationships.
- When necessary, take notes to help you remember certain crucial details.

To represent the functional needs of a system, use case diagrams are produced. Following the identification of the aforementioned items, we must follow the following guidelines to produce an effective use case diagram.

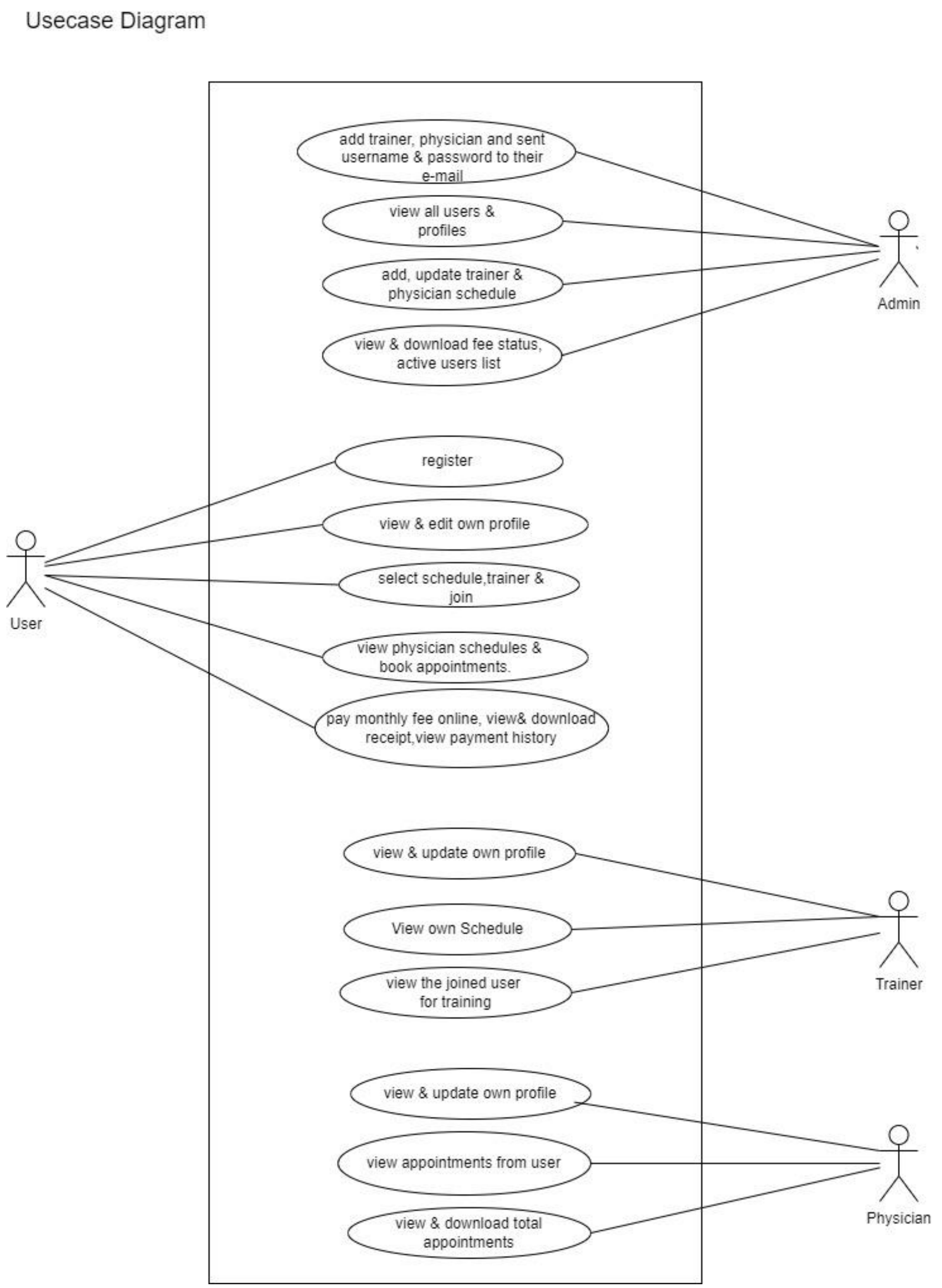
• The name of a use case is highly important. The name should be chosen such that it is obvious what tasks are being carried out.

• Give the performers appropriate names.

• Show dependencies and relationships in the diagram with clarity.

• Remember that the diagram's main purpose is to highlight the needs; do not try to include every relationship that could exist.

Fig 1 : Use case diagram for FitZone


Usecase Diagram
add trainer, physician and sent username & password to their e-mail
view all users & profiles
add, update trainer & physician schedule
view & download fee status, active users list
Admin
register
view & edit own profile
select schedule,trainer & join
view physician schedules & book appointments.
pay monthly fee online, view& download receipt,view payment history
User
view & update own profile
View own Schedule
view the joined user for training
Trainer
view & update own profile
view appointments from user
view & download total appointments
Physician

4.2.2 SEQUENCE DIAGRAM

A sequence diagram primarily depicts the sequential order in which events occur or how entities interact with one another. Event diagrams and event scenarios are other names for sequence diagrams. Sequence diagrams display the actions performed by a system's parts chronologically. These diagrams are widely used by businesspeople and software engineers to document and explain the requirements for new and existing systems.

Sequence Diagram Notations –

i. Actors – A specific type of role in which it interacts with the system's objects is represented by an actor in a UML diagram. An actor is always outside the scope of the system that we want to describe using the UML diagram. We use actors to play a range of characters, including human users and other external subjects. An actor is represented by a stick person in a UML diagram. A sequence diagram may contain multiple actors.

ii. Lifelines – A lifeline is a named component that displays a particular participant in a sequence diagram. A lifeline essentially stands in for each incident in a sequence diagram. The lifeline pieces are positioned at the top of a sequence diagram.

iii. Messages – Using messages, the demonstration of object communication. The messages are presented chronologically on the lifeline. Messages are shown via arrows. The two major elements of a sequence diagram are lifelines and messages.

The following categories broadly describe how messages are categorized:

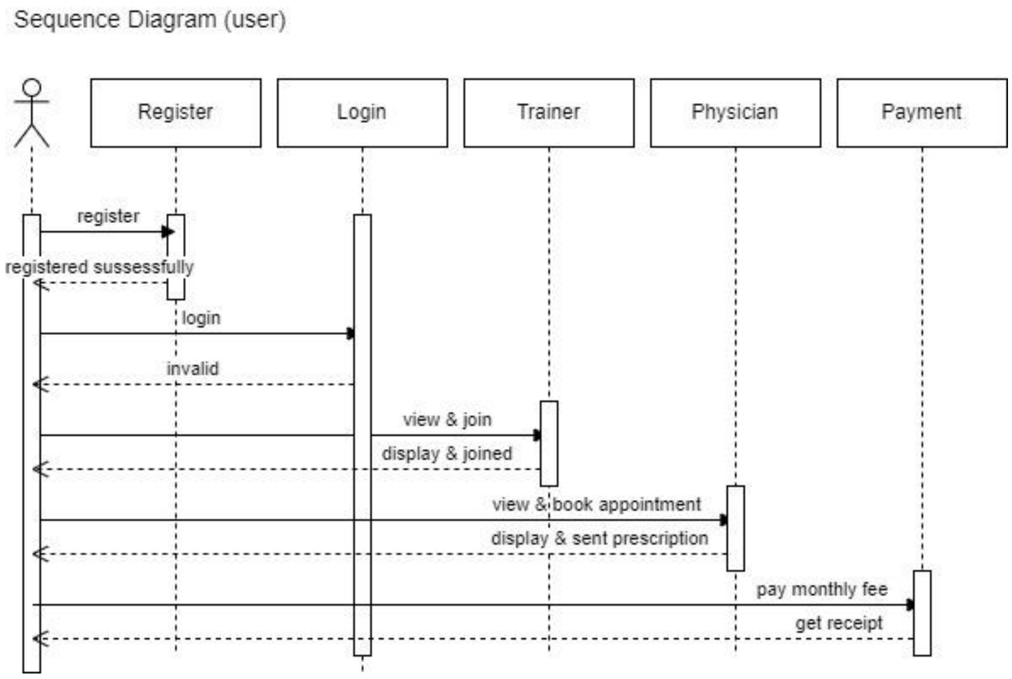
- Synchronous messages
- Asynchronous Messages
- Create message
- Delete Message
- Self-Message
- Reply Message
- Found Message
- Lost Message

iv. **Guards –** We use guards in the UML to model situations. We use them to restrict the flow of communications when we need to pretend that a condition has been satisfied. Guards are used by software developers to notify them of the restrictions imposed by a system or particular process.

Uses of sequence diagrams –

- They are also used to show details of UML use case diagrams and to define and show the logic underpinning intricate actions, procedures, or functions.
- used to fully understand how current or forthcoming systems operate.
- Visualize the information flow between various system components or objects.

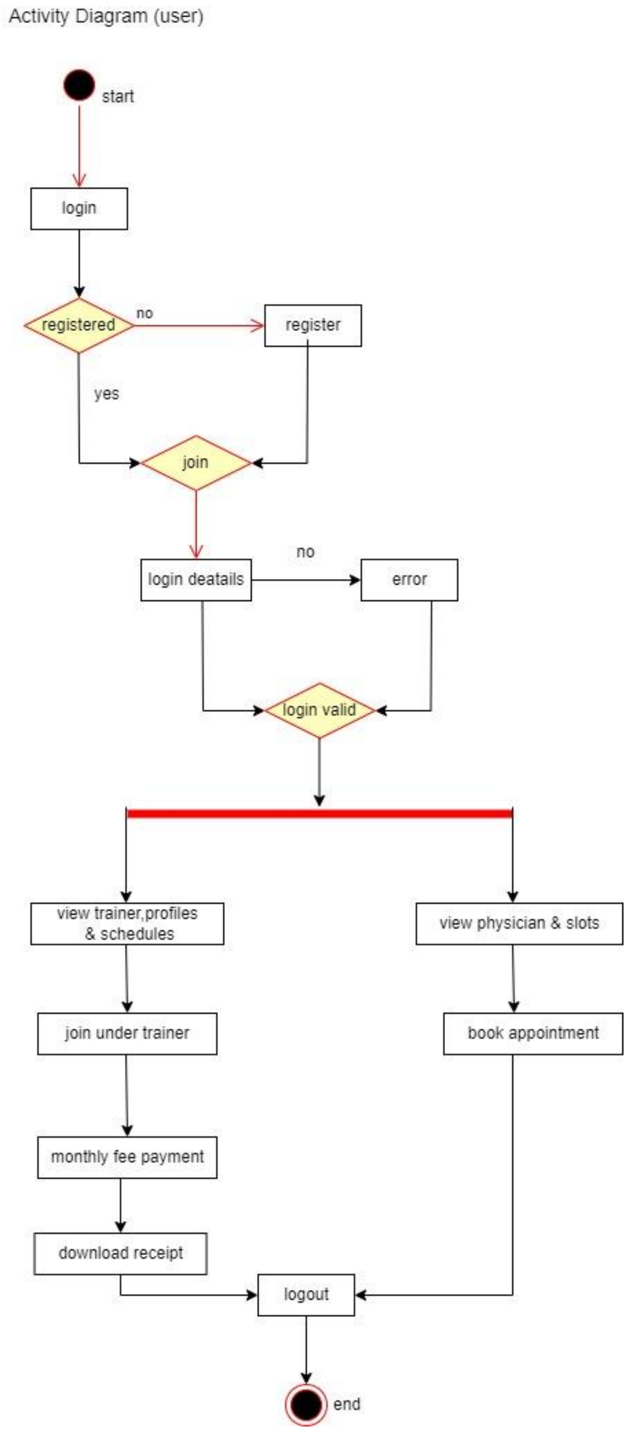
Fig 2 : Sequence diagram for FitZone



4.2.3 ACTIVITY DIAGRAM

Essentially, an activity diagram is a flowchart that illustrates how one activity leads to another. The procedure may be known as a system operation. In the control flow, one action leads to the next. This flow may be concurrent, parallel, or branching. Activity diagrams have a variety of characteristics to accommodate different sorts of flow control, like fork, join, etc.

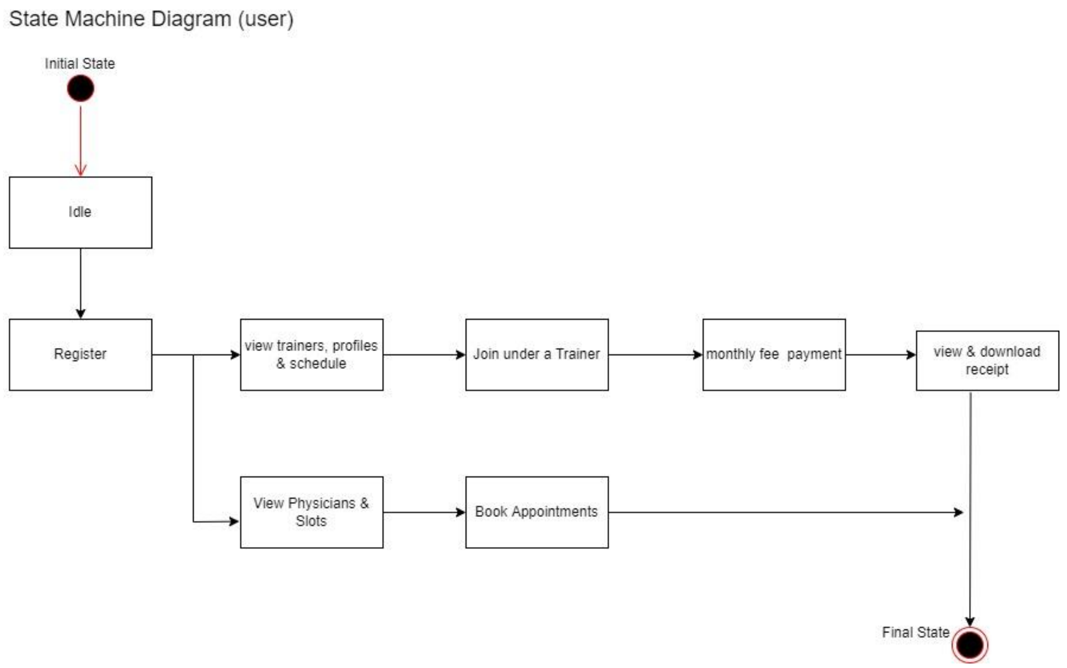
Fig 3 : Activity diagram for FitZone



4.2.3 STATE CHART DIAGRAM

One of the five UML diagrams used to depict a system's dynamic nature is the statechart diagram. Throughout an object's existence, they define several states, and these states are altered by events. The reactive systems can be modelled with statechart diagrams. A system that reacts to internal or external events is known as a reactive system.

Fig 4 : State Chart diagram for FitZone

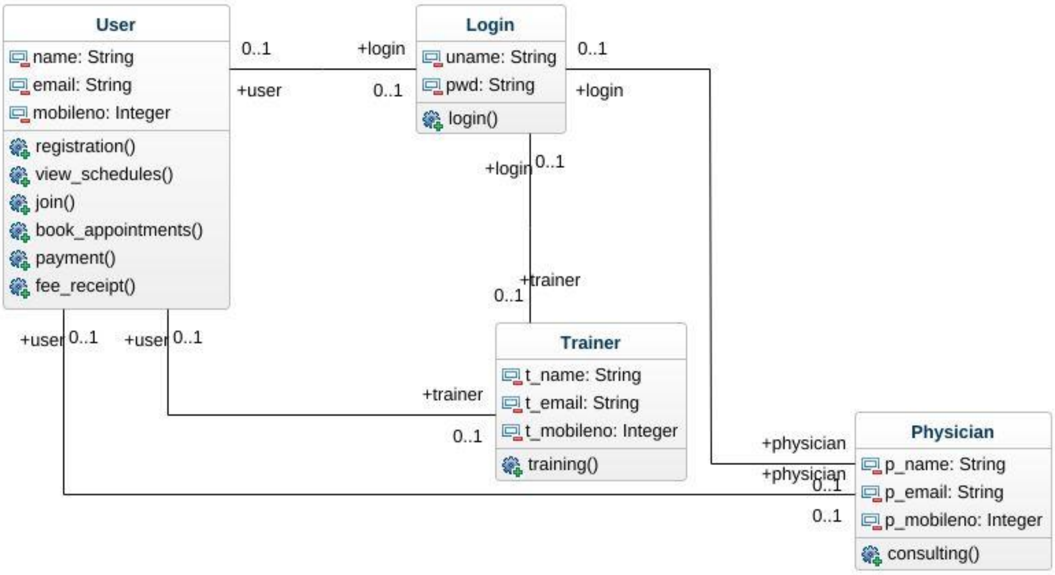


4.2.5 CLASS DIAGRAM

Class diagrams are a type of static diagram. It displays the static view of the application. Class diagrams are useful for visualising, describing, and documenting various system components as well as for writing executable code for software applications.

A class diagram describes a class's attributes and functions as well as the constraints the system is subject to. 5 Class diagrams are widely used in the modelling of object-oriented systems since they are the only UML diagrams that can be directly translated using object-oriented languages.

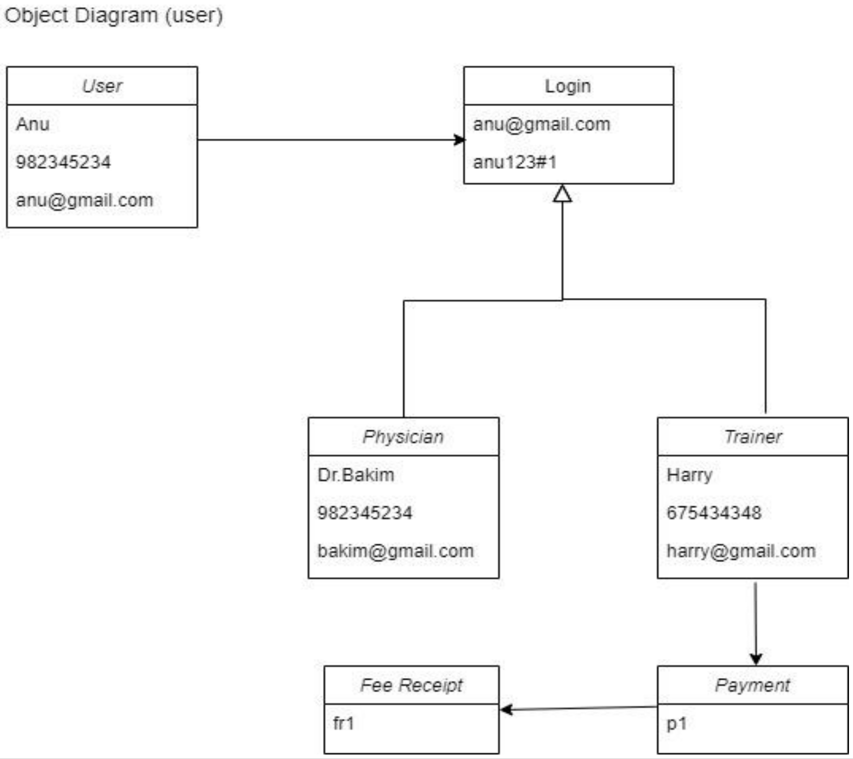
Fig 5 : Class diagram for FitZone



4.2.6 OBJECT DIAGRAM

An object diagram serves as a representation for a class diagram instance. The same basic concepts are used in both class and object diagrams. Object diagrams can also depict a system's static view, however this static view only provides a brief glimpse of the system.

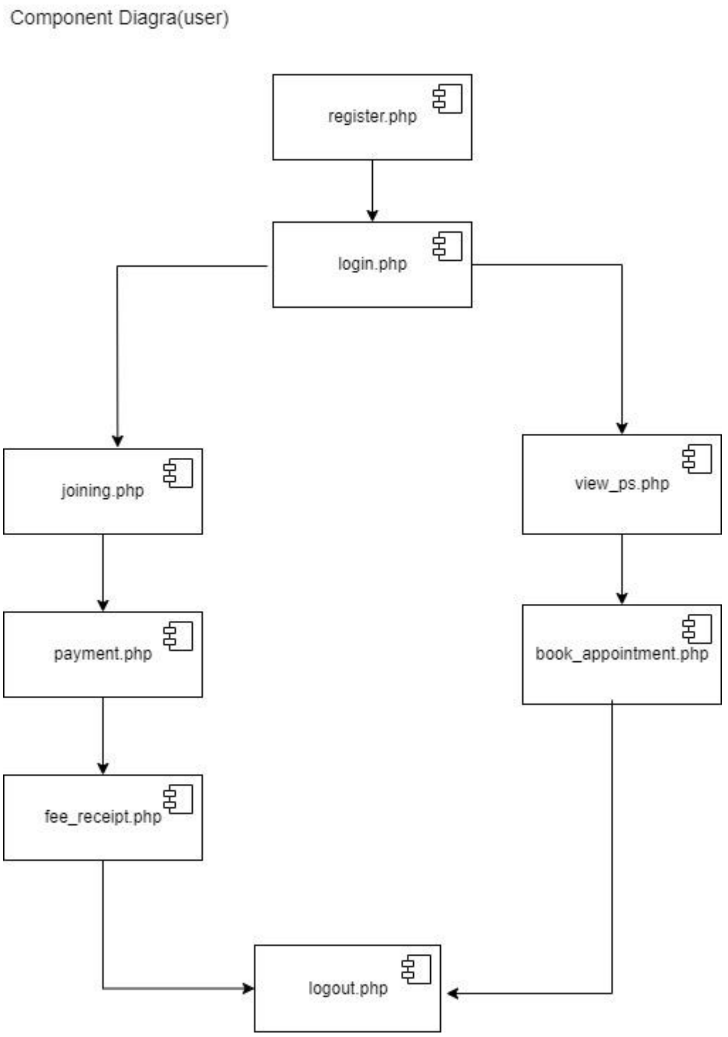
Fig 6 : Class diagram for FitZone



4.2.7 COMPONENT DIAGRAM

A component diagram, often called a UML component diagram, shows how the physical parts of a system are wired up and organised. Component diagrams are frequently used to model implementation details and confirm that all necessary system functions have been accounted for.

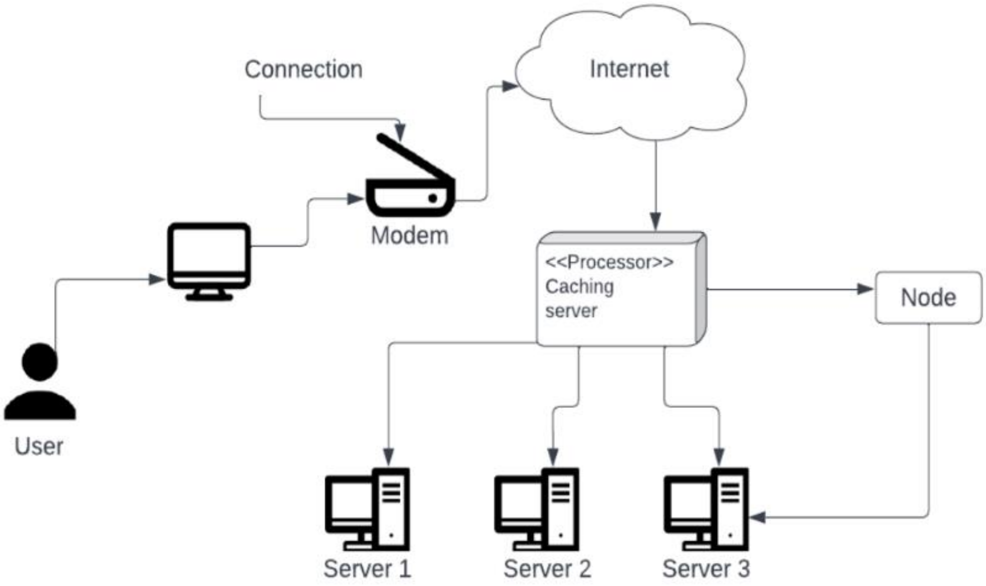
Fig 7 : Component diagram for FitZone



4.2.8 DEPLOYMENT DIAGRAM

A deployment diagram, a type of UML diagram, illustrates the execution architecture of a system, which consists of nodes such as hardware or software execution environments and the middleware connecting them. Typically, the real hardware and software of a system are represented using deployment diagrams.

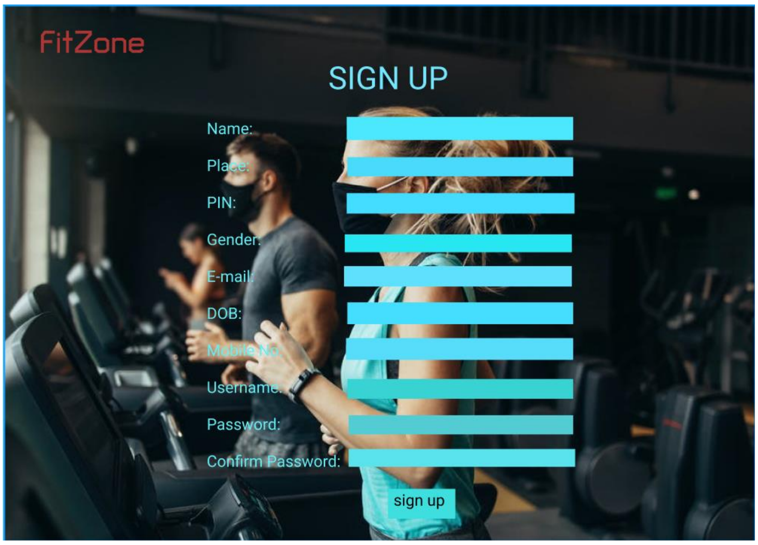
Fig 8 : Component diagram for FitZone



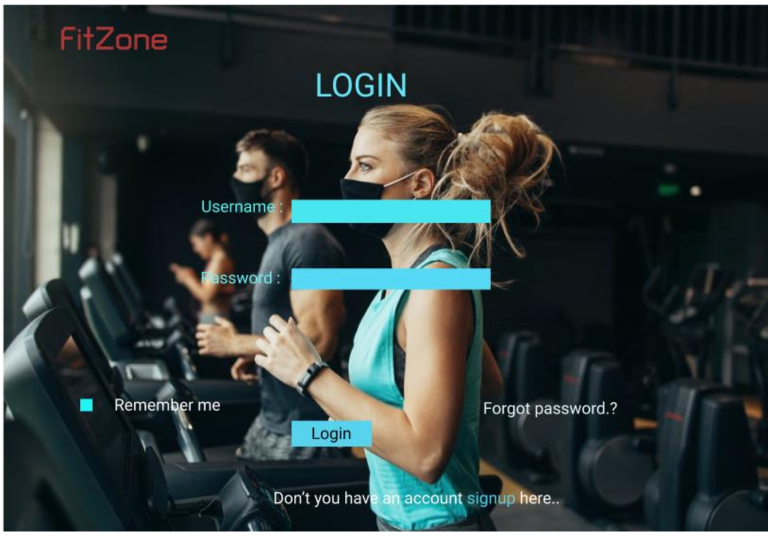
4.3 USER INTERFACE DESIGN

4.3.1-INPUT DESIGN

Form Name : User Registration

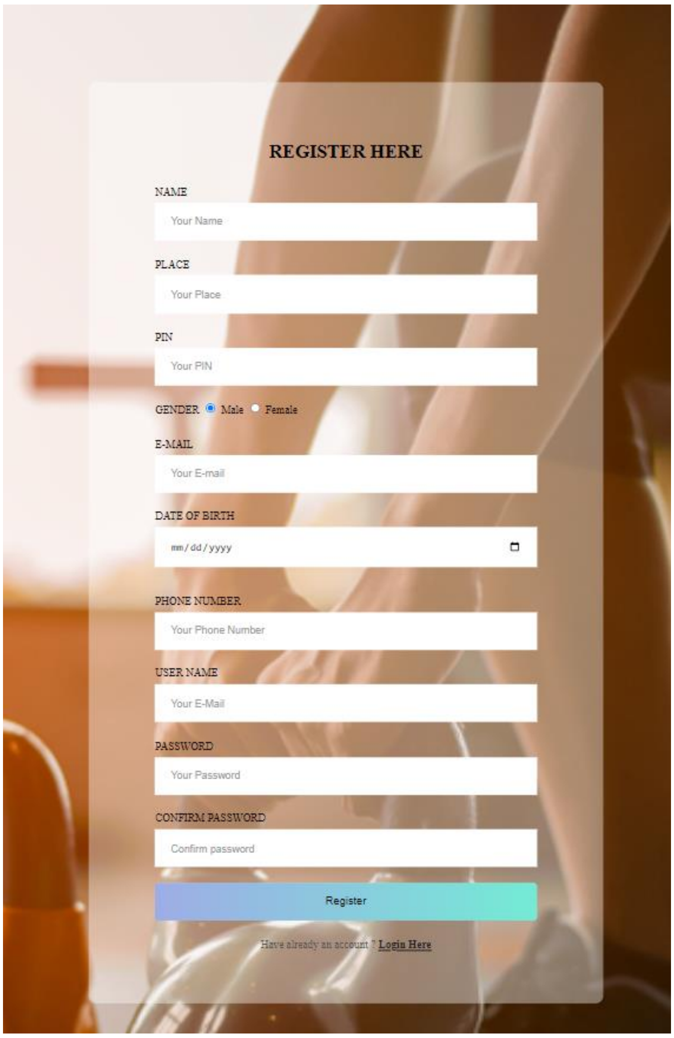


Form Name : User Login

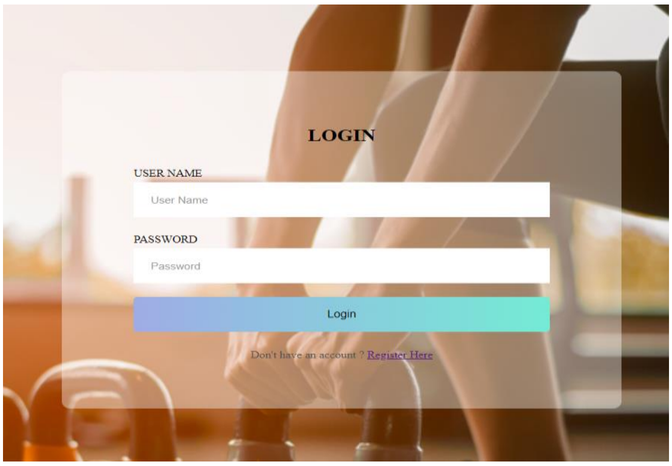


4.3.2 OUTPUT DESIGN

User-Registration



User Login



4.4 DATABASE DESIGN

A database is a structured system that has the ability to store information and enables users to quickly and efficiently access stored information. The main objective of any database is its data, which requires protection.

The process of designing a database involves two steps. First, the user demands are ascertained, and then a database is built to as clearly as possible satisfy these requirements. Information level design is a procedure that is carried out separately from all DBMSs.

In the second stage, the information level design for the particular DBMS that will be used to build the system in question is transformed into a design. The physical level design stage is where the specific DBMS that will be used's properties are discussed. A database design runs parallel to the system design. The database's data configuration seeks to achieve the two objectives indicated below.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A relational model presents the database as a collection of relationship. A file or table of records with values can be compared to each relation. 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. Numerous tables with unique names make up a relational database. Each row in a tale corresponds to a set of related values.

Relations, Domains & Attributes

A relation is a table. The units of a table's rows are known as tuples. A tuple is an ordered collection of n elements. Columns are the name given to attributes. There are already defined relationships between each table in the database. This ensures that referential and entity relationships are authentic. A domain D is a set of atomic values. A common method of defining a domain is by selecting a data type from which the domain's data values are derived. Additionally, naming the domain helps make its values simpler to comprehend. A relation's values are all atomic and cannot be divided.

Relationships

- Table associations are established using keys. The two primary keys that are most important are the primary key and the foreign key. Relationships for entity integrity and referential integrity can be made using these keys.
- For any Primary Key, Entity Integrity prohibits the usage of null values.
- No Primary Key may contain null values, according to Referential Integrity.
- Referential Integrity: Each distinct Foreign Key value must have a corresponding Primary Key value inside the same domain. Additional keys are Super Key and Candidate Keys.

4.6.2 Normalization

The simplest possible grouping of data is used to put them together so that future changes can be made with little influence on the data structures. The formal process of normalising data structures in a way that reduces duplication and fosters integrity. Using the normalisation technique, superfluous fields are removed and a huge table is divided into several smaller ones. Anomalies in insertion, deletion, and updating are also prevented by using it. Keys and relationships are two notions used in the standard form of data modelling. A row in a table is uniquely identified by a key. Primary keys and foreign keys are the two different kinds of keys. A primary key is an element, or set of components, in a table that serves as a means of distinguishing between records from the same table. A column in a table known as a foreign key is used to uniquely identify records from other tables. Up to the third normal form, all tables have been normalised. As the name suggests, it refers to returning things to their regular state. By using normalisation, the application developer aims to organise the data in a meaningful way into appropriate tables and columns, where user correlation of names to the data is simple. By removing recurring groups from data, normalisation prevents data redundancy, which places a heavy demand on the computer's resources. These consist of:

- ✓ Normalize the data.
- ✓ Select appropriate table and column names.
- ✓ Select a suitable name for the data.

First Normal Form

According to the First Normal Form, each attribute in a tuple must have a single value from the attribute's domain and its domain must only include atomic values. The 1NF forbids "relations within relations" or "relations as attribute values within tuples," in other words. By 1NF, only single atomic or indivisible values are allowed for attribute values. Putting the data into First Normal Form is the first step. Data that is of a similar type in each table can be moved into separate tables to solve this problem. According to the project's requirements, a Primary Key or Foreign Key is assigned to each table. For each nested or non-atomic attribute, we create additional relations in this. For each nested relation or non-atomic attribute, new relations are formed in this process. This got rid of data groups that were repeated. If a relation solely meets the constraints that include the primary key, it is said to be in first normal form.

Second Normal Form

In relations when the primary key comprises many properties, no non-key attribute should be functionally dependent on a component of the primary key. This entails creating a new connection for each partial key and dissecting it into its dependent properties. Keep your database's original primary key and any attributes that are totally dependent on it. This method helps to eliminate data that only depends on a small amount of the key. If and only if a relation satisfies all the requirements for first normal form for the primary key and all of the non-primary key qualities of the relation are completely dependent on the primary key alone, then that relation is said to be in second normal form.

Third Normal Form

A non-key attribute of a Relation should not be functionally determined by another non-key property or by a collection of non-key attributes, according to the Third Normal Form. In other words, the primary key shouldn't be transitively dependent on anything. In this, we break down the relation into its component parts and build up the non-key qualities that functionally determine the other non-key attributes. To eliminate anything that does not totally dependent on the Primary Key, this step is conducted. A relation is only considered to be in third normal form if it is in second normal form, and furthermore, the relation's non-key characteristics should not depend on other non-key attributes.

TABLE DESIGN

Table No 01

Table Name : tbl_login

Primary Key : login_id

Foreign Key :

Table Description : To store user Login information

Sl.no	Field name	Data type	Size	Constraints
1	login_id	int	11	P.K
2	uname	varchar	50	Not null
3	pwd	varchar	50	Not null
4	role	varchar	20	Not null

Table No 02

Table Name : user_reg
Primary Key : u_id

Foreign Key : login_id

Table Description: To store user registration information

Sl.no	Field name	Data type	Size	Constraints
1	u_id	int	11	P.K
2	login_id	int	11	F.K
3	name	varchar	50	Not null
4	place	varchar	50	Not null
5	pin	int	11	Not null
6	gender	varchar	10	Not null
7	email	varchar	50	Not null
8	dob	date		Not null
9	phone	bigint	11	Not null

Table No 03

Table Name : add_trainer

Primary Key : t_id

Foreign Key : login_id

Table Description: To store trainer information

Sl.no	**Field name**	**Data type**	**Size**	**Constraints**
1	t_id	int	11	P.K
2	login_id	int	11	F.K
3	t_name	varchar	50	Not null
4	t_place	varchar	50	Not null
5	t_pin	int	11	Not null
6	t_gender	varchar	10	Not null
7	t_email	varchar	50	Not null
8	certification	varchar	50	Not null
9	t_phone	bigint	11	Not null
10	t_experience	varchar	50	Not null
11	t_aadhar	bigint	20	Not null
12	t_upaadhar	varchar	50	Not null

Table No 04

Table Name : add_physician

Primary Key : p_id

Foreign Key : login_id

Table Description: To store physician details

Sl.no	Field name	Data type	Size	Constraints
1	p_id	int	11	P.K
2	login_id	int	11	F.K
3	p_name	varchar	50	Not null
4	p_place	varchar	50	Not null
5	p_pin	int	11	Not null
6	p_gender	varchar	10	Not null
7	p_email	varchar	50	Not null
8	p_qualification	varchar	50	Not null
9	p_phone	bigint	11	Not null
10	p_experience	varchar	50	Not null
11	license_no	varchar	10	Not null
12	up_license	varchar	50	Not null

Table No 05

Table Name : t_schedule

Primary Key : ts_id

Foreign Key : t_id

Table Description: To store schedules of trainer

Sl.no	Field name	Data type	Size	Constraints
1	ts_id	int	11	P.K
2	t_name	varchar	50	Not null
3	t_id	int	11	Not null
4	ts_day	varchar	10	Not null
5	ts_time	varchar	30	Not null

Table No 06

Table Name : user_tr

Primary Key : ut_id

Foreign Key : u_id

Table Description: To store joining schedule of user

Sl.no	**Field name**	**Data type**	**Size**	**Constraints**
1	ut_id	int	11	P.K
2	u_id	int	11	Not null
3	name	varchar	50	Not null
4	ts_day	varchar	10	Not null
5	ts_time	varchar	20	Not null
6	t_name	varchar	20	Not null

Table No 07

Table Name : p_schedule

Primary Key : ps_id

Foreign Key : p_id

Table Description: To store schedules of physician

Sl.no	**Field name**	**Data type**	**Size**	**Constraints**
1	ps_id	int	11	P.K
2	p_name	varchar	50	Not null
3	p_id	int	11	Not null
4	ps_day	varchar	10	Not null
5	ps_time	varchar	30	Not null

Table No 08

Table Name : appointment_booking

Primary Key : b_id

Foreign Key :u_id, p_id

Table Description: To store appointment booking details for physician

Sl.no	Field name	Data type	Size	Constraints
1	b_id	int	11	P.K
2	u_id	int	11	F.K
3	p_id	int	11	F.K
4	logid	int	11	Not null
5	name	varchar	30	Not null
6	email	varchar	20	Not null
7	phone	varchar	10	Not null
4	b_date	date		Not null
5	slot	varchar	10	Not null

Table No 09

Table Name : tbl_slot

Primary Key :slot_id

Foreign Key : p_id

Table Description: To store appointment slots

Sl.no	Field name	Data type	Size	Constraints
1	slot_id	int	11	P.K
2	p_id	int	11	F.K
3	slot	varchar	20	Not null
4	status	tinyint	1	Not null

Table No 10

Table Name : tbl_payment

Primary Key :payment_id

Foreign Key : u_id

Table Description: To store fee payment details

Sl.no	**Field name**	**Data type**	**Size**	**Constraints**
1	payment_id	int	11	P.K
2	name	varchar	50	F.K
3	u_id	int	11	Not null
4	amount	varchar	10	Not null
5	date	datetime		Not null
6	status	varchar	10	Not null
7	month	varchar	20	Not null

CHAPTER 5

SYSTEM TESTING

5.1 INTRODUCTION

The technique of closely monitoring the execution of software to check that it acts as intended is known as software testing. The words verification and validation are frequently used in conjunction with software testing. A product, including software, is examined or evaluated as part of the validation process to see if it complies with all pertinent specifications. Software testing uses techniques including reviews, analyses, inspections, and walkthroughs as one sort of verification. Validation is the process of ensuring that what has been specified corresponds to what the user actually wants.

Additional techniques that are typically connected to software testing include static analysis and dynamic analysis. Without actually running the code, static analysis examines the software's source code to look for errors and gather statistics. In order to provide information like execution traces, timing profiles, and test coverage details, dynamic analysis evaluates how software behaves while it is operating.

Individual components are tested first, then the entire computer-based system is integrated. Testing starts with individual modules and progresses to the integration of the full computer-based system. There are numerous regulations that can be utilised as testing goals, and testing is required for the goals of the system testing to be successful. They are:

Executing a programme during testing is a procedure used to look for errors.

- A excellent test case is one that has a decent chance of detecting an error that hasn't been found yet.
- A successful test is one that uncovers an undiscovered error.

A test that effectively achieves the aforementioned objectives will identify software flaws. Additionally, testing demonstrates that the programme appears to function in line with the specification and that the performance requirements appear to have been met. There are three ways to test program.

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

Testing for correctness is meant to ensure that a programme performs exactly as it was intended to. This is much harder than it might initially seem, especially for big programmes.

5.2 TEST PLAN

The procedures that must be followed to fulfil various testing approaches are suggested in a test plan. The test plan specifies the task that must be completed. Software developers produce a computer programme, its documentation, and any related data structures. It is always the software developers' job to test each of the program's individual components to ensure that it serves the intended function. There is an impartial test group in order to address the problems of letting the developer evaluate what they have created (ITG). The precise objectives of testing should be stated in quantitative terms. Information on the mean time to failure, the cost to detect and correct flaws, the remaining defect density or frequency of occurrence, and test work hours per regression test should all be included in the test plan.

The levels of testing include:

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

5.2.1 Unit Testing

The smallest unit of software design—the software component or module—is the focus of unit testing, which concentrates verification work. Important control pathways are examined in order to find faults inside the module's border using the component level design description as a guide. The scope set for unit testing and the relative complexity of tests. Unit testing can be carried out simultaneously for numerous components and is white-box focused. Information flow into and out of the programme unit under test is monitored by the modular interface to ensure appropriate operation. To make sure that temporary data is kept in its original form during all phases of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at least once, boundary conditions are evaluated. Finally, each path for managing errors is examined.

Testing of data flow through a module interface are important before beginning any additional tests. If data cannot correctly enter and exit the system, all other tests are useless. The unit test's selective analysis of execution routes is a crucial responsibility. To cleanly reroute or stop work when an issue does occur, error handling channels must be set up and error scenarios must be anticipated in excellent design. Boundary testing is the last stage of unit testing. At its boundaries, software frequently fails.

In the Sell-Soft System, unit testing was carried out by treating each module as a distinct entity and subjecting them to a variety of test inputs. The internal logic of the modules had some issues, which were fixed. Each module is tested and run separately after coding. All unused code was eliminated, and it was confirmed that every module was functional and produced the desired outcome.

5.2.2 Integration Testing

Integration testing is a methodical approach for creating the program's structure while also carrying out tests to find interface issues. The goal is to construct a programme structure that has been determined by design using unit tested components. The programme as a whole is tested. Correction is challenging since the size of the overall programme makes it challenging to isolate the causes. As soon as these mistakes are fixed, new ones arise, and the process repeats itself in an apparently unending cycle. All of the modules were integrated after unit testing was completed in the system to check for any interface inconsistencies. A distinctive programme structure also developed when discrepancies in programme structures were eliminated.

5.2.3 Validation Testing or System Testing

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, modules, and class modules. Popular names for this type of testing include system tests and black box testing.

The functional requirements of the software are the main emphasis of the black box testing approach. To completely exercise all functional requirements for a programme, the software engineer can create sets of input conditions using Black Box testing.

The following sorts of problems are targeted by black box testing: erroneous or missing functions, interface errors, data structure or external data access errors, performance errors, initialization errors, and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system under consideration has its user acceptance assessed; in this case, it must meet the needs of the company. When creating and making modifications as needed, the software should stay in touch with the user and viewpoint system. The following points are considered in this:

Output screen designs, input screen designs, and

The testing mentioned above is carried out using several test data types. In the system testing process, the preparation of test data is crucial. The system under study is evaluated using the test data after it has been prepared. During system testing, faults in the test data are once again found and fixed using the testing procedures described above. The fixes are also logged for use in the future.

Automation Testing

Automation testing is the process of testing software and other tech products to ensure it meets strict requirements. Essentially, it’s a test to double-check that the equipment or software doesexactly what it was designed to do. It tests for bugs, defects, and any other issues that can arisewith product development. Automation testing can be run at any time of the day. It uses scripted sequences to examine the software. It then reports on what’s been found, and this information can be compared with earlier test runs.

Benefits of Automation Testing

Capacities for thorough reporting - Well-written test cases for numerous scenarios are used in automation testing. These planned sequences can cover a lot of ground and produce in-depth reports that are simply impossible for a human to produce.

Improved bug detection - Finding bugs and other flaws in a product is one of the key reasons to test it. This approach is simplified by automation testing. Additionally, it can examine a greater test coverage than perhaps people can.

- Simplifies testing - Most SaaS and tech organisations regularly test their products as part of their operations. The key is to make things as easy as you can. Automation has a lot of advantages. The test scripts can be reused when automating test tools.
- Speeds up the testing process - Automation and machines complete tasks more quickly than people. This is why we employ them, along with more accuracy. Consequently, your software development cycles are shortened.

- Reduces human intervention - Without human supervision, tests can be carried out whenever it is convenient, including overnight. Additionally, this can lessen the possibility of human error when it is carried out mechanically.

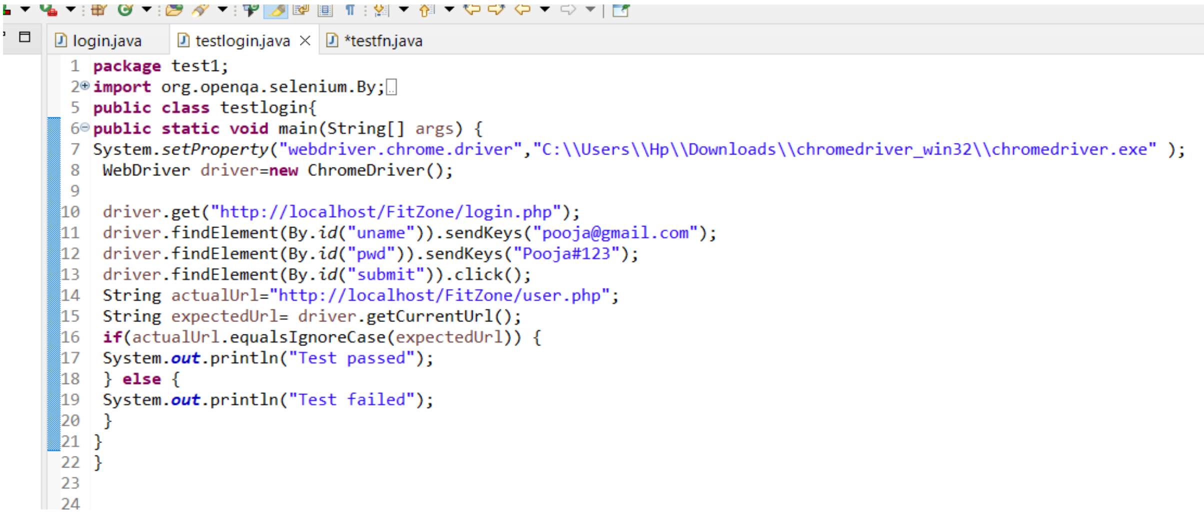
5.2.5 Selenium Testing

An open-source programme called Selenium automates web browsers. It offers a single interface that enables you to create test scripts in a number of different programming languages, including Ruby, Java, NodeJS, PHP, Perl, Python, and C#. The Selenium testing tool is used to automate tests for web applications across browsers. It is used to verify that online apps, whether responsive, progressive, or standard, are of the highest calibre. Selenium is a free software programme.

Test cases for a Login Page

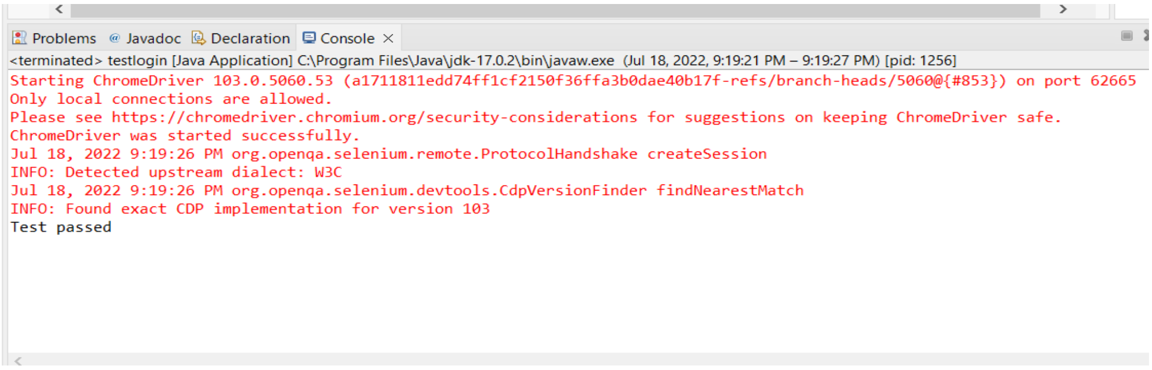
Project Name: FitZone					
Login Test Case					
Test Case ID: testlogin			**Test Designed By: Swarna Mol K K**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 18-07-2022		
Module Name: Login Screen			**Test Executed By: Shelly Shiju George**		
Test Title: Verify login with valid username and password			**Test Execution Date:** 18-07-2022		
Description: Test the Login Page					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Login Page should be display ed	Login page displayed	Pass
2	Provide Valid username	Username : pooja@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboard	Pass
3	Provide Valid Password	Password: Pooja#123			
4	Click on Login button				
5	Provide Invalid username or password	Username: poojaa@gmail.com Password: Pooja%1	User should not be able to Login	Message for enter valid email id or password displayed	Pass
6	Provide Null username or Password	Username : null Password: null			
7	Click on Sign In button				
Post-Condition: User is successfully authenticated with the database and logged into the account. Account session information is recorded in the database.					

Code package



```
package test1;
import org.openqa.selenium.By;
public class testlogin{
public static void main(String[] args) {
System.setProperty("webdriver.chrome.driver","C:\\Users\\Hp\\Downloads\\chromedriver_win32\\chromedriver.exe" );
 WebDriver driver=new ChromeDriver();

 driver.get("http://localhost/FitZone/login.php");
 driver.findElement(By.id("uname")).sendKeys("pooja@gmail.com");
 driver.findElement(By.id("pwd")).sendKeys("Pooja#123");
 driver.findElement(By.id("submit")).click();
 String actualUrl="http://localhost/FitZone/user.php";
 String expectedUrl= driver.getCurrentUrl();
 if(actualUrl.equalsIgnoreCase(expectedUrl)) {
 System.out.println("Test passed");
 } else {
 System.out.println("Test failed");
 }
}
}
```

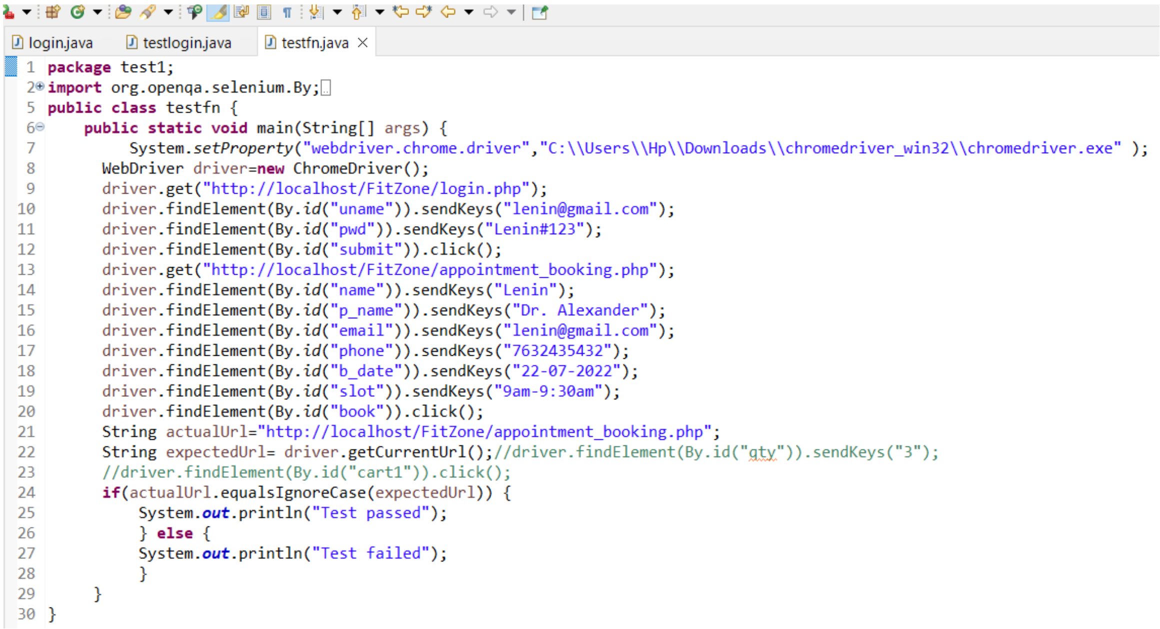


```
<terminated> testlogin [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe  (Jul 18, 2022, 9:19:21 PM – 9:19:27 PM) [pid: 1256]
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 62665
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 18, 2022 9:19:26 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 9:19:26 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```

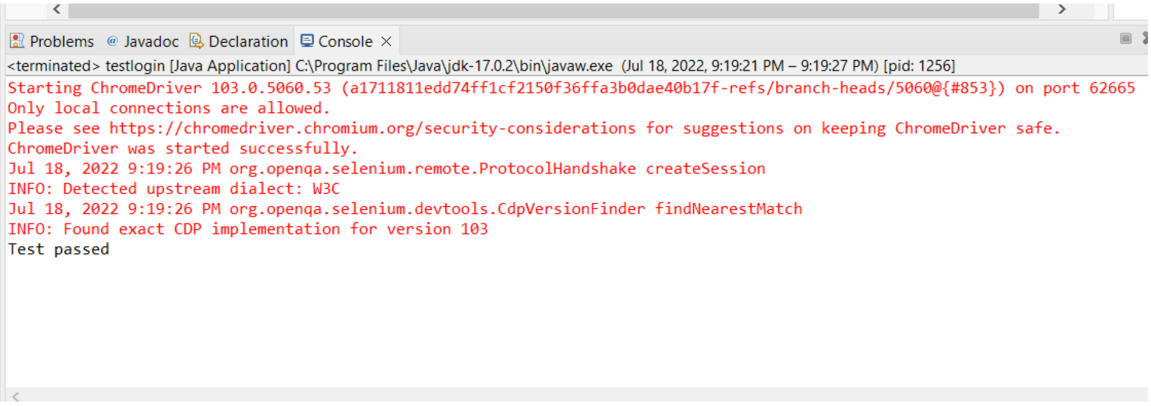
Test cases for a Appointment booking page

Project Name: FitZone					
booking Test Case					
Test Case ID: testbook			**Test Designed By: Swarna Mol K K**		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 21-07-2022		
Module Name: Booking Screen			**Test Executed By: Shelly Shiju George**		
Test Title: Verify booking with valid details			**Test Execution Date:** 21-07-2022		
Description: Test the Booking Page					
Pre-Condition: User has valid booking details					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Booking Page		Booking Page should be displayed	Booking page displayed	Pass
2	Provide Valid p_name	p_name : Dr. Alexander	User should be able to Book	User Booked in and navigated to User Dashboard	Pass
3	Provide Valid b_date	b_date: 22-07-2022			
4	Provide Valid slot	slot : 9am-9:30am			
5	Click on book button				
5\	Provide Invalid p_name or b_date or slot	p_name: Dr. Alen b_date: 01-07-2002 slot: 12am-12:30am	User should not be able to Book	Message for enter valid p_name or b_date or slot displayed	Pass
6	Provide Null P_name or b_date or slot	p_name : null b_date: null slot: null			
7	Click on book button				
Post-Condition: User is successfully authenticated with the database and book the appointment. Account session information is recorded in the database.					

Code package



```
package test1;
import org.openqa.selenium.By;
public class testfn {
    public static void main(String[] args) {
        System.setProperty("webdriver.chrome.driver","C:\\Users\\Hp\\Downloads\\chromedriver_win32\\chromedriver.exe" );
      WebDriver driver=new ChromeDriver();
      driver.get("http://localhost/FitZone/login.php");
      driver.findElement(By.id("uname")).sendKeys("lenin@gmail.com");
      driver.findElement(By.id("pwd")).sendKeys("Lenin#123");
      driver.findElement(By.id("submit")).click();
      driver.get("http://localhost/FitZone/appointment_booking.php");
      driver.findElement(By.id("name")).sendKeys("Lenin");
      driver.findElement(By.id("p_name")).sendKeys("Dr. Alexander");
      driver.findElement(By.id("email")).sendKeys("lenin@gmail.com");
      driver.findElement(By.id("phone")).sendKeys("7632435432");
      driver.findElement(By.id("b_date")).sendKeys("22-07-2022");
      driver.findElement(By.id("slot")).sendKeys("9am-9:30am");
      driver.findElement(By.id("book")).click();
      String actualUrl="http://localhost/FitZone/appointment_booking.php";
      String expectedUrl= driver.getCurrentUrl();//driver.findElement(By.id("qty")).sendKeys("3");
      //driver.findElement(By.id("cart1")).click();
      if(actualUrl.equalsIgnoreCase(expectedUrl)) {
          System.out.println("Test passed");
          } else {
          System.out.println("Test failed");
          }
     }
}
```



```
<terminated> testlogin [Java Application] C:\Program Files\Java\jdk-17.0.2\bin\javaw.exe (Jul 18, 2022, 9:19:21 PM – 9:19:27 PM) [pid: 1256]
Starting ChromeDriver 103.0.5060.53 (a1711811edd74ff1cf2150f36ffa3b0dae40b17f-refs/branch-heads/5060@{#853}) on port 62665
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
Jul 18, 2022 9:19:26 PM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected upstream dialect: W3C
Jul 18, 2022 9:19:26 PM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
INFO: Found exact CDP implementation for version 103
Test passed
```

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

The project's implementation phase is where the conceptual design is transformed into a functional system. It can be regarded as the most important stage in creating a successful new system since it gives users assurance that the system will operate as intended and be reliable and accurate. User documentation and training are its main concerns. Usually, conversion happens either during or after the user's training. Implementation is the process of turning a newly revised system design into an operational one, and it simply refers to placing a new system design into operation.

The user department is currently the one bearing the bulk of the workload, experiencing the most interruption, and having the greatest impact on the system. Confusion and mayhem may come from a poorly thought-out or managed implementation.

The entire process of moving from the old system to the new one is referred to as implementation. The new system may replace an existing human or automated system, be completely different, or just be improved upon. It is necessary to implement a reliable system that meets organizational requirements. System use or system implementation is the act of putting a built-in system into practice. This includes each step taken to transition from the old to the new system. The system cannot be used until after rigorous testing and if it is confirmed that it is performing in compliance with the standards. The system personnel determine whether the system is viable. As a system is put into place, the complexity of the system analysis and design work required to implement the three essential elements of education and training, system testing, and changeover will rise.

The following tasks are part of the implementation state:

- □ Meticulous planning.
- □ Examination of the system and its limitations.
- □ Designing strategies to implement the transition.

6.2 IMPLEMENTATION PROCEDURES

Software implementation refers to the whole setup of the package in its intended environment, as well as to the system's functionality and satisfaction of its intended uses. In many organizations, the software development project will be commissioned by someone who will not be operating it. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before viewing the system, the user must be aware that the server programme needs to be running on the server in order to view the results. The actual process won't happen if the server object is not active and functioning on the server.

6.2.1 User Training

The purpose of user training is to get the user ready to test and modify the system. It is crucial that the participants in the new system have faith in their abilities to fulfil the goals and reap the benefits of the computer-based system. The necessity for training becomes increasingly critical as systems get more sophisticated. The user learns how to enter data, handle error messages, query the database, call up routines to generate reports, and execute other important tasks through user training.

6.2.2 Training on the Application Software

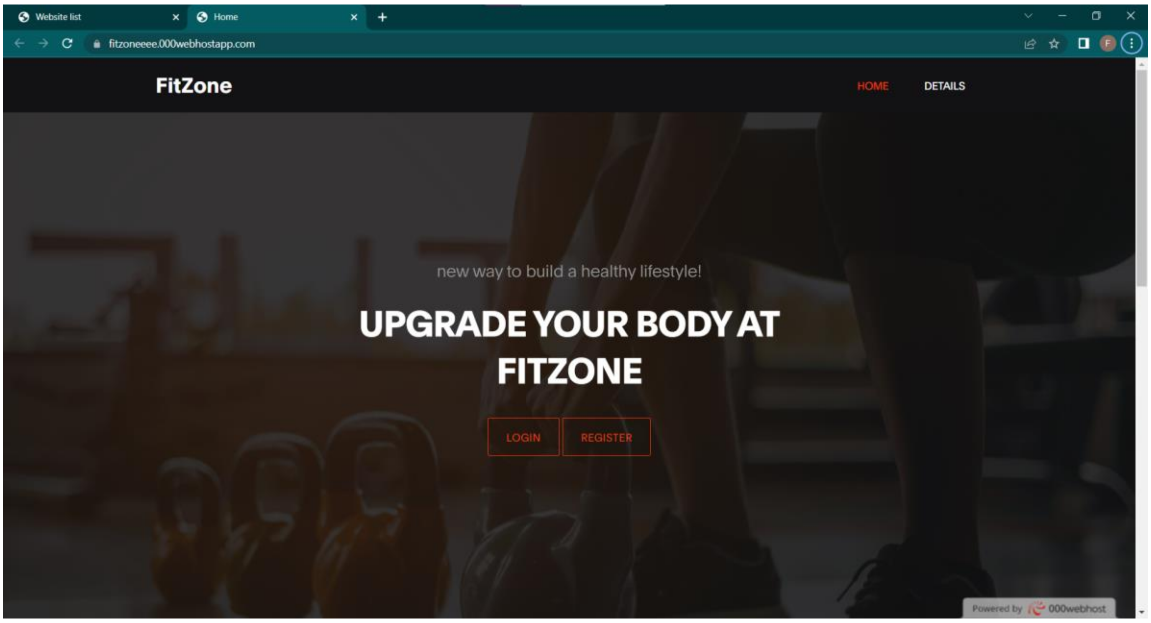
The user will need to receive training on the new application software after receiving the requisite basic instruction on computer awareness. This will explain the fundamental principles of how to utilize the new system, including how the screens work, how they are designed, what kinds of errors can occur while entering data, how each entry is validated, and how to change the date that was entered. Then, while delivering the program's training on the application, it should cover the information required by the particular user or group to use the system or a component of the system. It's possible that this training will vary depending on the user group and the level of hierarchy.

6.2.3 System Maintenance

The mystery of system development is maintenance. When a software product is in the maintenance stage of its lifecycle, it is actively working. A system should be properly maintained after it has been effectively implemented. An essential part of the software development life cycle is system maintenance. In order for a system to be flexible to changes in the system environment, maintenance is required. Of course, software maintenance involves much more than just "Finding Mistakes".

6.2.4 Hosting

000webhost is a free website hosting solution that provides an array of valuable features, including a website builder, WordPress support, and no ads.


Website list
Home
fitzoneeee.000webhostapp.com
FitZone
HOME
DETAILS
new way to build a healthy lifestyle!
UPGRADE YOUR BODY AT FITZONE
LOGIN
REGISTER
Powered by 000webhost

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The technology utilized to run the existing system is outdated, and digital currency and the internet are not used. The proposed system introduces facility for fully-fledged system that will be the backbone of the whole management of the gym so ignoring the risk or error is not an option as later it can make a greater form of itself. Therefore, it is preferable to reduce the issues that the administration and staff in the organisation experience.

7.2 FUTURE SCOPE

- Users can able to add complaints and feedbacks etc.
- Data security can be enhanced.
- Manages member information independently from all bill information while taking the needs of the gym into account.
- Information on common pharmaceuticals is kept in a database.
- Diet recommendation

20% Overall Similarity

Top sources found in the following databases:

- 15% Internet database
- 6% 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...** Crossref	5%
2	**issuu.com** Internet	4%
3	**coursehero.com** Internet	2%
4	**dev.mysql.com** Internet	1%
5	**scribd.com** Internet	1%
6	**essaysauce.com** Internet	<1%
7	**globalapptesting.com** Internet	<1%
8	**ru.scribd.com** Internet	<1%
9	**pdfcoffee.com** Internet	<1%

10	lovelycoding.org Internet	<1%
11	slideshare.net Internet	<1%
12	geeksforgeeks.org Internet	<1%
13	Verdi Yasin, Muhammad Zarlis, Opim Salim Sitompul, Poltak Sihombin... Crossref	<1%
14	docplayer.net Internet	<1%
15	digital.com Internet	<1%
16	researchpublish.com Internet	<1%
17	sliit.lk Internet	<1%
18	nanopdf.com Internet	<1%