

Project Report

On

AGRICULTURE EQUIPMENT RENTAL MANAGEMENT SYSTEM

Project work submitted in partial fulfillment of the requirement
for the award of the degree

B.Sc. Computer Science

By

Y. MANOJ KUMAR (107222468032)



BHAVAN'S VIVEKANANDA COLLEGE
Of Sciences, Humanities & Commerce

SAINIKPURI, SECUNDERABAD

Reaccredited with 'A' grade by NAAC

Autonomous college

Affiliated to Osmania University



Bharatiya Vidya
Bhavan

BHAVAN'S VIVEKANANDA COLLEGE

OF SCIENCE , HUMANITIES AND COMMERCE

AUTONOMOUS COLLEGE - Affiliated to Osmania University

Reaccredited with 'A' Grade by NAAC

Sainikpuri, Secunderabad - 500 094.

Department of Computer Science

CERTIFICATE

This is to Certify that the Project Work entitled

was successfully carried out by

bearing the Hall Ticket No.

*in partial fulfillment of the sessional requirements
leading to the award of the degree in*

Bachelor of Science (B.Sc)

during the academic year 20 - 20

Internal Guide

Head of Department

External Examiner

Date :

ACKNOWLEDGEMENT

Apart from the efforts of me, the success of any project depends largely on the encouragement and guidelines of many others. I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of the project.

Firstly, I would like to thank the Principal, **Dr. G. S. V. R. K. CHOUDARY, Bhavan's Vivekananda College**, for having faith in me and thinking that I am capable enough to go on with the project.

Secondly, I would like to thank **Ms. K. V. B. SARASWATHI** Head of the Department of Computer Science, lecturer and other faculty members for their help during my project completion in the college. I take it as a great privilege for having done the project for **Bhavan's Vivekananda College, Sainikpuri**.

Finally, I would like to convey my profound gratitude to my project guide **Mr. D. Ganesh, Lecturer, Department of Computer Science** for giving me this project in the college and enabling me with her valuable guidance and support.

The guidance and support received from all the members who have contributed to this project were vital for the success of the project. I am grateful for their constant support and help.

Y. MANOJ KUMAR (107222468032)

DECLARATION

I hereby declare that the project work entitled “**Agriculture Equipment Rental Management System**” submitted to the Bhavan’s Vivekananda College, Sainikpuri, Secunderabad, is a record of an original work done by me under the guidance of **Mr. D. Ganesh Kumar**, Lecturer, Department of Computer Science, Bhavan’s Vivekananda College, Sainikpuri, Secunderabad.

I further certify that the work has not been submitted to any other Institution for the award of any degree or diploma.

Y. MANOJ KUMAR (107222468032)

ABOUT ORGANIZATION

Bhavan's Vivekananda College of Science, Humanities and Commerce was in 1993 under the aegis of the Bhartiya Vidya Bhavan. Ever since its inspection, the college has been providing quality education to all students in a large variety of courses. The college continues to provide all students a solid foundation for further educational opportunities and the skill for various career opportunities upon graduation.

The college offers a peaceful atmosphere which is ideal for academic pursuits and for the overall development of the students. The college seeks to integrate the student's program of study and the development of skills which includes critical thinking, problem solving, written and oral communication and encouraging research techniques in various fields. It seeks to learn, to adapt and to lead in the creation of a pool of committed and competent individuals indicated to the process of nation building.

The college has a dedicated set of staff, lecturers and support staff. The lecturers, apart from teaching, also imbibe principles of discipline, commitment and hard work and to strive towards achieving the goals. They support the students in all possible ways and help students to solve their various problems and encourage the students to perform well in both academics and other co-curricular activities. Overall, the college, with its holistic atmosphere aided for learning, the principles and values that are imbibed into the students, the excellent staff headed by an able and strong Principle dedicated to making the students the citizens of the future, Bhavan's Vivekananda College of Science, Humanities and Commerce is the college, where we enter in as students and pass out as molded and well-read persons, ready to take on the challenges that come by us in the future.

INDEX

S.NO.	TOPIC	Pg. No.
1.	ABSTRACT	07 - 08
2.	PROBLEM STATEMENT	09
3.	PROPOSED SYSTEM	10 - 11
4.	HARDWARE AND SOFTWARE REQUIREMENTS	12
5.	ABOUT SOFTWARE USED	13 - 18
6.	SYSTEM ANALYSIS	19
6.1	DFD Diagram	19 - 21
7.	SYSTEM DESIGN	22
7.1	Use case Diagrams	22 - 25
7.2	Class Diagrams	26 - 32
7.3	Object Diagrams	33 - 37
7.4	Sequence Diagrams	38 - 41
7.5	Activity Diagrams	42 - 47
8.	SCREENS	48 - 52
9.	SAMPLE CODE	53 - 55
10.	TESTING	56 - 58
11.	IMPLEMENTATION	59
12.	MAINTENANCE	60
13.	CONCLUSION	61
14.	BIBLIOGRAPHY	62

1. ABSTRACT

The Agriculture Equipment Rental Management System is a comprehensive platform designed to streamline the process of renting agricultural equipment. The system addresses the challenges faced by farmers and equipment owner (Admin) in accessing and managing agricultural machinery. By providing an efficient, user-friendly interface, the system facilitates the easy rental of various agricultural tools and machinery, promoting the optimal use of resources and reducing the financial burden on farmers who cannot afford to purchase expensive equipment.

This system allows equipment owner (Admin) to list their machines with detailed descriptions, rental prices, and availability. Farmers can browse the listings, compare prices, and book the equipment that best suits their needs. This system benefits the agricultural community by enhancing the accessibility of modern farming tools, increasing the productivity and efficiency of agricultural operations. Additionally, it promotes the sharing economy within the farming sector, contributing to the sustainable development of agriculture.

This application contains various modules like: -

- **ADMIN MODULE:** -

The Admin Module in the Agriculture Equipment Rental Management System plays a vital role in managing and overseeing the platform's operations. It is responsible for handling user accounts, monitoring rentals, and ensuring smooth communication between farmers and equipment owner (Admin). The admin has access to a centralized dashboard that provides real-time insights into total users, available equipment, and ongoing rentals. This module streamlines administrative tasks, enhances operational efficiency, and ensures a secure and transparent equipment rental process.

- **USER MODULE: -**

The User Module in the Agriculture Equipment Rental Management System is designed to facilitate seamless interactions between farmers and equipment owner (Admin). It provides a user-friendly interface where users can register, log in, and access various platform features. For farmers, the system enables them to browse available equipment, view detailed specifications, check availability, and submit rental requests. Renters can also check their rental status and completed rentals and communicate with equipment owner (Admin) for inquiries. This module ensures a smooth and efficient user experience, making it easier for farmers and equipment owner (Admin) to connect, rent equipment, and contribute to the agricultural ecosystem.

2. PROBLEM STATEMENT

Farmers, particularly small and marginal farmers, often face challenges in accessing high-quality and advanced agricultural equipment due to the high costs of ownership. The lack of affordable options for accessing modern tools negatively impacts their productivity, efficiency, and overall yield. Many farmers resort to outdated methods, which can result in lower profitability and slower agricultural growth.

Moreover, the absence of a streamlined booking and management system leads to inefficiencies such as delayed responses, equipment mismanagement, and increased costs for farmers. A technology-driven solution is needed to streamline the equipment rental process, ensuring accessibility, efficiency, and optimal resource utilization. A digital Agriculture Equipment Rental Management System can bridge the gap between farmers and equipment owner (Admin) by offering a structured, transparent, and user-friendly platform to facilitate seamless equipment rentals.

Benefits of implementing agriculture equipment rental management system: -

- **Equipment Booking System:**
 - Enables farmers to rent equipment easily without manual coordination.
 - Displays a list of available agriculture equipment, ensuring users can book equipment without conflicts.
- **Role-Based Access Control:**
 - Ensures that only authorized users (farmers) and Admin can access relevant data.
 - Enhances platform security and prevents unauthorized modifications.
- **Cost Efficiency:**
 - Reduces operational costs by eliminating the need for middlemen.
 - Minimizes paperwork and manual efforts through digital record-keeping.
- **Data accuracy:**
 - Prevents errors in rental records, equipment availability, and bookings.
 - It will prevent double bookings of equipment.
- **Data security:**
 - Protects user and equipment data from unauthorized access, modification, or disclosure.
 - Restricts access through role-based access control.
- **Improved Equipment Utilization:**
 - Helps farmers access modern equipment without ownership costs, improving overall productivity.
 - Increases profitability for equipment owners by maximizing rentals.

3. PROPOSED SYSTEM

Based on the problem statement and the desired features, here's a proposed Agriculture Equipment Rental Management System (AERMS):

1. User Module: -

- a. **Equipment Booking:** Farmers can browse and book available equipment based on their needs.
- b. **Real-Time Availability:** Users can check equipment availability before making a booking.
- c. **Booking Confirmation:** User can track whether the booking has been approved or disapproved by the admin.

2. Administrative Dashboard: -

- a. **Data Security:** Ensures sensitive user information is protected with role-based access restrictions.
- b. **Equipment Management:** Admin can approve listings, update details, and track rented equipment.
- c. **Cost Management:** Helps in managing costs related to equipment maintenance and platform operations.
- d. **Performance Monitoring:** Tracks system performance, rental activity, and platform usage to ensure smooth operation and identify areas for improvement.

3. Reporting and Analytics: -

- a. **Data Accuracy:** Eliminates human errors through automated data entry and tracking.
- b. **Rental Insights:** Generates reports on rental history, user activity, and equipment demand.
- c. **Cost Efficiency:** Reduces paperwork and manual efforts with a centralized system.
- d. **Equipment Utilization Reports:** Helps admin analyze the usage frequency of their equipment.

4. **System Integration:** -

- a. **Cloud-Based Storage:** Ensures data is stored securely and accessible from anywhere.
- b. **API Support:** Allows integration with third-party services for enhanced functionality.

5. **Scalability and Flexibility:** -

- a. **Cloud-Based Deployment:** Enables seamless scalability without physical server costs.
- b. **Customization:** Allows modifications based on different agricultural regions and rental policies.
- c. **Multi-Language Support:** Supports different languages for wider accessibility.

6. **Equipment Listings:** -

- a. **Equipment Addition:** Equipment owners (Admin) can add equipment details such as type, model, condition, rental price, and availability.
- b. **Equipment Images and Descriptions:** Admin can upload images and provide detailed descriptions to help farmers make informed decisions.

By implementing this **AERMS**, the platform can streamline agricultural equipment rentals, enhance cost efficiency, and improve accessibility for both farmers and equipment owner (Admin), leading to increased agricultural productivity.

4. HARDWARE AND SOFTWARE REQUIREMENTS

- **HARDWARE REQUIREMENTS: -**
 - **For Input** : Keyword, Mouse
 - **For Output** : Monitor
 - **RAM Size** : 2GB Minimum or 8GB Preferred.
 - **Processor** : 2.0GHz
 - **Storage** : Minimum of 256 GB HDD or higher.

- **SOFTWARE REQUIREMENTS: -**
 - **Operating System** : All OS Which has browser support.
 - **Tool** : Microsoft Visual Studio Code
 - **Front End Software** : HTML, CSS, JavaScript
 - **Backend Language** : PHP and MySQL

5. ABOUT SOFTWARE USED

In today's digital age, the Internet has become an integral part of our lives, connecting thousands of computers worldwide. As network connections continue to grow rapidly, so does the volume of network traffic. However, these computers come from various manufacturers, operate on different systems, and vary in architecture and computing power.

Recognizing the challenges posed by this heterogeneous environment, SUN Microsystems Corporation identified the need for a versatile programming language. Java emerged as the solution to bridge the gap between disparate systems, chips, and operating platforms. With Java, applications gain compatibility across all operating systems, ensuring seamless functionality regardless of the underlying infrastructure.

- **HTML: -**

HTML, short for Hypertext Markup Language, serves as the standard markup language for documents intended for web browser display. It can be complemented by technologies like Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers retrieve HTML documents from a web server or local storage and translate them into multimedia web pages. HTML defines the structural semantics of a webpage and initially included instructions for document appearance.

HTML elements form the foundation of HTML pages. These constructs enable embedding images, interactive forms, and other objects into the rendered page. HTML facilitates the creation of structured documents by specifying semantic structures for text elements like headings, paragraphs, lists, links, quotes, and more. Tags, enclosed in angle brackets, delineate HTML elements. Tags like `` and `<input />` directly incorporate content into the page, while others like `<p>` encompass and provide context for document text, potentially containing additional tags as sub-elements. Although browsers do not visibly display HTML tags, they utilize them to interpret page content.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997. A form of HTML, known as HTML5, is used to display video and audio, primarily using the <canvas> element, in collaboration with JavaScript.

- **CSS:** -

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .CSS file, which reduces complexity and repetition in the structural content; and enable the .CSS file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

- **JavaScript:**

JavaScript is a versatile, high-level programming language primarily used for web development. It enables interactive and dynamic user experiences by controlling webpage behavior, handling events, and manipulating the Document Object Model (DOM).

Alongside HTML and CSS, JavaScript is one of the core technologies of the web, allowing developers to create responsive interfaces, animations, and real-time updates. It is an interpreted language with dynamic typing and supports asynchronous programming through callbacks, Promises, and async/await. With the advent of modern frameworks like React, Angular, and Vue.js, JavaScript has become essential for building complex web applications.

Beyond frontend development, JavaScript is widely used on the server side with Node.js, allowing developers to build full-stack applications using a single language. It also extends into mobile app development (React Native), game development (Phaser.js), and even machine learning (TensorFlow.js). JavaScript's ecosystem is rich with libraries and tools that streamline development and enhance functionality. As a cross-platform language, it runs in all major browsers without requiring compilation, making it an accessible and powerful tool for developers worldwide.

- **PHP:** -

PHP, initially standing for Personal Home Page, is a versatile scripting language primarily designed for web development. It was originally crafted by Danish-Canadian programmer Rasmus Lerdorf in 1994 and is currently maintained by The PHP Group. Over time, the acronym PHP has evolved to represent PHP: Hypertext Preprocessor.

Typically, PHP code is processed on a web server by a PHP interpreter, which may function as a module, daemon, or CGI executable. Upon interpretation and execution, PHP code generates various forms of data, such as HTML or binary image data, forming part or the entirety of an HTTP response. Numerous web template systems, content management systems, and frameworks exist to facilitate the generation of these responses.

Beyond web development, PHP finds utility in diverse programming tasks, extending to standalone graphical applications and even robotic drone control. Moreover, PHP code can be executed directly from the command line, showcasing its versatility across different contexts.

The PHP interpreter, powered by the Zend Engine, is open-source software released under the PHP License. PHP boasts extensive portability and can be deployed across a wide array of web servers and operating systems.

Until 2014, PHP evolved without a written formal specification or standard, relying on the original implementation as the de facto benchmark for other implementations. However, efforts have been underway since 2014 to establish a formal PHP specification.

According to W3Techs' data as of January 2022, PHP is utilized by 78.1% of websites employing server-side programming languages. Among PHP versions, 7.4 holds the majority share, with support for version 7.3 being discontinued on December 6, 2021.

- **MySQL:** -

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Wideness's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage

system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Wideness forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter and YouTube.

6. SYSTEM ANALYSIS

Systems analysis is the process by which an individual (s) studies a system such that an information system can be analyzed, modeled, and a logical alternative can be chosen. Systems analysis projects are initiated for three reasons: problems, opportunities, and directives. The people involved include systems analysts, sponsors, and users. The process by which systems are developed can be described by the systems development life cycle.

6.1 ABOUT DATA FLOW DIAGRAM (DFD)

A Data Flow diagram is a graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed.

The transformation of the data from input to output, through processing, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagram.

The physical data flow diagrams show the actual implements and movement of the data between people, department and workstation. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name.

Process Notations. A process transforms incoming data flow into outgoing data flow.



Datastore Notations. Datastores are repositories of data in the system. They are 15, sometimes also referred to as files.

Data Flow Diagram Level - 0 for (AERMS):

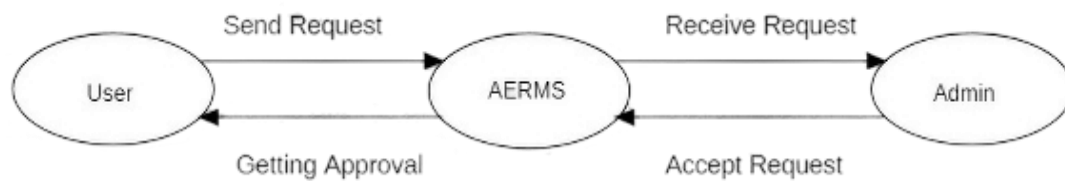


Fig: LEVEL - 0 Diagram for AERMS

Data Flow Diagram Level – 1 for (AERMS):

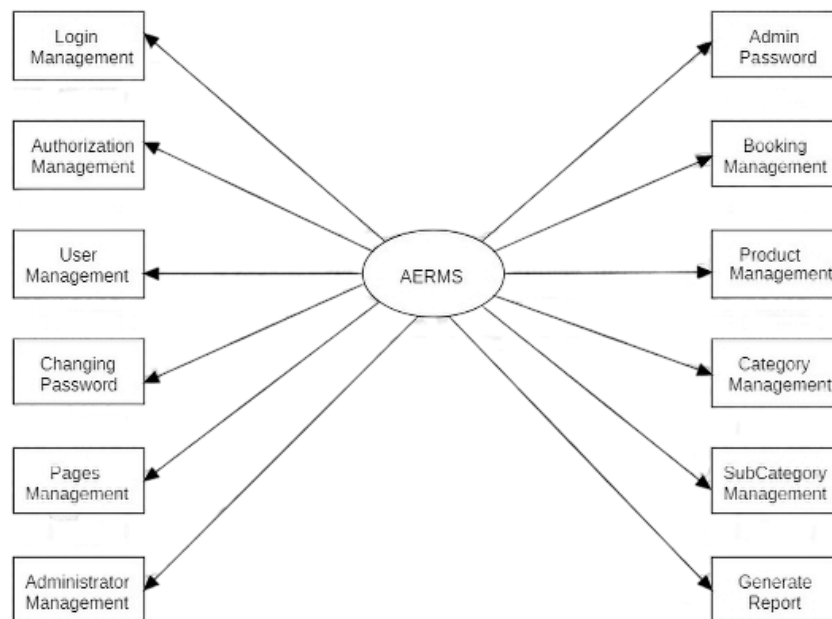


Fig: LEVEL - 1 Diagram for AERMS

Data Flow Diagram Level – 2 for (AERMS):

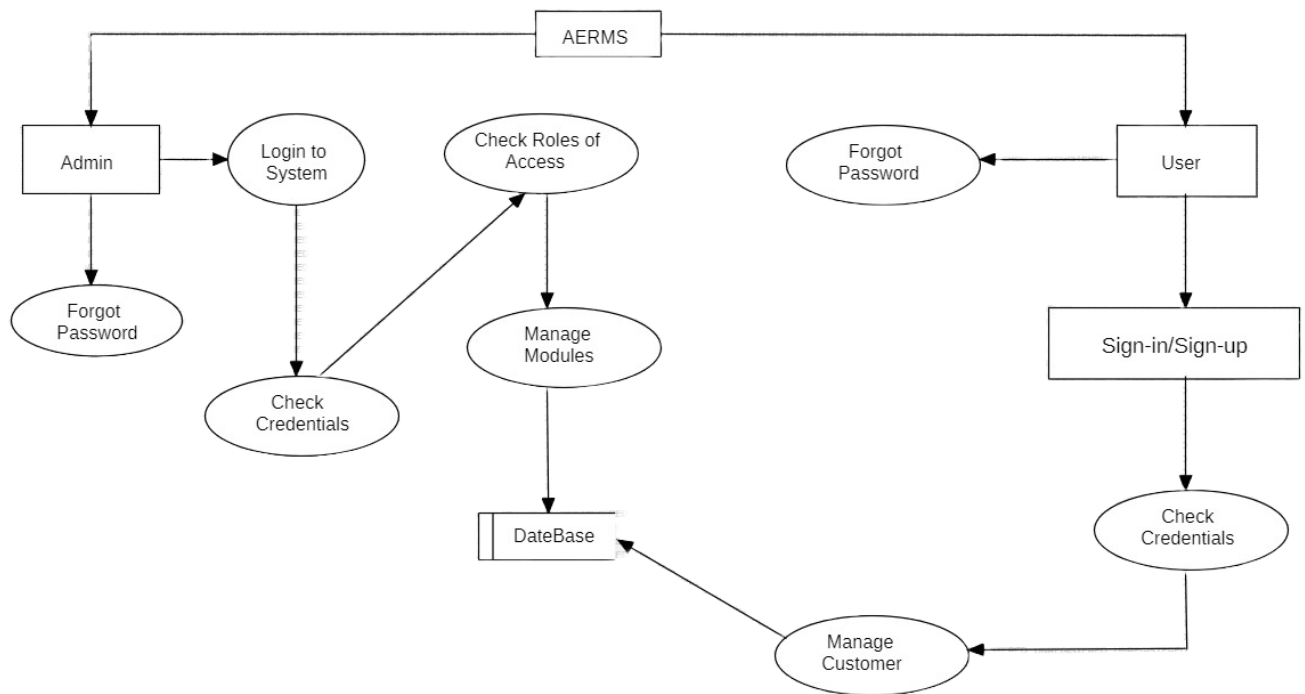


Fig: LEVEL - 2 Diagram for AERMS

7. SYSTEM DESIGN



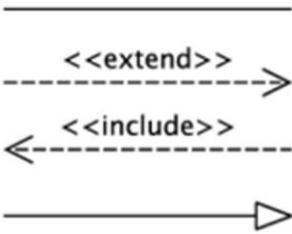
ABOUT USE CASE DIAGRAMS

In Unified Modelling Language (UML), use case diagrams are used to show the interactions between actors and a system. The system can be a website or another similar entity being developed. An actor is a person or entity that interacts with the system - normally, this is a user. In essence, use case diagrams depict the system, who is using it, how they are using it, or what they are doing. On a macro scale, the information you can glean from use case diagrams allow you to set the scope of your system or application and what goals it can help the actors achieve.

Use case diagrams are typically used in the following scenarios:

- Depicting the scope and goal of the interactions between the actor and the system.
- Modeling the interactions between actors and the system, as well as the general flow of events.
- Identifying internal or external factors that might impact the outcome of interaction and, by extension, the system itself.

Use Case Diagram Symbols:

Symbol	Reference Name
	Actor
	Use case
	Relationship

The notation for a use case diagram is pretty straightforward and doesn't involve as many types of symbols as other UML diagrams.

- **Use cases:** Horizontally shaped ovals that represent the different uses that a user might have.
- **Actors:** Stick figures that represent the people actually employing the use cases.
- **Relationship:** A line between actors and use cases. In complex diagrams, it is important to know which actors are associated with which use cases.

7.1. Use Case Diagram for AERMS:

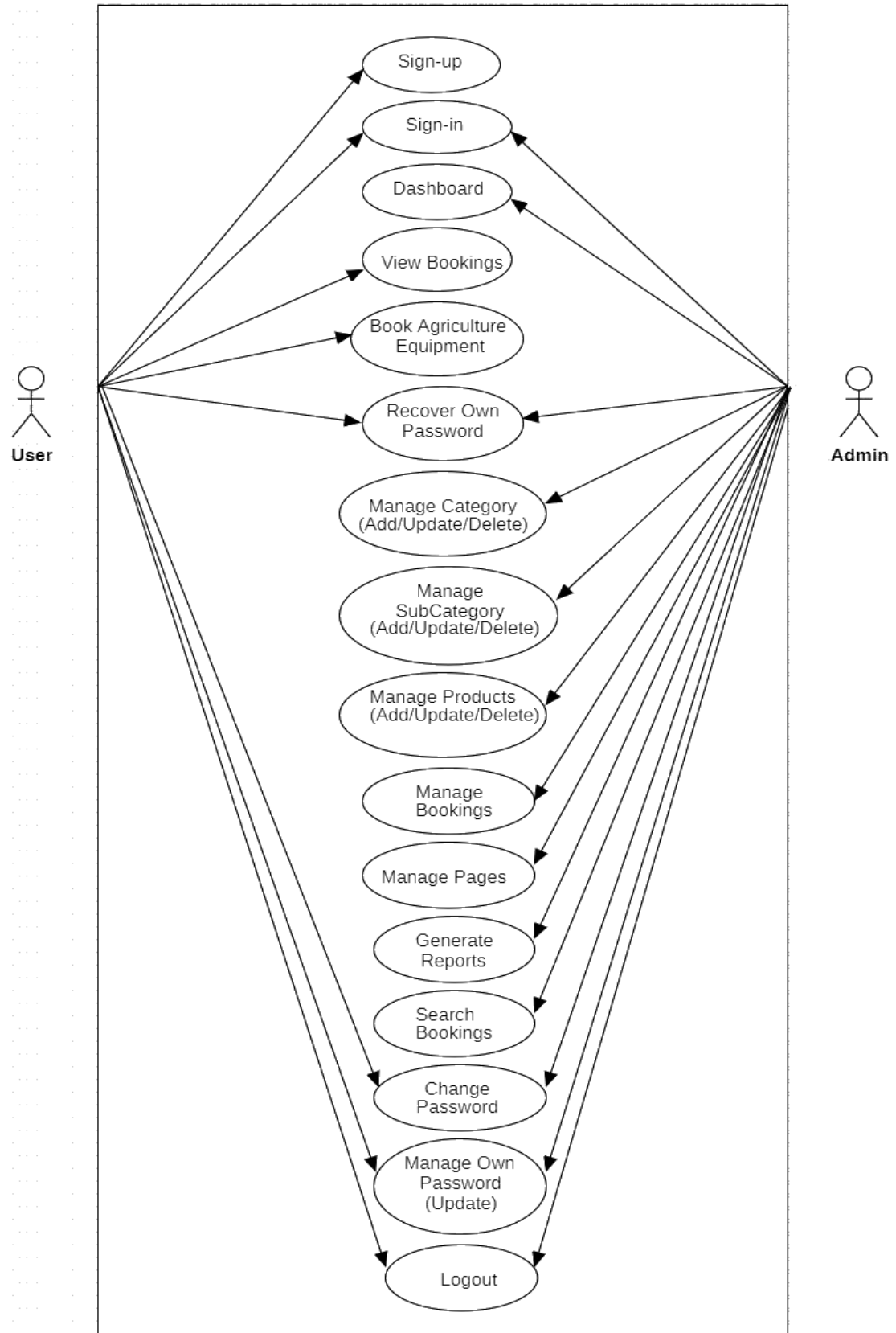


Fig 7.1 Use Case Diagram for AERMS

- The above use case diagram describes the roles of users i.e., farmers and admin in the system.
- New users (farmers) can register, while registered users can log in directly to the platform.
- Farmers can browse available equipment, request rentals, and view their booking history.
- Admin can list their equipment, manage availability, and approve or disapprove rental requests.
- Admin can verify new equipment listings, update availability, and handle disputes or maintenance requests.

ABOUT CLASS DIAGRAMS

The UML Class diagram is a graphical notation used to construct and visualize object-oriented systems. A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's:

- classes,
- their attributes,
- operations (or methods),
- and the relationships among objects.

Class: -

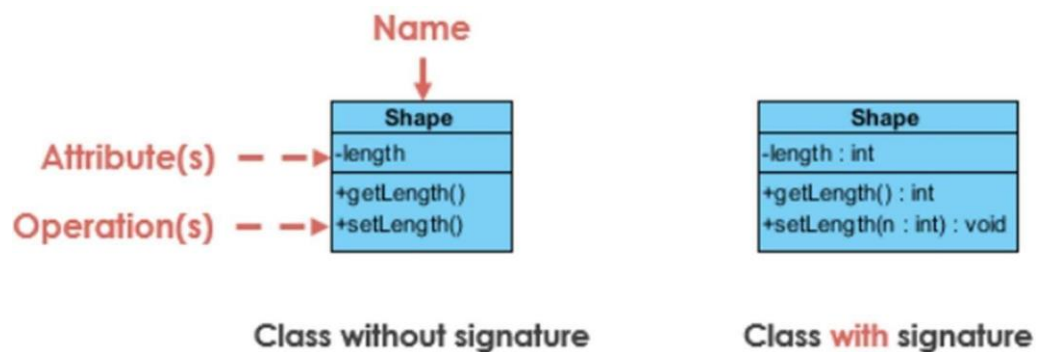
A Class is a blueprint for an object. Objects and classes go hand in hand. We can't talk about one without talking about the other. And the entire point of Object- Oriented.

Design is not about objects, it's about classes, because we use classes to create objects.

So, a class describes what an object will be, but it isn't the object itself.

UML Class Notation: -

A class represents a concept which encapsulates state (attributes) and behavior (operations). Each attribute has a type. Each operation has a signature. The class name is the only mandatory information.



Class Name: -

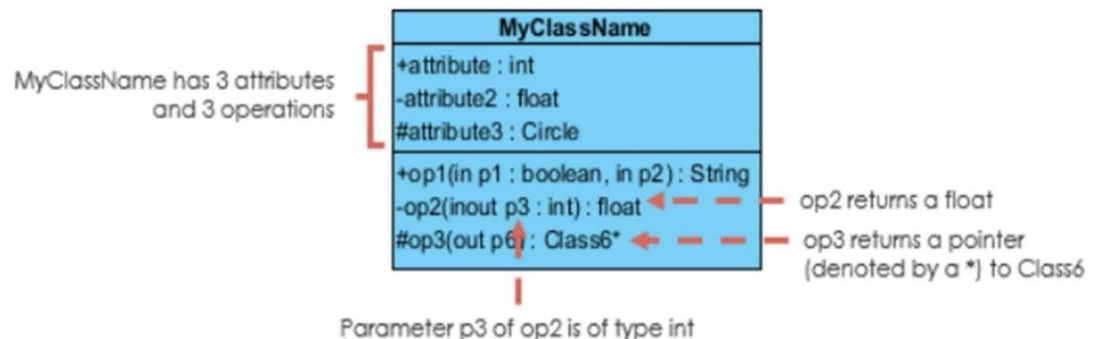
- The name of the class appears in the first partition.

Class Attributes: -

- Attributes are shown in the second partition.
- The attribute type is shown after the colon.
- Attributes map onto member variables (data members) in code.

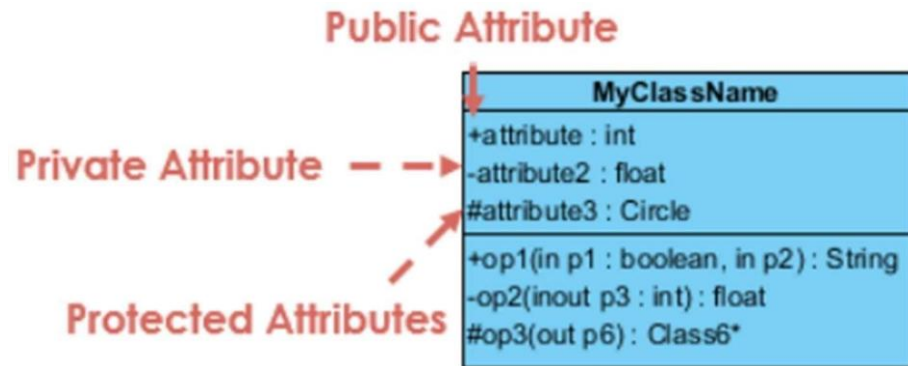
Class Operations (Methods): -

- Operations are shown in the third partition. They are services the class provides.
- The return type of a method is shown after the colon at the end of the method signature.
- The return type of method parameters is shown after the colon following the parameter name. Operations map onto class methods in code.

**Class Visibility: -**

The +, - and # symbols before an attribute and operation name in a class denote the visibility of the attribute and operation.

- + denotes public attributes or operations
- - denotes private attributes or operations
- # denotes protected attributes or operations.

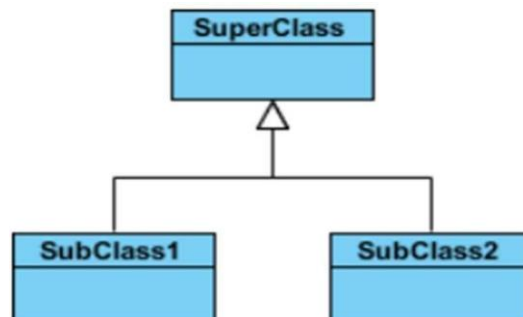


Relationships between classes: -

I) Inheritance or Generalization:

A generalization is a relationship between a more general classifier and a more specific classifier. Each instance of the specific classifier is also an indirect instance of the general classifier. Thus, the specific classifier inherits the features of the more general classifier.

The figure below shows an example of inheritance hierarchy. SubClass1 and SubClass2 are derived from Super Class. The relationship is displayed as a solid line with a hollow arrowhead that points from the child element to the parent element.



Association:

Associations are relationships between classes in a UML Class Diagram. They are represented by a solid line between classes. Associations are typically named using a verb or verb phrase which reflects the real-world problem domain.

Simple Association:

- A structural link between two peer classes.
- There is an association between Class1 and Class2.

**Cardinality:**

Cardinality is expressed in terms of:

- one to one
- one to many
- many to many

Aggregation:

A special type of association.

- It represents a "part of" relationship.
- Class2 is part of Class1.
- Many instances (denoted by the *) of Class2 can be associated with Class1.
- Objects of Class1 and Class2 have separate lifetimes.

The figure below shows an example of aggregation. The relationship is displayed as a solid line with an unfilled diamond at the association end, which is connected to the class that represents the aggregate.

**II) Composition: -**

A special type of aggregation where parts are destroyed when the whole is destroyed.

- Objects of Class2 live and die with Class1.
- Class2 cannot stand by itself.

The figure below shows an example of composition. The relationship is displayed as a solid line with a filled diamond at the association end, which is connected to the class that represents the whole or composite.



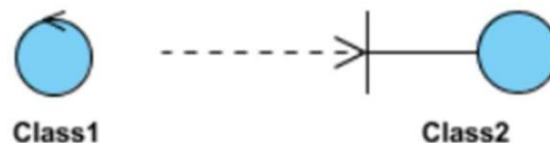
III) Dependency: -

An object of one class might use an object of another class in the code of a method. If the object is not stored in any field, then this is modeled as a dependency relationship.

A special type of association.

- Exists between two classes if changes to the definition of one may cause changes to the other (but not the other way around).
- Class1 depends on Class2

The figure below shows an example of dependency. The relationship is displayed as a dashed line with an open arrow.



The figure above shows another example of dependency. The Person class might have a has Read method with a Book parameter that returns true if the person has read the book (perhaps by checking some databases).

IV) Realization: -

Realization is a relationship between the blueprint class and the object containing its respective implementation level details. This object is said to realize the blueprint class.

In other words, you can understand this as the relationship between the interface and the implementing class.

For example, the Owner interface might specify methods for acquiring property and disposing of property. The Person and Corporation classes need to implement these methods, possibly in very different ways.

7.2. Class Diagram for AERMS:

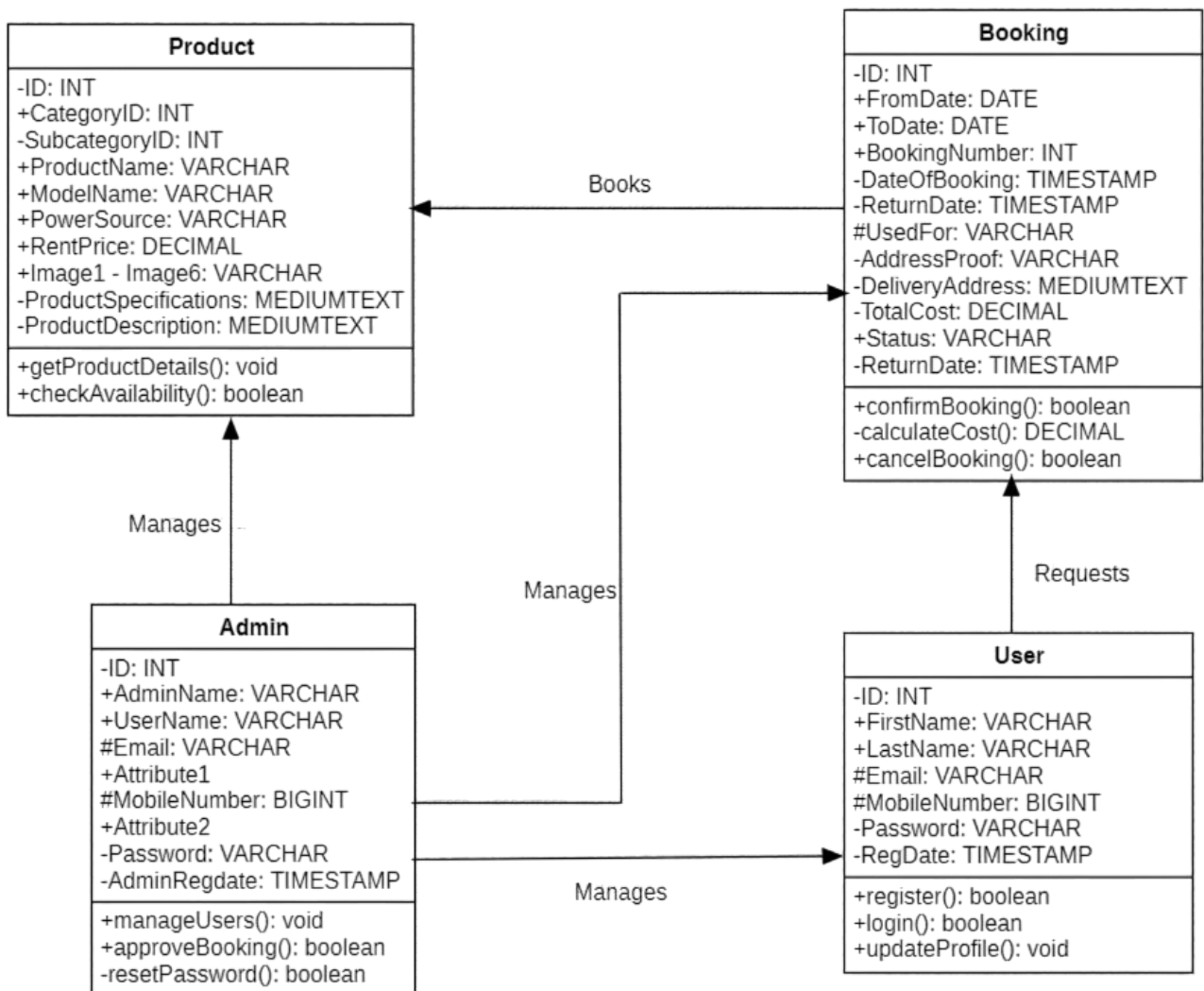


Fig 7.2 Class Diagram for AERMS

The above class diagram describes the structure of the Agriculture Equipment Rental Management System, including its classes, attributes, and operations.

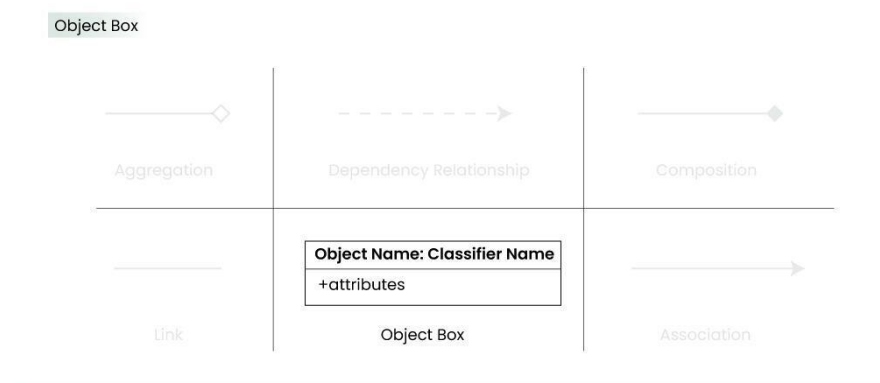
The classes in this module are Register, Login, User, Equipment, Booking and AERMS.

ABOUT OBJECT DIAGRAMS

Object diagrams are a visual representation in UML (Unified Modeling Language) that illustrates the instances of classes and their relationships within a system at a specific point in time. They display objects, their attributes, and the links between them, providing a snapshot of the system's structure during execution. Since object diagrams depict behavior when objects have been instantiated, we can study the behavior of the system at a particular instant.

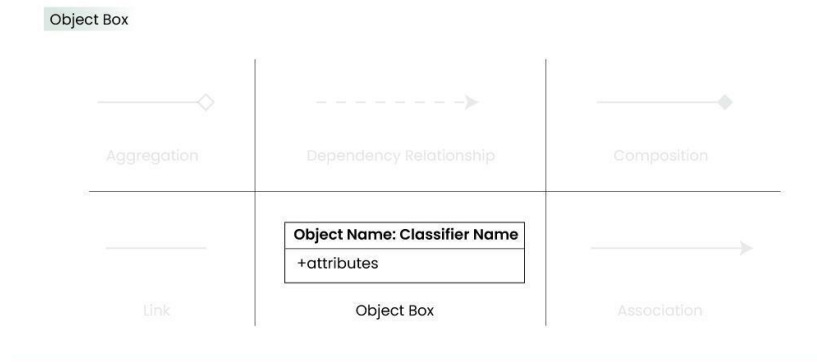
Basic Object Diagram Notations:

The object diagram in UML uses specific notations to represent instances of classes and their relationships at a particular moment in time.



i) Objects or Instance specifications:

When we instantiate a classifier in a system, the object we create represents an entity which exists in the system. We can represent the changes in object over time by creating multiple instance specifications. We use a rectangle to represent an object in an object diagram.

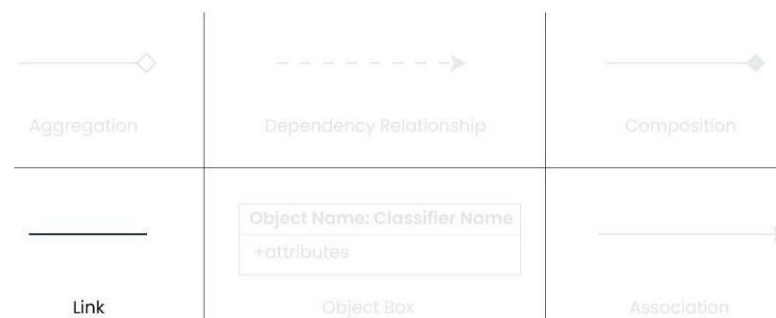


ii) **Attributes and Values:**

Inside the object box, attributes of the object are listed along with their specific values

iii) **Link:**

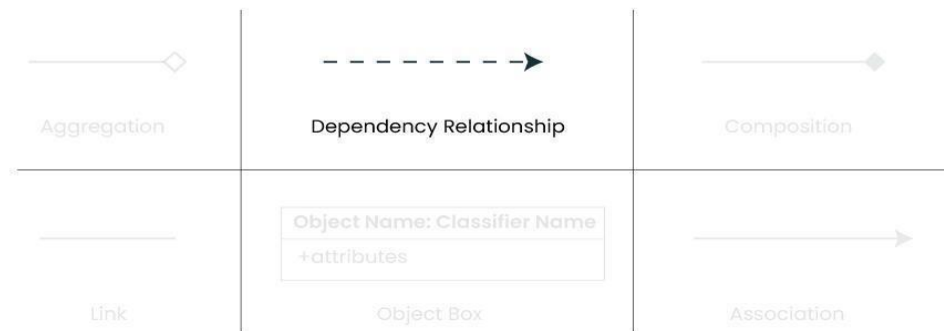
We use a link to represent a relationship between two objects. We represent the number of participants on the link for each, at the end of the link. The term link is used to specify a relationship between two instance specifications or objects. We use a solid line to represent a link between two objects.



iv) **Dependency Relationship:**

We use a dependency relationship to show when one element depends on another element. A dependency is used to depict the relationship between dependent and independent entities in the system.

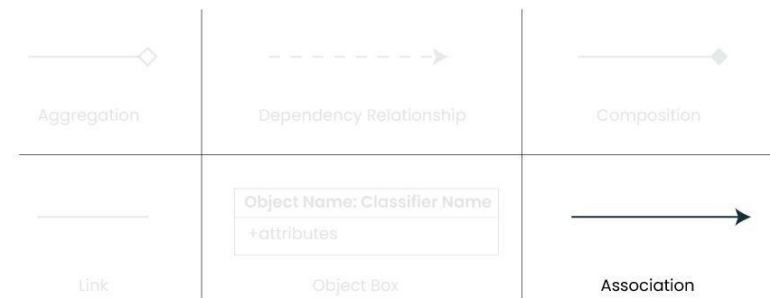
Dependency Relationship



v) **Association:**

Association is a reference relationship between two objects (or classes). An association line connects two object boxes, representing a relationship between instances of two classes. We use association when one object references members of the other object. Association can be uni-directional or bi-directional. We use an arrow to represent association.

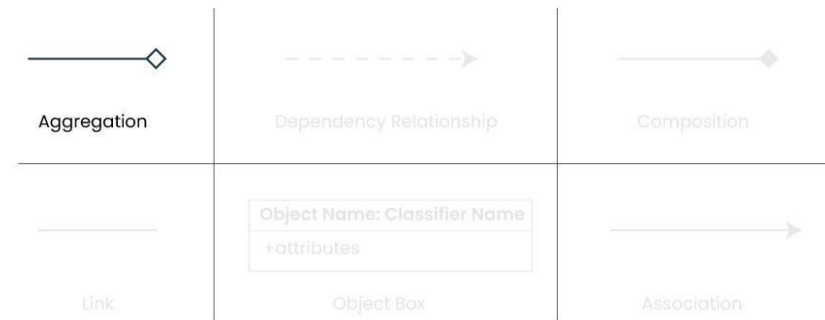
Association



vi) **Aggregation:**

Aggregation represents a “has a” relationship. We use a hollow diamond on the containing object with a line which joins it to the contained object.

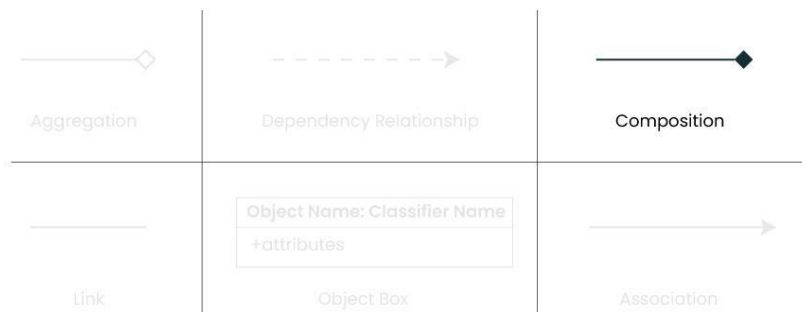
Aggregation



vii) **Composition:**

Composition is a type of association where the child cannot exist independent of the other. We use a filled diamond on the containing object with a line which joins it to the contained object. Composition is also a special type of association. It is also a kind of parent child relationship but it is not inheritance. So whenever independent existence of the child is not possible, we use a composition relationship.

Composition



7.3. Object Diagram for AERMS:

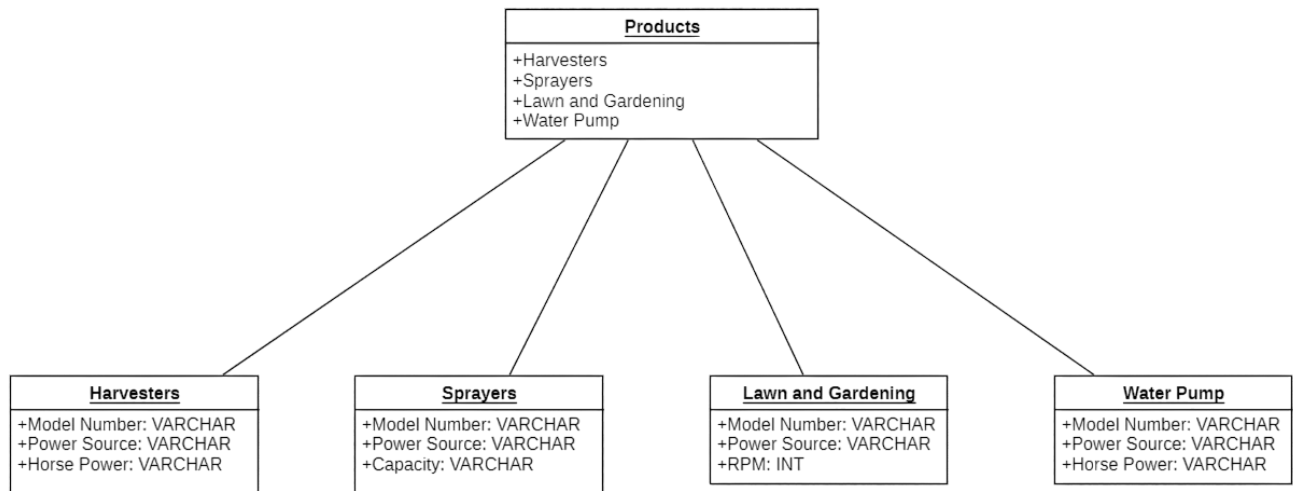


Fig 7.3 Object Diagram for AERMS

- The object diagram provides a snapshot of how different agricultural equipment objects interact within the system, helping to visualize the real-time structure of the equipment available for rent.
- These objects have specific attributes such as Model Number, Power Source, Horse Power, Capacity, and RPM, which define the characteristics of the equipment.
- Each product category has individual objects such as Harvesters, Sprayers, Lawn and Gardening Equipment, and Water Pumps, which inherit attributes from the Products class.
- The Products object contains different categories of agricultural equipment, including Harvesters, Sprayers, Lawn and Gardening tools, and Water Pumps etc.

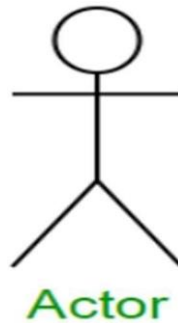
ABOUT SEQUENCE DIAGRAM

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

Basic Sequence Diagram Notations:

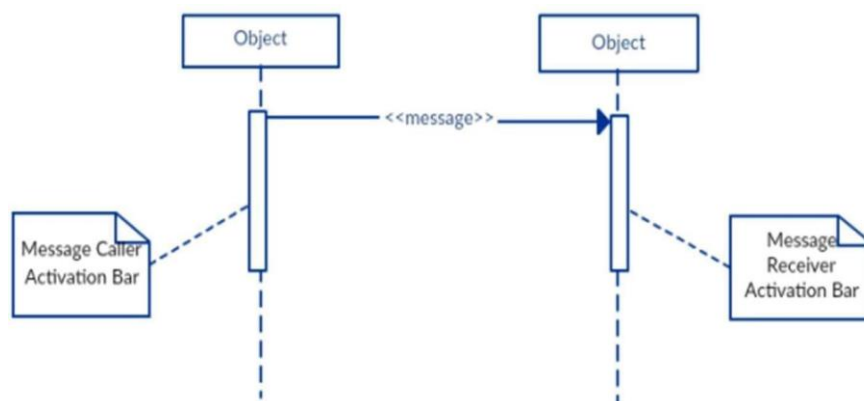
i) Actor symbol:

Shows entities that interact with or are external to the system.



ii) Activation:

Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifeline.

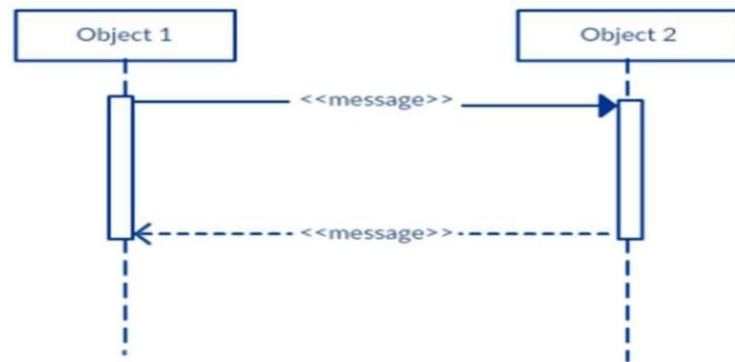


iii) Messages:

Messages are arrows that represent communication between objects. We use half arrowed lines to represent asynchronous messages. Asynchronous messages are sent from an object that will not wait for a response from the receiver before continuing its tasks.

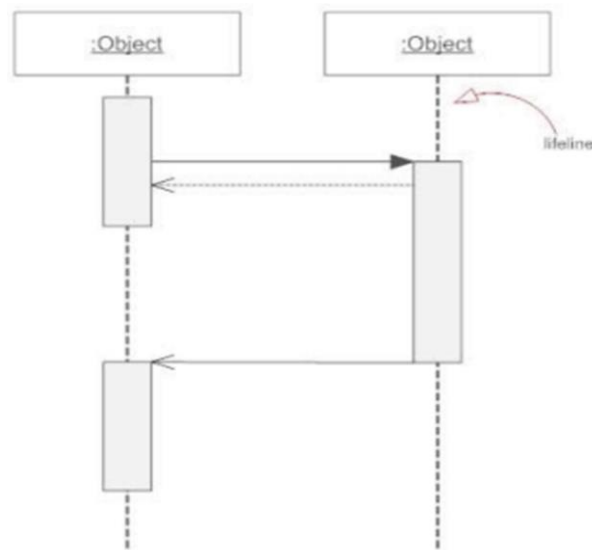
iv) Return message:

A return message is used to indicate that the message receiver is done processing the message and is returning control over to the message caller. Return messages are optional notation pieces, for an activation bar that is triggered by a synchronous message always implies a return message.



v) Lifelines:

Lifelines are vertical dashed lines that indicate the object's presence over time.



vi) Comment:

UML diagrams generally permit the annotation of comments in all UML diagram types.

The comment object is a rectangle with a folded-over corner as shown below. The comment can be linked to the related object with a dashed line.

**vii) Loops:**

A repetition or loop within a sequence diagram is depicted as a rectangle. Place the condition for exiting the loop at the bottom left corner in square brackets [].

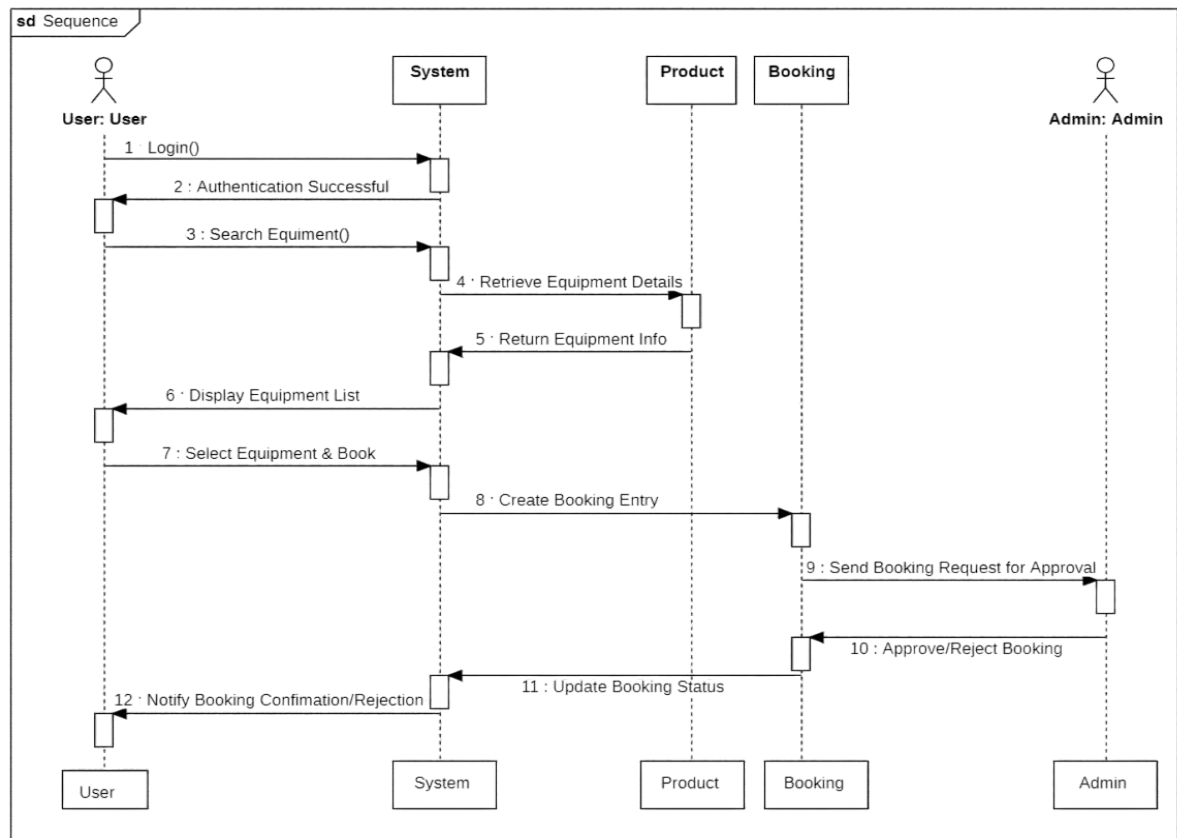
SEQUENCE DIAGRAM for AERMS:

Fig 7.4 SEQUENCE DIAGRAM for AERMS

- This sequence diagram illustrates the interaction between objects in the Agriculture Equipment Rental Management System, including Farmers, Admin, System Interface, and Database.
- The diagram describes how users (farmers) and (Admin) can log in, browse available equipment, and request rentals.
- Farmers can search for equipment, check availability, and place rental requests.
- Admin receives rental requests and can approve or disapprove them.
- The system interface processes these interactions and updates the database accordingly.
- The admin can monitor and manage equipment listings, user activities, and disputes when necessary.

ABOUT ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams: The basic purpose of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but the activity diagram is used to show message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

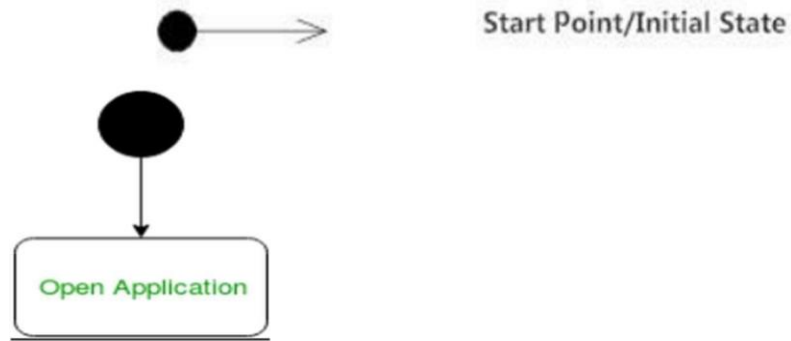
The purpose of an activity diagram can be described as : –

- Draw the activity flow of a system.
- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

Components of an Activity Diagram: -

i) Initial State or Start Point:

A small filled circle followed by an arrow represents the initial action state or the start point for any activity diagram.



ii) Activities:

The categorization of behavior into one or more actions is termed as an activity. In other words, it can be said that an activity is a network of nodes that are connected by edges.

The edges depict the flow of execution. It may contain action nodes, control nodes, or object nodes.

The control flow of activity is represented by control nodes and object nodes that illustrates the objects used within an activity. The activities are initiated at the initial node and are terminated at the final node.



iii) Fork:

Fork nodes are used to support concurrent activities.

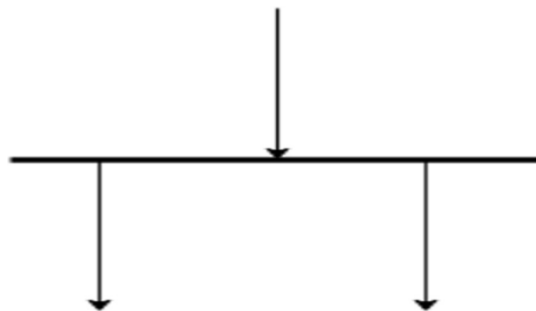


Figure – fork notation

When we use a fork node when both the activities get executed concurrently i.e., no decision is made before splitting the activity into two parts. Both parts need to be.

We use a rounded solid rectangular bar to represent a Fork notation with incoming arrows from the parent activity state and outgoing arrows towards the newly created activities.

For example: In the example below, the activity of making coffee can be split into two concurrent activities and hence we use the fork notation.

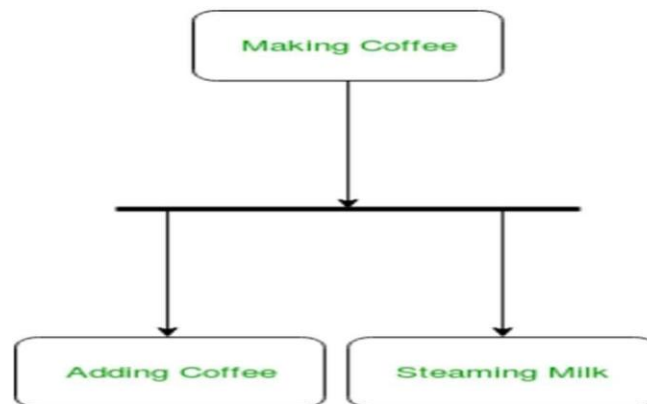
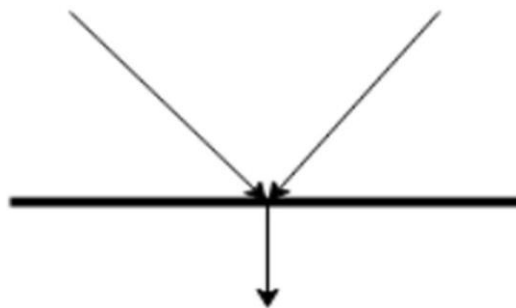


Figure – a diagram using fork

iv) **Join:**

Join nodes are used to support concurrent activities converging into one.

For join notations we have two or more incoming edges and one outgoing



edge.

Figure – join notation

For example – When both activities i.e., steaming the milk and adding coffee get completed, we converge them into one final activity.

v) Action Flow:

Action flows, also called edges and paths, illustrate the transitions from one action state to another. They are usually drawn with an arrowed line.



vi) Decisions and Branching:

When we need to make a decision before deciding the flow of control, we use the decision node.

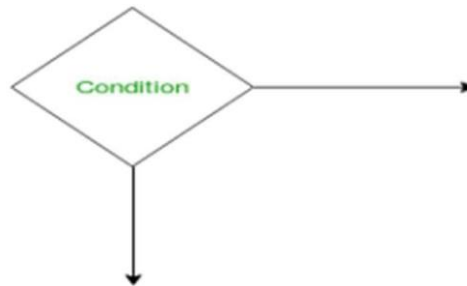
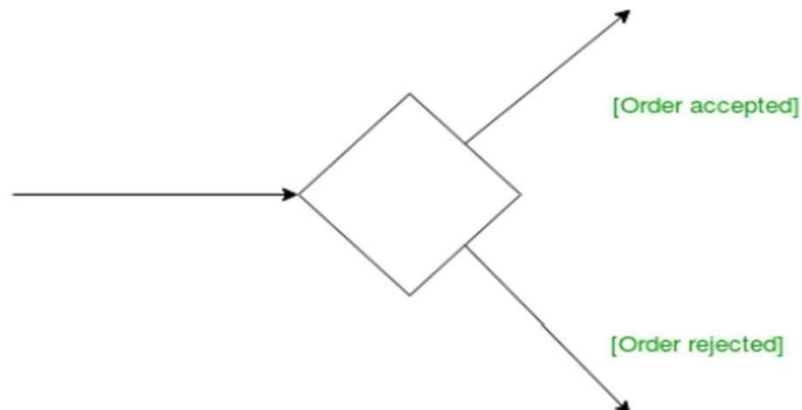


Figure – notation for decision node

The outgoing arrows from the decision node can be labeled with conditions or guard expressions. It always includes two or more output arrows.

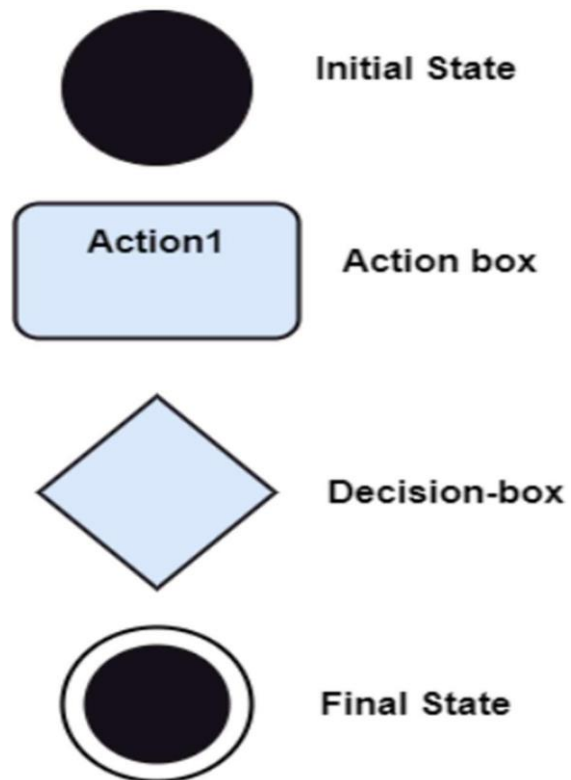
vii) Guards:

A Guard refers to a statement written next to a decision node on an arrow sometimes within square brackets.



viii) Final State or End Point:

An arrow pointing to a filled circle nested inside another circle represents the final action state.



Activity Diagram for AERMS:

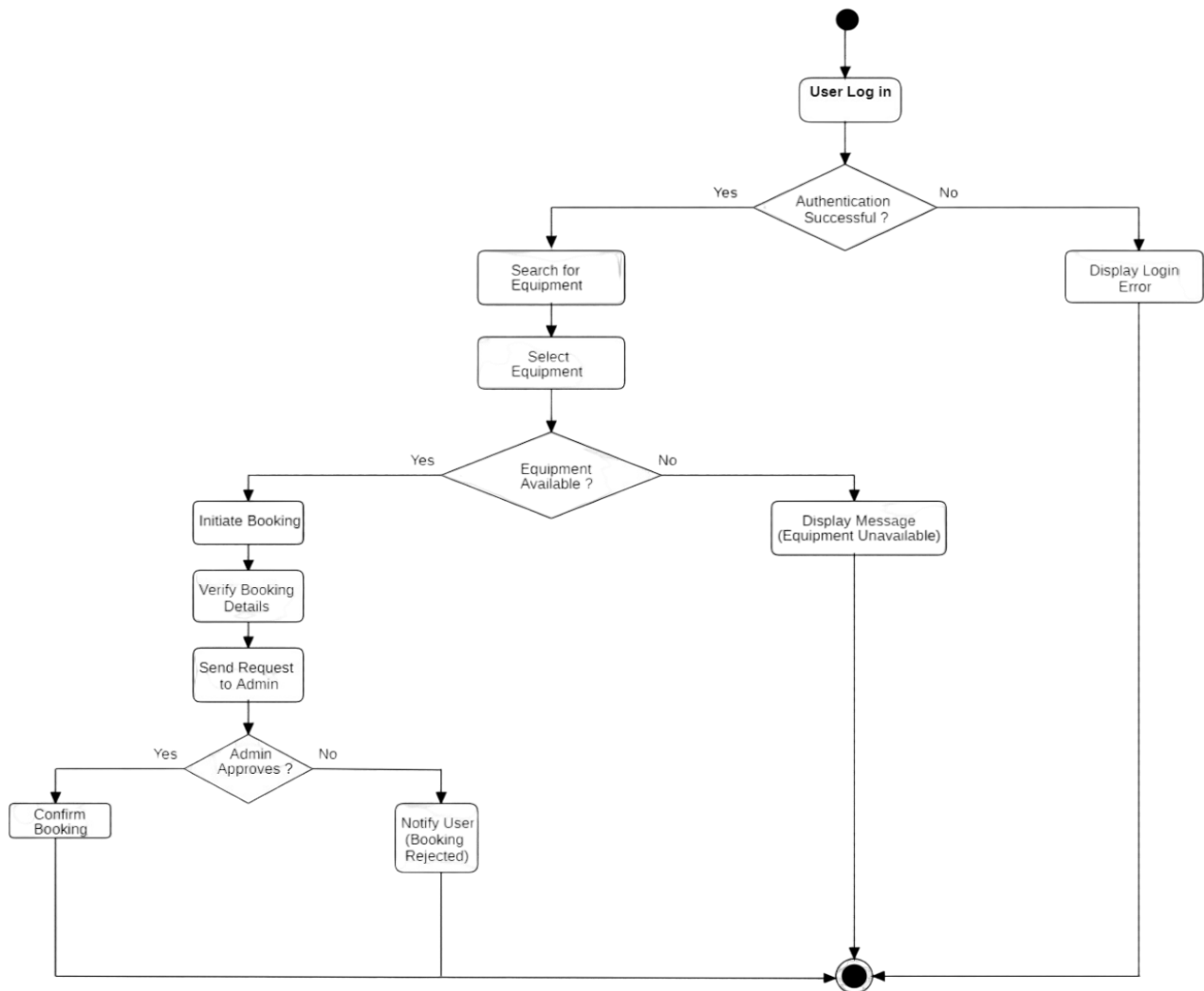


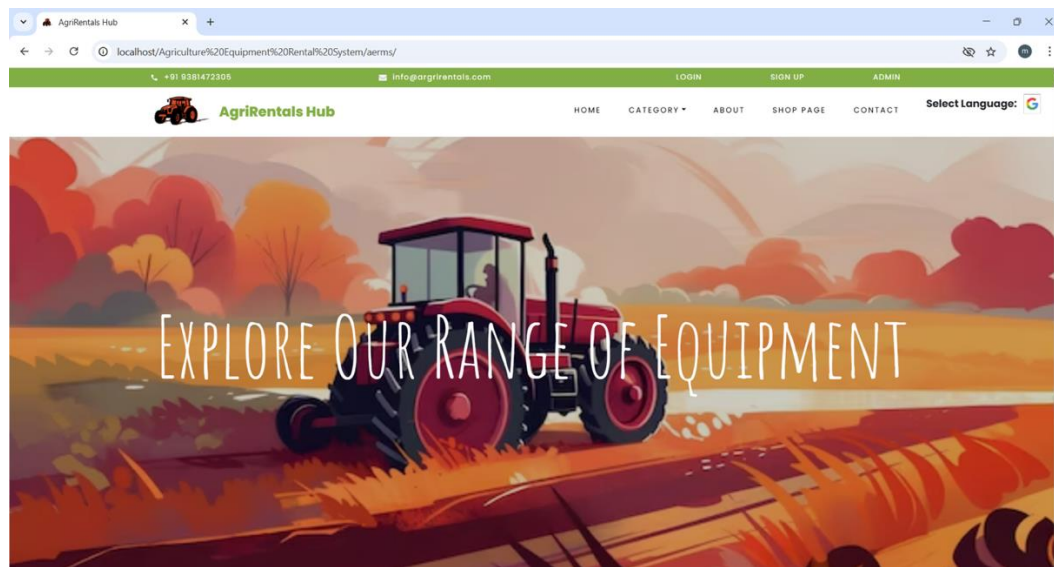
Fig 7.5 Activity Diagram for AERMS

- This activity diagram illustrates the workflow of the Agriculture Equipment Rental Management System, showing the sequence of actions users take within the system.
- The first step is logging in, where users (farmers) enter their registered username and password. If the credentials do not match, the system displays an "Invalid username or password" message. If the credentials are correct, the user is successfully logged in.
- After login, a farmer can search for available equipment by filtering based on type and price.
- If suitable equipment is found, the farmer can request a rental. If the requested equipment is unavailable, the system suggests alternative available equipment or allows the farmer to modify the search criteria.
- Equipment owners can approve or disapprove rental requests based on availability and renter eligibility.

8. SCREENS

- **HOME PAGE:** -

This is the homepage of the Agriculture Equipment Rental Management System. It provides easy navigation for users to browse, rent, and manage agricultural machinery efficiently.

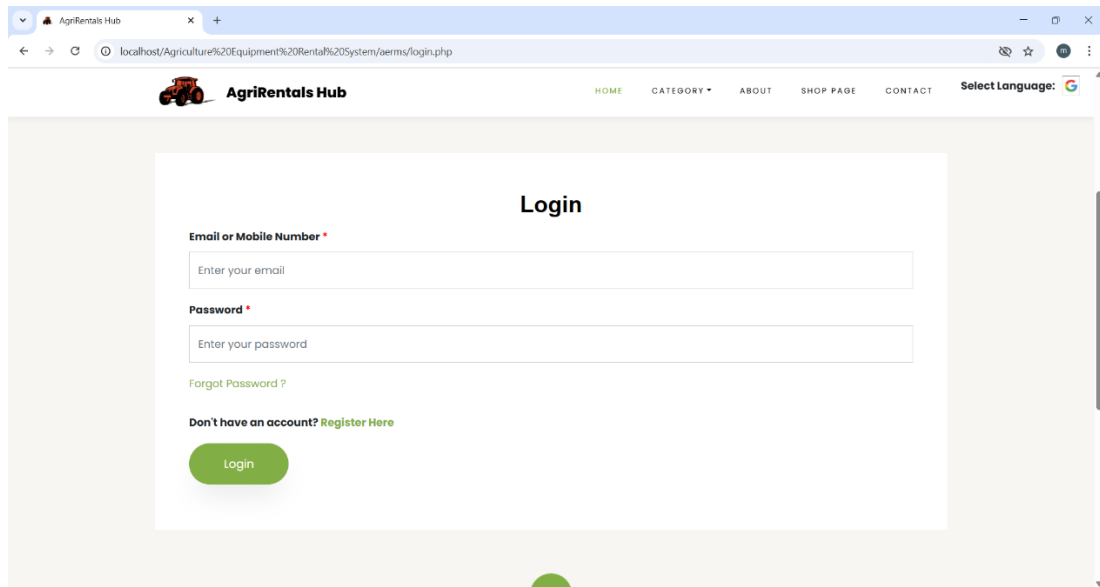


- **SIGN-UP PAGE:** -

This is the sign-up page of the Agriculture Equipment Rental Management System. New users can create an account to rent or list agricultural equipment easily.

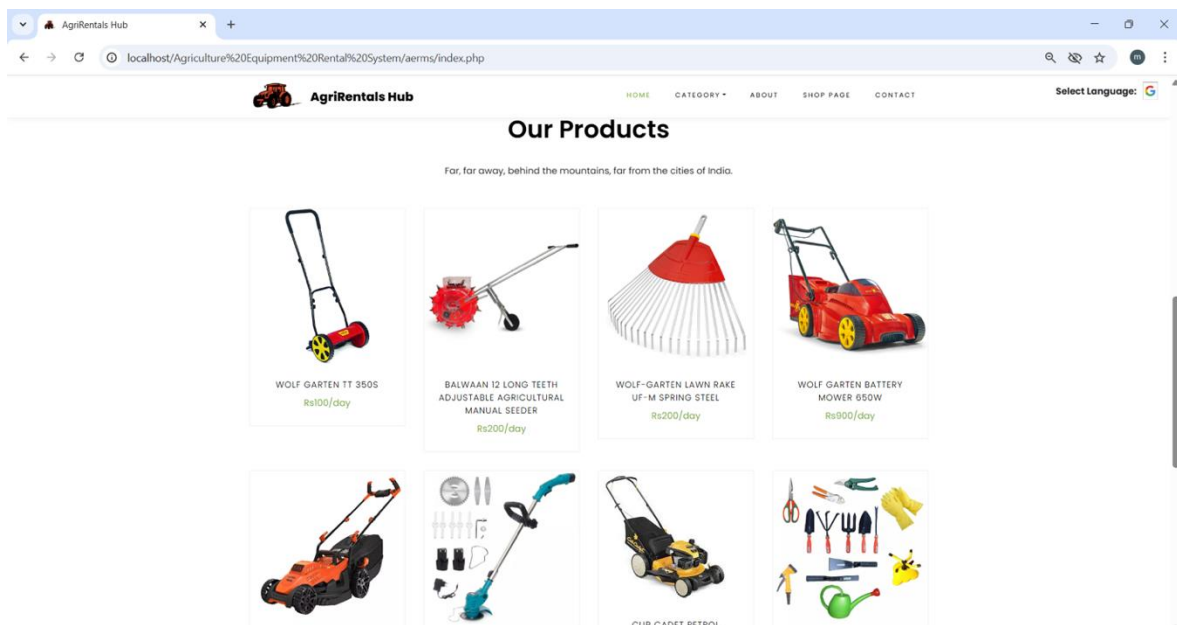
- **LOGIN PAGE:** -

This is the login page of the Agriculture Equipment Rental Management System. Users can securely sign in to access their accounts, manage equipment rentals, and track bookings.



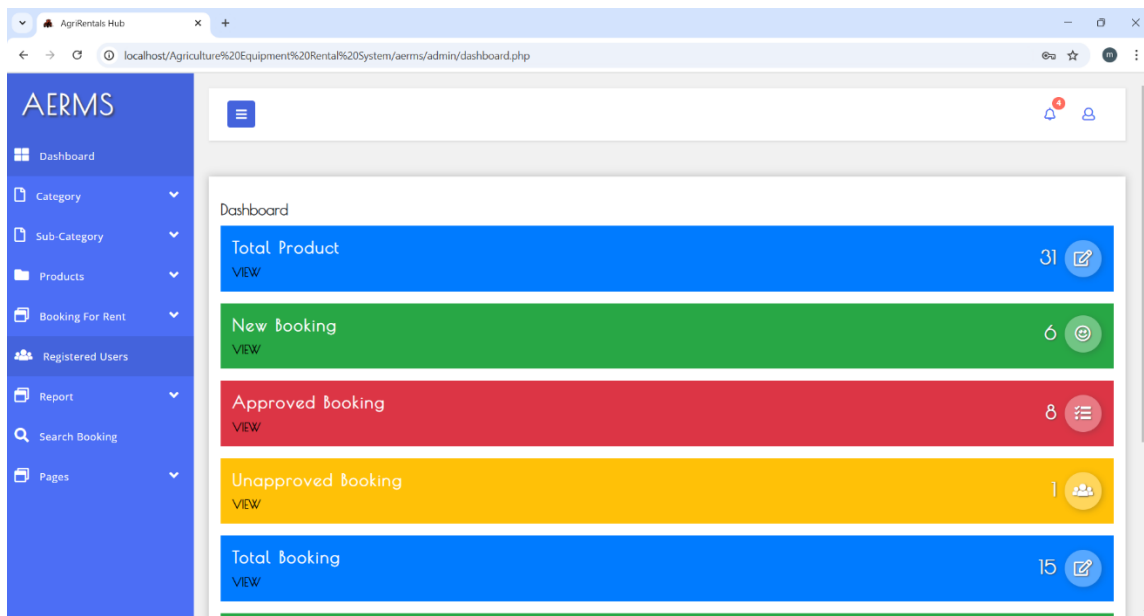
- **PRODUCTS:** -

This is the products page of the Agriculture Equipment Rental Management System. It showcases a variety of agricultural equipment available for rent, including tractors, harvesters, sprayers, hand tools and many more.



- **ADMIN – DASH BOARD: -**

This is the admin panel dashboard of the Agriculture Equipment Rental Management System. It allows administrators to manage users, equipment listings, overview of bookings, registered users and rental status in one place.

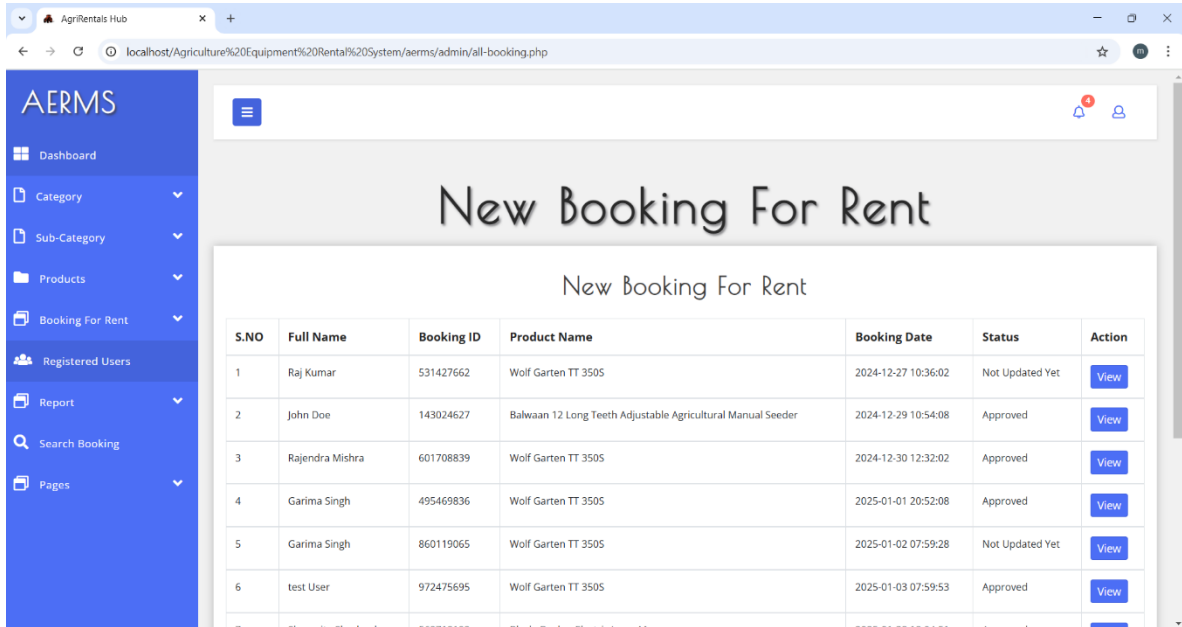


- **REGISTERED USERS: -**

The screenshot displays the 'View Register Users' page in the AERMS Admin Dashboard. The page has a white header with a hamburger menu and notification bell. Below the header, the title 'View Register Users' is centered. Underneath, there is a section titled 'Users Details' containing a table with the following data:

S.NO	Full Name	MobileNumber	Email Address	Reg Date	Action
1	Rishi Bhardwaj	9845632786	rishi@gmail.com	2024-12-29 12:43:47	Bookings
2	Meenakshi Singh	9878775464	singh@gmail.com	2024-12-29 12:43:47	Bookings
3	Devesh Gupta	5656565656	devesh@gmail.com	2024-12-29 12:43:47	Bookings
4	Abir Rahwans	8745665723	abir@gmail.com	2024-12-29 12:43:47	Bookings
5	Perth Kaushal	6345278453	perth@gmail.com	2024-12-29 12:43:47	Bookings
6	Shagun Mishra	7897978798	shagun@gmail.com	2024-12-30 12:43:47	Bookings
7	Andherson	6070100470	andh@gmail.com	2024-12-30 12:43:47	Bookings

• BOOKING OVERVIEW: -



S.NO	Full Name	Booking ID	Product Name	Booking Date	Status	Action
1	Raj Kumar	531427662	Wolf Garten TT 350S	2024-12-27 10:36:02	Not Updated Yet	View
2	John Doe	143024627	Balwaan 12 Long Teeth Adjustable Agricultural Manual Seeder	2024-12-29 10:54:08	Approved	View
3	Rajendra Mishra	601708839	Wolf Garten TT 350S	2024-12-30 12:32:02	Approved	View
4	Garima Singh	495469836	Wolf Garten TT 350S	2025-01-01 20:52:08	Approved	View
5	Garima Singh	860119065	Wolf Garten TT 350S	2025-01-02 07:59:28	Not Updated Yet	View
6	test User	972475695	Wolf Garten TT 350S	2025-01-03 07:59:53	Approved	View
7	Chandan Chakraborty	660718103	Black Power Electric Farm Mower	2025-01-10 18:04:45	Approved	View

• RENTAL STATUS – ADMIN: -

AgriRentals Hub

localhost/Agriculture%20Equipment%20Rental%20System/aerms/admin/view-booking.php?viewid=4

Power Source	Manual	RentalPrice(per day)	200	Product Model	5-12
Product Description	Agriculture Manual Hand Operated Seeding Machine is a premium quality product. All Manual Hand Operated Seeding Machines, are manufactured by using quality-assured material and advanced techniques, which make them up to the standard in this highly challenging field. The materials utilized to manufacture this are sourced from the most reliable and official vendors, chosen after performing detailed market surveys, products are widely acknowledged in the market for their high quality				
Product Specifications	Model: 5-12 Color: Red Number of Teeth: 12 Suitable For: Field and gardening crops				
Booking Details					
Booking Date	2024-12-29 10:54:08				
From Date	2022-07-30	To Date	2022-08-05	Order Quantity	1
Total Days of Rent	6	Rental Price	200	Total Cost	1200
Booking Number	143024627	Order Final Status	Your Booking has been approved		
Response History					
#	TotalCost	Remark	Status	Response Time	
1	3000	Booking has been confirmed	Approved	2025-02-21 17:42:00	

Agriculture Equipment Rental Management System

- **RENTAL STATUS -USER:-**

#805325468 Booking Details
Booking #805325468
Booking Date: 2025-02-21 20:20:06
Booking Status: Not Responded Yet

Invoice

Booking

Booking Number	Booking Date	From Date	To Date	Product Image	Product Name	Quantity	Rental Price	Total Price (Excl. GST)	GST (18%)	Total Price (Incl. GST)
					Neptune 6.5 HP	Simplify				

- **CONTACT US:-**

This is the Contact Us page of the Agriculture Equipment Rental Management System. Users can reach out for support, inquiries, or assistance regarding rentals and services.

HOME CONTACT US
CONTACT US

Flat No: 207, Alluri Residential complex, bhagyanagar colony, KPHB, Hyderabad - 500072

+91 9381472305

info@agrireals.com

9. SAMPLE CODE

```

<?php
session_start();
error_reporting(0);
include('includes/dbconnection.php');
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <title>AgriRentals Hub</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="icon" href="images/logo.png" type="image/x-icon" sizes="32x32">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css" rel="stylesheet">
    <link
href="https://fonts.googleapis.com/css?family=Poppins:200,300,400,500,600,700,800&display=swap"
rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Lora:400,400i,700,700i&display=swap"
rel="stylesheet">
    <link href="https://fonts.googleapis.com/css?family=Amatic+SC:400,700&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="css/open-iconic-bootstrap.min.css">
    <link rel="stylesheet" href="css/animate.css">
    <link rel="stylesheet" href="css/owl.carousel.min.css">
    <link rel="stylesheet" href="css/owl.theme.default.min.css">
    <link rel="stylesheet" href="css/magnific-popup.css">
    <link rel="stylesheet" href="css/aos.css">
    <link rel="stylesheet" href="css/ionicons.min.css">
    <link rel="stylesheet" href="css/bootstrap-datepicker.css">
    <link rel="stylesheet" href="css/jquery.timepicker.css">
    <link rel="stylesheet" href="css/flaticon.css">
    <link rel="stylesheet" href="css/icomoon.css">
    <link rel="stylesheet" href="css/style.css">
</head>

<body class="goto-here">
    <?php include_once('includes/header.php'); ?>

    <section id="home-section" class="hero">
        <div class="home-slider owl-carousel">
            <div class="slider-item" style="background-image: url(images/bg_1.jpg);">
                <div class="overlay"></div>
                <div class="container">
                    <div class="row slider-text justify-content-center align-items-center">
                        <div class="col-md-12 ftco-animate text-center">
                            <h1 class="mb-2">Explore Our Range of Equipment</h1>
                        </div>
                    </div>
                </div>
            </div>
            <div class="slider-item" style="background-image: url(images/bg_2.jpg);">

```

```

<div class="overlay"></div>
<div class="container">
  <div class="row slider-text justify-content-center align-items-center">
    <div class="col-sm-12 ftco-animate text-center">
      <h1 class="mb-2">Your Trusted Partner for Farming Tools & Equipment</h1>
    </div>
  </div>
</div>
</div>
</div>
</section>

<section class="ftco-section">
  <div class="container">
    <div class="row no-gutters ftco-services">
      <div class="col-md-3 text-center d-flex align-self-stretch ftco-animate">
        <div class="media block-6 services mb-md-0 mb-4">
          <div class="icon bg-color-1 active d-flex justify-content-center align-items-center mb-2">
            <span class="flaticon-shipped"></span>
          </div>
          <div class="media-body">
            <h3 class="heading">Free Shipping</h3>
            <span>On order over ₹ 2000</span>
          </div>
        </div>
      </div>
      <div class="col-md-3 text-center d-flex align-self-stretch ftco-animate">
        <div class="media block-6 services mb-md-0 mb-4">
          <div class="icon bg-color-2 d-flex justify-content-center align-items-center mb-2">
            <span class="flaticon-box"></span>
          </div>
          <div class="media-body">
            <h3 class="heading">100% Safe</h3>
            <span>Product well package</span>
          </div>
        </div>
      </div>
      <div class="col-md-3 text-center d-flex align-self-stretch ftco-animate">
        <div class="media block-6 services mb-md-0 mb-4">
          <div class="icon bg-color-3 d-flex justify-content-center align-items-center mb-2">
            <span class="flaticon-award"></span>
          </div>
          <div class="media-body">
            <h3 class="heading">Superior Quality</h3>
            <span>Quality Products</span>
          </div>
        </div>
      </div>
      <div class="col-md-3 text-center d-flex align-self-stretch ftco-animate">
        <div class="media block-6 services mb-md-0 mb-4">
          <div class="icon bg-color-4 d-flex justify-content-center align-items-center mb-2">
            <span class="flaticon-customer-service"></span>
          </div>
          <div class="media-body">

```

```

        <h3 class="heading">Support</h3>
        <span>24/7 Support</span>
    </div>
</div>
</div>
</div>
</div>
</section>

<?php include_once('includes/footer.php'); ?>

<script src="js/jquery.min.js"></script>
<script src="js/jquery-migrate-3.0.1.min.js"></script>
<script src="js/popper.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/jquery.easing.1.3.js"></script>
<script src="js/jquery.waypoints.min.js"></script>
<script src="js/jquery.stellar.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
<script src="js/jquery.magnific-popup.min.js"></script>
<script src="js/aos.js"></script>
<script src="js/jquery.animateNumber.min.js"></script>
<script src="js/bootstrap-datepicker.js"></script>
<script src="js/scrollax.min.js"></script>
<script src="js/google-map.js"></script>
<script src="js/main.js"></script>

</body>

</html>

```

10. TESTING

A process of executing a program with the explicit intention of finding errors, that is making the program fail.

TESTING METHODOLOGIES:

- Black Box Testing
- Software Testing
- White Box Testing

SOFTWARE TESTING:

It is the process of testing the functionality and correctness of software by running it. A good test case is one that has a high probability of finding an as yet undiscovered error. Software Testing is usually performed for one of two reasons:

- Defect Detection
- Reliability Estimation

BLACK BOX TESTING:

Applies to software systems or modules, tests functionality in terms of inputs and outputs at interfaces. Test reveals if the software function is fully operational with reference to requirements specification.

WHITE/GLASS BOX TESTING:

Knowing the internal working i.e., to test if all internal operations are performed according to the program structure and data structure. To test if all internal components have been adequately exercised.

SOFTWARE TESTING STRATEGIES:

A Strategy for the software testing will begin in the following order:

- Unit Testing

- Integration Testing
- Validation Testing
- System Testing

UNIT TESTING:

It concentrates on each unit of the software as implemented in source code and is a white box oriented. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module.

Unit Testing Benefits:

- Unit testing increases confidence in changing/ maintaining code. If good unit tests are written and if they are run every time any code is changed, we will be able to promptly catch any defects introduced due to the change. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.
- Codes are more reusable. In order to make unit testing possible, codes need to be modular. This means that codes are easier to reuse.

INTEGRATION TESTING:

Here focus is on design and construction of the software architecture. The objective is to take unit tested components and build a program structure that has been dictated by design.

Analogy:

During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced separately and unit tested separately.

When two or more units are ready, they are assembled and Integration Testing is performed. For example, whether the cap fits into the body or not

VALIDATION TESTING:

Requirements established as part of software requirements analysis are validated against the software that has been constructed i.e., validation succeeds when software functions in a manner that can be reasonably expected by the customer. Objective is to ensure that the product actually meets the user's needs and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.

SYSTEM TESTING:

In this software and other system elements are tested as a whole.

Analogy:

During the process of manufacturing a ballpoint pen, the cap, the body, the tail, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. When the complete pen is integrated, System Testing is performed.

11. IMPLEMENTATION

Implementation is the carrying out, execution or practice of a plan, to implement the project into our system we will need to perform the following operation:

- It works for both android and PCs (Laptop).
- Go to the updated browser.
- Search for the Intrasy website.
- Number of website links are available.
- Search for our only trustable website.
- Open the website and login with the credentials for further usage.
- Visit different pages according to their requirements.

12. MAINTENANCE

It is possible to develop a system that makes all the requirements of the user, User requirements keep changing as the system is being used.

Software will definitely go through change once when it is delivered to the customer.

There are a large number of reasons for the change. Change could happen due to some unpredicted input values into the system. In addition to this the change in the system directly has an effect on the software operations. The software should be implemented to accommodate changes that could be happen during the post development period Some of the future enhancements that can be done to this system are:

As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.

Based on the future security issues, Security can be improved using emerging technologies.

13. CONCLUSION

In conclusion, The Agriculture Equipment Rental Management System (AERMS) represents a significant advancement in agricultural resource management, transforming how farmers can access essential equipment. By digitizing the rental process, optimizing equipment utilization, and ensuring a seamless connection between farmers and equipment owner (Admin), the system enhances efficiency and accessibility. Additionally, its structured approach to equipment listing, booking, and availability management streamlines operations, reducing manual efforts and improving overall productivity.

In summary, The Agriculture Equipment Rental Management System (AERMS) is a digital platform designed to bridge the gap between farmers and equipment owner (Admin), making modern agricultural machinery more accessible and affordable. It enables farmers, particularly small and marginal ones, to rent equipment as needed, reducing the financial burden of ownership while improving productivity and efficiency. The system provides a centralized platform for equipment listing, and seamless booking, ensuring transparency and ease of use. By automating the rental process, AERMS optimizes equipment utilization, minimizes idle resources, and enhances overall agricultural operations. With features such as structured equipment management, and user-friendly interfaces, AERMS simplifies the rental experience for both farmers and equipment owner (Admin). As a cost-effective and scalable solution, it contributes to the modernization of agriculture, promoting sustainable farming practices and economic growth.

14. **BIBLIOGRAPHY**

- <https://openai.com/index/chatgpt/>
- <https://www.youtube.com/@OnlineTutorialsYT>
- <https://www.youtube.com/@MrWebDesignerAnas>
- <https://www.mysql.com/about/>
- https://www.w3schools.com/php/php_mysql_intro.asp
- https://www.w3schools.com/bootstrap/bootstrap_ver.asp