PROJECT REPORT

ON

Morphowiz

FOR

Maharashtra Education Society's
Institute of Management And Career
Courses

BY

Supriya Morankar

SAVITRIBAI PHULE PUNE UNIVERSITY

MASTER IN COMPUTER APPLICATION

MAHARASHTRA EDUCATION SOCIETY's

INSTITUTE OF MANAGEMENT AND CAREER COURSES

(IMCC), PUNE-411038

2023-24

MAHARASHTRA EDUCATION SOCIETY'S
(SINCE 1860)

INSTITUTE OF MANAGEMENT & CAREER COURSES (IMCC)

Approved by AICTE and Recognized by Savitribai Phule Pune University, Pune
131, Mayur Colony, Kothrud, Pune 411038, Maharashtra, India | Ph.: 020-25466271/73 | e-mail: info.imcc@mespune.in | https://imcc.mespune.in/

NAAC Accredited with Grade A+

Ref. No. MES IMCC / 399/ 2023 – 24/ **Date:13/04/2024**

CERTIFICATE

This is to certify that the Project Report entitled

"Morphowiz"

is prepared by

Supriya Morankar

M.C.A. Semester IV Course for the Academic Year 2023-24 at M.E. Society's Institute of Management & Career Courses (IMCC), Pune - 411038.
M.C.A Course is affiliated to Savitribai Phule Pune University.

To the best of our knowledge, this is original study done by the said student and important sources used by herhave been duly acknowledged in this report.

The report is submitted in partial fulfillment of M.C.A Course for the Academic Year 2023-224 as per the rules andprescribed guidelines of Savitribai Phule Pune University.

Dr. Ravikant Zirmite
Head, Dept of
MCAMES IMCC

Dr. Santosh Deshpande
Director,
MES
IMCC

Internal Examiner

External Examiner

Pictel Ai

PICTELAI PVT LTD

Internship Certificate

12th April 2024

To Whom It May Concern

This letter is to certify that Ms. Supriya Morankar has been offered a position of Intern under the guidance of Ms. Anu Shivanand Raje (Team Lead). Her internship program is of 12 months and tenure is from Sept 01, 2023 to August 31, 2024.

She is actively & diligently involved in the projects and tasks assigned to her for the project name '**Morphowiz**'.

As per company policy, any kind of source code or executables cannot be shared with the Intern as its solely belongs to PictelAi Pvt. Ltd.

Sincerely,

Anu Shivanand Raje
Team Lead
PictelAi Pvt. Ltd.

PICTELAI PRIVATE LIMITED

S112/1, H-1085, Opp Pragati Steel, Baner Road, Baner, Pune 411045
U72900PN2022PTC214230, Email id- sagar@pictelai.com, Ph No – 88888 11661.

CERTIFICATE

This is to certify that the student Supriya Morankar has completed the project work entitled "*Morphowiz*" under my guidance. The report is submitted in partial fulfillment of M.C.A. Course for the Academic Year 2023-2024 as per the rules & prescribed guidelines of Savitribai Phule Pune University.

her work is found to be satisfactory and complete in all respects.

Internal Guide Name
(Ms. Darshana Yadav)

ACKNOWLEDGEMENT

It is indeed with a great pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to our Director Dr. Sontosh Deshpande sir Deputy Director Dr.Mansi Bhate Ma'am for the facilities provided to accomplish this main project.

We would like to thank our Head of the Department of MCA Dr Ravikant Zirmite Sir for this constructive criticism throughout our project.

We feel elated to express our sense of gratitude to our internal project guide Abhay Damle Sir he has been a constant source of inspiration for us and we are very deeply thankful to him for his support and valuable advice.

We are extremely grateful to our Departmental staff members, Lab technicians and non-teaching staff members for their extreme help throughout our project.

Finally, we express our heart full thanks to all our friends who helped us in the successful completion of this project.

INDEX

1.INTRODUCTION

1.1 Company Profile

Pictel Al is a blend of two words, representing our core values and vision:

"Pictel" is derived from "pixel," the basic unit of any image. This word underscores our dedication to revolutionizing image processing and analysis. We strive to develop cutting-edge technologies that empower businesses and individuals to extract valuable insights from images, optimizing their workflows and unlocking new possibilities.

"Al" stands for "Artificial Intelligence," reflecting our commitment to harnessing the power of Al and machine learning to augment image processing capabilities. We aim to create state-of-the-art Al-driven solutions that enable our clients to stay ahead of the curve in their respective industries.

Together, Pictel Al symbolizes our mission to deliver advanced image processing and Al solutions that transform the way our clients interact with images and data. We are committed to helping them turn their visual information into actionable insights for smarter, data-driven decisions.

1.2 Abstract

1. Particle size measurements, ranging from manual to automatic, are crucial in various scientific and industrial applications. Manual measurement with an optical microscope involves comparing calibrated dimensions in the eyepiece with the object or particle of interest, typically applicable for particles of 1 µm and larger.

2. Microscopic investigations, especially when coupled with computer-based image analysis systems, offer a practical means of determining particle size and shape. While other particle size analysis instruments may be faster and more precise, the microscope remains indispensable for determining particle shape.

3. Microscope Image Analysis (MIA) systems consist of a microscope, digital camera, and analysis software, enabling the capture of images of drug samples for particle analysis. The software utilizes unique algorithms to detect particles, calculate parameters, and provide valuable statistics, offering rapid and accurate results.

4. Microscopic techniques have gradually replaced conventional methods like laser diffraction, especially for irregularly shaped particles. Optical microscopy excels in characterizing non-spherical particles, such as elongated particles (e.g., needles or rods), where laser diffraction may yield inaccurate results.

1.3 Existing System

In the Pharmaceutical industry particle characterization of drug substances and the drug products is an important factor in R&D, production, and quality control of pharmaceuticals. It is becoming increasingly important for compliance with requirements of FDA and European Health Authorities. Particle Size (PS), Particle size Distribution (PSD) and Particle shape (Morphology) of pharmaceutical ingredients is known to strongly affect the stability and aesthetics of the sample. Consequently, most industries have some interest in determining size and shape.

Particle size of pharmaceutical products have an influence on chemical and physical behavior. Particle size is therefore relevant for the behavior of powders, granulates, creams, emulsions, liquids, etc. in relation to Bioavailability, Flow ability, Adhesive strength, Drying properties, Solubility, Filterability, Thermal conductivity etc.

Laser Diffraction for Particle Size Analysis

Laser diffraction measurements capture information about particle size distribution by measuring scattering intensity as a function of the scattering angle, wavelength and polarization of light based on applicable scattering models. Laser diffraction offers a number of advantages, including ease-of-use, fast operation, high reproducibility. Laser Diffraction (LD): Laser diffraction measures particle size distributions by measuring the

angular variation in intensity of light scattered as a laser beam passes through a dispersed particulate sample. Large particles scatter light at small angles relative to the laser beam and small particles scatter light at large angles. The angular scattering intensity data is then analysed to calculate the size of the particles responsible for creating the scattering pattern, using the Mie theory of light scattering. The particle size is reported as a volume equivalent sphere diameter.

Typical schema of a laser diffraction instrument

The process begins with a light source that generates a monochromatic beam. After passing through several optical components, the raw beam creates an expanded, collimated beam that illuminate's particles in the scattering volume. The particles scatter light, generating unique angular scattering patterns. These scattering patterns are transformed into a spatial intensity pattern that is detected by a multi-element photo-detector array.

Need for System

Particle size measurements on the microscope range from completely manual to completely automatic. Manual measurement with the optical microscope simply requires one to compare a line or shape in the eyepiece (calibrated for each objective) with dimensions of the object or particle of interest. Microscopic investigations can generally be applied to particles of 1 μm and larger.

The microscope, particularly coupled with computer-based image analysis systems, is well suited for determination of particle size even though other types of particle size analysis instruments may be faster and more precise. The microscope is really the only practical means of determining particle shape. Image analysis systems today are so powerful, that it is now a straight-forward and relatively short task to develop robust and dependable particle size and shape methods for all sorts of materials.

Microscope Image Analysis (MIA): The system comprises of a microscope, a digital camera and analysis software. Digital camera takes images of drug sample kept under the microscope and transfers to connected computer. The software having unique algorithms detects particles and calculates various parameters. It is a direct technique which measures 2D images of a 3D particles. Image analysis give a number weighted distribution where each particle is given equal weighting irrespective of its size. The software having unique algorithm separates agglomerations and

isolated particles. Morphology of every particle can be seen and valuable statistics are generated. The automated system provides results in few seconds.

Why to choose microscopic technique?

Over the past few years, microscopic technique has replaced conventional methods, such as laser diffraction technique because for irregular and irregular spherical particles results were found closely similar by both the technique but for elongated particles (needle or rod shape) particle size measurement by laser diffraction may not be accurate. Optical microscopy is particularly useful for characterization of particles that are not spherical. Microscope Image Analysis is an orthogonal technique for validation of particle size measurement obtained by Laser diffraction technique.

Image Analysis:

Generate the proper image on the microscope first and then ensure that this image is replicated on the image analysis computer screen. Image analysis systems detect and measure features based on contrast between the feature and background. For example, particles and immersion liquid on the optical microscope. It is common practice to adjust the microscope based on the image generated on the computer screen. If we generate a good contrast image using proper illumination techniques on the microscope, then it is generally a simple matter to get a good image on the computer screen. The optimum number of separated features

having good contrast with the background is necessary for an accurate, precise, robust image analysis method.

Measurement Parameters

In most cases, some are interested in the distribution of feature (particle) size as well as the distribution of feature (particle) shape. The equivalent circular diameter, aspect ratio, and circularity (how close the particle fits a circle) are all common measurement parameters.

The number of direct and derived measurement parameters is effectively infinite - more than one can reasonably count.

1) Direct parameters include area, count, perimeter, Fret diameters, length, width etc.

2) Derived parameters include circular diameter, aspect ratio, circularity, roughness, etc.

1.4 Scope of the System

1. **Range of Particle Size Measurements:** The system will cater to particle size measurements ranging from completely manual to completely automatic. It will support manual measurements with optical microscopes for particles of 1 µm and larger, as well as fully automated measurements using computer-based image analysis systems.
2. **Microscope-Camera-Software Integration**: The system will integrate a microscope, digital camera, and analysis software to facilitate particle analysis. It will enable the parameters ages of samples under the microscope, transfer them to a connected computer, and utilize unique algorithms to detect particles, calculate various parameters, and generate valuable statistics.
3. **Direct Measurement of 2D Images**: Utilizing Microscope Image Analysis (MIA), the system will directly measure 2D images of 3D particles. It will provide a number-weighted distribution where each particle is given equal weighting irrespective of its size, and will accurately separate agglomerations and isolated particles.
4. **Rapid and Automated Results**: The system will provide results in a few seconds, offering rapid and

automated particle size and shape analysis. It will streamline the process of developing robust and dependable methods for various materials.

5. **Orthogonal Technique for Validation**: Microscope Image Analysis will serve as an orthogonal technique for validation of particle size measurements obtained by conventional methods such as laser diffraction. It will be particularly useful for characterizing non-spherical particles, ensuring accurate and reliable measurements.

6. **Measurement Parameters**: The system will support a wide range of measurement parameters, including both direct and derived parameters. Direct parameters may include area, count, perimeter, and dimensions like length and width, while derived parameters may include circular diameter, aspect ratio, circularity, and roughness, among others.

Overall, the system will provide comprehensive support for particle size and shape analysis, leveraging the capabilities of microscopy, digital imaging, and advanced analysis algorithms to deliver accurate and rapid results for various materials.

1.5 Operating Environment

Hardware Environment:

Hard Disk -512GB

Camera, Microscope, Connecting Cables

Software Environment:

Operating System: Windows

Development tool: Visual Studio Code.

1.6 Brief Description of Technology Used

Backend - GoLang

Go, also known as Golang, is an open-source programming language developed by Google in 2007 and released in 2009. It was created by Robert Griesemer, Rob Pike, and Ken Thompson, all of whom were involved in the development of Unix and other programming languages like C and Plan 9.

Here's a brief description of Go:

1. **Concurrency**: Go is designed with built-in support for concurrency. It utilizes goroutines, which are lightweight threads managed by the Go runtime. Goroutines make it easy to write concurrent programs without the complexity of traditional threading models.
2. **Efficiency**: Go compiles to machine code, resulting in high-performance executables. It has a garbage collector for memory management and provides efficient support for multi-core processing.
3. **Simplicity**: Go emphasizes simplicity and readability. It has a minimalistic syntax with a small number of keywords and features. This simplicity makes it easy to learn and maintain large codebases.

Overall, Go is a modern programming language that prioritizes simplicity, efficiency, and concurrency. It is well-suited for

building a wide range of applications, from web servers and command-line tools to distributed systems and cloud-native applications.

Front-end Anguler

Angular is a popular open-source framework maintained by Google for building web applications. It is written in TypeScript and offers a structured approach to web development, emphasizing modularity, reusability, and maintainability.

Here's a brief description of Angular:

1. Modular Architecture: Angular follows a modular architecture, allowing developers to break down their applications into reusable and maintainable components. Components encapsulate HTML templates, CSS styles, and TypeScript logic, making it easy to manage complex UI elements.
2. Two-way Data Binding: Angular provides two-way data binding, enabling automatic synchronization between the model (JavaScript objects) and the view (HTML templates). This simplifies data manipulation and reduces boilerplate code.

1.6.1 Operating systems used Windows

1.6.2 No Sql used to build database (MongoDB)

2.PROPOSED SYSTEM

2.1 Study of Similar Systems

Particle Detector Applications in Medicine

Selected particle detectors are described which find an application in medicine and have been the topic of presentations at the 2013 Vienna Conference of Instrumentation (VCI).

Particle detector applications in medicine are characterized by low energies and relatively small systems (at least until commercialization). They are still profiting from developments in High-Energy Physics and (due to the low energy), even more so in Astrophysics, although as shown in this paper, medicine-specific projects are becoming common.

This paper will concentrate on semiconductor systems; gaseous detectors in the WCC heritage have been covered e.g. in talks by V. Peskov and D. Nygren; the in addition, many recent developments of scintillator materials have been reported at VCI 2013. The first section will deal with photon detectors in dosimetry and imaging, including X-ray computed tomography (CT), single-positron emission CT (SPECT), positron emission tomography (PET) and time-of-flight PET (ToF-PET). This will be followed by a discussion of detectors in hadron therapy, a topical subject with the advent of the MedAustron, and their application in interaction vertex imaging (IVI), and proton CT (pCT).

2.2 Feasibility Study

1. **Technical Feasibility:**
    - Equipment Availability: Assess the availability and suitability of the necessary equipment, including microscopes, digital cameras, and image analysis software.
    - Technology Compatibility: Evaluate the compatibility of the microscopy and image analysis technologies with the intended application and sample types.
    - Algorithm Development: Determine the feasibility of developing or acquiring algorithms capable of accurately detecting and analysing particles in 2D images.
    - System Integration: Investigate the feasibility of integrating microscopy, digital imaging, and analysis software into a cohesive system.
2. **Operational Feasibility:**
    - User Training: Assess the feasibility of training users to operate the microscopy system, capture images, and perform image analysis tasks effectively.

- Workflow Integration: Evaluate the feasibility of integrating the proposed system into existing laboratory workflows and processes.
- Resource Requirements: Estimate the resources required, including personnel, time, and materials, to operate and maintain the system.

3. **Economic Feasibility:**

- Cost-Benefit Analysis: Conduct a cost-benefit analysis to determine the financial feasibility of implementing the system. Consider the costs of equipment, software, training, and maintenance against the expected benefits, such as improved accuracy and efficiency in particle analysis.
- Return on Investment (ROI): Calculate the potential ROI based on projected cost savings, increased productivity, and other quantifiable benefits over time.
- Budgetary Constraints: Evaluate whether the project falls within the organization's budgetary constraints and financial resources.

2.3 Objectives of Proposed System

The objective of the proposed system for microscopy-based particle size and shape analysis is to provide a comprehensive and efficient solution for characterizing particles in various materials. The primary goals and objectives of the system include:

Accurate Particle Analysis: Develop algorithms and techniques for accurately detecting, measuring, and analyzing the size and shape of particles in 2D images captured by the microscope.

Automated Analysis: Implement automation features to streamline the particle analysis process, reducing manual intervention and improving efficiency.

Robust and Reliable Results: Ensure that the system generates robust and reliable results by employing advanced image analysis algorithms and validation techniques.

Fast Turnaround Time: Provide rapid results generation, with minimal processing time, to support timely decision-making in research, development, and quality control processes.

User-Friendly Interface: Design an intuitive and user-friendly interface for the microscopy system and analysis software,

enabling researchers and technicians to easily operate the system and interpret results.

Comprehensive Parameter Measurement: Support the measurement of various particle parameters, including size, shape, aspect ratio, circularity, and roughness, to provide a comprehensive characterization of particle populations.

Integration with Existing Workflows: Ensure seamless integration of the system with existing laboratory workflows and data management systems, minimizing disruptions and facilitating adoption.

Validation and Calibration: Implement validation procedures to verify the accuracy and reliability of the system's measurements, including comparison with reference methods and validation against known standards.

Scalability and Flexibility: Design the system to be scalable and adaptable to different sample types, materials, and research requirements, allowing for versatility and future expansion.

2.4 User of system

1. Super Admin:
    - Role: The Super Admin has the highest level of access and authority within the system. They oversee and manage all system privileges and functionalities.
    - Responsibilities:
        - Manage system privileges and access levels for different user roles.
        - Oversee the overall functioning of the system, including user management, organization management, subscription management, and addon management.
        - Manage all organizations, units, plans, and addons according to subscription levels.
        - Ensure smooth operation and performance of the system.
        - Handle any technical issues or escalations that may arise.
2. Admin:
    - Role: The Admin role has administrative privileges within the system, but their scope is typically limited to managing modules and users.

- Responsibilities:
    - Manage modules within the system, including configuration, customization, and updates.
    - Manage user accounts, permissions, and access rights.
    - Assist users with system-related queries and issues.
    - Collaborate with the Super Admin in overseeing system operations and ensuring compliance with organizational policies and procedures.

3. User:
    - Role: The User role has basic access privileges within the system and is primarily focused on accessing and interacting with the system's products or services.
    - Responsibilities:
        - Access and utilize the products or services offered by the system.
        - View information, data, or content relevant to their role or responsibilities.
        - Interact with the system's features, functionalities, and modules as permitted by their access level.
        - Collaborate with other users and contribute to organizational goals and objective.

3.ANALYSIS AND DESIGN

3.1 System Requirements

Functional Requirement

Functional requirements for a system managing user roles, organizations, units, plans, addons, and modules typically include:

1. User Management:
    - Ability to create, update, and delete user accounts.
    - User authentication and authorization mechanisms.
    - Password management features, such as password reset and account recovery.
2. Organization Management:
    - Capability to create, edit, and delete organizations.
    - Assignment of users to organizations.
3. Unit Management:
    - Functionality to create, modify, and remove organizational units or departments.
    - Assignment of users to specific units or departments.
4. Plan Management:
    - Ability to define subscription plans with various features, limits, and pricing tiers.
5. Addon Management:

- Capability to offer additional features or services as addons to subscription plans.

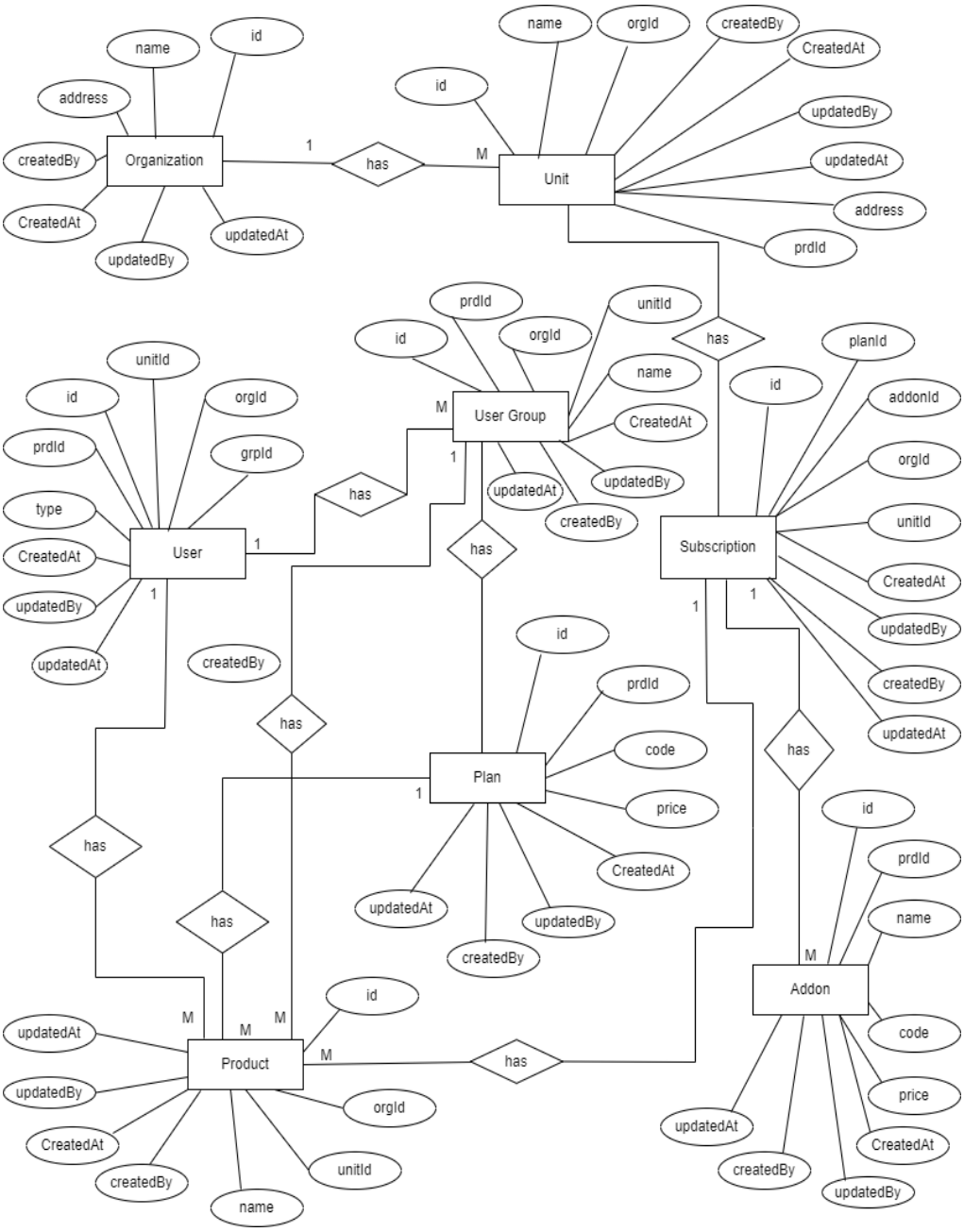
6. Module Management:
    - Functionality to configure and customize system modules based on organizational requirements.
7. System Privileges:
    - Definition of system privileges and permissions for each user role (e.g., Super Admin, Admin, User).
8. Subscription Management:
    - Handling of subscription lifecycle, including creation, renewal, upgrade, downgrade, and cancellation.
9. Audit Trail and Logging:
    - Logging of all user actions, system events, and administrative activities.
10. Integration and APIs:
    - Support for integration with external systems, services, or third-party applications.

These functional requirements ensure that the system effectively manages user roles, organizations, subscriptions, and modules while providing flexibility, scalability, and security for the users and administrators.

Non-functional requirement

1. **Security**:
    - Ensure data security and privacy by implementing encryption, secure authentication mechanisms, and access controls.
    - Comply with industry standards and regulations (e.g., GDPR, HIPAA) for handling sensitive data.
2. **Performance**:
    - Provide responsive and high-performance system operations, with minimal latency and downtime.
    - Handle concurrent user interactions and large volumes of data efficiently.
3. **Reliability**:
    - Ensure system availability and reliability through redundancy, failover mechanisms, and disaster recovery strategies.
4. **Scalability**:
    - Design the system to scale horizontally and vertically to accommodate increasing user loads and data volumes.

3.2 ERD


Organization
name
id
address
createdBy
CreatedAt
updatedBy
updatedAt
has
1
M
Unit
name
orgId
createdBy
CreatedAt
updatedBy
updatedAt
address
prdId
id
User
unitId
id
orgId
prdId
grpId
type
CreatedAt
updatedBy
updatedAt
User Group
prdId
unitId
id
orgId
name
CreatedAt
updatedAt
updatedBy
createdBy
Subscription
id
planId
addonId
orgId
unitId
CreatedAt
updatedBy
createdBy
updatedAt
Plan
id
prdId
code
price
CreatedAt
updatedAt
updatedBy
createdBy
Addon
id
prdId
name
code
price
CreatedAt
updatedBy
createdBy
updatedAt
Product
id
updatedAt
updatedBy
CreatedAt
createdBy
name
orgId
unitId
createdBy
has

3.3 Table Structure

Organization

```
{
   "_id": "ObjectId",
   "name": "String",
   "addresses": [
      {
         "addressLine1": "String",
         "city": "String",
         "pin": "String",
         "state": "String",
         "country": "String"
      }
   ],
   "contacts": {
      "name": "String",
      "phoneNumber": "String",
      "email": "String"
   },
   "createdBy": "String",
   "updatedBy": "String",
   "createdAt": "Date"
}
```

Unit

```
{
   "_id": "ObjectId",
   "name": "String",
   "description": "String",
   "logo": "String",
   "addresses": [
      {
         "type": "Number",
         "addressLine1": "String",
         "addressLine2": "String",
         "city": "String",
         "pin": "String",
         "state": "String",
         "country": "String"
      }
   ],
   "contacts": "Array",
   "tags": "Array",
   "createdBy": "String",
   "updatedBy": "String",
   "createdAt": "Date"
}
```

Addon

```
{
   "_id": "ObjectId",
   "code": "String",
   "price": "String",
   "prdIds": {
      "Example": [
         "64ec8b89b8cf3a741b0a1d3e"
      ]
   },
   "Name": "String",
   "OrgId": "String",
   "UnitId": "String"
}
```

Subscription

```
{
   "_id": "ObjectId",
   "planId": "String",
   "orgId": "String",
   "unitId": "String",
   "addonIds": {
      "Example": [
         "64ec8b89b8cf3a741b0a1d3e"
      ]
   },
   "activatedAt": "Date",
   "expiredAt": "Date",
   "createdBy": "String",
   "updatedBy": "String",
   "createdAt": "Date",
   "updatedAt": "Date",
   "productId": "String"
}
```

Group

```
 {
    "Id": "ObjectId",
    "OrgId": "String",
    "UnitId": "String",
    "PrdId": "String",
    "CreatedBy": "String",
    "CreatedAt": "Date",
    "UpdatedBy": "String",
    "UpdatedAt": "Date"
 }
```

Plan

```
{
   "Code": "String",
   "AddonIds": {
      "addonID": "String"
   },
   "PrdId": "String",
   "Price": "Integer"
}
```

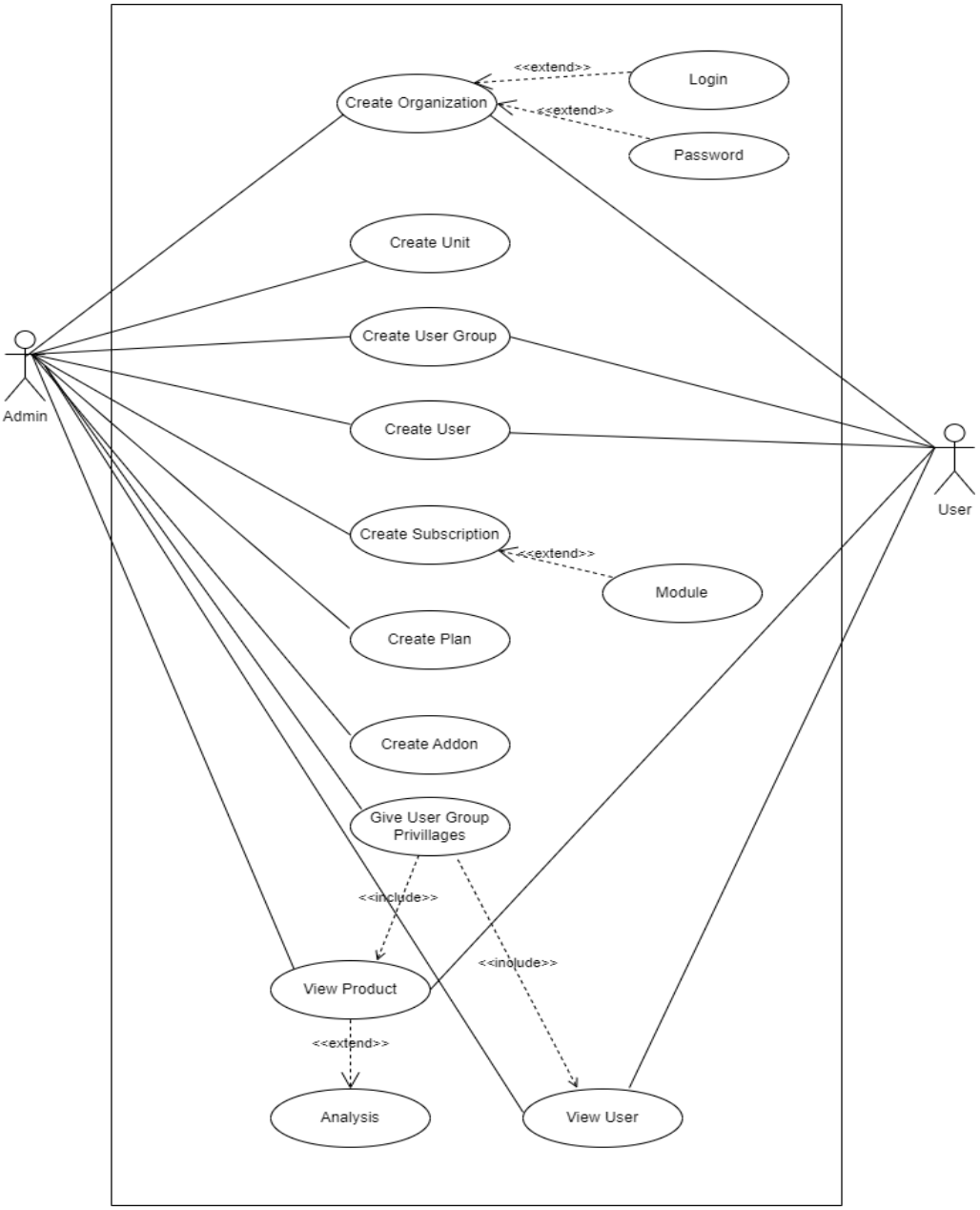
User

```
{
   "Id": "ObjectId",
   "Fname": "String",
   "Lname": "String",
   "Email": "String",
   "groupIds": "Array",
   "prdIds": "Array",
   "privillages": {
      "Data type": "Array",
      "Example": [
         0,
         1,
         2
      ],
      "Description": "0 for superadmin, 1 for admin, 2 for user"
   }
}
```

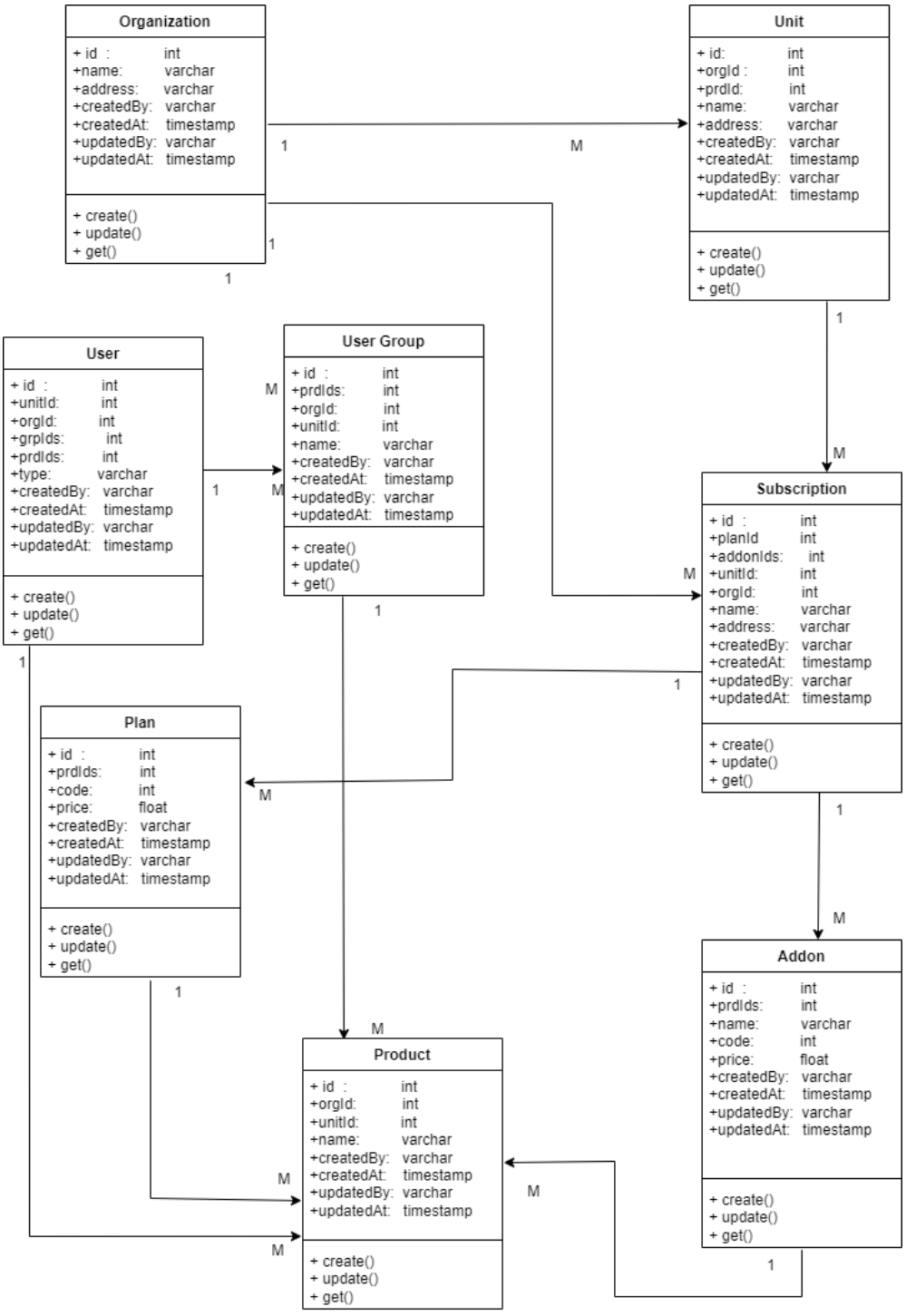
Product

```
{
   "Id": "ObjectId",
   "OrgId": "String",
   "UnitId": "String",
   "Name": "String",
   "Created By": "String",
   "Created At": "Date",
   "Updated By": "String",
   "Updated At": "Date"
}
```

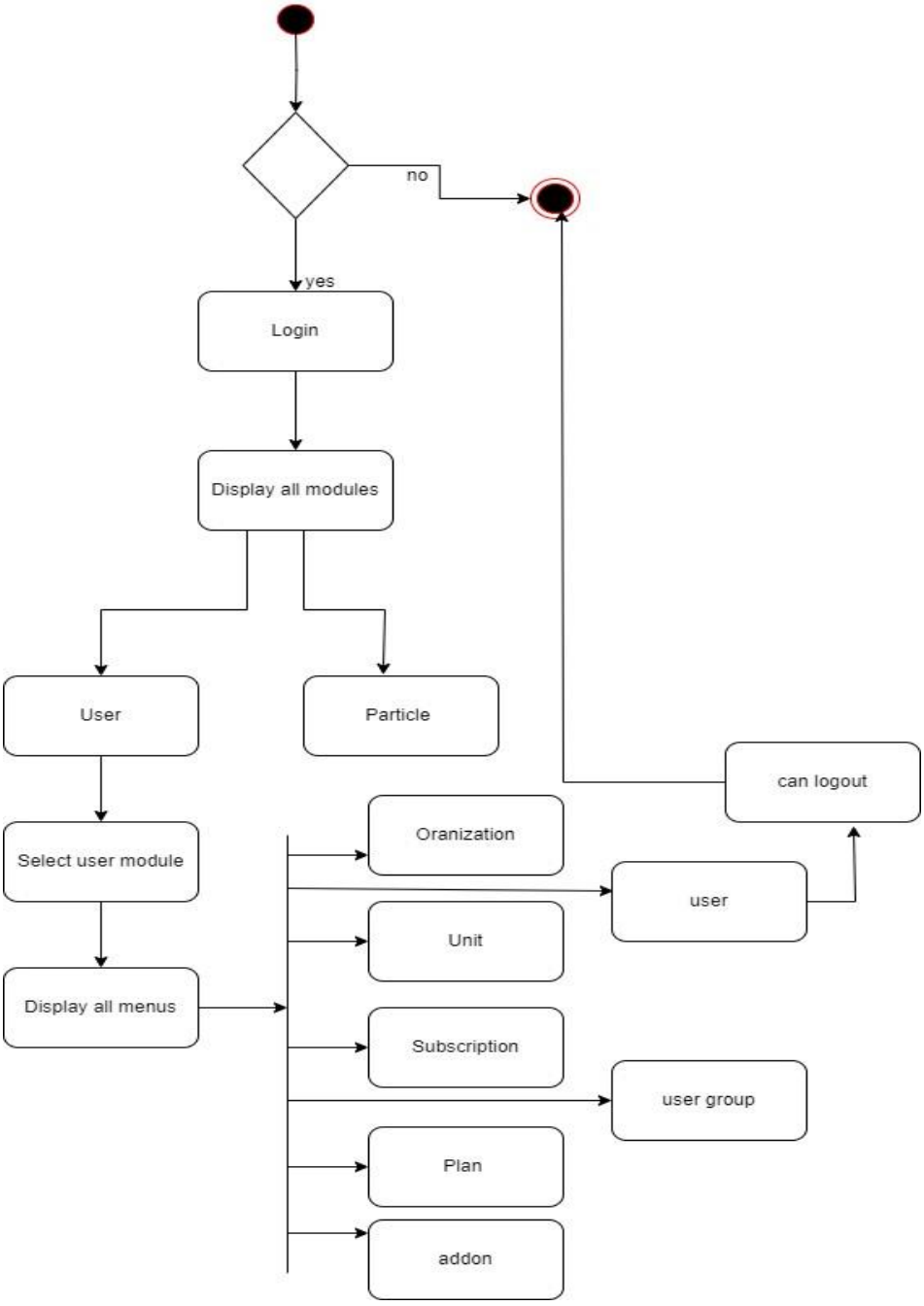
3.4 Use Case Diagram



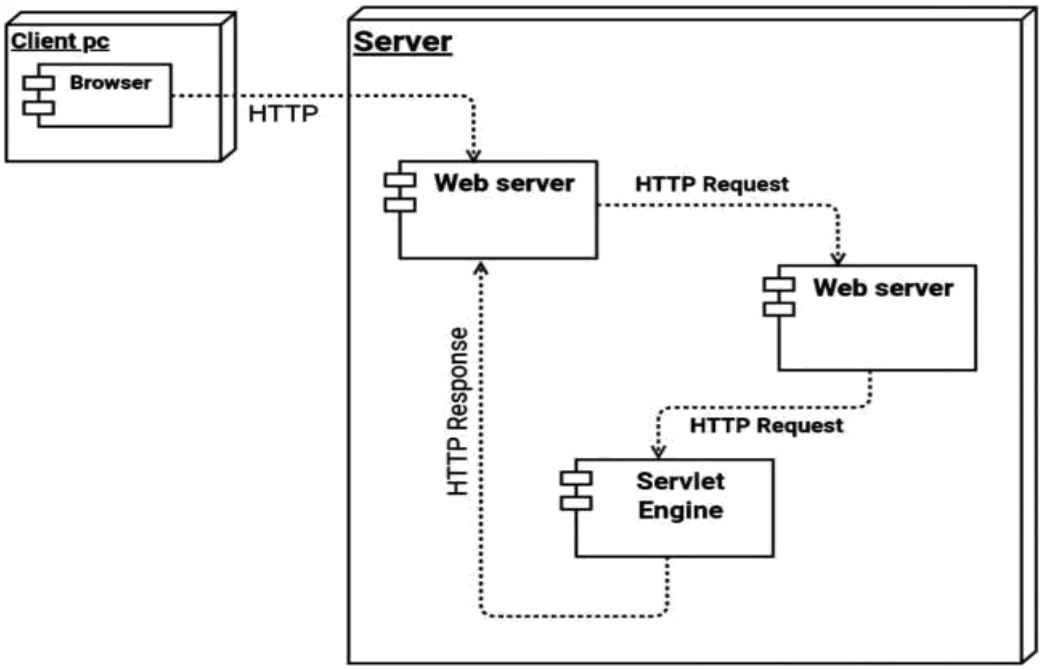
3.5 Class Diagram



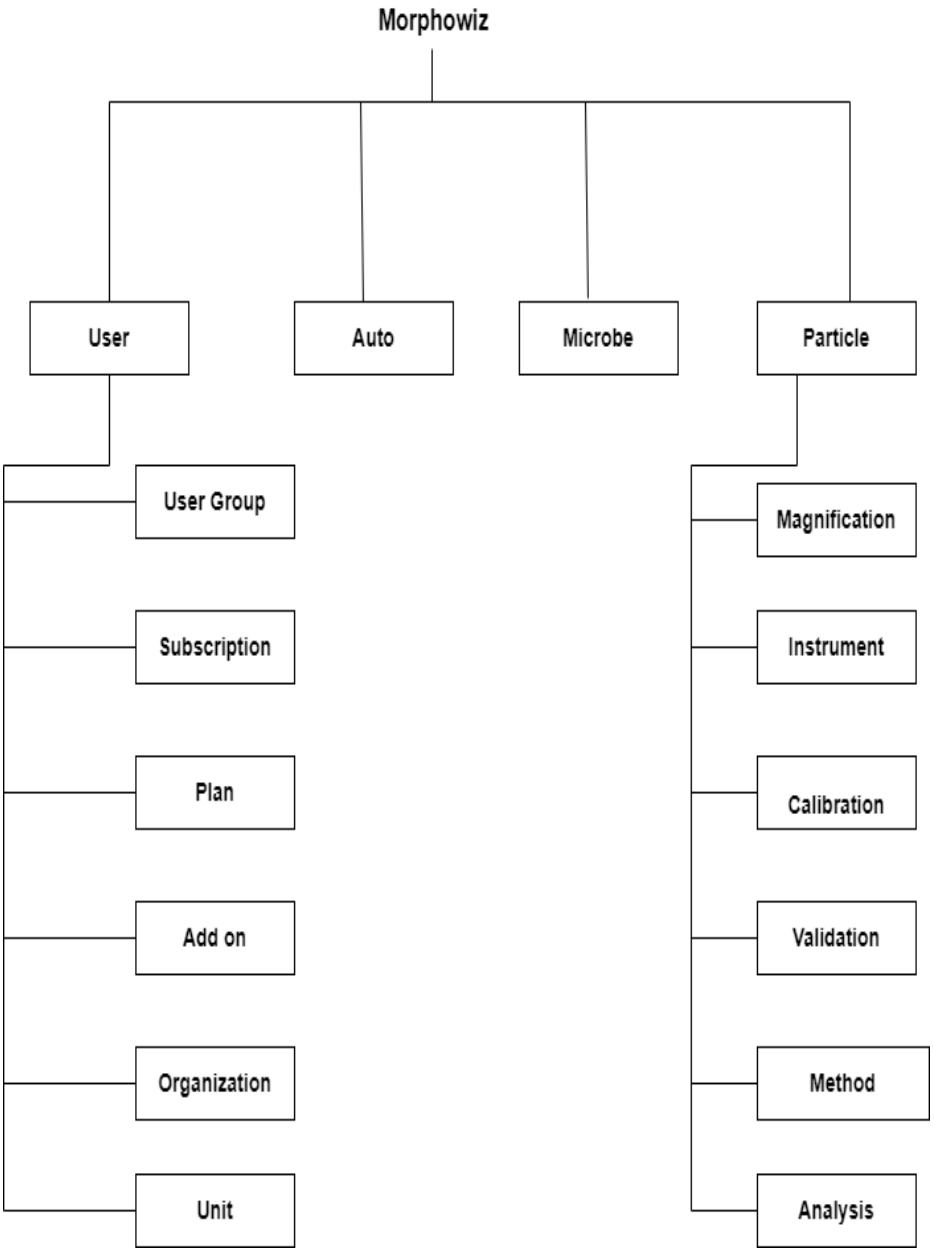
3.6 Activity Diagrams



3.7 Deployment Diagram

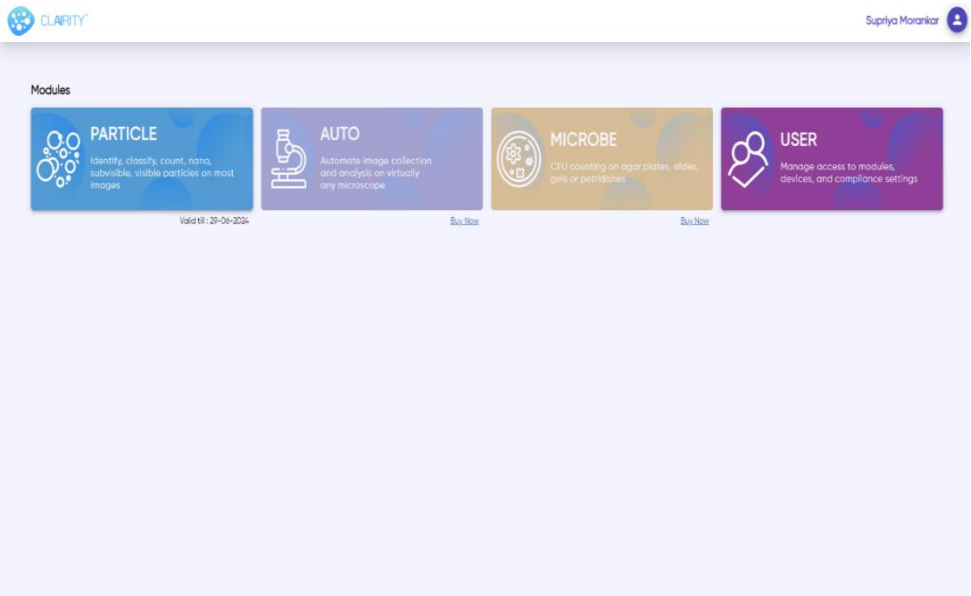


3.8 Module Hierarchy Diagram

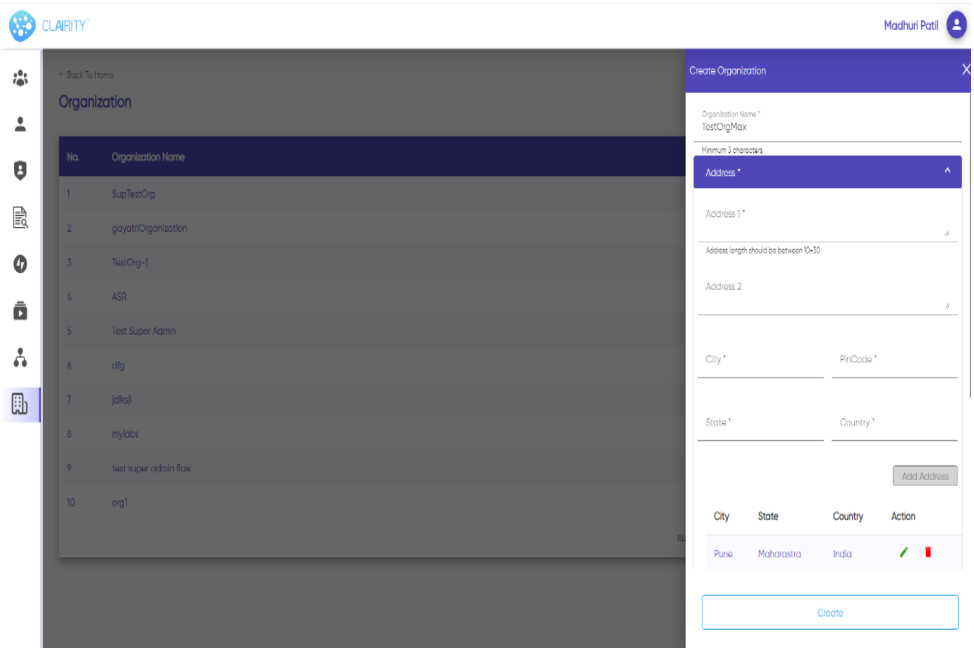


3.9 Input/Output Design

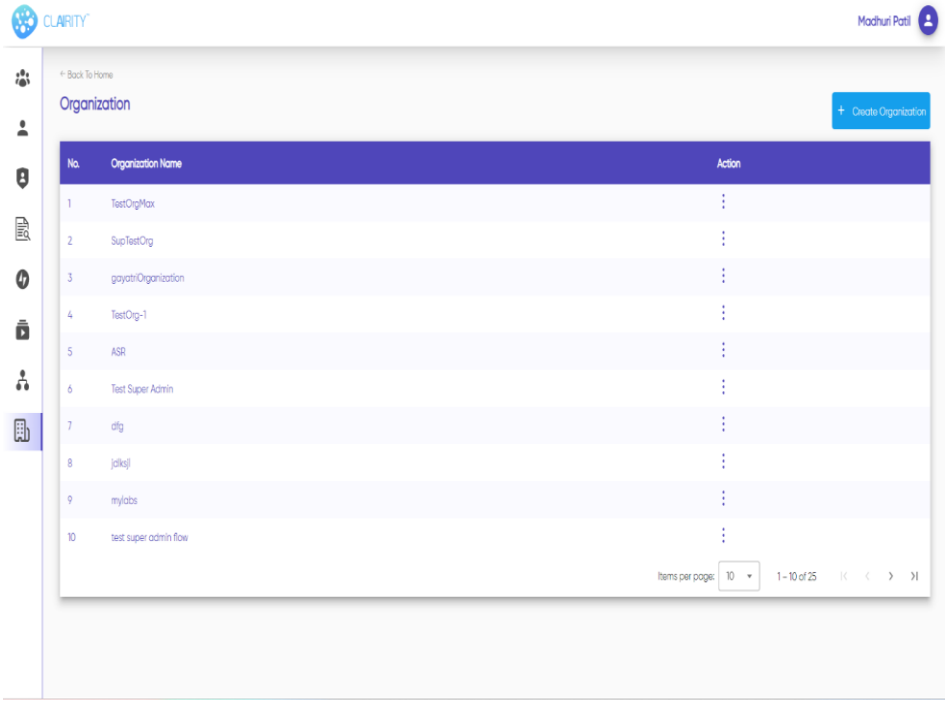
Modules



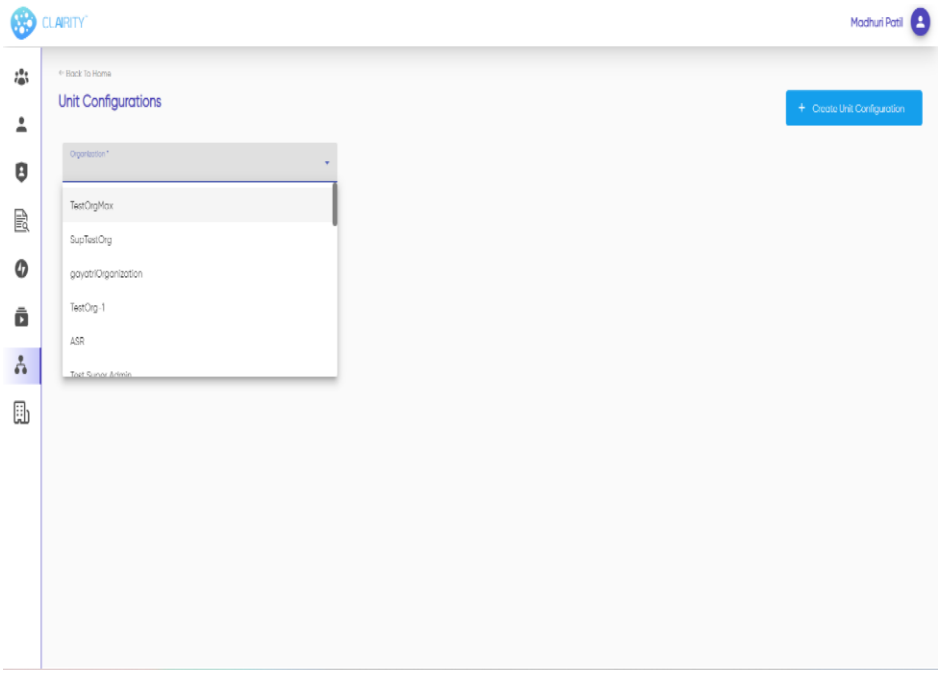
Organization (create organization)



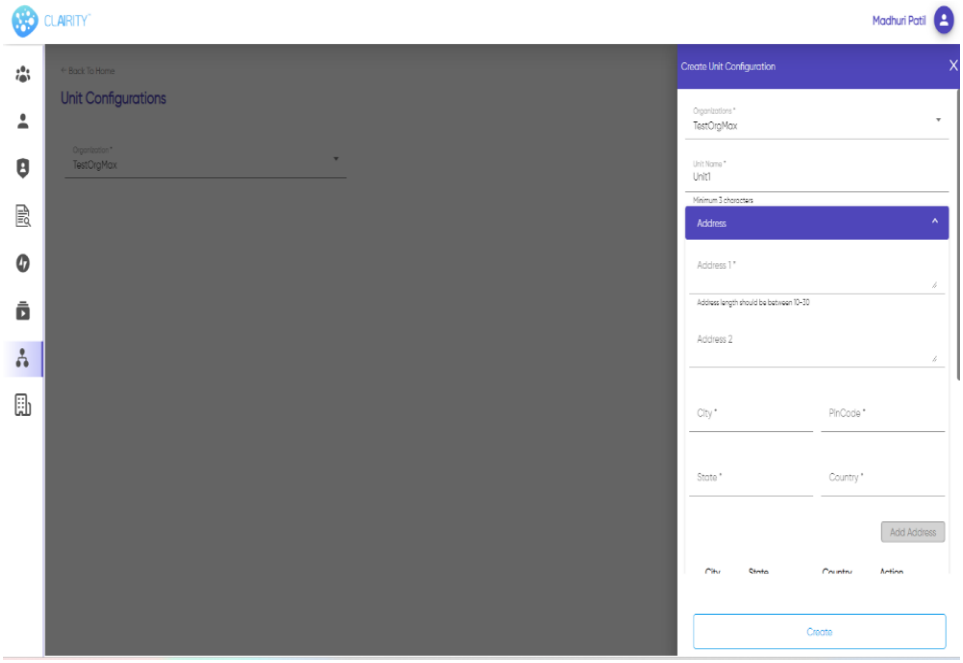
Organization created.



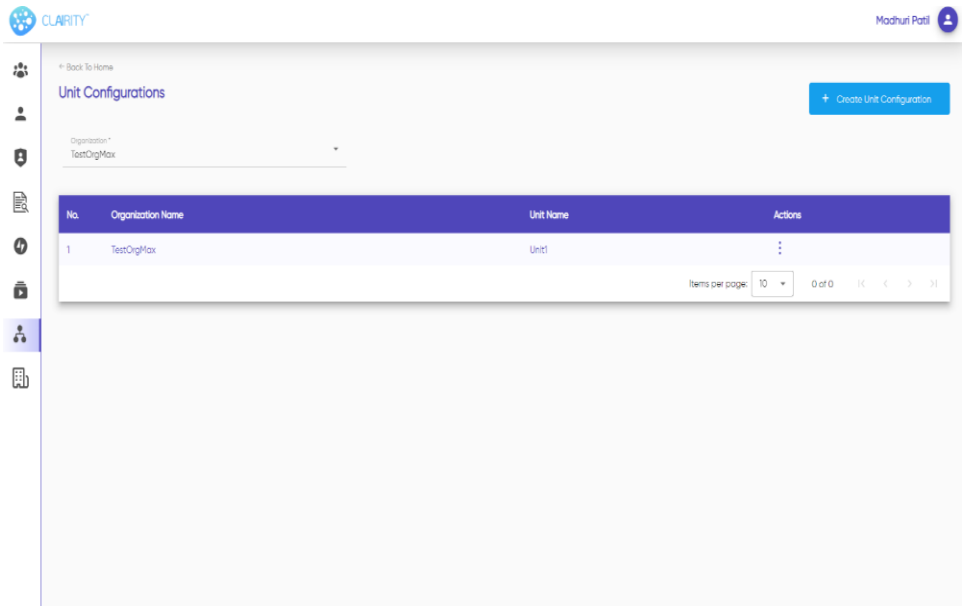
For creating unit select organization



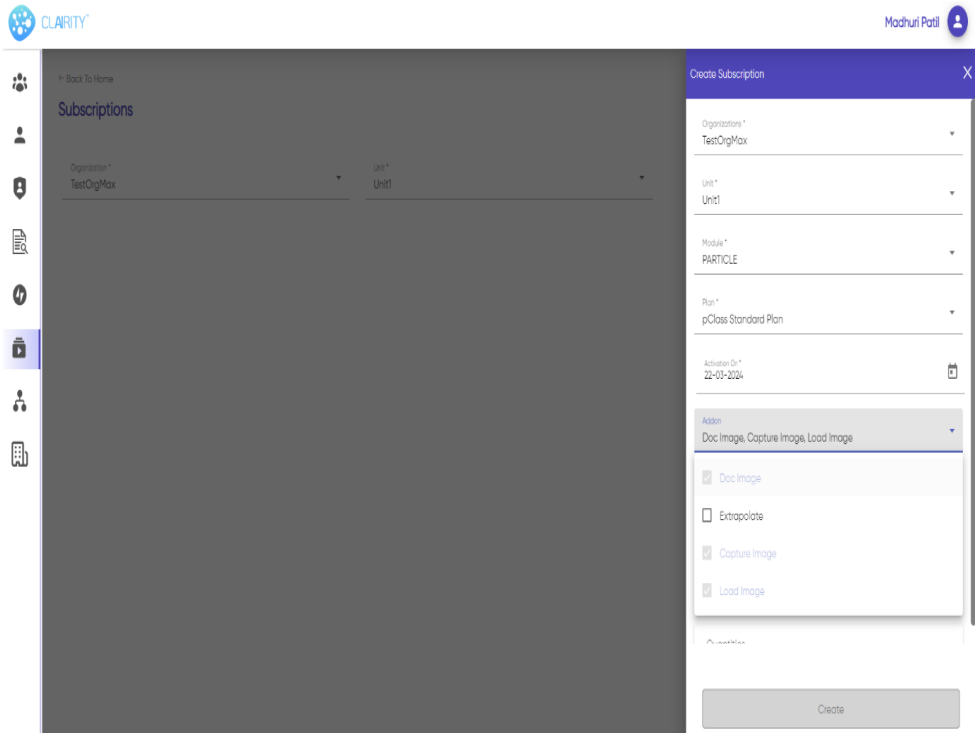
Creating unit



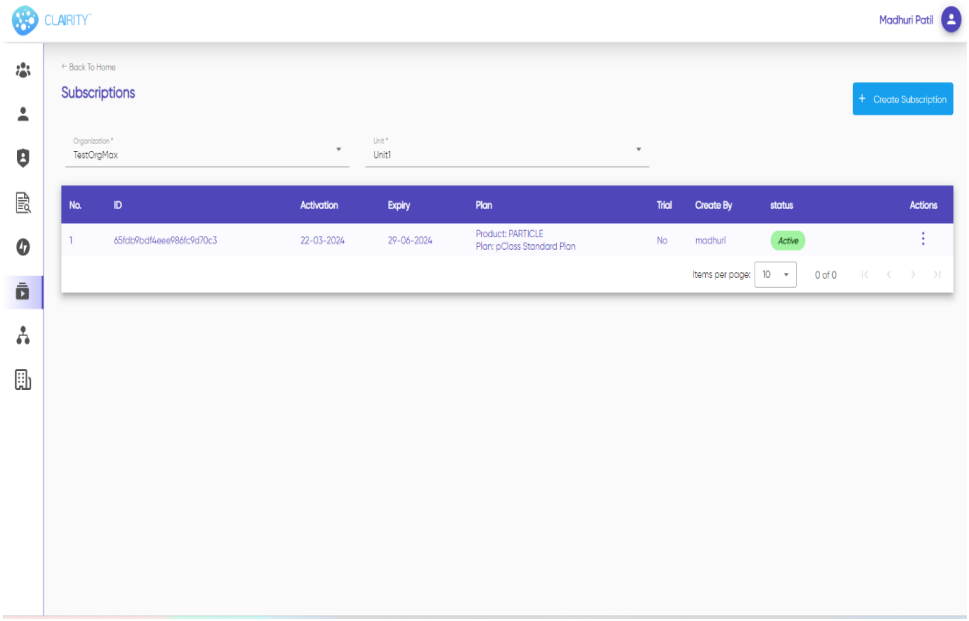
Unit created.



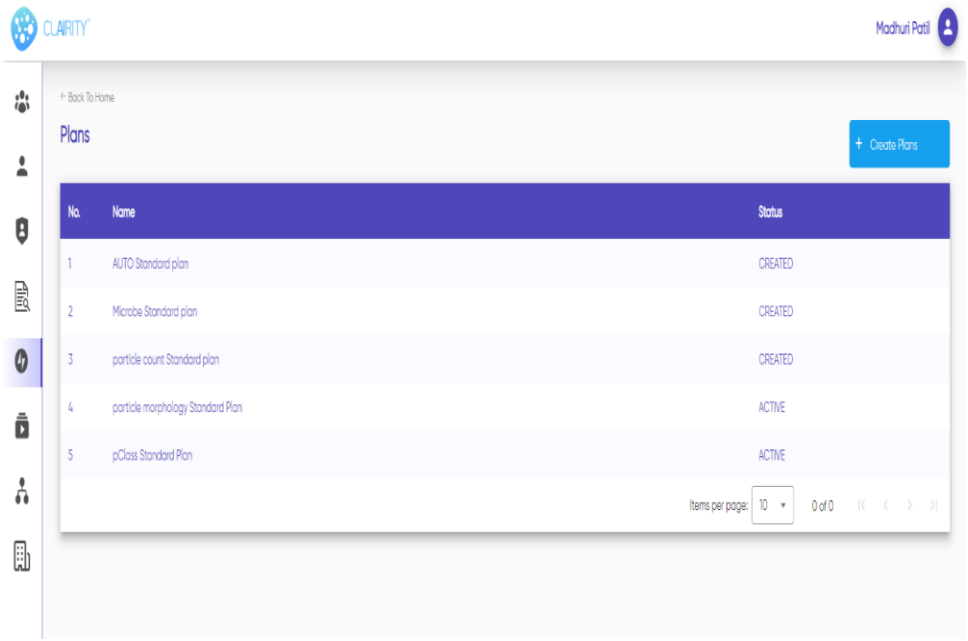
Create subscription for unit in that organization.



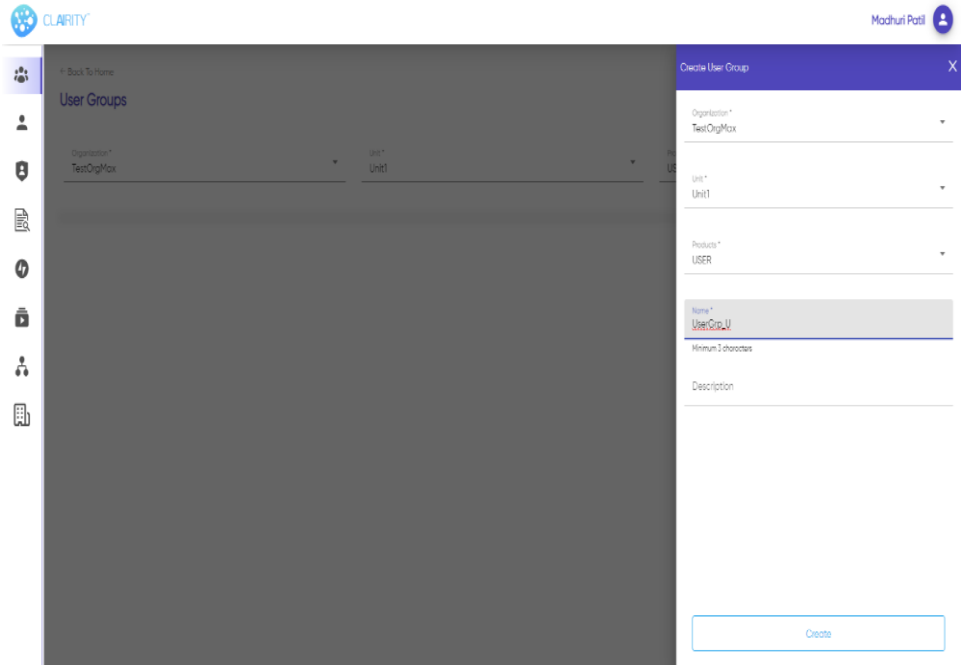
Subscription created.



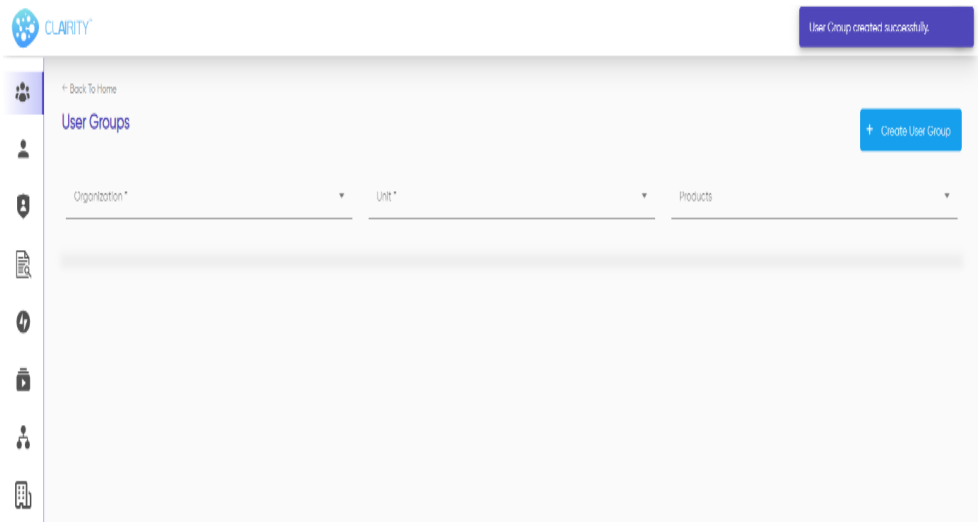
Create a plan.



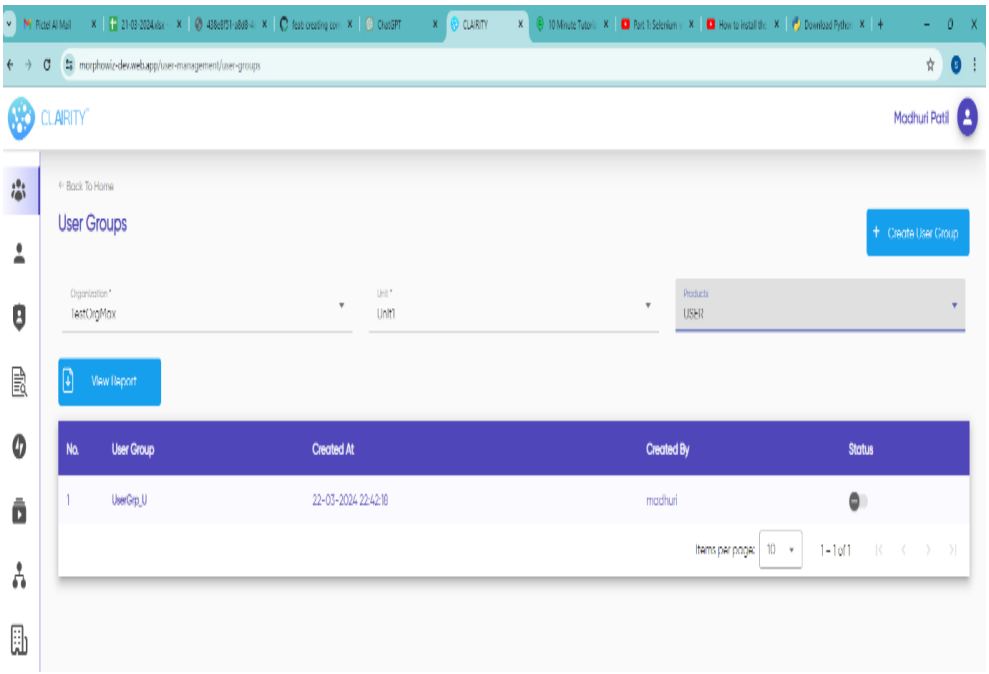
Create user group under unit and organization.



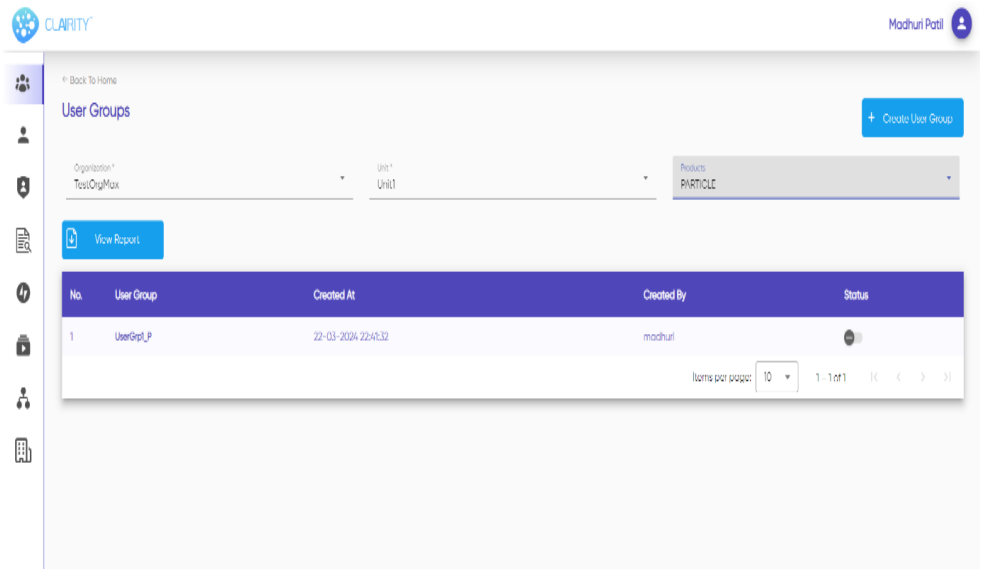
User group created.



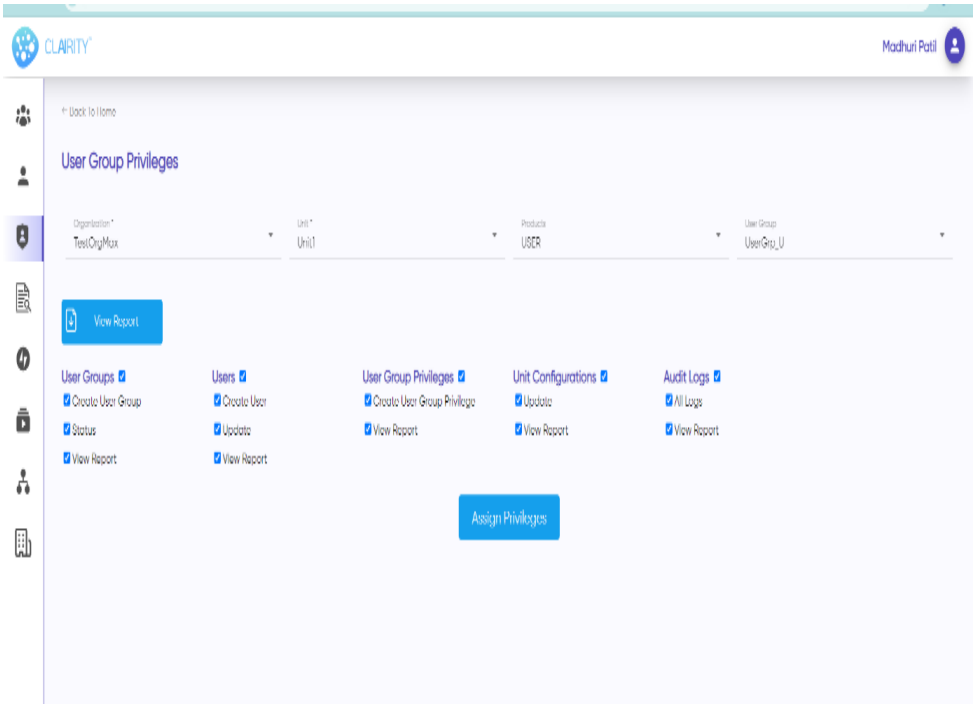
User group created under user module.



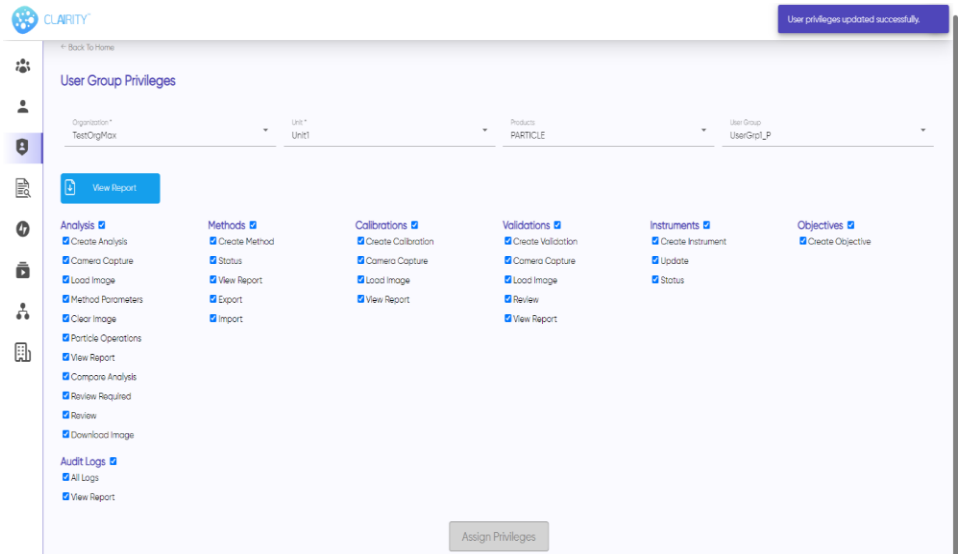
User group created under particle module



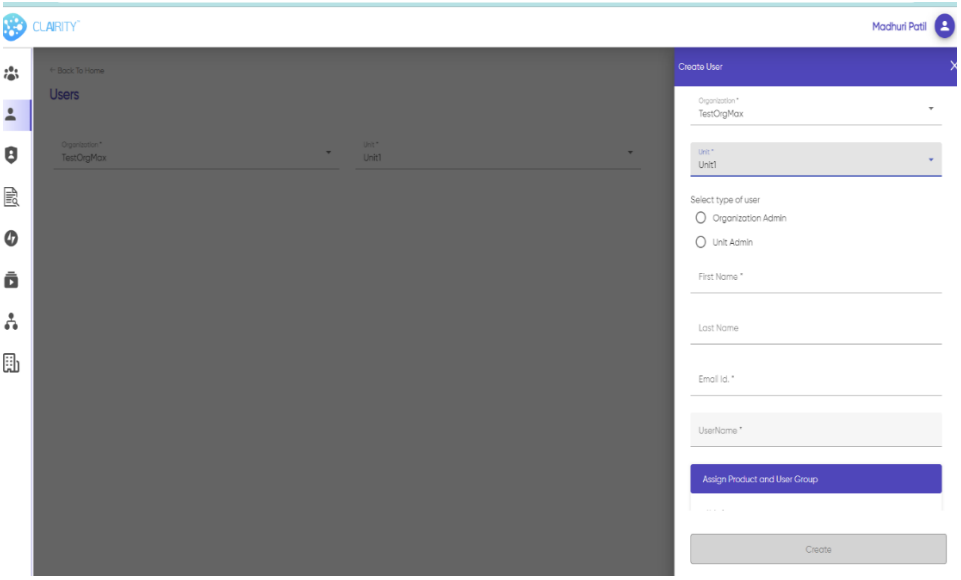
Assign user group privileges to user module.



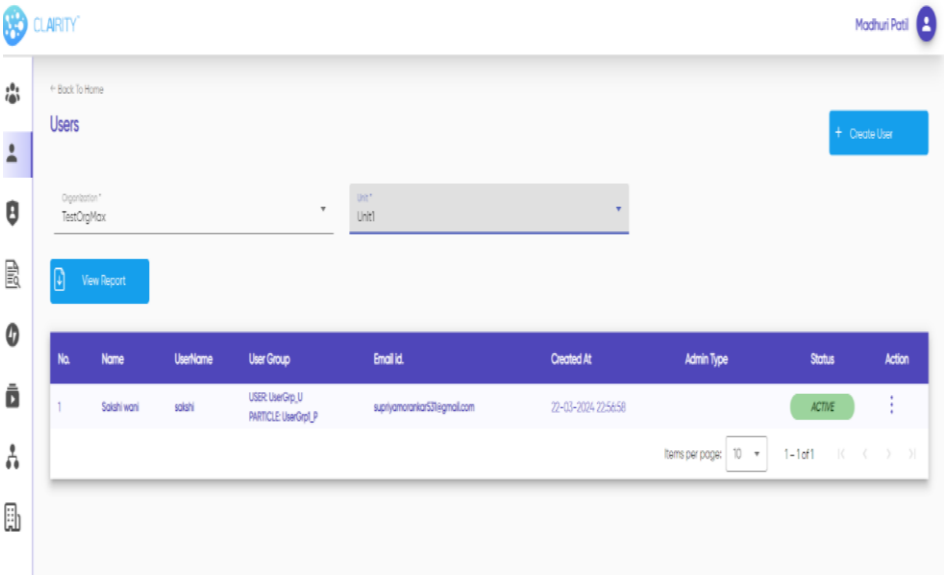
Assign user group privileges to particle module.



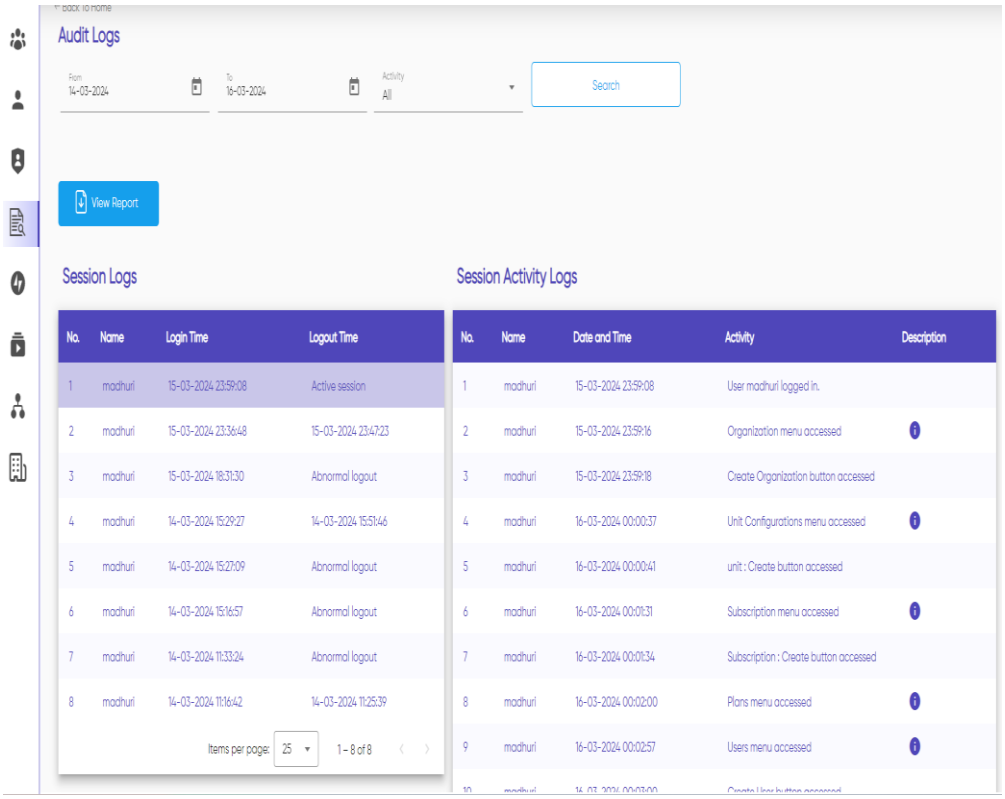
Crete user under unit and organization



User Created



Audit logs.



4. CODING

4.1 Algorithms

Algorithm: User Login

Step 1: Start

Step 2: Input User and Password

Step 3: Validate User and Password

If User and Password are valid:

Step 4: Successful Login

Else:

Step 5: Invalid Credentials

Step 6: End

Algorithm: Create Organization

Step 1: Start

Step 2: Input Organization Details

- Prompt the user to input organization details such as name, description, location, etc.

Step 3: Validate Input

- Check if all required fields are filled and meet any validation criteria.

If input is valid:

Step 4: Save Organization Details

Else:

Step 5: Prompt User to Correct Input

- Display error messages for any invalid fields.

- Return to Step 2 to allow the user to correct the input.

Step 6: End

Algorithm: Create Unit Under Organization

Step 1: Start

Step 2: Select Organization

- Display a list of organizations and prompt the user to select the organization under which the unit will be created.

Step 3: Input Unit Details

- Prompt the user to input unit details such as name, description, location, etc.

Step 4: Validate Input

- Check if all required fields are filled and meet any validation criteria.

If input is valid:

Step 5: Save Unit Details Under Selected Organization

Else:

Step 6: Prompt User to Correct Input

- Display error messages for any invalid fields.

- Return to Step 3 to allow the user to correct the input.

Step 7: End

Algorithm: Create Subscription for Unit

Step 1: Start

Step 2: Select Unit

- Display a list of units and prompt the user to select the unit for which the subscription will be created.

Step 3: Input Subscription Details

- Prompt the user to input subscription details such as plan, duration, price, etc.

Step 4: Validate Input

- Check if all required fields are filled and meet any validation criteria.

If input is valid:

Step 5: Save Subscription Details for Selected Unit

Else:

Step 6: Prompt User to Correct Input

- Display error messages for any invalid fields.

- Return to Step 3 to allow the user to correct the input.

Step 7: End

Algorithm: Create User

Step 1: Start

Step 2: Input User Details

- Prompt the administrator to input user details such as username, email, password, role, etc.

Step 3: Validate Input

- Check if all required fields are filled and meet any validation criteria.

If input is valid:

Step 4: Save User Details

Else:

Step 5: Prompt Administrator to Correct Input

- Display error messages for any invalid fields.

- Return to Step 2 to allow the administrator to correct the input.

Step 6: End

4.2 Code snippets

```
func (s *SubscriptionServer) CreateOrganization(ctx context.Context, in *subscription.CreateOrganizationRequest) (*subscription.CreateOrganizationResponse, error) {
  log := logging.Extract(ctx)
  md, ok := metadata.FromIncomingContext(ctx)
  if !ok {
    return nil, status.Error(codes.InvalidArgument, errors.MissingRequiredHeadersErrMsg)
  }
  userIDs := md.Get(constants.UserIDHeader)
  if len(userIDs) == 0 {
    return nil, status.Errorf(codes.InvalidArgument, errors.MissingRequiredHeaderErrMsg, constants.UserIDHeader)
  }

  userID := userIDs[0]
  check, err := s.checkPermission(ctx, userID)
  if err != nil {
    log.Error("failed to check permission", zap.Error(err))
    return nil, errors.InternalServerErr
  }
  if !check {
    return nil, errors.InsufficientPermissionErr(fmt.Sprintf("Create %s", organizationModelName))
```

```
  }
  //Check if organization with same name already exists
  sr := s.organizations.FindOne(ctx, bson.D{{Key: "name", Value:
in.Name}})
  if sr.Err() == nil {
      return  nil,  errors.AlreadyExistsErr(organizationModelName,
"name")
  } else if sr.Err() != mongo.ErrNoDocuments {
      log.Error("failed to check if organization with same name
already exists", zap.Error(sr.Err()))
    return nil, errors.InternalServerErr
  }

  req := &Organization{
    Name:      in.Name,
    Logo:      in.Logo,
    Tags:      in.Tags,
    Addresses: fromProtoToAddresses(in.Addresses),
    Contacts:  fromProtoToContacts(in.Contacts),
  }
  if in.Description != nil {
    req.Description = *in.Description
  }
  req.Created(userIDs[0])
  if err != nil {
    log.Error("failed to insert organization", zap.Error(err))
```

```
func (s *SubscriptionServer) CreateSubscription(ctx context.Context, in *subscription.CreateSubscriptionRequest) (*subscription.CreateSubscriptionResponse, error) {
  log := logging.Extract(ctx)
  md, ok := metadata.FromIncomingContext(ctx)
  if !ok {
    return nil, status.Error(codes.InvalidArgument, errors.MissingRequiredHeadersErrMsg)
  }
  userIDs := md.Get(constants.UserIDHeader)
  if len(userIDs) == 0 {
    return nil, status.Errorf(codes.InvalidArgument, errors.MissingRequiredHeaderErrMsg, constants.UserIDHeader)
  }
  check, err := s.checkPermission(ctx, userIDs[0])
  if err != nil {
    log.Error("failed to check permission", zap.Error(err))
    return nil, errors.InternalServerErr
  }
  if !check {
    return nil, errors.InsufficientPermissionErr(fmt.Sprintf("Create %s", subscriptionModelName))
  }
  oid, err := primitive.ObjectIDFromHex(in.OrgId)
  if err != nil {
```

```
        return    nil,    errors.NotFoundErr(organizationModelName,
in.OrgId)
  }
  _, err = s.findOrganization(ctx, oid)
  if err != nil && err != mongo.ErrNoDocuments {
                    log.Error(fmt.Sprintf(errors.FailedToFindErrMsg,
"organization", in.OrgId), zap.Error(err))
    return nil, errors.InternalServerErr
  } else if err == mongo.ErrNoDocuments {
        return    nil,    errors.NotFoundErr(organizationModelName,
in.OrgId)
  }
  uid, err := primitive.ObjectIDFromHex(in.UnitId)
  if err != nil {
    return nil, errors.NotFoundErr(unitModelName, in.UnitId)
  }
  _, err = s.findUnit(ctx, uid, in.OrgId)
  if err != nil && err != mongo.ErrNoDocuments {
        log.Error(fmt.Sprintf(errors.FailedToFindErrMsg,    "unit",
in.UnitId), zap.Error(err))
    return nil, errors.InternalServerErr
  } else if err == mongo.ErrNoDocuments {
    return nil, errors.NotFoundErr(unitModelName, in.UnitId)
  }
```

```
func (s *SubscriptionServer) CreateUnit(ctx context.Context, in
*subscription.CreateUnitRequest)
(*subscription.CreateUnitResponse, error) {
  log := logging.Extract(ctx)
  md, ok := metadata.FromIncomingContext(ctx)
  if !ok {
                return       nil,       status.Error(codes.InvalidArgument,
errors.MissingRequiredHeadersErrMsg)
  }

errors.AlreadyExistsErr(addonModelName, "name")
  } else if sr.Err() != mongo.ErrNoDocuments {
   log.Error("failed to check if unit with same name already exists",
zap.Error(sr.Err()))
   return nil, errors.InternalServerErr
  }
req := &Unit{
   OrgId:              in.OrgId,
   ParentId:           in.ParentId,
   Name:               in.Name,
   Logo:               in.Logo,
   Tags:               in.Tags,
   Addresses:          fromProtoToAddresses(in.Addresses),
   Contacts:           fromProtoToContacts(in.Contacts),
                                                ProductConfigurations:
fromProtoToParameters(in.ProductConfigurations),
```

```
    ProductSupports:       fromProtoToProductSupports(in.Product
Supports),
  }
  if in.Description != nil {
    req.Description = *in.Description
  }
  req.Created(userIDs[0])
```

```
func (s *SubscriptionServer) CreatePlan(ctx context.Context, in
*subscription.CreatePlanRequest)
(*subscription.CreatePlanResponse, error) {
  log := logging.Extract(ctx)
  md, ok := metadata.FromIncomingContext(ctx)
  if !ok {
                return       nil,       status.Error(codes.InvalidArgument,
errors.MissingRequiredHeadersErrMsg)
  }
  userIDs := md.Get(constants.UserIDHeader)
  if len(userIDs) == 0 {
                return       nil,       status.Errorf(codes.InvalidArgument,
errors.MissingRequiredHeaderErrMsg, constants.UserIDHeader)
  }
  check, err := s.checkPermission(ctx, userIDs[0])
  if err != nil {
    log.Error("failed to check permission", zap.Error(err))
    return nil, errors.InternalServerErr}
```

5.TESTING

5.1 Test Strategy

Software testing is a critical element of software quality assurance & represents the ultimate review of specification, design and code generation.

It is the process of executing a program with a primary objective of finding errors. Testing gives the guarantee that the software does not fail and runs according to its specification and in the way the end user expects.

This can be done by various software testing techniques which provide a systematic guidance for designing tests that exercise the internal logic of software components and exercise the input and output domains of the program to uncover errors in programming. functions, behavior, and performance.

Testing is the exposure of system to trial input to see whether it produces correct output. Testing is the process of detecting presence of faults. Once the source code has been generated, software must be tested to uncover as many errors as possible before delivery to your customer. Our goal is to design a series of test cases that have likelihood of finding errors. That's where Software testing Techniques enter the picture. A set of test cases designed to exercise both internal login and external requirements is designed and documented, expected results are defined and actual results are recorded.

5.2 Unit Test Plan

1. Introduction:
    - The unit test plan aims to verify the correctness and reliability of individual components/modules within the system as per the defined scope.
2. Test Objectives:
    - Validate the functionality and behaviour of each component/module.
    - Ensure that direct and derived measurement parameters are accurately calculated.
3. Scope:
    - In-scope Items:
        - Functions/methods responsible for capturing images, detecting particles, and calculating parameters.
        - Integration points between microscope, camera, and software components.
    - Out-of-scope Items:
        - External dependencies such as hardware drivers or third-party libraries.
4. Test Cases:

- Test Case 1: Test Image Capture
    - Verify that images captured by the digital camera meet the specified resolution and quality standards.
- Test Case 2: Particle Detection
    - Ensure that particles are accurately detected and segmented from background noise in captured images.
- Test Case 3: Parameter Calculation
    - Validate the accuracy of calculated parameters such as area, perimeter, diameter, and aspect ratio.
- Test Case 4: Statistical Analysis
    - Verify the correctness of statistical analysis algorithms for generating particle size distribution and morphology data.

5. Test Execution:
    - Procedure:
        - Set up the testing environment with mock objects or simulated data.

- Execute individual unit tests for each component/module using a testing framework.
- Record test results and identify any failures or deviations from expected behaviour.

6. Defect Management:
    - Report defects identified during unit testing using the designated issue tracking system.
    - Collaborate with developers to resolve defects and verify fixes.
7. Schedule and Resources:
    - Allocate time and resources for conducting unit testing activities based on project timelines and priorities.
    - Identify resources needed for testing, including testers and testing tools.
8. Roles and Responsibilities:
    - Testers: Write and execute unit tests, report defects.
    - Developers: Assist with test case design, resolve defects.
    - Project Manager: Coordinate unit testing activities, allocate resources.

5.3 Acceptance Test Plan

1. Introduction:

- The acceptance test plan outlines the strategy and procedures for validating that the system meets the specified requirements and is ready for deployment.

2. Test Objectives:

- Verify that the system functionalities align with the defined scope and acceptance criteria.
- Ensure that the system accurately performs particle size and shape analysis as per user expectations.
- Validate the integration and interaction between microscope, digital camera, and analysis software components.

3. Scope:

- In-scope Items:
    - End-to-end testing of the entire system, including user interfaces, workflows, and analytical capabilities.
    - Validation of system performance, usability, and accuracy of results.
- Out-of-scope Items:

- Testing of external dependencies not directly controlled by the system.

4. Test Cases:

- Test Case 1: System Initialization
    - Objective: Verify that the system initializes successfully, and all components are operational.
    - Steps:
        1. Power on the system and verify startup sequence.
        2. Ensure all hardware components (microscope, camera) are functioning.
        3. Launch the analysis software and check for any initialization errors.
    - Expected Outcome: System initializes without errors, and all components are ready for operation.
- Test Case 2: Image Acquisition
    - Objective: Ensure that images are captured accurately and transferred to the analysis software without loss of quality.
    - Steps:
        1. Capture images using the digital camera.

2. Verify that images are transferred to the analysis software.
3. Check for any loss of image quality during transfer.

- Expected Outcome: Images are captured and transferred seamlessly without loss of quality.

- Test Case 3: Particle Analysis
  - Objective: Validate the accuracy of particle detection, parameter calculation, and statistical analysis.
  - Steps:
    1. Analyse sample images containing particles of various shapes and sizes.
    2. Verify that particles are accurately detected and segmented.
    3. Validate the correctness of calculated parameters such as area, perimeter, and diameter.
    4. Verify the accuracy of statistical analysis for generating particle size distribution and morphology data.
  - Expected Outcome: Particle analysis results are accurate and consistent with expected values.

- Test Case 4: User Interface
    - Objective: Verify that the user interface is intuitive, responsive, and provides necessary feedback to users.
    - Steps:
        1. Navigate through different sections of the user interface.
        2. Perform common tasks such as image capture, analysis initiation, and result viewing.
        3. Verify that the interface provides appropriate feedback for user actions.
    - Expected Outcome: User interface is easy to use, responsive, and provides clear feedback to users.

5. Acceptance Criteria:

- Criteria for Pass/Fail:
    - All acceptance test cases must pass without any critical defects.
    - The system must meet or exceed performance targets specified in the requirements.
    - User interfaces must be intuitive and meet usability standards.

- Exit Criteria:
    - All acceptance test cases are executed and pass successfully.
    - Any identified defects are resolved or documented for future resolution.
    - Stakeholders provide formal approval to proceed with deployment based on acceptance test results.

6. Test Execution:

- Procedure:
    - Execute acceptance test cases based on the predefined test plan.
    - Record test results and document any deviations or issues encountered.
    - Verify that all acceptance criteria are met.

7. Defect Management:

- Report defects identified during acceptance testing using the designated issue tracking system.
- Collaborate with development teams to resolve defects and verify fixes.

8. Schedule and Resources:

- Allocate time and resources for conducting acceptance testing activities based on project timelines and priorities.

5.4 Test Cases

Test case ID	Scenario to test	Steps	Expected Result	Actual Result	Result Pass/Fail
TC-01	URL test	1.Open Browser 2.Enter correct URL	Open Site	Open site	Pass
TC-02	URL test	1.Open Browser 2.Enter incorrect URL	Site not open	Site not open	Pass
TC-03	Login	1.Enter correct URL 2.Login with correct username and password	Login successfully and module page is open.	Login successfully and module page is open.	Pass
TC-04	Login	1.Enter correct URL 2.Login with incorrect username	Login unsuccessful	Login unsuccessful	Pass

		and password			
TC-05	Login	1.Enter correct URL 2.Login with incorrect username and correct password	Login unsuccessful	Login successful	Fail
TC-05	Login	1.Enter correct URL 2.Login with correct username and incorrect password	Login unsuccessful	Login unsuccessful	Pass

TC-6	Module	1.Enter correct Username and password. 2.module page open click on particle module.	Login successfully. module page is open. Particle module should display.	1.Login successfully. module page is open. Particle module should display.	Pass
TC-7		1.Enter correct Username and password. 2.module page open click on user module.	Login successfully. module page is open. User module should display.	Login successfully. module page is open. User module should display.	Pass
TC- 8		1.In user module click on create organization. 2.Fill correct details	Organization is created	Organization is created	Pass

TC-9		1.In user module click on create organization. 2.Fill incorrect details	Organization is created	Organization is not created	Fail
TC-10		1.In user module click on create organization. 2.Fill incorrect details	Organization is not created	Organization is not created	Pass
TC-11		1.Create unit Under that created organization. 2.Enter organization.	Select that organization. Organization is displayed.	Select that organization. Organization is displayed.	Pass

TC-12		1.select organization 2.click on create unit button 3.Enter correct details for creating unit.	In that organization Unit created successfully.	In that organization Unit created successfully.	Pass
TC-13		1.select organization 2.click on create unit button 3.Enter incorrect details for creating unit.	In that organization Unit is not created.	In that organization Unit is not created.	Pass

TC-14		1.select organization. 2.select unit under organization.	In that organization and selected unit Is displayed	In that organization and selected unit Is displayed	Pass
TC-15		1.select organization 2.select unit under organization 3.click on create subscription. 4.filled correct details and all mandatory fields.	Subscription is created successfully under unit and organization.	Subscription is created successfully under unit and organization.	Pass

TC-15		1.select organization 2.select unit under organization 3.click on create subscription. 4.filled incorrect details and all mandatory fields.	Subscription is not created.	Subscription is not created.	Pass
TC-15		1.select organization 2.select unit under organization 3.create user group with correct data	User group created successfully.	User group created successfully.	Pass

TC-15		1.select organization 2.select unit under organization 3.create user group with incorrect data	User group not created	User group not created	Pass
TC-16		1.select organization 2.select unit under organization 3.Update user group with incorrect data	User group not updated.	User group not updated.	Pass

TC-16		1.select organization 2.select unit under organization 3.Update user group with correct data	User group updated successfully	User group updated successfully	Pass
TC-16		1.select organization 2.select unit under organization 3.select user group. 4.under that user group create user. 5.click on Create user button. 6.fill incorrect data	User not created	User not created	Pass

TC-16		1.select organization 2.select unit under organization 3.select user group. 4.under that user group create user. 5.click on Create user button. 6.fill correct data	User created	User created	Pass

5.5 Defect report

Defect ID	Summary	Severity	Priority
D01	Validation on some fields not available.	High	High

Limitations of proposed system

1. Cost and Resource Requirements: Setting up and maintaining a microscopy-based image analysis system can be expensive, requiring investment in specialized equipment, software licenses, and skilled personnel. Additionally, the system may require regular calibration, maintenance, and updates to ensure optimal performance.
2. User Expertise: Effective utilization of the system may require users to have a certain level of expertise in microscopy, image analysis techniques, and software operation. Training and proficiency in sample preparation, image acquisition, and data interpretation may be necessary to obtain accurate and reliable results.

Proposed Enhancements

1. Real-Time Analysis: Implement real-time analysis capabilities to allow users to view analysis results as images are captured, providing immediate feedback during sample inspection.
2. Automated Calibration: Introduce automated calibration features to streamline the process of calibrating the microscope and camera settings for accurate particle size measurement.
3. Advanced Statistical Analysis: Enhance statistical analysis capabilities to include more advanced metrics and visualizations, such as standard deviation, confidence intervals, and histograms.
4. Customizable Reporting: Introduce customizable reporting options that allow users to generate customized reports with specific analysis parameters, metrics, and visualizations.
5. Integration with External Devices: Enable integration with external devices such as robotic sample handlers or automated stage controllers to automate sample preparation and handling processes.
6. Machine Learning Algorithms: Incorporate machine learning algorithms for particle detection and classification to improve accuracy and efficiency in identifying particles of interest.

7. Cloud-Based Analysis: Implement cloud-based analysis capabilities to offload processing tasks to remote servers, enabling scalability and reducing the computational burden on local hardware.
8. Enhanced User Interface: Improve the user interface with intuitive navigation, interactive visualization tools, and customizable layouts to enhance user experience and usability.
9. Multi-Language Support: Add support for multiple languages to accommodate users from diverse linguistic backgrounds and regions.

Conclusion

In conclusion, the particle size analysis system described within the project scope offers a comprehensive solution for accurate and efficient particle size and shape analysis. By leveraging microscopy, digital imaging, and advanced analysis algorithms, the system addresses the need for both manual and automated measurements, catering to particles ranging from 1 µm and larger.

The integration of a microscope, digital camera, and analysis software facilitates seamless image capture, transfer, and analysis, enabling users to obtain valuable insights into particle morphology and distribution. Microscope Image Analysis (MIA) techniques, coupled with unique algorithms, ensure precise measurement parameters and statistical analysis, leading to reliable results.

The system's versatility is further enhanced by its ability to handle particles of various shapes and sizes, including non-spherical particles, which are challenging to analyze using conventional methods like laser diffraction. Moreover, the proposed enhancements, such as real-time analysis, automated calibration, and advanced statistical analysis, promise to enhance the system's capabilities and user experience.

Overall, the particle size analysis system represents a significant advancement in particle characterization technology, offering researchers and analysts a powerful tool for a wide range of applications in fields such as pharmaceuticals, materials science, and environmental monitoring. Its ability to provide accurate,

rapid, and reliable results makes it an invaluable asset for laboratories and research facilities seeking to gain deeper insights into particle behavior and properties.

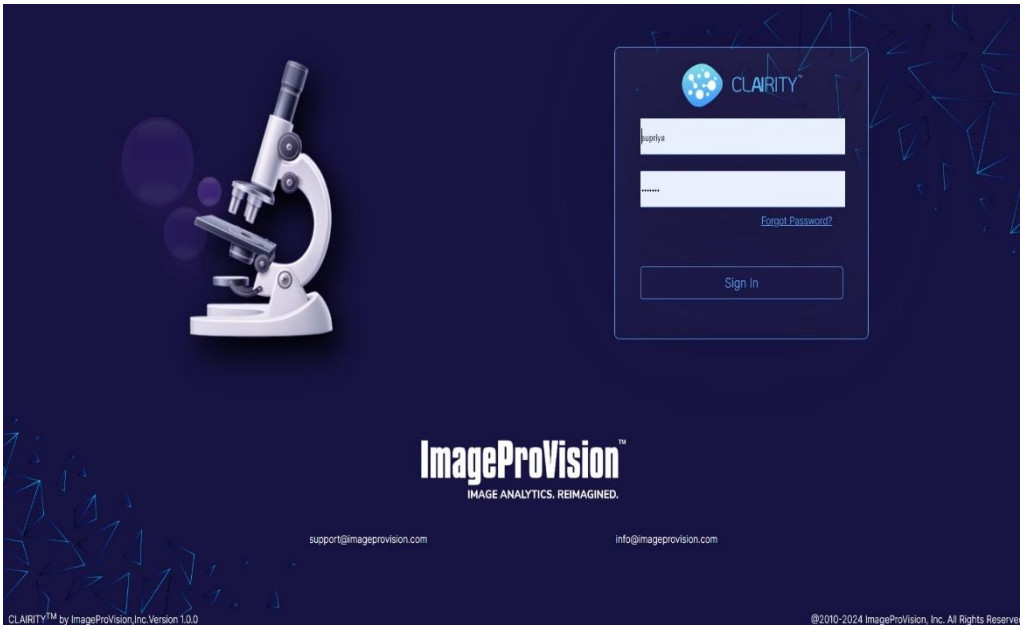
9. Bibliography

Websites:

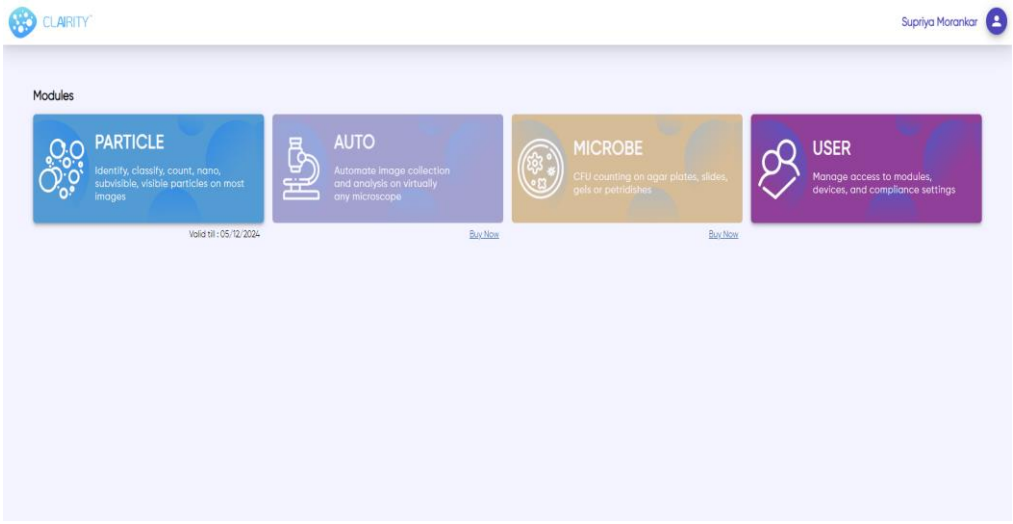
- https://go.dev/doc/effective_go
- https://docs.nats.io/
- https://grpc.io/docs/
- https://protobuf.dev/getting-started/gotutorial/
- https://authzed.com/docs/spicedb/getting started/discovering-spicedb
- https://www.geeksforgeeks.org/go-programming-language-introduction/
- https://docs.nats.io/nats-concepts/core-nats/pubsub/pubsub_walkthrough

10. User Manual

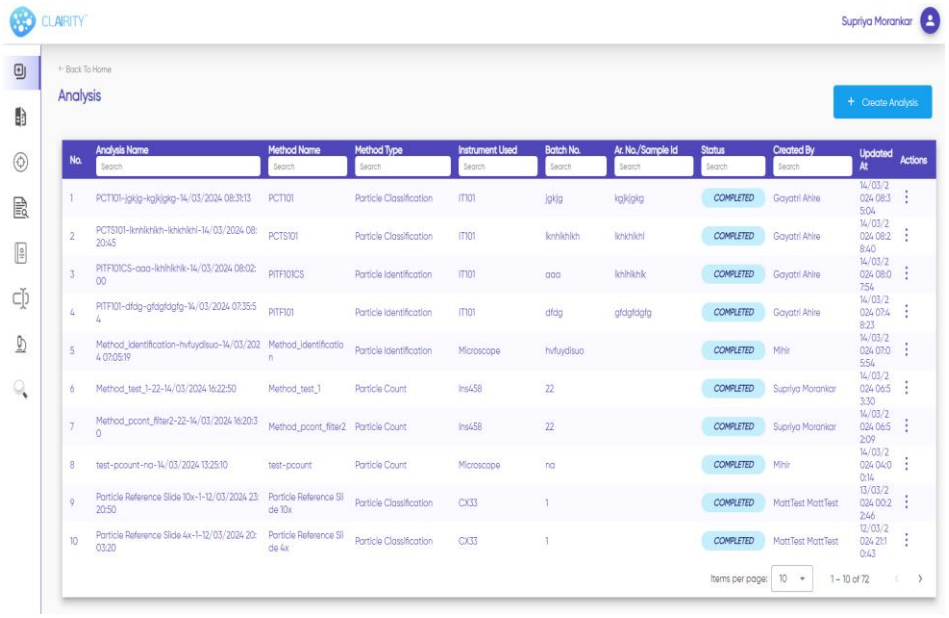
Firstly, login with created user from admin/super admin



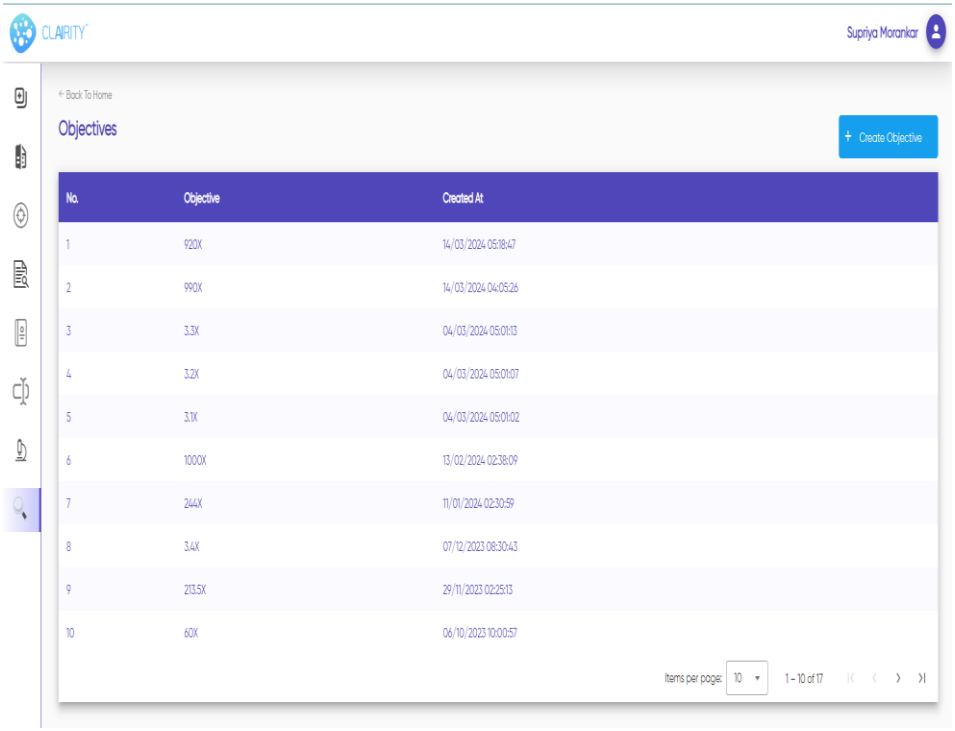
User can have access according to there subscription so they can use that module accordingly.



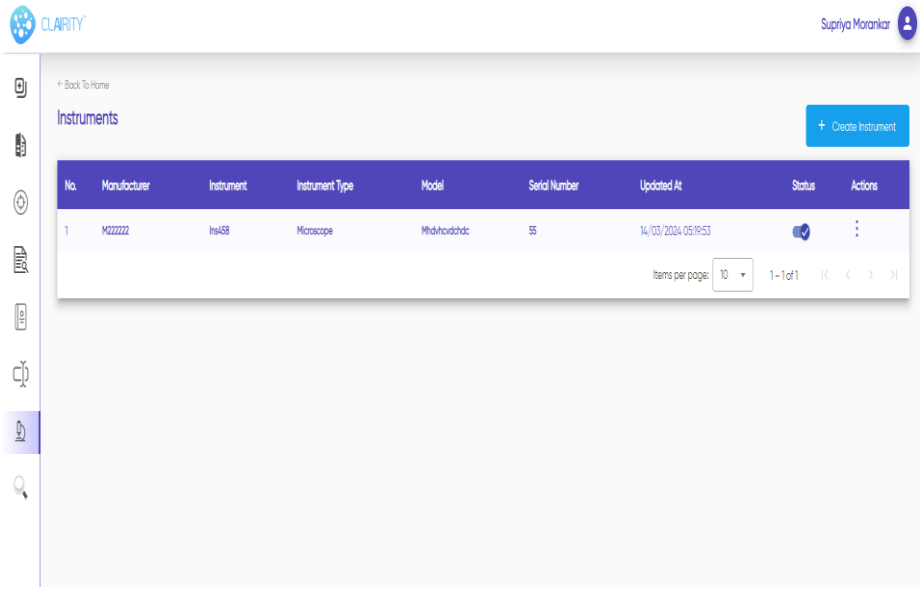
If user having access of particle module, then they can access accordingly.



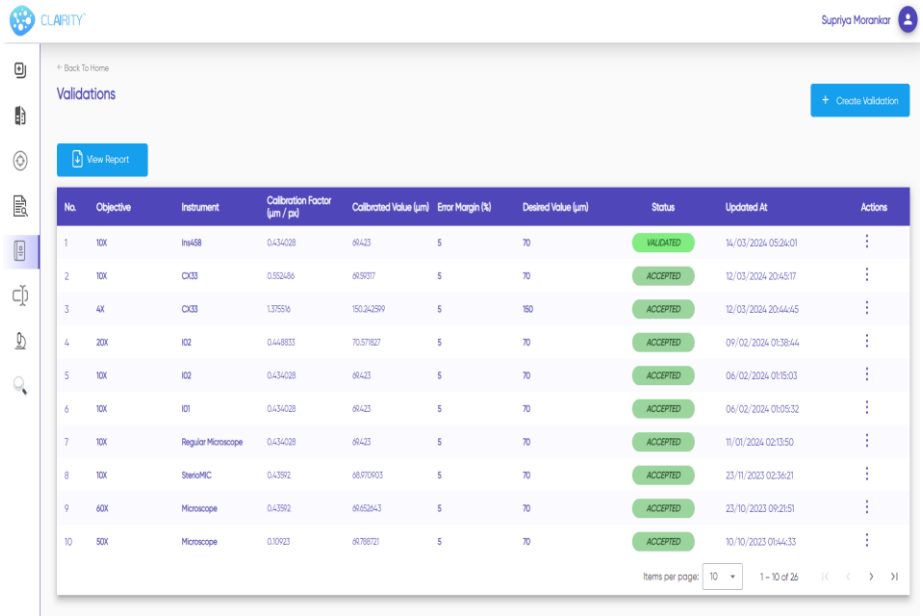
For creating analysis firstly create objective



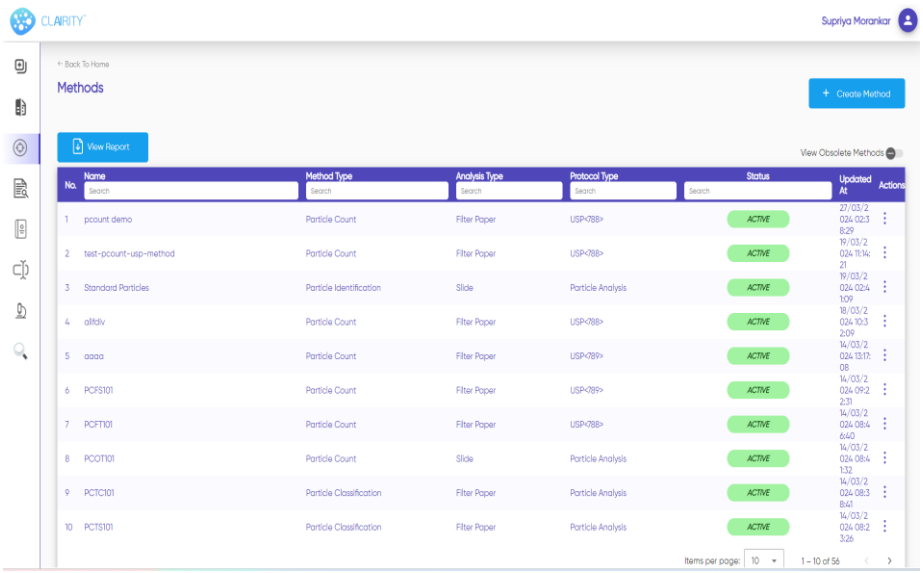
After creating objective create instrument and use that instrument for calibration



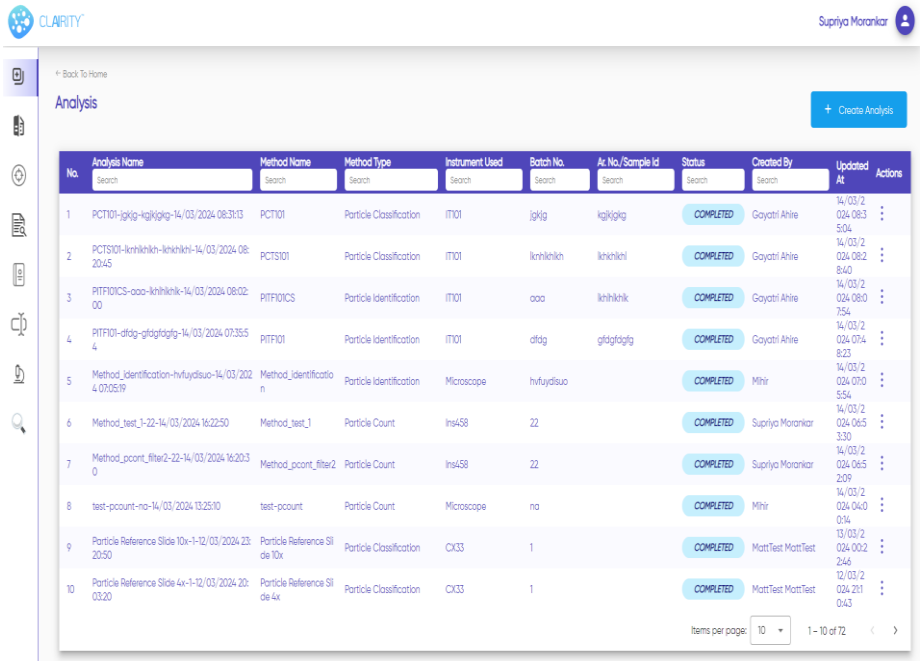
After creating calibration user select calibration and instrument for creating validation (instrument validation, instrument valid up to).



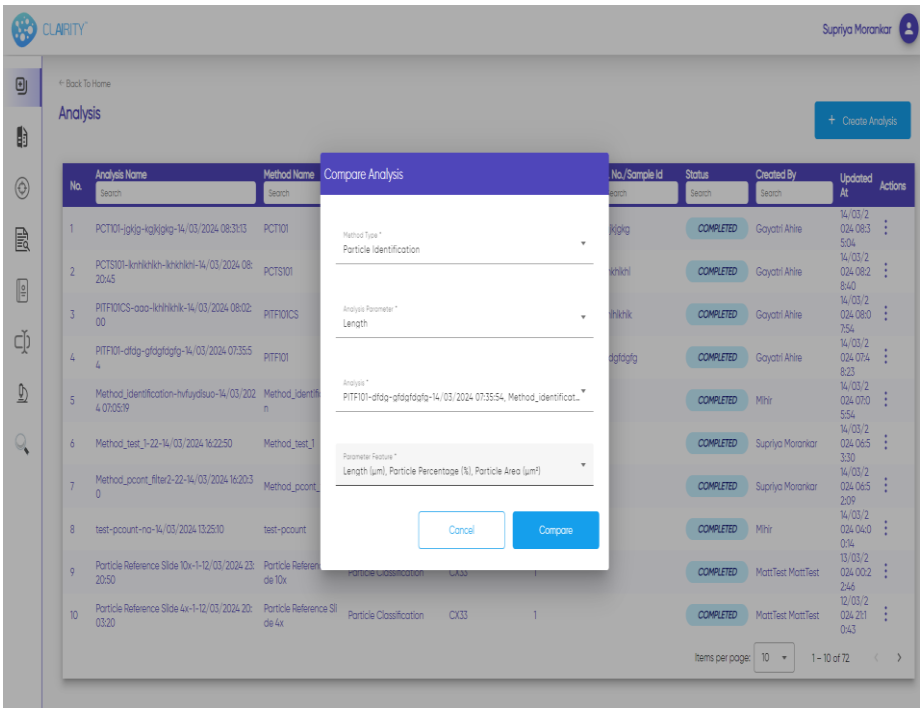
After creating validation user select objective for creating method (there are three types of method according to use user can select it particle count, identification and classification)



Creating analysis



User can also compare analysis.



After that user can logout also.