Water-Bacteria Project

Group 1

Alm Robert, Lavdim Imeri, Singh Vipin

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 30/03/2022 | 0.1 | Initiating the project artefact. Wrote a draft of Vision and Introduction. | Alm Robert, Lavdim Imeri, Singh Vipin |
| 11/04/2022 | 0.2 | Updating Introduction. Initiating Requirements, Risks and Tests. | Alm Robert, Lavdim Imeri, Singh Vipin |
| 20/04/2022 | 0.3 | Updating Introduction, Requirements, Risks and Tests. Fixing the tables. | Alm Robert, Lavdim Imeri, Singh Vipin |
|  |  |  |  |

Contents

[Revision History 1](#_Toc101053604)

[Vision 5](#_Toc101053605)

[Introduction 6](#_Toc101053606)

[Flow cytometry 6](#_Toc101053607)

[Process of flow cytometry 6](#_Toc101053608)

[Requirements 8](#_Toc101053609)

[Requirement 1. User registration: Register new user to the system. 8](#_Toc101053610)

[Requirement 2. User login: Log the user into the system. 9](#_Toc101053611)

[Requirement 3. Admin login: Log the admin into the system. 9](#_Toc101053612)

[Requirement 4. Upload FCS files in batch mode. 9](#_Toc101053613)

[Requirement 5. Visualize the pre-processed results through web application. 9](#_Toc101053614)

[Requirement 6. Filter out the noise in the FCS files. 9](#_Toc101053615)

[Requirement 7. Store pre-processed data in cloud database. 9](#_Toc101053616)

[Requirement 8. Analyzing the fcs files at the server and parse .FCS to .CSV. 10](#_Toc101053617)

[Requirement 9. Clustering the data by using different clustering methods. 10](#_Toc101053618)

[Requirement 10. Providing a written report of the session. 10](#_Toc101053619)

[Requirement 11. Full CRUD on Users. 10](#_Toc101053620)

[Requirement 12. Full CRUD on FCS files. 10](#_Toc101053621)

[Requirement 13. Full CRUD on Users. 10](#_Toc101053622)

[Supplementary Requirements 11](#_Toc101053623)

[Supplementary Requirement 1. Availability 11](#_Toc101053624)

[Supplementary Requirement 2. Robustness 11](#_Toc101053625)

[Supplementary Requirement 3. Graciousness 11](#_Toc101053626)

[Supplementary Requirement 4. Scalability 11](#_Toc101053627)

[Supplementary Requirement 5. User Friendliness 11](#_Toc101053628)

[Supplementary Requirement 6. Efficiency 11](#_Toc101053629)

[Risks 12](#_Toc101053630)

[Risk 1. Excessive Workload 12](#_Toc101053631)

[Risk 2. Compromised Personal Productivity 12](#_Toc101053632)

[Risk 3. Git Related Issues 12](#_Toc101053633)

[Risk 4. New Technologies 13](#_Toc101053634)

[Risk 5. External Factors 13](#_Toc101053635)

[Risk 6. Third Party Services 13](#_Toc101053636)

[Risk 7. Bugs and Vulnerabilities 13](#_Toc101053637)

[Risk 8. Compromised Performance 13](#_Toc101053638)

[Design 15](#_Toc101053639)

[The basic design idea 15](#_Toc101053640)

[Design Item 1. Front-End 15](#_Toc101053641)

[Design Item 2. Back-End 15](#_Toc101053642)

[Design Item 3. File Storage 16](#_Toc101053643)

[Design Item 4. Database 16](#_Toc101053644)

[Features 16](#_Toc101053645)

[Feature 1. User Registration 17](#_Toc101053646)

[Feature 2. User Login 17](#_Toc101053647)

[Feature 3. Admin Login 18](#_Toc101053648)

[Feature 4. Upload FCS files, (in batch mode) 18](#_Toc101053649)

[Feature 5. FCA Result Visualization 18](#_Toc101053650)

[Feature 6. Noise Filtering from the FCS data 18](#_Toc101053651)

[Feature 7. Store preprocessed Data to Cloud Database 18](#_Toc101053652)

[Feature 8. FCS file analysis 18](#_Toc101053653)

[Feature 9. FCS to CSV parsing 19](#_Toc101053654)

[Feature 10. Clustering of the preprocessed data 19](#_Toc101053655)

[Feature 11. Generating a report object 19](#_Toc101053656)

[Feature 12. Create User 19](#_Toc101053657)

[Feature 13. Read, (Display), the Users. 19](#_Toc101053658)

[Feature 14. Update, (Edit), a User. 20](#_Toc101053659)

[Feature 15. Delete a User. 20](#_Toc101053660)

[Feature 16. Read, (Display), The FCS files. 20](#_Toc101053661)

[Feature 17. Rename a FCS file. 20](#_Toc101053662)

[Feature 18. Delete an FCS file. 20](#_Toc101053663)

[Feature 19. Display the Reports. 20](#_Toc101053664)

[Feature 20. Rename a Report. 21](#_Toc101053665)

[Feature 21. Delete a Report. 21](#_Toc101053666)

[Feature 22. Display the details of a User. 21](#_Toc101053667)

[Feature 23. Display the contents of an FCS file. 21](#_Toc101053668)

[Feature 24. Display the contents of a Report. 21](#_Toc101053669)

[Tests 22](#_Toc101053670)

[Test 1. User Registration 23](#_Toc101053671)

[Test 2. User Login 23](#_Toc101053672)

[Test 3. Admin Login 23](#_Toc101053673)

[Test 4. Upload FCS files, (in batch mode) 23](#_Toc101053674)

[Test 5. FCA Result Visualization 24](#_Toc101053675)

[Test 6. Noise Filtering from the FCS data 24](#_Toc101053676)

[Test 7. Store preprocessed Data to Cloud Database 24](#_Toc101053677)

[Test 8. FCS file analysis 24](#_Toc101053678)

[Test 9. FCS to CSV parsing 24](#_Toc101053679)

[Test 10. Clustering of the preprocessed data 25](#_Toc101053680)

[Test 11. Generating a report object 25](#_Toc101053681)

[Test 12. Create User 25](#_Toc101053682)

[Test 13. Read, (Display), the Users 25](#_Toc101053683)

[Test 14. Update, (Edit), a User. 25](#_Toc101053684)

[Test 15. Delete a User. 26](#_Toc101053685)

[Test 16. Read, (Display), The FCS files. 26](#_Toc101053686)

[Test 17. Rename a FCS file. 26](#_Toc101053687)

[Test 18. Delete an FCS file. 26](#_Toc101053688)

[Test 19. Display the Reports 26](#_Toc101053689)

[Test 20. Rename a Report. 26](#_Toc101053690)

[Test 21. Delete a Report. 27](#_Toc101053691)

[Feature 22. Display the details of a User. 27](#_Toc101053692)

[Test 23. Display the contents of an FCS file. 27](#_Toc101053693)

[Test 24. Display the contents of a Report. 27](#_Toc101053694)

[Evaluation or Analysis of Test Results 27](#_Toc101053695)

[Conclusion 28](#_Toc101053696)

[References 29](#_Toc101053697)

# Vision

*(Write shortly what this document includes – as an abstract)*

Our vision is to offer a cheap, automatic, safe, and ongoing improved method of analyzing water, using flow cytometry, producing in that way an output that can be used to draw useful conclusions about the safeness or the purity of the water and if it is possible to be used for further research, (other methods of analysis).

As a deeper goal we aim to the possibility to isolate characteristics that can indicate things like hazardless of a bacteria or clear indicators if the water is contaminated or not, to be able to analyze the water faster and more efficiently with a lower computational cost.

# Introduction

Water is one of the most essential elements for the preservation of life, the survival of humankind and the development of human civilization. In modern era, while water still can’t be considered yet as a something granted for everyone, human civilization is at ongoing effort to make access to water safer, easier, and cheaper. One way to make water safer and cheaper, improving that way the access to water, is by improving the method of analyzing the safeness of the water.

That is a goal of great importance as the access to clean water is one of the major goals of the 2030 Agenda and its 17 Sustainable Development Goals, (SDGs), with SDG 16 being called “Clean Water and Sanitation” and at least other 4 of the 17 SDGs being affected by the availability and access to clean water. For example, the access to clean water is essential to the production of food, thing that calls for SDG 2, (Zero Hunger), and it is an integrated part for the goal to build sustainable cities, a matter that is addressed by SDG 11, (Sustainable Cities and Communities).

A solution to the challenge of analyzing water efficiently comes from a process that is called “Flow Cytometry” and it is a process that contains of a water sample are beamed with a laser beam in order to take measurements from the illuminance and the color that several particles emit due to their fluorescent agents. This methodology was used in the past for the analysis of blood samples but now is introduced to the field of water analysis.

## Flow cytometry

Flow cytometry is a technique for detecting and quantifying the physical and chemical properties of a population of cells or particles. A sample, including particles, is suspended in a fluid and injected into a flow cytometer equipment during this procedure. The flow cytometry method is used for identifying to assessing peripheral blood, bone marrow, and other bodily fluids in solution.

A flow cytometer is comparable to a microscope, except instead of creating a picture of the cell, it provides high-throughput, automated quantification of specified optical characteristics on a cell-by-cell basis. A single-cell suspension must first be created before solid tissue analysis can begin.

## Process of flow cytometry

A cell or particle sample is suspended in a fluid and injected into a flow cytometer equipment.

The sample is concentrated such that it flows through a laser beam one cell at a time, with the light dispersed being unique to the cells and their components.

Fluorescent markers are frequently used to label cells, causing light to be absorbed and subsequently released in a spectrum of wavelengths.

Thousands of cells may be analyzed fast, and the information acquired is processed by a computer.

Diagram

Description automatically generated

A flow cytometer is comprised of five key aspects:

1. Flow cell
2. Measurement system
3. Detector
4. Amplification system,
5. Computer for signal processing.

A liquid stream (sheath fluid) runs through the flow cell, carrying and aligning the cells such that they pass a single file through the light beam for sensing.

Measurement of impedance (or conductivity) and optical systems are often used in the measuring system. - mercury and xenon lamps; high-power water-cooled lasers (argon, krypton, dye laser); low-power air-cooled lasers (argon (488 nm), red-HeNe (633 nm), green-HeNe, HeCd (UV)); diode lasers (blue, green, red, violet).

Analog measurements of forward-scattered light (FSC) and side-scattered light (SSC), as well as dye-specific fluorescence signals, are converted into digital signals by the detector and analogue-to-digital conversion (ADC) system. A linear or logarithmic amplification mechanism can be used.

The word "acquisition" refers to gathering data from samples using a flow cytometer. A computer physically linked to the flow cytometer and software that handles the digital interface with the cytometer is used to facilitate the acquisition.

The program may alter settings for the sample being tested (e.g., voltage, compensation) and show initial sample information while obtaining data to check that parameters are set appropriately. Flow cytometers were first considered experimental equipment, but technological advancements have allowed them to be widely used for therapeutic and scientific objectives.

As a result of these discoveries, a sizable industry for instruments, analytic software, and acquisition reagents such as fluorescently tagged antibodies has emerged.

# Requirements

|  |  |
| --- | --- |
| **Requirement items** | **Priority** |
| **R1. User registration: Register new user to the system.** | **Essential** |
| **R2. User login: Log the user into the system.** | **Essential** |
| **R3. Admin login: Log the admin into the system.** | **Essential** |
| **R4. Upload FCS files in batch mode.** | **Essential** |
| **R5. Visualize the pre-processed results through web application.** | **Desirable** |
| **R6. Filter out the noise in the FCS files.** | **Essential** |
| **R7. Store pre-processed data in cloud database.** | **Essential** |
| **R8. Analysing the fcs files at the server and parse .FCS**  **to .CSV.** | **Essential** |
| **R9. Clustering the data by using different clustering methods.** | **Essential** |
| **R10. Providing a Report of the session.** | **Desirable** |
| **R11. Full CRUD on Users, (and Admins)** | **Essential** |
| **R12. Full CRUD on FCS files** | **Essential** |
| **R13. Full CRUD on Reports** | **Desirable** |

## Requirement 1. User registration: Register new user to the system.

**Description**: A new User shall be able to register to the system. Logically they will be 2 different types of registration. The one that is made by the users themselves for themselves, and a secondary that is applicable from the panel of the admin, (as a “Create a User” action). When the registration happens through the admin panel, the admin can give the Admin property to the user with the privileges that an Admin has. This Requirement is described by Requirement 11 as well.

## Requirement 2. User login: Log the user into the system.

**Description:** The User shall be able to log in to the system. It is the act when the user, (or the admin), provides the correct credentials to the system and gains access to the system. The login credentials are expected to be the email and the password that the user has defined for their account.

## Requirement 3. Admin login: Log the admin into the system.

**Description:** The Admin shall be able to log in to the system. It is the act when the admin, (admins are users with more privileges), provides the correct credentials to the system and gains access to the system. The login credentials are expected to be the email and the password that the user has defined for their account.

## Requirement 4. Upload FCS files in batch mode.

**Description:** FCS files shall be uploaded from the user interface to a Google Storage Service. In this process the user chooses the files that needs to be uploaded. While uploading, a status bar will show the progress of upload for each file.

## Requirement 5. Visualize the pre-processed results through web application.

**Description:** The User should be able to choose the file that needs to be shown. A scatter plot will show the cells in the sample using two values FL1-A, Fl2-A as the two coordinates.

## Requirement 6. Filter out the noise in the FCS files.

**Description:** The FCS files’ data shall be filtered to clear possible noise from them. The uploaded by the user data is expected to contain unnecessary information and impurities that can affect the results of the analysis. A user defined gating process will be used define the useful data and omit the data that does not contribute to the analysis.

## Requirement 7. Store pre-processed data in cloud database.

**Description:** The pre-processed data shall be stored to a cloud database. The collected and filtered data, is useful to be stored into a database so it will be available for processing or analysis. Understandably the data on the database needs to be mapped with the corresponding FCS file.

## Requirement 8. Analyzing the fcs files at the server and parse .FCS to .CSV.

**Description:** It is necessary for our system and for any further analysis to have a conversion from FCS to CSV especially when the data is processed and analyzed properly.

## Requirement 9. Clustering the data by using different clustering methods.

**Description:** Clustering is an important analysis tool that our system wish to utilize. To make the system more effective, the use should have the option to select between different clustering algorithms.

## Requirement 10. Providing a written report of the session.

**Description:** The user should have the analyzed data in a human readable report, so the system is should be able to deliver a structured documentation of the analysis and its results.

## Requirement 11. Full CRUD on Users.

**Description:** The system shall be able to Create, Read, Update and Delete the Users and the Admins of the System, through the admin panel.

## Requirement 12. Full CRUD on FCS files.

**Description:** The system shall be able to Create, Read, Update and Delete the FCS files of the System, through the user dashboard and the admin panel.

## Requirement 13. Full CRUD on Users.

**Description:** The system should be able to Create, Read, Update and Delete the reports of the System, through the user dashboard and the admin panel.

# Supplementary Requirements

|  |  |
| --- | --- |
| **Supplementary requirements** | **Priority** |
| SR1. Availability | **Essential** |
| SR2. Robustness | **Essential** |
| SR3. Graciousness | **Desirable** |
| SR4. Scalability | **Essential** |
| SR5. User Friendliness | **Desirable** |
| SR6. Efficiency | **Essential** |

## Supplementary Requirement 1. Availability

**Description:** The system shall be available to the users 24/7.

## Supplementary Requirement 2. Robustness

**Description:** The code shall be well written and documented with as few bugs or errors as possible.

## Supplementary Requirement 3. Graciousness

**Description:** The system should have a smooth progress without sudden activity spikes or unexplained behavior.

## Supplementary Requirement 4. Scalability

**Description:** The system shall be able to handle large amounts of workload.

## Supplementary Requirement 5. User Friendliness

**Description:** The user interface should be intuitive and easy to use.

## Supplementary Requirement 6. Efficiency

**Description:** The system shall perform as efficiently as possible and to consume as less resources as possible.

# Risks

|  |  |
| --- | --- |
| **Risk items** | **Priority** |
| **R1. Excessive Workload** | **High** |
| **R2. Compromised Personal Productivity** | **High** |
| **R3. Git Related Issues** | **High** |
| **R4. New Technologies** | **High** |
| **R5. External Factors** | **High** |
| **R6. Third Party Services** | **High** |
| **R7. Bugs and Vulnerabilities** | **High** |
| **R8. Compromised Performance** | **High** |

## Risk 1. Excessive Workload

Description: Even if we are confident that we can proceed to the completion of the project without difficulties, it is clear to us that it is a complicated project that requires high standards of quality, and a lot of effort. The possibility of assign to ourselves too many tasks and an amount of workload that we can't deliver, it is a factor that we should be very aware of.

Mitigation Strategy: we need to be aware about the size of workload, and how long time it will take us to complete each task

## Risk 2. Compromised Personal Productivity

Description: We are really confident that everyone in the team will do their very best while working on the project, but we should always be aware of the unforeseen factors that can affect someone's productivity, like sickness, personal life problems, exhaustion, external factors, etc.

Mitigation Strategy: Each task should involve the three of us, with one of us doing the task the two serving as consultants. That tactic increases our knowledge pool, and ensures that if someone of us is unable to complete the task, someone else will jump and take over from the spot the other person stopped.

## Risk 3. Git Related Issues

Description: Git is an important part of our programming culture, and we are highly trained on the use of Git/GitHub, but factors like, external promoted strategies, third party integrations, and the typical "human mistake" factor can lead to conflicts, that are not going to compromise our project but they are capable to delay us, sometimes even for days.

Mitigation Strategy: Ownership of our own strategy, even if our strategy will have to conform to other strategies. We have the be the rulers of our repositories at all times.

## Risk 4. New Technologies

Description: New technologies is a common point of failure, as we are not fully educated them about them, we are not fully aware of all the issues that may be hidden behind the new technology, and of course there is a whole set of unforeseen factor behind each new technology.

Mitigation Strategy: Avoid new technologies. Study extensively the documentation behind each new technology that we tend to use.

## Risk 5. External Factors

Description: From Dramatic global events, like pandemics or wars to more subtle events like incosistencies on the past research or change of the requirements, can affect our project in various ways.

Mitigation Strategy: A good documentation and good cooperation between the team members can assure that whatever happens the team will be consistent towards the completion of the project.

## Risk 6. Third Party Services

Description: Depedency on third party products or services is always something negative because of the depedency to their contribution. If a third party service change or cease to exist, it is expected that they will not take into consideration the needs of our project.

Mitigation Strategy: Avoidance of third party services if possible. Usage of the most reliable third part services. Study throughly the documentation of the use third party services, (or products).

## Risk 7. Bugs and Vulnerabilities

Description: Software Bugs can compromise the performance of a system dramatically or even damage the system and its reliability. System vulnerabilities, when not detected or fixed can compromize the security of the system.

Mitigation Strategy: Extensive Testing, andlysis of the results, patching, and of course peer review.

## Risk 8. Compromised Performance

Description: There are many reasons why a performance of a system is compromized. Sometimes it can be a software bug, a badly designed component of the system, faulty hardware, or simply inadquate design.

Mitigation Strategy: The understanding of the reasons why systems perform badly sometimes can help us having ways to mitigate the issue. Access to several experts and various technical support services can mitigate this risk drastically. Algorithmic analysis and data structures need to be taken into serious consideration as they are usually the number one reason for bad perfomance.

# Design

|  |  |
| --- | --- |
| **Design items** | **Priority** |
| **D1. Front-End** | **Essential** |
| **D2. Back-End** | **Essential** |
| **D3. File Storage** | **Essential** |
| **D4. Database** | **Essential** |

Diagram

Description automatically generated

## The basic design idea

Text

## Design Item 1. Front-End

Description: It is clear that the front end will be On React.js Instance, and a special attention will be given to the responsive design, possibly with the use of responsive design.

## Design Item 2. Back-End

Description: Back End is expected to be A Python instance with utilization of the Django Framework, in order to be able to serve as a standalone backend server with the ability to receive, process and return large amounts of data in different forms.

It is expected that the Python instance will be formatted inside a GitHub repository, so it will be able to be cloned inside a VM, (Linux probably), and with the support of Server Software that is not specified yet.

## Design Item 3. File Storage

Description: While, for the file storage, we will not have to implement a special instance, The google cloud storage instance has to be set up correctly, and 2 different interfaces, (backend and frontend), will have to have specific functionality implemented to be able to communicate and interact with the google cloud storage image.

## Design Item 4. Database

Description: Database has to have 2 different functionalities, namely the storage of the processed data, and the storage of user credentials. Because the different functionalities have different requirements for reliability and performance, the use of two different databases may be recommended. The Database component need to have interface on both front- and backend.

# Features

|  |  |  |
| --- | --- | --- |
| Features | Parent Requirement | Implementation |
| F1. User Registration | R1, R11 | **Not Implemented** |
| F2. User Login | R2 | **Not Implemented** |
| F3. Admin Login | R3 | **Not Implemented** |
| F4. Upload FCS files, (in batch mode) | R4, R12 | **Not Implemented** |
| F5. FCA Result Visualization | R5 | **Not Implemented** |
| F6. Noise Filtering from the FCS data | R6 | **Not Implemented** |
| F7. Store preprocessed Data to Cloud Database | R7 | **Not Implemented** |
| F8. FCS file analysis | R8 | **Not Implemented** |
| F9. FCS to CSV parsing | R8 | **Not Implemented** |
| F10. Clustering of the preprocessed data | R9 | **Not Implemented** |
| F11. Generating a report object | R10, R13 | **Not Implemented** |
| F12. Create User | R1, R11 | **Not Implemented** |
| F13. Read, (Display), the Users | R11 | **Not Implemented** |
| F14. Update, (Edit), a User. | R11 | **Not Implemented** |
| F15. Delete a User. | R11 | **Not Implemented** |
| F16. Read, (Display), The FCS files. | R12 | **Not Implemented** |
| F17. Rename a FCS file. | R12 | **Not Implemented** |
| F18. Delete an FCS file. | R12 | **Not Implemented** |
| F19. Display the Reports | R13 | **Not Implemented** |
| F20. Rename a Report. | R13 | **Not Implemented** |
| F21. Delete a Report. | R13 | **Not Implemented** |
| F22. Display the details of a User. | R11 | **Not Implemented** |
| F23. Display the contents of an FCS file. | R12 | **Not Implemented** |
| F24. Display the contents of a Report. | R13 | **Not Implemented** |

### Feature 1. User Registration

**Description:** The User shall be able to perform a registration to the system from his end.

**Requirement**: R1, R11

**Corresponded Test**: T1

## Feature 2. User Login

**Description**: The User shall be able to log in to the system by providing the necessary credentials.

**Requirement**: R2

**Corresponded** Test: T2

## Feature 3. Admin Login

**Description**: The Admin shall be able to log in to the system by providing the necessary credentials.

**Requirement**: R3

**Corresponded Test**: T3

## Feature 4. Upload FCS files, (in batch mode)

**Description**: FCS files shall be uploaded from the user interface to a Google Storage Service.

**Requirement**: R4, R12

**Corresponded Test**: T4

## Feature 5. FCA Result Visualization

**Description**: The User should be able to choose the file that needs to be shown. A scatter plot will show the cells in the sample using two values FL1-A, Fl2-A as the two coordinates.

**Requirement**: R5

**Corresponded Test**: T5

## Feature 6. Noise Filtering from the FCS data

**Description**: The FCS files’ data shall be filtered to clear possible noise from them. The uploaded by the user data is expected to contain unnecessary information and impurities that can affect the results of the analysis. A user defined gating process will be used define the useful data and omit the data that does not contribute to the analysis.

**Requirement**: R6

**Corresponded Test**: T6

## Feature 7. Store preprocessed Data to Cloud Database

**Description**: The pre-processed data shall be stored to a cloud database. The collected and filtered data, is useful to be stored into a database so it will be available for processing or analysis. Understandably the data on the database needs to be mapped with the corresponding FCS file.

**Requirement**: R7

**Corresponded Test**: T7

## Feature 8. FCS file analysis

**Description**: Analyzing the fcs files at the server

**Requirement**: R8

**Corresponded Test**: T8

## Feature 9. FCS to CSV parsing

**Description**: Parsing the analyzed data from the FCS files to CSV file format.

**Requirement**: R8

**Corresponded Test**: T9

## Feature 10. Clustering of the preprocessed data

**Description**: Clustering is an important analysis tool that our system wishes to utilize. To make the system more effective, the use should have the option to select between different clustering algorithms.

**Requirement**: R9

**Corresponded Test**: T10

## Feature 11. Generating a report object

**Description**: A report object contain the results of the analysis need to be generated.

**Requirement**: R10, R13

**Corresponded Test**: T11

## Feature 12. Create User

**Description:** The system shall be able to Create, the Users and the Admins of the System, through the admin panel or the registration feature.

**Requirement**: R1, R12

**Corresponded Test**: T12

## Feature 13. Read, (Display), the Users.

**Description**: The system shall be able to display the Users and the Admins of the System, through the admin panel.

**Requirement**: R11

**Corresponded Test**: T13

## Feature 14. Update, (Edit), a User.

**Description**: The system shall be able to edit, (Update), the Users and the Admins of the System, through the admin panel.

**Requirement**: R11

**Corresponded Test**: T14

## Feature 15. Delete a User.

Description: The system shall be able to Delete the Users and the Admins of the System, through the admin panel.

**Requirement**: R11

**Corresponded Test**: T15

## Feature 16. Read, (Display), The FCS files.

Description: The system shall be able to display, (Read), the FCS files of the System, through the user dashboard and the admin panel.

**Requirement**: R12

**Corresponded Test**: T16

## Feature 17. Rename a FCS file.

**Description:** The system shall be able to rename, (Update), the FCS files of the System, through the user dashboard and the admin panel.

**Requirement**: R12

**Corresponded Test**: T17

## Feature 18. Delete an FCS file.

**Description**: The system shall be able to Delete the FCS files of the System, through the user dashboard and the admin panel.

**Requirement**: R12

**Corresponded Test**: T18

## Feature 19. Display the Reports.

**Description**: The system should be able to display, (Read), the reports of the System, through the user dashboard and the admin panel.

**Requirement**: R13

**Corresponded Test**: T19

## Feature 20. Rename a Report.

**Description**: The system should be able to rename, (Update), the reports of the System, through the user dashboard and the admin panel.

**Requirement**: R13

**Corresponded Test**: T20

## Feature 21. Delete a Report.

**Description**: The system should be able to Delete the reports of the System, through the user dashboard and the admin panel.

**Requirement**: R13

**Corresponded Test**: T21

## Feature 22. Display the details of a User.

Description: The system shall be able to display the details of a User through the Admin panel, or the User’s account page.

Requirement: R11

**Corresponded Test**: T22

## Feature 23. Display the contents of an FCS file.

Description: The system shall be able to display the details of an FCS file through the Admin panel, or the User’s dashboard.

**Requirement**: R12

**Corresponded Test**: T23

## Feature 24. Display the contents of a Report.

Description: The system should be able to display the details of a report file through the User’s dashboard, (or the admin panel).

**Requirement**: R13

**Corresponded Test**: T25

# Tests

|  |  |  |  |
| --- | --- | --- | --- |
| Tests | Parent Feature | Parent Requirement | Passed/Failed |
| T1. User Registration | F1 | R1, R11 | **Not Implemented** |
| T2. User Login | F2 | R2 | **Not Implemented** |
| T3. Admin Login | F3 | R3 | **Not Implemented** |
| T4. Upload FCS files, (in batch mode) | F4 | R4, R12 | **Not Implemented** |
| T5. FCA Result Visualization | F5 | R5 | **Not Implemented** |
| T6. Noise Filtering from the FCS data | F6 | R6 | **Not Implemented** |
| T7. Store preprocessed Data to Cloud Database | F7 | R7 | **Not Implemented** |
| T8. FCS file analysis | F8 | R8 | **Not Implemented** |
| T9. FCS to CSV parsing | F9 | R8 | **Not Implemented** |
| T10. Clustering of the preprocessed data | F10 | R9 | **Not Implemented** |
| T11. Generating a report object | F11 | R10, R13 | **Not Implemented** |
| T12. Create User | F12 | R1, R11 | **Not Implemented** |
| T13. Read, (Display), the Users | F13 | R11 | **Not Implemented** |
| T14. Update, (Edit), a User. | F14 | R11 | **Not Implemented** |
| T15. Delete a User. | F15 | R11 | **Not Implemented** |
| T16. Read, (Display), The FCS files. | F16 | R12 | **Not Implemented** |
| T17. Rename a FCS file. | F17 | R12 | **Not Implemented** |
| T18. Delete an FCS file. | F18 | R12 | **Not Implemented** |
| T19. Display the Reports | F19 | R13 | **Not Implemented** |
| T20. Rename a Report. | F20 | R13 | **Not Implemented** |
| T21. Delete a Report. | F21 | R13 | **Not Implemented** |
| T22. Display the details of a User. | F22 | R11 | **Not Implemented** |
| T23. Display the contents of an FCS file. | F23 | R12 | **Not Implemented** |
| T24. Display the contents of a Report. | F24 | R13 | **Not Implemented** |

## Test 1. User Registration

**Requirement:** R1, R12

**Feature**: F1

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 2. User Login

**Requirement**: R2

**Feature:** F2

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 3. Admin Login

**Requirement**: R3

**Feature**: F3

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 4. Upload FCS files, (in batch mode)

**Requirement**: R4, R12

**Feature**: F4

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 5. FCA Result Visualization

**Requirement**: R5

**Feature**: F5

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 6. Noise Filtering from the FCS data

**Requirement**: R6

**Feature**: F6

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 7. Store preprocessed Data to Cloud Database

**Requirement**: R7

**Feature**: F7

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 8. FCS file analysis

**Requirement**: R8

**Feature**: F8

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 9. FCS to CSV parsing

**Requirement**: R8

**Feature**: F9

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 10. Clustering of the preprocessed data

**Requirement**: R9

**Feature**: F10

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 11. Generating a report object

**Requirement**: R10, R13

**Feature**: F11

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 12. Create User

**Requirement**: R1, R12

**Feature**: F12

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 13. Read, (Display), the Users

**Requirement**: R11

**Feature**: F13

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 14. Update, (Edit), a User.

**Requirement**: R11

**Feature**: F14

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 15. Delete a User.

**Requirement**: R11

**Feature**: F15

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 16. Read, (Display), The FCS files.

**Requirement**: R12

**Feature:** F16

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 17. Rename a FCS file.

**Requirement**: R12

**Feature**: F17

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 18. Delete an FCS file.

**Requirement**: R12

**Feature**: F18

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 19. Display the Reports

**Requirement**: R13

**Feature**: F19

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 20. Rename a Report.

**Requirement**: R13

**Feature**: F20

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 21. Delete a Report.

**Requirement**: R13

**Feature**: F21

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Feature 22. Display the details of a User.

**Requirement**: R11

**Feature**: F22

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 23. Display the contents of an FCS file.

**Requirement**: R12

**Feature**: F23

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

## Test 24. Display the contents of a Report.

**Requirement**: R13

**Feature**: F24

**Testing** Procedure: Not Defined Yet

**Expected Results**: Not Defined Yet

# Evaluation or Analysis of Test Results

# Conclusion

# References

*(Use IEEE reference system. It is in-built in Word.)*

(Books)

(Articles and Journals)

(WWW)