



---

# 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
01/04/2022	0.4	Updating Introduction, Requirements, Tests, and Design	Alm Robert, Lavdim Imeri, Singh Vipin



## Contents

Revision History.....	1
Vision.....	5
Introduction .....	6
Unsafe Water.....	6
What Are the Effects of Water Pollution?.....	7
In terms of human health.....	7
Concerning the environment.....	7
SDG 6 Water and sanitation for all.....	7
SDG 2 Zero Hunger .....	8
SDG 11 Sustainable Cities and Communities.....	8
Flow cytometry .....	8
Process of flow cytometry .....	8
Requirements .....	10
Requirement 1. User registration: Register new user to the system.....	10
Requirement 2. User login: Log the user into the system.....	11
Requirement 3. Admin login: Log the admin into the system. ....	11
Requirement 4. Upload FCS files in batch mode. ....	11
Requirement 5. Visualize the pre-processed results through web application. ....	11
Requirement 6. Filter out the noise in the FCS files.....	11
Requirement 7. Store pre-processed data in cloud database. ....	11
Requirement 8. Analyzing the fcs files at the server and parse .FCS to .CSV.....	12
Requirement 9. Clustering the data by using different clustering methods.....	12
Requirement 10. Providing a written report of the session.....	12
Requirement 11. Full CRUD on Users. ....	12
Requirement 12. Full CRUD on FCS files.....	12
Requirement 13. Full CRUD on Users. ....	12
Supplementary Requirements.....	13
Supplementary Requirement 1. Availability.....	13
Supplementary Requirement 2. Robustness .....	13
Supplementary Requirement 3. Graciousness.....	13
Supplementary Requirement 4. Scalability.....	13
Supplementary Requirement 5. User Friendliness.....	13
Supplementary Requirement 6. Efficiency.....	13
Risks .....	14

Risk 1. Excessive Workload.....	14
Risk 2. Compromised Personal Productivity .....	14
Risk 3. Git Related Issues .....	14
Risk 4. New Technologies .....	15
Risk 5. External Factors .....	15
Risk 6. Third Party Services .....	15
Risk 7. Bugs and Vulnerabilities.....	15
Risk 8. Compromised Performance .....	15
Design .....	17
The basic design idea.....	17
Design Item 1. Front-End.....	17
Design Item 2. Back-End .....	18
Design Item 3. File Storage.....	18
Design Item 4. Database .....	18
Features .....	18
Feature 1. User Registration .....	19
Feature 2. User Login.....	20
Feature 3. Admin Login.....	20
Feature 4. Upload FCS files, (in batch mode).....	20
Feature 5. FCA Result Visualization.....	20
Feature 6. Noise Filtering from the FCS data .....	20
Feature 7. Store preprocessed Data to Cloud Database .....	21
Feature 8. FCS file analysis.....	21
Feature 9. FCS to CSV parsing .....	21
Feature 10. Clustering of the preprocessed data .....	21
Feature 11. Generating a report object.....	21
Feature 12. Create User .....	21
Feature 13. Read, (Display), the Users. ....	22
Feature 14. Update, (Edit), a User. ....	22
Feature 15. Delete a User.....	22
Feature 16. Read, (Display), The FCS files. ....	22
Feature 17. Rename a FCS file.....	22
Feature 18. Delete an FCS file.....	23
Feature 19. Display the Reports.....	23
Feature 20. Rename a Report.....	23
Feature 21. Delete a Report. ....	23
Feature 22. Display the details of a User. ....	23
Feature 23. Display the contents of an FCS file.....	23

Feature 24. Display the contents of a Report.....	24
Tests .....	25
Test 1. User Registration.....	26
Test 2. User Login.....	26
Test 3. Admin Login.....	26
Test 4. Upload FCS files, (in batch mode).....	27
Test 5. FCA Result Visualization.....	27
Test 6. Noise Filtering from the FCS data .....	27
Test 7. Store preprocessed Data to Cloud Database .....	28
Test 8. FCS file analysis .....	28
Test 9. FCS to CSV parsing .....	28
Test 10. Clustering of the preprocessed data .....	28
Test 11. Generating a report object .....	28
Test 12. Create User .....	29
Test 13. Read, (Display), the Users .....	29
Test 14. Update, (Edit), a User. ....	29
Test 15. Delete a User. ....	29
Test 16. Read, (Display), The FCS files. ....	29
Test 17. Rename a FCS file.....	30
Test 18. Delete an FCS file.....	30
Test 19. Display the Reports.....	30
Test 20. Rename a Report.....	30
Test 21. Delete a Report. ....	31
Feature 22. Display the details of a User. ....	31
Test 23. Display the contents of an FCS file.....	31
Test 24. Display the contents of a Report.....	31
Test 25. Handshake with a remote database, (firebase).....	31
Evaluation or Analysis of Test Results .....	32
Conclusion.....	33
References.....	34

## 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.

## Unsafe Water

Water is essential for economic growth, food and energy production, ecological sustainability, and even human survival. Water is also crucial to climate change adaptation as a vital link between people and the environment.

Freshwater is also a matter of rights. As the world's population rises, the need to balance competing economic demands on water resources develops, ensuring that communities have enough to meet their needs. Women and children, in particular, require clean, private sanitary facilities to handle menstruation and maternity in dignity and safety.

"Ensure availability and sustainable management of water and sanitation for all," says Sustainable Development Goal 6 (SDG 6). The goals involve every area of the water cycle and sanitation systems, and their success will help advance several other SDGs, including health, education, economics, and the environment. Contaminated water and a lack of basic sanitation are jeopardising efforts in the world's poorest countries to overcome severe poverty and disease.

In 2017, 2 billion people worldwide lacked access to basic sanitation like toilets and latrines. Six hundred seventy-three million individuals still practised open defecation. According to the WHO/UNICEF Joint Monitoring Programme for Water Supply and Sanitation, at least 1.2 billion people globally drink water that has not been treated to prevent faecal contamination. Even more drinkable water is distributed through a system that lacks proper sanitary protection.

A primary cause of child mortality is contaminated water and poor sanitation. Inadequate water supply, inadequate sanitation, water polluted with infectious disease agents, and poor hygiene habits are linked to childhood diarrhoea. Diarrhoea is projected to kill 1.5 million children each year, most of whom are under the age of five and live in underdeveloped nations.

When dangerous substances—often chemicals or microorganisms—contaminate a stream, river, lake, ocean, aquifer, or other body of water, the water quality deteriorates, and the water becomes toxic to humans or the environment.

Water is very susceptible to pollution. Water, also known as a "universal solvent," can dissolve more chemicals than any other liquid. We have Kool-Aid and vivid blue waterfalls because of it. It's also the reason why water is so easily contaminated. Toxic compounds from farms, cities, and factories quickly dissolve and combine with it, polluting the water.

## What Are the Effects of Water Pollution?

### In terms of human health

To put it clearly, contamination of water kills. According to research published in The Lancet in 2015, it was responsible for 1.8 million fatalities. Water that has been contaminated can also make you sick. Every year, nearly 1 billion people become ill due to contaminated water. Low-income groups are particularly vulnerable because their dwellings are frequently located near polluting enterprises.

### Concerning the environment

Healthy ecosystems rely on a complex web of animals, plants, microbes, and fungus to exist, interacting with one another, either directly or indirectly. Any harm to these organisms can set off a chain reaction that puts entire aquatic ecosystems at risk.

## SDG 6 Water and sanitation for all

SDG 6 is crucial for long-term development. Human rights include access to safe drinking water and adequate sanitation. The availability of these services, especially water and soap for handwashing, is critical to human health and well-being. They are essential for improving nutrition, preventing disease, facilitating health care, and ensuring the proper functioning of schools, workplaces, and political institutions, as well as women, girls, and marginalised groups' total involvement in society. SDG 6 is crucial for long-term development.

Human rights include access to safe drinking water and adequate sanitation. The availability of these services, especially water and soap for handwashing, is critical to human health and well-being. They're necessary for boosting nutrition, preventing disease, providing access to health care, and guaranteeing the smooth operation of schools, businesses, and political institutions.

However, SDG 6 encompasses the entire water cycle and water and sanitation services. Water is used for more than just residential purposes. It is required in all sections of society for food, energy, goods, and services to be produced. These activities generate wastewater as well. If not adequately treated, it has the potential to spread diseases and introduce extra nutrients and harmful substances into rivers, lakes, and other bodies of water oceans. In the end, ecosystems supply a substantial portion of society's water. For the ecosystems to function properly, water must remain within the ecosystem. They need to stay healthy. Ecosystems are in good shape. As a result, ensuring the quantity and quality of freshwater and overall resistance to climate change Changes brought on by humans and the environment.

## SDG 2 Zero Hunger

The food aid program of the World Food Programme provides a lifeline to 87 million disadvantaged individuals around the world. The right to food goes beyond productivism, which is the ideology that underpins Goal 2 (zero hunger). To realise this right, we must address the historical and structural inequalities that jeopardise food systems' availability, sufficiency, accessibility, and sustainability. Need has increased globally since 2015, affecting over 820 million people. Adult obesity rates have risen year after year, and only 5% of countries are on track to reach childhood obesity targets. Millions of people are affected by hidden hunger or micronutrient deficiencies, such as the 151 million children under five who experienced stunted growth in 2017.

## SDG 11 Sustainable Cities and Communities

Cities are expected to double in size by 2030, according to projections. Be home to 60% of the world's population, a proportion expected to rise to around. By 2050, 68.4 per cent will have been achieved. Between the years 2010 and 2050, It is believed that between 2.5 and 3 million people will be affected. The urban population will grow by a billion people. With the most incredible people in the world, Less developed countries are expected to grow faster. East Asia, South Asia, and Southeast Asia are examples of such regions. Sub-Saharan Africa, to name a few. From the Millennium Development Goals to the Sustainable Development Goals In the last several years, the international community has adopted the Sustainable Development Goals (SDGs). The previous two decades have seen the emergence of As a crucial development trend; urbanisation is becoming more prevalent workforce.

## 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

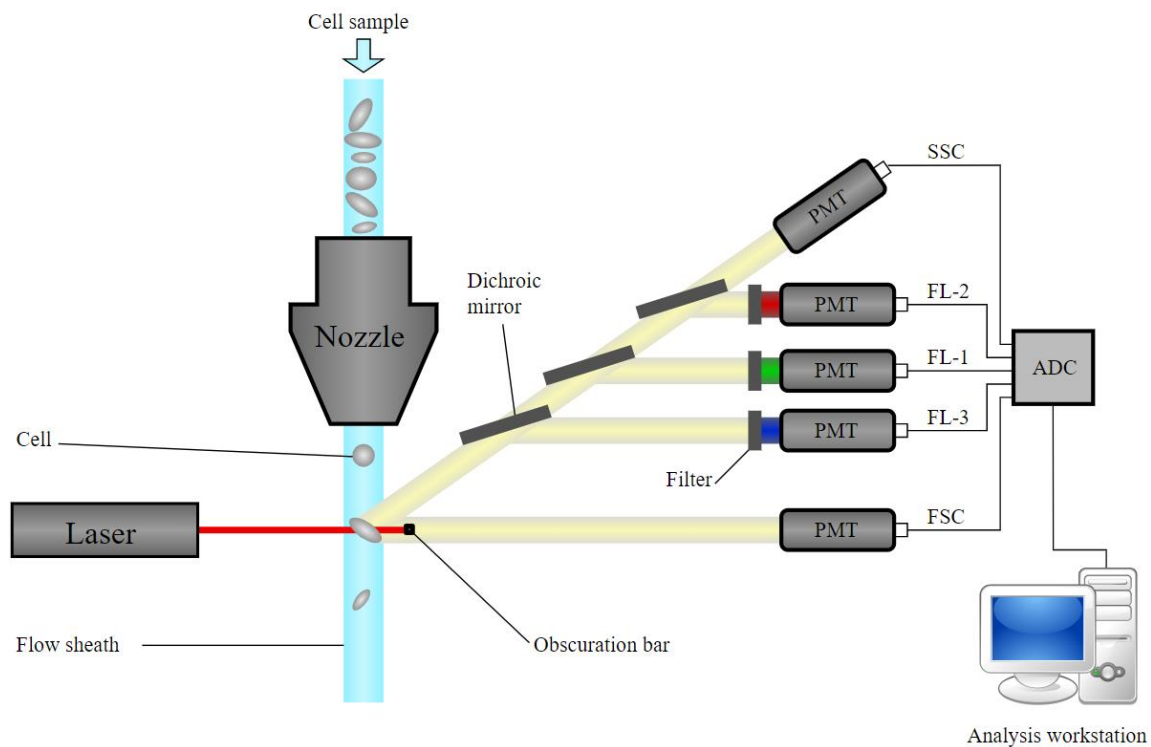
A cell or particle sample is suspended in a fluid and injected into a flow cytometer equipment.

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.





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

### 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
<b>R1. Excessive Workload</b>	<b>High</b>
<b>R2. Compromised Personal Productivity</b>	<b>High</b>
<b>R3. Git Related Issues</b>	<b>High</b>
<b>R4. New Technologies</b>	<b>High</b>
<b>R5. External Factors</b>	<b>High</b>
<b>R6. Third Party Services</b>	<b>High</b>
<b>R7. Bugs and Vulnerabilities</b>	<b>High</b>
<b>R8. Compromised Performance</b>	<b>High</b>

### 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 to 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 inconsistencies 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: Dependency on third party products or services is always something negative because of the dependency 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 party services. Study thoroughly 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 compromise the security of the system.

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

#### Risk 8. Compromised Performance

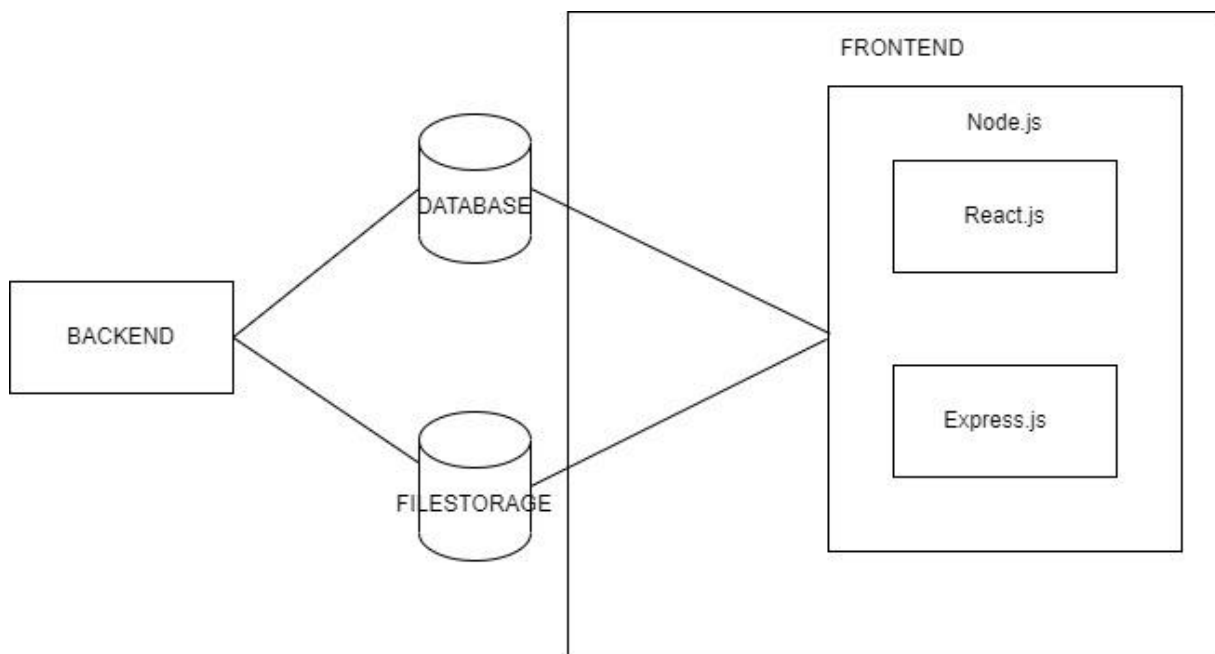
Description: There are many reasons why a performance of a system is compromised. Sometimes it can be a software bug, a badly designed component of the system, faulty hardware, or simply inadequate design.

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 performance.



## Design

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



### 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. For now it is clear that firebase will be used as a database, at least for the users and the administrative data, and probably we will extend the database to the entirety of the data, as firebase is a NoSQL and it is recommended for large amounts of data.

Front end follows a single page architecture that displays different panels in the same page, keeping in that way a robust engine and improved security. Extra React components weren't necessary yet, but they will probably be used in the future for reasons of advanced display.

## 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.

The Python part of the design consists of python server that utilizes sockets to handle http request. At the same time, effort is given to build a core module that will handle the machine learning and the heavy computational load of the system. Having the computational parts as a separated module, releases the computation from any web dependency and makes it possible to use the same functionality to other python instances.

## 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. At the moment the selected database for our needs is Google Firebase.

## Features

Features	Parent Requirement	Implementation
<b>F1. User Registration</b>	R1, R11	<b>Implemented</b>
<b>F2. User Login</b>	R2	<b>Implemented</b>
<b>F3. Admin Login</b>	R3	<b>Implemented</b>
<b>F4. Upload FCS files, (in batch mode)</b>	R4, R12	<b>Not Implemented</b>
<b>F5. FCA Result Visualization</b>	R5	<b>Not Implemented</b>

<b>F6. Noise Filtering from the FCS data</b>	R6	<b>Not Implemented</b>
<b>F7. Store preprocessed Data to Cloud Database</b>	R7	<b>Not Implemented</b>
<b>F8. FCS file analysis</b>	R8	<b>Not Implemented</b>
<b>F9. FCS to CSV parsing</b>	R8	<b>Not Implemented</b>
<b>F10. Clustering of the preprocessed data</b>	R9	<b>Not Implemented</b>
<b>F11. Generating a report object</b>	R10, R13	<b>Not Implemented</b>
<b>F12. Create User</b>	R1, R11	<b>Implemented</b>
<b>F13. Read, (Display), the Users</b>	R11	<b>Not Implemented</b>
<b>F14. Update, (Edit), a User.</b>	R11	<b>Not Implemented</b>
<b>F15. Delete a User.</b>	R11	<b>Not Implemented</b>
<b>F16. Read, (Display), The FCS files.</b>	R12	<b>Not Implemented</b>
<b>F17. Rename a FCS file.</b>	R12	<b>Not Implemented</b>
<b>F18. Delete an FCS file.</b>	R12	<b>Not Implemented</b>
<b>F19. Display the Reports</b>	R13	<b>Not Implemented</b>
<b>F20. Rename a Report.</b>	R13	<b>Not Implemented</b>
<b>F21. Delete a Report.</b>	R13	<b>Not Implemented</b>
<b>F22. Display the details of a User.</b>	R11	<b>Not Implemented</b>
<b>F23. Display the contents of an FCS file.</b>	R12	<b>Not Implemented</b>
<b>F24. Display the contents of a Report.</b>	R13	<b>Not Implemented</b>

### 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
<b>T1. User Registration</b>	F1	R1, R11	<b>Passed.</b>
<b>T2. User Login</b>	F2	R2	<b>Passed.</b>
<b>T3. Admin Login</b>	F3	R3	<b>Passed.</b>
<b>T4. Upload FCS files, (in batch mode)</b>	F4	R4, R12	<b>Not Tested Yet.</b>
<b>T5. FCA Result Visualization</b>	F5	R5	<b>Not Tested Yet.</b>
<b>T6. Noise Filtering from the FCS data</b>	F6	R6	<b>Not Tested Yet.</b>
<b>T7. Store preprocessed Data to Cloud Database</b>	F7	R7	<b>Not Tested Yet.</b>
<b>T8. FCS file analysis</b>	F8	R8	<b>Not Tested Yet.</b>
<b>T9. FCS to CSV parsing</b>	F9	R8	<b>Not Tested Yet.</b>
<b>T10. Clustering of the preprocessed data</b>	F10	R9	<b>Not Tested Yet.</b>
<b>T11. Generating a report object</b>	F11	R10, R13	<b>Not Tested Yet.</b>
<b>T12. Create User</b>	F12	R1, R11	<b>Passed.</b>
<b>T13. Read, (Display), the Users</b>	F13	R11	<b>Not Tested Yet.</b>
<b>T14. Update, (Edit), a User.</b>	F14	R11	<b>Not Tested Yet.</b>
<b>T15. Delete a User.</b>	F15	R11	<b>Not Tested Yet.</b>
<b>T16. Read, (Display), The FCS files.</b>	F16	R12	<b>Not Tested Yet.</b>
<b>T17. Rename a FCS file.</b>	F17	R12	<b>Not Tested Yet.</b>
<b>T18. Delete an FCS file.</b>	F18	R12	<b>Not Tested Yet.</b>
<b>T19. Display the Reports</b>	F19	R13	<b>Not Tested Yet.</b>

<b>T20. Rename a Report.</b>	F20	R13	<b>Not Tested Yet.</b>
<b>T21. Delete a Report.</b>	F21	R13	<b>Not Tested Yet.</b>
<b>T22. Display the details of a User.</b>	F22	R11	<b>Not Tested Yet.</b>
<b>T23. Display the contents of an FCS file.</b>	F23	R12	<b>Not Tested Yet.</b>
<b>T24. Display the contents of a Report.</b>	F24	R13	<b>Not Tested Yet.</b>
<b>T25. Handshake with a remote database, (firebase).</b>	F1, F2, F3, F7, F10, F11, F12, F14, F15, F19, F20, F21, F22, F24	R1, R2, R3, R7, R9, R10, R11, R13	<b>Passed.</b>

### Test 1. User Registration

**Requirement:** R1, R12

**Feature:** F1

**Testing Procedure:** It can be tested by a user through the graphical user interface or by script running on the frontend, either way is legit. The test is performed if the user or the script is capable to create a new account without being an admin, and without having direct access to the backend or the database.

**Expected Results:** True if the registration is successful of false if the registration is unsuccessful.

### Test 2. User Login

**Requirement:** R2

**Feature:** F2

**Testing Procedure:** It can be tested by a user through the graphical user interface or by script running on the frontend, either way is legit. The test is performed if the user or the script is capable to login. That means to give credentials that correspond to a specific user from the database and gain the status of "logged in", and to try one more time with false credentials in order to fail to acquire the status of "logged in". Testing script may be more successful as it can check more cases but through the user interface, it is more realistic.

**Expected Results:** True if the login is successful of false if the login is unsuccessful. At the same time true if a faulty login is unsuccessful, and false if a faulty login is successful.

### Test 3. Admin Login

**Requirement:** R3

**Feature:** F3

**Testing Procedure:** It can be tested by a user through the graphical user interface or by script running on the frontend, either way is legit. The test is performed if the user or the script is capable to login as an admin. That means to give credentials that correspond to a specific user with admin privileges from the database and gain the status of “logged in” and “admin”, and to try one more time with false credentials in order to fail to acquire the status of “logged in” or “admin”. Testing script may be more successful as it can check more cases but through the user interface, it is more realistic.

**Expected Results:** True if the login is successful or false if the login is unsuccessful. At the same time true if a faulty login is unsuccessful, and false if a faulty login is successful.

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

**Requirement:** R4, R12

**Feature:** F4

**Testing Procedure:** The test is performed if the user is able to select several files from his computer and upload them successfully to a specific remote storage location through the frontend of the system

**Expected Results:** True if the user is capable to send multiple selected files from his computer through the frontend of the system to specific file storage destination that is determined by the system.

#### Test 5. FCA Result Visualization

**Requirement:** R5

**Feature:** F5

**Testing Procedure:** The system takes the selected preprocessed data and generates a dotplot of 2 dimensions from it.

**Expected Results:** If a dotplot is generated that corresponds to the given data, then the result is true.

#### Test 6. Noise Filtering from the FCS data

**Requirement:** R6

**Feature:** F6

**Testing Procedure:** As a part of the processing. The data from the FCS files is filtered from any noise.

**Expected Results:** True if the noise is below the acceptable level.

## Test 7. Store preprocessed Data to Cloud Database

**Requirement:** R7

**Feature:** F7

**Testing Procedure:** The test controls if the system, (the backend), is capable to upload the selected preprocessed data to a specific database.

**Expected Results:** True if the data is uploaded successfully, false if the data fail to be uploaded.

## 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:** We test here if the system is capable to parse the data of and FCS file to a CSV format.

**Expected Results:** True if the system can parse data from FCS to CSV. False otherwise.

## Test 10. Clustering of the preprocessed data

**Requirement:** R9

**Feature:** F10

**Testing Procedure:** This test confirms the ability of the system to cluster the preprocessed data, by using at least on clustering mechanism.

**Expected Results:** True, if a successful clustering occurs.

## 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:** We test the ability of the system to create new user accounts. As the registration is tested at other point. This test will test the ability to create a new user account through the admin panel.

**Expected Results:** True if the system can create a new user account, through the admin panel.

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

**Requirement:** R11

**Feature:** F13

**Testing Procedure:** We test the ability of the system to display the registered user through the admin panel.

**Expected Results:** True if the system is capable to display the registered users through the admin panel.

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

**Requirement:** R11

**Feature:** F14

**Testing Procedure:** We test if the admin is capable to edit the details of a registered user. To pass this test the admin should be able to edit at least the majority of the user details, through the admin panel.

**Expected Results:** True if the admin is capable to change the details of a user through the admin panel.

### Test 15. Delete a User.

**Requirement:** R11

**Feature:** F15

**Testing Procedure:** We test if the admin has the option to delete a user.

**Expected Results:** Pass if the admin can delete a registered user. Fail otherwise.

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

**Requirement:** R12

**Feature:** F16

**Testing Procedure:** We test the ability of the system to display the titles of the FCS files that are stored on the remote storage location of the system.

**Expected Results:** Pass if the system successfully displays the list of the fcs files that are stored in its predefined storage location.

#### Test 17. Rename a FCS file.

**Requirement:** R12

**Feature:** F17

**Testing Procedure:** We test if the system can change the name of a stored fcs file, through the frontend.

**Expected Results:** Pass if the user can change the name of a stored FCS file through the frontend.

#### Test 18. Delete an FCS file.

**Requirement:** R12

**Feature:** F18

**Testing Procedure:** We test if the system is capable to delete a stored FCS file from the defined storage location.

**Expected Results:** Pass if the system is capable to delete an FCS file from the ones that are stored in the predefined storage location.

#### Test 19. Display the Reports

**Requirement:** R13

**Feature:** F19

**Testing Procedure:** The system should be able to display the generated reports.

**Expected Results:** Pass if there is a way to display the generated reports.

#### Test 20. Rename a Report.

**Requirement:** R13

**Feature:** F20

**Testing Procedure:** We test if the system is capable to rename one of the generated reports.

**Expected Results:** Pass if the system is capable to rename a report.

### Test 21. Delete a Report.

**Requirement:** R13

**Feature:** F21

**Testing Procedure:** We test if the system is capable to delete a report.

**Expected Results:** Pass if the system is able to delete one of the generated reports.

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

**Requirement:** R11

**Feature:** F22

**Testing Procedure:** The system should be able to display the details of user account through the front end.

**Expected Results:** Pass if the details of a user account are displayed on the frontend.

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

**Requirement:** R12

**Feature:** F23

**Testing Procedure:** FCS files are files that contain large amounts of data. Displaying their data is both challenging and essential. A way to overcome the difficulty to display the contents, if to display the head or the tail lines, or simple metadata about the FCS file.

**Expected Results:** Pass if the details of the FCS file are displayed on the frontend of the system.

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

**Requirement:** R13

**Feature:** F24

**Testing Procedure:** A generated report serves a role, if its contents are displayed to the frontend of the system.

**Expected Results:** Pass if the details of the report are displayed successfully on the front end of the system.

### Test 25. Handshake with a remote database, (firebase).

**Requirement:** R1, R2, R3, R7.R9, R10, R11, R13

**Feature:** F1, F2, F3, F7, F10, F11, F12, F14, F15, F19, F20. F21, F22 F24

**Testing Procedure:** while there is no feature that describes the connection of the system with a remote database, it is clear that a connection with a remote database is essential., and while we

describe the test as a handshake, the truth is that this test includes more than a handshake. This procedure tests not only the connection with a remote database, (google firebase in our case), but also tests the ability of the system to send data back and forward to the database.

**Expected Results:** Pass if the system is able to send and fetch data from and to the remote database, (Google Firebase).

## Evaluation or Analysis of Test Results



## Conclusion

## References

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

(Books)

(Articles and Journals)

(WWW)