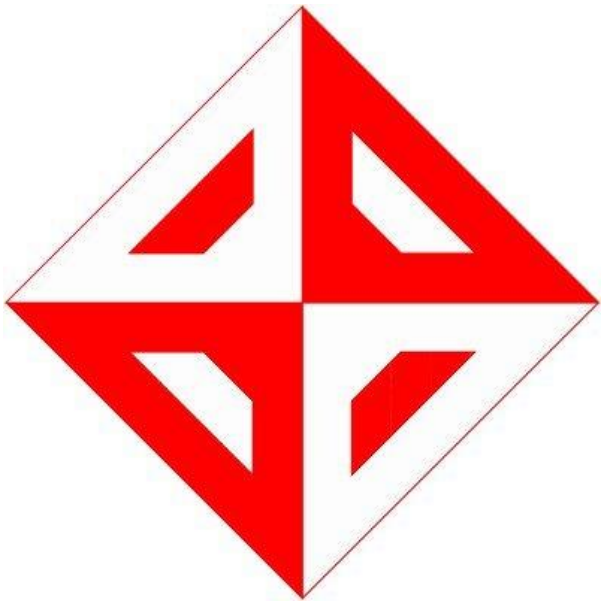

MIDDLE EAST TECHNICAL UNIVERSITY

Spring 2021

Computer Engineering Department

PROJECT:

OPENFLEXURE MICROSCOPE

Software Design Description

Version 1.1

Prepared by:

Zeynepsu Kesim - 2292340

Table of Contents

List of Figures

List of Tables

Version History

Version	Date	Explanation
1.0	28.05.2021	'Purpose of the system', 'scope' and 'stakeholders and their concerns' is written. Context diagram, use case diagram and use case tables are renewed.
1.1	11.06.2021	Full document delivered.

Table 1: Version History

1 Introduction

This document is the Software Design Document of the OpenFlexure Microscope which is co-developed by University of Bath and the Tanzanian engineering company STICLab.

1.1 Purpose of the System

The purpose of this project is to develop a cross-platform controllable microscope that will not rely on outdated hardware, or any equipment and reduce the work of microscopists that need to write complicated experiment codes and bind them together with inconsistent interfaces to run their experiments.

1.2 Scope

In the scope of the system, scientists can run their experiments without depending on any outdated hardware or inconsistent interfaces. To accomplish this, scientists register the system from Web GUI via entering their scientific credentials. When they are approved by the system, they can control the microscope, run experiments, and generate reports. During their work, system runs number of services that are connected to the Web GUI to fulfil scientist's needs. Once scientists are in the system, they can monitor the microscope with data coming from the streaming service. Also, they can control the camera, microscope position via HTTP calls to IO Controller which is running on a Raspberry Pi. All these features are for making the scientist's work less than before.

The main services of the system are:

- ❖ Web GUI
- ❖ Microscope IO Controller
- ❖ Streaming Service
- ❖ Analytics Service
- ❖ Experiment Processor Service

1.3 Stakeholders and Their Concerns

OpenFlexure Microscope is a smart microscope to be mainly used in experiments and research purposes. Thus, the main concern of the microscope project can be divided into two for study and research purposes. Study side of the concerns includes running experiments and controlling the microscope and its features efficiently while the research purposes includes collecting data, analysing data and observing the usage of the microscope for assessment and research. Study side consists of 'scientists' which are divided into 4 sub-groups, and research side consists of 'researchers'. Moreover, we have 'IT Staff'

Scientists

Scientists, all of the following sub-groups, are users of the system. Their first concern is to be able to use the microscope efficiently and as easy as possible. Therefore, the Web GUI shall be designed in a simplistic way to get users away from complexity and satisfy the 'easiness of use purpose'. Moreover, users concern about the design of the Web GUI to make them bored and reluctant to do the work. A good-looking design shall help users to engage in the system joyfully, and thus, helping them to conduct their works more willingly. Their second concern is to use the microscope fast and contemporaneously, and thus, reach and exercise their wants in the minimum time possible. The second concern is handled by the system limitations that are determined by the needs of the users by the IT Staff. Their third concern is about the 'learning side of the system'. Users would like to learn the system well as fast as possible, and to be able to use the system accordingly. The third concern is handled by the 'demo' coming with the project. The demo will explain the basics of the system and how users will use the system efficiently. Their last concern is about to get informed and helped when a technical problem occurs in the system. This concern is sufficiently handled by the IT Staff, reachable 7/24 hours. When a problem occurs, the IT Staff get in touch with the users and explain how to fix it, or when the system will be available successfully.

- **Professors:** Professors are the academics and/or the main experimenters, observers and researchers about the findings. They guide students and assistants about what to do in experiments, or how to analyse the findings after the experiments. Besides their help to students, they conduct academic research and write articles according to the findings.

- **Students:** Students are the students in universities. They use the microscope for study purposes for their classes with the guidance of their professors.
- **Research Assistants:** Research assistants are the ones mostly in their master's or doctorate's degree students. They use the microscope according to their own thesis research, conduct experiments accordingly and/or use the microscope for their classes with the guidance of the professors.

Researchers

Researchers are different from the 'research purposes scientists'. These researchers are the ones who collect data about the usage of the microscope. The scientists use the microscope mainly for the 'biological research' while researchers here collects data about the microscope itself. Their main concerns are gather information and analyse the information about the OpenFlexure Microscope and its features/usage. They are primarily concerned with the OpenFlexure Microscope's outcomes and the real-world performance. Researchers use machine learning algorithms and other statistical evaluation methods to evaluate the data and build new models and test them for future products.

IT Staff

IT Staff is mainly responsible for the efficient usage of the system. Their concems are satisfying the limitations determined by the users and ensuring the system work flawlessly, consistently and continuously. When a technical problem occurs with the system, IT Staff get in touch with the users to give them information about the problem and how long it will last. Moreover, they are concerned about the requirements that are determined well, and complete.

2 References

This document is prepared with respect to **IEEE 1016-1998 standard**: IEEE standard for information technology–systems design–software design descriptions. (2009). New York, NY: Institute of Electrical and Electronics Engineer

Collins, J. T., Knapper, J., Stirling, J., McDermott, S., & Bowman, R. (2021). *Modern Microscopy with the Web of Things: The OpenFlexure Microscope Software Stack*. http://arxiv.org/abs/2101.00933

Committee, S. & S. E. S. (2009). 1016-2009 - IEEE Standard for Information Technology--Systems Design--Software Design Descriptions. In *Middle East* (Vol. 2009, Issue July). https://standards.ieee.org/standard/1016-2009.html

Stirling, J., Sanga, V. L., Nyakyi, P. T., Mwakajinga, G. A., Collins, J. T., Bumke, K., Knapper, J., Meng, Q., McDermott, S., & Bowman, R. (2020, October 29). The OpenFlexure Project. The technical challenges of Co-Developing a microscope in the UK and Tanzania. *2020 IEEE Global Humanitarian Technology Conference, GHTC 2020*. https://doi.org/10.1109/GHTC46280.2020.9342860

3 Glossary

Term	Definitions
IT Staff	Information Technology Staff
API	Application Programming Interface
SD Card	Secure Digital Memory Card
HTTP	Hyper Text Transfer Protocol There can be different types of HTTP requests like POST (update or add a resource), DELETE (delete a resource). Commonly, this protocol returns a status code 200.
Web GUI	Graphical User Interface for scientists to use the system
UDP	User Datagram Protocol is a communication protocol used across the internet for video streaming etc.
IO Controller	Input Output Controller
XSS	Cross Site Scripting
TCP	Transmission Control Protocol
USB	Universal Serial Bus
HDMI	High-Definition Multimedia Interface

Table 2: Glossary

4 Architectural Views

4.1 Context View

In this viewpoint, all system use-cases and descriptions explained with details. These descriptions show how system should behave in specific situations and how the functionalities implemented. Context diagram shows how subsystems interacts with each other and use case diagram and use case descriptions specified below the context diagram.

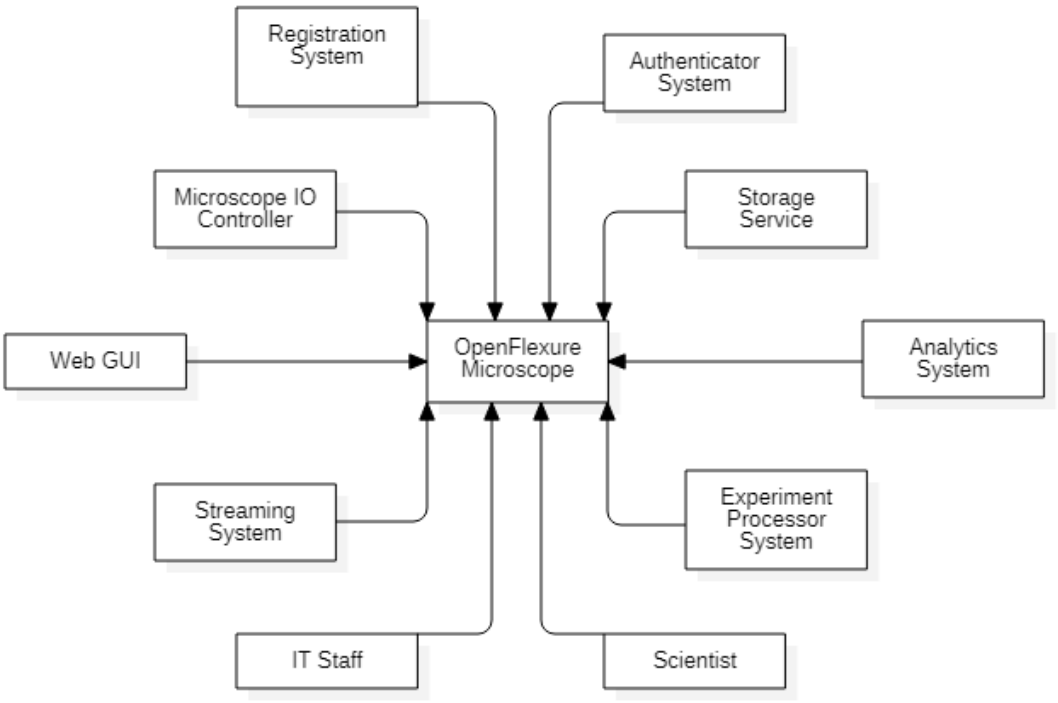

Figure 1: Context Diagram

Figure 2: Use Case Diagram

Use Case Name	**Get error logs and manage data**
Actors	IT Staff, Storage Service
Description	IT staff checks the error logs and data to make sure all services up and running
Data	-
Preconditions	-
Stimulus	-
Basic Flow	1. IT Staff enters to the Database Management System, and gather the logs 2. IT Staff checks the error logs of the application 3. Staff make sure there are no critical errors
Alternative Flow #1	1. IT Staff enters to the internal storage (or SD Card) to make sure there are no corrupted data
Exception Flow	-
Postconditions	-

Table 3: Get error logs and manage data

Use Case Name	**System Recovery**
Actors	IT Staff, Storage Service
Description	IT staff recovers the system storage from last backup
Data	-
Preconditions	-
Stimulus	-
Basic Flow	1. IT Staff enters to the Database Management System, and checks the last system backup 2. IT Staff checks the application availability 3. Staff starts a recovery process if there is an application which is in down
Alternative Flow #1	-
Exception Flow	-
Postconditions	-

Table 4: System Recovery

Use Case Name	**Set User Privileges**
Actors	IT Staff, Storage Service
Description	IT staff sets user privileges and permissions
Data	-
Preconditions	-
Stimulus	-
Basic Flow	1. IT Staff enters to the Database Management System, and checks system users 2. IT Staff updates the user permissons if there is a need
Alternative Flow #1	-
Exception Flow	-
Postconditions	User permissions updated

Table 5: Set User Privileges

Use Case Name	**Create Scientific Group**
Actors	Scientist, Web GUI, Storage Service
Description	Scientists can create new scientific groups with other scientists for their researches
Data	-
Preconditions	Scientist should be authenticated
Stimulus	Click on “create scientific group“ button
Basic Flow	1. Scientist enters the scientific group page on the Web GUI and clicks create scientific group. 2. Scientist enters scientific group details and creates an invitation list via sms or email, then creates the group. 3. Scientist can also invite other scientists later to his research group and systems sends an email or sms invite. 4. Scientist can set research topics and targets in the organisation group properties.
Alternative Flow #1	-
Exception Flow	-
Postconditions	New scientific group is created

Table 6: Create Scientific Group

Use Case Name	**Join Scientific Group**
Actors	Scientist, Web GUI, Storage Service
Description	Scientists can join different scientific groups
Data	-
Preconditions	-
Stimulus	Scientist receives an invitation email.
Basic Flow	1. Scientist enters the link that has been given in the email. 2. Scientist enters the credentials on the page that link redirects. 3. Scientist previews the scientific group details and joins the group.
Alternative Flow #1	-
Exception Flow	-
Postconditions	Scientist joined the scientific group

Table 7: Join scientific group

Use Case Name	**Register the system**
Actors	Scientist, Registration system
Description	Users gives their scientific credentials to the registration system to be verified
Data	Information about the scientist
Preconditions	-
Stimulus	Filled information sent from Web GUI
Basic Flow	1. Scientist gives their personal and scientific credentials 2. Registration system checks whether the credentials scientist has supplied are correct. 3. A verification code goes to the email of the scientist. 4. Scientist enters the verification code to the system 5. Registration system verifies the scientist
Alternative Flow #1	2. Given credentials are wrong. Then, scientist is prompted to enter correct information
Alternative Flow #2	5. System fails to verify scientist's verification code due to wrong entrance of the verification code
Exception Flow	3. E-mail did not reach the scientist's email due to error on email service. Then, scientist requests verification code again
Postconditions	Scientist is verified

Table 8: Register the System

Use Case Name	**Authenticate User**
Actors	Scientist, Authentication System
Description	Authentication system checks the given login credentials are valid
Data	Login credentials
Preconditions	Scientist should be registered
Stimulus	-
Basic Flow	1. Scientist enters login information. 2. Auth system check if the login credentials are correct and exist. 3. System make user authenticated in the system
Alternative Flow #1	2. User enters wrong credentials. Then system prompts user to give correct credentials
Alternative Flow #2	2. User does not exist in the system. Then system redirects him/her to register page
Exception Flow	-
Postconditions	User is authenticated

Table 9: Authenticate User

Use Case Name	**Run Data Analysis on the Experiment Results**
Actors	Scientist, Analytics Service, Storage System, Web GUI
Description	When scientist wants an analysis of the past data, system makes the wanted computations according to the parameters that scientist has supplied and generates the reports.
Data	Data in the storage system and the parameters that scientist supplied to the system.
Preconditions	-
Stimulus	Entering the parameters to the system by the scientist
Basic Flow	1. Scientist enters the Web GUI. 2. Enters the parameters that being wanted to be computed. 3. System gets the past data from the storage system. 4. System makes the analysis according to the parameters. 5. Generates a report with the findings
Alternative Flow #1	3. Data is insufficient to compute the wanted parameters and make the analysis. Then, system gives a warning to the scientist. 4. System offers 2 options to ask the scientist if s/he wants to continue with the existing insufficient data and make the computations accordingly, or enters new parameters to compute from the beginning.
Exception Flow	3. System may not be able to reach the past data in the storage system due to connection or time-out error. 4. System retries to get data from the storage system (retry limit is 1)
Postconditions	A final report is generated

Table 10: Run Data Analysis on the Experiment Result

Use Case Name	**Run Scientific Experiment**
Actors	Scientist, Experiment Processor System, Web GUI
Description	Scientists can run pre-defined experiments from the platform or upload a custom experiment to be executed.
Data	Experiment features, parameters supplied to the system by the scientist
Preconditions	There should be a sample to experiment on the microscope
Stimulus	Run an experiment command from the scientist
Basic Flow	1. Scientist selects which experiment to run. 2. Scientist decides parameters to be computed on that experiment. 3. Experiment processor system computes the parameters. 4. Generates a report with the findings of that experiment
Alternative Flow #1	1. Scientist can upload a custom experiment to the platform. 2. Scientist decides parameters to be computed on that experiment. 3. Experiment processor system computes the parameters. 4. Generates a report with the findings of that experiment.
Exception Flow	-
Postconditions	A final report is generated

Table 11: Run Scientific Experiment

Use Case Name	**Manage account**
Actors	Scientist, Web GUI, Storage Service
Description	Scientists can update their account information.
Data	-
Preconditions	Scientist should be authenticated
Stimulus	Update request from Web GUI
Basic Flow	1. Scientist enters the manage account page. 2. Scientist updates the account information. 3. Web GUI posts an HTTP POST request to storage service. 4. Storage service handles the request and updates the account information in the Database
Alternative Flow #1	5. Scientist may want to delete their account. 6. Web GUI posts an HTTP DELETE request to storage service. 7. Storage service handles the request and make user disabled.
Exception Flow	-
Postconditions	Account information is updated

Table 12: Manage Account

Use Case Name	**Get activity history**
Actors	Scientist, Web GUI, Storage Service
Description	Scientist can see their activity history, past experiments etc. or the activity history of the scientists of the same organisation.
Data	-
Preconditions	Scientist should be authenticated
Stimulus	Scientist should go to the activity history page
Basic Flow	1. Scientist enters the activity history page on the Web GUI. 2. Web GUI gathers data from storage service and returns the activity history
Alternative Flow #1	3. Scientist enters the activity history page on the Web GUI, and selects the 'see organisation activity history' option. 4. Web GUI gathers data from storage and returns the activity history.
Exception Flow	-
Postconditions	Activity history displayed to the scientist

Table 13: Get Activity History

Use Case Name	**Take snapshots**
Actors	Scientist, Streaming service, Web GUI
Description	Scientist can take camera snapshots of the microscope and then save it on his/her account or locally
Data	-
Preconditions	Scientist should be in the live streaming feed
Stimulus	Snapshot command from Web GUI
Basic Flow	1. Scientist should go to the streaming page in Web GUI. 2. Scientist should click the snapshot button. 3. Web GUI sends an HTTP request to the streaming service. 4. Streaming service creates another thread instead of live feed thread and sends that specific snapshot in the payload. 5. Scientist takes the snapshot and save it locally.
Alternative Flow #1	5. Scientist can save the snapshot in the platform using Raspberry Pi instead of downloading locally.
Exception Flow	4. Streaming service unable to create new thread due to insufficient processor power. System generates an error log message for the investigations of IT Staff.
Postconditions	Scientist takes the snapshot

Table 14: Take Snapshot

Use Case Name	**Get Camera Information**
Actors	Scientist, Microscope IO Controller
Description	Scientist can see camera information like angle, position, speed etc.
Data	-
Preconditions	There should be a monitor that scientist can see the stream (laptop, or external monitor)
Stimulus	Scientist should go to the live streaming page.
Basic Flow	1. Scientist enters the streaming page. 2. Web GUI sends a request to microscope IO controller to gather camera information. 3. Microscope IO Controller checks the hardware properties and sends the information to the Web GUI. 4. Web GUI displays the camera information on the live streaming page.
Alternative Flow #1	-
Exception Flow	2. Web GUI may not be able to reach the microscope IO controller due to connection or time-out error. 3. System retries to reach the microscope IO controller (retry limit is 1)
Postconditions	Information about the camera is taken and displayed on the live streaming page on Web GUI

Table 15: Get Camera Information

Use Case Name	**Get Live Camera Feed**
Actors	Scientist, Streaming Service
Description	Streaming service streams the live camera feed to the Web GUI for scientists
Data	-
Preconditions	There should be a monitor that scientist can see the stream (laptop, or external monitor)
Stimulus	Scientist should go to the live streaming page.
Basic Flow	1. Streaming service finds all the connected devices (laptops, displays etc.) 2. Streaming service sends all the frames that captured by the camera via UDP. 3. Web GUI gets the frames and displays the camera feed.
Alternative Flow #1	-
Exception Flow	1. Streaming service may not be able to find connected devices or stream. In that case IT staff should be informed.
Postconditions	Scientist can see live camera feed.

Table 16: Get Live Camera Feed

Use Case Name	**Control the Microscope**
Actors	Scientist, Microscope IO Controller System
Description	When scientist wants to change the direction, position, or angle of the microscope, s/he can do it by using the directives in the Web GUI.
Data	New direction, position, or angle of the microscope supplied by the scientist
Preconditions	Scientist should be in live streaming page
Stimulus	Incoming update directive from the scientist
Basic Flow	1. Scientist sets the new values for the microscope properties. 2. GUI sends the update request to the IO Controller system 3. System sets new values for the microscope properties with using Arduino. 4. System returns an HTTP response with status 200.
Alternative Flow #1	-
Exception Flow	1. Microscope IO Controller system may not be reachable due to a hardware failure. In case that happens, Web GUI generates an error log for the IT staff.
Postconditions	New microscope position, direction, or angle value is set.

Table 17: Control the Microscope

4.2 Composition View

In this view, components and interfaces between them are presented in a high level view. Detailed explanations are given in the related sections.

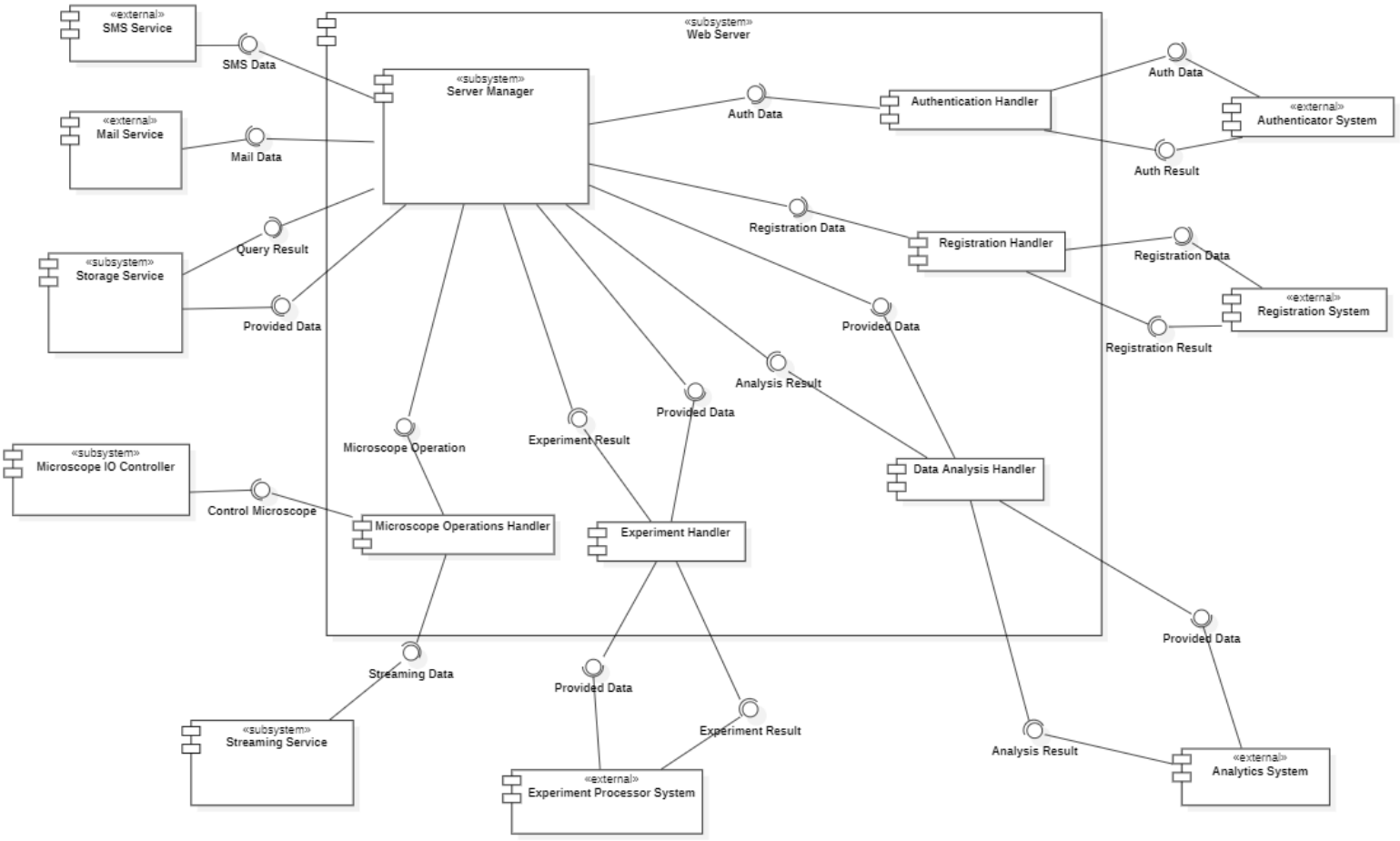

Figure 3: Component Diagram

Design Rationale:

- Server Manager is the fundamental component, which manages/handles information flow and data flow in the Web Server. Moreover, it authorizes/authenticates users.
- Server Manager controls components which have external/internal interfaces. It invokes components based on incoming request.
- Communications between components are mostly passed through server manager because of security considerations.
- Storage Service securely stores all the information about the whole system, like users, microscopes, experiments, analysis reports etc…
- Mail service is dedicated for register and invite scientist flows and it is an external service, Amazon SES.
- SMS service is dedicated for invite scientist flow and it is also an external service, Amazon SNS.

- Microscope IO Controller is responsible for changing position of the microscope and the camera. It writes current microscope situation to the database every time microscope operations handler invokes it.
- Streaming Service is responsible for supplying live camera feed to the user. It is always invoked by the microscope operations handler and transmits the feed to the server manager.
- Experiment Processor System is invoked by experiment handler and is responsible with running scientific experiments and generating reports.
- Experiment Processor System can also take recently uploaded experiments from storage service to run custom experiments.
- Analytics system is responsible with running data analysis on the past experiment data and it is invoked by data analysis handler.
- Authenticator system is invoked by authentication handler and is responsible with authenticating users on the system.
- Authenticator system also provides setting new user privileges feature to the admins.
- Registration system is invoked by registration handler and it takes place in the registration process.

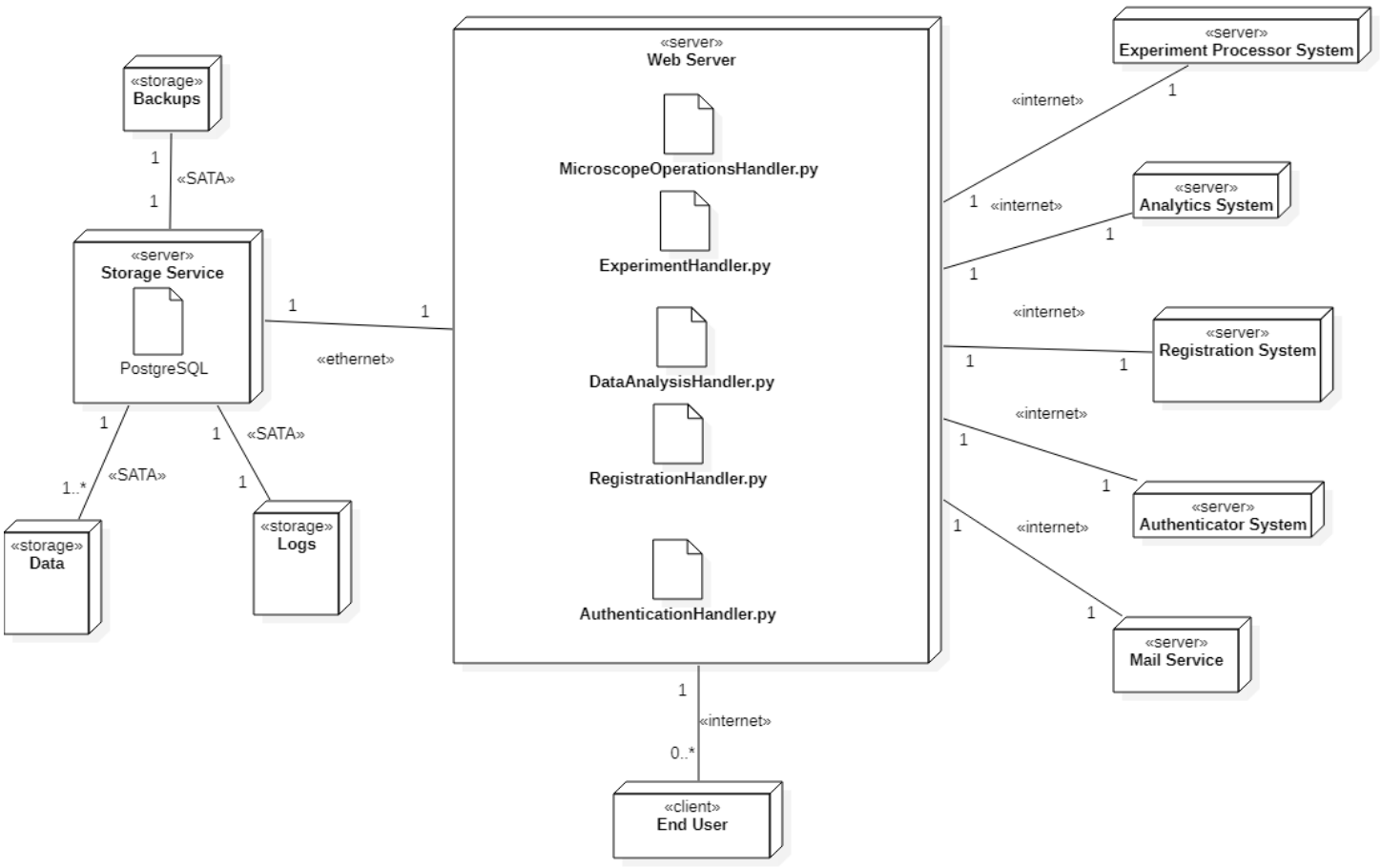

Figure 4: Deployment Diagram

Design Rationale:

- ❖ Python/Flask is used in communication between components and PostgreSQL is used as DBMS
- ❖ Handlers are also implemented by using python since crucial external services like experiment processor system and analytics system is implemented with python.
- ❖ All communication with external systems is done with REST API Calls between handlers and systems.
- ❖ There are three separate DB storage:
    - Backups
    - Application and error logs
    - System Data
- ❖ System backup is taken when no user is connected to the microscope.
- ❖ End user is connected over internet from the website.
- ❖ SATA is used to have a fast communication between server and storage.
- ❖ All connection between all components (storage, server manager, external systems) are done via HTTPS protocol.

4.3 Information View

In this view, all the data organization and relations are explained as well as the CRUD operations of the systems on the data. Moreover, the effects of the system operations will be examined.

4.3.1 Interfaces

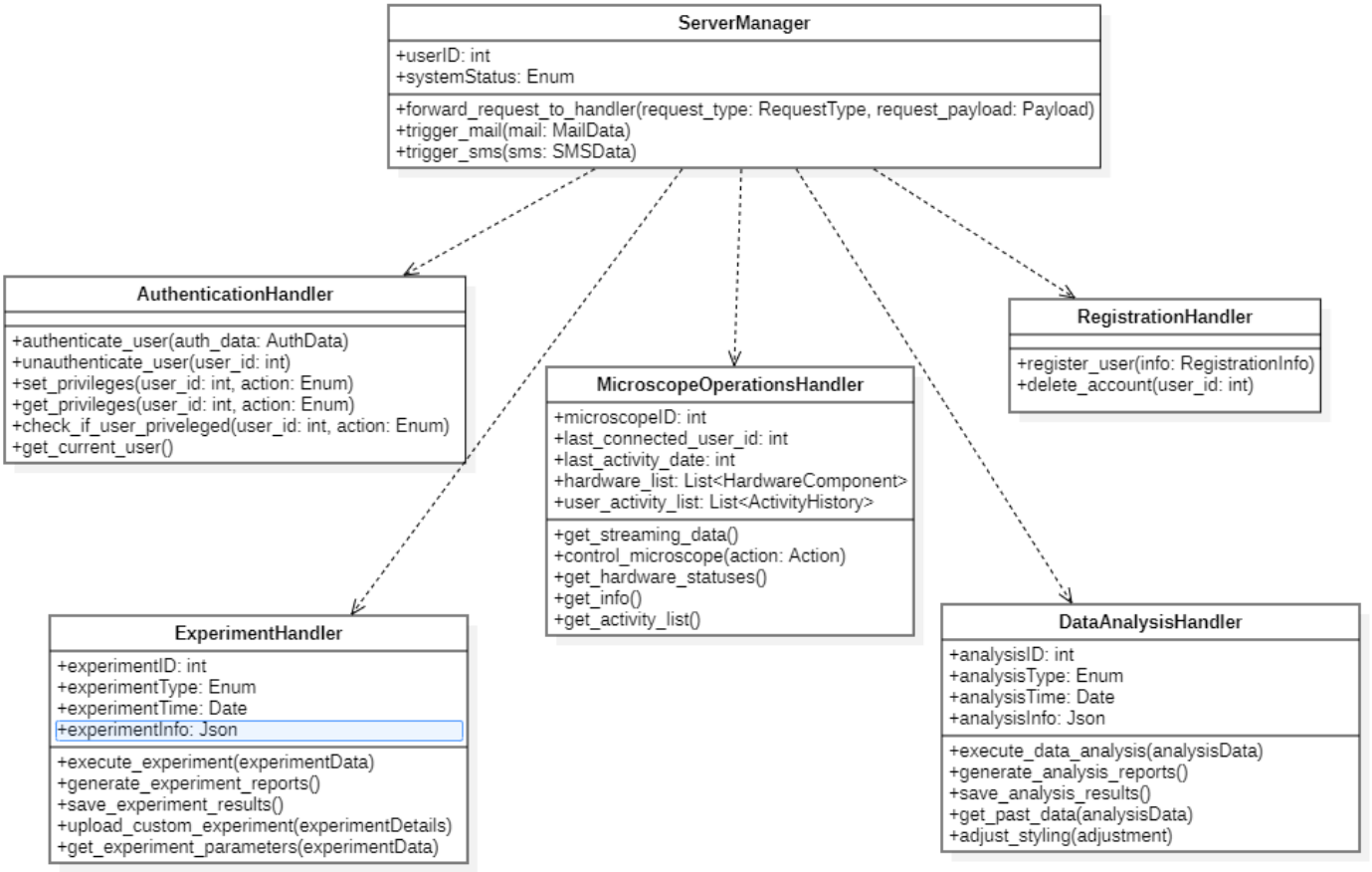

Figure 5: Interface Class Diagram

Operations	**Descriptions**
forward_request_to_handler	Forwards incoming request to related handlers.
trigger_mail	Sends a call to Amazon SES with a mail body to be received by a user.
trigger_sms	Sends a call to Amazon SNS with a mail body to be received by a user.
get_streaming_data	Gets live camera feed from microscope camera to the WEB GUI.
control_microscope	Updates microscope position or camera angle with respect to given action.

get_hardware_statuses	Get all up to date hardware statuses of the microscope.
get_info	Get information about microscope hardware.
get_activity_list	Get user's activity history of the microscope.
execute_data_analysis	Execute data analysis with given data analysis parameters.
generate_analysis_reports	Generate data analysis report.
save_analysis_results	Saves analysis results to the database.
get_past_data	Get past data from the DB according to analysis data
adjust_styling	Adjust data analysis report's stylings.
execute_experiment	Execute experiment on the microscope.
generate_experiment_reports	Generate reports for the experiment that ran.
save_experiment_results	Saves experiment results to the DB.
upload_custom_experiment	Uploads new type of experiment to the DB for future reuse.
get_experiment_parameters	Get experiment parameters that is being computed by the system automatically.
authenticate_user	Logs in the user.
unauthenticated_user	Logs out the user.
set_priveleges	Sets user's privileges on the system for a specified action.
get_priveleges	Gets user's current privileges.
check_if_user_priveleged	Checks if a user is privileged to do a specified action.
get_current_user	Gets current authenticated user.
register_user	Create an account for the user.
delete_account	Delete user's account.

Table 18: Operations Descriptions

Operations	Inputs	Outputs	Exceptions
forward_request_to_handler	request_type, request_payload	Success/failure message, or analysis report or experiment report according to request type	Connection error on services
trigger_mail	mail_data	Success/failure message	Amazon SES connection error
trigger_sms	sms_data	Success/failure message	Amazon SNS connection error
get_streaming_data	-	Live streaming feed (images)	Microscope camera is not reachable
control_microscope	Action	Success/failure message	Hardware error, or invalid action
get_hardware_statuses	-	True if all hardware online, false otherwise	One of the hardware is broken.
get_info	-	Hardware information list	Microscope device is not online
get_activity_list	-	List of activity history of the users connected to the microscope	Database connection error
execute_data_analysis	Data analysis parameters	Analysis information as json	Analysis can not be done because lack of sufficient data
generate_analysis_reports	Json analysis information in that handler instance	Analysis report as pdf	Generated report is too long
save_analysis_results	Json analysis information in that handler instance	True if save to DB successful, False otherwise	DB connection error
get_past_data	Analysis parameters	Analysis data	DB connection error
adjust_styling	Report Styling parameters, and json analysis information in that handler instance	Analysis report	Given stylings is not supported
execute_experiment	Experiment parameters	Experiment info as json	Experiment cant be done because lack of reliable data
generate_experiment_reports	Json experiment info in that handler instance	Experiment report as pdf	Report is too long

save_experiment_results	Json experiment info in that handler instance	True if write DB successful, False otherwise	DB connection error
upload_custom_experiment	Experiment parameters	True if write DB successful, False otherwise	DB connection error, or experiment is too complex to be processed
get_experiment_parameters	Experiment data	Computed parameters if successful, None otherwise	Parameters cant be computed automatically due to complexity
authenticate_user	Auth data	True if auth success, False otherwise	Auth Data is incorrect
unauthenticated_user	user id	True if success, False otherwise	User already unauthenticated
set_priveleges	User id, action	True if success, False otherwise with an error message	Current user not authorized to set this user's privileges
get_priveleges	User id, action	Privilege info	DB read error
check_if_user_priveleged	User id, action	True if user authorized, False otherwise	DB read error
get_current_user	-	Session(with user_id)	DB connection error
register_user	Registration Data	True if success, False otherwise	Identity cannot be validated in external service
delete_account	User_id	True if success, False otherwise	User already deactivated, or user not exists

Table 19: Operations Design

Design Rationale:

- Server manager is responsible for all the database transactions.
- Handlers are responsible for intercommunication between external/subsystems and server manager; therefore, they have operations providing functionality of that.
- All the provided methods are asynchronously working methods that are awoken by components.
- forward_request_to_handler function always executed before the other methods in the interfaces.
- get_streaming_data runs completely in a different thread for not blocking other functionalities of the application

4.3.2 Database Operations

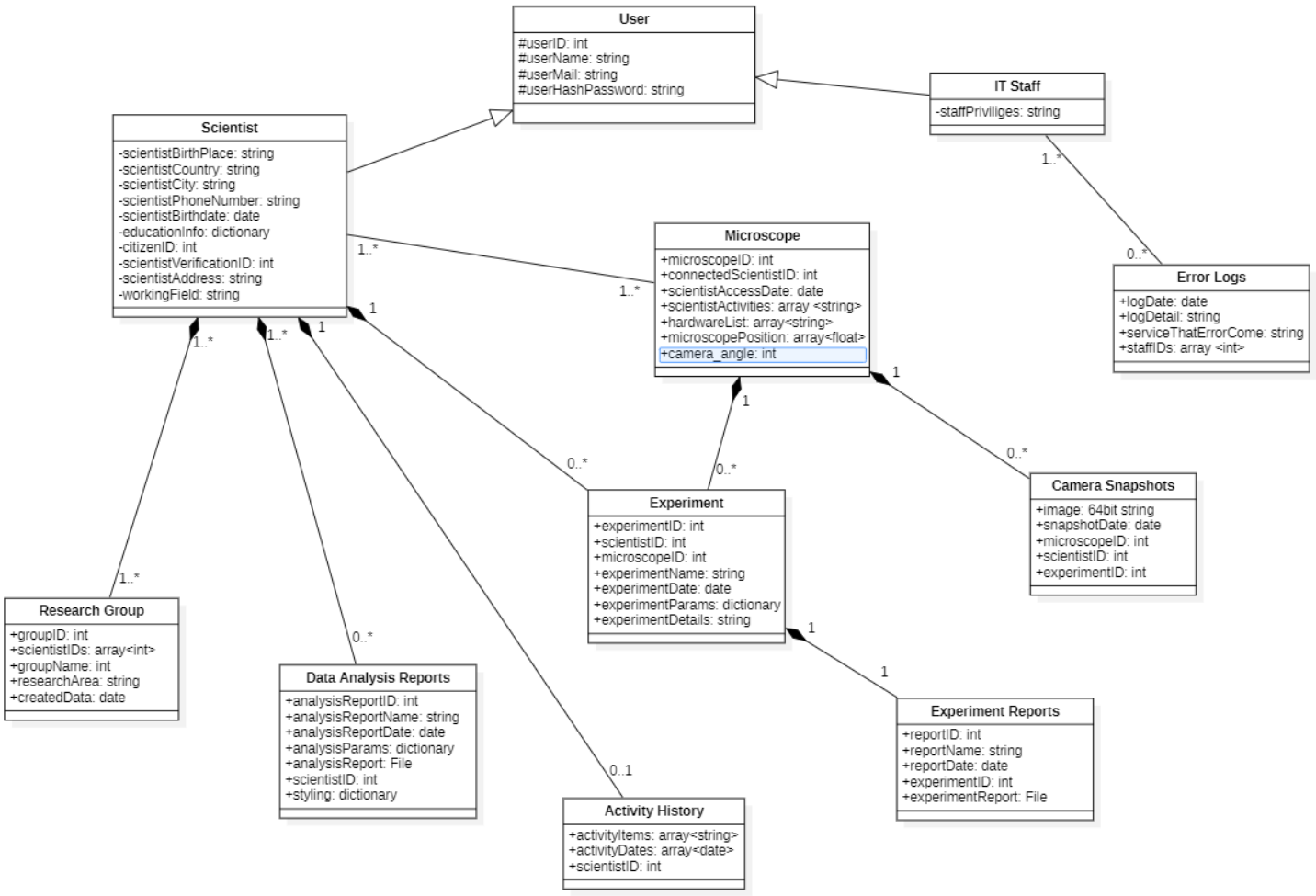

Figure 6: Database Class Diagram

Operations	**CRUD Operations**
forward_request_to_handler	Create: - Read: - Update: - Delete: -
trigger_mail	Create: - Read: User Update: - Delete: -
trigger_sms	Create: - Read: User Update: - Delete: -
get_streaming_data	Create: CameraSnapshots Read: CameraSnapshots, Scientist Update: - Delete: -

control_microscope	Create: - Read: Microscope Update: Microscope Delete: -
get_hardware_statuses	Create: - Read: Microscope Update: - Delete: -
get_info	Create: - Read: Microscope Update: - Delete: -
get_activity_list	Create: - Read: Microscope, Scientist, ActivityHistory Update: - Delete: -
execute_data_analysis	Create: - Read: - Update: - Delete: -
generate_analysis_reports	Create: - Read: - Update: - Delete: -
save_analysis_results	Create: DataAnalysisReports Read: DataAnalysisReports, Scientist, ResearchGroup Update: - Delete: -
get_past_data	Create: - Read: DataAnalysisReports Update: - Delete: -
adjust_styling	Create: - Read: DataAnalysisReports Update: DataAnalysisReports Delete: -
execute_experiment	Create: - Read: - Update: - Delete: -
generate_experiment_reports	Create: - Read: - Update: - Delete: -
save_experiment_results	Create: ExperimentReports Read: - Update: - Delete: -

upload_custom_experiment	Create: Experiment Read: Experiment Update: - Delete: -
get_experiment_parameters	Create: - Read: Experiment, ExperimentReports Update: - Delete: -
authenticate_user	Create: - Read: User Update: User Delete: -
unauthenticated_user	Create: - Read: User Update: User Delete: -
set_priveleges	Create: - Read: - Update: User Delete: -
get_priveleges	Create: - Read: User Update: - Delete: -
check_if_user_priveleged	Create: - Read: - Update: User Delete: -
get_current_user	Create: - Read: User Update: - Delete: -
register_user	Create: User Read: - Update: ResearchGroup Delete: -
delete_account	Create: - Read: - Update: User Delete: User

Table 20: CRUD Operations

Design Rationale:

- PostgreSQL is the DBMS used in the project
- If registering is not via invite with a research group, the update on ResearchGroup is skipped in register_user
- Every entity is stored in database in order to provide persistency of the data.
- Camera snapshots are saved in a separate thread to check experiment later.

4.4 Interface View

In this view, the internal interfaces between the components of the system and the external interfaces between web server and the other systems will be specified in detail.

4.4.1 Internal Interfaces

- **Interface between Microscope IO Controller and Microscope Operations Handler:**
  Microscope operations handler will use the microscope IO controller in order to control the microscope. When 'microscope_action' request has come to the handler, it will redirect that request to the IO Controller and updates the microscope position or camera angle according to the request.

  Design Rationale:
  - Microscope Operations Handler is crucial for communicating IO Controller.
  - Microscope shall be available to use in order to use this interface properly

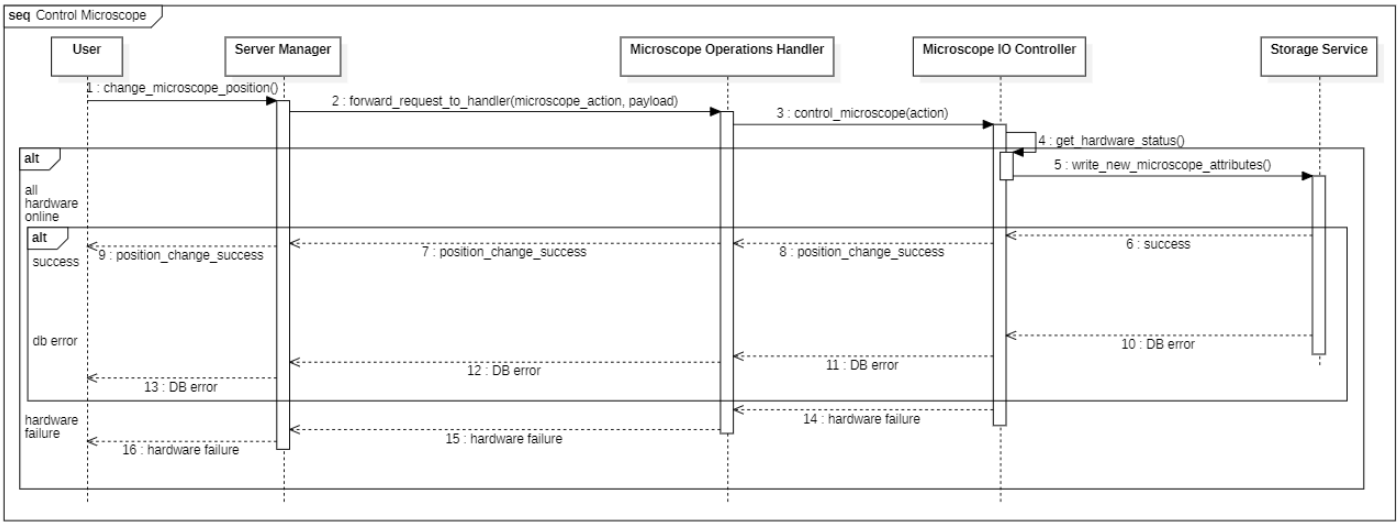

Figure 7: Interface between Microscope IO Controller and Microscope Operations Handler

- **Interface between Streaming Service and Microscope Operations Handler:**
  Microscope operations handler will use the streaming service in order to get live feed from microscope camera. When 'live_stream' request has come to the handler, it will redirect that request to the streaming service and start gathering feed.

  Design Rationale:
  - All requests should run in separate threads to serve same camera feed to multiple users.
  - Saving snapshots also should be in separate thread to not produce latency in the camera feed.

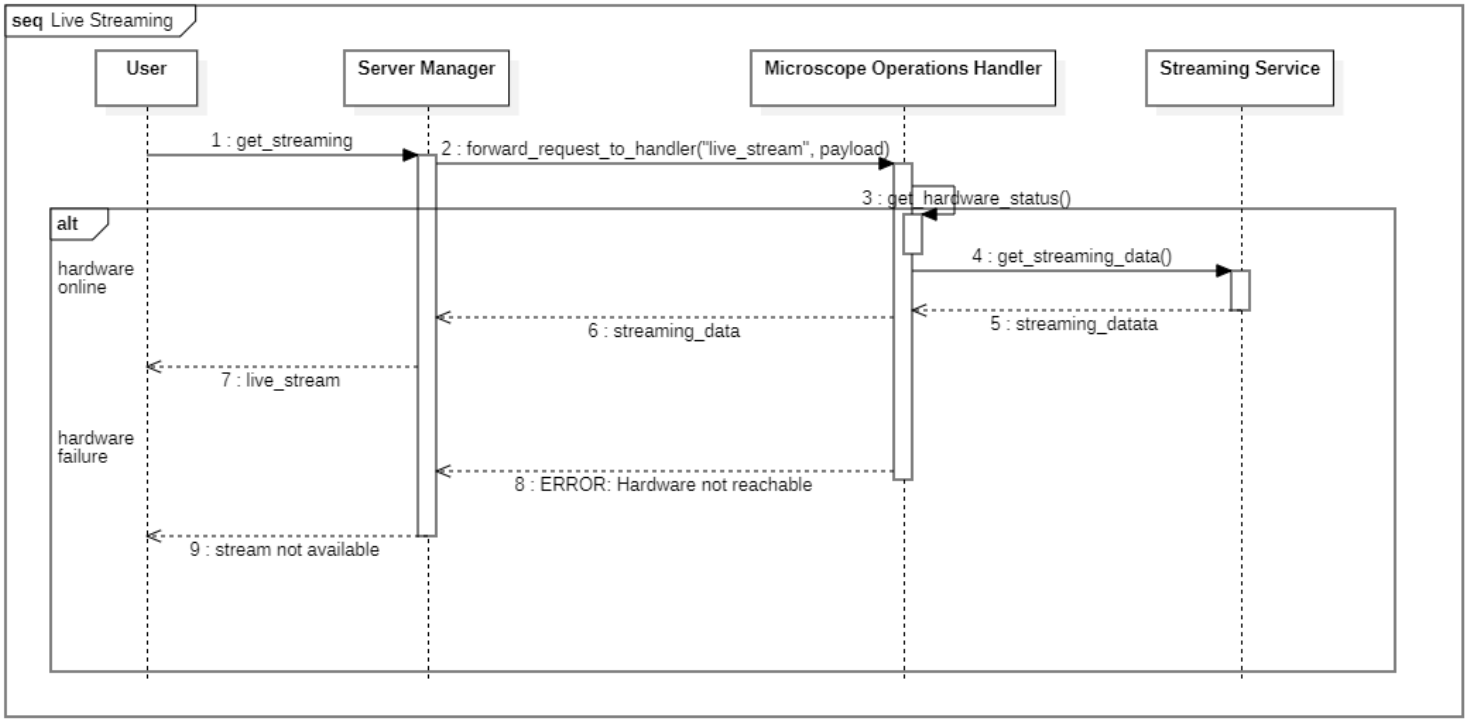

Figure 8: Interface between Streaming Service and Microscope Operations Handler

- **Interface between Server Manager and Storage Service:**
  Server manager queries the Database Server to get specific information for an operation. The query passed as string and dbms runs it as SQL. If query fails, server returns error message, if query is successful, result would be sent back.

  Design rationale:
  - All query results shall be stored in case of a failure
  - All data shall be persistent in the database; hence, server manager shall not sent delete queries

- **Interface between Server Manager and Authentication Handler:**
  Server manager uses the authentication handler to communicate with external interfaces. Authentication handler requires auth_data from the server manager to make users authenticated/unauthenticated. Besides from that, authentication handler can manage user privileges for admins.

  Design rationale:
  - Server manager shall pass auth_data to handler
  - Authentication handler shall check privilege of the users in any actions in case of a malicious activity.

- **Interface between Server Manager and Registration Handler:**
  Server manager uses the registration handler to register users to the system. This flow has a few parts from passing input to the registration handler to the writing user info

to the database. This handler also signals server manager to update research groups if the registration process is started via an email invite to a research group.

Design rationale:

- Registration data shall be passed to the handler
- Whenever a registration fails, this should be logged to investigated further by the admins

❖ **Interface between Server Manager and Data Analysis Handler:**
Server manager passes analysis data in case of the request type is 'analysis', and uses data analysis handler to communicate with external system. This communication should also be done async to not block other operations since this operation may take time depending on the analysis parameters. Also, server manager should write any related data analysis report to the database.

Design rationale:

- Should run concurrently
- Should save related reports to the database.
- Errors and exceptions should be logged for further investigations

❖ **Interface between Server Manager and Experiment Handler:**
Server manager passes experiment data in case of the request type is 'experiment', and uses experiment handler to communicate with external system. This communication should also be done async to not block other operations since this operation may take time depending on the experiment parameters. Also, server manager should write any related experiment report to the database.

Design rationale:

- Should run concurrently
- Should save related reports to the database.
- Errors and exceptions should be logged for further investigations

4.4.2 External Interfaces

4.4.2.1 User Interfaces

- **Scientist Interface:**
  This interface can be divided into 2 interfaces.
  - **Professor & Research Assistant Interface:** This interface allows professors/research assistants to interact with all functionalities with the system. All interactions are done via WEB GUI and they can conduct their experiments and do scientific studies from here. Also, they can also track their students within this interface.
    Design Rationale:
    - Each operation can be handled less than 5 button operation.
    - Design shall be simple to reduce misuse.
    - Interface shall contain an easy to use student management page.
  - **Student interface:** Student interface provides students a limited version of the system. For example, they cannot see all the activity history of the research group they belong. This interface is more simplified and for a minimal use.
    Design Rationale:
    - Each operation can be handled less then 3 button operation.
    - Design shall be more simpler than professor interface

- **Researcher Interface:**
  In this interface, researchers can access all system data while system is up and running. This interface provides researchers to see the performance and usage of the system features by showing various charts and graphics. Researchers can run their own machine learning algorithms in this interface and then collect these data to improve the microscope in future. Researchers also can search for specific user's usage instead of the overall usage and hence they can analyse specific user's interaction to the system.
  Design Rationale:
  - Interface shall be supported on all browsers.
  - Pie, bar, line charts are supported.
  - Database access shall be read-only.

- **IT Staff Interface:** This interface allows it staff to do operations on scientists. They can directly add new users to the database, or they can existing ones from database. This interface operates on a web browser and to login this interface, admin should enter id and password. And then, they can do operations on users.
  Design Rationale:
  - Single paged interface, all functionality is in one place.
  - Reduced misuse with simple design.

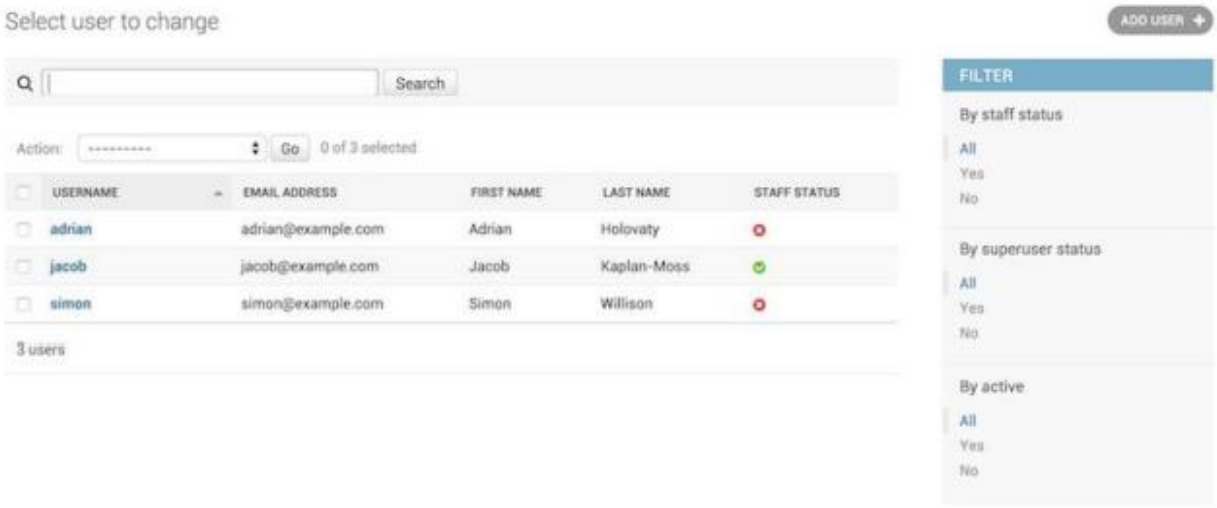

Figure 9: Admin Interface

4.4.2.2 System Interfaces

OpenFlexure Microscope has external systems which are done critical tasks for Web Server. In this subsection, detailed explanation of interfaces for external systems which is designed for communication is explained.

- **Interface between experiment handler and experiment processor system:**
  When the experiment handler gets the experiment data to be ran, it passes this request to experiment processor system and takes the report once process is finished, and then sends it back to the server manager.

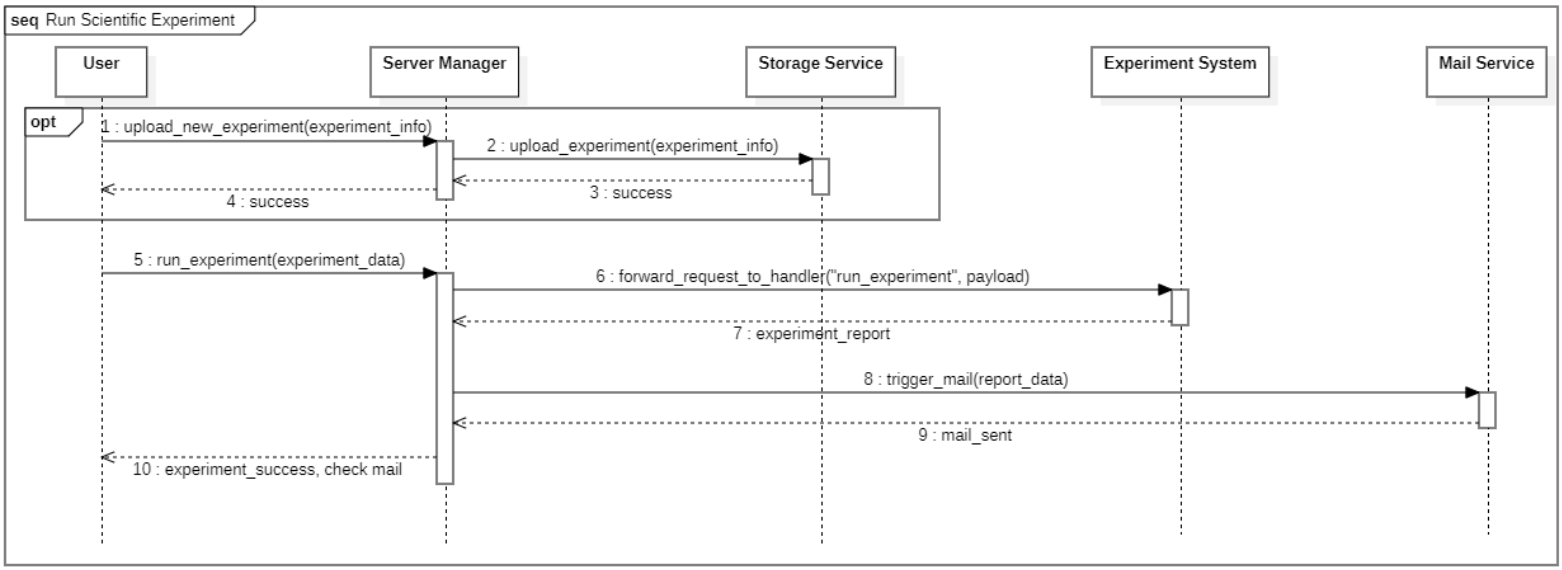

Figure 10: Interface between experiment handler and experiment processor system

Design Rationale:

- Experiment handler doesn't stores any user or scientific group related data while running the experiment for security reasons.

❖ **Interface between server manager and mail service:**
The mail service is one of the crucial external services is the subsystem. For instance, it may take place in a scenario where a scientist creates a scientific group and invites the other scientists within this process. This interface allows server manager to send different types of emails to end users.

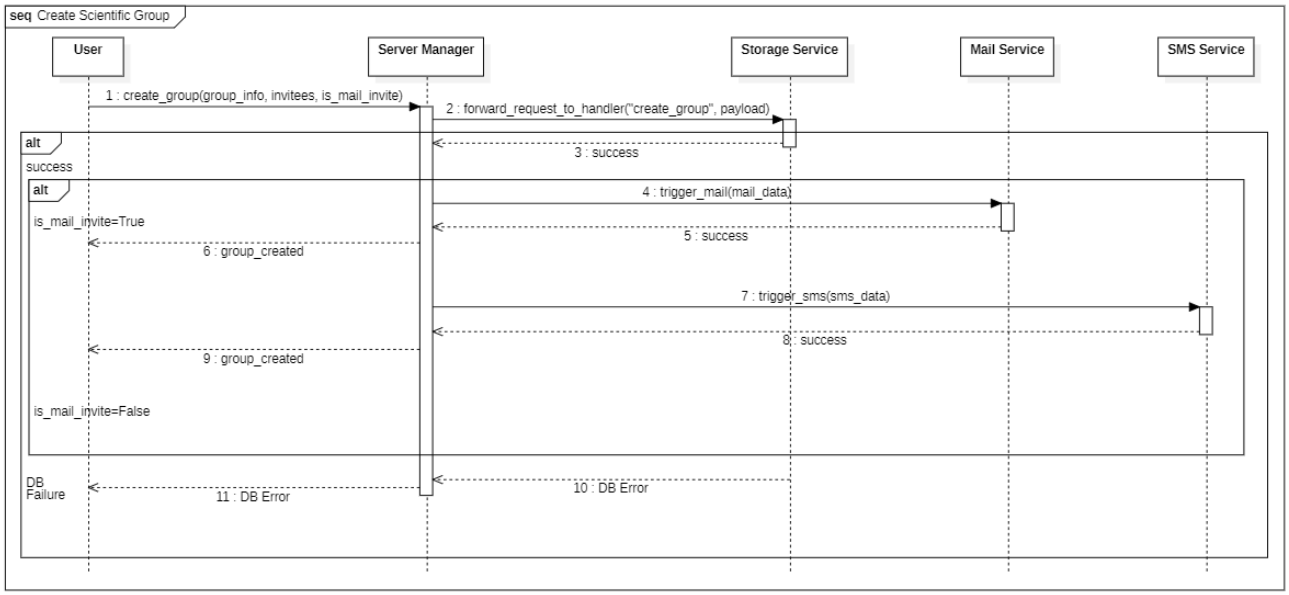

Figure 11: Interface between Server Manager and Mail Service

Design Rationale:

- Server manager shall check the mail_data format before sending request to the mail service
- In case of a failure(like db failure), logs should be updated and investigated by IT staff.

❖ **Interface between authentication handler and authenticator service:**
This interface not only using in authenticating/unauthenticating cases, but also using in setting privileges of new scientists/researchers by the admins. Auth handler gets all the authentication related data from storage and checks user authorization statuses

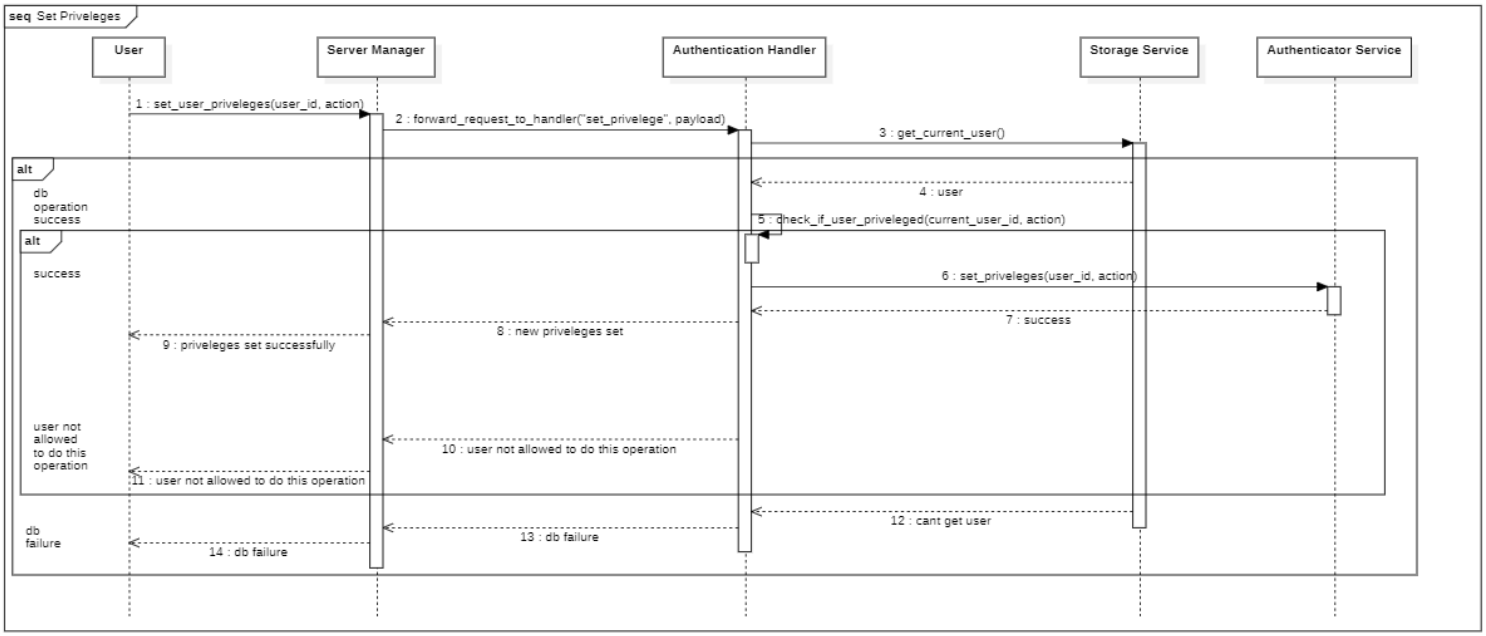

Figure 12: Interface between Authentication Handler and Authenticator Service

Design Rationale:

- All authdata must be kept secured inside auth handler with encryption protocols.
- IT Staff must be informed in case of an unauthorized access immediately.

❖ **Interface between server manager and sms service:**
The sms service is one of the crucial external services is the subsystem. For instance, it may take place in a scenario where a scientist creates a scientific group and invites the other scientists within this process. This interface allows server manager to send different types of SMSes to end users.

Design Rationale:

- Server manager shall check the sms_data format before sending request to the Amazon SNS service
- In case of a failure(like a connection failure to Amazon SNS), logs should be updated and investigated by IT staff.

❖ **Interface between data analysis handler and analytics system:**
Analytics system takes a part when the analytics request comes to the analysis handler. Analysis handler expects some analysis data and preprocesses it according to analytics system's needs, and then passes it to the external system. After that, analytics system creates and returns an analysis report.

Design Rationale:

- Communication between analytics system and handler is only alive when there is a data analysis request.
- In case of a failure(like a connection failure), logs should be updated and investigated by IT staff.

❖ **Interface between registration handler and registration system:**
Registration system takes a part in the registration process of the users. It accepts data in a specified format so there is a contract in the interface between registration handler and registration system. Once registration handler takes the information from server manager, it adjusts the format of the input data and send to the registration system to make user validated.

Design Rationale:

- Registration for the same user is blocked for 3 days once the user validation failed.
- Personal data, like identity information, shall not be stored in the system, it shall only be used for checking user detail validity on the fly.