Software Requirements Specification OpenFlexure Microscope

Ahmet Burak Baraklı, 2309730
Ozan Akın, 2309599

April 23, 2021

Contents

List of Figures

List of Tables

Revision History

Date	Reason For Changes	Version
15.04.2021	Initial Setup	0.1
23.04.2021	Final Version	1.0

Table 1: Revision History of Software Requirements Specification Document

1 Introduction

The document is a Software Specification Requirements (SRS) of a smart microscope called OpenFlexure, a microscope that is connected to the internet and can be controlled remotely. The OpenFlexure project's website can be found at `https://openflexure.org`.

1.1 Purpose of the System

The purpose of the project is to build a 3D-printed laboratory grade microscope that can be used in environments from Antarctica to Amazon Forests. With the motorized parts and easily printable parts of the microscope, the system can be easily be placed in areas with restricted access to decrease the risks of both patients' and researchers' health. Parts of the microscope can be seen in the figure below. Note that the figure includes both printable and non-printable parts.

Figure 1: Parts of an unassembled OpenFlexure Microscope [3]

For the sake of durability and stability of the inner parts where camera and electronic components, such as Arduino and Raspberry Pi, OFM system has stages that has a wide bottom. Since stages are printed with 3D-printers like most of the parts, the stage part can have various colors. In the following the stage part of several OpenFlexure Microscopes can be seen:

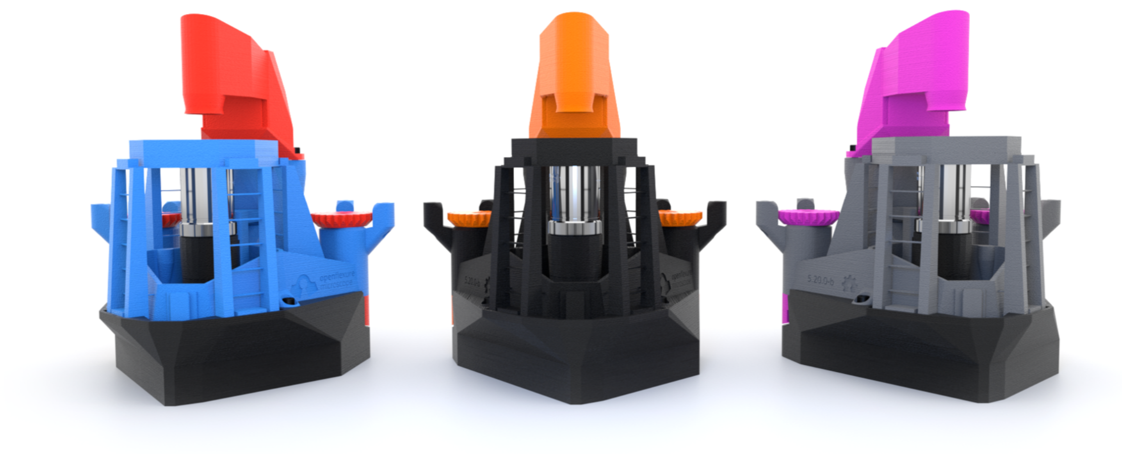

Figure 2: The stage part of OpenFlexure Microscope [5]

1.2 Scope

The project is called OpenFlexure, a 3D printed, remotely controlled microscope. The researchers will be able to watch and record the live stream coming from the microscope. Researchers will be able to change the microscope's location and field of view with motorized parts using two programmable electronic boards, an Arduino to control the motors and a Raspberry Pi to connect to the internet and communicate with the researchers. The camera part of an OpenFlexure Microscope that is connected to motors can be seen in the figure below.

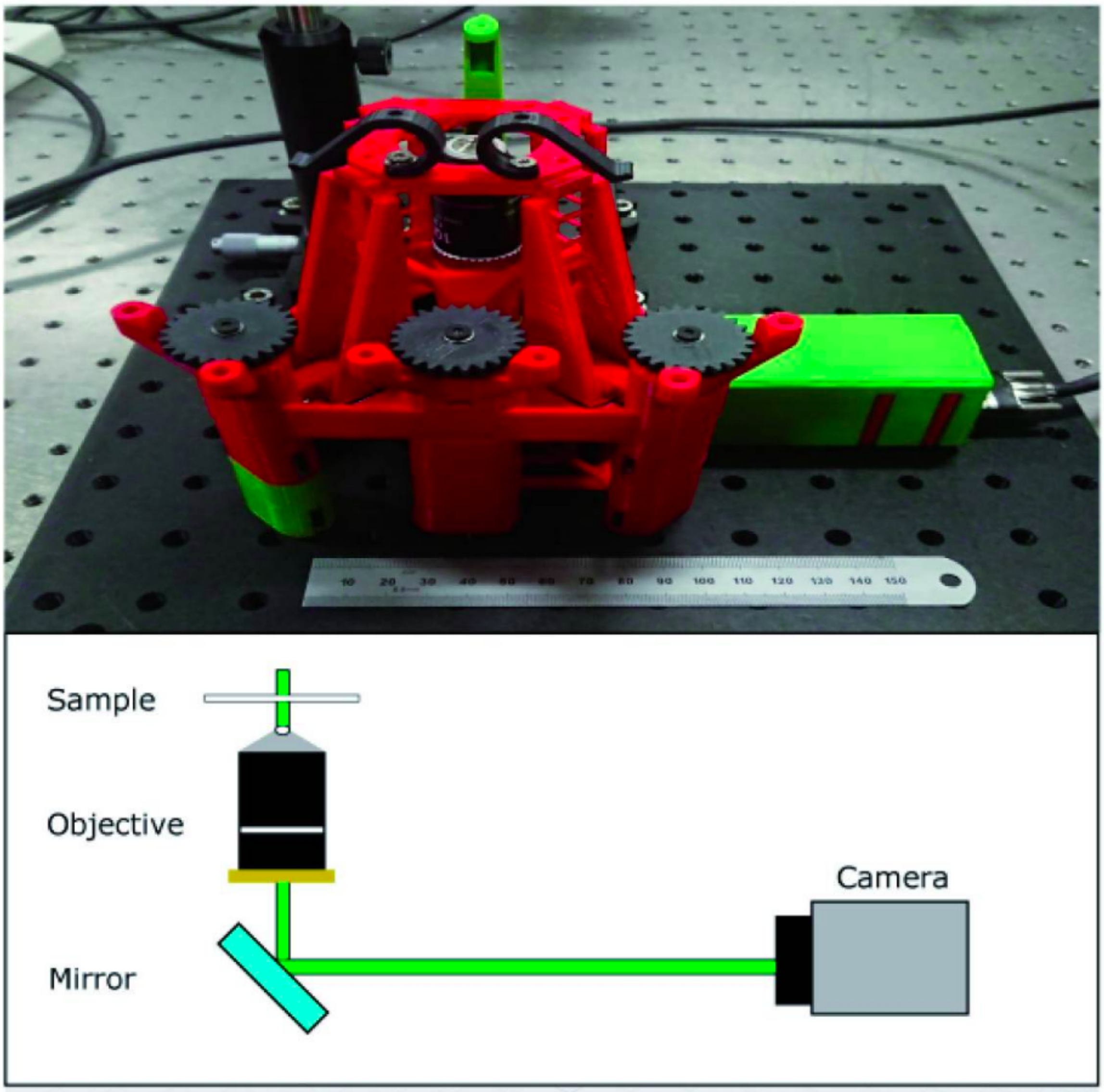

Figure 3: Camera part of an OpenFlexure Microscope [2]

The scope of the project can be listed as follows:

- Providing a remotely controlled microscope environment to help researchers do their research without risking their health.
- Providing a recordable live stream to make it possible to examine the results by more than one person while creating a reproducible environment with the record options of the live stream.
- Providing a cloud-based architecture built-in to the microscope to help researchers combine their results, projects, and visuals.

1.3 System Overview

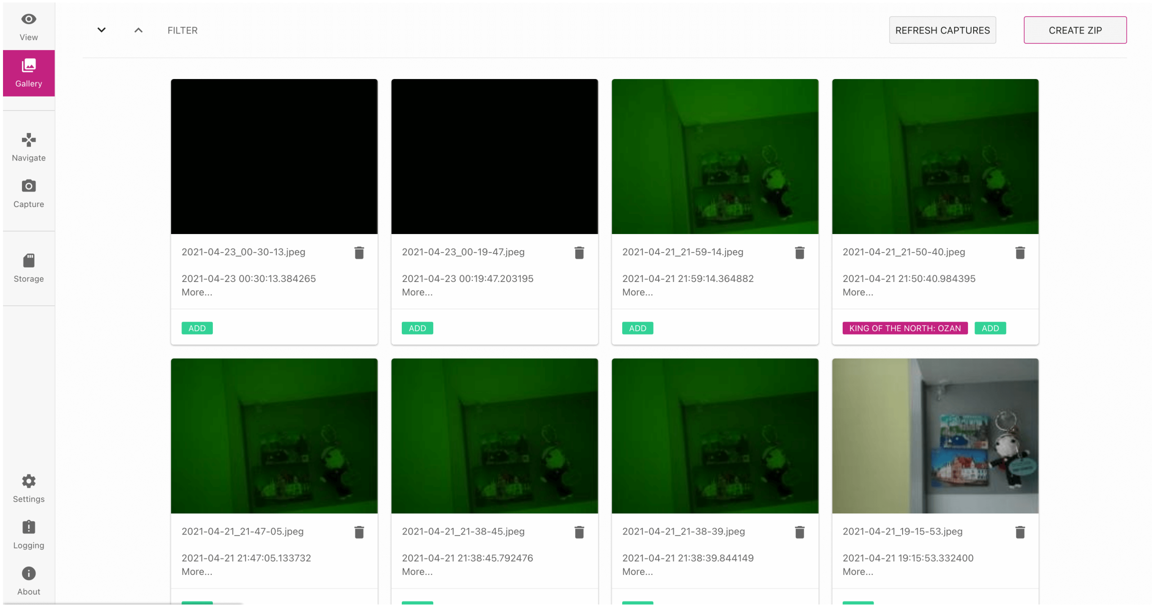

Figure 4: An example screenshot from the OpenFlexure Web Interface

This section of the SRS document provides information about the components and functions of the project.

1.3.1 System Perspective

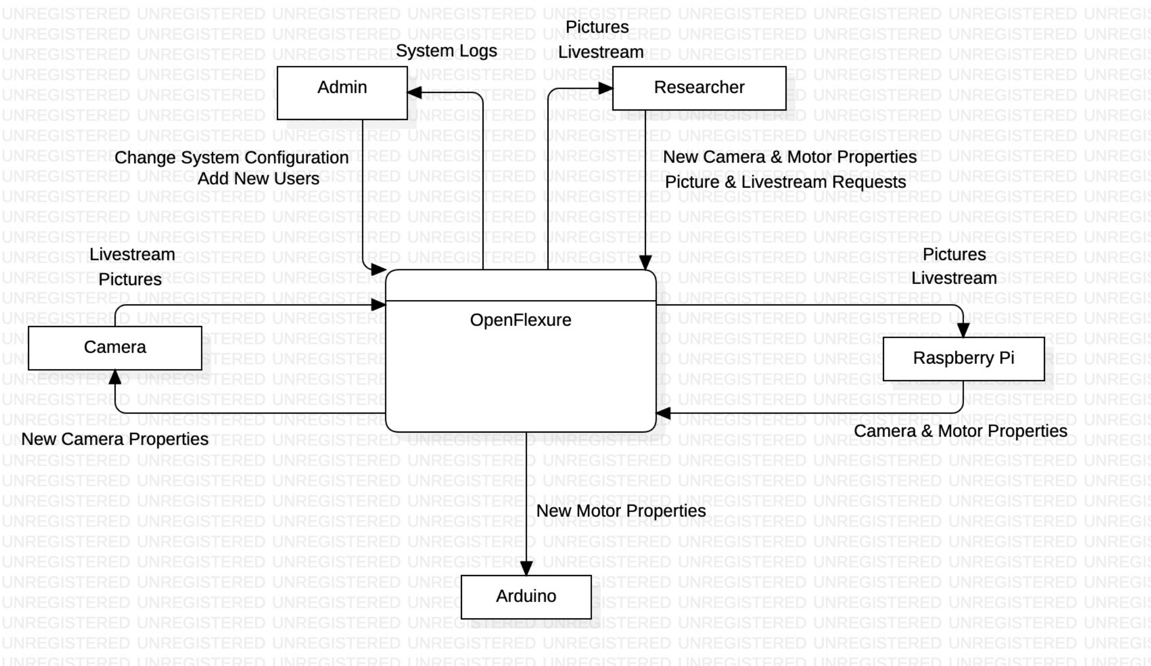

Figure 5: Context Diagram for OpenFlexure Microscope

OpenFlexure is a system that can be built in any environment that has electricity and an internet connection to work. The system includes its cloud architecture to make the system independent from any other cloud systems and available for everyone that has been granted access by the system [6].

The system of the project provides necessary tools and user interfaces to control the project remotely from anywhere in the world. Options such as recordable live streams, controllable motorized parts, and saving the system information with the live streams into storage such as SD Cards, make the system more wanted, desired, and necessary to everyone that works in areas such as restricted access areas or some areas with highly infectious viruses [6].

The project mainly consists of three parts to work with [1]:

- A microscope and a camera to catch the visuals.
- A highly advanced programmable electronic board called Raspberry Pi process the video coming from the camera and runs a server that is connected to the internet to serve videos, live streams and make the options and properties of the system available and configurable by researchers.
- A second programmable electronic board called Arduino to listen to events from Raspberry Pi and control the motorized parts of the system accordingly.

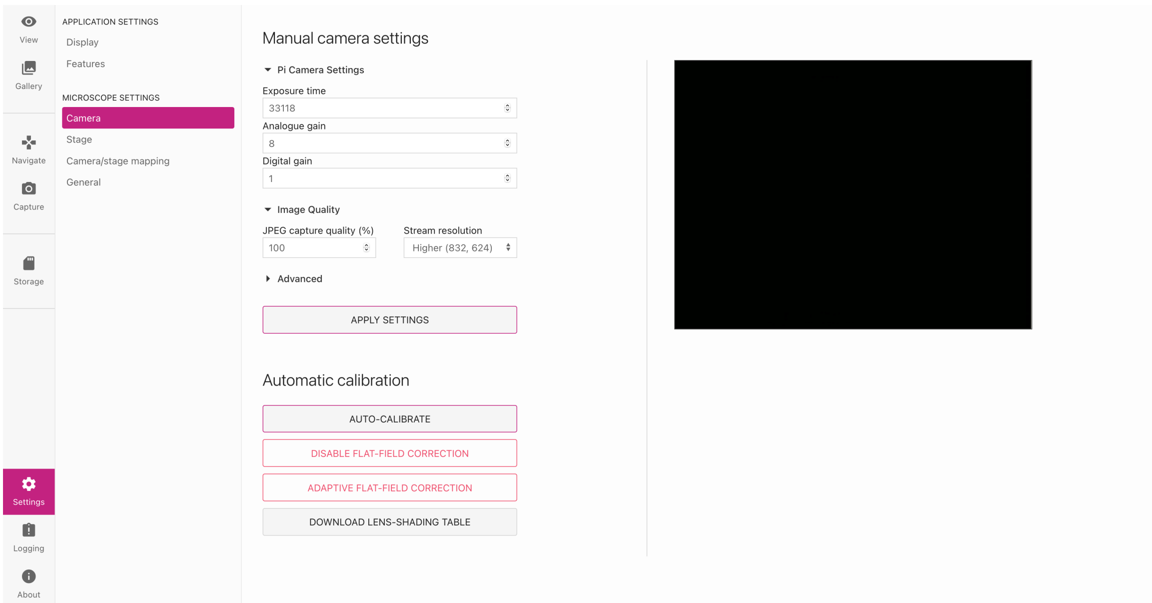

Figure 6: An example screenshot from the OpenFlexure Web Interface

1.3.2 System Functions

Function	**Summary**
Add Microscope	The system admin adds a microscope to the system for researchers to view.
Add User	The system admin adds an user to the system.

Add Plugin	When a system admin wants to add a plugin to the system, they can add a plugin.
Authorize Researcher to Microscope	When a system admin wants to authorize a researcher to see a microscope, they can add an authorization.
Change Angle of the Camera	When a researcher wants to change the angle of the camera to see from other perspectives, they can change the angle.
Change Position of the Camera	When a researcher wants to change the position of the camera to see from other perspectives, they can change the position.
Create Researcher	The system admin creates a researcher to the system.
Create System Admin	The super admin creates a system admin for the system.
Download Gallery	The researcher downloads the gallery.
Refresh Pictures	When the researchers in the gallery page, they can refresh the pictures in the gallery.
Remove Plugin	When a system admin wants to remove a plugin from the system, they can remove a plugin.
Reset the Position of the Camera	When a researcher wants reset the position of the camera, the camera position is being set to (0, 0, 0).
Reset Password	A system admin resets a researcher's password.
Resize Picture	The researcher can resize any picture.
Save Pictures to Gallery	A researcher wants to save a picture into the gallery to be able to reach it later.
Schedule a Task	The user creates a scheduled/recurring task to perform an action.
Select Theme for GUI	The user changes the theme of the user interface.
Start Auto-calibration	When the researcher wants to calibrate the camera, the system automatically calibrates the camera lens.
Start Auto-focus	When a researcher wants to auto-focus camera to make visuals more clear, they can trigger the action.
Start Recording	When a researcher wants to record a video from live stream and save to SD Card.
Stop Recording	When a researcher wants to stop recording the video from the live stream and encode the already saved SD Card Video.

Take a Picture	When a researcher sees an important image at microscope, they can send request to microscope to take picture of it.
Update a Property of the System	When a researcher wants to update some properties or configuration of the system, they change these values remotely.
View Gallery	A researcher wants to view pictures in a gallery.
View a Picture	A researcher wants to view previously saved picture.
View System Logs	The authorized system admin views the logs of the system.
Watch Livestream	When a researcher wants to see what is microscope is seeing, they can send request to microscope to stream the camera outputs.

Table 2: System Functions

1.3.3 User Characteristics

System users can be classified into three parts:

- **Researchers:** Researchers are the main users of the OFM system. They should have basic computer skills as they are required to interact with OFM GUI. Moreover, they can use the microscope system for any field they need.
- **System Admins:** A system admin is responsible for arranging database of the system, working over system errors, managing the back-end of the system, creating and enforcing security patches and viewing systems logs. They should have enough technical skills to make sure that OFM system runs as expected. They also have authority to change system-wide settings. In addition to that they can authenticate or deauthenticate the researchers.
- **Super Admin:** A super admin is a system admin that can authenticate or deauthenticate the system admins.

1.3.4 Limitations

- **Regulatory policies:** All information that given to the system or generated using the system is stored in the servers. Therefore, they must be preserved using Personal Data Protection Law (in Turkish ”Kişisel Verilerin Korunması Kanunu” or shortly KVKK).
- **Hardware limitations:** As server needs to respond quickly and process the data, it need to be enough to handle this operations. As for the camera, it must be able to serve high quality images since these outputs will be used in analysis. Motors should be durable.

- **Parallel operation:** The system should allow multiple researchers use system at the same time. It also should support multiple microscopes. Thus, the system must be able to process data from multiple microscopes and serve it into multiple researchers.

- **Audit functions:** The system should produce adequate and accurate logs to allow auditing work smoothly.

- **Control functions:** Changing and viewing system-wide configurations must be an privileged to system admins and super admins. So, they must bu inaccessible to unauthorized users.

- **Higher-order technology requirements:** The server and client side applications uses Electron, Python, Flask, JavaScript and Vue. Therefore, they are required.

- **Signal handshake protocols:** The communication between server and client side applications uses HTTPS protocol.

- **Criticality of the application:** If the system gives errors or starts to behave unexpectedly, the system admins can view the system logs in order to find out and solve the problem.

- **Safety and security considerations:** System admins are main responsible to keep the system and the information kept safe and secure. If an unauthorized access occurs, they can do whatever it takes to block this access.

1.4 Definitions

Researcher	A user who uses the microscope to analysis data.
System admin	The administrators of the system.
Super admin	A system admin that has control over other system admins.
Arduino	Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices.
Raspberry Pi	Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom.
SD card	Secure Digital, officially abbreviated as SD, is a proprietary non-volatile memory card format developed by the SD Association for use in portable devices

GUI	Graphical User Interface
OFM	OpenFlexure Microscope
Electron	Electron is an open-source software framework developed and maintained by GitHub. It allows for the development of desktop GUI applications using web technologies.
Node.js	Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.
GitHub	GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features.
Git	Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows.
Python	Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation.
Flask	Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions
JavaScript	JavaScript is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions
ECMAScript	ECMAScript is a general-purpose programming language, standardised by Ecma International according to the document ECMA-262
Vue	Vue.js is an open-source model–view–viewmodel front end JavaScript framework for building user interfaces and single-page applications
HTTPS	Hypertext Transfer Protocol Secure(HTTPS) is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using Transport Layer Security or, formerly, Secure Sockets Layer.

RAID	RAID is a data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for the purposes of data redundancy, performance improvement, or both.
VPN	The term virtual private network describes any technology that can encapsulate and transmit network data, typically Internet Protocol data, over another network. Such a system enables users to access network resources that may otherwise be inaccessible from the public internet.
Argon2	Argon2 is a key derivation function that was selected as the winner of the Password Hashing Competition in July 2015.

Table 3: Definitions

2 References

This document is prepared with respect to IEEE 29148 2018-11 standard:
29148 2018-11 - ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering.

References

[1] Joel T Collins et al. "Modern Microscopy with the Web of Things: The OpenFlexure Microscope Software Stack". In: *arXiv preprint arXiv:2101.00933* (2021).

[2] Stephen D Grant et al. "Adapting the 3D-printed Openflexure microscope enables computational super-resolution imaging". In: *F1000Research* 8 (2019).

[3] *https://wikifactory.com/@rwbowman/openflexure-microscope.* `https://wikifactory.com/@rwbowman/openflexure-microscope`. (Accessed on 04/22/2021).

[4] *Install the Software.* `https://openflexure.org/projects/microscope/install`. (Accessed on 04/22/2021).

[5] *OpenFlexure Microscope.* `https://openflexure.org/projects/microscope/`. (Accessed on 04/22/2021).

[6] Julian Stirling et al. "The OpenFlexure Project. The technical challenges of Co-Developing a microscope in the UK and Tanzania". In: *2020 IEEE Global Humanitarian Technology Conference (GHTC).* IEEE. 2020, pp. 1–4.

3 Specific Requirements

3.1 External Interfaces

The system admins who want to install OpenFlexure Microscope can easily install the Raspberry Pi OS based image from their website [4]. They need a Raspberry Pi to install the system. Then, the system admins can install the 3D schematics of the stages from the same website. After the 3D printed parts are ready, an Arduino is required to create the control motors in the system. After the motors, Arduino and the 3D printed parts assembled and necessary connections between the electronic parts made the Raspberry Pi system and the Arduino system can be integrated to each other. After the setup of the system, system admins let researchers to access the system from the control panel. System admins can introduce the microscope and stages to the system easily. Lastly, the system shows researchers a quick introduction of the system.

- **Administration Interface:** In this interface, system admins could setup and connect stages, microscope and camera together. With this interface, system can be connected to a local microscope and camera, or a remote one from the network. System admins can also create researchers and authorize researchers to access the microscopes.

- **Researcher Interface:** In this interface, researcher can view, record, capture and watch the stream from the camera of the microscope. Researchers can control the angle and the position of the camera. Moreover, the camera can be auto-focused and auto-calibrated using this interface. Researcher can view the gallery, download the recordings and captured pictures and control the system settings.

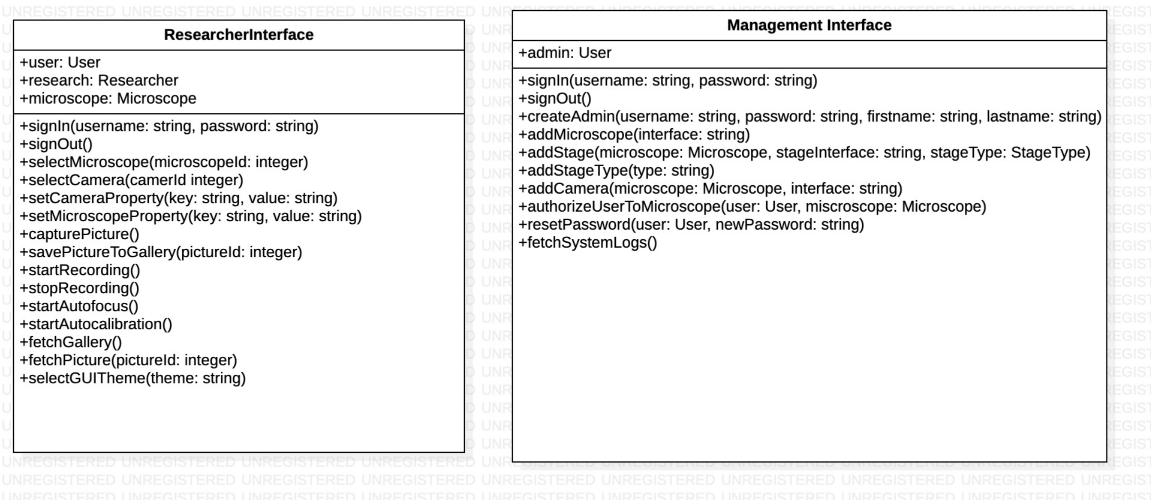

Figure 7: The Class Diagram of External Interfaces

3.2 Functions

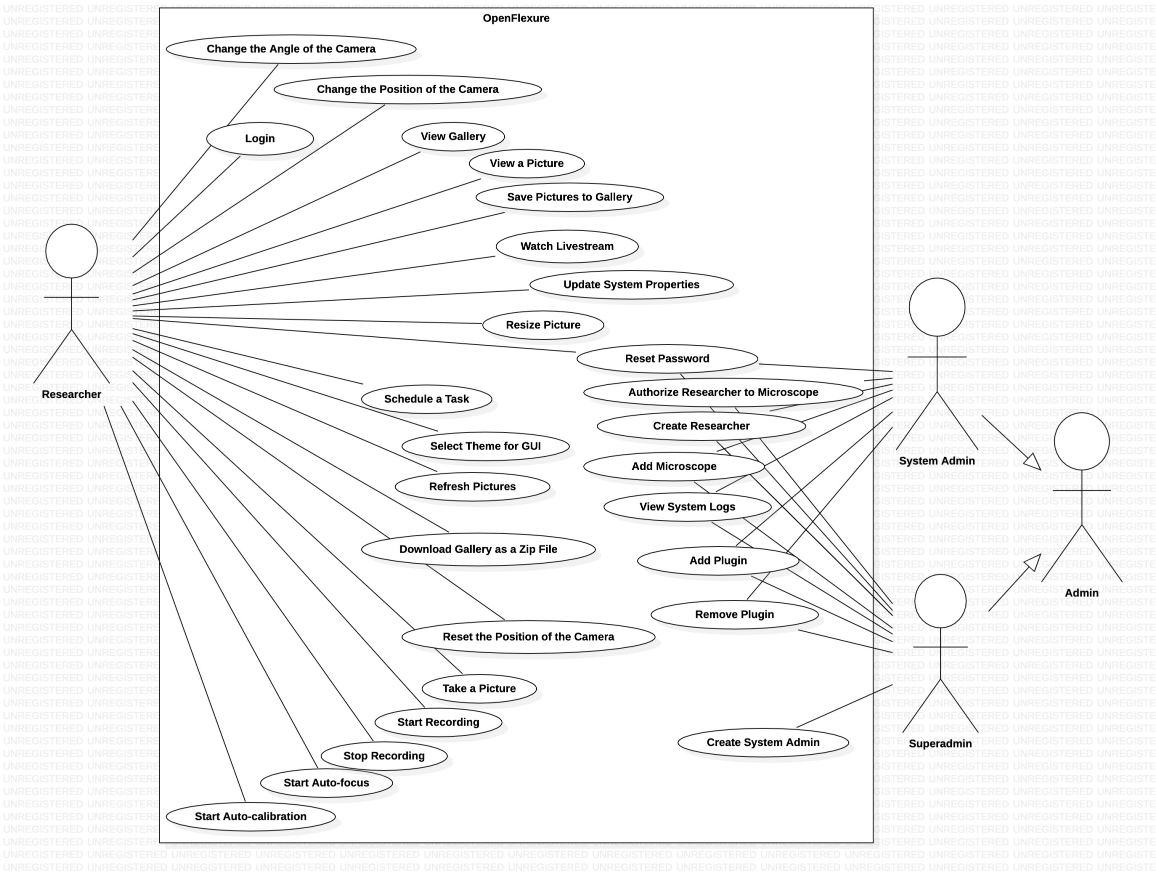

Figure 8: Use-Case Diagram for OpenFlexure Microscope

Use-Case Name	Change Angle of the Camera
Actors	Researcher
Description	When a researcher wants to change the angle of the camera to see from other perspectives, they can change the angle.
Data	The degree of the angle for the camera, the time that the request will be applied.
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the ”Change Angle” button.
Basic Flow	Step 1 - A researcher wants to change the angle to see from other perspective. Step 2 - The system receives the request about changing the angle. Step 3 - The system controls motors accordingly to apply the new angle. Step 4 - The user is informed about the change has been applied successfully.
Alternative Flow#1	Step 1 - A researcher wants to change the angle to see from other perspective in a future time. Step 2 - The system receives the request about changing the angle. Step 3 - The system sets a timer to the given timer. Step 4 - When given time is reached, the system controls motors accordingly to apply the new angle. Step 5 - The user is informed about the change has been applied successfully.
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The angle of the camera now changed.

Table 4: Change the Angle of the Camera

Use-Case Name	Change Position of the Camera
Actors	Researcher
Description	When a researcher wants to change the position of the camera to see from other perspectives, they can change the position.
Data	The x, y, z coordinates for the new camera location, the time that the request will be applied.
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the ”Change Position” button.
Basic Flow	Step 1 - A researcher wants to change the position to see from other perspective. Step 2 - The system receives the request about changing the position. Step 3 - The system controls motors accordingly to apply the new position. Step 4 - The user is informed about the change has been applied successfully.
Alternative Flow#1	Step 1 - A researcher wants to change the position to see from other perspective in a future time. Step 2 - The system receives the request about changing the position. Step 3 - The system sets a timer to the given timer. Step 4 - When given time is reached, the system controls motors accordingly to apply the new position. Step 5 - The user is informed about the change has been applied successfully.
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The position of the camera now changed.

Table 5: Change the Position of the Camera

Use-Case Name	Save Pictures to Gallery
Actors	Researcher
Description	A researcher wants to save a picture into the gallery to be able to reach it later.
Data	Filename of the picture to be saved into gallery, gallery which image will be saved.
Preconditions	The user must be authenticated.
Stimulus	When looking at a picture, researcher pressing the "Save Picture to Gallery" button.
Basic Flow	Step 1 - A researcher wants to save a picture into the gallery. Step 2 - The researcher send a request which includes name picture to the system. Step 3 - The system adds the picture into the gallery.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If there is no picture with that name, then the researcher receives an error.
Post Conditions	The picture must be in the gallery.

Table 6: Save Pictures to Gallery

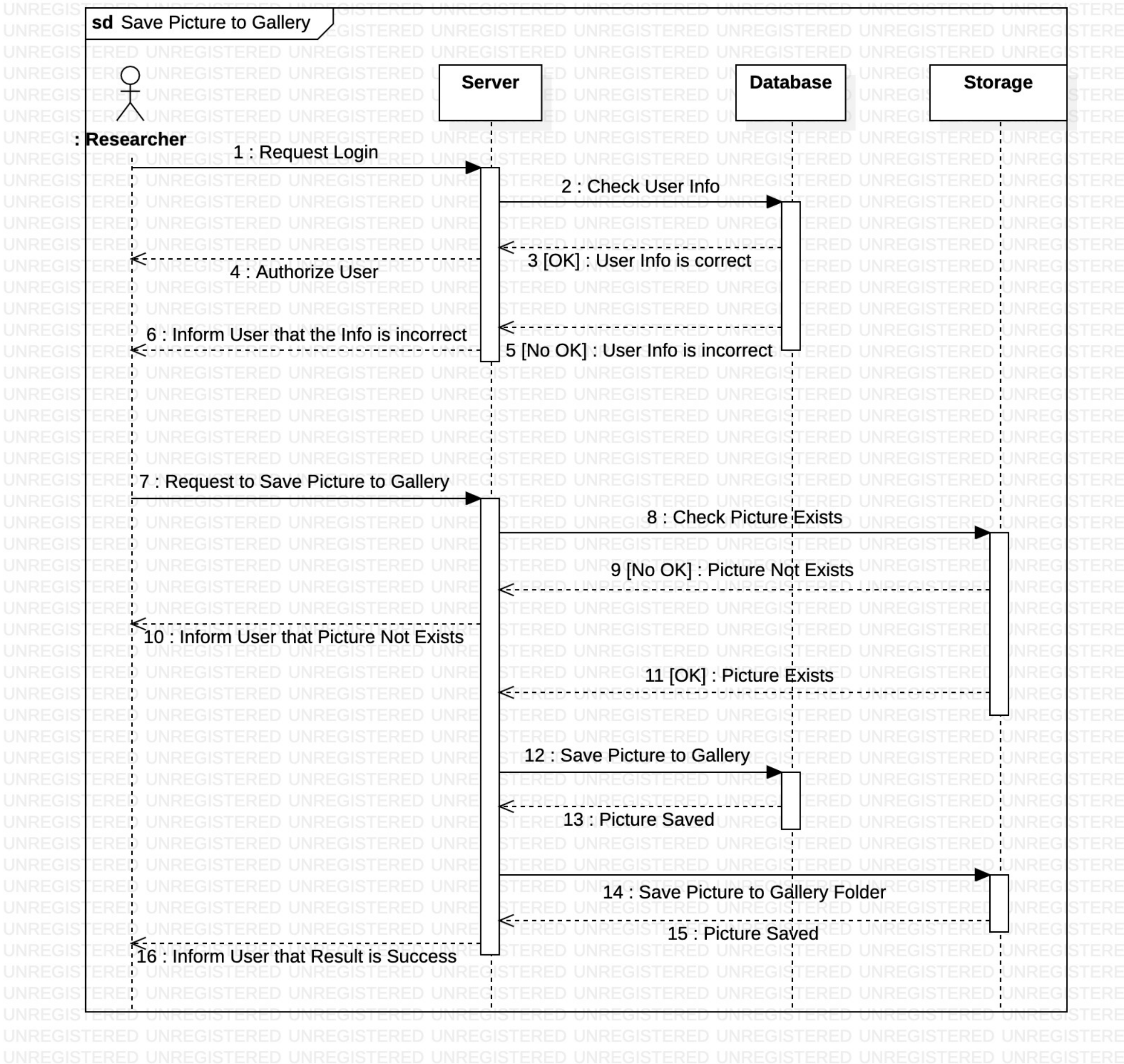

Figure 9: Save Pictures to Gallery Sequence Diagram

Use-Case Name	Start Auto-focus.
Actors	Researcher
Description	When a researcher wants to auto-focus camera to make visuals more clear, they can trigger the action.
Data	The auto-focus button action from the web server.
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the ”Automatic Focus” button.
Basic Flow	Step 1 - A researcher wants to calibrate the camera according to the environment. Step 2 - The researcher presses the ”Automatic Focus” button to start automatic focus. Step 3 - The system starts to the auto-focus based on the environment properties. Step 4 - The user is informed about camera has been focused successfully.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The camera is now focused and the visuals are much more clear.

Table 7: Start Auto-focus

Use-Case Name	Start Auto-calibration.
Actors	Researcher
Description	When the researcher wants to calibrate the camera, the system automatically calibrates the camera lens.
Data	The auto-calibrate button action from the web server.
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the ”Automatic calibration” button.
Basic Flow	Step 1 - A researcher wants to calibrate the camera according to the environment. Step 2 - The researcher presses the ”Automatic calibration” button to start automatic calibration. Step 3 - The system starts to the auto-calibration based on the environment properties. Step 4 - The user is informed about camera has been calibrated successfully.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The camera is now calibrated.

Table 8: Start Auto-calibration

Use-Case Name	Start Recording.
Actors	Researcher
Description	When a researcher wants to record a video from live stream and save to SD Card.
Data	Time that the "Start Recording" button is pressed.
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the "Record" button.
Basic Flow	Step 1 - A researcher wants to record a Video and save it to the SD Card. Step 2 - The researcher presses the "Record" button to start recording. Step 3 - The system starts to capture the images coming from the camera and saves them into the SD Card. Step 4 - The user is informed about a recording has started.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The system continues to record the content.

Table 9: Start Recording

Use-Case Name	Stop Recording.
Actors	Researcher
Description	When a researcher wants to stop recording the video from the live stream and encode the already saved SD Card Video.
Data	Time that the ”Stop Recording” button is pressed.
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the ”Stop Recording” button.
Basic Flow	Step 1 - A researcher wants to stop recording of a video. Step 2 - The researcher presses the ”Stop Recording” button to stop recording. Step 3 - The system receives the record button’s action. Step 4 - The system stops the recording. Step 5 - The system starts the encoding of the saved video to convert the record format to something more common such as MP4. Step 6 - The system saves the new encoded video to the SD Card.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The SD Card contains the recorded video.

Table 10: Stop Recording

Use-Case Name	Take a Picture
Actors	Researcher
Description	When a researcher sees an important image at microscope, they can send request to microscope to take picture of it.
Data	Time the button was pressed. Desired size of the Picture.
Preconditions	The user must be authenticated. The microscope and camera must be working.
Stimulus	Researcher pressing the "Take Picture" button.
Basic Flow	Step 1 - A researcher sees an important image at camera. Step 2 - The researcher send request to take picture. Step 3 - The system saves the image when the button was pressed into an image file in SD Card. Step 4 - The user sees the picture.
Alternative Flow#1	Step 4 - The picture file is saved into USB instead.
Alternative Flow#2	Step 3 - The system saves the image when the button was pressed and resize it into desired size an save it into an image file in SD Card.
Alternative Flow#3	–
Exception Flow	–
Post Conditions	The picture which contains the data when researcher pressed the button must be saved.

Table 11: Take a Picture

Use-Case Name	Add Plugin
Actors	System Admin
Description	When a system admin wants to add a plugin to the system, they can add a plugin.
Data	Zip file of the plugin added.
Preconditions	The user must be authenticated. Also, the user must be a superadmin or admin.
Stimulus	The System Admin presses the "Add the Plugin" button.
Basic Flow	Step 1 - The system admin presses the "Add Plugin" button. Step 2 - A popup is shown to admin. Step 3 - The system admin selects which plugin will be added to the system. Step 4 - The system admin adds the plugin by uploading the zip file and presses the "Save" button. Step 5 - The system adds the uploaded plugin.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	-
Post Conditions	The uploaded plugin now is added to the system.

Table 12: Add a Plugin

Use-Case Name	Authorize Researcher to Microscope
Actors	System Admin
Description	When a system admin wants to authorize a researcher to see a microscope, they can add an authorization.
Data	The researcher to be authorized and the microscope will be seen by the researcher.
Preconditions	The user must be authenticated. Also, the user must be an admin.
Stimulus	The System Admin presses the "Authorize Researcher to Microscope" button.
Basic Flow	Step 1 - The system admin presses the "Authorize Researcher to Microscope" button. Step 2 - A popup is shown to admin. Step 3 - The system admin selects the researcher to be authorized and the microscope will be seen by the researcher. Step 4 - The system saves the new authorization.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	-
Post Conditions	The researcher is authorized to see the given microscope.

Table 13: Authorize Researcher to Microscope

Use-Case Name	Add Microscope
Actors	System Admin
Description	The system admin adds a microscope to the system for researchers to view.
Data	Microscope Data
Preconditions	The user must be authenticated. Also, the user must be an admin
Stimulus	The system admin presses "Add Microscope" button.
Basic Flow	Step 1 - The system admin presses add microscope button. Step 2 - The system connects to given microscope. Step 3 - The system admin is informed about success of adding a microscope.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If given microscope data is invalid or can not be connected, an error is shown instead.
Post Conditions	The newly added microscope is listed in the list of microscopes.

Table 14: Add Microscope

Use-Case Name	Create Researcher
Actors	System Admin
Description	The system admin creates a researcher to the system.
Data	Researcher's Data
Preconditions	The user must be authenticated. Also, The user that creates researcher is system admin
Stimulus	The system admin presses "Add Researcher" button.
Basic Flow	Step 1 - The system admin presses add researcher button. Step 2 - The system creates a new researcher with given data. Step 3 - The system admin is informed about success of creating an user.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The newly created user can use the system.

Table 15: Create Researcher

Use-Case Name	Create System Admin
Actors	Super Admin
Description	The super admin creates a system admin for the system.
Data	System Admin's Data
Preconditions	The system admin must be authenticated user. The system admin must be a super admin.
Stimulus	The user that creates another system admin presses "Add System Admin" button.
Basic Flow	Step 1 - The superadmin pressed the "Add System Admin" button. Step 2 - A popup is shown to the super admin. Step 3 - The user enters the information for the new system admin. Step 4 - The system creates a new system admin with the given information. Step 5 - The super admin is informed with the new system admin's information.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The new system admin can change and configure the system.

Table 16: Create System Admin

Use-Case Name	Reset Password
Actors	Researcher
Description	A system admin resets a researcher's password.
Data	–
Preconditions	The researcher must prove ownership of the account.
Stimulus	The researcher contacts with a system admin to reset his/her password.
Basic Flow	Step 1 - The researcher request a system admin to reset the password. Step 2 - The system admin resets the password. Step 3 - The system admin sends the new password to the researcher.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	-
Post Conditions	The password of the researcher is reset.

Table 17: Reset Password

Use-Case Name	Update a Property of the System
Actors	Researcher
Description	When a researcher wants to update some properties or configuration of the system, they change these values remotely.
Data	The new value for the property.
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the "Change Property" button.
Basic Flow	Step 1 - A researcher wants to change a property of the system. Step 2 - The researcher send request to web server to change property. Step 3 - The system updates the property accordingly. Step 4 - The user is informed about the change has been applied successfully.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The property of the system is now changed.

Table 18: Update a Property of the System

Use-Case Name	View Gallery
Actors	Researcher
Description	A researcher wants to view pictures in a gallery.
Data	The gallery which will be viewed.
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the "View Gallery" button.
Basic Flow	Step 1 - A researcher wants to see pictures in a gallery. Step 2 - The system returns the gallery. Step 3 - The gallery is sent the researcher.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If there is no previously saved pictures, then the researcher receives gallery is empty warning.
Post Conditions	The researcher receives the gallery.

Table 19: View Gallery

Use-Case Name	View a Picture
Actors	Researcher
Description	A researcher wants to view previously saved picture.
Data	Filename of the picture
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the "View Picture" button.
Basic Flow	Step 1 - A researcher wants to see previously captured picture. Step 2 - The researcher send a request which includes name picture to the system. Step 3 - The system returns the picture. Step 4 - The picture is sent the researcher.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If there is no picture with that name, then the researcher receives an error.
Post Conditions	The researcher receives the picture.

Table 20: View a Picture

Use-Case Name	Watch Livestream
Actors	Researcher
Description	When a researcher wants to see what is microscope is seeing, they can send request to microscope to stream the camera outputs.
Data	–
Preconditions	The user must be authenticated.
Stimulus	Researcher pressing the ”Watch Livestream” button.
Basic Flow	Step 1 - A researcher wants to see what microscope is seeing. Step 2 - The researcher send request to watch livestream. Step 3 - The system sends the camera output with web server until it receives a stop request.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The researcher receives the livestream of camera output until they send a stop request.

Table 21: Watch Livestream

Use-Case Name	Create System Admin
Actors	Super Admin
Description	The super admin creates a system admin for the system.
Data	System Admin’s Data
Preconditions	The system admin must be authenticated user. The system admin must be a super admin.
Stimulus	The user that creates another system admin presses ”Add System Admin” button.
Basic Flow	Step 1 - The superadmin pressed the ”Add System Admin” button. Step 2 - A popup is shown to the super admin. Step 3 - The user enters the information for the new system admin. Step 4 - The system creates a new system admin with the given information. Step 5 - The super admin is informed with the new system admin’s information.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The new system admin can change and configure the system.

Table 22: Create System Admin

Use-Case Name	Download Gallery as a Zip File
Actors	Researcher
Description	The researcher downloads the gallery.
Data	Gallery
Preconditions	A gallery must be exists. The user must be authorized to see that gallery.
Stimulus	The researcher presses the "Download" button while viewing a gallery.
Basic Flow	Step 1 - The researcher presses download button while viewing a gallery. Step 2 - The system compresses and starts the streaming the gallery as a zip file from the storage. Step 3 - The researcher downloads the compressed gallery as a zip file.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	-
Post Conditions	The researcher has the gallery as a zip file.

Table 23: Download Gallery as a Zip File

Use-Case Name	Refresh Pictures
Actors	Researcher
Description	When the researchers in the gallery page, they can refresh the pictures in the gallery.
Data	Gallery
Preconditions	A gallery must be exists. The user must be authorized to see that gallery.
Stimulus	The user presses the "Refresh Pictures" button while viewing a gallery.
Basic Flow	Step 1 - The researcher presses the "Refresh Pictures" button while viewing a gallery. Step 2 - The system refetches the pictures from the gallery.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	-
Post Conditions	The pictures in the gallery view are now refreshed.

Table 24: Refresh Pictures

Use-Case Name	Remove Plugin
Actors	System Admin
Description	When a system admin wants to remove a plugin from the system, they can remove a plugin.
Data	The plugin that will be removed.
Preconditions	The user must be authenticated. Also, the user must be a superadmin or admin.
Stimulus	The System Admin presses the "Remove the Plugin" button.
Basic Flow	Step 1 - The system admin presses the "Remove Plugin" button. Step 2 - A popup is shown to admin. Step 3 - The system admin selects which plugin will be removed from the system. Step 4 - The system admin removes the plugin by clicking the "Save" button. Step 5 - The system removes the selected plugin.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	-
Post Conditions	The selected plugin now is being removed from the system.

Table 25: Remove a Plugin

Use-Case Name	Reset the Position of the Camera
Actors	Researcher
Description	When a researcher wants reset the position of the camera, the camera position is being set to (0, 0, 0).
Data	–
Preconditions	The user must be authenticated.
Stimulus	Researcher presses the ”Reset Position” button.
Basic Flow	Step 1 - A researcher wants to reset the position of the camera. Step 2 - The system receives the request about changing the position. Step 3 - The system controls motors accordingly to apply the new position. Step 4 - The researcher is informed about the change has been applied successfully.
Alternative Flow#1	Step 1 - A researcher wants to reset the position of the camera for a future time. Step 2 - The system receives the request about changing the position. Step 3 - The system sets a timer to the given timer. Step 4 - When given time is reached, the system controls motors accordingly to apply the new position. Step 5 - The researcher is informed about the change has been applied successfully.
Alternative Flow#2	–
Exception Flow	–
Post Conditions	The position of the camera now changed.

Table 26: Reset the Position of the Camera

Use-Case Name	Resize Picture
Actors	Researcher
Description	The researcher can resize any picture.
Data	The picture that will be resized, the width and height of the new picture size.
Preconditions	The researcher must be authenticated and has access to the gallery.
Stimulus	The user presses ”Resize” button while viewing a picture.
Basic Flow	Step 1 - The researcher presses the ”Resize” button while viewing a picture. Step 2 - A popup is shown to researcher. Step 3 - The researcher enters the width and height of the new picture size. Step 4 - The system resizes the image with the given width and height.
Alternative Flow#1	Step 3 - The researcher selects a pre-defined width and height. Step 4 - The system resized the image with the given width and height.
Alternative Flow#2	–
Exception Flow	-
Post Conditions	The picture is now being resized.

Table 27: Resize Picture

Use-Case Name	Schedule a Task
Actors	Researcher or System Admin
Description	The user creates a scheduled/recurring task to perform an action.
Data	The task type, action type, schedule time, and the recurring status.
Preconditions	The user must be authenticated.
Stimulus	The user presses "Schedule a Task" button.
Basic Flow	Step 1 - The user presses "Schedule a Task" button. Step 2 - A popup is shown to user. Step 3 - The user selects the type of the task. Step 4 - The user selects the action of the task. Step 5 - The user selects the time which the action will run. Step 6 - The user selects the recurring status of the task. If the task is a recurring task, then the user enters the recurring schedule. Step 7 - The user saves the task by clicking the "Save" button.
Alternative Flow#1	Step 3 - The user selects a pre-defined schemas for the tasks. Step 4 - The user presses the "Save" button to save the task.
Alternative Flow#2	–
Exception Flow	-
Post Conditions	A scheduled/recurring task is now being set and ready to execute.

Table 28: Schedule a Task

Use-Case Name	Select Theme for GUI
Actors	Researcher or System Admin
Description	The user changes the theme of the user interface.
Data	The selected user interface, Dark or Light.
Preconditions	The user must be an authenticated user.
Stimulus	The user presses "Select Theme" button and selects the preferred theme, Light or Dark.
Basic Flow	Step 1 - The user presses the "Select Theme" button. Step 2 - A dropdown is shown to user to select dark or light theme Step 3 - The user selects theme wanted. Step 4 - The system changes theme to the theme selected.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	-
Post Conditions	The theme of the user interface has changed to what user selected.

Table 29: Select Theme for GUI

Use-Case Name	View System Logs
Actors	System Admin
Description	The authorized system admin views the logs of the system.
Data	System Logs
Preconditions	The user must be an authenticated user. Also, the user is an system admin or super admin.
Stimulus	The user presses the "View Logs" button from the left panel.
Basic Flow	Step 1 - The user presses the "View Logs" button. Step 2 - The system logs retrieved from the database. Step 3 - The user sees the logs in the chronological order.
Alternative Flow#1	Step 4 - The user downloads logs from the user interface.
Alternative Flow#2	Step 3 - The user filters the logs with their log type. Step 4 - The user sees the logs in the chronological order.
Exception Flow	Step 2 - The system logs cannot be retrieved from the database. Step 3 - A popup is shown in the user interface with the title "The logs cannot be retrieved from the database".
Post Conditions	The authorized admin has the logs of the system.

Table 30: View System Logs

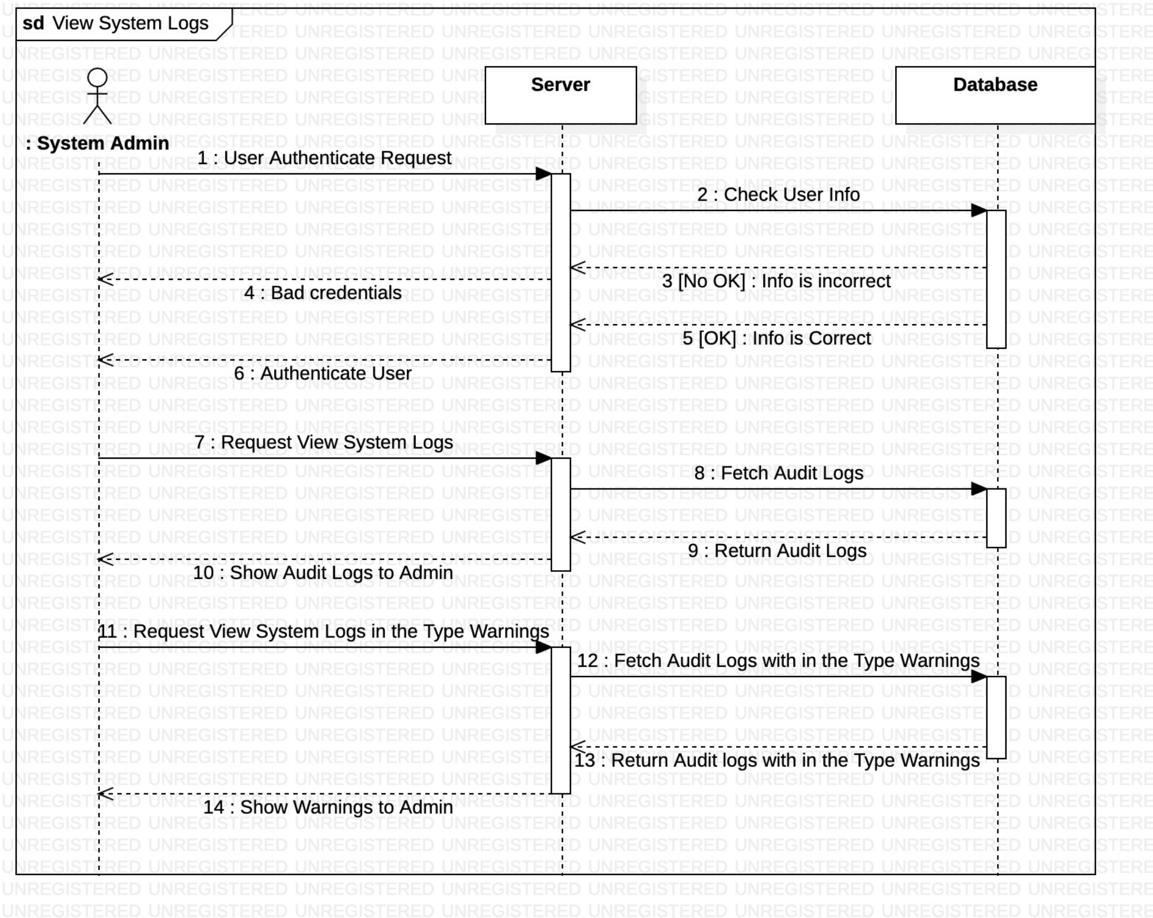

Figure 10: View System Logs Sequence Diagram

3.3 Usability Requirements

1. When the users open the application first time, a tutorial is required to show how to use the system properly.
2. The user interfaces should have dark-mode features if users prefer it.
3. The user interfaces should have automatic dark mode detection for operating systems to set dark-mode on or off at the initial setup.
4. The website and the Electron app should ask user feedback about features and usability of the system.
5. The system have to check if the user online or not and notify the user if the connection between the user interface and the server lost.
6. The system have to warn user if the storage usage is above 80%.
7. The system has to be connection tolerant. It means that if the internet connection goes down for the web server, the system have to keep doing its own tasks and obey the scheduled program.
8. The system has to warn user if the user wants to exit from the system while an ongoing task exists.

3.4 Performance Requirements

1. The initial opening of system have to take at most 2 seconds.
2. The actions done in the user interfaces have to reflect to the microscope at most 5 seconds delay.
3. The delay between live-stream and real-time video have to be at most 10 seconds.
4. The internet connection between the microscope and cloud have to be at least 100 Mbps and the latency have to be at most 50 milliseconds.
5. The storage capacity of the server have to be big enough to hold the images of the microscope visuals. The system have to warn user if the storage usage is above 80%.
6. The network system of the place where the microscope is has to be updated and have fiber optic cable system.
7. The place where the microscope is must have a predictable temperature. If the temperature is too dynamic then a automatic and scalable cooling system based on the current temperature have to exists.
8. The disk speed where the web server runs has to be fast enough to handle high disk usages, such as writing high quality videos.
9. The power supply of the system has to be stable. If a stable energy cannot be found, then a energy stabilizer for the web server and microscope is required.

10. The motors that in used for the microscope must be stable and durable, i.e. the speed of the motor must not be changed from time to time. Regular check and replacement of the motors are required.

3.5 Logical Database Requirements

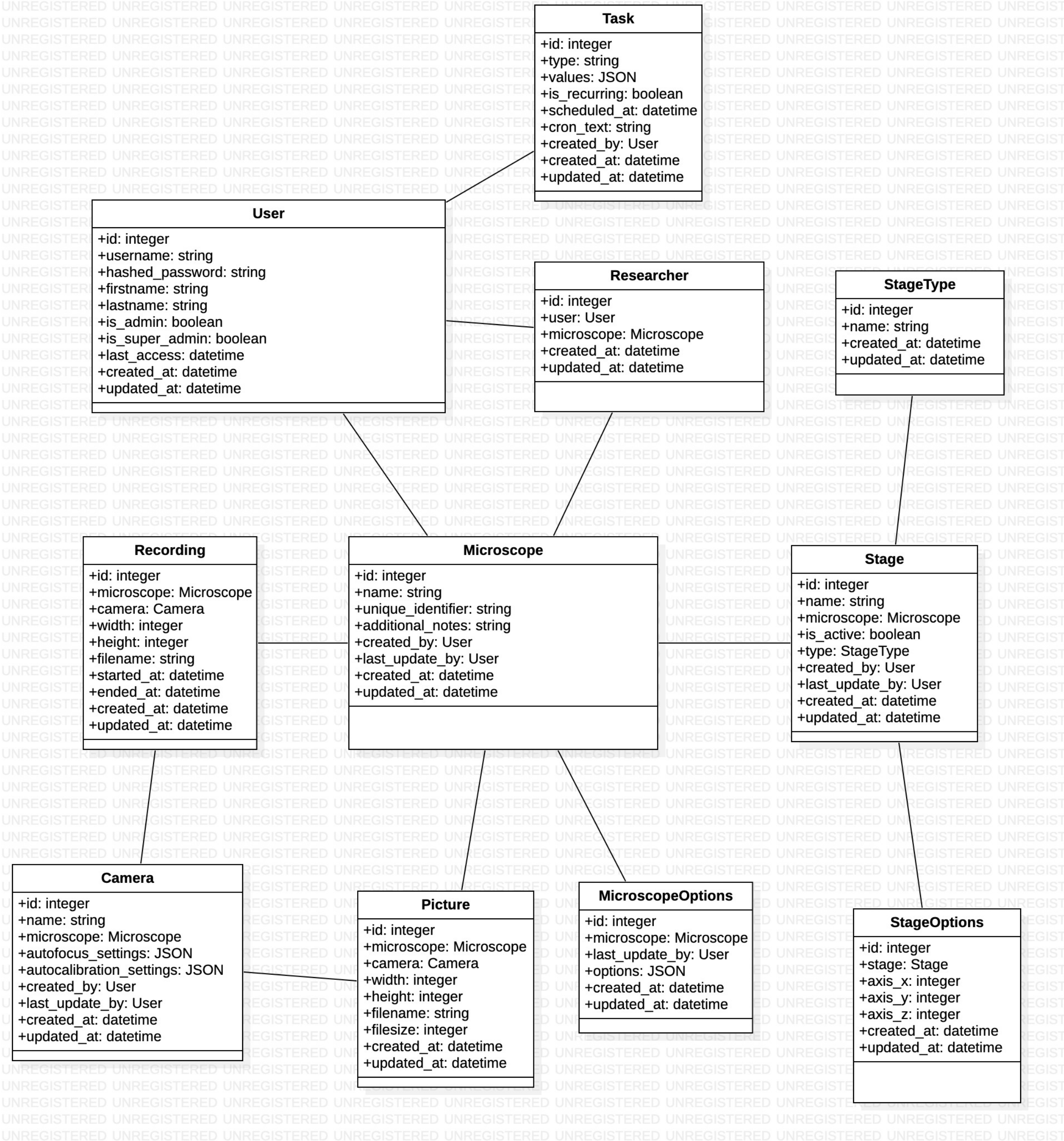

Figure 11: The Entity-Relationship Model of the Database of OFM

1. The database itself at first must be only accessible by the Super Admin.

2. The System Admin can define new users to access to database for Maintenance purposes.

3. Only defined users can access to the system.

4. All System Admins can define new users to access to the system.

5. Every user has a password to access the system. This password can be changed by the owner. In case of a password lost, the system admins can reset the user password.

6. The tables `tables`, `audit_logs`, `users`, `microscopes`, and `stage_types` are strong entities since they do not need any other entity to exist.

7. The table `microscope_options`, `stages`, `cameras`, `microscope_users`, `recordings` and `pictures` are weak entities since they need a microscope to exist.

8. The table `stage_options`, is a weak entity since it needs a *stage* to exist.

9. An *audit log* is a system database table that the entries are automatically added to the database.

10. The `hashed_password` field in the `users` table is the Argon2 hashed passwords of the raw versions.

11. A *user* can be identified as both a *regular admin*, or a *super admin*. Both *regular admins* and *super admins* can be called as *admins*.

12. A *user* can create infinite number of *microscopes*, *cameras*, and *stages*.

13. A *stage_type* can only be created, updated or deleted by an admin.

14. A *camera* can have infinite number of *stages* and *cameras*. However, only one *stage* can be active. Thus, the `is_active` property of a *stage* must be configured properly.

15. A *microscope* can only have one *microscope_option*.

16. A *microscope* can have infinite number of *microscope_user*.

17. A *stage* can only have one *stage_options*.

18. New users must be added to `users` table after they've added to the system. Also, fields `is_admin` and `is_super_admin` must be set properly.

19. New microscopes must be added to `microscopes` table after they've added to the system. Also, the `microscope_options` table entry must be created at the same time.

20. New stages must be added to `stages` table with their *microscopes* after they've added to the system. Also, the `stage_options` table entry must be created at the same time.

21. New cameras must be added to `cameras` table with their *microscopes* after they've added to the system.

22. New microscope users must be added to the `microscope_users` table with their *microscope* after they've added to the system.

23. New recordings can be created, updated, or deleted by the *creator user* of the *microscope*, any *microscope_user* of the same *microscope*, or any *admin*.

24. A *camera* can only be added by the *creator user* of the *microscope*, or by an *admin*.

25. A *stage* can only be added by the *creator user* of the *microscope*, or by an *admin*.

26. A *stage_option* can be updated by any *microscope_user* with belong to the same *microscope*.

27. The *creator user* of the *microscope* and the *admins* can add any number of users to the *microscope*.

28. The *creator user* of the *microscope*, the *microscope_users*, and the *admins* can create, update or delete any number of *pictures*.

29. A *task* can be created by any user, within the user's permissions.

30. The fields `axis_x`, `axis_y`, and `axis_z` in the table `stage_options` must be in the range of 0 and 360.

31. The fields `width` and `height` in the tables `recordings` and `pictures` must be greater than 0.

32. The database should be backed up every week in a different storage in case of a failure.

3.6 Design Constraints

1. All personal data of the `users` and all `audit logs` must be kept in accordance with the General Data Protection Regulation and KVKK.

2. The `users`, `pictures`, and `recordings` tables are confidential, so these table must be kept in very careful.

3. The storage that holds the recordings and pictures from the system is very confidential, so the storage must be kept with very careful precautions. Disk level encryption is necessary to make sure that data is safe.

4. According to the KVKK, a user can request to delete all marks of their existence in the system. In case of any user wants to delete all data in the system, this must be an allowed behaviour that is something necessary.

5. The IP address, the `user agents`, and the operating system info is confidential, and must not be kept unless it is necessary.

6. The database and the storage that holds recordings and pictures must be backed up regularly, i.e. once a week. This backup location must be a different disk rather than OFM storage. Cloud backup in this can be an option.

3.7 Software System Attributes

3.7.1 Reliability

1. The system must be durable and stable before using in the real life. The probability of having an error must not exceed 0.001%.

2. The SD Card used in the system must be durable and stable. The data loss probability of the SD Card must be selected as low as possible. The health of the SD Card must be checked regularly to not to have a data loss failure. Moreover, RAID 1 setup can be selected as primary storage option in case of data loss is important.

3. The electronic parts of the system should be checked regularly in case of a electronic shortcut or a failure. Regular tests for the Arduino and the Raspberry Pi should have be applied.

4. The regulators in the Raspberry Pi and the Arduino are the most sensible parts in the electronic parts. The regulators on them must be checked separately and more regularly then the others.

5. The system shall be tested if any component or module gets changed, updated, upgraded or if a new component gets installed.

3.7.2 Availability

1. The system must be running and operational during the research hours. This includes not just working hours, but also nights and weekends.

2. In case of a software failure, the system must restart itself in less than 5 minutes. Also, a notification must be sent to the admins in case of a restart.

3. In case of a hardware failure, the system must notify the system admins in order to replace parts and fix them.

4. The system must notify the admins in case of the CPU or RAM usage is above 80%.

5. At least one system admins must be operational and accessible in case of a system failure. The system admins can work remotely.

6. Any maintenance must be held in a time that the researchers that does not use the OFM. The maintenance hours that is no more than 3 hours must be told the researchers at least one week before. If the maintenance time required is more than 3 hours, then at least 15 days is required.

3.7.3 Security

1. The network where the OFM runs must be secure, and not publicly accessible. The researches must have a dedicated VPN system in order to connect the OFM remotely.

2. The user password hashes must be kept in Argon2 format, one of the state of the art password hashing algorithm. The password hashing algorithms must be checked in every five years and updated to a new one if that is necessary.

3. The SD Cards must have disk encryption in case of a robbery.

3.7.4 Maintainability

1. The development process must have a version control system, i.e. Git. Also, a well documented setup guide, maintenance guide etc. is required.

2. The system must be updatable remotely in case of a software bug or security problem. In every 3 months a new update must be released in order to keep system packages as updated. Also, the system must lock itself and wait for the upgrade if the system is not updated in 12 months.

3.7.5 Portability

1. The system must be available for at least last 4 years Raspberry Pi Operating System (OS) releases.

3.7.6 Usability

1. The system must be easily usable by anyone. The researcher have not to know about software architecture or programming. The tutorials and the user interface must be explanatory enough for everyone to understand them.

3.8 Supporting Information

The OFM is a 3D-printed microscope which aims to address many of the issues in research and clinical microscopy. Its primary benefits over conventional optical microscopes are that it is low cost, open-source and customisable for a range of laboratory or field applications. OFM has been used in many continents, and has been used for outreach, teaching and academic research.