Software Requirements Specification
OpenFlexure Microscope

Ali Doğan, 2237261
Doğukan Öztürk, 2310423

Contents

List of Figures

List of Tables

Revision History

Date	Reason For Changes	Version
16.04.2021	Initial Setup	0.1
23.04.2021	Final Version	1.0

Table 1: Revision History of Software Requirements Specification Document

1 Introduction

1.1 Purpose Of the System

OpenFlexure Microscope (OFM) is an open-source, 3D-printed, and fully-automated laboratory microscope that aims to make high precision mechanical positioning available to anyone with a 3D printer. The purpose of this project is to present a scientific instrument that can be controlled using existing, cross-platform, language-independent, industry-supported standards that adapts the Web of Things approach. With this advancements, the microscope can be deployed around the world in a wide range of operating environments with much cheaper costs compared to most other commercial microscopes. OFM offers an accessible, partially automated alternative with sufficient optical performance to identify and quantify microscopic animals such as parasitic protozoa of the Malaria, in regions with restricted access to conventional microscopes.

1.2 Scope

In the scope of this system, microscopists from different fields with different use cases, are able to obtain a 3D-printed fully-automated laboratory microscope with a very low cost compared to conventional high-performance microscopes. The microscope is designed to enable low-volume manufacturing and maintenance by local personnel.
The embedded hardware components along with installed software enables remotely and locally connection in a way that users are able to control the camera, observe the objective in different angles, capture the images of the object, and deploy different customized operations such as tile-scanning and more.
The microscope is used in the fields of clinical application, school teaching, biology teaching labs, academic research, and more. Having customized operations enable the range of the usage of the microscope unpredictable.
Scope of the project can be listed as:

- The system shall be assembled and configured by the local personnel.
- The system shall use a web API (aplication programming interface) to enable remote control through internet protocol (IP) networks.
- Users shall connect to and control the microscope either remotely or locally.
- Users shall be able to use a desktop application that enables features like navigation, live preview, image capturing, adding plug-ins.
- Users shall be able to update the software of the microscope without any need of hardware change.
- Users of the system shall be able to develop additional functionality and entirely new imaging modes without having to re-implement the more complex instrument control code.

Followings are **not** in the scope of the project:

- The microscope is not certificated for medical usage. Hence, the microscope is not an *in-vitro* diagnostic device.

1.3 System Overview

1.3.1 System Perspective

OpenFlexure Microscope is not a part of a larger system. However, it interacts with other applications such as desktop application and web application. Users shall control the objective and the positioning via a user interface. The user requests are processed in the main server of the microscope. Main server handles the issues by communication with camera control management and motor control subsystems.

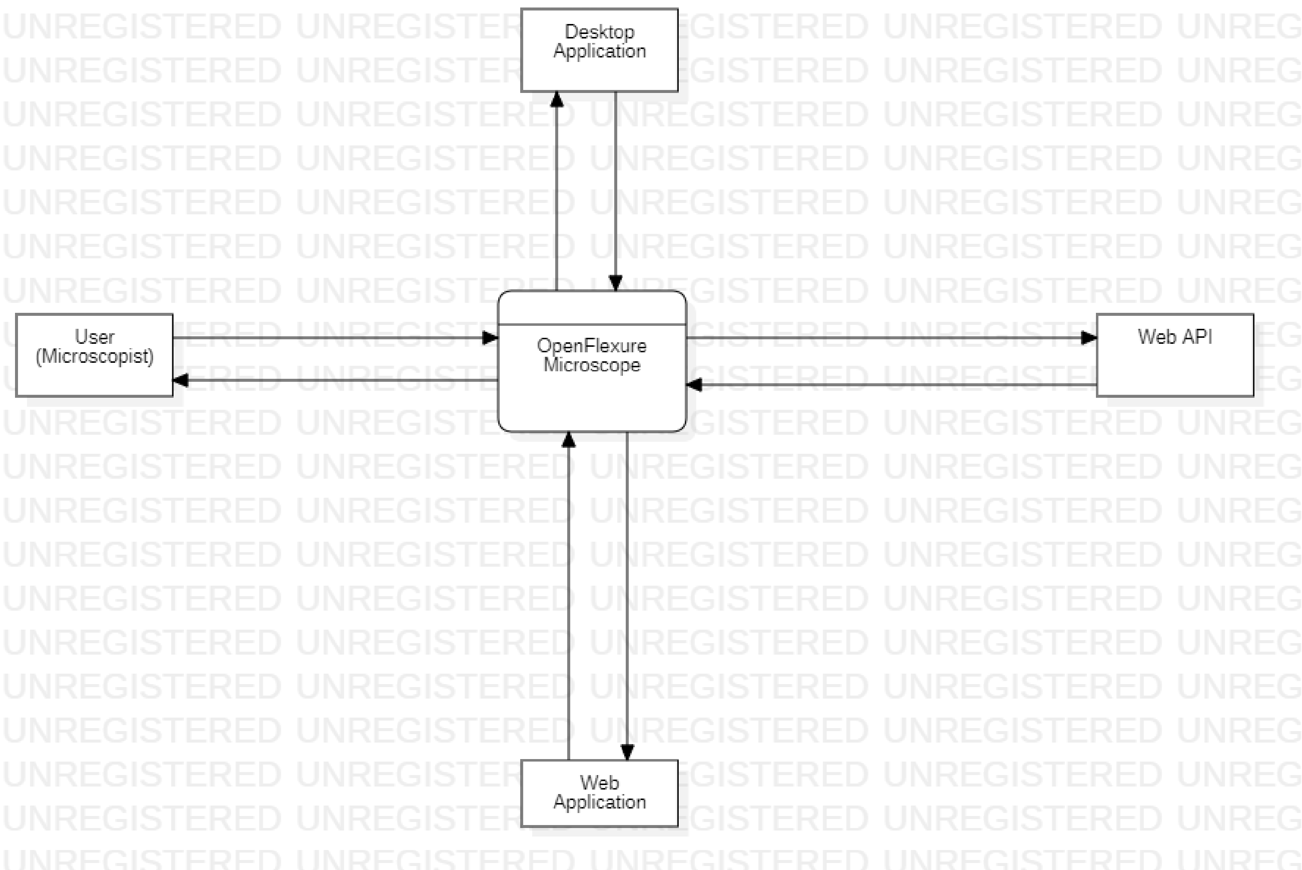

Figure 1: System Context Diagram for OFM

1.3.2 System Functions

1.3.3 User Characteristics

The target user of Open Flexure Microscope is mainly the microscopists who will use the microscope in their domain of interest.

Function	Summary
Setup server	User setups the server with already installed software from the SD disk.
Connect to microscope	User connects to the microscope either manually with peripheral I/O ports or remotely via Internet Protocol (IP) networks.
Control the microscope	User controls the microscope by navigating the camera, focusing and running scripted experiments.
Capture an image	User captures the current previewed image.
Browse the gallery of captured images	User browses the captured images from the gallery section of the application interface.
Filter the gallery	User filters the captured images according to their properties and metadata.
Download images	User downloads an image from the microscopes hardware storage remotely via HTTP API.
Change settings	User changes the settings of the microscope.
Upgrade server	User updates the main server using the main server kernel whenever there is internet connection.
Add plug-ins	User adds custom-plugins to the server for special custom use cases.

Table 2: System Functions

Users (microscopists) can vary from clinical practitioners to school teachers, from academic researcher to biology student. Users do not have to know programming, nor have a technical background. However, some additional features of the system is for users who are able to write automated experiment scripts with a technical background in couple of areas such as Internet Protocols (IPs), Web of Things and W3C standards, some scripting languages that are compatible with HTTP API's. Users should be able to update the microscope software with basic command line arguments as well.

1.3.4 Limitations

- **Regulatory policies:** The microscope is not certified as a medical usage.
- **Hardware limitations:** The hardware used in the microscope is an embedded system that has space restrictions for the portable usage which makes the hardware options limited.
- **Interface to other applications:** For the control and configuration of the microscope, an interface is needed.
- **Parallel operation :** Microscope must be able to process different tasks at the same time, however by limiting some cases. For example, navigating is not posible while tile-scan task is in process.
- **Audit functions :**There is no audit function in the system.
- **Control functions :** The control of the microscope is limited with the motor control system.

- **Higher-order language requirements :** There is no such limitation.
- **Signal handshake protocols :** All communication protocol between microscopes main server and client servers must meet WoT W3C standards.
- **Quality requirements :** There is no such limitation.
- **Criticality of the application :** There is no such limitation.
- **Safety and security considerations :** Due to the material used for the microscope, high temperature objects are not advised to be inspected.
- **Physical/mental considerations:** There is no such limitation.
- **Limitations that are sourced from other systems:** There is no such limitation.

1.4 Definitions

Term	Definition
API	Aplication Programming Interface
HTTP	Hyper-Text Transfer Protocol
WoT	Web of Things
Main Server	Main server is the embedded software inside the microscope. The main low-level code and simple functionalities of the microscope resides in the main server.
IP	Internet Protocol
GUI	Graphical User Interface

Table 3: Definitions

2 References

This document is written with respect to the specifications of the document below:

29148-2018 - ISO/IEC/IEEE International Standard - Systems and software engineering– Life cycle processes –Requirements engineering.

Other sources:

Jeretta Nord, Alex Koohang, and Joanna Paliszkiewicz. The internet of things: Review andtheoretical framework. *Expert Systems with Applications*, 133, 05 2019.

Toru Kawaguchi, Kazuo Kajimoto, Matthias Kovatsch, Michael Lagally, Ryuichi Matsukura, and Kunihiko Toumura. Web of things (wot) architecture. W3C recommendation, W3C, April 2020. https://www.w3.org/TR/2020/REC-wot-architecture-20200409/.

Collins, Joel & Knapper, Joe & Stirling, Julian & Mduda, Joram & Mkindi, Catherine & Mayagaya, Valeriana & Mwakajinga, Grace & Nyakyi, Paul & Sanga, Valerian & Carbery, Dave & White, Leah & Dale, Sara & Lim, Zhen Jieh & Baumberg, Jeremy & Cicuta, Pietro & McDermott, Samuel & Vodenicharski, Boyko & Bowman, Richard. (2019). Robotic microscopy for everyone: the Open-Flexure Microscope. 10.1101/861856.

Collins, Joel & Knapper, Joe & Stirling, Julian & McDermott, Samuel & Bowman, Richard. (2021). Modern Microscopy with the Web of Things: The OpenFlexure Microscope Software Stack.

Stirling, Julian & Sanga, Valerian & Nyakyi, Paul & Mwakajinga, Grace & Collins, Joel & Bumke, Kaspar & Knapper, Joe & Meng, Qingxin & McDermott, Samuel & Bowman, Richard. (2020). The OpenFlexure Project. The technical challenges of Co-Developing a microscope in the UK and Tanzania. 1-4. 10.1109/GHTC46280.2020.9342860.

3 Specific Requirements

3.1 External Interfaces

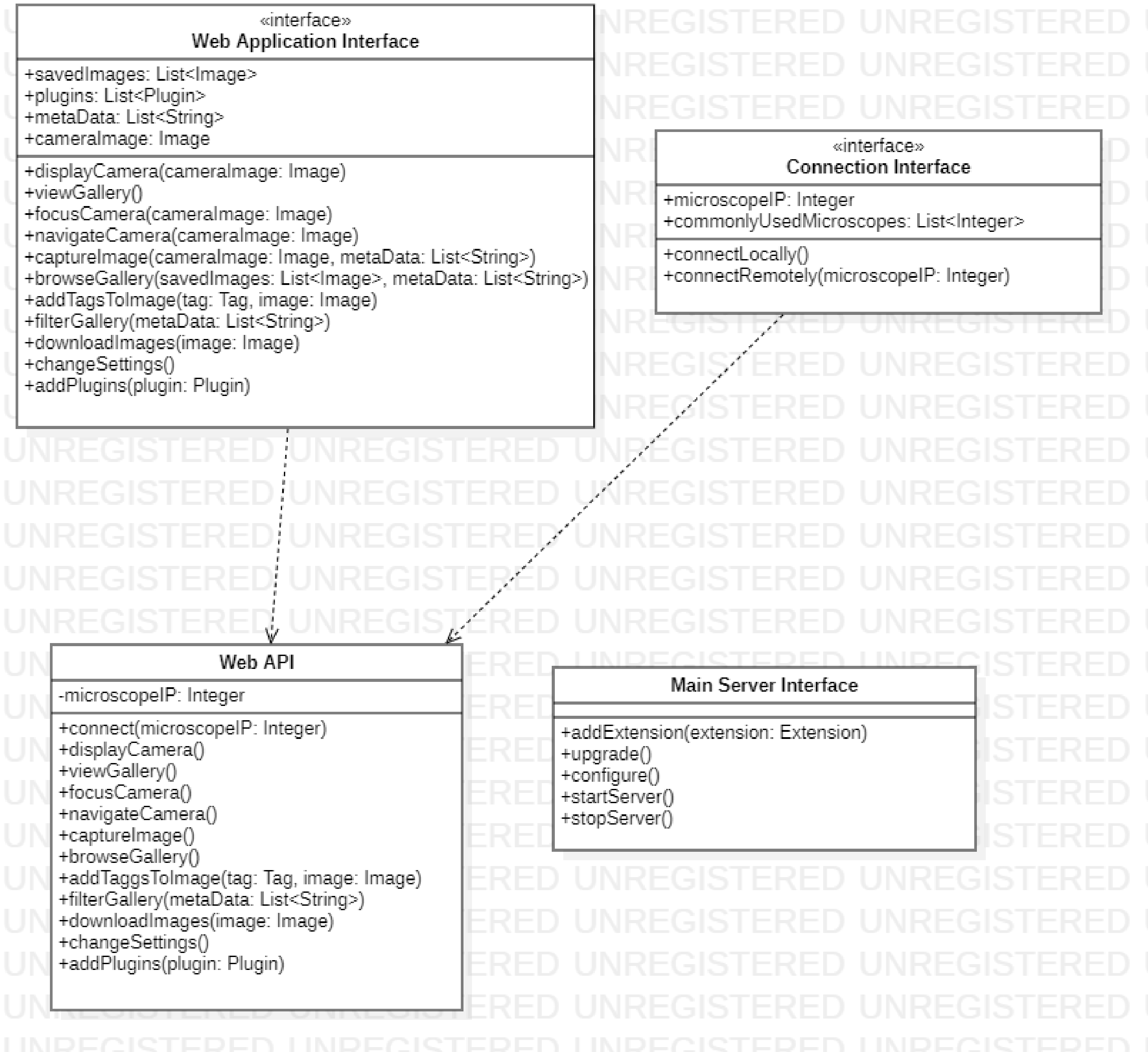

Figure 2: External Interface Diagram for OFM

3.2 Functions

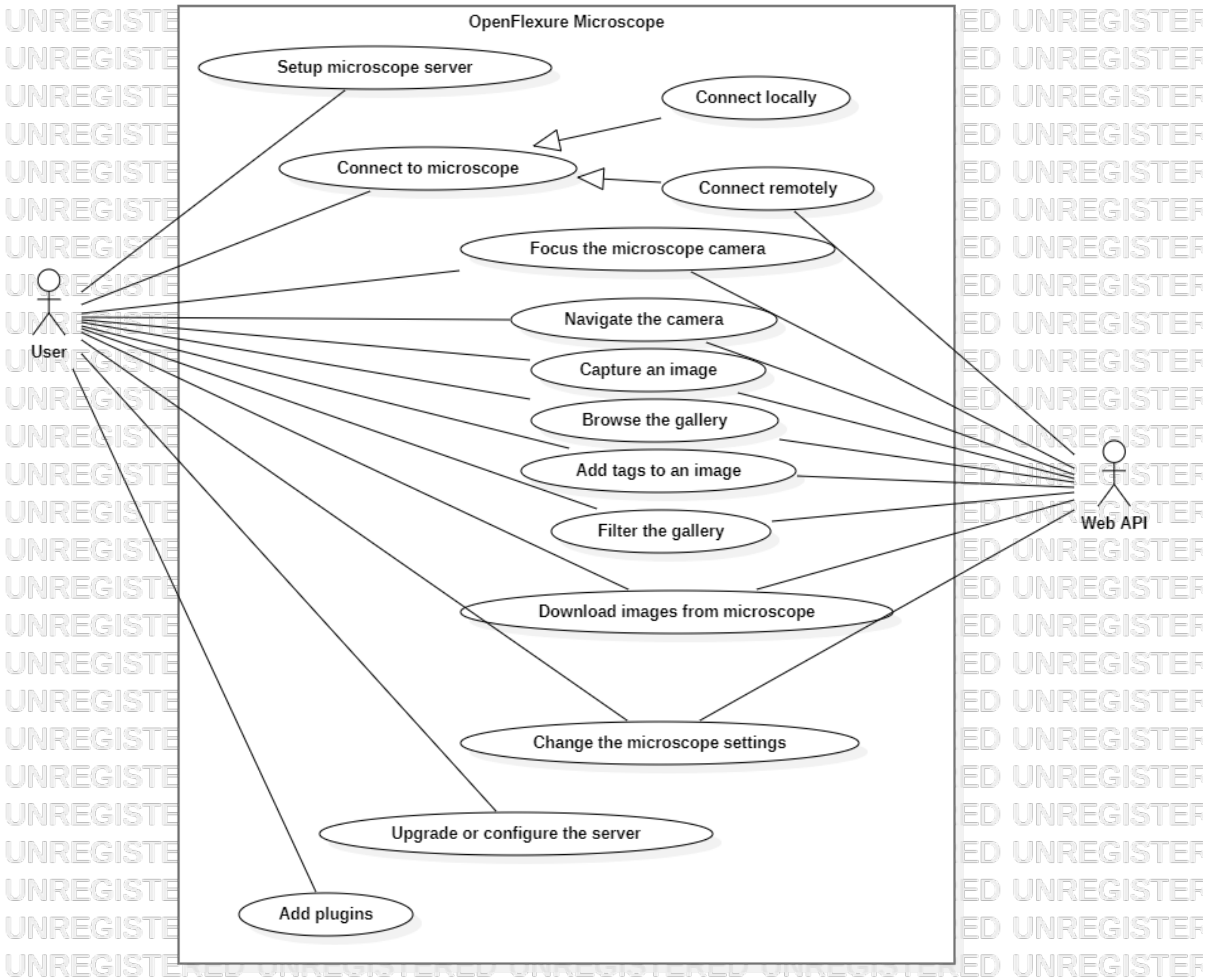

Figure 3: Use-Case Diagram for OFM

Use-Case Name	Setup Microscope Server
Actors	User
Description	User boots the microscope server for the first time and configure the connection settings.
Data	System configurations, connection settings
Preconditions	The hardware component of the microscope must be assembled and connected to power supply. It shall be connected to a external display, a keyboard and a mouse.
Stimulus	User boots the microscope.
Basic Flow	Step 1 - User needs to start the microscope server. Step 2 - User boots the microscope. Step 3 - User connects the system to internet via Ethernet cable. Step 4 - User sets the wanted configurations. Step 5 - Microscope server is started.
Alternative Flow#1	Step 3 - User connects the system to internet via wireless connection. Step 4 - User sets the wanted configurations. Step 5 - Microscope server is started.
Alternative Flow#2	–
Exception Flow	If system can't establish an internet connection, server cannot start.
Post Conditions	After completing the first boot and setup, the system saves all the configurations. The system is now capable of starting server automatically without needing to connect it to a display device.

Table 4: Setup Microscope Server

Use-Case Name	Connect To Microscope
Actors	User,Web API
Description	User connects to the microscope either locally or remotely.
Data	Microscope IP address
Preconditions	The hardware component of the microscope must be assembled and connected to power supply. It shall be connected to the external device physically if connection will be made locally.
Stimulus	User pressing the connect button from desktop application.
Basic Flow	Step 1 - User needs to connect the microscope. Step 2 - User opens desktop application. Step 3 - User chooses connection type which is either locally or remotely. Step 4 - If User chooses locally, user directly starts to see the microscope live preview in the GUI that is opened upon connection. Step 5 - If User chooses remotely, s/he waits for the application finding the microscope. Step 6 - User chooses the microscope s/he wants to connect. Step 7 - User presses the connect button. Step 8 - The microscope is connected and GUI is opened.
Alternative Flow#1	Step 5 - Application cannot find the microscope. Step 6 - User manually enters an IP address and connects to the microscope. Step 7 - The microscope is connected
Alternative Flow#2	–
Exception Flow	If the hardware components are not assembled correctly or the power supply is not provided, connection cannot occur.
Post Conditions	Upon connecting, the application finds and loads the microscope's graphical user interface. The connected microscope is saved in a list of commonly accessed microscopes.

Table 5: Connect To Microscope Function

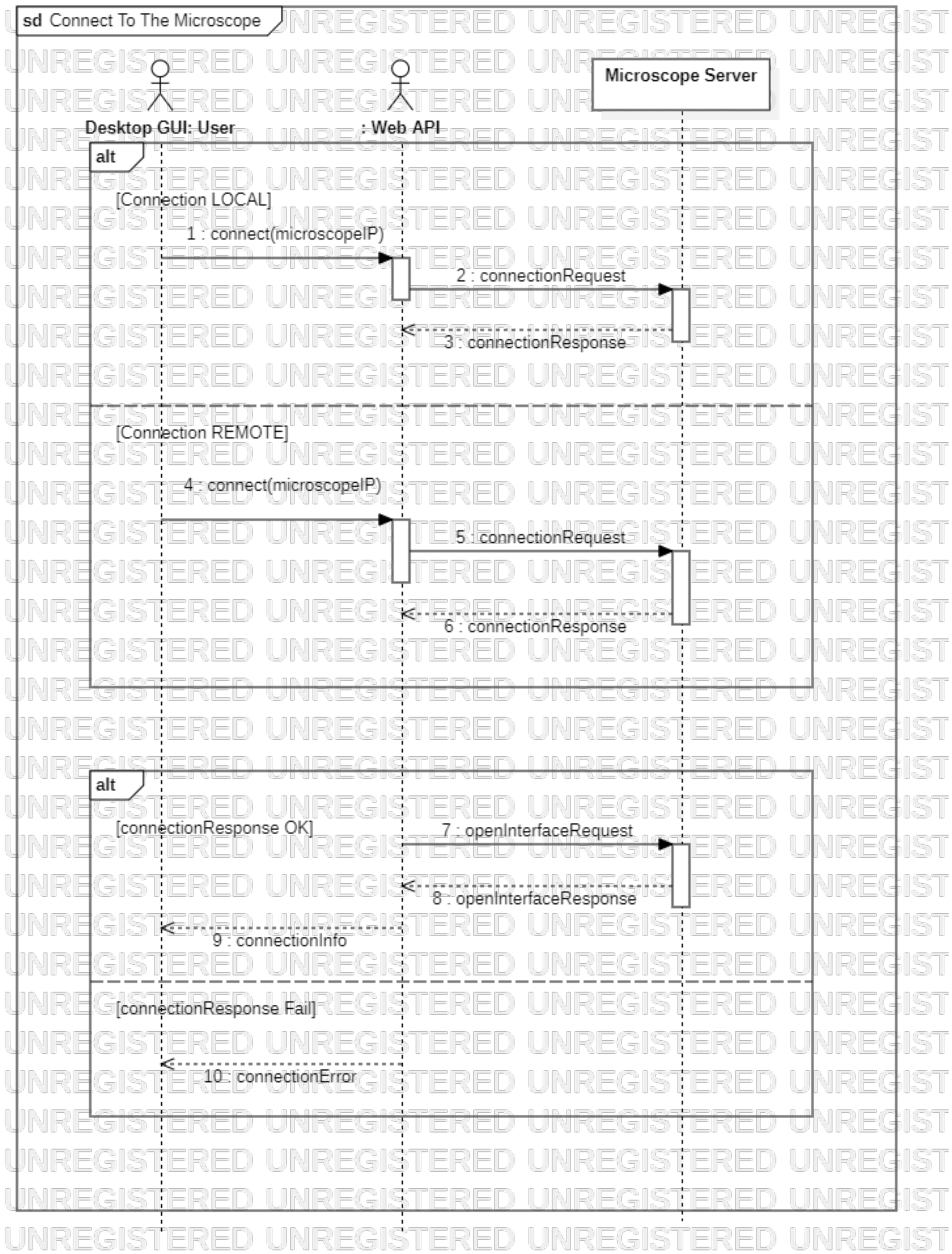

Figure 4: Sequence Diagram for Connect to Microscope use case

Use-Case Name	Auto-focus The Microscope
Actors	User, Web API
Description	User auto-focuses the microscopes camera to current location of the stage.
Data	Selected auto-focus type
Preconditions	The microscope must be assembled correctly and the user interface shall be open and connected to the microscope.
Stimulus	User pressing the chosen auto-focus button from user interface.
Basic Flow	Step 1 - User needs to focus the microscope. Step 2 - User selects the navigate tab. Step 3 - User chooses the fast auto-focus method and presses the appropriate button. Step 4 - Web API gets the requests from Web application and forwards to the microscope. Step 5 - The microscope is focused.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If the hardware components are not assembled correctly microscope may not focus.
Post Conditions	Upon focusing, the camera feed from microscope will now be focused to the current location of the microscope stage.

Table 6: Auto-focus The Microscope

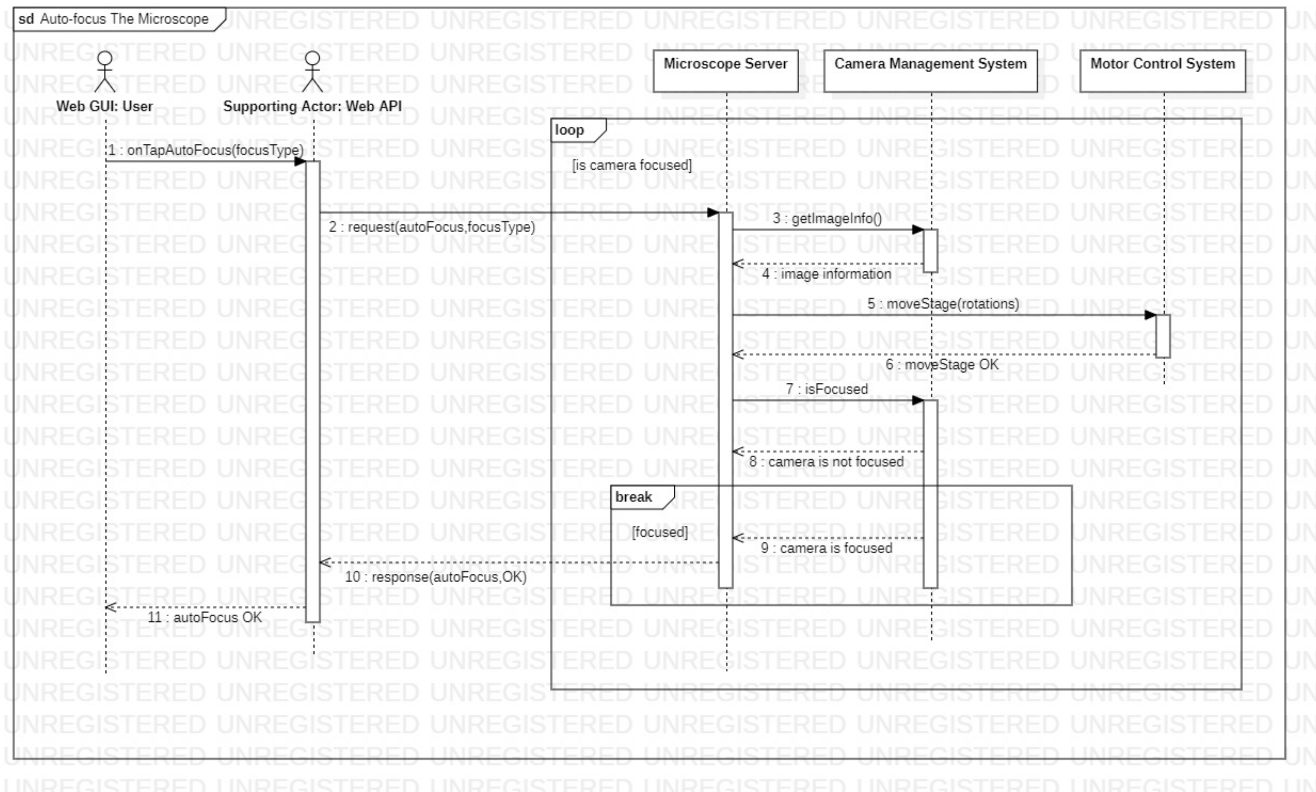

Figure 5: Sequence Diagram for Auto-focus the Microscope use case

Use-Case Name	Navigate The Camera
Actors	User, Web API
Description	User navigates the camera in x,y and z directions.
Data	Captured live image
Preconditions	User is already connected to a microscope that is assembled correctly, and the graphical user interface is already open.
Stimulus	User presses the navigate button from the interface.
Basic Flow	Step 1 - User needs to navigate the camera to see the object from different angles. Step 2 - User presses the Navigate button from the interface. Step 3 - User presses the navigation keys from the peripheral device. Step 4 - Web API gets the requests from Web application and forwards to the microscope. Step 5 - Control motors are used to apply the request by the microscope. Step 6 - Web API gets the response from microscope and sends it to the web application. Step 7 - User sees the updated live preview.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If user does not have a proper connection, the navigation may not work.
Post Conditions	The control motors are acted for the request and the preview is changed accordingly.

Table 7: Navigate The Camera

Use-Case Name	Capture An Image
Actors	User, Web API
Description	User captures the current frame from the microscopes camera feed.
Data	Captured image, image tags, image metadata, notes
Preconditions	The microscope must be assembled correctly and the user interface shall be open and connected to the microscope.
Stimulus	User pressing the capture button from the user interface.
Basic Flow	Step 1 - User needs to capture an image from the microscopes camera feed. Step 2 - User selects the capture tab. Step 3 - User chooses the appropriate settings like resolution and tags for the captured image. Step 4 - User adds notes. Step 5 - User presses the capture button. Step 6 - Web API gets the requests from Web application and forwards to the microscope. Step 7 - The image is captured.
Alternative Flow#1	Step 4 - User doesn't add notes. Step 5 - User presses the capture button. Step 6 - Web API gets the requests from Web application and forwards to the microscope. Step 7 - The image is captured.
Alternative Flow#2	–
Exception Flow	If the microscope has no empty space to save the captured image, it cannot save the image.
Post Conditions	Upon capturing, the captured image is saved to gallery with chosen settings, metadata and tags with the added notes.

Table 8: Capture An Image

Use-Case Name	Browse The Gallery
Actors	User, Web API
Description	User searches the captured images from the gallery.
Data	Captured Images and Metadata
Preconditions	The microscope should be assembled correctly and the user interface should be open and connected to the microscope.
Stimulus	User presses the gallery button from the interface.
Basic Flow	Step 1 - User needs to browse the previously captured images. Step 2 - User presses the Gallery button from the interface. Step 3 - User can browse the gallery of captured images previously
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If user does not have a proper connection, the gallery may not work.
Post Conditions	The captured images with their labels are shown in the interface

Table 9: Browse The Gallery

Use-Case Name	Add Tags To An Image
Actors	User, Web API
Description	User adds a tag to a captured image.
Data	Image tags
Preconditions	There must be at least one captured image present at the gallery.
Stimulus	User pressing the add button below the captured image.
Basic Flow	Step 1 - User needs to add a tag to an image from the gallery. Step 2 - User selects the capture tab. Step 3 - User presses the gallery button. Step 4 - User presses the add button below the image. Step 5 - User types the tag. Step 6 - User confirms the typed tag. Step 7 - Web API gets the requests from Web application and forwards to the microscope. Step 8 - Tags are added to the image.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If user types any other character than letters, process cannot be completed.
Post Conditions	Tags are added to the image.

Table 10: Add Tags To An Image

Use-Case Name	Filter The Gallery
Actors	User, Web API
Description	User filters the captured images according to their properties.
Data	Captured Images
Preconditions	The microscope must be assembled correctly and the user interface shall be open and connected to the microscope. The gallery should be open.
Stimulus	User presses the button in the gallery section
Basic Flow	Step 1 - User needs to filter the previously captured images according to some properties such as the tags of images or the date of capturing the image. Step 2 - User presses the filter button from the interface. Step 3 - User chooses the filtered property. Step 4 - Images are filtered accordingly are shown in the interface.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If user does not have a proper connection, the gallery and filtration may not work.
Post Conditions	The captured images that fit to the filtered property are shown in the interface

Table 11: Browse The Gallery

Use-Case Name	Download an Image From Microscope
Actors	User, Web API
Description	User downloads an image from the microscopes storage to the user's computer.
Data	Image
Preconditions	There must be at least one captured image present at the gallery.
Stimulus	User pressing the save image button.
Basic Flow	Step 1 - User needs to download an image from the gallery to user's computer. Step 2 - User selects the capture tab. Step 3 - User presses the gallery button. Step 4 - User double clicks the image. Step 5 - User right clicks the image. Step 6 - User presses the save image button. Step 7 - Web API gets the requests from Web application and forwards to the microscope. Step 8 - The image is downloaded.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If user does not have a proper connection, fails may occur.
Post Conditions	The image is downloaded to the user's computer.

Table 12: Download An Image From Microscope

Use-Case Name	Upgrade and Configure The Server
Actors	User
Description	User upgrades or configures the main server.
Data	Software package
Preconditions	The microscope must be assembled correctly and it must be connected.
Stimulus	User enters the required commands for the update or configuration of the server.
Basic Flow	Step 1 - User needs to configure the microscope server with new software packages. Step 2 - User connects to the main server via terminal or application. Step 3 - User enters the required commands for downloading the packages. Step 4 - User upgrades and configures the server.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If user does not enter a proper command. User is warned with a warning.
Post Conditions	The server is updated according to the command and packages configured.

Table 13: Upgrade and Configure The Server

Use-Case Name	Change the Microscope Settings
Actors	User, Web API
Description	User changes some settings about the microscope.
Data	Chosen settings
Preconditions	The microscope must be assembled correctly and the user interface shall be open and connected to the microscope.
Stimulus	User pressing the settings button from user interface.
Basic Flow	Step 1 - User needs to change the microscope settings. Step 2 - User presses settings button. Step 3 - User changes some values. Step 4 - User confirms these changes. Step 5 - Web API gets the requests from Web application and forwards to the microscope. Step 6 - The settings are changed.
Alternative Flow#1	Step 2 - User opens the settings file directly from the microscopes file system. Step 3 - User changes some values. Step 4 - User saves and exits the file. Step 5 - The settings are changed.
Alternative Flow#2	–
Exception Flow	If user inputs some incorrect values for settings, confirmation cannot be done.
Post Conditions	Upon changing settings, settings file is updated and saved.

Table 14: Change The Microscope Settings

Use-Case Name	Add plugins
Actors	User
Description	User adds new capabilities to the system by adding new plugins to the desktop application.
Data	Plugin file
Preconditions	User shall be able to change the files of the desktop application and received a plugin file from some sources.
Stimulus	User adds plugin file to the appropriate location.
Basic Flow	Step 1 - User needs to add a plugin to the desktop application. Step 2 - User locates the installation directory. Step 3 - User moves the plugin file to the plugin directory. Step 4 - User restarts the desktop application. Step 5 - The plugin is added to the application.
Alternative Flow#1	–
Alternative Flow#2	–
Exception Flow	If the plugin file is in incorrect place or corrupted, the plugin will not work.
Post Conditions	Upon adding the plugin, a new tab appears in the desktop application to use the installed plugin.

Table 15: Add Plugins

3.3 Usability Requirements

- Upon connection to the microscope, user must be navigated to the Graphical User Interface (GUI) for control over the microscope.
- User shall view the image from the GUI and shall be able to capture an image.
- User shall be able to focus the camera as well as navigate it in every direction using the GUI.
- User shall be able to browse the gallery in the GUI.
- User shall be able to add tags to the captured images, and filter them using the tags and other metadata about the image.
- User shall be able to download images from the microscope using an external device or through HTTP API.
- User shall be able to add custom plug-ins to the GUI as a graphical interface component.
- The GUI shall be used as a modular interface served by the microscope that allows the client to only render user interface elements for enabled functionality.
- New graphical interface components into the Web GUI must be able to be defined via server extensions.

3.4 Performance Requirements

- Latency between server and user for live camera feed shouldn't exceed 300ms.
- Server has at least 8 GB free space for captured images and plugins.
- Maximum number of 5 users can be simultaneously connect to the microscope.
- Web GUI should use less than 100 MB memory to ensure that application can be run from majority of modern devices.
- Server should be able to remain open indefinitely (without any external factors) to ensure that automated plugins can run without problems.

3.5 Logical Database Requirements

- When user connects to the microscope, a user entry and an active connection entry shall be created with unique IDs.
- When an connection ends, the connection shall be saved by creating a closed connections entry. After that the entry in the active connections shall be deleted.
- Every active/closed connection has an exactly one event log table which stores performed events.
- Every event performed by a user connection shall be saved by creating an event logs entry. If it is performed by a plugin it shall be stated in the event logs entry.

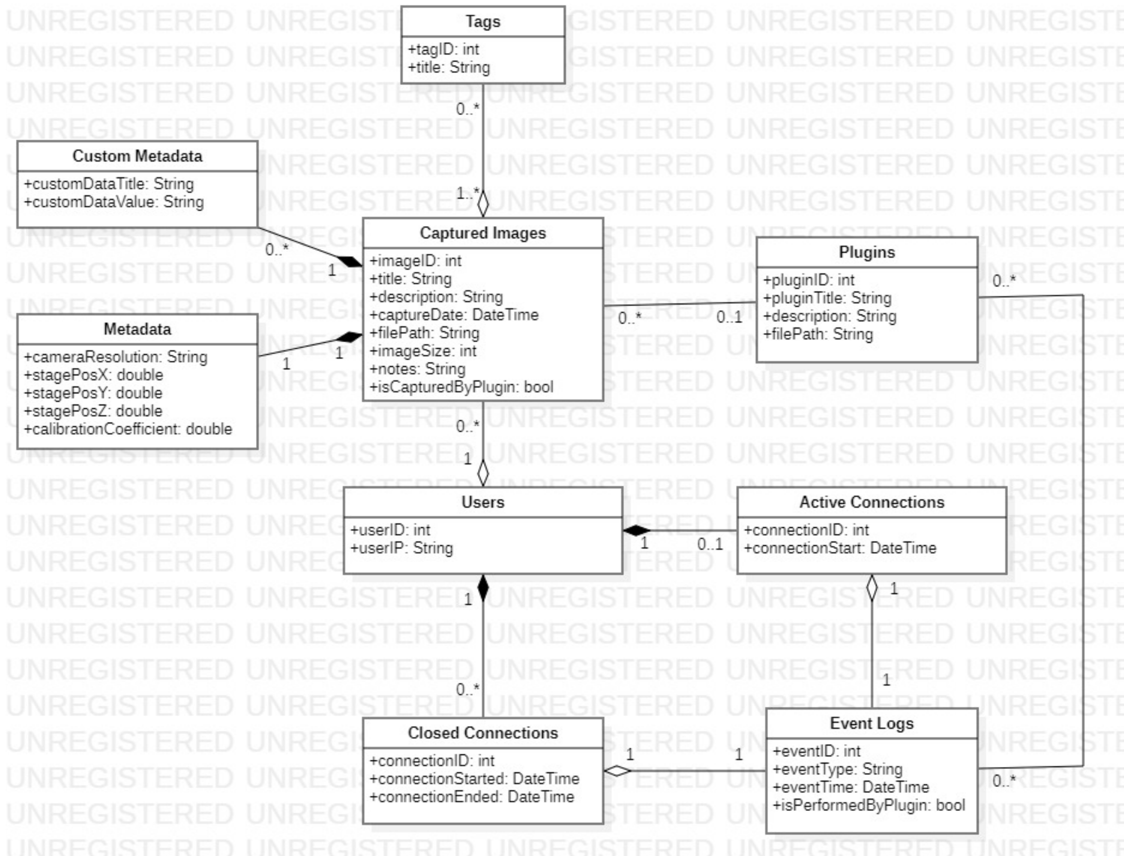

Figure 6: Logical Database Requirements Class Diagram

- Custom metadata and metadata is created when a captured image exists. Therefore, these tables are weak entities.
- Upon capturing an image, metadata is saved automatically by creating an entry. Users or plugins can create custom metadata or tags for the captured image.
- A captured image must have one metadata, but it can have zero or more custom metadata and tags.
- Whether an image is saved by a plugin or not must be saved after capturing.
- Users can have zero ore more closed connections, but they can have at most one active connections.
- A tag can appear in many images.
- Upon deleting an user, all the active and closed connections which started by s/he shall be deleted.

- Upon deleting a captured image, all the custom metadata and the original metadata shall be deleted.
- If a tag doesn't exist in any of captured images, it shall be deleted.
- All the columns in the custom metadata, metadata, users and active/closed connections shall not be NULL.
- An event logs table must associated with either an active connection or a closed connection.

3.6 Design Constraints

- Microscope's web API functionality must be compatible to W3C WoT standards.
- Users must be able to interact with the microscope using existing standard libraries in REST API.
- Custom plug-ins can be written in any programming languages as long as they conform to the W3C Web of Things (WoT) standard.
- Users can continue to interact with the microscope while an action is running, as long as they do not require a locked piece of hardware.
- Extensions must provide HTTP API endpoints and HTML interfaces that are displayed as part of the microscope's web app.

3.7 Software System Attributes

3.7.1 Reliability

- All of the hardware and software code of the system must be open-source.
- In case of a connection failure or unexpected application shutdown, the captured images will be stored in microscope hardware.

3.7.2 Availability

- Desktop and Web application must be always available whenever there is an internet connection.
- In the absence of an internet connection, applications still must be available via manual connection.
- A user shall be able to connect to multiple microscopes simultaneously.
- Different users shall be able to connect to a single microscope simultaneously.

3.7.3 Security

- Captured images must be stored locally including the metadata.
- Downloading process of captured images must conform to W3C WoT standards.

3.7.4 Maintainability

- Microscope software must be able to be updated via the terminal on the server in any time.
- Additional plug-ins must not intervene with the existing functionality of the microscope.

3.7.5 Portability

- User shall be able to connect to microscope and use all related functions of the system via IP networks whenever there is an internet connection.
- User shall be able to connect locally to microscope and use all related functions using the peripheral I/O ports of the embedded hardware inside the microscope.
- After each connection, the microscope is saved, if not already, in the list of commonly accessed microscopes.

3.8 Supporting Information

Open Flexure Microscope is an open-source laboratory project. A user can assemble the 3D-printed microscope components and hardware, along with the embedded software provided by OFM project website on a SD card. Furthermore, users with technical background can contribute to the project as an open-source developer.