Acceptance Criteria V2.0

– Based on V1.0 Prototype

No	User stories	Accpetance Criteria	Testing Cases		
			Normal	Boundary	Abnormal
1 & 2	A User wants to do some research on cells, so he/she decides to use our Napari plugins to segment the cell images. User opens the terminal in an environment-configured device, and input napari command to open the plugin.	The application should launch via terminal using the 'napari' command. The user should see the Napari interface without errors, with plugin loading successfully and ready for interaction.	Launch plugin successfully via terminal. The Napari interface appears with a plugin panel on the left.	Launch on a clean machine with a pre-installed environment setup.	Run without the correct environment. Plugin fails due to missing dependencies.
3	The Napari will automatically open and the user clicks on the plugin menu on the top left and then see below drop down menu	Napari should open and the plugin dropdown menu should be clickable from the top left. All plugins listed must be selectable.	Click the top left menu and see the plugin list with Annotator and Auto-Segmentation .	Open plugin list before loading any image.	Plugin list doesn't appear due to incomplete install.
4	Firstly the user clicks on Annotator 2d, and the right side widget will be opened. The user is also able to see a few masks in the left widgets.	Clicking Annotator 2D should open a right-side widget and show mask layers on the left panel.	Click Annotator 2D → right-side widget opens, masks appear on left.	Open Annotator 2D before loading any image.	Plugin crashes due to missing dependencies in `micro-sam`.
5 & 6	The user is able to click on the button 'File' on the top left and then 'open-file' button so that he can review the device libraries to choose a tiff image. (On the other cases, the user is also able to just drag the tiff image into the napari windows, it's also acceptable.) The user clicks the 'open' button and then the image will be displayed on the screen.	User opens a file using 'File' > 'Open-file' or drags the image into the window. Image appears on canvas.	Click open-file and select TIFF → image shows on canvas.	Drag image file to window directly.	Image format unsupported (e.g., .jpeg), or drag fails silently.

7	The user clicks the 'Compute Embedding' button firstly to compute the image. The image is successfully loaded if the user sees this display as below.	User clicks 'Compute Embedding' and sees status bar or visual feedback indicating embedding success.	Embedding completes within 10 seconds; no crash.	Embedding large images (>5MB).	Compute fails due to GPU memory limit or missing model.
8 & 9	Then the user clicks on the 'auto-segmentation' button on the right, the napari will segment the cell images. The user is able to see the result of auto-segmentation in the auto-segmentation mask on the left widget.	Clicking 'auto-segmentation' triggers the segmentation process and creates a new mask layer with colored regions.	Auto-segmentation mask appears with segmented cells.	Run auto-segmentation on grayscale low-contrast images.	Auto-segmentation fails silently if the image is not embedded.
10	The user can adjust the opacity, gamma and contrast limit to let the cell images be displayed on the screen.	User adjusts the display via opacity, gamma, and contrast sliders. Visual result changes in real-time.	Sliders update the displayed image instantly.	Adjust values to extreme limits (e.g. 0.01, 10).	UI crashes when slider values are out of bounds.
11-14	The user finds that the accuracy of auto-segmentation is a bit low. And then he/she uses box-segmentation to choose the cells that have been wrongly segmented. The user clicks on the prompt button in the plugin list of Napari. The user clicks on the Bounding box button on the left widget, and uses the draw box button on the top left of the window to draw the box. After the user draws the box on the image, he/she clicks overwrite on the right widget inside the box-segmentation widget, and then clicks on the segment button, he/she finds that two	Box-segmentation should allow users to draw bounding boxes on image, overwrite existing masks and generate new segmented results.	Draw box → overwrite → click segment → updated result appears.	Draw box overlapping existing masks.	Click segment without drawing box → nothing happens or error occurs.

	cells have been segmented into one.				
15 & 16	The user still does not get what he/she wants, then the user clicks the Prompt Points on the left widget. The user clicks the 'add-point' button so that the plugin can automatically add new points and segments itself. Then it will separate those two cells with points.	User clicks 'add-point' to create segmentation from clicked points. Point prompts refine mask result.	Click add-point → click image → mask refines with clicked region.	Add multiple close points in overlapping regions.	Click add-point plugin without any image loaded.
17 - 19	The user clicks the different mask and is able to only display specific layers of cells. The user is able to click on the 'seen/unseen' button on the left of each mask. The user is able to see only a few masks that what he/she wants.	User toggles visibility of masks using 'seen/unseen' buttons to control which layers are visible.	Click toggle → selected mask becomes hidden/shown.	Toggle multiple masks on/off simultaneously.	Click delete on mask and expect toggle instead – causes permanent loss.
20 & 21	The user selects a different image and auto-segmented the image again. The user gets the results and closes the Napari.	User loads a second image and re-runs auto-segmentation. Then the plugin exits gracefully.	Load new image → auto-segment masks appear again.	Repeat on 3+ images in one session.	Loading a new image without clearing the previous session leads to overlap/memory issues.