# AMI Invigilation Comments

Date: 27/03/2025

Status: Draft

Contents

[AMI Invigilation Comments 1](#_Toc193963853)

[Introduction, Important Notes 1](#_Toc193963854)

[COMM1: Student Portal>My Courses>Course>Assessments>Exam 2](#_Toc193963855)

[Student Portal>My Courses>Course>Assessments>Exam 2](#_Toc193963856)

[Student Portal>My Courses>Course>Assessments>Exam>Submit 2](#_Toc193963857)

## Introduction, Important Notes

## COMM1: Student Portal>My Courses>Course>Assessments>Exam

**1.1**

|  |  |
| --- | --- |
| UC1 | Student Portal>My Courses>Course>Assessments>Exam**SPEC:**IF CAMERA SHARE IS SELECTED IN AMI: 1. **Each time** a warning is logged, take a snap shot of video stream with attached warning.

IF AUDIO SHARE IS SELECTED IN AMI:1. Each time a warning is logged regarding audio, take 3 second audio recording from when warning is picked up.

**COMMENT:*** Image and Audio is not logged **EVERY TIME** a warning is logged
* Noise is not easily detected with noise cancelling headphones.

 |
| UC2 | Student Portal>My Courses>Course>Assessments>Exam>Submit**SPEC:**When a student submits their assessment, the following must be included:* 1. Add a folder called “Exam Warnings” with all snap shots and audio recordings taken during the assessment.
	2. Add a text file of **ALL** warnings with their timestamps during the assessment
	3. Attach above in the submissions zip file for the student.

**Comment:*** Snap shots and audio recordings are not being saved with file extensions and therefore do not open:

 |
| UC3 | When tab is changed in browser, it should be added to the warning logs (and also then must be in the log that is exported) It should state "Browser tab changed during assessment", but only if screenshare is enabled.useEffect(() => { const handleVisibilityChange = () => { if (document.hidden) { console.log('User switched away from the tab');// ADD TO WARNING LOGS } else { console.log('User switched back to the tab'); } }; document.addEventListener('visibilitychange', handleVisibilityChange); return () => { document.removeEventListener('visibilitychange', handleVisibilityChange); };}, []); |