Guide to use cases

2 required use cases: Print Document, Record Print Request

**Note:*

- question mark (?) shows conditional branching
- The `code` parts shows the recommended calls for the sequence diagram

Print Document

- **Initial state**: user logged in as "Student"
- **Activities/Sequences**

1. system shows list of printers
2. user selects a printer `SelectPrinter(printerID)`
    1. Check printer condition
        1. ? The printer is currently online
            1. Initiate the GUI for file uploading and setting printing properties
            2. Continues to *3.*
        2. ? The printer is unavailable
            1. User is notified about the problem `Error("Printer Unavailable!")`
            2. Return to *1.*
3. user choose to upload a file `UploadFile(FileName)`
    1. A window pop up for the user to choose a file
    2. The system check the file extension
        1. ? File extension is valid. File upload shows status "ok" `FileStatus("OK")`
            1. Process can still continues to *4*.
        2. ? File extension is valid. File upload shows status "Invalid file extension, valid ones are .docx, .doc, or .pdf" `FileStatus("Invalid");` `ShowMessage("Invalid file extension, valid ones are .docx, .doc, or .pdf")`
            1. Process can still continues to *4*.
    3. File sent to server
4. user specify the printing properties `SetProperties(paperSize, numCopy, isColor)`

1. Check if printer can print with color
    1. Yes. Continues to *5.*
    2. No.
        1. Notify user. `ShowMessage("Warning! This printer does not support color printing!")`
        2. Continues to *5.*
5. user hit button "Print document" `UserRequestPrint(filename, paperSize, numCopy, isColor)`
6. Check file condition
    1. ? File is invalid.
        1. Reject print request `Error("Invalid file extension!"); RejectPrint()`
        2. Go back to *3.*
    2. ? File is valid. Proceed to *7.*
7. Calculate the required number of page
8. Check balance
    1. ? Insufficient balance. Reject print request `Error("Insufficient page balance!"); RejectPrint()`
    2. ? Sufficient balance.
        1. Notify user of successful print request. `ShowMessage("Print request sent! Printing is underway!`.
        2. Continue to *9.*
9. Send print request to printer `SystemRequestPrint(file, paperSize, numCopy, isColor)`
10. Return the status from the printer
    1. ? Print successful. Notify the user `ShowMessage("Document printed successfully! Come pick up your document!`
    2. ? Print failed. Notify the user `ShowMessage("Unexpected error occurred at printer! Print failed!")`

Record Printing Request*

- **Initial state**: A printer just finished handling a print request
- **Activities/Sequences**

1. Printer request to record to the system database `RequestToRecord(printerID)`
2. The system put the request into a waiting queue
3. System get a waiting request from the queue to handle it
4. System handle the request
    1. Accept.
        1. System register a row for the request `Accept(printerID)`
        2. System request the printer to send the record. `WaitingForData(printerID)`
        3. Proceed to *4*.
5. The printer sends the printing record to the system `RecordPrint(printerID, filename, pageSize, NumCopy)`