A smart printing service for students at HCMUT

The university is intent to build a Student Smart Printing Service (HCMUT_SSPS) for serving students in its campuses to print their documents.

The system consists of some printers around the campuses. Each printer has ID, brand/manufacturer name, printer model, short description, and the location (campus name, building name, and room number).

The system allows a student to print a document by uploading a document file onto the system, choose a printer, and specifying the printing properties such as paper size, pages (of the file) to be printed, one-/double-sided, number of copies, etc. The permitted file types are limitted and configured by the Student Printing Service Officer (SPSO).

The system has to log the printing actions for all students, including student ID, printer ID, file name, printing start and end time, number of pages for each page size.

The system allows the SPSO to view the printing history (log) of all students or a student for a time period (date to date) and for all or some printers. Of course, a student can also view his/her printing log for a time period together with a summary of number of printed pages for each page size.

For each semester, the university give each student a default number of A4-size pages for printing. Students are allowed to buy some more using the feature Buy Printing Pages of the system and pay the amount through some online payment system like the BKPay system of the university. The system only allow a student to print some number of pages when it does not exeed his/her account (page) balance. Note that, one A3 page is equivalent to two A4 pages.

The SPSO has a feature to manage printers such as add/enable/disable a printer.

The SPSO also has a feature to manage other configuration of the system such as changing the default number of pages, the dates that the system will give the default number of pages to all students, the permitted file types accepted by the system.

The reports of the using of the printing system are generated automatically at the end of each month and each year and are stored in the system, and can be viewed by the SPSO anytime.

All users have to be authenticated by the HCMUT_SSO authentication service before using the system.

The system are provided through a web-based app and a mobile app.

---------Task ----------

Task 1: Requirement elicitation	1.1	Describe the domain context of a smart printing service for students at HCMUT. Who are relevant stakeholders? What are their current needs? In your opinion, what benefits HCMUT-SSPS will be for each stakeholder? Hint: At least three paragraphs have to be written (1) a paragraph about domain context, (2) a paragraph about stakeholders and their needs, (3) a paragraph about the benefits of HCMUT-SSPS for each stakeholder. The section has to be *understandable*, refer to *reliable* sources and information has to be *justified*.
	1.2	Describe all functional and non-functional requirements that can be inferred from the project description. Hint: At least 05 functional requirements for each stakeholder. Requirements to be written as single sentences. The requirements must be *complete, unambiguous, consistent and correct*. The use case diagram has to be *complete and correct syntactically*.
	1.3	Draw a use-case diagram for the whole system. Choose an important module and draw its use-case diagram, as well as describe the use-case using a table format Hint: one use case diagram shall be drawn. The use case diagram has to be *complete and correct syntactically*. One use-case description table for each use case using the example from the lecture. The number of steps in a use-case should be more than 3. At least one use-case should have an exception flow.
Task 2: System modelling	2.1	Draw an activity diagram to capture the business process between systems and the stakeholders in a particular module (choose a module in Task 1.3) Hint: draw an activity diagram for each use case with a swimlane (https://circle.visual-paradigm.com/activity-diagram-example-swimlane/) for different stakeholders. The number of diagrams should be maximum 5. Try to use as many notations of the diagrams as possible (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/). Write 01 paragraph for each diagram to describe it. The diagrams should be *correct and complete*.
	2.2	Draw a sequence diagram for a particular module (the same with the module used in task 2.1) Hint: draw 01 sequence diagram for each use case with a swimlane for different stakeholders. The number of diagrams should be maximum 5. Try to use as many notations of the diagram as possible (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/) Write 01 paragraph for each diagram to describe it. The diagrams should be *correct and complete*.
	2.3	Draw a class diagram of a particular module (the same with the module used in task 2.1) as comprehensive as possible Hint: draw 01 class diagram for the whole module. Try to use as many notations of the diagram as possible (https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/). The diagram should be *correct and complete*.
	2.4	Develop MVP 1 as user interfaces of either a Desktop-view central dashboard for a particular module (the same with the module used in task

		2.1). Decide yourself what to include in the view. Use a wireframe tool like Figma or Adobe XD, or Illustrator Hint: The wireframe shall be made from scratch. Language is Vietnamese. Any photos or materials taken from Internet has to give reference/ links. No implementation (backend or frontend) is needed at this stage. The user interface shall have *at least 05 screens*. There should be *clickable links* to navigate among the screen. The user interface should be as *close to a final software's screenshot as possible* (high fidelity wireframe).
Task 3: Architecture design	3.1	Use a layered architecture to design the HCMUT-SSPS system. Describe how will you present your User Interface. Describe how will you store your data. Describe how you will access to external services/ APIs. Hint: draw 01 architectural diagram for the overall design of HCMUT-SSPS system. Write 01 paragraph for your Presentation strategy, 01 paragraph for Data storage approach and 01 paragraph for API management. The architectural decisions in these paragraphs should be *justified* and *associated with external links* for detail approaches.
	3.2	Draw a component diagram for an important module (the same with the module used in task 2.1) Hint: draw 01 component diagram for the whole module. Try to use as many notations of the diagram as possible (https://online.visual-paradigm.com/diagrams/tutorials/component-diagram-tutorial/). See the notation description from the lecture. Write 01 paragraph for each diagram to describe it. The diagram should be *correct and complete*.
Task 4: Implementation – Sprint 1	4.1	Setting up an online repository (github, bitbucket, etc) for version control. Hint: setup Github (https://docs.github.com/en/get-started/quickstart/hello-world) and a hello world example. Every team shall *have an account with public/ shareable links* to the repository.
	4.2	Adding documents, materials and folders for Requirement, System modelling and Architectural design. Use the selected version control system to report the changes to these files. Hint: the repository shall have a *readme file*. The repository shall be frequently maintained (*basing on last updated time).*
	4.3	Conducted a usability test with the user interface you developed in MVP 1. Hint: follow the guideline https://www.nngroup.com/articles/usability-testing-101/ Key steps include: 1. Recruit participants/ testers. 2. Define tasks. 3. Define test strategy (qualitative vs. quantitative, remote vs. in-person. 4. Conduct the test. 5. Document the feedback from testers. All key steps shall be documented in 01 test report. The test report should be complete, understandable, having illustrative photos. The test report should be less than 10 A4 pages.
Task 5: Implementation – Sprint 2	5.1	Develop MVP 2 with input from Task 2.4 and Task 4.3. You are free to choose the programming language (HTML, Javascript, Python, C#, etc). It is not required to implement a database in the backend. Data can be hard coded in code files. Hint: the MVP should be demonstrable in either a desktop view or a mobile view. The MVP should capture *most important value* for all stakeholders from the project description. The MVP should have *at least 05 different screens/ views*. There should be *hyperlinks to navigate through views*.
	5.2	Demonstrate the whole project from Task 1 to Task 5

		Hint: a presentation slide shall be prepared. Each team should practice the demonstrations many times in advance. The presentation should be *straight to the point, contains lessons learned for the team*. The demonstration should be *prepared, correct, brief* and has *good quality*.