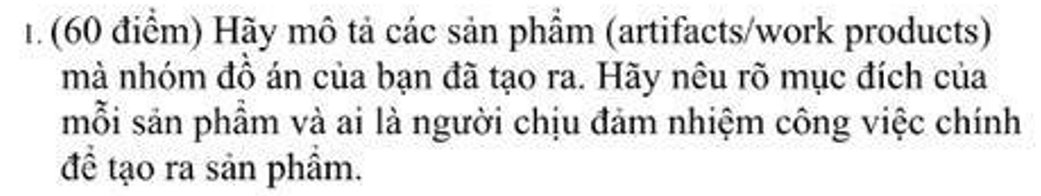
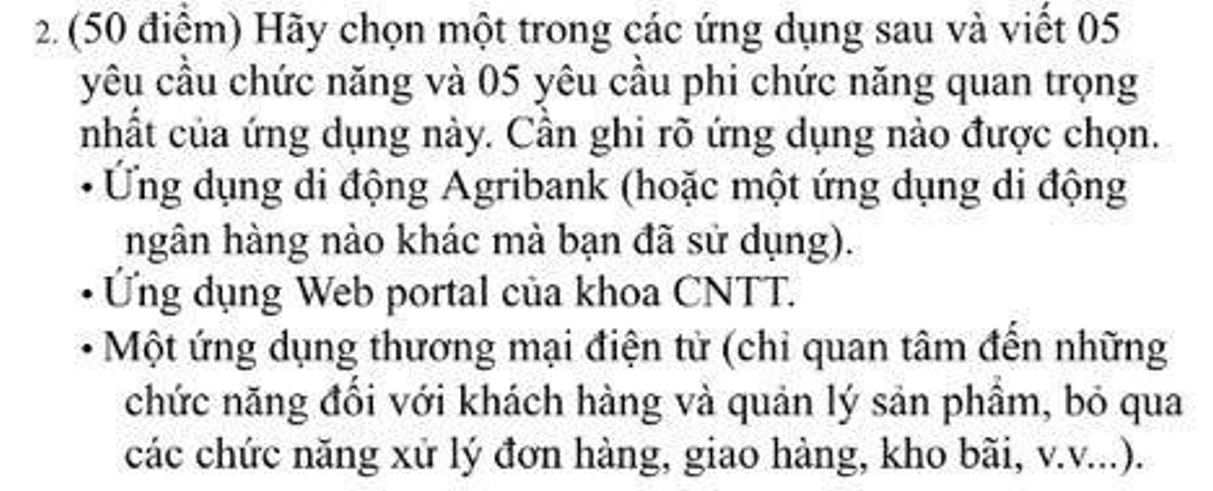
**NMCNPM**



* Sản phẩm của nhóm đồ án: Hệ thống online video conference như Zoom, Google meet.
* Mục đích: Cho phép tạo và tham gia cuộc họp trực tuyến, đặc biệt cần thiết trong làn sóng “Work from home” như hiện nay. Ngoài ra phần mềm cũng có những tính năng như bật webcam, chia sẻ màn hình, biểu tượng giơ tay, …
* Phân công công việc:
  + Em phụ trách về phần Backend, quản lý Database, viết báo cáo – các template. Ngoài ra cũng thực hiện phần Testing chung với cả nhóm.
  + Bạn Tiến: Phụ trách phần Frontend, thiết kế giao diện người dùng.
  + Bạn Bình và Trung thực hiện phần Backend, tổ chức các API, code các tính năng cần thiết của phần mềm.
  + Bạn An thực hiện xây dựng các requirement cho ứng dụng, quản lý chính việc testing.



* Ứng dụng Angribank:
  + Yêu cầu chức năng:
    - Chức năng tạo tài khoản, đăng nhập tài khoản.
    - Chức năng nạp / rút tiền.
    - Chức năng chuyển khoản.
    - Chức năng gửi tiết kiệm.
    - Chức năng báo lỗi giao dịch, trợ giúp khi gặp sự cố.
  + Yêu cầu phi chức năng:
    - Ứng dụng có hiệu năng cao, tốc độ chạy nhanh.
    - Ứng dụng có độ bảo mật cao, có các biện pháp bảo vệ cho tài khoản người dùng.
    - Giao dịch diễn ra nhanh chóng, an toàn, bảo mật.
    - Giao diện thân thiện với người dùng.
    - Có khả năng tương thích trên nhiều nền tảng, liên kết được với các ví điện tử.
    - Chi phí thực hiện và bảo trì rẻ.
* Ứng dụng Web portal:
  + Yêu cầu chức năng:
    - Chức năng tạo tài khoản, đăng nhập tài khoản.
    - Cập nhật thông tin cá nhân, chứng chỉ liên quan.
    - Đăng kí học phần.
    - Xem lịch thi, xem điểm thi, xem học phí.
    - Đánh giá môn học.
  + Yêu cầu phi chức năng:
    - Ứng dụng có hiệu năng cao, tốc độ chạy nhanh.
    - Bảo mật thông tin tốt.
    - Không bị nghẽn khi có lượng lớn truy cập vào cùng thời điểm.
    - Giao diện thân thiện với người dùng.
    - Có khả năng tích hợp với các trang web khác của trường, các khóa học online, …
    - Chi phí thực hiện và bảo trì rẻ.
* (30 points) Explain why Waterfall model can be more effective than Scrum if the project’s requirements are stable during the project.

Nếu yêu cầu về project ổn định, không thay đổi quá nhiều thì Waterfall có thể hoạt động tốt hơn Scrum. Mô hình Waterfall theo thứ tự tuần tự, công đoạn này hoàn thành mới bắt đầu công đoạn tiếp. Với các dự án ổn định, không có quá nhiều sự thay đổi thì Waterfall thực hiện đơn giản, ổn định hơn. Nó dễ hiểu, dễ triển khai, dễ đo lường các yếu tố như năng suất, …

* (30 points) Explain why Scrum is more effective than RUP if the project’s requirements change rapidly during the project?

Rational Unified Process (RUP)

Nếu yêu cầu project thay đổi nhiều, nhanh chóng thì Scrum tốt hơn. Do RUP cần xác định các scope, major milestones cũng như ngày cụ thể (Scrum thì dùng backlog chứ không phải scope). Ngoài ra RUP có 4 phases lớn trong vòng đời project còn Scrum thì dồn lại thành 1 lần lặp.

While both Scrum and Rational Unified Process (RUP) follow the Agile framework, RUP involves more formal definition of scope, major milestones, and specific dates (Scrum uses a project backlog instead of scope). In addition, RUP involves four major phases of the project lifecycle (inception, elaboration, construction, and transition), whereas Scrum dictates that the whole “traditional lifecycle” fits into one iteration.

* (30 points) What is the difference between high-level design and low-level design? Give one example output of each.

Với HLD, **phần cứng được mô hình hóa bằng chính ngôn ngữ dùng để phát triển phần mềm** như C/C++ hay SystemC. Điều này **giúp phần mềm và phần cứng có thể được tích hợp sớm để thử nghiệm và đánh giá hiệu quả và tính khả thi**. Sau khi hệ thống được tối ưu như yêu cầu, phần cứng mới bắt đầu được thiết kế trùng khớp với mô hình đã phát triển. Với cách này, phần cứng được đảm bảo tương thích phần mềm và đáp ứng tốt cho hệ thống mong muốn. Bên cạnh đó, phần mềm có thể tiếp tục được thử nghiệm và hoàn chỉnh song song với việc phát triển phần cứng.

Low level design giống như 1 phiên bản cụ thể hóa của high level design. Low-level design (LLD) là một quá trình trong quy trình phát triển phần mềm, trong đó **người lập trình xác định các chi tiết cụ thể về cách thức hệ thống sẽ thực hiện các tác vụ được yêu cầu**. Đây là một bước quan trọng trong quy trình phát triển phần mềm, vì nó cung cấp một khung cảnh rõ ràng về cách thức hệ thống sẽ hoạt động và cách thức nó sẽ đáp ứng các yêu cầu của người dùng.

Trong low-level design, người lập trình sẽ **xác định các module cần thiết để thực hiện các tác vụ được yêu cầu, các liên kết giữa các module và các giao diện giữa các module**. Người lập trình cũng sẽ xác định các thuộc tính của các module và các tham số cần thiết để thực hiện các tác vụ. Low-level design cũng có thể bao gồm việc xác định các thuật toán hoặc các giải thuật sẽ được sử dụng trong các module để thực hiện các tác vụ.

**The Low-level design is created after the High-Level Design**.

**(LLD tạo sau HLD)**

Trong high-level design, người lập trình sẽ xác định các tác vụ chính mà hệ thống sẽ thực hiện và cách thức chúng sẽ được thực hiện bởi hệ thống.

Sau khi high-level design đã hoàn thành, người lập trình sẽ tiến hành low-level design để xác định chi tiết hơn cách thức hệ thống sẽ thực hiện các tác vụ được yêu cầu. Low-level design sẽ xác định các module cần thiết để thực hiện các tác vụ, các liên kết giữa các module và các giao diện giữa các module. Nó cũng sẽ xác định các thuộc tính của các module và các tham số cần thiết để thực hiện các tác vụ.

