**KHOA AN TOÀN THÔNG TIN 1**

****

**Môn: Nhập Môn Công Nghiệp Phần Mềm**

**BÁO CÁO BÀI TẬP LỚN ĐỀ 8**

**Họ và tên sinh viên:**

**Đỗ Tiến Sĩ**

**Mã số sinh viên:**

**B20DCAT153**

**Họ và tên giảng viên:**

**Thầy: Đỗ Thị Ngọc**

**Hà Nội 5/2023 (tháng/năm)**

1. Thế nào là corrective maintenance?

- việc bảo trì chỉnh sửa là một hoạt động quản lý phần mềm nhằm khắc phục các lỗi và sự cố trong phần mềm.

2. Thế nào là adaptive maintenance?

- bảo trì thích ứng: là hoạt động quả lý phần mềm nhắm thay đổi, điều chỉnh hoặc cải tiến tiến độ để phù hợp với quá trình.

3. Thế nào là perfective maintenance?

- là một hoạt động trong bảo trì nhằm sửa lỗi hệ thống sau khi hệ thống đã được đưa vào sữ dụng

4. Thế nào là refactoring?

- là hành vi tái cấu trúc lại hệ thống nhằm cập nhật điều mới và sửa đổi cho hệ thống tốt hơn

5. Thế nào là "from scratch"?

- là loại dự án mà phát triển mà không có dựa trên bất kì nguồn hặc tài nguyên nào từ dự án trước đó.

6. Thế nào là moving target problem?

- là vấn đề trong bảo mật mang, khi mục tiêu bị tấn công không cố định và thay đổi theo tg.

7. Thế nào là regession fault?

- là lỗi trong phần mềm mà khi ta phát triển phần mềm thì nó sẽ lỗi khi sử dụng lại chức năng của các phiên bản phía trước.

8. Thế nào là một episode?

- là một sự kiện hoặc mootjh chuỗi các sự kiện diễn ra như một phần của sự kiện lớn.

9. Thế nào là một iteration?

- là một chu kì phát triển phần mềm.

10. Thế nào là một increasement?

- là sự tăng trưởng.

11. Thế nào là một artifact?

12. Thế nào là portability?

- Là khả năng của một ứng dụng hoắc phần mềm có thể chạy trên nhiều nền tảng mà không phải sửa đổi code và mã nguồn.

13. Thế nào là reusebility?

- Là sự tái sử dụng trong phát triển phần mềm. điều này giúp tránh lãng phí thời gian và chi phí phát triển trong phần mềm.

14. Thế nào là milestone?

- cột mốc hoặc là một sự kiện quan trong trong một đích đến trong quá trình phát triển dự án.

15. Thế nào là một story?

User story là motoj phần mềm nhỏ được miêu tả bằng ngôn ngữ tự nhiên nhằm mô tả tính năng của một người dùng.

16. Thế nào là refactoring?

- là tái cấu trúc: thiết kế code mà không thay đổi hành vi và phần mềm của nó.

17. Thế nào là concept exploration?

- điều tra sơ bộ về nhu cầu của khác hàng

18. Thế nào là business model?

- sau khi nhà phát triển hiểu rõ về nhu cầu phát triển của khách hàng thì họ bắn đầu hiểu đến mô hình kinh doanh của khác hàng.

19. Thế nào là traceability?

- là khả năng theo dõi và truy xuất các yêu cầu, thoogn tin và tài liệu liên quan trong quá trình phát triển phần mềm hoặc sản phẩm.

20. Thế nào là egoless programming?

- mô hình lập trình ngang bằng:

+) các thành viên bình đẳng về chức vụ.

+) mỗi người tự do thiết kế, code và test modul của mình.

+) việc có lỗi được coi là việc bình thường.

+) cả đội xây dựng một tính năng hay cả sản phầm và là của cả đội.

21. Thế nào là Project Manager?

- là người sắp xếp kế hoạch, thời gian thực hiện công viêc, trang thiết bị, ngân sách,…. Của một dự án từ khi bắt đầu tới kết thúc.

22. Thế nào là technical leader?

- trưởng nhóm kĩ thuật là kỹ sư giám sát chất lượng công việc của nhóm phát triển trong phần mềm và đảm bảo sự thành công của giải pháp được giao.

23. Thế nào là programming secrectary?

-

24. Thế nào là backup programmer?

- lập trình nhóm dự bị:

+) dự bị cho sếp của nhóm.   
+) có kĩ năng tương đương sếp trong quản lý code.

25. Thế nào là supper programmer? ???

- siêu lập trình…

26. Thế nào là một bản thiết kế còn ommision? ?????

- phần mềm hoặc một khía cạnh tính năng bị bỏ sót hoặc thiếu.

27. Thế nào là một bản thiết kế còn contradiction?

Contradiction là sự xung đột giữa 2 hay nhiều điều không thể cùng tồn tại, cùng đúng đắn một lúc. Bản thiết kế còn contradiction là bản thiết kế còn tồn tại các xung đột đôi khi trong bản còn chưa rõ những điều kiện gì ở trong.

28. Thế nào là một phần mềm COTS?

COTS: commercial off-the-shelf (COTS) software là phần mềm loại thương mại có sẵn.Phần mềm này loại phần mềm mà bán với giá rẻ cho số lượng lớn người sử dụng. Nhà phát hành sẽ thu hồi vốn khi mà bán với số lượng lớn.

29. Thế nào là SPMP?

software project management plan: đây là kế hoạch quản lý dự án phần mềm.

30. Thế nào là alpha release?

- trong phần mềm thì khi mà phát triển phần mềm thì khi ta làm xong thì đưa cho người sử dụng là sự phát hành alpha.

31. Thế nào là beta release?

- sau khi sự phát hành alpha thì sau quá trình phát triển, cải thiện thì sự phát hành sau là beta.

32. Thế nào là process?

- là tập hợp các hoạt động được thực hiện theo thứ tụ cụ thể để đạt được mục tiêu nào đó.

33. Thế nào là workflow?

- là chuỗi các bước, hoạt động và quy trình liên quan nhau được thực hiện để hoàn thành một công việc và quá trình cụ thể.

34. Luật Miller trong CNPM nói gì?

- Tại mỗi thời điểm, người ta chỉ có thể tập trung vào tối đa 7 vấn đề.

35. Luật Brooks trong CNPM nói gì?

- Đưa thêm người vào 1 project đang chậm, sẽ chỉ khiến nó càng chập hơn.

36. Luật Dijkstra trong CNPM nói gì?

+) kiểm thử một chương trình rất dễ chỉ ra rằng chương trình có lỗi, nhưng rất khó để chứng tỏ rằng chương trình không còn lỗi.

- tính hữu dụng.

- tính tin cậy.

- tính ổn định.

- tính hiệu quả.

- tính đúng đắn.

37. Verification và Validation (V&V) là gì?

- Verification là quá trình xác định phần mềm đã được phát triển đúng theo yêu cầu, quy chuẩn và quy trình đã định.

- Validation: là quá trình xác định xem phần mềm đã được phát triển có đáp ứng yêu cầu và mong đợi của người dùng.

38. Thế nào là inspection?

- kiểm tra, thẩm định lại tất cả các tài liệu.

- thống kê lỗi theo mức độ, số lượng.

39. Thế nào walkthrough?

- là sự rà soát

- là hướng tài liệu chứ không hướng vào người tham gia.

- mục tiêu là phát hiện lỗi.

- kết quả walkthrough không bao giờ được dùng để đánh giá hiệu năng người thực hiện.

40. Thế nào là một moderator trong nhóm inspection?

- là người quả lý chung.

41. Thế nào là một recorder trong nhóm inspection?

- là người ghi lại các phần sai sót.

42. Mô hình CMM là gì?

- mô hình phát triển và đánh giá quy trình phần mềm.

Được xây dựng trên các phần như sau:

1. Initial: mới bắt đầu
2. Managed: quản lý
3. Defind: định nghĩa
4. Quantitatively managed: quản lý định lượng.
5. Optimizing( tối ưu hóa):

43. Thế nào là test performance?

- là quy trình kiểm thử để đánh giá và đo lường hiệu suất và khả năng của một ứng dung, hệ thống hoặc thành pohaanf phần mềm.  
- các yếu tố:

1. thời gian phản hội

2. tải công suất

3. thời gian phục hồi

4. tài nguyên hệ thống.

44. Thế nào là test robustness?

- là một phương pháp kiểm thử để đánh giá và đảm bảo tính ổn định, khả năng chống chịu trong một môi trường không ổn định hoặc hệ thống dữ liệu không hợp lệ.

Các yếu tố:

1. xử lý lỗi.
2. kiểm tra dữ liệu đầu vào
3. tolerant testing: khả năng phần mềm hoạt động chịu được như nào
4. stress testing: kahar năng phần mềm chịu áp lực cao như nào.

45. Thế nào là cone of uncertainty?

- là khái niệm trong quản lý phần mềm, đặt biệt trong ước lượng và dự đoán thời gian.

- nó đề cập đến việc mô phỏng sự biến đối và giảm thiểu không chắc chắc trong quá trình phát triển phần mềm.

46. Thế nào là norminal effort?

- nó được dịnh nghĩa là nỗ lực bình thương là để chỉ số công sức ước tính cần thiết để hoàn thành nhiệm vụ, tác vụ của một dự án trong điều kiện bình thường.

47. Thế nào là phần mềm organic?

- là chỉ các dữ án mà yêu cầu và phạm vi không xác định hoàn toàn từ đầu. trong phần mềm thì các thông tin, phát triển không được kiểm soát hoặc khó dự đoán.

- đặc điểm:

1. môi trường không ổn định.

2. sự linh hoạt và thích ứng.

3. tính tương tác.

48. Thế nào là phần mềm embeded?

- phần mềm nhúng là một loại phần mềm được thiết kế và tích hợp vào các thiết bị điện tử hoặc hệ thống nhúng.

49. Thế nào là phần mềm semi-detached?

50. Thế nào là TCF? (technical complexity factor)

- là yếu tố được sử dụng để đánh giá độ phức tạp kĩ thuật của một dự án phần mềm.

51. Thế nào là UFP? (unadjusted function point):

- là phương pháp đo lường kích thước phần mềm dựa trên chức năng của nó.

52. Thế nào là flow trong FFP?

Flow trong FFP có thể được định nghĩa là sự di chuyển hoặc dòng chảy của thông tin, dữ liệu hoặc các hoạt động trong một chức năng.

53. Thế nào là process trong FFP?

Process thường được sử dụng để chỉ các chức năng hoặc quy trình mà người dùng có thể thực hiện trong hệ thống.

54. Tại sao không có pha kiểm thử?

55. Tại sao không có pha làm tài liệu?

56. Tại sao không có pha lập kế hoạch?

- vì mô sự phát triển linh hoạt và cập nhật thông tin mới, nâng cấp phần mềm lên nhiều. nên sự lập kế hoạch giống như không thực tế lắm, chúng ta phải thay thế bằng việc thực hiện kế hoạch theo từng vòng hoặc chu kì phát triển.

57. Nếu không áp dụng các mô hình vòng đời phần mềm thì có phát triển được phần mềm không? Tại sao?

58. Tại sao người ta phải dùng nhiều mô hình vòng đời khác nhau để phát triển phần mềm?

Vì mỗi phần mềm thì cũng cần nhiều loại mô hình để phát triển vì mỗi loại mô hình có nhiều ưu nhược điểm khác nhau.

59. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu thác nước?

Ưu điểm:

- dễ hiểu và triển khai.

- kiểm soát rủi ro

- phù hợp với dự án có yêu cầu ổn định.

Nhược điểm:  
- thiếu linh hoạt.

- khó khăn trong xác định yêu cầu ban đầu.

- khó trong việc đáp ứng sự thay đổi của khách hàng.

60. Mô hình vòng đời phần mềm kiểu thác nước thì phù hợp với những dự án có đặc điểm gì?

- yêu cầu xác định rõ ràng và ổn định.

- các công nghiệp và phương pháp đã được phát triển rõ ràng.

- khác hàng ko đòi hỏi trong sự tham gia liên tục trong sự phát triển.

- dự án nhỏ có sự ràng buộc về thời gian.

61. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu bản mẫu nhanh?

Ưu điểm:  
- phát triển nhanh

- tính tương tác cao

- sự linh hoạt và thích ứng.

Nhược điểm:

- yêu cầu phản hồi nhanh

- giới hạn quy mô

- rủi ro về chất lượng

62. Mô hình vòng đời phần mềm kiểu bản mẫu nhanh thì phù hợp với những dự án có đặc điểm gì?

- yêu cầu thay đổi thường xuyên.

- khả năng tương tác cao với khách hàng.

-dự án có phạn vi rõ ràng và giới hạn

- quy trình phát triển linh hoạt và thích ứng.

63. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu lặp và tăng trưởng?

Ưu điểm:

- linh hoạt.

- tạo ra giá trị sớm.

- giảm rủi ro.

Nhược điểm:

- đòi hỏi phản hồi tích cực.

- đòi hỏi tài nguyên và thời gian.

- quản lý phức tạp.

64. Mô hình vòng đời phần mềm kiểu lặp và tăng trưởng thì phù hợp với những dự án có đặc điểm gì?

- yêu cầu thay đổi thường xuyên

- sự không chắc chắc và sự phát triển độc lập.

- sản phẩm có thể phát triển từng phần.

- tình tương tác và phản hồi từ khách hàng.

65. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu xoắn ốc?

Ưu điểm:

- định hướng rủi ro.

- tính linh hoạt.

- điều chỉnh yêu cầu.

Nhược điểm:

- phức tạp quản lý.

- yêu cầu nguồn lực.

- khó khăn trong ước lượng.

66. Mô hình vòng đời phần mềm kiểu xoắn ốc thì phù hợp với những dự án có đặc điểm gì?

- dự án có mức độ phức tạp cao.

- yêu cầu không rõ ràng hoặc có khả năng thay đổi.

- dự án đòi hỏi kiểm soát rủi ro chặt chẽ.

- sự tương tác và phản hồi liên tục từ khách hàng.

- dự án có tính linh hoạt và sáng tạo cao.

67. Nêu ưu điểm, nhược điểm của mô hình vòng đời phần mềm kiểu tiến trình linh hoạt?

Ưu điểm:

- tính linh hoạt.

- tính tăng cường tương tác và sự hợp tác.

- tạo ra giá trị sớm.

- giảm rủi ro.

Nhược điểm:

- đòi hỏi sự tham gia và cam kết.

- khả năng quản lý yêu cầu.

- khản năng mở rộng

- đòi hỏi sự tương tác liên tục.

68. Mô hình vòng đời phần mềm kiểu tiến trình linh hoạt thì phù hợp với những dự án có đặc điểm gì?

- yêu cầu thay đổi và không chắc chắn.

- sự tương tác liên tục với khách hàng.

- nhóm phát triển tự quản lý và chủ đông.

- nhóm phát triển tự quả lý và chủ động.

- quy mô dự án quản lý từ nhỏ đến trung bình.

69. Trong mô hình tiến trình liên hoạt, luôn có đại diện của khác hàng trong nhóm phát triển thì có ưu điểm gì?

70. Nêu ưu điểm, nhược điểm của mô hình nhóm code bình đẳng?

71. Mô hình nhóm code bình đẳng thì phù hợp với những dự án có đặc điểm gì?

72. Nêu ưu điểm, nhược điểm của mô hình nhóm code có chef?

73. Mô hình nhóm code có chef thì phù hợp với những dự án có đặc điểm gì?

74. Nêu ưu điểm, nhược điểm của kĩ thuật pair programming?

75. Kĩ thuật pair programming thì phù hợp với những dự án có đặc điểm gì?

76. Nêu ưu điểm, nhược điểm của kĩ thuật time boxing?

77. Nêu ưu điểm, nhược điểm của kĩ thuật stand up meeting?

78. Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng LOC?

Ưu điểm:

- đơn giản: phương pháp ước lượng dựa trên LOC đơn giản và dễ hiểu. Nó chỉ đơn thuần đếm số dòng code trong dự án để ước lượng kích thước và công sức.

- Dễ thực hiện: việc đếm số lượng dòng code có thể thực hiện bằng công cụ phân tích code hoặc thông qua các công cụ quản lý phiên bản.

- dễ so sánh: với việc sử dụng LOC, có thể so sánh các dự án phần mềm với nhau dựa trên kích thước của chúng.

- phù hợp với việc ước lượng công sức: LOC có thể được sử dụng để ước lượng công sức thiết cho việc phát triển phần mềm.

Nhược điểm:

- không đo lường được chất lượng.

- không linh hoạt.

- không đồng nhất.

- không xem xét yếu tố ngữ cảnh.

79. Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng FFP?

Ưu điểm:

- đơn giản và dễ hiểu:

- chuẩn hóa:

- phù hợp cho việc so sánh dự án.

- cung cấp ước lượng sớm:

Nhược điểm:

- khó định nghĩa các điểm chức năng.

- không đo lường chất lượng

- Ảnh hưởng thay đổi yêu cầu.

- Phụ thuộc vào kinh nghiệm và hiểu biết.

80. Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng Function Point?

Ưu điểm:

- đo lường chức năng.

- đánh giá giá trị kinh tế.

- dễ hiểu và sử dụng.

- khả năng dự báo.

Nhược điểm:

- đòi hỏi kiến thức và kinh nghiệm.

- đánh giá chủ quan

- khó khăn trong việc đo lường chất lượng.

- phụ thuộc và đặc tả yêu cầu.

81. Nêu ưu điểm, nhược điểm của phương pháp ước lượng phần mềm bằng COCOMO?

Ưu điểm:

- đánh giá tin cậy và được chứng minh.

- dễ sự dụng

- phù hợp cho ước lượng sớm.

- tính linh hoạt.

Nhược điểm:

- yêu cầu dữ liệu đầu vào chính xác.

- phù thuộc vào kinh nghiệm.

- không xem xét một số yếu tố quan trong.

- giới hạn trong ước lượng dự án lớn.

82. Tại sao trong mô hình tiến trình linh hoạt, không cần có pha đặc tả?

83. Tại sao trong nhóm walkthrough và inspection, luôn phải có đại diện của workflow tiếp theo?

84. Nếu nhóm SQA phát hiện ra ít lỗi, thì có thể kết luận nhóm code giỏi hay nhóm SQA kém? Tại sao?

85. Tại sao nói inspection và walkthrough là hướng tài liệu, mà không phải hướng vào người tham gia?

86. Quality assurance thì khác gì với testing?

87. Tại sao nói function point chịu ảnh hưởng chủ quan của các chuyên gia?

88. COCOMO tính đến nhiều tiêu chí hơn hay là function point? Giải thích?

89. SW development multiplier của COCOMO thì khác gì TCF của function point? 4

90. TCF của function point thì khác gì hằng số b của FFP?

91. Tại sao nguyên lí Djiktra lại đúng?

92. Tại sao luật Brook lại đúng?

93. Người ta áp dụng luật Miller trong CNPM như thế nào?

94. Phát triển phần mềm thì khác gì sản xuất phần mềm?

95. Test trường hợp sai kiểu dữ liệu đầu vào thì thuộc thể loại test gì?

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA AN TOÀN THÔNG TIN 1**

****

**Môn: Nhập Môn Công Nghiệp Phần Mềm**

**BÁO CÁO BÀI TẬP LỚN**

**Họ và tên sinh viên:**

**Đỗ Tiến Sĩ**

**Mã số sinh viên:**

**B20DCAT153**

**Họ và tên giảng viên:**

**Thầy: Đỗ Thị Ngọc**

**Hà Nội 5/2023 (tháng/năm)**

1. **Pha lấy yêu cầu:**
2. Các từ chuyên ngành:

* hệ thống quản lý kết quả học tập của sinh viên: tên của hệ thống
* sinh viên: người sử dụng phần mềm để xem xét trong quá trình học tập.
* quản lý : người giáo vụ phụ trách việc quản lý việc quản lý học phần và lên lịch thời khóa biểu cho giáo viên.
* Giáo viên: người giảng dạy và có chức năng nhập điểm cho sinh viên.
* Môn học: là môn học mà student đăng kí học và teacher đăng kí dạy
* Lớp học phần : là thứ mà chỉ ra phòng học và thời gian mà giáo viên dạy môn học đó hoặc là sinh viên học ở đâu vào thời gian nào.
* Quản lý sinh viên: quản lý sinh viên.
* quản lý môn học: là thứ giúp quản lý kiểm soát môn học.
* nhập điểm: là hành động đưa điểm môn học dành cho giáo viên người sử dụng
* phòng học: là đối tượng mà xác định địa điểm của sinh viên học.
* xem thời khóa biểu
* lên thời khóa biểu là hoạt động người quản lý có thể tạo ra thời khóa biểu
* người quản lý hệ thống: có quyền quản lý tài khoản.
* đăng kí lớp học phần của sinh viên là sự kiện đăng kí môn học cho sinh viên
* Môn học: được student or teacher học hoặc dạy
* đăng nhập là hoạt động tham gia vào hệ thống.

1. Mô tả từng người:

+) Sinh viên được phép:

* Xem thời khóa biểu của mình.
* Đăng kí môn học.
* Đăng kí lớp học phần.
* Xem điểm của mình.

+) Giáo viên được phép:

* Đăng kí môn học.
* Đăng kí lớp học phần.
* Nhập điểm.

+) Quản lý hệ thống:

* Quản lý tài khoản của toàn hệ thống.
* Được xem báo cáo.

+) Quản lý:

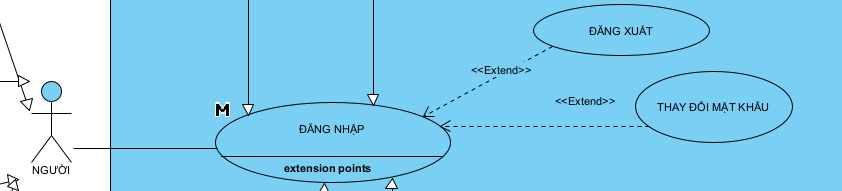
* Quản lý phòng học.
* Quản lý môn học.
* Quản lý lớp học phần.
* Quản lý lịch học môn học.

Mô tả các use case:

* Người sử dụng

Ta quy ước quản lý hệ thống, người quản lý, giáo viên, quản lý , sinh viên là người sử dụng

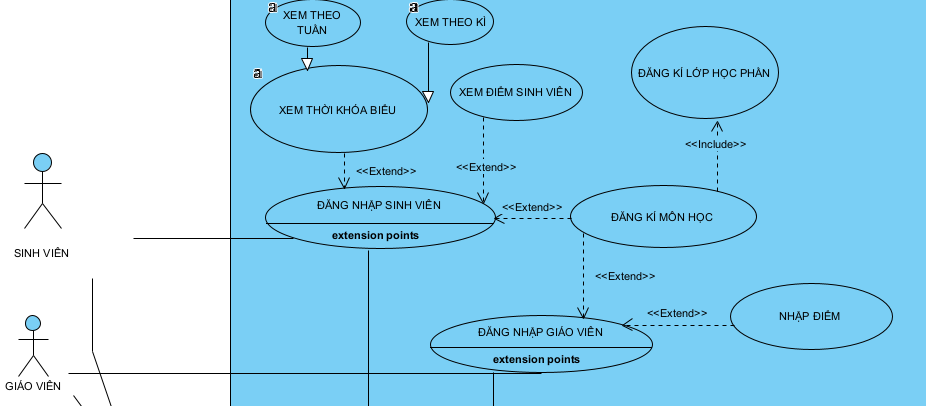
* Đăng nhập: use case này cho phép tài khoản người sử dụng đăng nhập vào tài khoản của mình.
* Đổi mật khẩu: use case này cho phép các người sử dụng thay đổi password
* Đăng xuất: cho phép người sử dụng sau khi hoàn thành việc thì có thể đăng xuất.



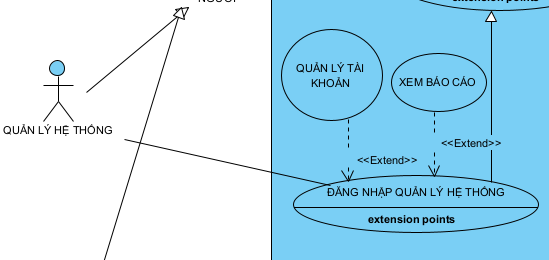
* Sinh viên:
* Đăng nhập: để xuất hiện được menu của student sau khi đăng nhập thì ta gọi đó là uc sinh viên đăng nhập.
* Xem thời khóa biểu: xem thời khóa biểu sau khi đăng kí môn học.
* Đăng kí môn học: use case này cho phép sinh viên thêm xóa các môn học từ hệ thống vào kì học của mình.
* Đăng kí lớp học phần: use case này cho phép sinh viên thêm, xóa các lớp học phần từ môn học mà mình đã đăng kí.
* Xem điểm:use case này cho phép sinh viên xem điểm các môn học mà mình đã đăng kí học.

|  |
| --- |
|  |

* Giáo viên
* Đăng nhập: đề xuất hiện menu của giáo viên ngay sau khi login thì tao gọi uc đăng nhập giáo viên
* Đăng kí môn học: uc này cho phép giáo viên thêm, xóa các môn học từ hệ thống.
* Đăng kí lớp học phần: uc cho phép giáo viên thêm, xóa các lớp học phần từ các môn học mà đã đăng kí.



* Quản lý hệ thống:
* Đăng nhập: để đề xuất menu của người quản lý ngay sau khi login thì ta gọi uc đăng nhập quản lý hệ thống.
* Xem báo cáo: uc này cho phép người quản lý hệ thống xem được các báo cáo của hệ thống ví dụ: thống kê sinh viên qua môn, thông kê thành tích học tập, báo cáo đăng kí môn học, ….



* Quản lý:
* Đăng nhập: để đề xuất ra menu của quản lý ngay sau khi login thì ta gọi uc đăng nhập quản lý.
* Quản lý môn học: uc cho phép người quản lý thêm sửa xóa các môn học trong hệ thống.
* Quản lý học phần: uc cho phép người quản lý thêm sửa xóa các lớp học phần trong hệ thống.
* Quản lý lịch học môn học: uc cho phép người quản lý thêm sửa xóa lịch học của lớp học phần trong môn học.

|  |
| --- |
|  |

1. Mịn hóa mô hình BM:  
   Xem xét lại các use case:

Giáo viên và sinh viên đều có cùng uc đăng kí môn học và đăng kí lớp học phần.

* Có thể gộp chung lại thành uc đăng kí môn học và đăng kí lớp học phần cho toàn cả sinh viên và giáo viên.

Use case tổng quan cho toàn bộ hệ thống:

|  |
| --- |
|  |
|  |

1. **Pha phân tích**
2. Mô hình hóa chức năng:

Scenario chuẩn cho xem thời khóa biểu của sinh viên:

* 1. Sinh viên A chọn chức năng xem thời khóa biểu sau khi login. A muốn xem thời khóa biểu của kì học.
  2. Giao diện thời khóa biểu hiện ra với 2 nút: xem theo tuần , xem theo kì học.
  3. Sinh viên A nhấn chọn xem theo tuần và click nút xem.
  4. Trang xem thời khóa biểu hiện ra thời khóa biểu với 1 bảng gồm 7 cột tương ứng 7 ngày, 6 hàng tương ứng với 6 kíp học mỗi ngày. Mỗi ô tương ứng hiển thị tên môn, nhóm môn, phòng học và khung giờ tương ứng.

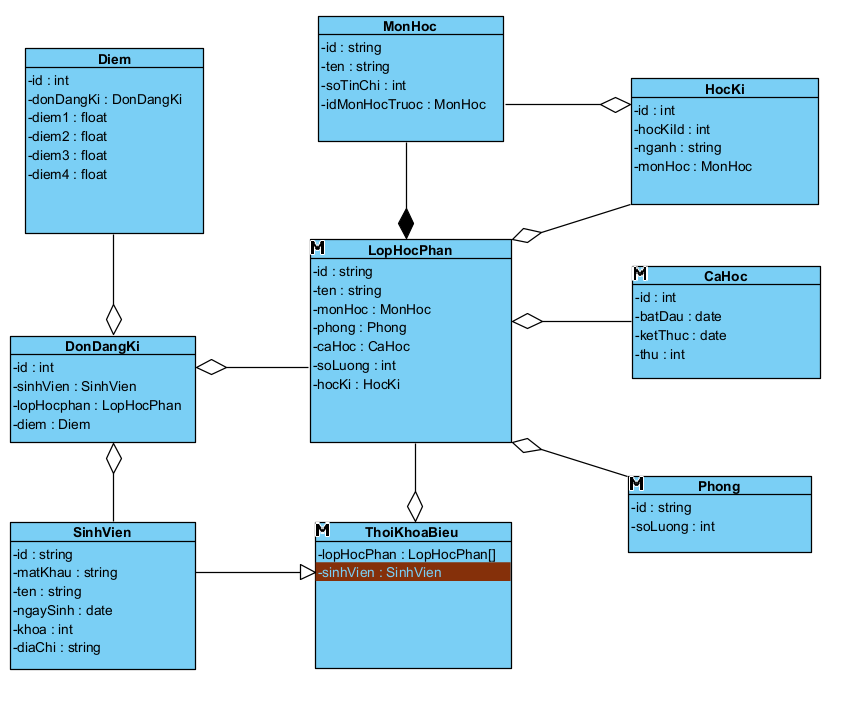
1. Trích lớp thực thể.

Đề bài mô tả như sau:

* Mỗi sinh viên (Mã SV, mật khẩu, tên, ngày sinh, khóa, quê quán, địa chỉ) được phép đăng kí tối thiểu 10 tín chỉ/học kì và tối đa 15 tín chỉ/học kì
* Mỗi sinh viên được đăng kí nhiều môn học (mã môn, tên môn, số tín chỉ)
* Mỗi môn học có thể có nhiều môn học yêu cầu sinh viên phải hoàn thành trước đó thì mới được đăng kí
* Mỗi môn học có thể có nhiều lớp học phần (mã lớp, tên lớp, số sv tối đa, phòng học, khung giờ học cố định trong tuần)
* Sinh viên không được phép đăng kí học hai lớp có trùng buổi học
* Với mỗi môn học, một sinh viên chỉ được đăng kí vào 1 lớp xác định
* Kết quả của sinh viên (điểm thành phần số 1, số 2, số 3, điểm thi, điểm cuối cùng=x% số1+ y% số2 + z% số3 + w% điểm thi) được lưu theo từng môn học.
* Điểm trung bình của sinh viên trong học kì được tính bằng trung bình có trọng số là số tín chỉ từng môn học.
* modul "Xem TKB của sinh viên" với các bước sau đây: SV chọn menu xem TKB → Giao diện xem TKB hiện lên với phía trên là ô chọn các cách xem TKB theo: tuần, học kỳ → SV chọn xem theo tuần → Phía dưới cập nhật hiển thị thời khóa biểu theo tuần hiện tại của SV: 1 bảng có 7 cột tương ứng 7 ngày, 6 hàng tương ứng 6 kíp học cho mỗi ngày. Trong mỗi ô của bảng hiển thị tên môn học, nhóm môn học, và tên phòng học tương ứng với khung giờ đó.

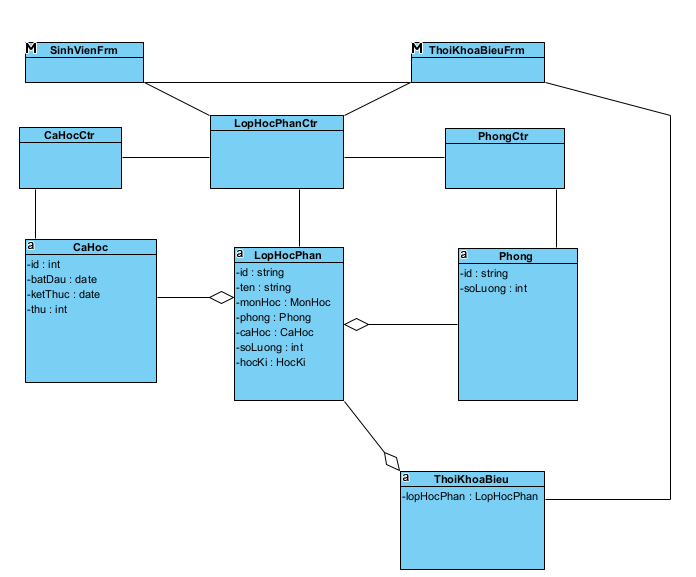
Từ những lớp ở trên và suy luận với modul mà ta cần làm thì ta có các entity sau:

* MonHoc: lớp môn học
* HocKi: lớp học kì.
* CaHoc: lớp ca học.
* LopHocPhan: lớp lớp học phần.
* PhongHoc: lớp phòng học.
* ThoiKhoaBieu: lớp thời khóa biểu
* SinhVien: lớp sinh viên
* Diem: lớp điểm
* DonDangKi: lớp đăng kí lớp học phần.

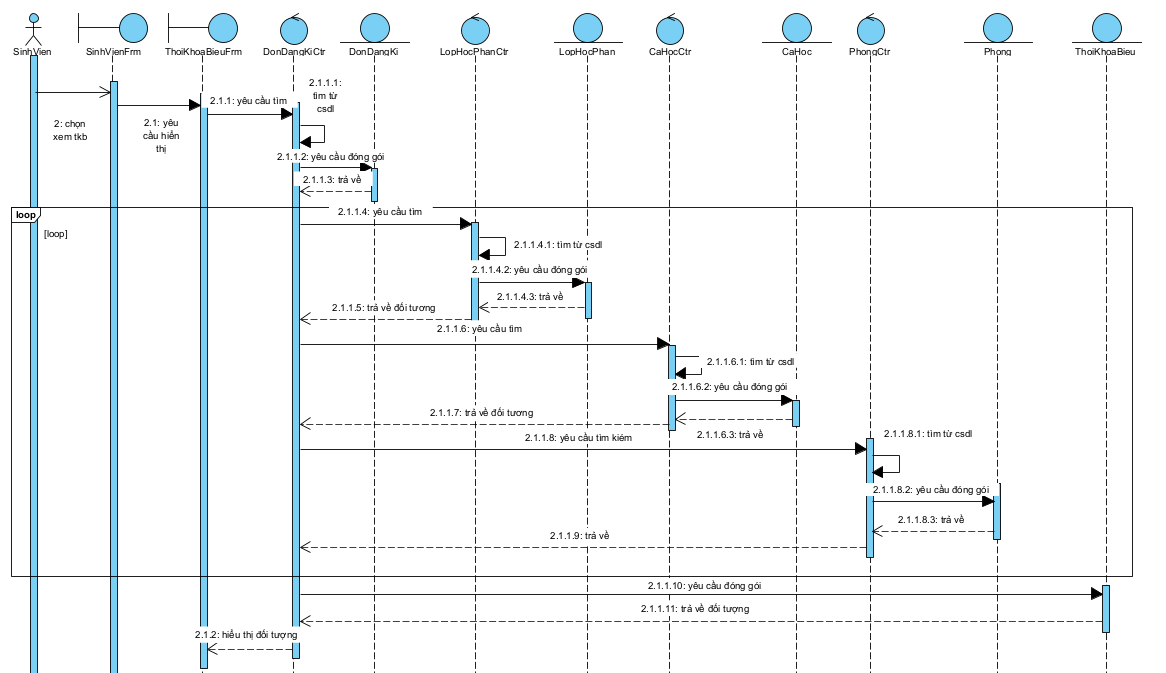


1. Trích lớp điều khiển.

* Lớp điều khiển:
* Lớp điều khiển cho modul MonHoc: MonHocCtr.
* Lớp điều khiển cho modul LopHocPhan: LopHocPhanCtr.
* Lớp điều khiển cho modul HocKi: HocKiCtr.
* Lớp điều khiển cho modul CaHoc: CaHocCtr.
* Lớp điều khiển cho modul Phong: PhongCtr.
* Lớp điều khiển cho modul DonDangKi: DonDangKiCtr.
* Lớp điều khiển cho modul SinhVien: SinhVienCtr.
* Lớp điều khiển cho modul Diem: DiemCtr.
* Đề xuất lớp biên cho module xem thời khóa biểu của sinh viên:
* Giao diện chính: SinhVienFrm
* Chắc năng xem: form xem (ThoiKhoaBieuFrm)



1. Sequence Diagram

+

Scenario bước pha phân tích:  
1. Sinh viên A chọn chức năng xem thời khóa biểu.  
2. Lớp SinhVienFrm hiện ra 2 nút: là xem theo tuần hoặc xem theo học kì.

3. Sinh viên A click vào nút xem theo tuần.

4. Lớp SinhVienFrm gọi lớp ThoiKhoaBieuFrm yêu cầu hiển thị.

5. Lớp ThoiKhoaBieuFrm gọi lớp DonDangKiCtr để yêu cầu tìm kiếm các lớp học phần mà sinh viên đã đăng kí.

6. Lớp DonDangKiCtr tìm kiếm các lớp học phần mà sinh viên đã đăng kí trong cơ sở dự liệu.

7. Lớp DonDangKiCtr gửi đến lớp DonDangKi để yêu cầu đóng gói thành danh sách các đối tượng DonDangKi.

8. Lớp DonDangKi gửi trả danh sách đối tượng donDangKi cho lớp DonDangKiCtr

9. Lớp DonDangKiCtr gửi yêu cầu tìm kiếm đến LopHocPhan ứng với các DonDangKi trả về.

10. Lớp LopHocPhanCtr tìm kiếm các section trong cơ sở dữ liệu tương ứng.

11. Lớp LopHocPhanCtr gửi yêu cầu đóng gói đến lớp LopHocPhan.

12. Lớp LopHocPhan đóng gói trả về đối tượng cho lớp LopHocPhanCtr.

13. Lớp LopHocPhanCtr trả về đối tượng section cho lớp DonDangKiCtr.

14. Lớp DonDangKiCtr gửi yêu cầu tìm kiếm tới lớp CaHocCtr để tìm kiếm các đối tượng caHoc tương ứng với đối tượng LopHocPhan được trả về.

15 Lớp CaHocCtr tìm kiếm đối tượng trong cơ sở dữ liệu.

16. Lớp CaHocCtr gửi yêu cầu đóng gói đến lớp CaHoc yêu cầu đóng gói.

17. Lớp CaHoc trả về đối tượng cho lớp CaHocCtr.

18. Lớp CaHocCtr trả về đối tượng cho lớp DonDangKiCtr

19. Lớp DonDangKiCtr gửi yêu cầu tìm kiếm lớp PhongCtr tương ứng với đối tượng LopHocPhan được trả về phía trên.

20. Lớp PhongCtr tìm kiếm đối tượng trong cơ sở dữ liệu.

21. Lớp PhongCtr yêu cầu đóng gói gửi đến lớp Phong.

22. Lớp Phong trả về đối tượng cho lớp PhongCtr.

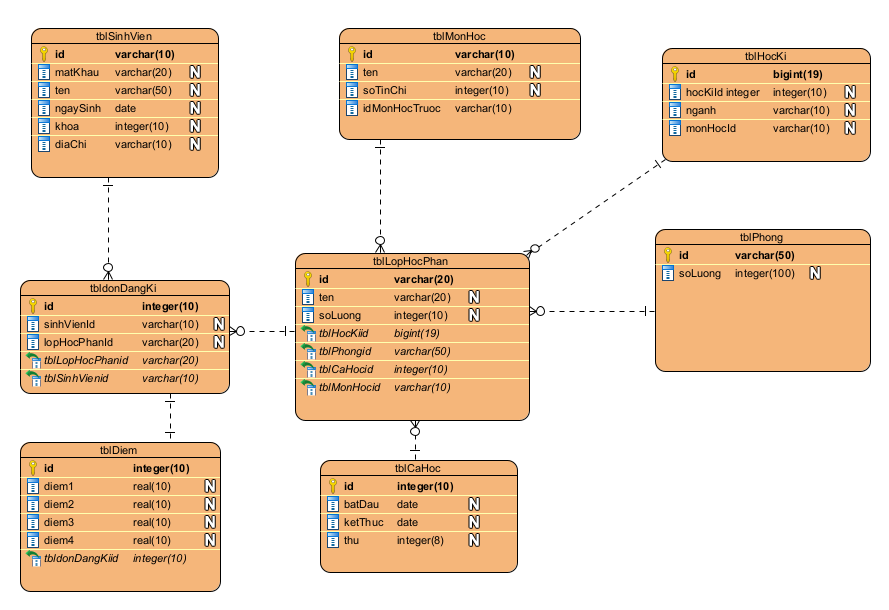
22. Lớp PhongCtr gửi đối tượng trả về cho lớp DonDangKiCtr.

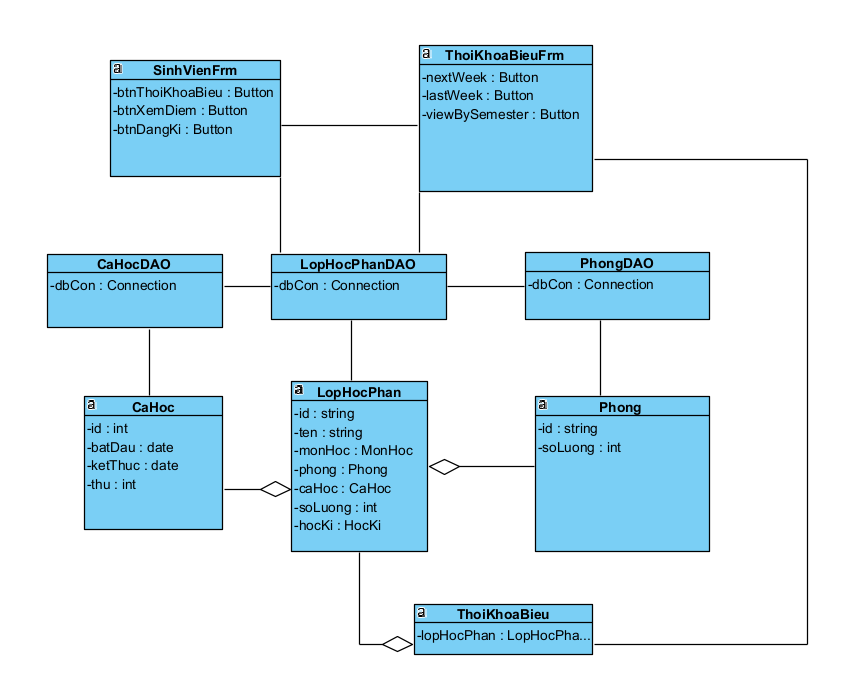
23. Lớp DonDangKiCtr gửi yêu cầu đóng gói tới lớp ThoiKhoaBieu.

24. Lớp ThoiKhoaBieu đóng gói và trả về đối tượng cho lớp DonDangKiCtr.

25. Lớp DonDangKiCtr gửi thông tin cho lớp ThoiKhoaBieuFrm thông tin cần.

1. Pha thiết kế:
2. Thiết kế cơ sở dữ liệu:



2. Mô hình MVC thiết kế:  
 

1. Pha cài đặt.

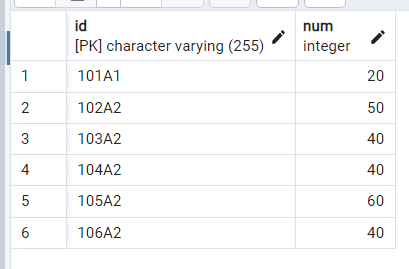
Em không code được phần phần kia phần mềm nên em xin show code trong chương trình ạ.

1. Pha Kiểm thử:

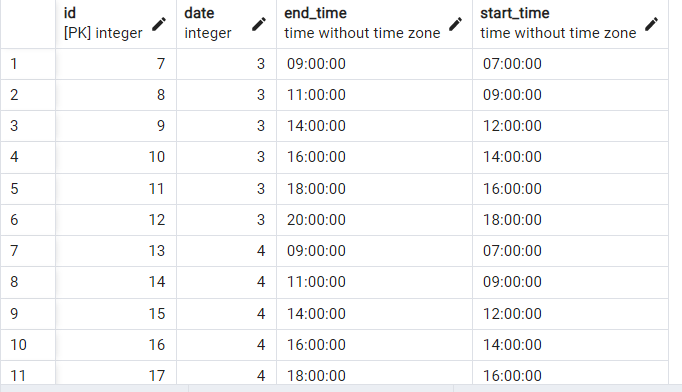
Test 1:

* 1. Bảng dữ liệu hiện thời:

Bảng Phong:



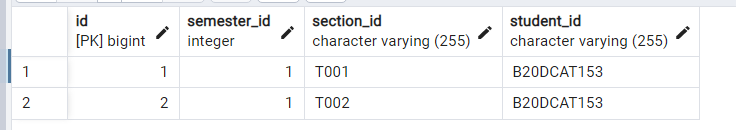
Bảng CaHoc:



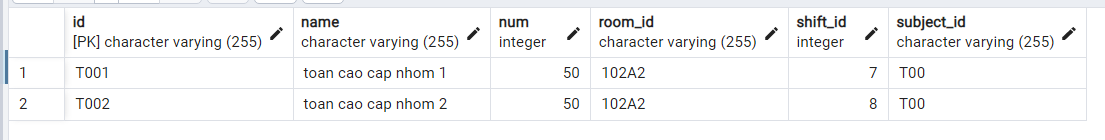
Bảng HocSinh:



Bảng DonDangKi:



Bảng LopHocPhan:



Kết quả mong muốn cho sinh viên “B20DCAT153”:

