HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA: CÔNG NGHỆ THÔNG TIN 1

NGÂN HÀNG CÂU HỎI THI THỰC HÀNH - VẤN ĐÁP

Tên học phần:	**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**
	INTRODUCTION TO SOFTWARE ENGINEERING
Mã học phần:	**INT1340_CLC**
Ngành đào tạo:	**CÔNG NGHỆ THÔNG TIN CHẤT LƯỢNG CAO**
Trình độ đào tạo:	**Đại học chính quy**

Cấu trúc đề

Mỗi đề có hai phần:

- Phần lí thuyết 2,5 điểm, do giáo viên hỏi ngay khi chấm phần bài tập của sinh viên
- Phần bài tập thực hành: 7,5 điểm, có dạng một bài tập hoàn chỉnh. Mỗi đề có 5 câu hỏi liên quan đến các bước của công nghệ phần mềm. Mỗi câu 1,5 điểm.

Thời gian chung cho tất các các đề là 60 phút.

Ngân hàng câu hỏi thi này đã được thông qua bộ môn và nhóm cán bộ giảng dạy học phần.

Hà Nội, ngày tháng năm 2022

Trưởng khoa	**Trưởng bộ môn**	**Giảng viên chủ trì biên soạn**
Nguyễn Duy Phương	Nguyễn Mạnh Hùng	Nguyễn Mạnh Hùng, Đỗ Thị Bích Ngọc

Theory section

- Each student will asked by 3 different questions, from easiest to hardest
- After a successful answer, the next question have to be more difficult than the last asked.
- After a failed answer, the next question sould be the same difficult as the last asked.

Questions of SE

1. What is corrective maintenance?
2. What is adaptive maintenance?
3. What is perfective maintenance?
4. What is refactoring?
5. What is "from scratch"?
6. What is the moving target problem?
7. What is regession fault?
8. What is an episode?
9. What is an iteration?
10. What is an increasement?
11. What is an artifact?
12. What is portability?
13. What is reusebility?
14. What is milestone?
15. What is a story?
16. What is concept exploration?
17. What is business model?
18. What is traceability?
19. What is egoless programming?
20. What is Project Manager?
21. What is technical leader?
22. What is programming secrectary?
23. What is backup programmer?
24. What is supper programmer?
25. What is an ommision analyse?
26. What is a contradiction in an analyse?
27. What is COTS?
28. What is SPMP?
29. What is alpha release?
30. What is beta release?
31. What is a process?
32. What is a workflow?
33. What is Miller law in SE?
34. What is Brooks law in SE?
35. What is Dijkstra law in SE?
36. What is Verification and Validation (V&V)?
37. What is inspection?
38. What is walkthrough?
39. What is a moderator in an inspection team?
40. What is a recorder in an inspection team?
41. What is CMM?
42. How to take a test performance?
43. How to take a test robustness?
44. What is cone of uncertainty?
45. What is norminal effort?
46. What is an organic software?
47. What is an embeded software?

48. What is a semi-detached software?
49. What is TCF?
50. What is UFP?
51. What is a flow in the FFT model?
52. What is a process in the FFP model?
53. Why there is no testing phase in the software life-cycle model?
54. Why there is no doccumentation phase in the software life-cycle model?
55. Why there is no planning phase in the software life-cycle model?
56. Could we develop a software without applying any software life-cycle model? Why?
57. Why many software life-cycle models are used?
58. Advanatage/disadvantage of the waterfall model?
59. Which type of software that the waterfall is suitable for?
60. Advanatage/disadvantage of the rapid prototype model?
61. Which type of software that the rapid prototype is suitable for?
62. Advanatage/disadvantage of the iteration and inscreasement model?
63. Which type of software that the iteration and inscreasement is suitable for?
64. Advanatage/disadvantage of the spiral model?
65. Which type of software that the spiral is suitable for?
66. Advanatage/disadvantage of the agile process?
67. Which type of software that the agile process is suitable for?
68. In the agile process, what is the advantage of the fact that thereis always a representative of client in the development team?
69. Advanatage/disadvantage of the democractive team?
70. Which type of software project that the democractive team is suitable for?
71. Advanatage/disadvantage of the chief-programing team?
72. Which type of software project that the chief-programing team is suitable for?
73. Advanatage/disadvantage of the pair programming?
74. Which type of software project that the pair programming is suitable for?
75. Advanatage/disadvantage of the time-boxing?
76. Advanatage/disadvantage of the stand up meeting?
77. Advanatage/disadvantage of the LOC?
78. Advanatage/disadvantage of the FFP?
79. Advanatage/disadvantage of the Function Point?
80. Advanatage/disadvantage of the COCOMO?
81. Why in the agile process, there is no specification phase?
82. Why in the walkthrough and inspection team, there is always a representative of the next workflow?
83. If the number of error founded by SQA is small, could we say that the SQA team is not good or the development team is good? Why?
84. Why do we say that inspection and walkthrough is doccument-orieted, but not participant-oriented?
85. What is the difference betwwen quality assurance and testing?
86. Why do they say that the function point is influenced by the expert subjective?
87. COCOMO or function point covers more citeria? Why?
88. What is the difference of the cost multiplier in the COCOMO in compare to the TCF of the function point?
89. What is the difference of the TCF of the function point in compare to the b constant of the FFP?
90. Why does the Djiktra law hold?
91. Why does Brook law hold?

92. How the Miller law is applied in the SE?
93. When a tester tries to enter the wrong value at the input of a variable, which method of test is being taken?

Project

- Duration: 60m
- Note: 1.5point/question. Total 7.5/10.

Software engineering
Subject No. 01
Duration: 60 minutes

A client requires us to develop a library management software, described as follows:

- Each book title (Code, name, author, publication year, cover price, quantity, barcode, description) can be borrowed many times by many different readers.
- Each reader has a reader card containing the reader's code, name, date of birth, address, phone number, barcode
- Up to 5 books can be borrowed at a time, and the total number of books being borrowed by one person cannot exceed 5 books
- The maximum duration to borrow a book is 1 month from the date of borrowing that book, if the reader returns it after this time, he will be fined 20% of the book cover price.
- Each time a reader returns borrowed books, he can return part or all of the borrowed books
- When borrowing new books, the librarian can still see a list of books that the reader has borrowed and paid or not paid before.

modulee "**Borrowing of books**": A staff chooses the book borowing menu → scans the reader card to get reader information → reader detail information appears + list unreturned books + returned book list → The staff scans selected books one by one → the list of borrowed books is added until the books to borrow (or up to 5 books) are exhausted, then submit → print out a loan slip containing the code, name, reader barcode, bar code of the loan slip, and a list of books that are still borrowed, each title book on one line: code, title, author, barcode, loan date, due date and The last line shows the total number of books being borrowed.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 02
Duration: 60 minutes

A client requires us to develop a library management software, described as follows:

- Each book title (Code, name, author, publication year, cover price, quantity, barcode, description) can be borrowed many times by many different readers.
- Each reader has a reader card containing the reader's code, name, date of birth, address, phone number, barcode
- Up to 5 books can be borrowed at a time, and the total number of books being borrowed by one person cannot exceed 5 books
- The maximum duration to borrow a book is 1 month from the date of borrowing that book, if the reader returns it after this time, he will be fined 20% of the book cover price.
- Each time a reader returns borrowed books, he can return part or all of the borrowed books
- When borrowing new books, the librarian can still see a list of books that the reader has borrowed and paid or not paid before.

modulee "**Returning of books**": A staff selects the return menu → scans the reader card to get reader information → reader detail information appears + a list of unreturned books + returned borrowed books → The staff scan returned books in turn → the list of borrowed books is shortened until the end of borrowed books (or all the books returned by readers are scanned) then submit → print out a loan slip (if there are still books on loan) containing the code, name, reader barcode, loan card barcode, and a list of books that are still borrowed, each title on one line: code, title, author, barcode, Borrowing date, due date and last line total number of books on loan + penalty slip (if fined) containing code, name, reader barcode, loan coupon barcode, and list of fine (late return books, damaged, lost...), each end books on one line: code, title, author, bar code, date borrowed, due date, pay date, fine amount and last line the total amount of the fine.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 03
Duration: 60 minutes

A client requires us to develop a library management software, described as follows:

- Each book title (Code, name, author, publication year, cover price, quantity, barcode, description) can be borrowed many times by many different readers.
- Each reader has a reader card containing the reader's code, name, date of birth, address, phone number, barcode
- Up to 5 books can be borrowed at a time, and the total number of books being borrowed by one person cannot exceed 5 books
- The maximum duration to borrow a book is 1 month from the date of borrowing that book, if the reader returns it after this time, he will be fined 20% of the book cover price.
- Each time a reader returns borrowed books, he can return part or all of the borrowed books
- When borrowing new books, the librarian can still see a list of books that the reader has borrowed and paid or not paid before.

Module "**Statistics of borrowed books**": The staff selects the statistics menu → selects the statistics of borrowed books → enter the time period (start - end) → the list of borrowed books are displayed in order of the number of loans from most to least, each line contains: code, book title, author, barcode, total number of loans. The staff clicks on a line of a book to display a detailed list of times that reader has borrowed that book, each line contains: reader name, borrowed day, returned day, the fine if any.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 04
Duration: 60 minutes

A client requires us to develop a library management software, described as follows:

- Each book title (Code, name, author, publication year, cover price, quantity, barcode, description) can be borrowed many times by many different readers.
- Each reader has a reader card containing the reader's code, name, date of birth, address, phone number, barcode
- Up to 5 books can be borrowed at a time, and the total number of books being borrowed by one person cannot exceed 5 books
- The maximum duration to borrow a book is 1 month from the date of borrowing that book, if the reader returns it after this time, he will be fined 20% of the book cover price.
- Each time a reader returns borrowed books, he can return part or all of the borrowed books
- When borrowing new books, the librarian can still see a list of books that the reader has borrowed and paid or not paid before.

Module "**Statistics of readers**": A staff selects the statistics menu → selects the statistics of readers → enter the time period (start - end) → list the most readers are displayed in order of the number of books borrowed from most to least, each line contains: code, name, date of birth, reader's address, total number of books borrowed. The staff clicks on a line of a reader, the details of the loan slips with information on the date of borrowing, the total number of books of each loan will appear.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 05
Duration: 60 minutes

A client requires us to develop a software to manage student results, described as follows:

- Each student (Student ID, password, name, date of birth, course, hometown, address) is allowed to register for a minimum of 10 credits/semester and a maximum of 15 credits/semester
- Each student can register for many subjects (subject code, subject name, number of credits)
- Each subject can have multiple subjects that students must complete before they can be registered
- Each subject can have multiple classes (class code, class name, maximum number of students, classrooms, fixed hours of the week)
- Students are not allowed to register for two classes with the same class time
- For each subject, a student is only allowed to enroll in a class
- Student results (part number 1, number 2, number 3, test score, final score=x% number1+ y % number2 + z% number3 + w% test score) are saved for each subject
- The student's grade point average for the semester is calculated as a weighted average of the number of credits per subject

Module "**Schedule a class**": A staff selects the menu to schedule a class → the class scheduler interface appears with the subject and class comboboxes. section, class, time frame, confirmation button → The staff clicks on a subject from the drop-down list → a list of classes in the selected subject has been updated → The staff clicks to add new class of the selected subject → selects classroom from the drop-down list of classrooms + clicks to select the time of the week from the drop-down list of time frames + click confirm → The system saves the class schedule to the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 06
Duration: 60 minutes

A client requires us to develop a software to manage student results, described as follows:

- Each student (Student ID, password, name, date of birth, course, hometown, address) is allowed to register for a minimum of 10 credits/semester and a maximum of 15 credits/semester
- Each student can register for many subjects (subject code, subject name, number of credits)
- Each subject can have multiple subjects that students must complete before they can be registered
- Each subject can have multiple classes (class code, class name, maximum number of students, classrooms, fixed hours of the week)
- Students are not allowed to register for two classes with the same class time
- For each subject, a student is only allowed to enroll in a class
- Student results (part number 1, number 2, number 3, test score, final score=x% number1+ y % number2 + z% number3 + w% test score) are saved for each subject
- The student's grade point average for the semester is calculated as a weighted average of the number of credits per subject

Module "**Enter grades by class**": A teacher selects the function to enter grades → the system displays a list of subjects taught by the teacher → The teacher clicks on a subject → the system shows a list of classes of the selected subject taught by the teacher → The teacher clicks on a class → The system displays a list of students in the course, each student on a line with the component/part score columns and the exam score column → The teacher enters all the scores of the students + click confirm → The system saves it to the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 07
Duration: 60 minutes

A client requires us to develop a software to manage student results, described as follows:

- Each student (Student ID, password, name, date of birth, course, hometown, address) is allowed to register for a minimum of 10 credits/semester and a maximum of 15 credits/semester
- Each student can register for many subjects (subject code, subject name, number of credits)
- Each subject can have multiple subjects that students must complete before they can be registered
- Each subject can have multiple classes (class code, class name, maximum number of students, classrooms, fixed hours of the week)
- Students are not allowed to register for two classes with the same class time
- For each subject, a student is only allowed to enroll in a class
- Student results (part number 1, number 2, number 3, test score, final score=x% number1+ y % number2 + z% number3 + w% test score) are saved for each subject
- The student's grade point average for the semester is calculated as a weighted average of the number of credits per subject

Module "**Register for classes**": Students log in → select the menu to register for the new semester → the registration page appears → students select a subject from the list of subjects + select the class in the list of classes (and associated lecturers) corresponding to the selected subject → repeat until all subjects are registered. If the above constraints are satisfied, then notify the success + print out the registration form for the student: student code, student name, course, semester + list of registered subjects, each subject has: code, name, credit number, class time, lecturer name

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 08
Duration: 60 minutes

A client requires us to develop a software to manage student results, described as follows:

- Each student (Student ID, password, name, date of birth, course, hometown, address) is allowed to register for a minimum of 10 credits/semester and a maximum of 15 credits/semester
- Each student can register for many subjects (subject code, subject name, number of credits)
- Each subject can have multiple subjects that students must complete before they can be registered
- Each subject can have multiple classes (class code, class name, maximum number of students, classrooms, fixed hours of the week)
- Students are not allowed to register for two classes with the same class time
- For each subject, a student is only allowed to enroll in a class
- Student results (part number 1, number 2, number 3, test score, final score=x% number1+ y % number2 + z% number3 + w% test score) are saved for each subject
- The student's grade point average for the semester is calculated as a weighted average of the number of credits per subject

Module "**View Student's schedule**": Students select the menu to view the schedule → The system shows up with a box to select the ways to view the schedule by: week, semester → Student selects weekly view → The system displays the current weekly schedule of students: 1 table has 7 columns corresponding to 7 days, 6 rows corresponding to 6 kips for each day. In each cell of the table shows the subject name, classrooms, and class name corresponding to that time slot. Clicks on a cell, the detailed information of the selected kip is shown: subject id, subject name, class id, classe name, class room, lecturer name, start time and end time.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 09
Duration: 60 minutes

A client requires us to develop a software to manage student results, described as follows:

- Each student (Student ID, password, name, date of birth, course, hometown, address) is allowed to register for a minimum of 10 credits/semester and a maximum of 15 credits/semester
- Each student can register for many subjects (subject code, subject name, number of credits)
- Each subject can have multiple subjects that students must complete before they can be registered
- Each subject can have multiple classes (class code, class name, maximum number of students, classrooms, fixed hours of the week)
- Students are not allowed to register for two classes with the same class time
- For each subject, a student is only allowed to enroll in a class
- Student results (part number 1, number 2, number 3, test score, final score=x% number1+ y % number2 + z% number3 + w% test score) are saved for each subject
- The student's grade point average for the semester is calculated as a weighted average of the number of credits per subject

Module "**Statistics of students**": A manage staff logins → selects the statistics menu → selects student statistics → the results page shows a list of students: student code, Name of student, course, semester, total number of credits studied in the semester, average score of the last semester, sorted by the average score of the whole semester, from high to low. The staff clicks on a line of a student to display the details score of each subject that the student has studied in the semester.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 10
Duration: 60 minutes

A client requires us to develop a software to manage student results, described as follows:

- Each student (Student ID, password, name, date of birth, course, hometown, address) is allowed to register for a minimum of 10 credits/semester and a maximum of 15 credits/semester
- Each student can register for many subjects (subject code, subject name, number of credits)
- Each subject can have multiple subjects that students must complete before they can be registered
- Each subject can have multiple classes (class code, class name, maximum number of students, classrooms, fixed hours of the week)
- Students are not allowed to register for two classes with the same class time
- For each subject, a student is only allowed to enroll in a class
- Student results (part number 1, number 2, number 3, test score, final score=x% number1+ y % number2 + z% number3 + w% test score) are saved for each subject
- The student's grade point average for the semester is calculated as a weighted average of the number of credits per subject

Module "**Statistics of subjects**": A manage staff logins → selects the statistics menu → selects the subject statistics → the results page shows a list of subjects: code, name, number of credits, average score of students in the subject, percentage of students passing the subject in all classes (calculated %). Results are sorted by the percentage of students passing that subject from high to low. The staff clicks on a line of 1 subject, the details score of all students who have studied the subject will appear.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 11
Duration: 60 minutes

A client requires us to develop a tour booking management software, described as follows:

- Each tour (Tour code, name, departure place, destination, description) can depart on many different days, depending on the departure date and the number of people buying the tour for each group will have different prices.
- Each tour can have a schedule to go through many different tourist sites. At each site, each tour can use different services of different providers.
- Each customer (Code, name, ID number, ID card type, phone number, email, address) can buy tickets for many different tours. Each tour can buy a different number of tickets. Each purchase is issued an invoice specifying tour information, departure date, tour price, number of guests, name of customer representative, total payment amount.
- The same customer can go on the same tour multiple times, only differing in departure date and ticket price.
- Customers can return tickets and have to pay the fine of ticket cancelation.

Module "**Buy tickets**": A staff selects the function to buy tickets requested by a customer→ tour search interface (by destination name) is displayed → The staff enters the destination name and click search → The results appeared include a list of available tours corresponding to the selected criteria, each tour displays compelete information + departure date + corresponding price at the time of search → The staff selects 1 tour according to customer's choice → Invoice (ticket) details will be displayed: tour name, departure place, destination, departure date, name of the delegation representative, ID number, ID type, guest address, phone number, email, number of guests, price ticket → The staff selects payment → customer pays → the system saves the result and prints the ticket for the customer.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

.

Software engineering
Subject No. 12
Duration: 60 minutes

A client requires us to develop a tour booking management software, described as follows:

- Each tour (Tour code, name, departure place, destination, description) can depart on many different days, depending on the departure date and the number of people buying the tour for each group will have different prices.
- Each tour can have a schedule to go through many different tourist sites. At each site, each tour can use different services of different providers.
- Each customer (Code, name, ID number, ID card type, phone number, email, address) can buy tickets for many different tours. Each tour can buy a different number of tickets. Each purchase is issued an invoice specifying tour information, departure date, tour price, number of guests, name of customer representative, total payment amount.
- The same customer can go on the same tour multiple times, only differing in departure date and ticket price.
- Customers can return tickets and have to pay the fine of ticket cancelation.

Module "**Cancel the ticket**": A staff selects the ticket cancelation function requested by a customer → the ticket search interface appears → enters the ticket code → the result appears. Detailed ticket: tour name, departure place, destination, departure date, name of the group representative, ID number, ID type, guest address, phone number, email, number of guests, ticket price → choose to cancel tickets → the system displays the fine invoice including the information as on the ticket + fine according to the cancel time → press Ok → the system saves the result to the system, and the staff sends the change back to the customer.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

.

Software engineering
Subject No. 13
Duration: 60 minutes

A client requires us to develop a tour booking management software, described as follows:

- Each tour (Tour code, name, departure place, destination, description) can depart on many different days, depending on the departure date and the number of people buying the tour for each group will have different prices.
- Each tour can have a schedule to go through many different tourist sites. At each site, each tour can use different services of different providers.
- Each customer (Code, name, ID number, ID card type, phone number, email, address) can buy tickets for many different tours. Each tour can buy a different number of tickets. Each purchase is issued an invoice specifying tour information, departure date, tour price, number of guests, name of customer representative, total payment amount.
- The same customer can go on the same tour multiple times, only differing in departure date and ticket price.
- Customers can return tickets and have to pay the fine of ticket cancelation.

Module "**Tour statistics by revenue**": A manager staff selects the function of tour statistics by revenue → the interface selects the periode (start date - end date) → The staff enters the start/end time and clicks view → results appear including a list of detailed tours, sorted by total revenue, from high to low: code, name, place of departure, destination, average number of guests/tour, total revenue. The staff clicks on a line of a tour -> the system displays a detailed list of customer invoices who have ordered that tour, each invoice on 1 line: id, customer's name, departure date and time, total number of guests, total amount of money.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

.

Software engineering
Subject No. 14
Duration: 60 minutes

A client requires us to develop a tour booking management software, described as follows:

- Each tour (Tour code, name, departure place, destination, description) can depart on many different days, depending on the departure date and the number of people buying the tour for each group will have different prices.
- Each tour can have a schedule to go through many different tourist sites. At each site, each tour can use different services of different providers.
- Each customer (Code, name, ID number, ID card type, phone number, email, address) can buy tickets for many different tours. Each tour can buy a different number of tickets. Each purchase is issued an invoice specifying tour information, departure date, tour price, number of guests, name of customer representative, total payment amount.
- The same customer can go on the same tour multiple times, only differing in departure date and ticket price.
- Customers can return tickets and have to pay the fine of ticket cancelation.

Module "Revenue statistics by site": A manager staff selects the function of revenue statistics by tourist site → interface selects statistical periode (start date - end date) appears → The satff enters the periode and clicks view statistics → the results appear including a list of detailed site, sorted by total revenue, from high to low: name, number of tours including that site, total number of visitors to that site, total revenue. The staff clicks on a line of a site, the system displays a detailed list of customer invoices who have ordered tours through that site, each invoice on 1 line: id, customer's name, departure date and time, name tour, total number of guests, total amount.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

.

Software engineering
Subject No. 15
Duration: 60 minutes

A client requires us to develop an order management software in a restaurant, described as follows:

- The restaurant has many tables (Table code, name, maximum number of guests, description). Many small tables can be merged into one large table upon request from a large group of guests.
- Each table, can be booked many times in a day, or in different days.
- Each customer (Code, name, phone number, email, address) can book many times, each time can book many tables (in this case, it will be merged into 1 table)
- Restaurants can make combos that combine a number of dishes that are enough for one meal for one person to eat. Customers can call available combos like this.
- Customers at each table can order multiple dishes (Code, type, name, description, current price) or combo. Each dish (combo) can be ordered with a different amount.
- When paying, the invoice contains all information: desk code, name and code of staff, customer name if any, then a table, each line contains information about an used item (combo): id, name, unit price, quantity, amount. The last line shows the total amount of the invoice.

Module "**Order**": A staff selects the ordering function → the table interface appears with a drop-down list of tables → selects the correct table for the customer who is ordering → The interface to enter the ordered dish appears → The staff asks the customer and enters the name of the dish + selects search → the results appear including a list of detailed dishes: code, type, name, price. → The staff selects 1 dish exactly as customer ordered → Required to enter quantity → enters quantity and click OK → Dish name + quantity + subtotal added to the list of selected dishes below. The staff repeats these steps of choosing dishes until all the dishes that customers in the table have ordered. The staff reads again to confirm with customer → click confirm → system saves.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 16
Duration: 60 minutes

A client requires us to develop an order management software in a restaurant, described as follows:

- The restaurant has many tables (Table code, name, maximum number of guests, description). Many small tables can be merged into one large table upon request from a large group of guests.
- Each table, can be booked many times in a day, or in different days.
- Each customer (Code, name, phone number, email, address) can book many times, each time can book many tables (in this case, it will be merged into 1 table)
- Restaurants can make combos that combine a number of dishes that are enough for one meal for one person to eat. Customers can call available combos like this.
- Customers at each table can order multiple dishes (Code, type, name, description, current price) or combo. Each dish (combo) can be ordered with a different amount.
- When paying, the invoice contains all information: desk code, name and code of staff, customer name if any, then a table, each line contains information about an used item (combo): id, name, unit price, quantity, amount. The last line shows the total amount of the invoice.

Module "**Book a table**": A staff selects the table reservation function when a customer calls → the interface to find an available table appears → The customer enters the date + time of booking + the number of guests and presses the button search → results appear including a list of available tables on that date and time: code, name, maximum number of guests, description → Customer selects a table according to customer's choice → Customer information input interface appears → The staff asks the customer and enters the code, name, phone number, email, address and click search → The system displays a list of customers whose name contains the entered keyword, each customer on 1 line: code, name, phone number, email, address → The staff clicks on the right line with the currently booked customer (if not, click add a new customer) → The system displays full table information + customer information + date and time of booking → The staff confirms with the customer and click confirm → The system saves the booking information in the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 17
Duration: 60 minutes

A client requires us to develop an order management software in a restaurant, described as follows:

- The restaurant has many tables (Table code, name, maximum number of guests, description). Many small tables can be merged into one large table upon request from a large group of guests.
- Each table, can be booked many times in a day, or in different days.
- Each customer (Code, name, phone number, email, address) can book many times, each time can book many tables (in this case, it will be merged into 1 table)
- Restaurants can make combos that combine a number of dishes that are enough for one meal for one person to eat. Customers can call available combos like this.
- Customers at each table can order multiple dishes (Code, type, name, description, current price) or combo. Each dish (combo) can be ordered with a different amount.
- When paying, the invoice contains all information: desk code, name and code of staff, customer name if any, then a table, each line contains information about an used item (combo): id, name, unit price, quantity, amount. The last line shows the total amount of the invoice.

Module "**Make a combo menu**": The manager staff selects the combo management menu → The management page appears → selects the function of adding new combos → the interface to add combos appears with the fields to enter the combo name and the button to add dishes to the combo → clicks to add dishes to the combo → the interface to find dishes by name appears → enters the name of the dish and click search → a list of dishes with the name containing the keyword appears → selects a dish → the system returns to the interface to add combos with the selected dish added to the combo → The staff repeats until the dishes are added to the combo and then click update → the system saves the information in the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 18
Duration: 60 minutes

A client requires us to develop an order management software in a restaurant, described as follows:

- The restaurant has many tables (Table code, name, maximum number of guests, description). Many small tables can be merged into one large table upon request from a large group of guests.
- Each table, can be booked many times in a day, or in different days.
- Each customer (Code, name, phone number, email, address) can book many times, each time can book many tables (in this case, it will be merged into 1 table)
- Restaurants can make combos that combine a number of dishes that are enough for one meal for one person to eat. Customers can call available combos like this.
- Customers at each table can order multiple dishes (Code, type, name, description, current price) or combo. Each dish (combo) can be ordered with a different amount.
- When paying, the invoice contains all information: desk code, name and code of staff, customer name if any, then a table, each line contains information about an used item (combo): id, name, unit price, quantity, amount. The last line shows the total amount of the invoice.

Module "**Payment**": Customer requests a staff to pay → Staff selects payment function → Table selection interface appears with a list of tables → Customer selects the table matches the customer's table → The detailed invoice interface of the table appears as described above (can add/edit the ordered items in the bill, if necessary) → The customer asks if the customer has a coupon. → if yes, then click add coupon + enter code → invoice interface add coupon line and update total amount to pay → The staff informs customer the amount → After payment, the staff clicks confirm → system saves and prints detailed invoices for customers.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 19
Duration: 60 minutes

A client requires us to develop an order management software in a restaurant, described as follows:

- The restaurant has many tables (Table code, name, maximum number of guests, description). Many small tables can be merged into one large table upon request from a large group of guests.
- Each table, can be booked many times in a day, or in different days.
- Each customer (Code, name, phone number, email, address) can book many times, each time can book many tables (in this case, it will be merged into 1 table)
- Restaurants can make combos that combine a number of dishes that are enough for one meal for one person to eat. Customers can call available combos like this.
- Customers at each table can order multiple dishes (Code, type, name, description, current price) or combo. Each dish (combo) can be ordered with a different amount.
- When paying, the invoice contains all information: desk code, name and code of staff, customer name if any, then a table, each line contains information about an used item (combo): id, name, unit price, quantity, amount. The last line shows the total amount of the invoice.

Module "**Statistics of visitors by time slot**": A manager staff selects the function of visitor statistics by time slot → interface selects statistical pẻiode (start - end date) appears → The staff enters the periode and clicks view statistics → the results appear including a list of detailed time slot: start time, end time of slot, average number of visitors, average revenue/per guest, total hourly revenue. The staff clicks on a time slot, the system shows the details of other bills used in that slot, each bill on 1 line: code, customer name, date, total number of orders, total payment amount.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 20
Duration: 60 minutes

A client requires us to develop an order management software in a restaurant, described as follows:

- The restaurant has many tables (Table code, name, maximum number of guests, description). Many small tables can be merged into one large table upon request from a large group of guests.
- Each table, can be booked many times in a day, or in different days.
- Each customer (Code, name, phone number, email, address) can book many times, each time can book many tables (in this case, it will be merged into 1 table)
- Restaurants can make combos that combine a number of dishes that are enough for one meal for one person to eat. Customers can call available combos like this.
- Customers at each table can order multiple dishes (Code, type, name, description, current price) or combo. Each dish (combo) can be ordered with a different amount.
- When paying, the invoice contains all information: desk code, name and code of staff, customer name if any, then a table, each line contains information about an used item (combo): id, name, unit price, quantity, amount. The last line shows the total amount of the invoice.

Module "**Statistical monthly revenue**": A manager staff selects the statistics menu → select monthly revenue statistics → list of last 12 months is displayed, one line for a month: name, total number of guest, total revenue, sorted in descending order of total revenue. The staff clicks on a line of a month, the system displays the details of the customer's bills for the month, each bill on the line: id, name of customer, date and time, total number of orders, total payment amount.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 21
Duration: 60 minutes

A client requires us to develop an order management software in a restaurant, described as follows:

- The restaurant has many tables (Table code, name, maximum number of guests, description). Many small tables can be merged into one large table upon request from a large group of guests.
- Each table, can be booked many times in a day, or in different days.
- Each customer (Code, name, phone number, email, address) can book many times, each time can book many tables (in this case, it will be merged into 1 table)
- Restaurants can make combos that combine a number of dishes that are enough for one meal for one person to eat. Customers can call available combos like this.
- Customers at each table can order multiple dishes (Code, type, name, description, current price) or combo. Each dish (combo) can be ordered with a different amount.
- When paying, the invoice contains all information: desk code, name and code of staff, customer name if any, then a table, each line contains information about an used item (combo): id, name, unit price, quantity, amount. The last line shows the total amount of the invoice.

Module "**Statistics of dishes**": A manager staff selects the function of statistics of dishes → the interface to choose the periode of statistics (start - end date) appears → After selecting the periode, the staff click statistics → the results appear including a list of detailed dishes: code, type, name, total number of sales, total revenue (Sorted by total revenue, from high to low). The staff clicks on a line of a dish, the system displays a detailed list of times the dish was ordered: id, guest name, date and time, quantity, amount.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 22
Duration: 60 minutes

A client requires us to develop a store management software, described as follows:

- Each item (Item code, name, description) can be imported many times, each time has different quantity and different price, coming from a supplier (code, name, address, phone number) are different
- You can import many different items at a time
- Each imported time there is an imported bill with supplier information, followed by a list of imported items, each item has full information: code, name, quantity, unit price, and total amount (automatically calculated) and the last line is the total amount of the imported bill
- Similarly, each item can be exported many times, each time to different sub-agents (code, brand name, address, phone number), with different quantities and different export prices.
- Each export time can export many items, as long as the export quantity does not exceed the number of goods in the store
- Each export time has an export bill with sub-agent information, followed by a list of exported items, each item has full information: code, name, quantity, unit price, and total amount (automatically calculated) and the last line is the total amount of the bill.

Module "**Exporting**": A staff selects the export menu → the export page appears with the search box for sub-agence → enters the name of the agent and clicks to search → the system displays a list of agents whose names contain the entered keyword → clicks the line of the correct sub-agence (in case of new sub-agence, it must be added) → the system displays the interface to find exported item → The staff enters the name item and click search → the system displays a list of items whose names containing the keywords entered → the employee selects the correct item in the list + enters the quantity + unit price → the item appears in the list of exported items → repeat until all the items need to be exported, then submit → the system saves and prints out the export bill as described.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 23
Duration: 60 minutes

A client requires us to develop a store management software, described as follows:

- Each item (Item code, name, description) can be imported many times, each time has different quantity and different price, coming from a supplier (code, name, address, phone number) are different
- You can import many different items at a time
- Each imported time there is an imported bill with supplier information, followed by a list of imported items, each item has full information: code, name, quantity, unit price, and total amount (automatically calculated) and the last line is the total amount of the imported bill
- Similarly, each item can be exported many times, each time to different sub-agents (code, brand name, address, phone number), with different quantities and different export prices.
- Each export time can export many items, as long as the export quantity does not exceed the number of goods in the store
- Each export time has an export bill with sub-agent information, followed by a list of exported items, each item has full information: code, name, quantity, unit price, and total amount (automatically calculated) and the last line is the total amount of the bill.

Module "**Importing**": A staff selects the import menu → the import page appears with the search box for providers → enters the name of the provider and clicks to search → the system displays a list of providers whose names contain the entered keyword → clicks the line of the correct provider (in case of new provider, it must be added) → the system displays the interface to find imported item → The staff enters the name item and click search → the system displays a list of items whose names containing the keywords entered (if there is no results, it must be added) → the employee selects the correct item in the list + enters the quantity + unit price → the item appears in the list of imported items → repeat until all the items need to be imported, then submit → the system saves and prints out the import bill as described.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 24
Duration: 60 minutes

A client requires us to develop a store management software, described as follows:

- Each item (Item code, name, description) can be imported many times, each time has different quantity and different price, coming from a supplier (code, name, address, phone number) are different
- You can import many different items at a time
- Each imported time there is an imported bill with supplier information, followed by a list of imported items, each item has full information: code, name, quantity, unit price, and total amount (automatically calculated) and the last line is the total amount of the imported bill
- Similarly, each item can be exported many times, each time to different sub-agents (code, brand name, address, phone number), with different quantities and different export prices.
- Each export time can export many items, as long as the export quantity does not exceed the number of goods in the store
- Each export time has an export bill with sub-agent information, followed by a list of exported items, each item has full information: code, name, quantity, unit price, and total amount (automatically calculated) and the last line is the total amount of the bill.

Module "**Statistics of items**": A manager staff selects the statistics menu → selects the function of item statistics → enter the statistical period (start - end) → the results show a list of item in order of total revenue from the most to the least in the selected period, each line has the following information: item code, item name, quantity sold, the total revenue during the selected time period. The staff clicks on a line of an item to display detailed statistics of invoices of sub-agents who have purchased that product.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 25
Duration: 60 minutes

A client requires us to develop a store management software, described as follows:

- Each item (Item code, name, description) can be imported many times, each time has different quantity and different price, coming from a supplier (code, name, address, phone number) are different
- You can import many different items at a time
- Each imported time there is an imported bill with supplier information, followed by a list of imported items, each item has full information: code, name, quantity, unit price, and total amount (automatically calculated) and the last line is the total amount of the imported bill
- Similarly, each item can be exported many times, each time to different sub-agents (code, brand name, address, phone number), with different quantities and different export prices.
- Each export time can export many items, as long as the export quantity does not exceed the number of goods in the store
- Each export time has an export bill with sub-agent information, followed by a list of exported items, each item has full information: code, name, quantity, unit price, and total amount (automatically calculated) and the last line is the total amount of the bill.

Module "**Statistics of sub-agence**": A manager staff selects the statistics menu → select the top sub-agence statistics function → enter the statistical period (start strat - end) → the results show a list of sub-agence in order of the most total revenue to the least in the selected time period, each line has the following information: sub-agence code, name, the total revenue from the sub-agence during the selected period. The staff clicks on a line of an sub-agence, a detailed list of invoices (date, total number of item, total amount) of each time the sub-agence has imported item will appear.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 26
Duration: 60 minutes

A client requires us to develop a software to manage the World Championship Tournament of Chess with the following description:

- Each tournament (Code, name, year, time held, location, description) allows multiple players (code, name, year of birth, nationality, Elo coefficient, notes) to participate.
- There may be hundreds of players involved, but each player must play 11 matches according to the Swiss rule.
- In the first game, the players are ranked in order of Elo coefficients from high to low. Then going from the top to the bottom of the arrangement, two players standing next to each other will form a pair for round 1.
- In each round, win 1 point, draw 0.5 point, lose 0 point. After each round, the results of each match are updated according to the previously scheduled matches. At the same time, the Elo coefficient increasing or decreasing after each round is also updated (Calculated by FIDE's formula, just enter the results).
- Starting from the 2nd round, the temporary standings after the previous round are ranked according to the following criteria: total score (descending), total score of opponents met (descending), Elo coefficient (decrease). And the match is determined as follows, going from top to bottom of the provisional standings, for each unpaired player, that player's opponent is the first player encountered and satisfied: no match, and haven't met the considered player.
- After 11 such rounds, the top player in the ranking will be the champion.

Module "**Update results**": A staff selects the results update menu → the results update page appears → The staff selects the round from the drop-down list + select a match from the list that comes out by round + enter the number of points and Elo points for the 2 players of the match + click Update → The system will notify you that the match result is successfully saved and return to the round + match selection interface.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 27
Duration: 60 minutes

A client requires us to develop a software to manage the World Championship Tournament with the following description:

- Each tournament (Code, name, year, time held, location, description) allows multiple players (code, name, year of birth, nationality, Elo coefficient, notes) to participate.
- There may be hundreds of players involved, but each player must play 11 matches according to the Swiss rule.
- In the first game, the players are ranked in order of Elo coefficients from high to low. Then going from the top to the bottom of the arrangement, two players standing next to each other will form a pair for round 1.
- In each round, win 1 point, draw 0.5 point, lose 0 point. After each round, the results of each match are updated according to the previously scheduled matches. At the same time, the Elo coefficient increasing or decreasing after each round is also updated (Calculated by FIDE's formula, just enter the results).
- Starting from the 2nd round, the temporary standings after the previous round are ranked according to the following criteria: total score (descending), total score of opponents met (descending), Elo coefficient (decrease). And the match is determined as follows, going from top to bottom of the provisional standings, for each unpaired player, that player's opponent is the first player encountered and satisfied: no match, and haven't met the considered player.
- After 11 such rounds, the top player in the ranking will be the champion.

Module "**View leaderboard**": A staff selects the statistics menu → select the function to view the standings after each round → select the round from the drop-down list → the results show a list of players, each with full information: id, name, year of birth, nationality, total score, total score of opponents met, instant Elo coefficient (Sorted in the order described above). Clicks on a player -> list of played match results appears, each match has: id, name of the opponent player, result (win, draw, loss), and the Elo increasement/decreasement after the match.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 28
Duration: 60 minutes

A client requires us to develop a software to manage the World Championship Tournament with the following description:

- Each tournament (Code, name, year, time held, location, description) allows multiple players (code, name, year of birth, nationality, Elo coefficient, notes) to participate.
- There may be hundreds of players involved, but each player must play 11 matches according to the Swiss rule.
- In the first game, the players are ranked in order of Elo coefficients from high to low. Then going from the top to the bottom of the arrangement, two players standing next to each other will form a pair for round 1.
- In each round, win 1 point, draw 0.5 point, lose 0 point. After each round, the results of each match are updated according to the previously scheduled matches. At the same time, the Elo coefficient increasing or decreasing after each round is also updated (Calculated by FIDE's formula, just enter the results).
- Starting from the 2nd round, the temporary standings after the previous round are ranked according to the following criteria: total score (descending), total score of opponents met (descending), Elo coefficient (decrease). And the match is determined as follows, going from top to bottom of the provisional standings, for each unpaired player, that player's opponent is the first player encountered and satisfied: no match, and haven't met the considered player.
- After 11 such rounds, the top player in the ranking will be the champion.

Module "**Pair scheduling**": A staff selects the scheduling menu → the schedule page appears → The staff selects the previous round in the dropdown list → the system shows the current standings after the selected round + the Schedule button → The staff clicks the Schedule button → The system automatically matches the players according to the rules described above, and shows the list of tables in the correct order of matches (table name, name of the two player) → The staff clicks Save → The system saves the schedule of the new round in the database

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 29
Duration: 60 minutes

A client requires us to develop a software to manage the World Championship Tournament with the following description:

- Each tournament (Code, name, year, time held, location, description) allows multiple players (code, name, year of birth, nationality, Elo coefficient, notes) to participate.
- There may be hundreds of players involved, but each player must play 11 matches according to the Swiss rule.
- In the first game, the players are ranked in order of Elo coefficients from high to low. Then going from the top to the bottom of the arrangement, two players standing next to each other will form a pair for round 1.
- In each round, win 1 point, draw 0.5 point, lose 0 point. After each round, the results of each match are updated according to the previously scheduled matches. At the same time, the Elo coefficient increasing or decreasing after each round is also updated (Calculated by FIDE's formula, just enter the results).
- Starting from the 2nd round, the temporary standings after the previous round are ranked according to the following criteria: total score (descending), total score of opponents met (descending), Elo coefficient (decrease). And the match is determined as follows, going from top to bottom of the provisional standings, for each unpaired player, that player's opponent is the first player encountered and satisfied: no match, and haven't met the considered player.
- After 11 such rounds, the top player in the ranking will be the champion.

Module "**Statistics of Elo change**": A staff selects the statistics menu → select the function of Elo change statistics of the chess players after the tournament → the results appear: List of players, each player is shown full information: code, name, year of birth, nationality, old Elo coefficient, new Elo coefficient, increased/decreased Elo coefficient (sorted in descending order of the increase or decrease of the Elo coefficient of the oddities, followed by the descending of the new Elo coefficient). Click on a line of a chess player → the system displays the details of the matches that the player has played, each match on 1 line: id, opponent's name, result (win, draw, loss), Elo increase or decrease.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 30
Duration: 60 minutes

A client requires us to develop a software to manage the F1 Formula Championship with the following description:

- There is a championship every year. A tournament consisting of many races taking place around the world (Race code, name, number of laps, location, time, description).
- Each tournament has many participating teams (Code, name, brand, description).
- Each racing team has many riders (code, name, date of birth, nationality, biography). But in each race, each team is only allowed to allow a maximum of 2 riders to participate.
- Each driver can play for many racing teams at different times. But at a time only play for 1 team.
- For each race, the results are ranked in order of finishing (time) and the score is only calculated for the top 10, respectively in the order of finishing 25, 18, 15, 12, 10, 8, 6, 4, 2, 1.
- If the driver is in the top 10 but does not finish due to a dropout or an accident, then 0 points.
- The score and time of each driver will be added up between the stages to decide the individual and team prizes of the season.

Module "**Register to racing**": A staff selects the racer registration function required by the team → the racer registration interface for each stage appears → The staff selects the race from the drop-down list + select a racing team from the drop-down list → a list of racers of the selected team appears, sorted by their alphabetic order of name → The staff ticks the correct 2 racers according to the team's request + click Save → The system saves information.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 31
Duration: 60 minutes

A client requires us to develop a software to manage the F1 Formula Championship with the following description:

- There is a championship every year. A tournament consisting of many races taking place around the world (Race code, name, number of laps, location, time, description).
- Each tournament has many participating teams (Code, name, brand, description).
- Each racing team has many riders (code, name, date of birth, nationality, biography). But in each race, each team is only allowed to allow a maximum of 2 riders to participate.
- Each driver can play for many racing teams at different times. But at a time only play for 1 team.
- For each race, the results are ranked in order of finishing (time) and the score is only calculated for the top 10, respectively in the order of finishing 25, 18, 15, 12, 10, 8, 6, 4, 2, 1.
- If the driver is in the top 10 but does not finish due to a dropout or an accident, then 0 points.
- The score and time of each driver will be added up between the stages to decide the individual and team prizes of the season.

Module "**Update results**": A staff selects the function of entering race results → the result input interface appears → The staff selects the race name from the list drop-down → The list of registered racers for the race appears in the form of a table, each line contains blank boxes to enter the time to the finish line, the number of laps completed → The staff enters all the results of all racers and click Save → The system saves the results to the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 32
Duration: 60 minutes

A client requires us to develop a software to manage the F1 Formula Championship with the following description:

- There is a championship every year. A tournament consisting of many races taking place around the world (Race code, name, number of laps, location, time, description).
- Each tournament has many participating teams (Code, name, brand, description).
- Each racing team has many riders (code, name, date of birth, nationality, biography). But in each race, each team is only allowed to allow a maximum of 2 riders to participate.
- Each driver can play for many racing teams at different times. But at a time only play for 1 team.
- For each race, the results are ranked in order of finishing (time) and the score is only calculated for the top 10, respectively in the order of finishing 25, 18, 15, 12, 10, 8, 6, 4, 2, 1.
- If the driver is in the top 10 but does not finish due to a dropout or an accident, then 0 points.
- The score and time of each driver will be added up between the stages to decide the individual and team prizes of the season.

Module "**View the racers' standings**": A staff selects the statistics function → Select to view the current racer rankings → The staff selects a stage from the dropdown list -> The system displays a list of racers in the form of a table, each line contains: Racer's name, nationality, team name, total score after the selected stages, total time after stages (sorted in descending order of total score, then in ascending order of total time). The staff clicks on a line of a racer → the system displays the detailed results of each race stage given by that racer, each stage on one line: stage name, finish rank, score, time to finish.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 33
Duration: 60 minutes

A client requires us to develop a software to manage the F1 Formula Championship with the following description:

- There is a championship every year. A tournament consisting of many races taking place around the world (Race code, name, number of laps, location, time, description).
- Each tournament has many participating teams (Code, name, brand, description).
- Each racing team has many riders (code, name, date of birth, nationality, biography). But in each race, each team is only allowed to allow a maximum of 2 riders to participate.
- Each driver can play for many racing teams at different times. But at a time only play for 1 team.
- For each race, the results are ranked in order of finishing (time) and the score is only calculated for the top 10, respectively in the order of finishing 25, 18, 15, 12, 10, 8, 6, 4, 2, 1.
- If the driver is in the top 10 but does not finish due to a dropout or an accident, then 0 points.
- The score and time of each driver will be added up between the stages to decide the individual and team prizes of the season.

Module "**View the team rankings**": A staff selects the statistics function → Select to view the current racing team rankings → The staff selects a stage from the dropdown list → The system displays a list of racing teams, in the form of a table, each line contains: Team name, team owner, total points of the team's drivers after stages, total time after stages (sorted in descending order of total score, then in ascending order of total time). The staff clicks on a line of a racing team → the system displays detailed results for each stage of that racing team, each stage on 1 line: race name, total score, total time of the 2 racers in the team.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 34
Duration: 60 minutes

A client requires us to develop a book rental management software with the following description:

- The store has many book titles. Each book title has a different quantity and a different rental price (rental per day).
- Each book title can be borrowed by many customers. Each customer can borrow many books each time.
- Each time of borrowing, the borrower will receive a loan voucher. In which, the first line contains the name of the customer and the date of the loan. Information for each borrowed book is written on a line: name, author, publisher, year of publication, rental price. The last line shows the number of borrowed book titles.
- When returning, the customer will receive a payment invoice. In it, the first line contains the customer's name and payment date. Information for each paid book is written on one line: name, author, publisher, year of publication, date of borrowing, date of payment, rent, amount. If fined, there is an additional column of fines. The last line shows the total amount of the payment.

Module "**Borrowing**": After selecting books to borrow, a customer takes them to the cashier counter to make loan slips. A librarian staff enters the customer's name and searches → The system returns a list of customers whose name contains the name entered → The staff clicks on the customer's name in the list (if the customer borrows for the first time, enter a new one) → The system displays the interface to add borrowed books: For each book title, the staff clicks to search for book by name → enter the title of the book + click search → the system displays a list of the book titles whose name contains the entered name → The staff clicks on the correct line with the book selected by the customer → System adds 1 line corresponding to the borrowing list. When finish the book list, the staff clicks to create a loan slip → The system saves the loan slip and displays the loan slip on the screen → The staff clicks to print → The system prints the loan slip for the customer.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 35
Duration: 60 minutes

A client requires us to develop a book rental management software with the following description:

- The store has many book titles. Each book title has a different quantity and a different rental price (rental per day).
- Each book title can be borrowed by many customers. Each customer can borrow many books each time.
- Each time of borrowing, the borrower will receive a loan voucher. In which, the first line contains the name of the customer and the date of the loan. Information for each borrowed book is written on a line: name, author, publisher, year of publication, rental price. The last line shows the number of borrowed book titles.
- When returning, the customer will receive a payment invoice. In it, the first line contains the customer's name and payment date. Information for each paid book is written on one line: name, author, publisher, year of publication, date of borrowing, date of payment, rent, amount. If fined, there is an additional column of fines. The last line shows the total amount of the payment.

Module "**Return and pay**": When a customer returns books, a staff selects the menu to find a list of borrowed stories by the customer's name → enter the customer's name + click search → system displays a list of customers whose names contains entered keyword → The staff selects the correct customer name → the system displays a list of book titles that the customer is borrowing, each title on a line with full information : code, name, the date of the loan, the loan price, and the rental amount up to the date of payment, the last column is the check box to choose to pay → The staff clicks on the return button for the book titles that the customer has returned (may not pay all 1 times), enters the status of the book and the fine if any, finally click the payment button → the system displays the invoice with full customer information + a list of returned book titles as described above + the last line is the total payment → The staff clicks confirm → the system updates to the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 36
Duration: 60 minutes

A client requires us to develop a book rental management software with the following description:

- The store has many book titles. Each book title has a different quantity and a different rental price (rental per day).
- Each book title can be borrowed by many customers. Each customer can borrow many books each time.
- Each time of borrowing, the borrower will receive a loan voucher. In which, the first line contains the name of the customer and the date of the loan. Information for each borrowed book is written on a line: name, author, publisher, year of publication, rental price. The last line shows the number of borrowed book titles.
- When returning, the customer will receive a payment invoice. In it, the first line contains the customer's name and payment date. Information for each paid book is written on one line: name, author, publisher, year of publication, date of borrowing, date of payment, rent, amount. If fined, there is an additional column of fines. The last line shows the total amount of the payment.

Module "**Statistics of book**": A staff selects the menu statistics of the most borrowed book → Enter the time period (start - end date) statistics → The system displays a list of borrowed titles in a table format, each line corresponds to a book title: code, name, author, publisher, year of publication, column total number of times borrowed, column total amount (Sorted in descending order of the total borrowed column, followed by the descending of the total income column). The staff clicks on 1 line of a book → the system displays the details of the invoice with that book borrowed, each invoice on 1 line: id, name of the borrower, borrowed date and time, payment date and time, total payment amount.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 37
Duration: 60 minutes

A client requires us to develop a book rental management software with the following description:

- The store has many book titles. Each book title has a different quantity and a different rental price (rental per day).
- Each book title can be borrowed by many customers. Each customer can borrow many books each time.
- Each time of borrowing, the borrower will receive a loan voucher. In which, the first line contains the name of the customer and the date of the loan. Information for each borrowed book is written on a line: name, author, publisher, year of publication, rental price. The last line shows the number of borrowed book titles.
- When returning, the customer will receive a payment invoice. In it, the first line contains the customer's name and payment date. Information for each paid book is written on one line: name, author, publisher, year of publication, date of borrowing, date of payment, rent, amount. If fined, there is an additional column of fines. The last line shows the total amount of the payment.

Module "**Statistics of borrowers**": A staff selects the menu of statistics of customers → Enter the time period (start - end date) statistics → system displays a list of customers who borrow a lot in the form of a table, each line corresponds to a customer: code, name, idcard number, phone number, address, followed by the column of total number of loans, column of total amount paid (Sorted in descending order of the total number of loans, followed by the descending direction of the total amount paid). The staff clicks on 1 line of a customer → the system displays the details of the invoices that customer has borrowed, each invoice on 1 line: borrowed date, total number of books borrowed, total payment amount.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 38
Duration: 60 minutes

A client requires us to develop a book rental management software with the following description:

- The store has many book titles. Each book title has a different quantity and a different rental price (rental per day).
- Each book title can be borrowed by many customers. Each customer can borrow many books each time.
- Each time of borrowing, the borrower will receive a loan voucher. In which, the first line contains the name of the customer and the date of the loan. Information for each borrowed book is written on a line: name, author, publisher, year of publication, rental price. The last line shows the number of borrowed book titles.
- When returning, the customer will receive a payment invoice. In it, the first line contains the customer's name and payment date. Information for each paid book is written on one line: name, author, publisher, year of publication, date of borrowing, date of payment, rent, amount. If fined, there is an additional column of fines. The last line shows the total amount of the payment.

Module "**Revenue Statistics**": A staff selects the menu of sales statistics by time (month, quarter, year) → the system displays a box to select statistics by month, quarter, or year → The staff clicks by month → the system displays monthly revenue statistics in the form of a table, each line corresponds to 1 month (corresponding to quarter, year): month name, total revenue (Sorted by the order from the nearest month (respectively quarter, year) to the oldest month (respectively quarter, year)). The staff clicks on a line → the system displays the details of invoices in that line, each invoice on 1 line: id, customer name, loan date, total number of stories borrowed, total amount of the invoice .

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 39
Duration: 60 minutes

A client requires us to develop a software to manage the (part-time) employees of stores with the following description:

- The restaurant chain has many restaurants. Each restaurant has many hourly employees. Each working day has 2 shifts, 1st shift from 8am to 4pm, 2nd shift from 4pm to 0am. Hourly rates are the same for all hourly employees.
- Each employee, after signing the contract, is allowed to register his available working time. The number of sessions that can work in each week that each registered employee must meet the prescribed minimum threshold. This information may change on a weekly basis, before scheduling work for the next week.
- Management will be based on the registration schedule of each employee to schedule the next week. Make sure each shift has enough N employees to work. If there is a shift where the number of registered employees is greater than N, priority will be given to the employees who are working fewer hours. The next week schedule will be announced to all staff.
- When coming to work, the employee scans the check-in card to work, when returning, the employee scans the checkout card to return.
- Employee wages are calculated based on the actual hours worked by the employee and are paid weekly. If an employee works more than 8 hours, the salary for the extra time will be calculated. If an employee arrives late or leaves early, the time of absence will be deducted.

Module "***Register for next week***": A staff selects the function to register for the next week's shift as reqired by an employee → Employee search interface appears → The staff enters the employee's name and clicks search → The interface shows up a list of employees whose names contain the entered keyword → The staff click the correct employee -> The interface to register next week's shift for the selected employee shows up, containing the employee information and a table with 7 lines corresponding to 7 days of the next week, each line has 2 check boxes corresponding to the shift → The staff clicks on the boxes corresponding to the shifts that the employee registered to do and click save → The system saves.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 40
Duration: 60 minutes

A client requires us to develop a software to manage the (part-time) employees of stores with the following description:

- The restaurant chain has many restaurants. Each restaurant has many hourly employees. Each working day has 2 shifts, 1st shift from 8am to 4pm, 2nd shift from 4pm to 0am. Hourly rates are the same for all hourly employees.
- Each employee, after signing the contract, is allowed to register his available working time. The number of sessions that can work in each week that each registered employee must meet the prescribed minimum threshold. This information may change on a weekly basis, before scheduling work for the next week.
- Management will be based on the registration schedule of each employee to schedule the next week. Make sure each shift has enough N employees to work. If there is a shift where the number of registered employees is greater than N, priority will be given to the employees who are working fewer hours. The next week schedule will be announced to all staff.
- When coming to work, the employee scans the check-in card to work, when returning, the employee scans the checkout card to return.
- Employee wages are calculated based on the actual hours worked by the employee and are paid weekly. If an employee works more than 8 hours, the salary for the extra time will be calculated. If an employee arrives late or leaves early, the time of absence will be deducted.

Module "**Schedule next week**": A staff selects the function to schedule the next week → The scheduling interface appears including a table with 7 lines corresponding to 7 days of the next week, each row has 2 columns corresponding to 2 shifts of the day. Each column contains the names of employees registered for that shift → The staff clicks on a shift → The interface shows a list of employees who have registered to that shift and have not been assigned to that shift, one employee per line: name, phone number, total scheduled hours for next week, sort in ascending order of total scheduled hours for next week → The staff clicks on some employee and clicks the select button → Interface returns to schedule page with the information of selected employees is added to the column of the corresponding shift → The staff repeats the above selection steps until the end of the next week's shift and clicks save → The system saves.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 41
Duration: 60 minutes

A client requires us to develop a software to manage the (part-time) employees of stores with the following description:

- The restaurant chain has many restaurants. Each restaurant has many hourly employees. Each working day has 2 shifts, 1st shift from 8am to 4pm, 2nd shift from 4pm to 0am. Hourly rates are the same for all hourly employees.
- Each employee, after signing the contract, is allowed to register his available working time. The number of sessions that can work in each week that each registered employee must meet the prescribed minimum threshold. This information may change on a weekly basis, before scheduling work for the next week.
- Management will be based on the registration schedule of each employee to schedule the next week. Make sure each shift has enough N employees to work. If there is a shift where the number of registered employees is greater than N, priority will be given to the employees who are working fewer hours. The next week schedule will be announced to all staff.
- When coming to work, the employee scans the check-in card to work, when returning, the employee scans the checkout card to return.
- Employee wages are calculated based on the actual hours worked by the employee and are paid weekly. If an employee works more than 8 hours, the salary for the extra time will be calculated. If an employee arrives late or leaves early, the time of absence will be deducted.

Mmodule "**Checkin/Checkout**": Checkin and checkout can be done by the employee's card scanning, or by a staff updating directly on the computer: The staff chooses checkin (or checkout) → The employee's code input screen appears → The staff enters the employee's code and clicks submit → The system saves and reports the checkin (checkout) time.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 42
Duration: 60 minutes

A client requires us to develop a software to manage the (part-time) employees of stores with the following description:

- The restaurant chain has many restaurants. Each restaurant has many hourly employees. Each working day has 2 shifts, 1st shift from 8am to 4pm, 2nd shift from 4pm to 0am. Hourly rates are the same for all hourly employees.
- Each employee, after signing the contract, is allowed to register his available working time. The number of sessions that can work in each week that each registered employee must meet the prescribed minimum threshold. This information may change on a weekly basis, before scheduling work for the next week.
- Management will be based on the registration schedule of each employee to schedule the next week. Make sure each shift has enough N employees to work. If there is a shift where the number of registered employees is greater than N, priority will be given to the employees who are working fewer hours. The next week schedule will be announced to all staff.
- When coming to work, the employee scans the check-in card to work, when returning, the employee scans the checkout card to return.
- Employee wages are calculated based on the actual hours worked by the employee and are paid weekly. If an employee works more than 8 hours, the salary for the extra time will be calculated. If an employee arrives late or leaves early, the time of absence will be deducted.

Module "**Calculate this week's wages**": A staff selects the function of calculating wages for employees during the week → The calculation interface appears with a box to enter the time period for calculation → The staff enters the starting date, the end of the last week → The interface shows a list of wages for all employees in that week, each employee on one line, in the alphabetic order of name: code, name, phone number, total hours worked in shift, total money in shift, total overtime, total number of hours late coming home early, total fines, total last received → The staff clicks on a line of an employee→ The interface shows a detailed statistics table of working hours of the selected employee in that week, each line corresponds to 1 working shift: day, date, shift, checkin time, checkout time, number hours in shift, amount in shift, number of overtime hours, overtime amount, number of hours of late arrival and early return, fines, total amount received for the shift.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 43
Duration: 60 minutes

A client requires us to develop a software to manage the (part-time) employees of stores with the following description:

- The restaurant chain has many restaurants. Each restaurant has many hourly employees. Each working day has 2 shifts, 1st shift from 8am to 4pm, 2nd shift from 4pm to 0am. Hourly rates are the same for all hourly employees.
- Each employee, after signing the contract, is allowed to register his available working time. The number of sessions that can work in each week that each registered employee must meet the prescribed minimum threshold. This information may change on a weekly basis, before scheduling work for the next week.
- Management will be based on the registration schedule of each employee to schedule the next week. Make sure each shift has enough N employees to work. If there is a shift where the number of registered employees is greater than N, priority will be given to the employees who are working fewer hours. The next week schedule will be announced to all staff.
- When coming to work, the employee scans the check-in card to work, when returning, the employee scans the checkout card to return.
- Employee wages are calculated based on the actual hours worked by the employee and are paid weekly. If an employee works more than 8 hours, the salary for the extra time will be calculated. If an employee arrives late or leaves early, the time of absence will be deducted.

Module "Statistics of employees": A staff selects the function of employee statistics → Statistical interface appears with a box to enter the period of statistics → The staff enters the start date, end date of the period → The interface shows a list of statistics tables for all employee in that period, each on 1 line, sorted in descending order of the total number of hours: code, name, phone number, total number of hours worked in shift, total overtime hours, total number of hours late to return early, total hours worked last, total amount received last → The staff clicks on a line to view details → The interface shows detailed statistics of working hours of selected employees in that period, each line corresponds to 1 working shift: day, date, shift, checkin time, hour checkout, the number of hours in the shift, the number of overtime hours, the number of hours that go late and return early, the total time actually worked, the total amount actually received for the shift.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 44
Duration: 60 minutes

A client requires us to develop a software to manage the (part-time) employees of stores with the following description:

- The restaurant chain has many restaurants. Each restaurant has many hourly employees. Each working day has 2 shifts, 1st shift from 8am to 4pm, 2nd shift from 4pm to 0am. Hourly rates are the same for all hourly employees.
- Each employee, after signing the contract, is allowed to register his available working time. The number of sessions that can work in each week that each registered employee must meet the prescribed minimum threshold. This information may change on a weekly basis, before scheduling work for the next week.
- Management will be based on the registration schedule of each employee to schedule the next week. Make sure each shift has enough N employees to work. If there is a shift where the number of registered employees is greater than N, priority will be given to the employees who are working fewer hours. The next week schedule will be announced to all staff.
- When coming to work, the employee scans the check-in card to work, when returning, the employee scans the checkout card to return.
- Employee wages are calculated based on the actual hours worked by the employee and are paid weekly. If an employee works more than 8 hours, the salary for the extra time will be calculated. If an employee arrives late or leaves early, the time of absence will be deducted.

Module "**Statistics of best employees**": A staff selects the function of statistics of best employees → Statistical interface appears with a box to enter the period of statistics → The staff enters the start date, end date of the period → The interface shows a list of employee in that period, each on 1 line, sorted in ascending order of the total number of hours late/early: code, name, phone number, total hours actually worked, total money received, total hours late, total fines → The staff clicks on a line to see details → The interface appears on the dashboard detailed list of working hours of employees selected during that period, each line corresponds to 1 working shift: day, date, shift, checkin time, checkout time, actual working hours, amount actually received, the number of hours going late, the amount of the fine.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 45
Duration: 60 minutes

A client requires us to develop a cinema chain management software with the following description:

- The company has a chain of cinemas (cinema code, theater name, address, introduction).
- Each cinema has many screening rooms (room code, number of seats, room characteristics)
- Each movie (Movie ID, movie title, type, year of production, description) can be shown in different cinemas at different times
- Each screening room can show many movies at different times
- At the time, in a screening room only one movie is shown, and sold at a fixed ticket price.
- The same movie, showing at the same screening room, but in different time slot and dates may have different ticket prices.
- Same show, different seats may have different ticket prices.
- Employees only sell tickets to customers when the screening room at the showtime requested by the customer is still full of empty seats for the customer.
- When buying tickets, customers are issued an invoice containing the tickets purchased. Each ticket on one line: movie name, screening room, showtime, seats, offer, price. The next line is the total amount.
- The cinema also sells fast food services (such as popcorn, drinking water...). Customers can purchase with movie tickets (in which case, the bill will include these services), or buy separately. If buying separately, issue a separate invoice, each line is an item: code, name, unit price, quantity, incentives, money. The next line is the total amount.

Module "**Selling movie tickets**": A staff selects the ticketing menu when a customer requires → the ticketing page appears → The staff selects a screening room or movie name in the drop-out list (as the request of the guest) + select the show time slot → The staff asks the customer to choose the available seats in the screening room → print out the ticket and invoice for the customer which contains a list of ticket, a ticket has: room name, seat number, date showtime, movie name, price per ticket. The next line is total bill amount.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 46
Duration: 60 minutes

A client requires us to develop a cinema chain management software with the following description:

- The company has a chain of cinemas (cinema code, theater name, address, introduction).
- Each cinema has many screening rooms (room code, number of seats, room characteristics)
- Each movie (Movie ID, movie title, type, year of production, description) can be shown in different cinemas at different times
- Each screening room can show many movies at different times
- At the time, in a screening room only one movie is shown, and sold at a fixed ticket price.
- The same movie, showing at the same screening room, but in different time slot and dates may have different ticket prices.
- Same show, different seats may have different ticket prices.
- Employees only sell tickets to customers when the screening room at the showtime requested by the customer is still full of empty seats for the customer.
- When buying tickets, customers are issued an invoice containing the tickets purchased. Each ticket on one line: movie name, screening room, showtime, seats, offer, price. The next line is the total amount.
- The cinema also sells fast food services (such as popcorn, drinking water...). Customers can purchase with movie tickets (in which case, the bill will include these services), or buy separately. If buying separately, issue a separate invoice, each line is an item: code, name, unit price, quantity, incentives, money. The next line is the total amount.

Module "**Schedule showing**": A staff selects the showtime management menu → selects a new showtime schedule → the show scheduler interface shows out → The staff selects movie from the drop-down list + selects a screening room from the drop-down list + showtimes and selects ticket prices from the drop-down list + click add showtimes available room of the time slot or available time slot of the selected screening room → The fare pricing interface appears with the default fare for all seats of the show -> The staff can select some seats and prices and confirm -> System save to database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 47
Duration: 60 minutes

A client requires us to develop a cinema chain management software with the following description:

- The company has a chain of cinemas (cinema code, theater name, address, introduction).
- Each cinema has many screening rooms (room code, number of seats, room characteristics)
- Each movie (Movie ID, movie title, type, year of production, description) can be shown in different cinemas at different times
- Each screening room can show many movies at different times
- At the time, in a screening room only one movie is shown, and sold at a fixed ticket price.
- The same movie, showing at the same screening room, but in different time slot and dates may have different ticket prices.
- Same show, different seats may have different ticket prices.
- Employees only sell tickets to customers when the screening room at the showtime requested by the customer is still full of empty seats for the customer.
- When buying tickets, customers are issued an invoice containing the tickets purchased. Each ticket on one line: movie name, screening room, showtime, seats, offer, price. The next line is the total amount.
- The cinema also sells fast food services (such as popcorn, drinking water...). Customers can purchase with movie tickets (in which case, the bill will include these services), or buy separately. If buying separately, issue a separate invoice, each line is an item: code, name, unit price, quantity, incentives, money. The next line is the total amount.

Module "**Selling food**": A staff selects the food service sale menu when a customer requires → a sales page appears → The staff repeats the following steps until all the items requested by the customer: enter the name of the item and click search → the interface of the list of items containing the entered keyword appears → clicks on an item → the interface choose the size, the quantity will appear → selects the size of the food and drink, enters the quantity and clicks OK → the interface of the selected items will appear as the invoice contains the items, each line contains: code, name, size, unit price, quantity, amount. The last line is the total amount → The staff clicks to pay. The system prints out invoices for customers.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 48
Duration: 60 minutes

A client requires us to develop a cinema chain management software with the following description:

- The company has a chain of cinemas (cinema code, theater name, address, introduction).
- Each cinema has many screening rooms (room code, number of seats, room characteristics)
- Each movie (Movie ID, movie title, type, year of production, description) can be shown in different cinemas at different times
- Each screening room can show many movies at different times
- At the time, in a screening room only one movie is shown, and sold at a fixed ticket price.
- The same movie, showing at the same screening room, but in different time slot and dates may have different ticket prices.
- Same show, different seats may have different ticket prices.
- Employees only sell tickets to customers when the screening room at the showtime requested by the customer is still full of empty seats for the customer.
- When buying tickets, customers are issued an invoice containing the tickets purchased. Each ticket on one line: movie name, screening room, showtime, seats, offer, price. The next line is the total amount.
- The cinema also sells fast food services (such as popcorn, drinking water...). Customers can purchase with movie tickets (in which case, the bill will include these services), or buy separately. If buying separately, issue a separate invoice, each line is an item: code, name, unit price, quantity, incentives, money. The next line is the total amount.

Module "**Revenue Statistics**": A staff selects the statistics menu → selects revenue statistics by movie (or by theater) → enters the start and end time of statistics → a list of available movies (cinemas) appears, one line for each movie: Code, movie name, total number of tickets sold, total revenue earned, sorted in descending order of total revenue -> clicks on a line of a movie (theater) -> The system displays the details of the total income for each showing of the movie, each corresponding line: showtime, quantity tickets sold, total income, sorted in the order of showing from old to new -> clicks on a show to display a list of invoices sold for that show, each invoice on 1 line arranged by payment time: code, customer name if any, total number of tickets, total amount of the bill (only those tickets related to that show in the invoice).

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 49
Duration: 60 minutes

A client requires us to develop a cinema chain management software with the following description:

- The company has a chain of cinemas (cinema code, theater name, address, introduction).
- Each cinema has many screening rooms (room code, number of seats, room characteristics)
- Each movie (Movie ID, movie title, type, year of production, description) can be shown in different cinemas at different times
- Each screening room can show many movies at different times
- At the time, in a screening room only one movie is shown, and sold at a fixed ticket price.
- The same movie, showing at the same screening room, but in different time slot and dates may have different ticket prices.
- Same show, different seats may have different ticket prices.
- Employees only sell tickets to customers when the screening room at the showtime requested by the customer is still full of empty seats for the customer.
- When buying tickets, customers are issued an invoice containing the tickets purchased. Each ticket on one line: movie name, screening room, showtime, seats, offer, price. The next line is the total amount.
- The cinema also sells fast food services (such as popcorn, drinking water...). Customers can purchase with movie tickets (in which case, the bill will include these services), or buy separately. If buying separately, issue a separate invoice, each line is an item: code, name, unit price, quantity, incentives, money. The next line is the total amount.

Module "**Statistics of services**": A staff selects the statistics menu → selects statistics list items sold → enters the start and end time of statistics → a list of service items appears, one line for each item: Code, name, total quantity sold, total revenue, sorted in descending order of total revenue. The staff clicks on a line of an item to display the details of the sale of that item, each line corresponds to the information: sale date, unit price, quantity, total amount, sorted in the order of sale date from old to new. For items that are exchanged for points, when statistics are still converted into money as usual for statistics.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 50
Duration: 60 minutes

A client requires us to develop a rental management software for a mini football field with the following description:

- The football field has many mini courts for rent. Depending on customer requirements, it is possible to combine 2 or 4 adjacent small courts into 1 large court for rent.
- Each court can be rented by many customers at different time slot. Each customer can rent many different courts.
- Customers can rent the court by session of the week or by month (on one or a number of fixed sessions a week, within a few specific months).
- When making a contract to rent a court, customers receive a rental voucher. In it, the first line records the date of the contract, the owner's information, and the customer's information. The next lines, each line record a mini court with full information about the court, rental price per session, rental time slot of the week, start date, end date of the rental period, total expected rent. The last line shows the expected rental amount
- When booking a court, customers must deposit in advance. And this deposit information is also clearly stated in the invoice/bill.
- When customers come to play football at the court, the owner can serve refreshments and snacks. What kind of products do customers use each session, how many bottles (packs) of each type, and how much total money is updated into the system. The customer will pay this incidental fee at the end of the rental period.
- When paying for the court rental, the customer receives an invoice detailing the rental information and the cost of the rental, just like the booking slip. There may be some additional sessions arising or rescheduled according to customer requirements. In addition, the next part of the invoice states the food and drink used in each session, each session is listed in a table, in which each line of the table describes an item: code, name, price, quantity used, total money. The total amount of each session and the total amount for the whole booking.
- The yard manager must import the items for sale from many different suppliers (code, name, address, email, phone, description). Each time of importing goods, there is an import invoice specifying supplier information and a list of items, each line: id, name, unit price, quantity, amount. The last line is the total amount.

Module "**Booking**": A customer comes to book a → A staff selects the booking function → the system displays the interface to find an empty court according to the time slot → The staff enters the time slot + select the type of court as requested by the customer + click search → the system displays a list of available courts according to the selected time slot → clicks on a courts → the system displays an interface to fill in customer information → The staff enters a name and search → the system displays a list of customers whose names contains the entered keyword → clicks on the correct customer name with the current customer (if the customer first comes to book a court, must add a new one) → the system displays the interface to enter the time period of the start date, End date of the booking (preferred to book by quarter) → clicks confirm → the system displays a booking slip with full customer information, booking information, booking price, booking time slot, total number session according to the selected time, the estimated total amount and the deposit amount → clicks confirm → the system prints the booking slip and updates it to the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 51
Duration: 60 minutes

A client requires us to develop a rental management software for a mini football field with the following description:

- The football field has many mini courts for rent. Depending on customer requirements, it is possible to combine 2 or 4 adjacent small courts into 1 large court for rent.
- Each court can be rented by many customers at different time slot. Each customer can rent many different courts.
- Customers can rent the court by session of the week or by month (on one or a number of fixed sessions a week, within a few specific months).
- When making a contract to rent a court, customers receive a rental voucher. In it, the first line records the date of the contract, the owner's information, and the customer's information. The next lines, each line record a mini yard with full information about the yard, rental price per session, rental time slot of the week, start date, end date of the rental period, total expected rent. The last line shows the expected rental amount
- When booking a yard, customers must deposit in advance. And this deposit information is also clearly stated in the invoice/bill.
- When customers come to play football at the yard, the owner can serve refreshments and snacks. What kind of products do customers use each session, how many bottles (packs) of each type, and how much total money is updated into the system. The customer will pay this incidental fee at the end of the rental period.
- When paying for the yard rental, the customer receives an invoice detailing the rental information and the cost of the rental, just like the booking slip. There may be some additional sessions arising or rescheduled according to customer requirements. In addition, the next part of the invoice states the food and drink used in each session, each session is listed in a table, in which each line of the table describes an item: code, name, price, quantity used, total money. The total amount of each session and the total amount for the whole booking.
- The yard manager must import the items for sale from many different suppliers (code, name, address, email, phone, description). Each time of importing goods, there is an import invoice specifying supplier information and a list of items, each line: id, name, unit price, quantity, amount. The last line is the total amount.

Module "**Goods importing**": A staff selects the import menu whem import goods from a provider → the import page appears with a box to search for provider by name → The staff enters a name + clicks to search → the system displays a list of the providers whose name contains the entered keyword → clicks on the currently imported provider (if the provider is new, add a new one) →

Repeat the following steps for all imported goods: clicks to search for goods by name → enter name + click search → the system displays a list of the goods whose name contains the name just entered → the staff selects the name of the goods in the list of available goods (if the goods are new, choose to add new) + enter the unit price and quantity → that item will be added to the list of imported goods of the invoice → repeat until all the imported goods are finished, submit → successful import report and print the imported invoice as described.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 52
Duration: 60 minutes

A client requires us to develop a rental management software for a mini football field with the following description:

- The football field has many mini courts for rent. Depending on customer requirements, it is possible to combine 2 or 4 adjacent small courts into 1 large court for rent.
- Each court can be rented by many customers at different time slot. Each customer can rent many different courts.
- Customers can rent the court by session of the week or by month (on one or a number of fixed sessions a week, within a few specific months).
- When making a contract to rent a court, customers receive a rental voucher. In it, the first line records the date of the contract, the owner's information, and the customer's information. The next lines, each line record a mini yard with full information about the yard, rental price per session, rental time slot of the week, start date, end date of the rental period, total expected rent. The last line shows the expected rental amount
- When booking a yard, customers must deposit in advance. And this deposit information is also clearly stated in the invoice/bill.
- When customers come to play football at the yard, the owner can serve refreshments and snacks. What kind of products do customers use each session, how many bottles (packs) of each type, and how much total money is updated into the system. The customer will pay this incidental fee at the end of the rental period.
- When paying for the yard rental, the customer receives an invoice detailing the rental information and the cost of the rental, just like the booking slip. There may be some additional sessions arising or rescheduled according to customer requirements. In addition, the next part of the invoice states the food and drink used in each session, each session is listed in a table, in which each line of the table describes an item: code, name, price, quantity used, total money. The total amount of each session and the total amount for the whole booking.
- The yard manager must import the items for sale from many different suppliers (code, name, address, email, phone, description). Each time of importing goods, there is an import invoice specifying supplier information and a list of items, each line: id, name, unit price, quantity, amount. The last line is the total amount.

Module "**Update used items of the rental session**": When the customer arrives to receive the court and return the court for that session, the staff selects the menu to find the booking ticket by the customer's name → enter the customer's name + click search → the system displays a list of customers with the name entered → selects the correct customer name with the current customer information → the system displays a list of orders that the customer is booking → clicks on the checkout button rental session 1 booking ticket → the system displays an interface to enter the court reception time, return time, and rent (early payment will not be reduced, but late payment will be charged more) + repeat the following steps until the list of food products that customers have used during the rental sessions: click more items used → the interface to search for goods by name appears → enters the name of the goods and search → the interface for the list of goods with the name entered appears → clicks on 1 item → the interface to enter the unit price and quantity appears → enters and confirms → the used item information is added to the list of used items of the session -> The last line is the total amount of customers → clicks to confirm → the system updates to the database (no payment required).

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 53
Duration: 60 minutes

A client requires us to develop a rental management software for a mini football field with the following description:

- The football field has many mini courts for rent. Depending on customer requirements, it is possible to combine 2 or 4 adjacent small courts into 1 large court for rent.
- Each court can be rented by many customers at different time slot. Each customer can rent many different courts.
- Customers can rent the court by session of the week or by month (on one or a number of fixed sessions a week, within a few specific months).
- When making a contract to rent a court, customers receive a rental voucher. In it, the first line records the date of the contract, the owner's information, and the customer's information. The next lines, each line record a mini yard with full information about the yard, rental price per session, rental time slot of the week, start date, end date of the rental period, total expected rent. The last line shows the expected rental amount
- When booking a yard, customers must deposit in advance. And this deposit information is also clearly stated in the invoice/bill.
- When customers come to play football at the yard, the owner can serve refreshments and snacks. What kind of products do customers use each session, how many bottles (packs) of each type, and how much total money is updated into the system. The customer will pay this incidental fee at the end of the rental period.
- When paying for the yard rental, the customer receives an invoice detailing the rental information and the cost of the rental, just like the booking slip. There may be some additional sessions arising or rescheduled according to customer requirements. In addition, the next part of the invoice states the food and drink used in each session, each session is listed in a table, in which each line of the table describes an item: code, name, price, quantity used, total money. The total amount of each session and the total amount for the whole booking.
- The yard manager must import the items for sale from many different suppliers (code, name, address, email, phone, description). Each time of importing goods, there is an import invoice specifying supplier information and a list of items, each line: id, name, unit price, quantity, amount. The last line is the total amount.

Module "**Customer paying**": When a customer comes to pay, a staff selects the menu to find the booking slip by the customer's name → enter the customer's name + click search → the system displays a list of customers have the name just entered → selects the correct customer name with the current customer information → the system displays a list of booking tickets that the customer is booking → clicks on the payment button for 1 booking ticket → the system displays the invoice full customer information + 1 list of food and beverage products that the customer has used during the rental sessions as described above + the last line is the total amount paid (if the customer complains about a change in the quantity or information information about used items, the staff must change, update the detailed list in the corresponding invoice) → clicks confirm → the system updates to the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 54
Duration: 60 minutes

A client requires us to develop a rental management software for a mini football field with the following description:

- The football field has many mini courts for rent. Depending on customer requirements, it is possible to combine 2 or 4 adjacent small courts into 1 large court for rent.
- Each court can be rented by many customers at different time slot. Each customer can rent many different courts.
- Customers can rent the court by session of the week or by month (on one or a number of fixed sessions a week, within a few specific months).
- When making a contract to rent a court, customers receive a rental voucher. In it, the first line records the date of the contract, the owner's information, and the customer's information. The next lines, each line record a mini yard with full information about the yard, rental price per session, rental time slot of the week, start date, end date of the rental period, total expected rent. The last line shows the expected rental amount
- When booking a yard, customers must deposit in advance. And this deposit information is also clearly stated in the invoice/bill.
- When customers come to play football at the yard, the owner can serve refreshments and snacks. What kind of products do customers use each session, how many bottles (packs) of each type, and how much total money is updated into the system. The customer will pay this incidental fee at the end of the rental period.
- When paying for the yard rental, the customer receives an invoice detailing the rental information and the cost of the rental, just like the booking slip. There may be some additional sessions arising or rescheduled according to customer requirements. In addition, the next part of the invoice states the food and drink used in each session, each session is listed in a table, in which each line of the table describes an item: code, name, price, quantity used, total money. The total amount of each session and the total amount for the whole booking.
- The yard manager must import the items for sale from many different suppliers (code, name, address, email, phone, description). Each time of importing goods, there is an import invoice specifying supplier information and a list of items, each line: id, name, unit price, quantity, amount. The last line is the total amount.

Module "**Revenue statistics**": A staff selects the menu of revenue statistics by time (month, quarter, year) → the system displays a box to select statistics by month, quarter, or year → clicka by month → the system displays revenue statistics for the last 12 months in the form of a table, each line corresponds to 1 month (corresponding to quarter, year): month name, total revenue, sorted by time.

The staff clicks on 1 line of the results → the system details the customer's invoices during the click line, each invoice on 1 line: id, customer name, court name, date and time, total payment.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 55
Duration: 60 minutes

A client requires us to develop a rental management software for a mini football field with the following description:

- The football field has many mini courts for rent. Depending on customer requirements, it is possible to combine 2 or 4 adjacent small courts into 1 large court for rent.
- Each court can be rented by many customers at different time slot. Each customer can rent many different courts.
- Customers can rent the court by session of the week or by month (on one or a number of fixed sessions a week, within a few specific months).
- When making a contract to rent a court, customers receive a rental voucher. In it, the first line records the date of the contract, the owner's information, and the customer's information. The next lines, each line record a mini yard with full information about the yard, rental price per session, rental time slot of the week, start date, end date of the rental period, total expected rent. The last line shows the expected rental amount
- When booking a yard, customers must deposit in advance. And this deposit information is also clearly stated in the invoice/bill.
- When customers come to play football at the yard, the owner can serve refreshments and snacks. What kind of products do customers use each session, how many bottles (packs) of each type, and how much total money is updated into the system. The customer will pay this incidental fee at the end of the rental period.
- When paying for the yard rental, the customer receives an invoice detailing the rental information and the cost of the rental, just like the booking slip. There may be some additional sessions arising or rescheduled according to customer requirements. In addition, the next part of the invoice states the food and drink used in each session, each session is listed in a table, in which each line of the table describes an item: code, name, price, quantity used, total money. The total amount of each session and the total amount for the whole booking.
- The yard manager must import the items for sale from many different suppliers (code, name, address, email, phone, description). Each time of importing goods, there is an import invoice specifying supplier information and a list of items, each line: id, name, unit price, quantity, amount. The last line is the total amount.

Module "**Statistics of time slot**": A staff selects the menu of the most rented time slot → Enter the period (start - end date) statistics → The system displays the list of time slot in e table format, each line corresponds to a time slot: time slot, date, column total number of book, column total amount earned, sorted in descending order descending of the total number of book column, followed by the descending of the total revenue column. The staff clicks on a line of a time slot → the system displays a detailed list of times when customer book a court in that time slot, each time on a line: id, customer's name, court name, date and time, price, total amount .

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 56
Duration: 60 minutes

A client requires us to develop a software to manage installment loans with the following description:

- The company cooperates with many partners which are retailers of products with many categories from phones, computers, electronics, refrigeration, home appliances, cars, real estate .. .
- When a customer buys one or several phone models and needs to use the installment service, the staff will carry out procedures to sign an installment loan contract for that customer. The contract contains information about the company's representative, customer information, signing date, and a list of items, each item on one line: code, name, unit of measure, unit price, quantity, amount. The next line is the total amount and loan term. Next is a list of payment times, each installment on 1 line: payment date, total payment amount, total outstanding balance.
- Each item has its own list price, the company paying for the item will receive a discount, the company will collect it from the customer at an interest rate based on the listed price of the item.
- Customers can pay installments for each contract once a month, for the optional period of the contract.
- Customers can pay before the due date but the payment value remains unchanged
- If the customer makes late payment compared to the monthly deadline, the late balance will be included in the principal and interest will be calculated according to the principal.
- The company can pay the item bill for each item or in installments for a period of 1 week, 1 month... Each payment will save the invoice with full information of the company representative, the representative of the partner, payment date, total payment and list of paid items, each customer's purchase on 1 line: code, name of customer, date of purchase, unit of measure, quantity, unit price, cost money.

Module "**Signing a contract**": A staff selects the function to sign a new contract with the customer → Customer search interface appears → enters the customer's name and clicks to search → The interface displays a list of customers whose name contains the entered keyword -> selects the correct customer (if not, it is has to enter new customer information to enter and continue) → Repeat the following steps for all the items purchased by the customer: item search interface appears → enters the item name and clicks search → The interface shows a list of item with the names containing the entered keywords -> Chooses the correct item and enters the quantity and unit price -> After finishing all the items, chooses to continue - > Interface for entering loan term and interest rate -> The system automatically calculates the time to pay monthly and the payment amount in the form of a table, each line corresponds to: the time to pay, the total amount to be paid. payment, outstanding balance the rest → reconfirms with the customer and clicks save → The system saves and prints the contract to the customer.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 57
Duration: 60 minutes

A client requires us to develop a software to manage installment loans with the following description:

- The company cooperates with many partners which are retailers of products with many categories from phones, computers, electronics, refrigeration, home appliances, cars, real estate .. .
- When a customer buys one or several phone models and needs to use the installment service, the staff will carry out procedures to sign an installment loan contract for that customer. The contract contains information about the company's representative, customer information, signing date, and a list of items, each item on one line: code, name, unit of measure, unit price, quantity, amount. The next line is the total amount and loan term. Next is a list of payment times, each installment on 1 line: payment date, total payment amount, total outstanding balance.
- Each item has its own list price, the company paying for the item will receive a discount, the company will collect it from the customer at an interest rate based on the listed price of the item.
- Customers can pay installments for each contract once a month, for the optional period of the contract.
- Customers can pay before the due date but the payment value remains unchanged
- If the customer makes late payment compared to the monthly deadline, the late balance will be included in the principal and interest will be calculated according to the principal.
- The company can pay the item bill for each item or in installments for a period of 1 week, 1 month... Each payment will save the invoice with full information of the company representative, the representative of the partner, payment date, total payment and list of paid items, each customer's purchase on 1 line: code, name of customer, date of purchase, unit of measure, quantity, unit price, cost money.

Module "**Customers paying**": A staff selects the function of receiving payment from Customer (Customer can pay directly at the counter, transfer or online - this is the description for direct payment) → Displays interface to find contract information -> The staff enters contract code -> Show details of contract, history of payments, total outstanding balance and payable amount -> The staff informs customer the amount to be paid payment and total outstanding balance, asks the customer how many installments or total amount (customer can pay in advance for the next installments, or pay not enough for this installment) → selects payment due that customer wants to pay payment, enter the amount → displays payment invoice containing information of customer, company representative, list of items as in the original contract, total payment, total outstanding balance, and a list of remaining payments -> The staff confirms with customer and click save -> System saves and prints the invoice for the staff to deliver to customer.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 58
Duration: 60 minutes

A client requires us to develop a software to manage installment loans with the following description:

- The company cooperates with many partners which are retailers of products with many categories from phones, computers, electronics, refrigeration, home appliances, cars, real estate .. .
- When a customer buys one or several phone models and needs to use the installment service, the staff will carry out procedures to sign an installment loan contract for that customer. The contract contains information about the company's representative, customer information, signing date, and a list of items, each item on one line: code, name, unit of measure, unit price, quantity, amount. The next line is the total amount and loan term. Next is a list of payment times, each installment on 1 line: payment date, total payment amount, total outstanding balance.
- Each item has its own list price, the company paying for the item will receive a discount, the company will collect it from the customer at an interest rate based on the listed price of the item.
- Customers can pay installments for each contract once a month, for the optional period of the contract.
- Customers can pay before the due date but the payment value remains unchanged
- If the customer makes late payment compared to the monthly deadline, the late balance will be included in the principal and interest will be calculated according to the principal.
- The company can pay the item bill for each item or in installments for a period of 1 week, 1 month... Each payment will save the invoice with full information of the company representative, the representative of the partner, payment date, total payment and list of paid items, each customer's purchase on 1 line: code, name of customer, date of purchase, unit of measure, quantity, unit price, cost money.

Module "**Payment to partners**": A staff selects the payment function for a partner → The partner search interface appears → The staff enters the partner name and clicks to search → The interface pops up a list of partner with the name containing the entered keyword -> Select the correct partner → The interface shows a list of Contracts of the customer who buys from that partner but the company has not paid appears, each contract on 1 line: id, customer name, total item, total amount of customer, amount to pay for patner -> The staff selects a number of contracts to pay for partner and clicks next -> Shows partner payment invoice interface with full enough information as described above -> confirms with item and click save → System saves and prints invoice for partner to sign for the staff to save.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 59
Duration: 60 minutes

A client requires us to develop a software to manage installment loans with the following description:

- The company cooperates with many partners which are retailers of products with many categories from phones, computers, electronics, refrigeration, home appliances, cars, real estate .. .
- When a customer buys one or several phone models and needs to use the installment service, the staff will carry out procedures to sign an installment loan contract for that customer. The contract contains information about the company's representative, customer information, signing date, and a list of items, each item on one line: code, name, unit of measure, unit price, quantity, amount. The next line is the total amount and loan term. Next is a list of payment times, each installment on 1 line: payment date, total payment amount, total outstanding balance.
- Each item has its own list price, the company paying for the item will receive a discount, the company will collect it from the customer at an interest rate based on the listed price of the item.
- Customers can pay installments for each contract once a month, for the optional period of the contract.
- Customers can pay before the due date but the payment value remains unchanged
- If the customer makes late payment compared to the monthly deadline, the late balance will be included in the principal and interest will be calculated according to the principal.
- The company can pay the item bill for each item or in installments for a period of 1 week, 1 month... Each payment will save the invoice with full information of the company representative, the representative of the partner, payment date, total payment and list of paid items, each customer's purchase on 1 line: code, name of customer, date of purchase, unit of measure, quantity, unit price, cost money.

Module "**Customer statistics by dept**": A staff selects the function of customer statistics by dept → The interface displays a list of statistics for all customers, each customer on 1 line, sorted by order of total dept in descending order: code, name, phone number, total outstanding balance, total dept remain → clicks on 1 line to view details → The interface shows the list of contracts of that customer, each line corresponds to 1 contract: code, signing date, total loan amount, total number of payments, total outstanding balance, total dept remain -> click on 1 line -> display details corresponding contract: customer information, phone number, list of items, quantity, unit price, total amount; list of payments, status of payment completed or not.

1. Write a standard scenario for this module

2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 60
Duration: 60 minutes

A client requires us to develop a software to manage installment loans with the following description:

- The company cooperates with many partners which are retailers of products with many categories from phones, computers, electronics, refrigeration, home appliances, cars, real estate .. .
- When a customer buys one or several phone models and needs to use the installment service, the staff will carry out procedures to sign an installment loan contract for that customer. The contract contains information about the company's representative, customer information, signing date, and a list of items, each item on one line: code, name, unit of measure, unit price, quantity, amount. The next line is the total amount and loan term. Next is a list of payment times, each installment on 1 line: payment date, total payment amount, total outstanding balance.
- Each item has its own list price, the company paying for the item will receive a discount, the company will collect it from the customer at an interest rate based on the listed price of the item.
- Customers can pay installments for each contract once a month, for the optional period of the contract.
- Customers can pay before the due date but the payment value remains unchanged
- If the customer makes late payment compared to the monthly deadline, the late balance will be included in the principal and interest will be calculated according to the principal.
- The company can pay the item bill for each item or in installments for a period of 1 week, 1 month... Each payment will save the invoice with full information of the company representative, the representative of the partner, payment date, total payment and list of paid items, each customer's purchase on 1 line: code, name of customer, date of purchase, unit of measure, quantity, unit price, cost money.

Module "**Statistics of partners**": A staff selects the function of partner statistics by sales → Statistical interface appears with the box to enter the statistical period → enters the start date, the end date of the statistical period → The interface shows a list of partner, one per line, in descending order of total revenue: code, name, address/branch, total number of invoices, total sales payment, total outstanding balance → clicks on 1 line to see details → The interface shows a list of contracts related to that patner, each line corresponds to 1 contract: code, customer's name, signing date, total loan amount, total number of payments, total outstanding balance, total overdue balance -> click on 1 line -> display corresponding contract details: customer information, phone number, list of items, quantity, unit price, total amount; list of payments, status of payment completed or not.

1. Write a standard scenario for this module

2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 61
Duration: 60 minutes

A client requires us to develop a software to manage installment loans with the following description:

- The company cooperates with many partners which are retailers of products with many categories from phones, computers, electronics, refrigeration, home appliances, cars, real estate .. .
- When a customer buys one or several phone models and needs to use the installment service, the staff will carry out procedures to sign an installment loan contract for that customer. The contract contains information about the company's representative, customer information, signing date, and a list of items, each item on one line: code, name, unit of measure, unit price, quantity, amount. The next line is the total amount and loan term. Next is a list of payment times, each installment on 1 line: payment date, total payment amount, total outstanding balance.
- Each item has its own list price, the company paying for the item will receive a discount, the company will collect it from the customer at an interest rate based on the listed price of the item.
- Customers can pay installments for each contract once a month, for the optional period of the contract.
- Customers can pay before the due date but the payment value remains unchanged
- If the customer makes late payment compared to the monthly deadline, the late balance will be included in the principal and interest will be calculated according to the principal.
- The company can pay the item bill for each item or in installments for a period of 1 week, 1 month... Each payment will save the invoice with full information of the company representative, the representative of the partner, payment date, total payment and list of paid items, each customer's purchase on 1 line: code, name of customer, date of purchase, unit of measure, quantity, unit price, cost money.

Module "**Statistics of product**": A staff selects the function of item statistics by revenue → Statistical interface appears with a box to enter the statistical period → enters the start date, the end date of the statistical period → The interface displays a list of statistics for all lines/categories of item in that period, each item per line, sorted in descending order of total sales. income: item, total invoice number, total interest revenue (only the interest earned) → clicks on 1 line to see details → The interface shows a detailed statistics table of item are selected in that period, each line corresponds to 1 item, in descending order of total profit and loss: code, name, total invoice, total profit -> Click on 1 line, pop up list of contracts of that item in the statistical period, arranged in chronological order: code, customer name, total loan value, total interest earned.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 62
Duration: 60 minutes

A client requires us to develop a software to help them manage their rental costumes with the following description:

- The store has many costumes, of different categories, a costume can have different quantities.
- Costume is ordered or pre-imported from suppliers. Each suplier can provide different costume types. Each import can import many types of costume from the same provider, each costume has a different amount.
- Customers can rent many times, each time renting many different costumes, each costume has a different number. If renting for the first time, a deposit must be equal to the total original value of the rental costumes, if renting many times (customers), the deposit will be decided by the staff making the invoice.
- When paying, customers can pay part or all of the rental costumes in one time, each payment has a payment voucher corresponding to the returned costumes. The deposit is only returned to the customer when all rental costumes have been returned. In case the customer pays a part of the costumes, after paying, the remaining deposit is more than the original value of the rental costumes, the customer is entitled to receive the remaining balance, only keeping the maximum deposit equal to the original value of the rental costumes.
- When paying, if the costume is damaged or dirty, the customer must pay a fine. A costume can have multiple errors concurrently. Fines for each error are estimated by the the staff.

Module "**Import costume**": A staff selects the function to import costumes from a provider → The interface to find provider by name appears -> The staff enters the name of the provider and finds it -> Shows a list of providers containing the new name -> Clicks the correct provider (if not in the list of results, switch to the interface to enter new provider information and continue) → Repeat until all costumes need to be imported from that provider: select search costumes by name - > select and enter quantity, unit price → confirms the invoice entered with provider and pay to the provider, receive costumes → The system saves and prints the invoice to ask the provider to sign and save.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 63
Duration: 60 minutes

A client requires us to develop a software to help them manage their rental costumes with the following description:

- The store has many costumes, of different categories, a costume can have different quantities.
- Costume is ordered or pre-imported from suppliers. Each suplier can provide different costume types. Each import can import many types of costume from the same provider, each costume has a different amount.
- Customers can rent many times, each time renting many different costumes, each costume has a different number. If renting for the first time, a deposit must be equal to the total original value of the rental costumes, if renting many times (customers), the deposit will be decided by the staff making the invoice.
- When paying, customers can pay part or all of the rental costumes in one time, each payment has a payment voucher corresponding to the returned costumes. The deposit is only returned to the customer when all rental costumes have been returned. In case the customer pays a part of the costumes, after paying, the remaining deposit is more than the original value of the rental costumes, the customer is entitled to receive the remaining balance, only keeping the maximum deposit equal to the original value of the rental costumes.
- When paying, if the costume is damaged or dirty, the customer must pay a fine. A costume can have multiple errors concurrently. Fines for each error are estimated by the the staff.

Module "**Costume renting**": After choosing the costumes to borrow, a customer brings them to the cashier's counter to make a loan slip. The staff enters the customer's name and searches → The system returns a list of customers with the name entered → The staff clicks on the customer's name in the list (if the customer borrows for the first time, enter a new one) → The system displays the interface to add borrowed costumes: For each costume, the staff clicks to find the costume by name → enter the name of the costume + click search → the system displays a list of costumes with the name entered → clicks on the right line with the costume selected by the customer + enter the quantity → The system adds a line corresponding to that costume in the rental slip as described. The total deposit is equal to the total cost of the costumes and is automatically calculated at the end of the bill. -> clicks to create a loan slip → The system saves it in the database and prints out the loan slip for the customer and receives the deposit.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 64
Duration: 60 minutes

A client requires us to develop a software to help them manage their rental costumes with the following description:

- The store has many costumes, of different categories, a costume can have different quantities.
- Costume is ordered or pre-imported from suppliers. Each suplier can provide different costume types. Each import can import many types of costume from the same provider, each costume has a different amount.
- Customers can rent many times, each time renting many different costumes, each costume has a different number. If renting for the first time, a deposit must be equal to the total original value of the rental costumes, if renting many times (customers), the deposit will be decided by the staff making the invoice.
- When paying, customers can pay part or all of the rental costumes in one time, each payment has a payment voucher corresponding to the returned costumes. The deposit is only returned to the customer when all rental costumes have been returned. In case the customer pays a part of the costumes, after paying, the remaining deposit is more than the original value of the rental costumes, the customer is entitled to receive the remaining balance, only keeping the maximum deposit equal to the original value of the rental costumes.
- When paying, if the costume is damaged or dirty, the customer must pay a fine. A costume can have multiple errors concurrently. Fines for each error are estimated by the the staff.

Module "**Customer returns and pays**": When a customer brings the costumes back to return, a staff chooses the menu to find the list of borrowed costumes by the customer's name → enter the customer's name + click search → the system displays a list of customers whose names contains the entered keyword → The staff selects the correct customer name with the current customer information → the system displays a list of the costumes that the customer is borrowing, each costume on a line with full information about the costume, loan date, loan price per day, number of days borrowed, and rental amount up to the date of payment, the last column is the check box to select pay → The staff clicks on the pay button for the costumes that the customer returns (with may not pay in full), enter the status of the costume and the fine if any, finally clicks the payment button → the system displays the invoice with full customer information + a list of the costumes to be returned as described above + the last line is the total amount paid, the amount deposited, the amount the customer has to pay or return to the customer → clicks confirm → the system updates to the database.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 65
Duration: 60 minutes

A client requires us to develop a software to help them manage their rental costumes with the following description:

- The store has many costumes, of different categories, a costume can have different quantities.
- Costume is ordered or pre-imported from suppliers. Each suplier can provide different costume types. Each import can import many types of costume from the same provider, each costume has a different amount.
- Customers can rent many times, each time renting many different costumes, each costume has a different number. If renting for the first time, a deposit must be equal to the total original value of the rental costumes, if renting many times (customers), the deposit will be decided by the staff making the invoice.
- When paying, customers can pay part or all of the rental costumes in one time, each payment has a payment voucher corresponding to the returned costumes. The deposit is only returned to the customer when all rental costumes have been returned. In case the customer pays a part of the costumes, after paying, the remaining deposit is more than the original value of the rental costumes, the customer is entitled to receive the remaining balance, only keeping the maximum deposit equal to the original value of the rental costumes.
- When paying, if the costume is damaged or dirty, the customer must pay a fine. A costume can have multiple errors concurrently. Fines for each error are estimated by the the staff.

Module "**Statistics of costumes**": A staff selects the menu of statistics of costumes → Enter the time period (start - end date) statistics → The system displays the list of costumes borrowed in the form of a table, each line corresponds to a costume with complete information: code, name, model, genre, column total number of loans, column total amount collected (Sorted in descending order of the total borrowed column, followed by the descending of the total proceeds column). The staff clicks on a line of an costume → the system displays the details of the invoice with the borrowed costume, each invoice on 1 line: id, name of the borrower, borrowed date and time, payment date and time, total amount.

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module

Software engineering
Subject No. 66
Duration: 60 minutes

A client requires us to develop a software to help them manage their rental costumes with the following description:

- The store has many costumes, of different categories, a costume can have different quantities.
- Costume is ordered or pre-imported from suppliers. Each suplier can provide different costume types. Each import can import many types of costume from the same provider, each costume has a different amount.
- Customers can rent many times, each time renting many different costumes, each costume has a different number. If renting for the first time, a deposit must be equal to the total original value of the rental costumes, if renting many times (customers), the deposit will be decided by the staff making the invoice.
- When paying, customers can pay part or all of the rental costumes in one time, each payment has a payment voucher corresponding to the returned costumes. The deposit is only returned to the customer when all rental costumes have been returned. In case the customer pays a part of the costumes, after paying, the remaining deposit is more than the original value of the rental costumes, the customer is entitled to receive the remaining balance, only keeping the maximum deposit equal to the original value of the rental costumes.
- When paying, if the costume is damaged or dirty, the customer must pay a fine. A costume can have multiple errors concurrently. Fines for each error are estimated by the the staff.

Module "**Revenue Statistics**": A staff selects the menu of revenue statistics by time (month, quarter, year) → the system displays a box to select statistics by month, quarter, or year → The staff clicks by month → the system displays the monthly revenue statistics in the form of a table, each line corresponds to 1 month (corresponding to the quarter, year): month name, total revenue (Sorted by chronological order from the nearest month (respectively quarter, year) to the oldest month (respectively quarter, year)). The staff clicks on a line → the system displays the details of invoices in that line, each invoice on 1 line: id, customer name, borrowed date, total number of borrowed clothes, total amount of the bill

1. Write a standard scenario for this module
2. Extract and build class diagram for all related entity classes
3. Static design: Design UI and the detailed class diagram of the design phase for this module
4. Dynamic design: Build the sequence diagram of the design phase for this module
5. Write a standard blacklox testcase for this module