 

Software Engineering-2 Cover Sheet

Project Title: Book for You

Row Number (in PDF): 6 Time: 10:15

|  |  |  |  |
| --- | --- | --- | --- |
| # | Member Name (printed in arabic) | Member Id (printed) | Handwritten signature |
| 1 | الاء حمدي حسن منسي | 20210167 |  |
| 2 | الاء حسن عبد الرحمن عبد الرحمن | 20210166 |  |
| 3 | الاء ممدوح احمد عثمان | 20210172 |  |
| 4 | علا إبراهيم عبد المنعم | 20210569 |  |
| 5 | ساره علي مرسي راجح | 20210398 |  |
| 6 | زينه محمود محمد كمال | 20210391 |  |
| 7 |  |  |  |

Project Requirements(grades)

|  |  |  |
| --- | --- | --- |
| SRS | 2 |  |
| SDD | 2 |  |
| Validation | 3 |  |
| OCL | 4 |  |
| AOP | 4 |  |
| Microservices | 1 |  |
| Cloud | 4 |  |

20

**System Requirements Specification**

**For Online Book Store**

**1. Introduction**

**1.1. Purpose:**

The purpose of this document is to present a detailed description of the Online Book Store. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, this document is intended for both the stakeholders and the developers of the system.

**1.2. Scope:**

This system allows the customers to maintain their cart for add or remove the product over the internet.

# Book Store Function Requirements

**Admin**

**First Admin must log in to the system and can do the following:**

|  |  |
| --- | --- |
| Name | Description |
| login |  |
| Forget password |  |
| View books rate |  |
| View all contact messages |  |
| And replay |  |

**Manage Books**

|  |  |
| --- | --- |
| Name | Description |
| Add Books |  |
| Modify Books |  |
| Remove Books |  |

**Manage Users**

|  |  |
| --- | --- |
| Name | Description |
| Edit User account |  |
| Delete User account |  |

**Manage Orders**

|  |  |
| --- | --- |
| Name | Description |
| View all orders |  |
| Update order statuses | Like pending, processing, shipped |

**Manage offers**

|  |  |
| --- | --- |
| Name | Description |
| Add offer |  |
| Modify offer |  |
| Remove offer |  |

**User**

|  |  |
| --- | --- |
| Name | Description |
| Registration | If a user wants to buy any book, then he/she must be registered. |
| Login |  |
| View account details |  |
| Forget password | Users are presented with a form where they can input their registered email address associated with their account to change password |
| Logout |  |
| Search |  |
| View cart |  |
| Add to shopping cart |  |
| Delete from shopping cart. |  |
| Proceed to check out | if the user wants to continue the payment process. |
| Payment | user choose way to pay |
| Place order | user can order own selected books. |
| Display bill |  |
| Rate | customers can rate and add a review for a product. |
| View offers |  |
| Contact with admin |  |

3. Non-Function Requirements

1- Performance:

● The system shall respond to the request from the user in less than 3 seconds.

● The system shall be reliable and flexible.

●Application shall handle more than 1user simultaneously. ● Application shall on average operate without failure for as long a time as possible.

2- Maintainability:

● system shall be easy to change/ update.

3- Serviceability:

● system shall be easy to maintain, including tasks such as monitoring the system, repairing problems that arise, adding and removing users from the system.

4- Data integrity:

● system shall be described by accuracy, completeness, and consistency of data mean safety of data and security.

5- Manageability:

● system shall be efficient and easy to monitor and maintain to keep the system performing, secure, and running smoothly.

6- interface:

● The system must be with a good interface; any user can understand it.

7- look and feel:

● The system shall be painted in comfortable colors for the eyes.

8- Safety and Security:

● The system shall ensure that only authorized users are able to gain access.

● The system shall distinguish between authorized and non- authorized users.

9- Usability:

● The system shall have a user-friendly interface.

● The system shall be easy to use by different users.

10- Availability:

● The system shall be available 24 hours in a day.

11- Reliability:

● The system shall be reliable and do not have data validation failures.

4- Use case diagram

**صورة تحتوي على نص, لقطة شاشة, رسم بياني, شكل

تم إنشاء الوصف تلقائياً**

**Login**

|  |  |
| --- | --- |
| **Identifier and name:** | Login |
| **Initiator:** | User |
| **Goal:** | The user wants to access their account and use the system's features. |
| **Description**: | The user wants to access their account on the application that requires authentication. To do so, they must provide their login credentials. The system will then verify the credentials and allow the user to access their account. If the user forgets their password, they can use the Forgot Password use case to recover it. If the user is new and doesn't have an account, they can use the Register use case to create a new account. If the user enters incorrect login credentials, the system will show a wrong password message and prompt the user to retry. |
| **Preconditions**: | - The user has a valid account on the application, or they have completed the registration process.  - The user knows their login credentials or has completed the forgot password process. |
| **Postconditions:** | - If the login is successful, the user can access their account and perform actions that require authentication.  - If the login fails, the user can’t access their account and must retry the login process, use the Forgot Password use case to recover their password, or use the Register use case to create a new account. |
| **Basic flow (**Main success scenario**):** | 1. The user navigates to the login page.  2. The system presents a form for the user to enter their login credentials.  3. The user enters their username and password.  4. The system verifies the credentials using the Verify Login use case.  5. If the credentials are valid, the system logs the user in and redirects them to their account homepage.  6. If the credentials are invalid, the system shows a wrong password message and prompts the user to retry. |
| **Alternate flows:** | - If the user enters an incorrect username, the system may prompt them to try again or provide additional assistance, such as suggesting they create a new account.  - If the system detects suspicious activity, such as repeated failed login attempts or login attempts from an unfamiliar device or location, it may prompt the user to perform additional security checks, such as entering a verification code or answering security questions.  **Included Verify Login use case:**  1. The system receives the user's login credentials from the Login use case.  2. The system verifies the credentials by comparing them to the user's account information in the database.  3. If the credentials are valid, the system returns a success message to the Login use case.  4. If the credentials are invalid, the system returns an error message to the Login use case. |
| **Extended flows:** | **Extends Register use case:**  1. If the user doesn't have an account, they can click the "Register" link.  2. The system redirects the user to the registration page.  3. The user fills out the registration form and submits it.  4. The system creates a new account and redirects the user to the login page.  **Extends Forgot Password use case:**  1. If the user forgets their password, they can click the "Forgot Password" link.  2. The system prompts the user to enter their email address or username.  3. The user enters their email address or username.  4. The system sends a password reset link to the user's email address.  5. The user clicks the password reset link and follows the instructions to reset their password.  6. The system prompts the user to enter a new password.  7. The user enters a new password.  8. The system updates the user's password and redirects them to the login page. |

**Logout**

|  |  |
| --- | --- |
| **Identifier and name:** | Logout |
| **Initiator:** | User |
| **Goal:** | The user wants to safely end their session and prevent unauthorized access to their account. |
| **Description**: | The user wants to end their current session on the application. To do so, they must log out of their account. The system will then terminate the session and return the user to the login page. |
| **Preconditions**: | - The user is currently logged in to their account. |
| **Postconditions:** | - The user is no longer logged in to their account and cannot access actions that require authentication until they log in again. |
| **Basic flow (**Main success scenario**):** | 1. The user clicks the "Logout" button.  2. The system terminates the user's session and logs them out of their account.  3. The system returns the user to the login page or displays a message confirming that they have been logged out. |
| **Alternate flows:** | - If the user is inactive for a certain period, the system may automatically log them out to prevent unauthorized access.  - If the user is logged in to multiple devices or locations, they may need to log out of each one separately to fully end their session. |

**Update Account**

|  |  |
| --- | --- |
| **Identifier and name:** | Update Account |
| **Initiator:** | User |
| **Goal:** | The user wants to modify their account information to keep it up-to-date and accurate. |
| **Description**: | The user wants to update their account information on an application. This may include changing their name, email address, profile picture, or other details. The system will then update the user's account information with the new data. |
| **Preconditions**: | - The user is currently logged in to their account.  - The user has permission to update their account information. |
| **Postconditions:** | -The user's account information is updated with the new data.  -The user can log in to the system using their updated account information. |
| **Basic flow (**Main success scenario**):** | 1. The user navigates to the "Account Settings" page.  2. The system presents a form for the user to update their account information.  3. The user makes the desired changes to their account information.  4. The system verifies that the changes are valid.  5. If the verification is successful, the system updates the user's account information and displays a message confirming that the changes have been saved.  6. If the verification is unsuccessful, the system displays an error message and prompts the user to retry. |
| **Alternate flows:** | - If the user wants to delete their account, they may need to use the Delete Account use case instead. |

**View Account Details**

|  |  |
| --- | --- |
| **Identifier and name:** | View Account Details |
| **Initiator:** | User |
| **Goal:** | The user wants to view their account information and details. |
| **Description**: | The user wants to view their account details on the application. This may include details such as their name, email address, profile picture, or other information. The system will then display the user's account details to the user and any other users who have permission to view them. |
| **Preconditions**: | - The user is currently logged in to their account.  - The user has permission to view their own account details.  - Other users have permission to view the user's account details. |
| **Postconditions:** | - The user can view and/or update their account details as needed.  - Other users who have permission to view the user's account details can see the information as well.  - Any changes made to the user's account details are reflected in the system and any features or actions that rely on the updated information. |
| **Basic flow (**Main success scenario**):** | 1. The user navigates to the "Account Details" page.  2. The system displays the user's account details, including their name, email address, profile picture, or other information.  3. The user reviews their account details and/or updates them using the Update Account use case. |
| **Alternate flows:** | - If the user doesn’t have permission to view their own account details, the system may display an error message or prompt the user to request access from an admin or other authorized user. |

**Change Password**

|  |  |
| --- | --- |
| **Identifier and name:** | Change Password |
| **Initiator:** | User |
| **Goal:** | The user wants to update their password to ensure the security of their account. |
| **Description**: | The user wants to change their password for their account on the application. To do so, they must provide their current password and choose a new password. The system will then update their account information with the new password. |
| **Preconditions**: | - The user is currently logged in to their account.  - The user knows their current password. |
| **Postconditions:** | - The user's account information is updated with the new password.  - The user can log in to their account using the new password.  - The user's old password is no longer valid and cannot be used to access their account. |
| **Basic flow (**Main success scenario**):** | 1. The user navigates to the "Change Password" page.  2. The system presents a form for the user to enter their current password and choose a new password.  3. The user enters their current password and chooses a new password.  4. The system verifies that the current password is correct and that the new password meets the system's password requirements (such as length, complexity, etc.).  5. If the verification is successful, the system updates the user's account information with the new password and displays a message confirming that the password has been changed.  6. If the verification is unsuccessful, the system displays an error message and prompts the user to retry. |
| **Alternate flows:** | - If the user forgets their current password, they may need to use the Forgot Password use case to recover it before they can change their password. |

**Manage Customers Database**

|  |  |
| --- | --- |
| **Identifier and name:** | Manage Customers Database |
| **Initiator:** | Admin |
| **Goal:** | The admin wants to manage the customers database, including editing and deleting customer records |
| **Description**: | The admin wants to manage the customers database on the application. This may be necessary to edit or delete customer information, or to perform other administrative tasks related to customer data. |
| **Preconditions**: | - The admin has the necessary permissions and access to manage the customers database.  - The application has a customer’s database available for management. |
| **Postconditions:** | - The customers database is updated based on the admin's actions.  - Any actions or features that rely on the updated customer information will reflect the updated database.  - The admin can continue to manage the customers database as needed. |
| **Basic flow (**Main success scenario**):** | 1. The admin navigates to the "Customers Database" page.  2. The system presents a list or menu of available tasks for managing the customers database, such as editing an existing customer, or deleting a customer.  3. The admin selects the desired task from the list or menu.  4. The system presents a form for entering or editing the customer information or prompts the admin to confirm the deletion.  5. The admin enters or edits the required information or confirms the deletion.  6. The system updates the customers database based on the admin's actions and displays a message confirming the successful completion of the task. |

**Manage Books**

|  |  |
| --- | --- |
| **Identifier and name:** | Manage Books |
| **Initiator:** | Admin |
| **Goal:** | The admin wants to manage books in the product catalog. |
| **Description**: | The admin can view, add, edit, and delete books records in the book category. The system validates the books information and updates the book category accordingly. |
| **Preconditions**: | - The admin has access to the book management functionality. |
| **Postconditions:** | - The book category is up-to-date and accurate. |
| **Basic flow (**Main success scenario**):** | 1. The admin navigates to the "book Management" page.  2. The admin can view a list of all books in the category, including their basic information such as name, description, price.  3. The admin can add new products to the category by entering their information manually or importing a list of books from a file.  4. The admin can edit existing book records by updating their information, such as name, description, price, or quantity.  5. The admin can delete book records from the category.  6. The user can manage book images, including uploading, editing.  7. The system validates the book information and updates the category accordingly. |

**Choose Category**

|  |  |
| --- | --- |
| **Identifier and name:** | Choose Category |
| **Initiator:** | Customer |
| **Goal:** | The customer wants to select a category for a book. |
| **Description**: | The user wants to select a category on the website. This may be necessary for various reasons, such as browsing book. The system will then display the relevant information or content based on the selected category. |
| **Preconditions**: | - The customer is currently on a page or interface that requires a category selection.  - The customer has permission to view or access the selected category. |
| **Postconditions:** | - The customer can view the relevant information or content based on the selected category.  - Any actions or features that rely on the selected category will reflect the chosen category. |
| **Basic flow (**Main success scenario**):** | 1. The customer navigates to the page or interface where the category selection is required.  2. The customer selects the desired category from the list or menu.  3. The system displays the relevant information or content based on the selected category. |
| **Alternate flows:** | - If the user selects an invalid or inaccessible category, the system may not display any thing |

**Display invoice**

|  |  |
| --- | --- |
| **Identifier and name:** | Display invoice |
| **Initiator:** | Customer |
| **Goal:** | The customer wants to view and manage invoice in the website. |
| **Description**: | The customer wants to view their invoice on the website. This may include details such as the amount owed, due date, payment history, or other information. The system will then display the user's invoice and any relevant information associated with it. |
| **Preconditions**: | - The Customer has an account on the website.  - The Customer has a bill associated with their account.  - The Customer has permission to view their invoice. |
| **Postconditions:** | - The Customer can view their invoice and any relevant information associated with it.  - Any actions or features that rely on the invoice information will reflect the details displayed.  - The Customer can continue to view these invoice information as needed. |
| **Basic flow (**Main success scenario**):** | 1. The Customer navigates to the "invoice" page or interface.  2. The system presents a list or menu of available invoice associated with the customer's account.  3. The system displays the invoice and any relevant information associated with it, such as the amount owed, due date, payment history, or other details. |
| **Alternate flows:** | - If the user has completed payment on a invoice, the system may display a message confirming that the bill has been paid and any associated payment history. |

**Make Payment**

|  |  |
| --- | --- |
| **Identifier and name:** | Make Payment |
| **Initiator:** | Customer |
| **Goal:** | The Customer wants to make a payment for an order. |
| **Description**: | The Customer wants to make a payment on the website. This may be necessary if the Customer owes a balance, needs to pay a invoice, or wants to purchase books or services. The system will then guide the Customer through the process of making a payment. |
| **Preconditions**: | - The Customer has a valid payment method on file with the website.  - The Customer has identified the amount to be paid and the reason for the payment. |
| **Postconditions:** | - The customer's payment is processed and applied to the associated invoice, or order.  - Any actions or features that rely on the completed payment will reflect the updated payment status.  - The Customer can continue to use the website with the payment successfully processed. |
| **Basic flow (**Main success scenario**):** | 1. The Customer navigates to the "Payment" page or interface.  2. The system presents a form or interface for entering payment details, including the payment amount and any relevant information such as a invoice address or payment method.  3. The Customer enters the required payment details and confirms the payment.  4. The system processes the payment and displays a message confirming that the payment is complete. |
| **Alternate flows:** | - If the Customer needs to modify their payment method or information, they may need to use additional forms or interfaces to update their payment information before making the payment.  - If the payment is for a specific book or service, the system may need to verify that the payment matches the associated order or invoice before processing the payment. |

**Place Order**

|  |  |
| --- | --- |
| **Identifier and name:** | Place Order |
| **Initiator:** | Customer |
| **Goal:** | The customer wants to place an order for the selected books and complete the purchase process. |
| **Description**: | The customer wants to place an order and complete the purchase process on the website. The customer selects the book they want to purchase, enters the shipping address, invoice information, and payment details, and confirms the order and payment details. The website processes the payment, updates the order status, and sends a confirmation email to the customer with the order details and confirmation number. |
| **Preconditions**: | - The customer has selected the books they want to purchase.  - The customer has access to the shopping cart or checkout functionality.  - The customer is logged in to the website. |
| **Postconditions:** | - The customer has placed an order and completed the purchase process.  - The customer receives a confirmation email with the order details.  - The app updates the inventory levels and shipping status of the ordered books. |
| **Basic flow (**Main success scenario**):** | 1. The customer navigates to the shopping cart or checkout page.  2. The website displays the selected books, along with their quantities and prices.  3. The customer reviews the books details, quantities, and prices, and can update the cart contents as needed.  4. The customer enters the shipping address, invoice information, and payment details.  5. The customer confirms the order and payment details.  6. The app processes the payment and updates the order status.  7. The app sends a confirmation email to the customer with the order details.  8. The website updates the inventory levels and shipping status of the ordered books. |
| **Alternate flows:** | Invalid Payment Information  1. The customer enters the shipping address, invoice information, and payment details.  2. The app validates the payment information and detects an error or issue, such as an expired or invalid credit card, insufficient funds, or a declined transaction.  3. The website displays an error message to the customer, informing them of the issue and prompting them to correct the payment information.  4. The customer updates the payment information and confirms the order and payment details.  5. The website processes the payment and updates the order status.  6. The website sends a confirmation email to the user with the order details. |

**Proceed to Checkout**

|  |  |
| --- | --- |
| **Identifier and name:** | Proceed to Checkout |
| **Initiator:** | Customer |
| **Goal:** | The Customer wants to finalize the order and proceed to the checkout process in the website. |
| **Description**: | The Customer wants to proceed to check out on the website. This may be necessary if the user wants to purchase books or services. The system will then guide the Customer through the checkout process to complete the transaction. |
| **Preconditions**: | - The Customer has added books or services to their cart.  - The Customer is currently on the "Cart" page or interface.  - The Customer has permission to proceed to checkout. |
| **Postconditions:** | - The customer's transaction is complete and any purchased books or services are processed according to the system's policies.  - The Customer can view their order history or receipt for the transaction.  - Any actions or features that rely on the completed transaction will reflect the purchased booksor services. |
| **Basic flow (**Main success scenario**):** | 1. The Customer navigates to the "Checkout" page or interface.  2. The system presents a summary of the customer's cart contents and total cost.  3. The Customer enters their shipping information, invoice information, and payment details as required.  4. The system verifies that the information is valid and meets any applicable requirements or restrictions.  5. If the verification is successful, the system processes the payment and displays a message confirming that the transaction is complete.  6. If the verification is unsuccessful, the system displays an error message and prompts the user to retry. |
| **Alternate flows:** | - If the user wants to modify their cart contents, they may need to use the View Cart use case before proceeding to checkout. |

**Search Book**

|  |  |
| --- | --- |
| **Identifier and name:** | Search book |
| **Initiator:** | Customer |
| **Goal:** | The customer wants to search for a book in the website. |
| **Description**: | The Search website use case involves users searching for specific book or items on the website. The customer enters a search query into a search bar or field, and the platform returns a list of relevant books that match the user's query. |
| **Preconditions**: | - User is looking for a specific book.  - The website has a search feature. |
| **Postconditions:** | - Customer has found the desired book.  - The website has provided accurate search results to the user. |
| **Basic flow (**Main success scenario**):** | 1. Customer navigates to the website search bar or field.  2. Customer enters a search query for the desired books.  3. The website search algorithm processes the query and returns a list of relevant books  4. Customer reviews the list of books and selects the desired item.  5. Customer proceeds to view the selected book's details and may choose to purchase it |
| **Alternate flows:** | - If the Customer 's search query returns no results, the website may suggest alternative search terms or provide other bboks recommendations  - If the Customer finds the search results to be inaccurate or irrelevant, they may refine their search query or adjust their search filters to narrow down the results |

**Offers**

|  |  |
| --- | --- |
| **Identifier and name:** | Offers |
| **Initiator:** | Customer |
| **Goal:** | The Customer wants to view the current offers and promotions in the website. |
| **Description**: | The Customer wants to view offers on the application. This may include discounts is only available for a limited time. The system will then display the available offers and any relevant information associated with them. |
| **Preconditions**: | - The Customer is currently using the website.  - The website has today's offers available. |
| **Postconditions:** | - The Customer can offers and any relevant information associated with them.  - Any actions or features that rely on the offer details will reflect the displayed information.  - The Customer can continue to view or take advantage of the available offers until they expire or are no longer available. |
| **Basic flow (**Main success scenario**):** | 1. The Customer navigates to the "Offers" page or interface.  2. The system presents a list or menu of available offers for the day.  3. The Customer selects the desired offer from the list or menu.  4. The system displays the offer and any relevant information associated with it, such as the discount amount. |
| **Alternate flows:** | - If the offer is time-sensitive, the system may display a countdown or other indicator of the remaining time to take advantage of the offer. |

**Rate**

|  |  |
| --- | --- |
| **Identifier and name:** | Rate |
| **Initiator:** | Customer |
| **Goal:** | The customer wants to provide rate or suggestions to the website. |
| **Description**: | The customer wants to provide rate on the website. This may include suggestions for improvement, bug reports, or other comments. The system will then use this rate to improve the customer experience or address any issues. |
| **Preconditions**: | - The customer is currently using the application.  - The customer has permission to submit rate. |
| **Postconditions:** | - The customer's rate is saved and can be used by the system to improve the customer experience or address any issues.  - Any actions or features that rely on the user rate will reflect the submitted rate. |
| **Basic flow (**Main success scenario**):** | 1. The customer navigates to the "rate" page or interface.  2. The system presents a form or interface for the customer to enter their rate.  3. The customer enters their rates in the provided text field or interface.  4. The system verifies that the rate is valid and meets any applicable requirements or restrictions.  5. If the verification is successful, the system saves the rate and displays a message confirming that the rate has been submitted. |
| **Alternate flows:** | - If the user wants to provide rate on a specific feature or aspect of the website, they may need to use a more specific rate form or interface. |

**View Cart**

|  |  |
| --- | --- |
| **Identifier and name:** | View Cart |
| **Initiator:** | Customer |
| **Goal:** | The Customer wants to view the items they have added to their shopping cart and manage their cart contents. |
| **Description**: | The View Cart use case involves users viewing the items they have added to their shopping cart on website. The Customer navigates to the cart and can view the books they have added to their cart, along with their quantities and prices. The Customer may also have the option to edit or remove items from their cart. |
| **Preconditions**: | - Customer has added items to their shopping cart.  - The website has a cart or checkout page |
| **Postconditions:** | - Customer has reviewed the items in their shopping cart  - Customer has the option to edit or remove items from their cart  - Customer may proceed to the checkout page to complete their purchase |
| **Basic flow (**Main success scenario**):** | 1. Customer navigates to the website's cart or checkout page  2. The application displays the items the Customer has added to their cart, along with their quantities and prices.  3. Customer reviews the items in their cart and may choose to edit or remove items as needed  4. Customer proceeds to the checkout page to complete their purchase |
| **Alternate flows:** | - If the customer encounters any issues or errors while viewing their cart, the platform may provide error messages. |

**Add book to cart**

|  |  |
| --- | --- |
| **Identifier and name:** | Add book to cart |
| **Initiator:** | Customer |
| **Goal:** | The Customer wants to add books to their cart for future purchase on thewebsite. |
| **Description**: | The Customer wants to add books to their cart. The user selects the desired book and adds it to their cart by entering the quantity and clicking the "Add to Cart" button. The platform adds the selected book to the customer's cart and displays a confirmation message. The customer can then view the cart contents and proceed to checkout when ready to purchase. |
| **Preconditions**: | - The Customer is logged in to the website.  - The Customer has accessed the book page on the website. |
| **Postconditions:** | - The Customer has added the book to their cart for future purchase.  - The website updates the cart contents and displays the updated cart summary. |
| **Basic flow (**Main success scenario**):** | 1. The Customer selects the book they want to add to their cart.  2. The website displays the book details, including the price and quantity available.  3. The Customer enters the desired quantity of the book.  4. The Customer clicks the "Add to Cart" button.  5. The website adds the selected book to the user's cart and displays a confirmation message. |
| **Alternate flows:** | - If the Customer wants to remove a product from their cart, they can select the "Remove from Cart" button from View Cart use case. |

5-Activity Diagram

صورة تحتوي على نص, رسم بياني, موازِ, رسم تقني

تم إنشاء الوصف تلقائياً

صورة تحتوي على رسم بياني, نص, رسم تقني, خطة

تم إنشاء الوصف تلقائياً

صورة تحتوي على رسم بياني, رسم تقني, خط, رسم

تم إنشاء الوصف تلقائياً

صورة تحتوي على رسم بياني, رسم تقني, خطة, رسم

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, رسم تقني, لقطة شاشة

تم إنشاء الوصف تلقائياً

صورة تحتوي على رسم بياني, نص, رسم, رسم تقني

تم إنشاء الوصف تلقائياً

6. Sequence diagram

صورة تحتوي على نص, رسم بياني, خطة, رسم تقني

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, موازِ, خط

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, لقطة شاشة, موازِ

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, لقطة شاشة, موازِ

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, موازِ, خط

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, موازِ, رسم تقني

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, موازِ, لقطة شاشة

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, موازِ, لقطة شاشة

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, لقطة شاشة, موازِ

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, موازِ, خط

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, لقطة شاشة, موازِ

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, موازِ, خطة

تم إنشاء الوصف تلقائياً

صورة تحتوي على نص, رسم بياني, موازِ, خط

تم إنشاء الوصف تلقائياً

7- Class Diagram:

صورة تحتوي على رسم بياني, رسم, خطة, رسم تقني

تم إنشاء الوصف تلقائياً

9- Database Specification:

ERD :

صورة تحتوي على رسم, رسم بياني

تم إنشاء الوصف تلقائياً

10- OCL

