Delivery Notes:

- This is a group assignment of 4 members (at most). If you submit as a group of more than 4 members, ALL the group members will get 0 in the assignment.
- ALL team members students should work and fully understand everything in the submitted solution.
- **Team members could be from the same lab group, or from different lab groups.**
- **If you have any questions about the assignment, you need to submit your questions through this form link**
  https://docs.google.com/forms/d/e/1FAIpQLSfKYVn41UZRFBJNZxqfBBHG3QNvisei09LeAGCRyesLAiFJAg/viewform
- No late submission is allowed.
- Submission will be done through a Google form link. Such link will be posted on the Google classroom before the deadline. Once you submit through the form, you will receive a copy of your submission by email. It is your responsibility to make sure that your submission went through properly. If you found a problem in your submission, you can still edit it before the deadline.
- If your submission was not uploaded properly while marking, you will not receive a grade for the assignment.
- No submission through e-mails.
- You will create TWO Word files that using both templates (SRS) and (SAS), and you, and should include the full names/IDs of the team members as well. You will save both files as a PDF files. You will put those files **(two Word files and two PDF files)** in a folder named CS251_Assign1_firstStudentID_SecondStudentID_ThirdStudentID_FourthStudentID and compress them to a .zip file with the same folder name. The compressed file would be the file to be submitted.
- **Failing to abide by the naming conventions of the file or failing to submit the files as per the requested extension, would result in a ZERO for both team members.**

- **In case of cheating, you will get a negative grade whether you give your solution to someone, take the solution from someone/internet, or even send it to someone for any reason.**

Assignment 1

This assignment consists of two parts as follows.

Part 1 [Requirements Elicitation]

Consider the following problem statement:

A university member uses university library web site to borrow library books online within some library book borrowing system. The University member is considered an uncertified university member until he certifies. A University member can **browse books**, **borrow books, return books,** and **certify** as a **certified university member**. The University member could use browse books if the university member only wants to find and see the currently available books within the library to borrow. This functionality could also be used as a part of the borrow book function. When a university member certifies as a certified university member, he can get extended borrowing periods based on belonging to specific institutes offered by the Web site borrowing period extension service (e.g., if the certified university member is a member of some syndicates-organizations, he could be offered extended borrowing periods). Note that **the checkout** function is not available by itself - checkout is part of the borrow book.

The university member can **explore**, **search**, **borrow library books**, **join waiting list**, and **browse recommended books** during browsing books. Borrowing books, and joining the waiting list demand **authentication** as a certified university member. User authentication could be done through the library book borrowing system´s login page or through the user´s google based university account. Using university accounts requires external identity provider participation.

To checkout, the university member needs to be certified, and a payment function should be involved. The payment could be done either through some credit card payment service, or using certified university member reward points. University member earn reward points from reward points party through various actions such as borrowing books, participating in library events, or contributing to the library community in other ways.

Part 1 deliverable: Write an Software Requirements Document (SRS), for the above problem statement, using the attached template. Such SRS document would contain the following items:

1. A listing of the functional requirements of the system.
2. A listing of the non-functional requirements of the system. For the non-functional requirements, you need to mention at least (2x team size) non-functional requirements.
3. A Use case model describing the above system.
4. Create use case descriptions/table for at least (**1 x team size) use cases** for the **most complex** use cases, where each team member would be responsible **for submitting one use case description/tables**. *Note: The selected use cases should be the ones explained within Part 2 of the assignment. You should utilize the details within part 2 (below) when creating your use case descriptions/tables.*

Part 2 [Requirements Analysis]

For the same library management system explained within Part 1, consider the following additional details about some of the system's functionality:

- Each book shall be identified by a name, a target-age-range, a publisher, a publishing year, a number of copies, borrowing fees, and a book ID.
- Each book should have a number of copies, where each copy should have a copy ID and should be linked to its corresponding book.
- Borrow book(s) function that reflects borrowing a specific amount of book copies, within a specified allowed limit for borrowed books. Such limit is specified in advance through the system. Furthermore, for children less than 12 years old, books target-age-range apply, and can lead to rejecting borrowing specific books. Also, accumulated fines above a specified system-defined amount would prohibit borrowing books.
- Checkout function that marks calculating the fees for books selected to be borrowed at a specific time.
- During the checkout, calculate the fees based on the number of borrowed books, and overdue fines less than a specific amount.
- Calculate the total income of the library management system, as well as the total number of borrowed books across the system at any point in time.
- Return book(s) function that reflects returning previously borrowed books, as well as calculating the fines for exceeding the borrowing duration. The fines should be incremental (i.e., they increase with the increase in exceeding the borrowing duration past the given deadline. This duration should be system-defined).
- Display the number of available books, as well as the borrowed books along with their expected return dates.

Part 2 deliverable: Write an Software Analysis Document(SAS), for the above problem statement, using the attached template. Such SAS document would contain the following items:

1. A class diagram for the description given within Part 1 and Part 2.
2. Sequence diagrams for the use cases given within Part 2 only. Create sequence diagrams for the **most complex** scenarios. The submitted sequence diagrams should be **1 x team size**, where each team member would be responsible **for submitting one sequence diagram**. *The covered sequence diagrams should be for the same use cases that had use case descriptions within Part 1*.

Grading Criteria

- Correctness and coverage of the submitted items for the features of the system of interest.
- Correctness and variation of the used notations for the submitted items.
- Consistency among the different submitted items.

Due date and submission

- Assignment 1 is due on Wednesday, March 27th at 11:55 PM (Cairo Local time).
- Submission needs to be done through the course's Google classroom form only.