IT314 - Software Engineering

Lab : Domain Analysis Modeling & Sequence Diagram

Group 3:

Chirag Gupta-201801188
Arkaprabha Banerjee-201801408
Rahil Shah-201801252
Kartavi Shah-201801426
Archit Agrawal-201801043
Amruthsai Jilla -201801069
Meet Patel-201801415
Bhagyesh Ganatra-201801047
Ridham Suvagiya-201801006
Udit Meena -201801095

Project Name: Automatic Answer Checker (Yosemite)

Textual Description of “Issue Book”,
Actors: Librarian, Student,

1.) Student wants to Issue book:
    - Login to the system
    - wil search for a book
    - Visit Librarian
2.) Librarian will do the below process of book issue
    - Issue book and mark the due date
    - It includes that librarian needs to check whether student is valid or not
    - It includes that librarian needs to check whether student is not exceeding the limit of number of book that he can issue.

- Signup
- Login
- Browse Book
- GIve feedback
- Update Book details
- Issue book
- Availability of required books
- Verify details
- Check book limits
- Return Item
- Check Applicable file
- Verify return details

Use case	Signup
Actor	Student
Pre-condition	Students must have an internet connection.
Post-condition	Students will be registered in the database.
Steps	1.Student opens the app and fills the signup form. 2.System sends a request to the server to store the details of the student.
Alternate flow	1.Server failed: student needs to sign up again. 2.Lost internet connection: needs to sign up again
Relate Use cases	1. Validation error <<extend>>

Use case name	Login to the system
Actors	1.Student 2.Librarian
Pre-condition	Student and Librarian are already registered in the database
Post-condition	App goes into logged state
Steps	1. Student and Librarian need to login with credentials 2. After the login they will see the homepage
Alternate flow	1. Wrong credentials: app shows error and needs to login again 2. User is not pre-registered: need to ask admin to make his account
Related use cases	1. Validation error <<extend>> 2. Issue book <<include>> 3. Return Item <<include>>

Use case	Browse Book
Actor	Student
Pre-condition	Students must have internet connection
Post-condition	Students will be able to see the book details
Steps	Students need to search book by its name and then able to see the details of the book
Alternate flow	1.Book does not exist: Shows an error
Relate Use cases	-

Use case	Give feedback
Actor	Student
Pre-condition	1. Students must be logged in.
Post-condition	Feedback will be stored in the database.
Steps	1.Student opens the app and logs in via credentials.

	2.Writes feedback and submit it.
Alternate flow	1.Wrong credentials: Shows an error 2.Server failed: Feedback won't get stored in the database and the student needs to write it again.
Relate Use cases	-

Use case	Update book details
Actor	Librarian
Pre-condition	Librarian must be logged in
Post-condition	Updated details of the book will get stored in the database.
Steps	1.Librarian opens the app and logs in. 2.Update the book details
Alternate flow	1.Wrong credentials: Shows an error 2.Server failed: Details won't get stored in the database, Librarian need to update it again
Relate Use cases	1. Validation error <<extend>>

Use case name	Issue book
Actors	1.Student 2.Librarian
Pre-condition	Student and Librarian are already logged in.
Post-condition	Students can issue books.
Steps	1. Students need to login via their credentials. 2. And then search for book 3. System checks whether book is available or not 4. If it is available then librarian issue this book to that student
Alternate flow	1. Wrong credentials: app shows error and needs to login again 2. Book does not exist: shows error 3. Book is not available: Librarian add student into waiting queue

Related use cases	1. Login <<include>> 2. Availability of the required items <<include>>

Use case	Availability of required books
Actor	Student
Pre-condition	Students need to login and issue a book.
Post-condition	Availability of books is shown and system verifies the student details
Steps	1.Students open the website and login into the system via credentials provided to them. 2.Students issue a book. 3.And then system checks the availability of the book
Alternate flow	1. Students enter the wrong credentials - System shows error and ask them to login again 2.Book is not available - Librarian add the student into the waiting queue.
Relate Use cases	1. Issue book <<include>> 2. Verify Details <<include>>

Use case	Verify Details
Actor	Student
Pre-condition	Students need to login into the system and issue a book and that book should be available.
Post-condition	Student details are verified and then the system checks the issue book limit.
Steps	1.Students need to login into the system via credentials provided to them. 2.Student issues a book and the system checks the availability of the book. 3.If a book is available then the system checks the student details. 4.If details provided by the students are correct then the system proceeds further and checks for the issue book limit.

Alternate flow	1.Student enters the wrong credentials - the system shows error and asks them to login again. 2.Student didn't issue a book - system asks him to issue a book 3.Book is not available - system adds the student to the waiting queue. 4. Students details are wrong - system shows error and returns to the home page.
Relate Use cases	1. Check book limit <<include>> 2. Availability of the required items <<include>>

Use case	Check book limit
Actor	Student
Pre-condition	1. Students need to login into the system. 2. Students need to issue a book. 3. Issued books should be available. 4. And the details of the students should be verified.
Post-condition	Students will be allotted the issued book and then the librarian adds the entry in the database.
Steps	1. Student opens the website and logs in via credentials provided to him. 2. Student issues the book. 3. System checks the availability of the book. 4. Then the system verifies the student details. 5. And then checks for the limit.
Alternate flow	1. Student enters wrong credentials - system shows an error and asks him to login again. 2. Student didn't issue a book - the system asks him to issue a book before proceeding further. 3. Book is not available - System adds the student into the waiting queue. 4. Students provided wrong details - system shows an error and redirect to the home page. 5. Student has exceeded the limit - shows an error message and redirects to the home page.
Relate Use cases	1. Verify details <<include>>

Use case	Return Item

Actor	Student Librarian
Pre-condition	1. Student and librarian should be logged in.
Post-condition	System checks for the fine that should be imposed on this student.
Steps	1. Student opens the website and logs in via credentials provided to him. 2. And then makes the return request for the item that he has issued before.
Alternate flow	1. Student enters the wrong credentials -the system shows an error and again asks him to login. 2. Student made a return item request that he hadn’t issued - the system shows an error and redirects to the home page.
Relate Use cases	1. Login <<include>> 2. Check applicable fine<<include>>

Use case	Check applicable fine
Actor	Student
Pre-condition	1. Students need to be logged in. 2. He should have made the return request.
Post-condition	System verifies the return details.
Steps	1. Student opens the website and logs in via credentials. 2. He makes the return requests. 3. Then the system checks for a fine.
Alternate flow	1. Wrong credentials - the system shows an error and then asks the student to log in. 2. Student made a return item request that he hadn’t issued - the system shows an error and redirects to the home page.
Relate Use cases	1. Return Item <<include>> 2. Verify return details <<include>>

Use case	Verify return details
Actor	Student
Pre-condition	1. Student should be logged in. 2. He should have made the return request. 3. All the applicable fines should be calculated.
Post-condition	System makes the return entry in the database.
Steps	1. Student opens a website and logs into the system via credentials provided to him. 2. He makes the return item request. 3. System checks for fine if there is any. 4. System verifies the return details.
Alternate flow	1. Wrong credentials - the system shows an error and asks the student to log in again. 2. Student made the return request for the item he hadn't issued - the system shows an error and redirects him to the home page. 3. Some return details are not correct - system shows an error and redirects him to the home page.
Relate Use cases	1. Check applicable fine <<include>>

Contribution :

1. Use case diagram
   Contributors ID : 201801415 , 201801408

2. Use case documentation
   Contributors ID : 201801069, 201801188

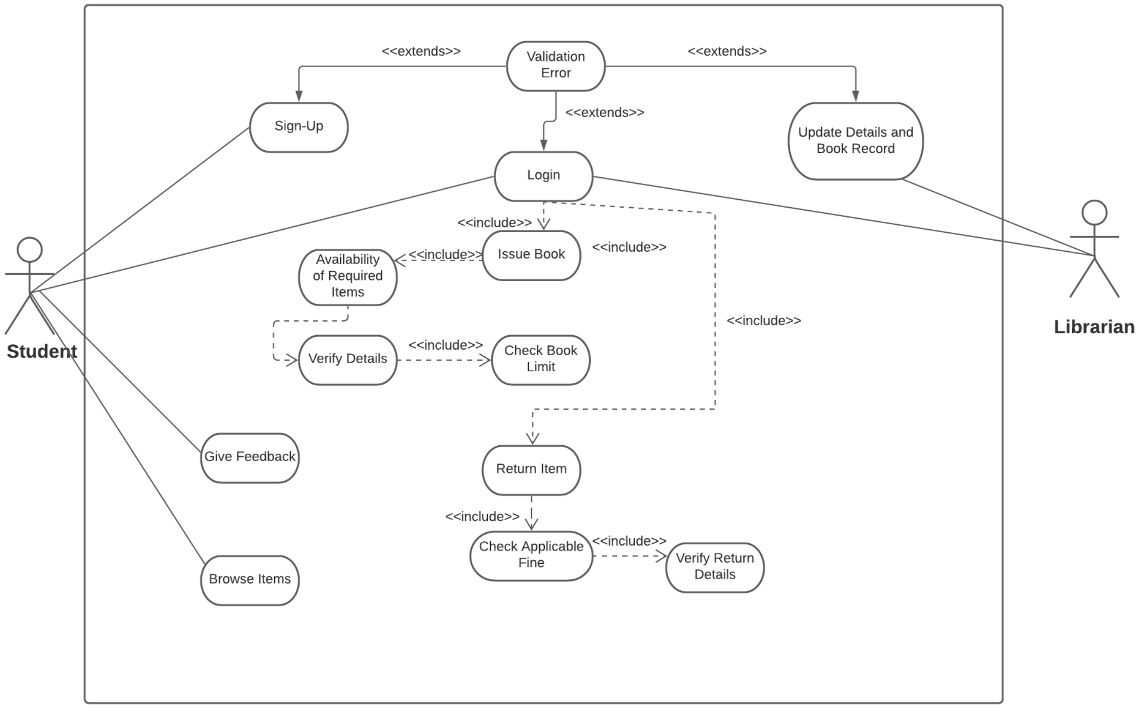
3. Sequence Diagram
   Contributors ID : 201801252, 201801047

4. Analysis Diagram
   Contributors ID : 201801426, 201801043

5. Q2 sequence diagram
   Contributors ID : 201801252, 201801415, 201801426

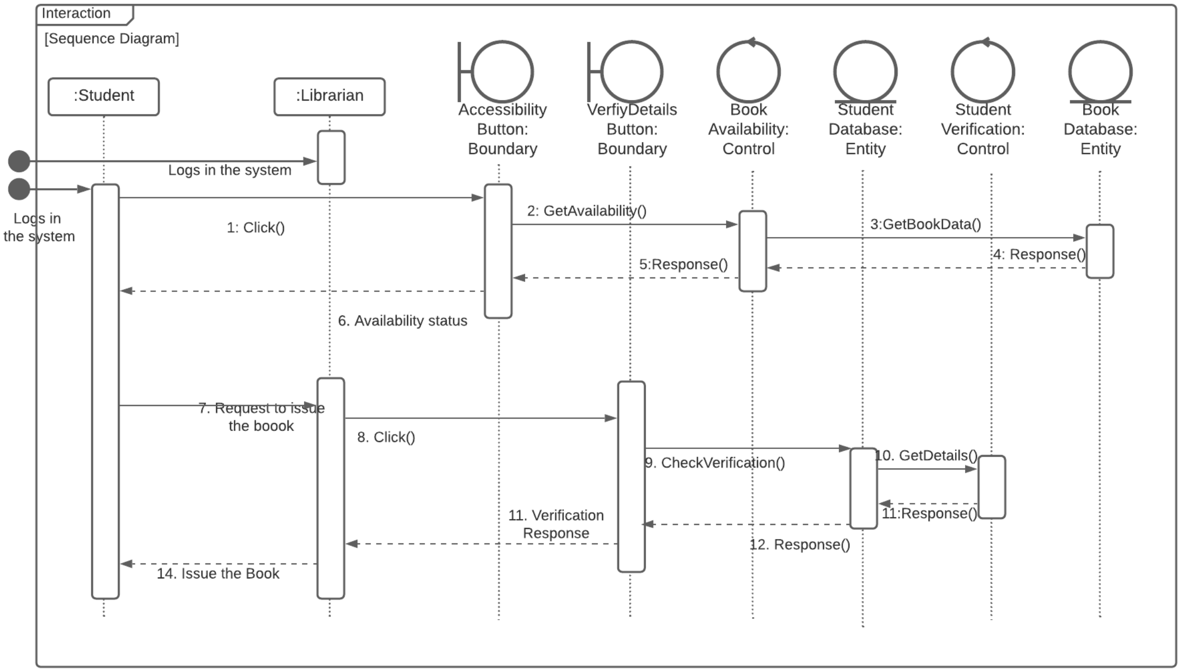
Q1)

Use case diagram:

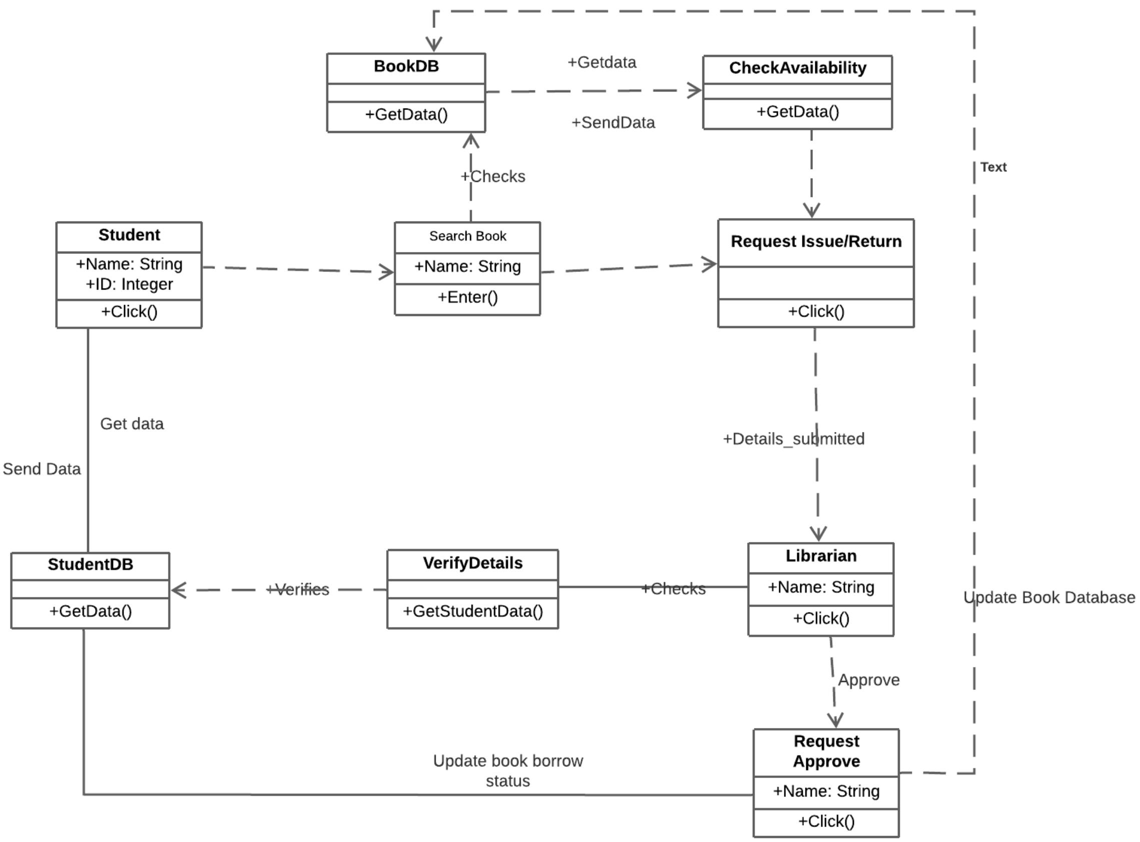


Sequence Diagram

2. The sequence diagram for the **IssueBook** Use Case.

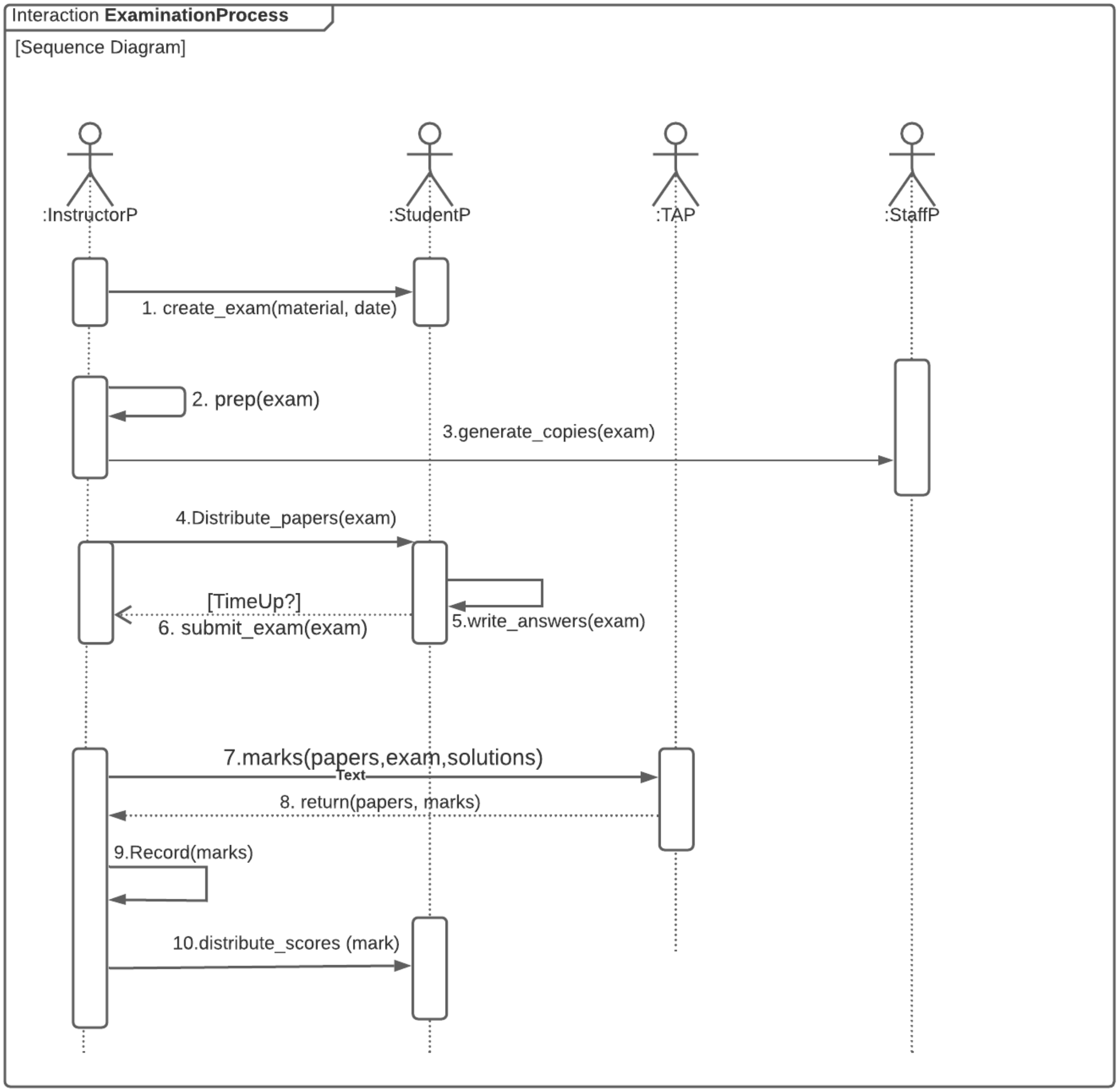


Analysis Diagram



Q2) Sequence Diagram

Question-2. The sequence diagram for the **ExaminationProcess**.


Interaction **ExaminationProcess**
[Sequence Diagram]
:InstructorP
:StudentP
:TAP
:StaffP
1. create_exam(material, date)
2. prep(exam)
3.generate_copies(exam)
4.Distribute_papers(exam)
5.write_answers(exam)
[TimeUp?]
6. submit_exam(exam)
7.marks(papers,exam,solutions)
Text
8. return(papers, marks)
9.Record(marks)
10.distribute_scores (mark)