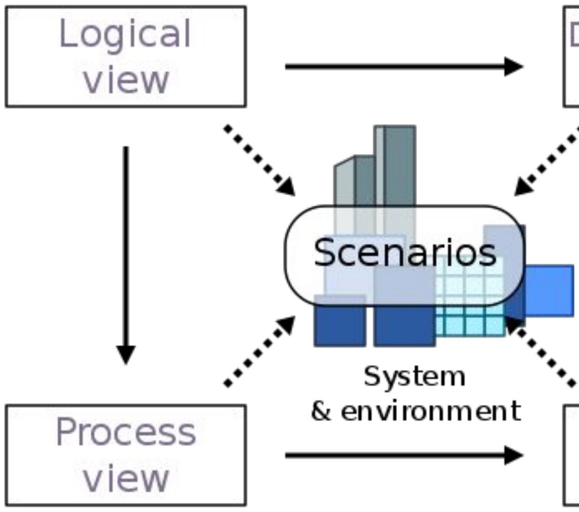
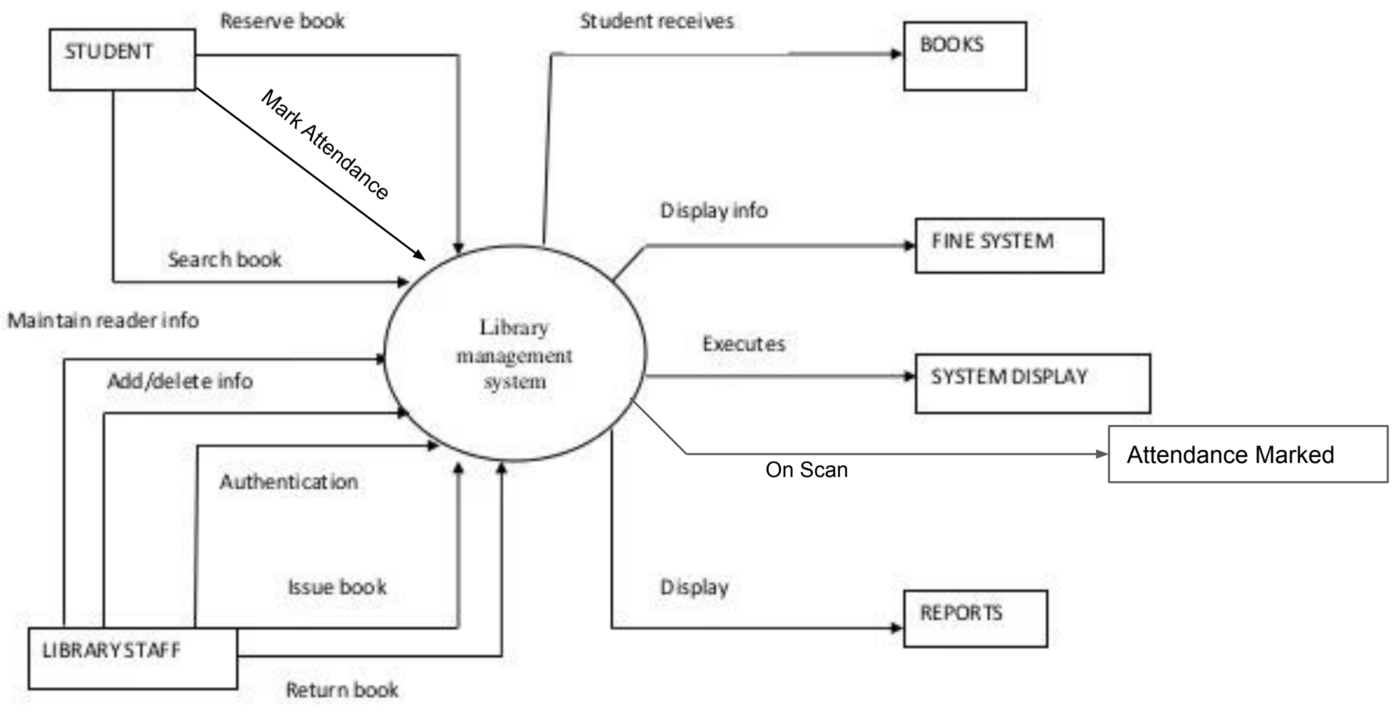
4+1 Views

- 4+1 is a view model used for describing the architecture of software-intensive systems, based on the use of multiple, concurrent views.
- It provides a clear picture of the architecture involved in the development of software systems using various multiple and concurrent views stated earlier.
- The four views of the model are logical, development, process and physical view.
- In addition, selected **use cases** or **scenarios** are used to illustrate the architecture serving as the **'plus one'** view.
- Hence, the model contains 4+1 views.

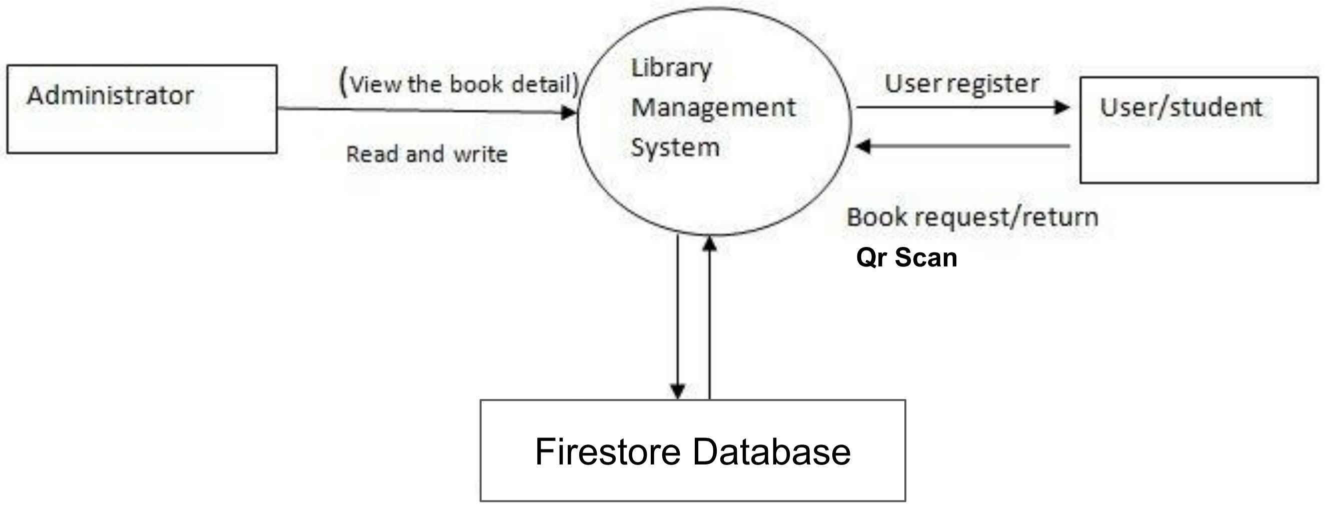


- **Logical View:**
    - The logical view is concerned with the functionality that the system provides to end-users.
    - End to end test cases are prepared keeping the logical view in mind.
    - UML diagrams are used to represent the logical view, and include class diagrams, and state diagrams.

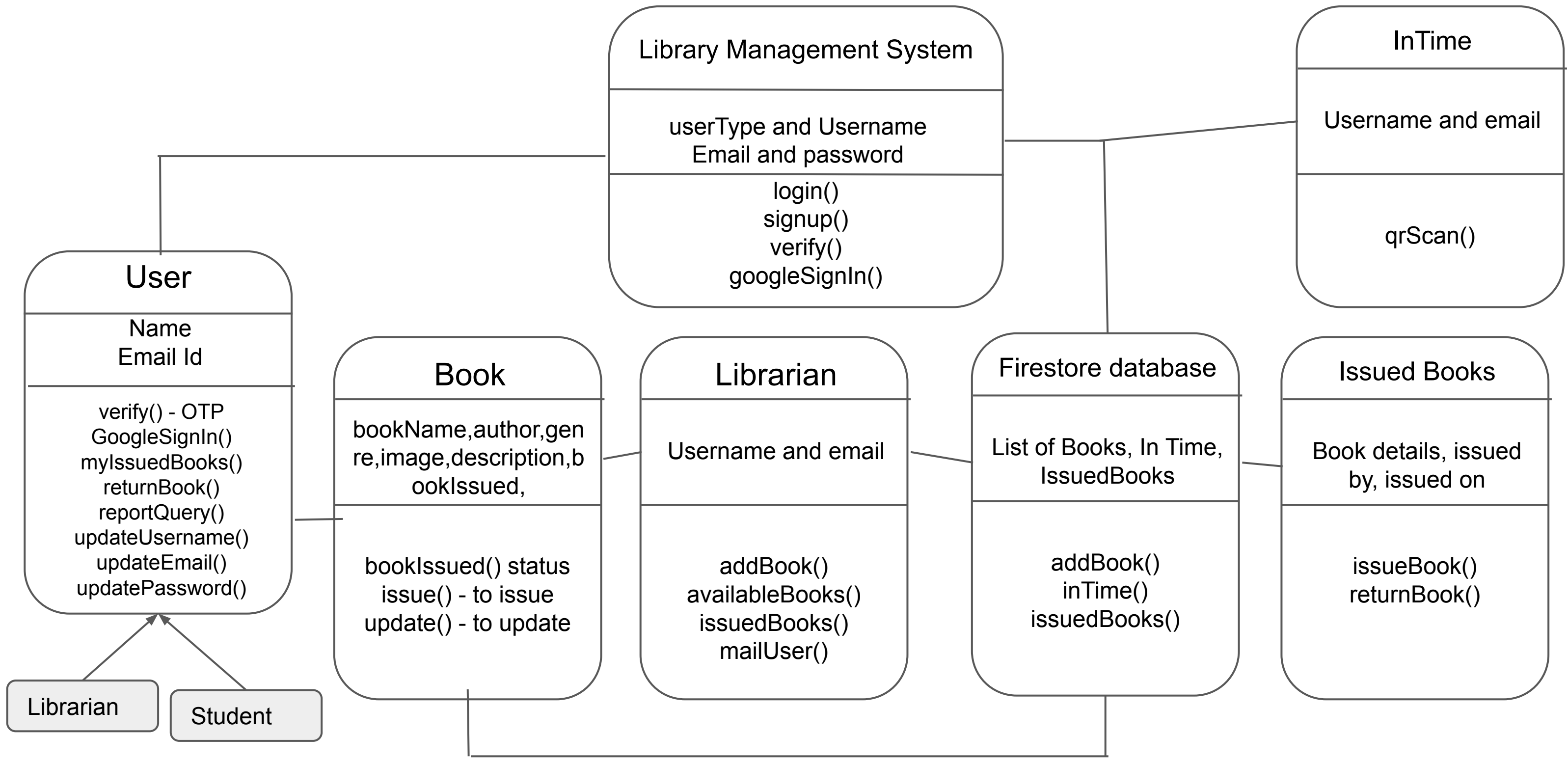


- The logical view provides us the clear information about the flow of the application for every kind of user whether it be student or library staff/admin.

- **Process View:**
    - The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the run time behavior of the system.
    - The process view addresses concurrency, distribution, integrator, performance, and scalability, etc.
    - UML diagrams to represent process view include the sequence diagram, communication diagram, activity diagram.
    - Process view depicts the essential aspects for the software to communicate smoothly and for a better performance.

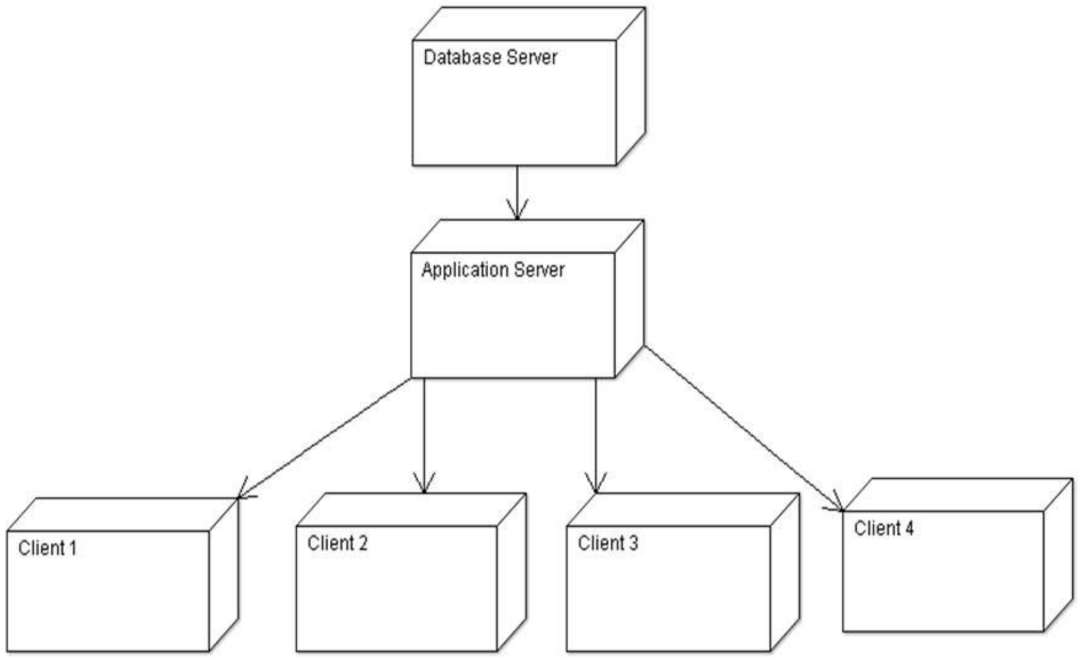


- **Development View:**
    - The development view illustrates a system from a programmer's perspective and is concerned with software management.
    - This view is also known as the implementation view.
    - UML Diagrams used to represent the development view include the Package diagram and the Component diagram.
    - Development view plays a vital role in understanding the software package from the programmer's perspective and helps to understand the flow as per the requirement of programmer.


Library Management System
userType and Username Email and password
login() signup() verify() googleSignIn()
InTime
Username and email
qrScan()
User
Name Email Id
verify() - OTP GoogleSignIn() myIssuedBooks() returnBook() reportQuery() updateUsername() updateEmail() updatePassword()
Book
bookName,author,genre,image,description,bookIssued,
bookIssued() status issue() - to issue update() - to update
Librarian
Username and email
addBook() availableBooks() issuedBooks() mailUser()
Firestore database
List of Books, In Time, IssuedBooks
addBook() inTime() issuedBooks()
Issued Books
Book details, issued by, issued on
issueBook() returnBook()
Librarian
Student

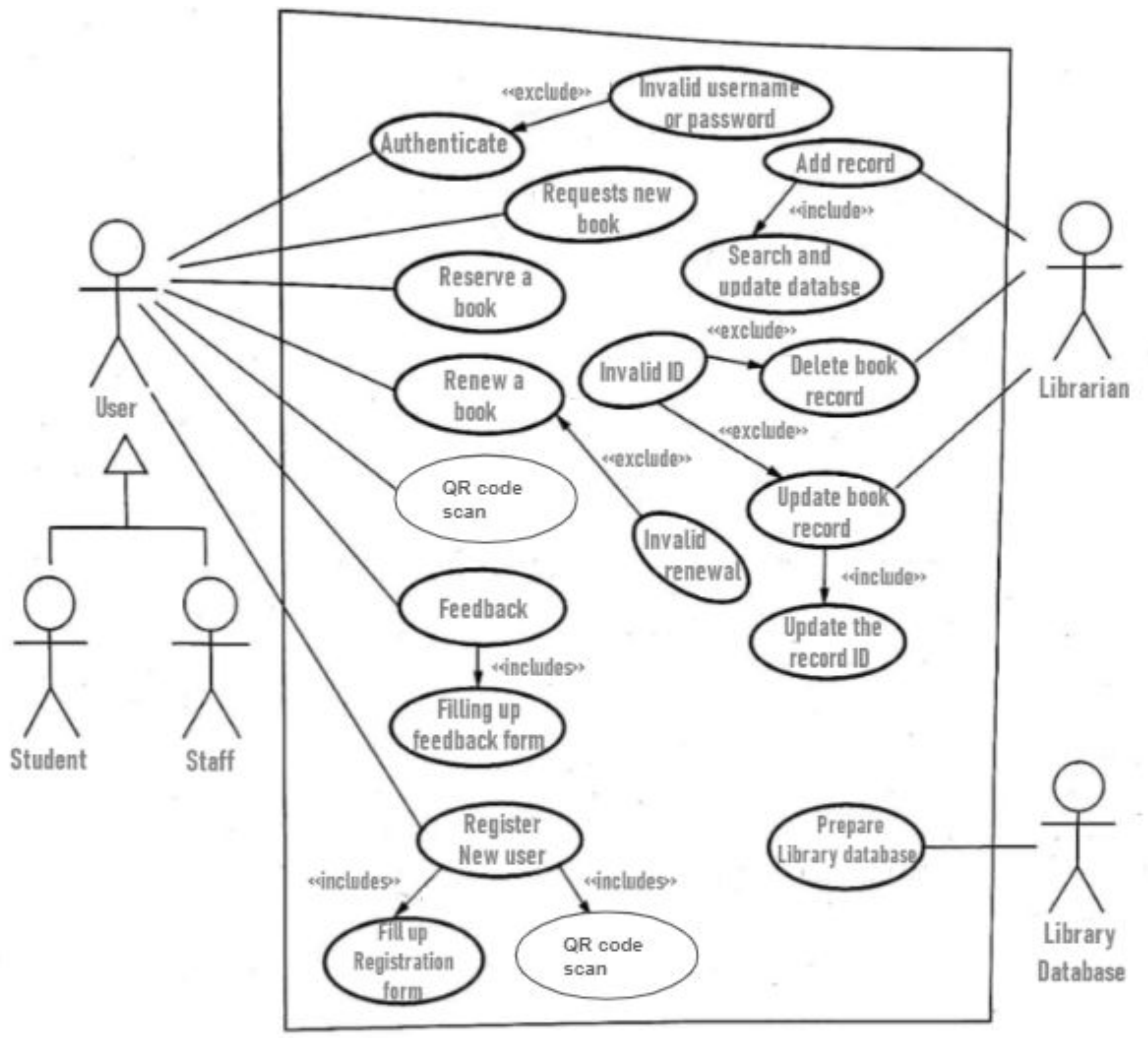
- The above given development view provides the view from the programmer's perspective.
- We can see that the firestore database is linked to every other model present in it as it is the super class of all the different models.
- Then we can see that librarian and students are linked to User model indicating user having to different sub classes i.e. librarian and student.
- Every model has some functions return at the end, which shows in which function this models are either read or write.
- For example in Books model we can see addBook() function which implies that addBook() function is used to write the model Books and availableBooks() function is used to read the books available in library to issue.

- **Deployment View:**
  - The deployment view depicts the system from a system engineer's point of view.
  - It is concerned with the topology of software components on the physical layer as well as the physical connections between these components. UML diagrams used to represent the physical v



  - The above given diagram depicts the workflow to be followed during the deployment time, wherein the database server is set and an application server is provided in user's mobile phone which gets splitted to several clients.

- **Use Case Scenarios:**
  - The description of an architecture is illustrated using a small set of use cases, or scenarios, which become a fifth view.
  - The scenarios describe sequences of interactions between objects and between processes.
  - They are used to identify architectural elements and to illustrate and validate the architecture design.
  - They also serve as a starting point for tests of an architecture prototype.
  - This view is also known as the **use case view**.
  - As you can see the diagram on next page tries to explain all the user case scenarios which can occur in our mobile application
  - User case scenarios are of much importance because it specifies the target audience which then affects the UI/UX and the simplicity or complexity of the application.


«exclude»
Invalid username or password
Authenticate
Add record
Requests new book
«include»
Search and update databse
Reserve a book
«exclude»
Invalid ID
Delete book record
Renew a book
«exclude»
«exclude»
QR code scan
Update book record
Invalid renewal
«include»
Feedback
Update the record ID
«includes»
Filling up feedback form
Register New user
Prepare Library database
«includes»
«includes»
Fill up Registration form
QR code scan
User
Student
Staff
Librarian
Library Database