**CHAPTER TWO**

2.1 High Level Sequence Diagram

A **high-level sequence diagram** is a UML diagram that provides a broad, abstract view of interactions within a system. Unlike detailed sequence diagrams focusing on specific objects and method calls, it emphasizes the communication flow between major actors and significant subsystems or components. Its primary goal is to illustrate the overall sequence of key actions or data exchanges for a particular use case or scenario, without delving into the intricate implementation details within each component.

These diagrams typically depict actors (external entities like users or other systems) interacting with the main functional blocks of the system. The messages exchanged between these high-level participants represent significant steps or data transfers necessary to achieve the use case's objective. The temporal ordering of these messages is still crucial, showing the progression of the interaction from initiation to completion.

High-level sequence diagrams are valuable for early-stage system design and communication with stakeholders. They offer a simplified understanding of system behavior, facilitating discussions and ensuring everyone has a common understanding of the major interactions. By abstracting away complexities, they help in identifying the key responsibilities of different subsystems and the overall flow of control and information within the system. They serve as a road map for more detailed design and development efforts, providing a context for the finer-grained interactions that will be elaborated in lower-level sequence diagrams. These diagrams are often used to document the core workflows and user journeys within the system.

2.2 Components Of High Level Sequence Diagram

Here are the **components** We have used in this high-level sequence diagram:

**1.Actors:** These are the external entities interacting with the system. For the Kindergarten Management System, I've identified the primary actors as:

* 1. **Parent:** Represents the parent or guardian using the system.
  2. **Teacher:** Represents the teaching staff interacting with the system.
  3. **Administrator:** Represents the administrative personnel managing the system.

**2.Major Subsystems/Components:** These represent the significant functional blocks within the Kindergarten Management System. I've broken it down into these key areas:

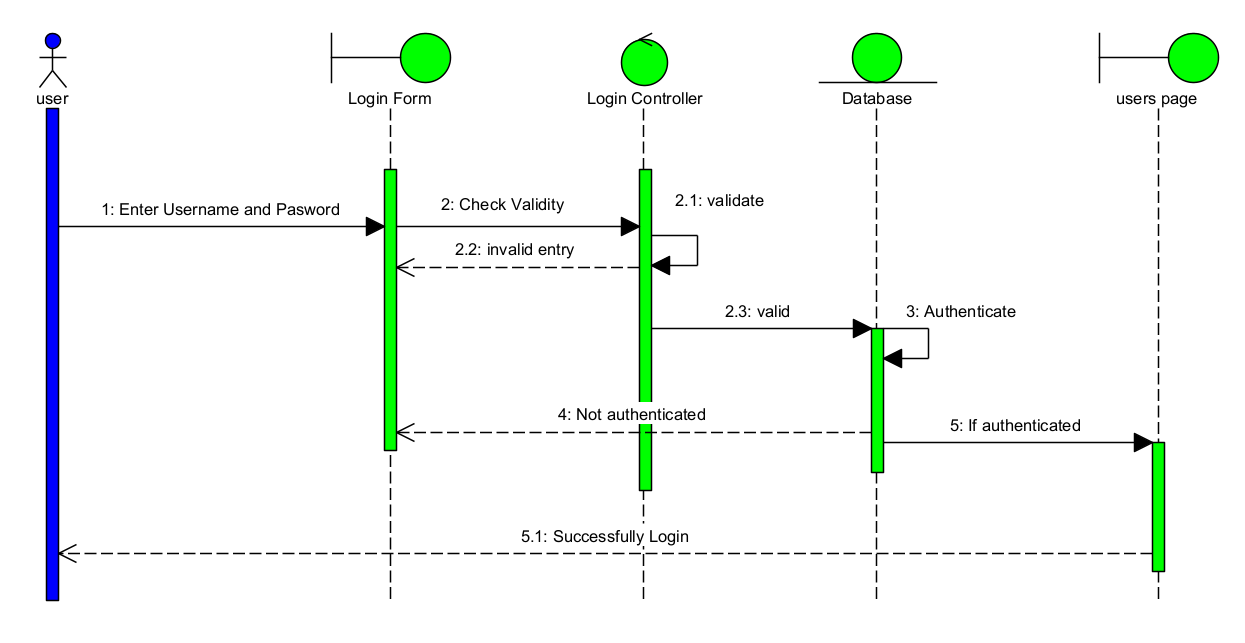
* 1. **User Interface (UI):** This is the presentation layer where actors interact with the system (e.g., web pages).
  2. **Student Management:** This component handles all operations related to student data (enrollment, records, etc.).
  3. **Attendance Management:** This component manages the recording and tracking of student attendance.
  4. **Fee Management:** This component deals with tuition fees, payments, and invoicing.
  5. **Communication:** This component facilitates communication between parents, teachers, and administrators.
  6. **Reporting:** This component generates various reports (e.g., attendance, fees).
  7. **Database:** This represents the system's data storage.

**3.High-Level Messages:** These are the significant interactions or data flows between the actors and the major subsystems. They are labeled with a brief description of the action.

4. **Lifelines:** These are the vertical dashed lines extending below each actor and major subsystem, indicating their participation in the sequence.

**2.3 EXAMPLES OF HIGH LEVEL SEQUENCE DIAGRAMS**

High Level Sequence Diagram For Login System

****

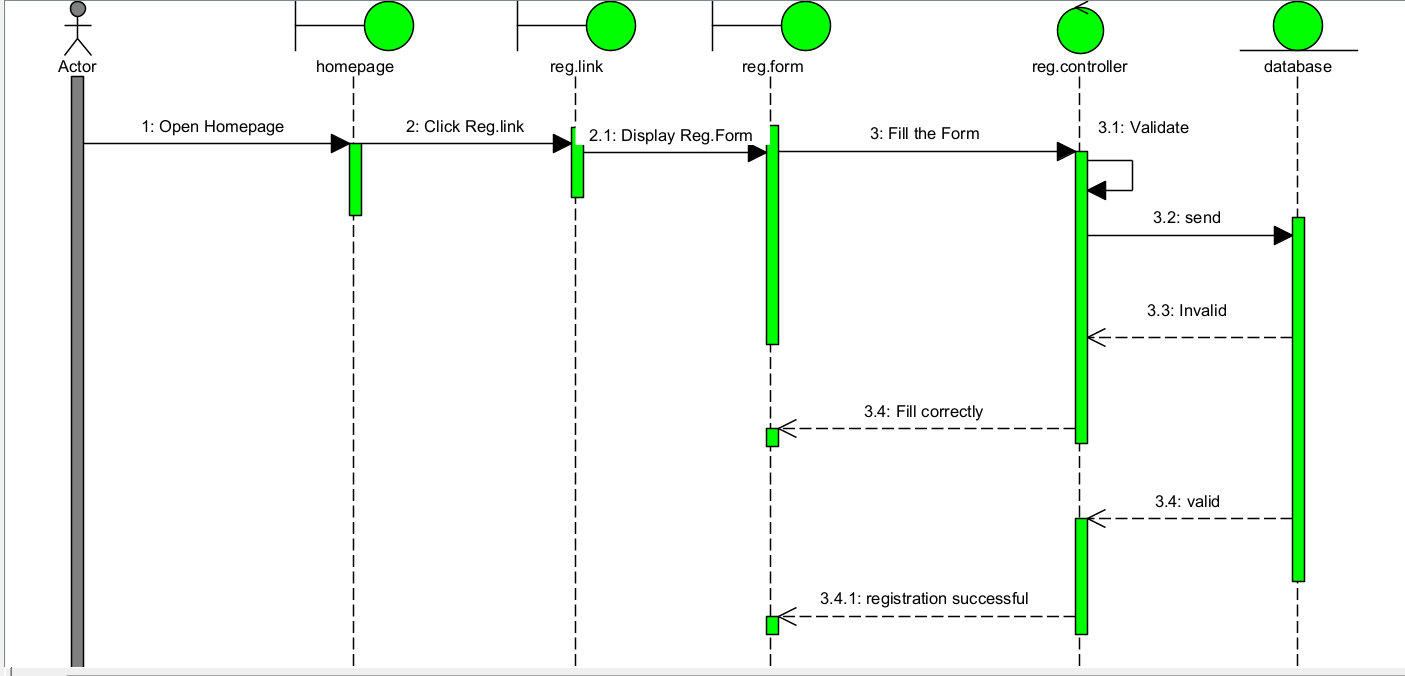
**Figure 2.2 High Level Sequence Diagram For Login**

**Tools and Steps to Draw High Level Sequence Diagram**

**Steps to Draw High Level Sequence Diagram**

* **Defining the Lifelines:** The first step was to establish the participants involved in the user login process. These are represented by the labeled rectangles at the top: "User" (actor), "Login Form," "Login Controller," "Database," and "Users Page." Below each rectangle, a vertical blue dashed line (the lifeline) extends downwards, indicating the existence of these entities throughout the process.
* **User Action (Entering Credentials):** A horizontal arrow originates from the "User" actor lifeline and points to the "Login Form" lifeline, labeled "1 : Enter Username and Password." This represents the user interacting with the login form by entering their credentials.
* **Initiating Validity Check:** A horizontal arrow originates from the "Login Form" lifeline and points to the "Login Controller" lifeline, labeled "2 : Check Validity." This indicates the login form sending the entered credentials to the login controller for validation.
* **Controller Validation:** A labeled box "4: Validate Data" is placed on the "Login Controller" lifeline. This rectangle indicates an internal action or processing step within the login controller to validate the received credentials.
* **Invalid Entry Scenario:** A horizontal dashed arrow originates from the "Login Controller" lifeline and points back to the "Login Form" lifeline, labeled "2.2 : invalid entry." This indicates a scenario where the login controller determines the entered credentials are invalid and sends an "invalid entry" response back to the login form.
* **Valid Credentials Scenario:** A horizontal arrow originates from the "Login Controller" lifeline and points to the "Database" lifeline, labeled "2.1 : validate." This indicates a scenario where the login controller considers the format of the credentials valid and sends them to the database for authentication.
* **Database Authentication:** A labeled box "3 : Authenticate" is placed on the "Database" lifeline. This rectangle indicates an internal action or processing step within the database to authenticate the provided username and password.
* **Authentication Failure:** A horizontal dashed arrow originates from the "Database" lifeline and points back to the "Login Controller" lifeline, labeled "4 : Not authenticated." This indicates a scenario where the database fails to authenticate the user based on the provided credentials.
* **Successful Authentication and Redirection:** A horizontal arrow originates from the "Login Controller" lifeline and points to the "Users Page" lifeline, labeled "5 : If authenticated". This represents the Login Controller directing the flow to the Users Page upon successful authentication.
* **Displaying Users Page to User:** A horizontal dashed arrow originates from the "Users Page" lifeline and points back to the "User" lifeline, labeled "5.1 : Successfully Login". This represents the Users Page being displayed to the User, signifying a successful login.

High Level Sequence Diagram For Student Registration System

****

**Figure 2.1 High Level Sequence Diagram For Student Registration**

**2.4 Tools and Steps to Draw High Level Sequence Diagram**

**Tool Used:visual paradigm**

**Steps to Draw High Level Sequence Diagram**

* **Defining the Lifelines:** The first step was to establish the participants involved in the student registration process.

These are represented by the labeled at the top:

* "**Registration of users**" (actor), "**Manager**" (actor), "**Home Page**," "**Registration Link**," "**Registration Form**," "**Registration Controller**," and "**Database**." Below each rectangle, a vertical blue line (the lifeline) extends downwards, indicating the existence of these entities throughout the process.
* **Initiating Action (User Interaction):** The process begins with a horizontal arrow originating from the "**Registration of users**" actor lifeline and pointing to the "**Manager**" actor lifeline, labeled 1: Open Homepage. This represents the user initiating the process by opening the homepage, likely managed by the "Manager" role or system.
* **Clicking Registration Link:** A horizontal arrow originates from the "**Manager**" actor lifeline and points to the "**Home Page**" lifeline, labeled 2: Click Registration Link. This indicates the manager (or user through the manager interface) interacting with the homepage by clicking a registration link.
* **Displaying Registration Form:** A horizontal arrow originates from the "**Home Page**" lifeline and points to the "**Registration Link**" lifeline, labeled 3: Display Registration Form. This suggests the homepage triggers the display of the registration form link.
* **Filling the Form:** A horizontal arrow originates from the "**Registration Link**" lifeline and points to the "**Registration Form**" lifeline, labeled 3.1: Fill the Form. This represents the user interacting with the registration link, which leads to the display and subsequent filling of the registration form.
* **Submitting Form Data:** A horizontal arrow originates from the "**Registration Form**" lifeline and points to the "**Registration Controller**" lifeline, labeled 4: validate. This indicates the registration form sending the filled data to the registration controller for validation.
* **Sending Data to Database:** A horizontal arrow originates from the "**Registration Controller**" lifeline and points to the "**Database**" lifeline, labeled 5: send. This represents the registration controller sending the (potentially validated) registration data to the database for storage.
* **Invalid Data Scenario:** A horizontal dashed arrow originates from the "**Database**" lifeline and points back to the "**Registration Controller**" lifeline, labeled 5.1: invalid. This indicates a scenario where the database identifies the sent data as invalid, and it sends an "invalid" response back to the controller.
* **Feedback to Registration Form (Invalid):** A horizontal arrow originates from the "**Registration Controller**" lifeline and points back to the "**Registration Form**" lifeline, labeled 5.1.1: Fill Correctly. This shows the controller instructing the registration form (and thus the user) to correct the invalid data.
* **Valid Data Scenario:** A horizontal dashed arrow originates from the "**Database**" lifeline and points back to the "**Registration Controller**" lifeline, labeled 5.2: Valid. This indicates a scenario where the database successfully processes the registration data as valid and sends a "valid" response back to the controller.
* **Registration Successful Feedback:** A horizontal arrow originates from the "**Registration Controller**" lifeline and points back to the "**Registration Form**" lifeline, labeled 5.2.1: Registration Successful. This shows the controller informing the registration form (and thus the user) that the registration was successful.

High Level Sequence Diagram For Class Enrollement

**Figure 2.3 High Level Sequence Diagram For Class Enrollement**

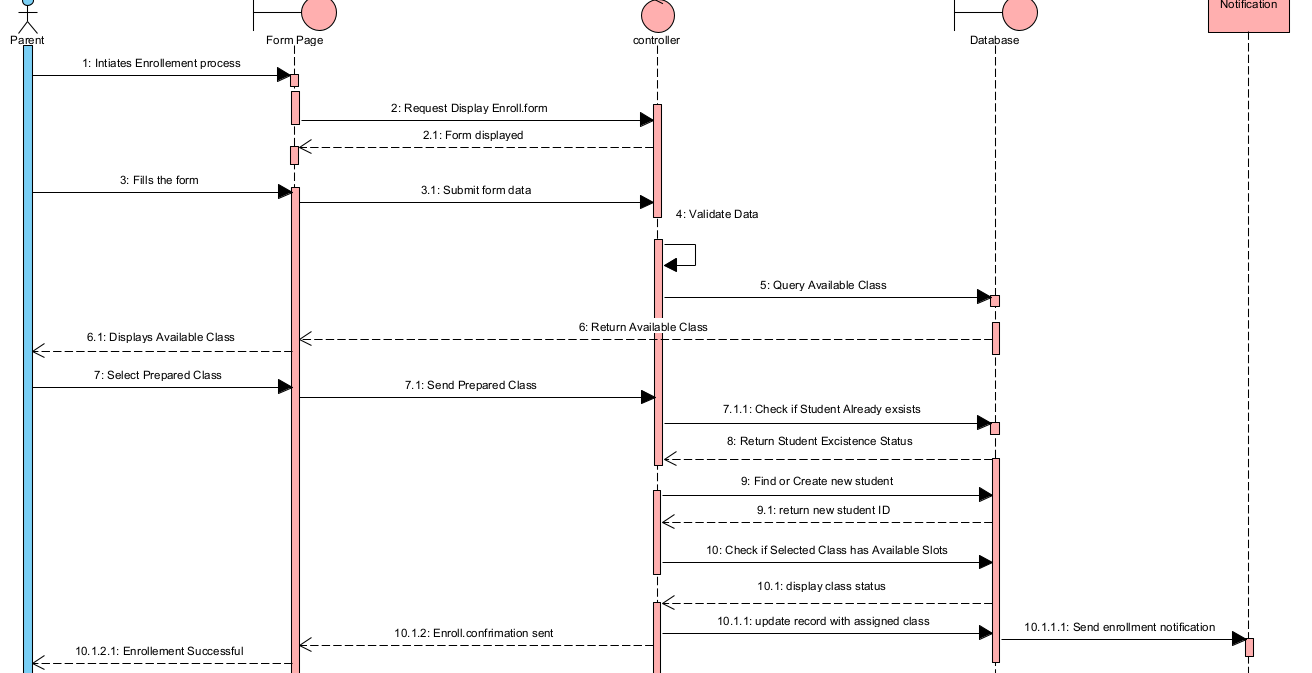
**Tools and Steps to Draw High Level Sequence Diagram**

**Tool Used:visual paradigm**

**Steps to Draw High Level Sequence Diagram**

·**Lifelines:** Draw 5 vertical dashed lines and label them "Parent," "Form Page," "controller," "Database," and "Notification" from left to right.

·**Message 1:** Draw a solid arrow from "Parent" to "Form Page" labeled "1: Initiates Enrollment process."

·**Message 2:** Draw a solid arrow from "Form Page" to "controller" labeled "2: Request Display Enroll form."****

·**Return Message 2.1:** Draw a dashed arrow from "controller" back to "Form Page" labeled "2.1: Form displayed."

·**Message 3:** Draw a solid arrow from "Parent" to "Form Page" labeled "3: Fills the form."

·**Message 3.1:** Draw a solid arrow from "Form Page" to "controller" labeled "3.1: Submit form data."

·**Internal Action 4:** Indicate "4: Validate Data" as an internal action within the "controller's" activation.

·**Message 5:** Draw a solid arrow from "controller" to "Database" labeled "5: Query Available Class."

·**Return Message 6:** Draw a dashed arrow from "Database" back to "controller" labeled "6: Return Available Class."

·**Message 6.1:** Draw a solid arrow from "controller" to "Form Page" labeled "6.1: Displays Available Class."

·**Message 7:** Draw a solid arrow from "Parent" to "Form Page" labeled "7: Select Preferred Class."

·**Message 7.1:** Draw a solid arrow from "Form Page" to "controller" labeled "7.1: Send Prepared Class."

·**Message 7.1.1:** Draw a solid arrow from "controller" to "Database" labeled "7.1.1: Check If Student Already exists."

·**Return Message 8:** Draw a dashed arrow from "Database" back to "controller" labeled "8: Return Student Existence Status."

·**Message 9:** Draw a solid arrow from "controller" to "Database" labeled "9: Find or Create new student."

·**Return Message 9.1:** Draw a dashed arrow from "Database" back to "controller" labeled "9.1: return new student ID."

·**Message 10:** Draw a solid arrow from "controller" to "Database" labeled "10: Check If Selected Class has Available Slots."

·**Return Message 10.1:** Draw a dashed arrow from "Database" back to "controller" labeled "10.1: display class status."

·**Message 10.1.1:** Draw a solid arrow from "controller" to "Database" labeled "10.1.1: update record with assigned class."

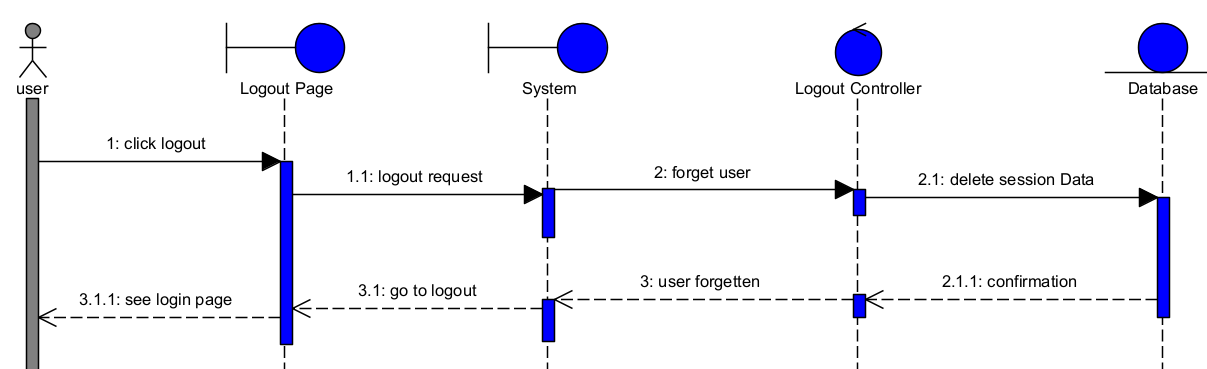
·**Message 10.1.1.1:** Draw a solid arrow from "controller" to "Notification" labeled "10.1.1.1: Send enrollment notification."

·**Return Message 10.1.2:** Draw a dashed arrow from "Database" back to "controller" labeled "10.1.2: Confrimation"

·**Return Message 10.2:** Draw a dashed arrow from "controller" back to "Form Page" labeled "10.2: Enroll confirmation sent."

·**Message 10.2.1:** Draw a solid arrow from "Form Page" to "Parent" labeled "10.2.1: Enrollment Successful."

High Level Sequence Diagram For Logout

****

**Figure 2.4 High Level Sequence Diagram For Logout**

**Tools and Steps to Draw High Level Sequence Diagram**

* **Tool used:visual paradigm**

**Steps to Draw High Level Sequence Diagram**

·**Lifelines:** Draw 5 vertical dashed lines and label them "user," "Logout Page," "System," "Logout Controller," and "Database" from left to right.

·**Message 1:** Draw a solid arrow from the "user" to the "Logout Page" labeled "1: click logout."

·**Message 1.1:** Draw a solid arrow from the "Logout Page" to the "System" labeled "1.1: logout request."

·**Message 2:** Draw a solid arrow from the "System" to the "Logout Controller" labeled "2: forget user."

·**Message 2.1:** Draw a solid arrow from the "Logout Controller" to the "Database" labeled "2.1: delete session Data."

·**Return Message 2.1.1:** Draw a dashed arrow from the "Database" back to the "Logout Controller" labeled "2.1.1: confirmation."

·**Return Message 3:** Draw a dashed arrow from the "Logout Controller" back to the "System" labeled "3: user forgotten."

·**Return Message 3.1:** Draw a dashed arrow from the "System" back to the "Logout Page" labeled "3.1: go to logout."

·**Message 3.1.1:** Draw a solid arrow from the "Logout Page" back to the "user" labeled "3.1.1: see login page."