JEDI-07 GROUP 9
COURSE REGISTRATION SYSTEM

STAKEHOLDERS

1 **Sponsors**

- Flipkart

2 **SME's**

- Amit Balyan

3 **Coordinator**

- Deepika Gajaraj
- Shree Lakshmi

AGENDA

01 Our Journey

02 Our Team

03 Team Structure

04 Problem Statement

05 Engineering Practices

06 Tech Stack

07 Demo

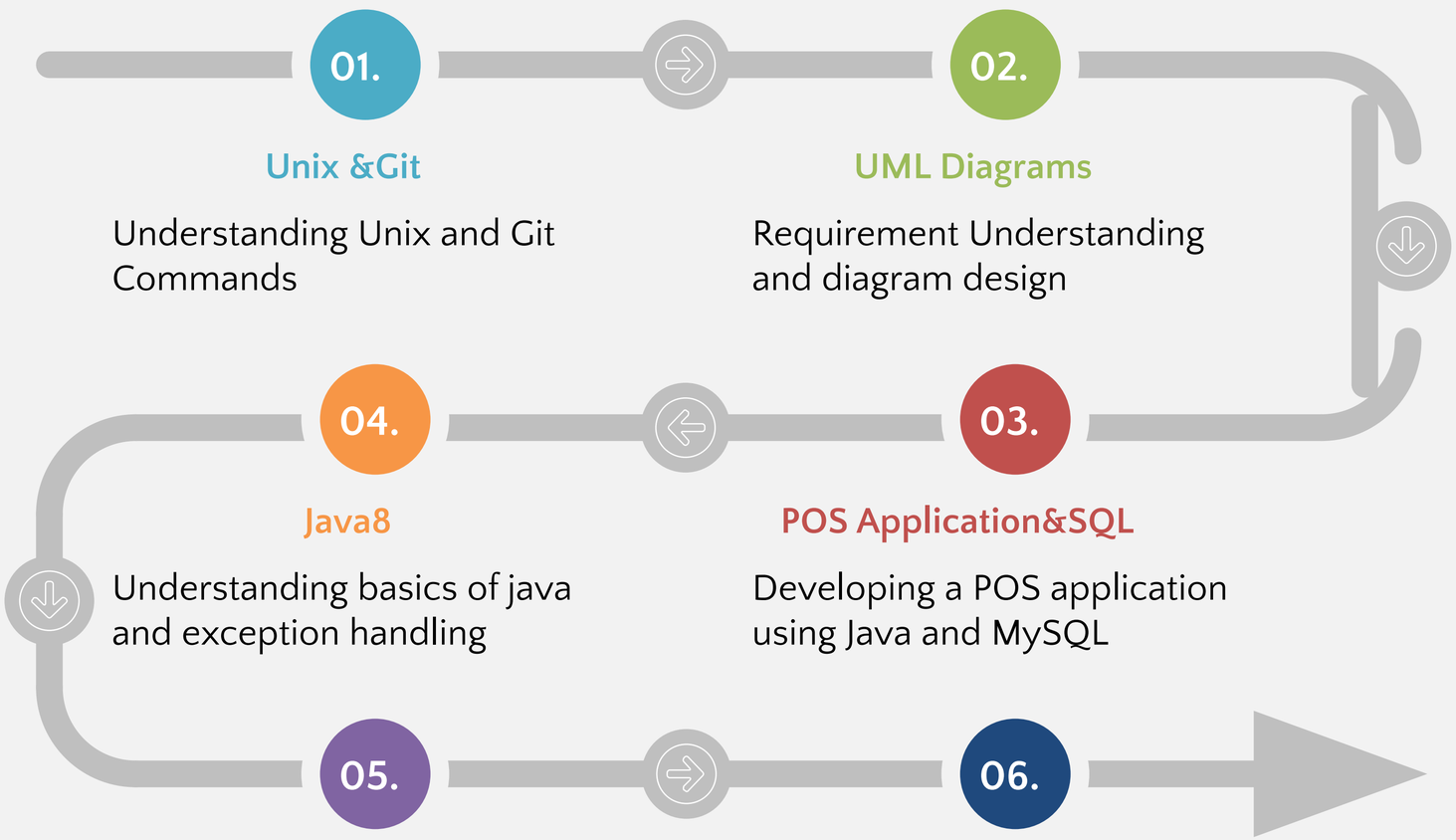
08 Development

09 Challenges & Learnings

10 Questions

OUR JOURNEY

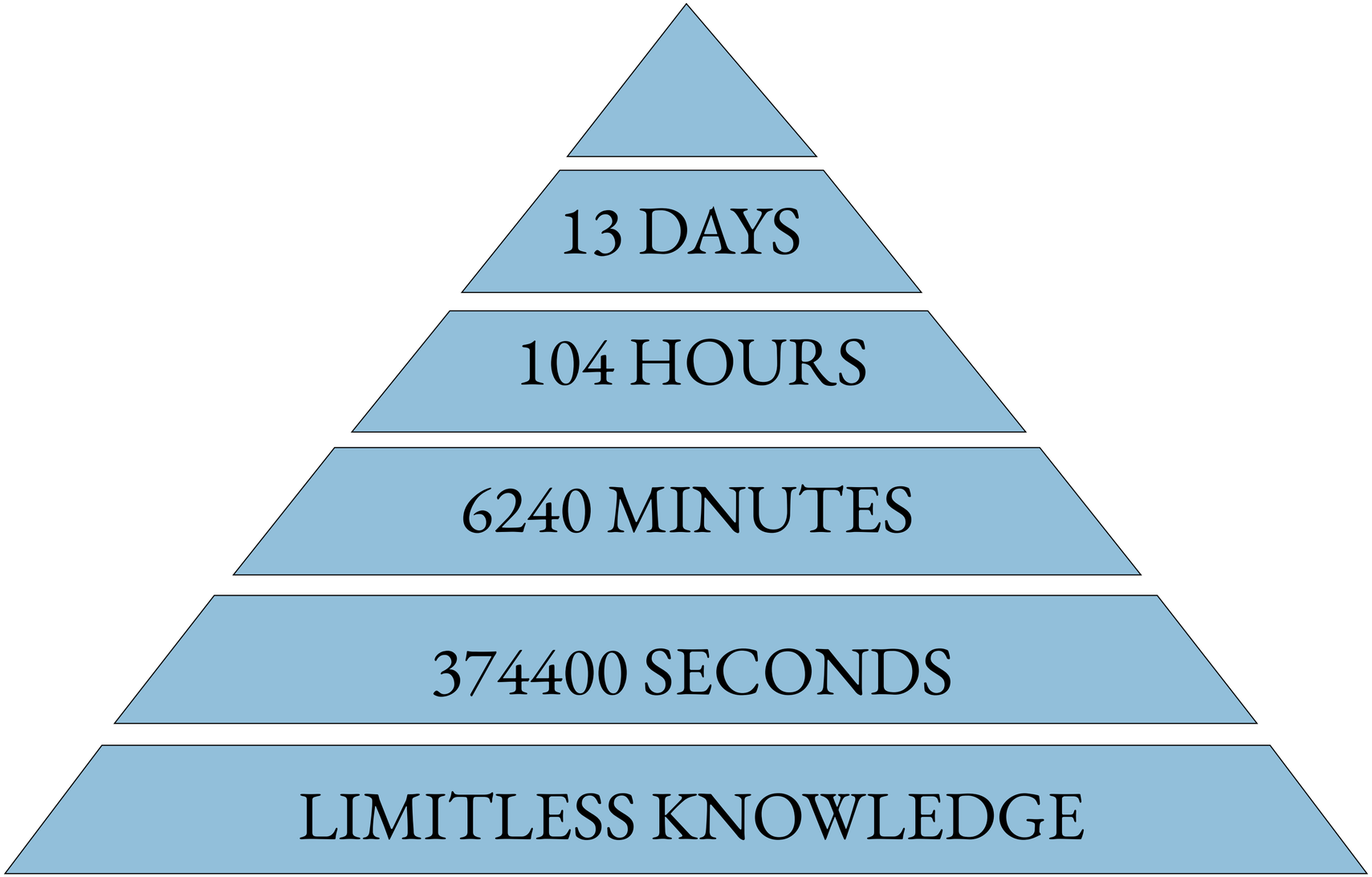
Action Plan


01.
Unix &Git
Understanding Unix and Git Commands
02.
UML Diagrams
Requirement Understanding and diagram design
03.
POS Application&SQL
Developing a POS application using Java and MySQL
04.
Java8
Understanding basics of java and exception handling
05.
Java REST
Understanding about Java RESTful services
06.
Jersey&Drop Wizard
REST API development using jersey and DropWizard

Timeline	Modules	Description
17th Jan – 20th Jan	• Unix commands • Git commands • SDLC • UML Diagrams	• Environment Setup • Understanding Unix and Git commands • Requirement analysis and modelling
21st Jan – 27th Jan	• Java 8 • SQL	• Developing a POS Application using Java and MySQL. • Understanding Java 8 features and collections
28th Jan – 3rd feb	• Rest • Jersey • DropWizard	• Developing REST API using Jersey and DropWizard framework

OUR NEXT STARTS NOW

3 weeks of Training + Project demo



THE TEAM

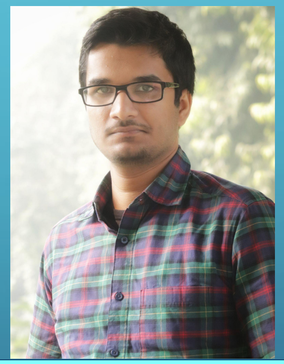
Arunesh Sarker

Venkat Karthik

Anshika Bansal

Tushar Jaiswal

Ruchika Swain



Jayant Kishore

Vishal Thirwani

Mehak Goel

PROJECT GOALS

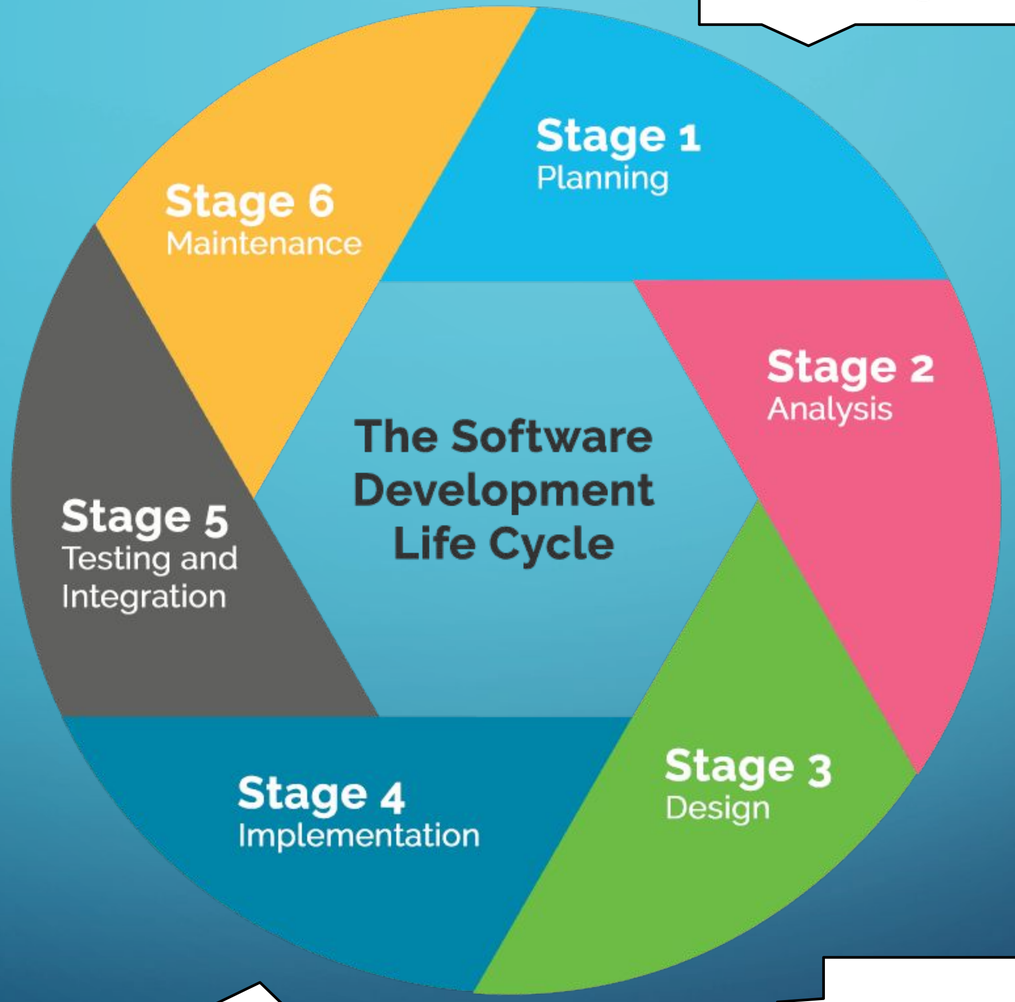
PROBLEM STATEMENT

Develop a **Course Registration System**

- Entities involved are: Student, Professor, Admin
- Students should be able to Register for courses and view an electronic report card at the end of the semester.
- Professors access the system to choose and teach courses, record grades.
- The Admin should be able to Approve a Student, Register a Professor, add/remove course from course catalog.
- The System should allow students to select courses (4 minimum with choice to add 2 extra courses at max).
- Course offerings will have a maximum of ten students and a minimum of three students.
- After the registration, the system sends information to the billing system so that the student can be billed for the semester.

ENGINEERING PRACTICES

Requirement Gathering



Tech Stack Selection

UML Artifacts, Architecture Design

Code development

Testing & Bug Fixes


GitHub


Java


git


MySQL


Unified
Modeling
Language


TECHSTACK


Jersey
RESTful Web Services in Java.


Apache Tomcat


REST

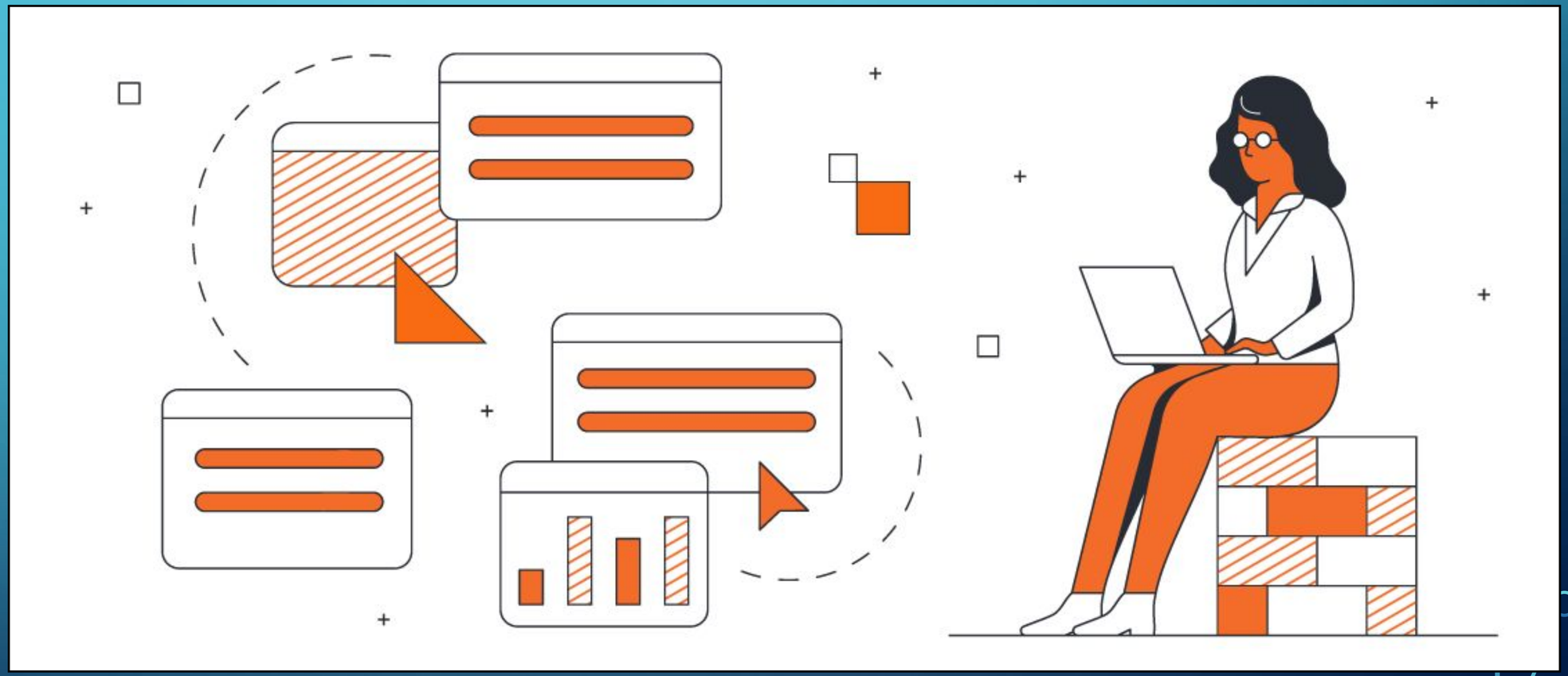

Dropwizard

UML (Unified Modelling Language)

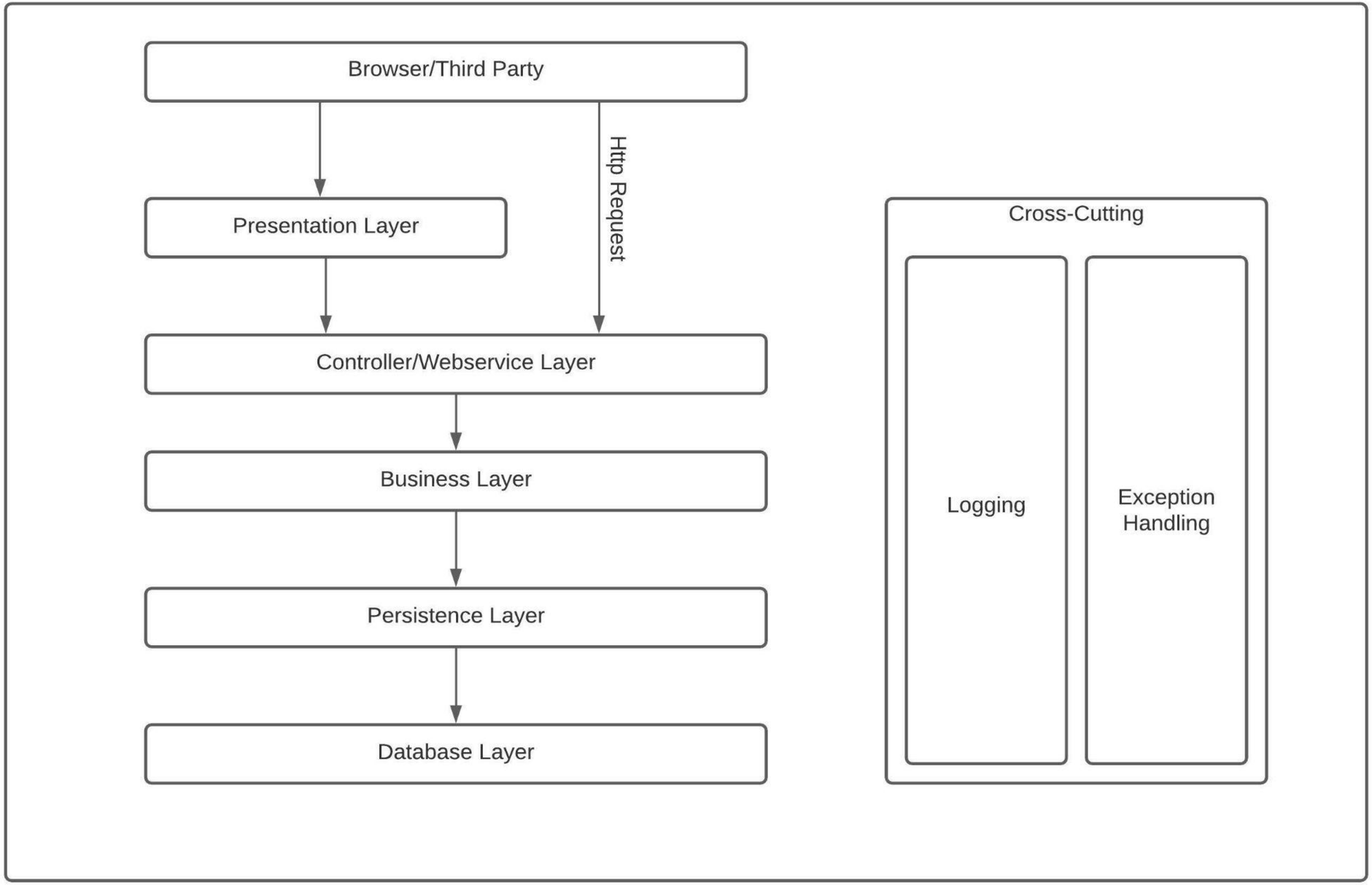
UML artifacts:

1. Use Case Diagrams
2. Activity Diagrams
3. Sequence Diagrams
4. Class Diagrams

ARCHITECTURE DESIGN



LAYERED ARCHITECTURE



DESIGN DESCRIPTION

- Components are organized into horizontal layers, each layer performing a specific role within the application. Components within a specific layer deal only with logic that pertains to that layer.
- Separation of Concern: Layers are isolated.Each layer is independent of the other layers, thereby having little or no knowledge of the inner workings of other layers in the architecture.
- Why we used this pattern?
  - Code Organization: Code is more readable.
  - Ease of development: Team members can work on various layers independently.
  - Testing: Layers can be tested independently.
- Challenges with this pattern:
  - Deployment: Tightly coupled code, may end up doing monolithic deployment.
  - Scalability: Expensive to scale due to monolithic deployment.


DEMO

LEARNINGS AND CHALLENGES

CHALLENGES

- Understanding the problem statement and initial implementation using Use Case, activity, sequence and class diagram.
- Installing software on multiple environments.
- Git merge conflicts.
- Resolving jar and version dependencies.
- Migration of project from one tech stack to another.
- Database integration challenges.
- Migration from one tech stack to another

LEARNINGS

- Familiarization with Unix Commands.
- Getting hang of the basic Git commands.
- Software Development Life Cycle.
- Knowledge and Implementation of Activity, Sequence and Use Case Diagram.
- Java 8 features.
- Integration of REST Services with Java code using Tomcat and Jersey.
- Dropwizard Integration and Advantages.
- Usage of Postman for testing APIs.
- Javadoc Generation & basic Application Monitoring.

ANY QUESTIONS?

THANK YOU!!