

Flipkart



JEDI 2.0 / GROUP B

The Problem Statement

Design a UI and a backend system for a new enterprise application that Flipkart is launching, FlipFit.

Flipkart is partnering up with gyms across Bangalore to enter into the fitness space.

Framework for 6 days

Day 1:

- Discussion on project objectives and requirements
- Overview of UML diagrams and initial project design
- Doubt clearance session

Day 2:

- Discussion on technologies to be used (Java 17, JDBC, MySQL)
- Understanding Java frameworks and database integration
- SME/Trainer session on core project aspects

Day 3:

- Review of UML diagrams and transformation into project structure
- Begin coding key components (DAO, Business Logic, Beans)
- Trainer feedback on progress

Day 4:

- Implementation of business logic and role-based access control
- Exception handling and collection API usage
- Continuous integration discussion

Day 5:

- Implementing stream API and advanced Java features
- Code review session with SME/Trainer

Day 6:

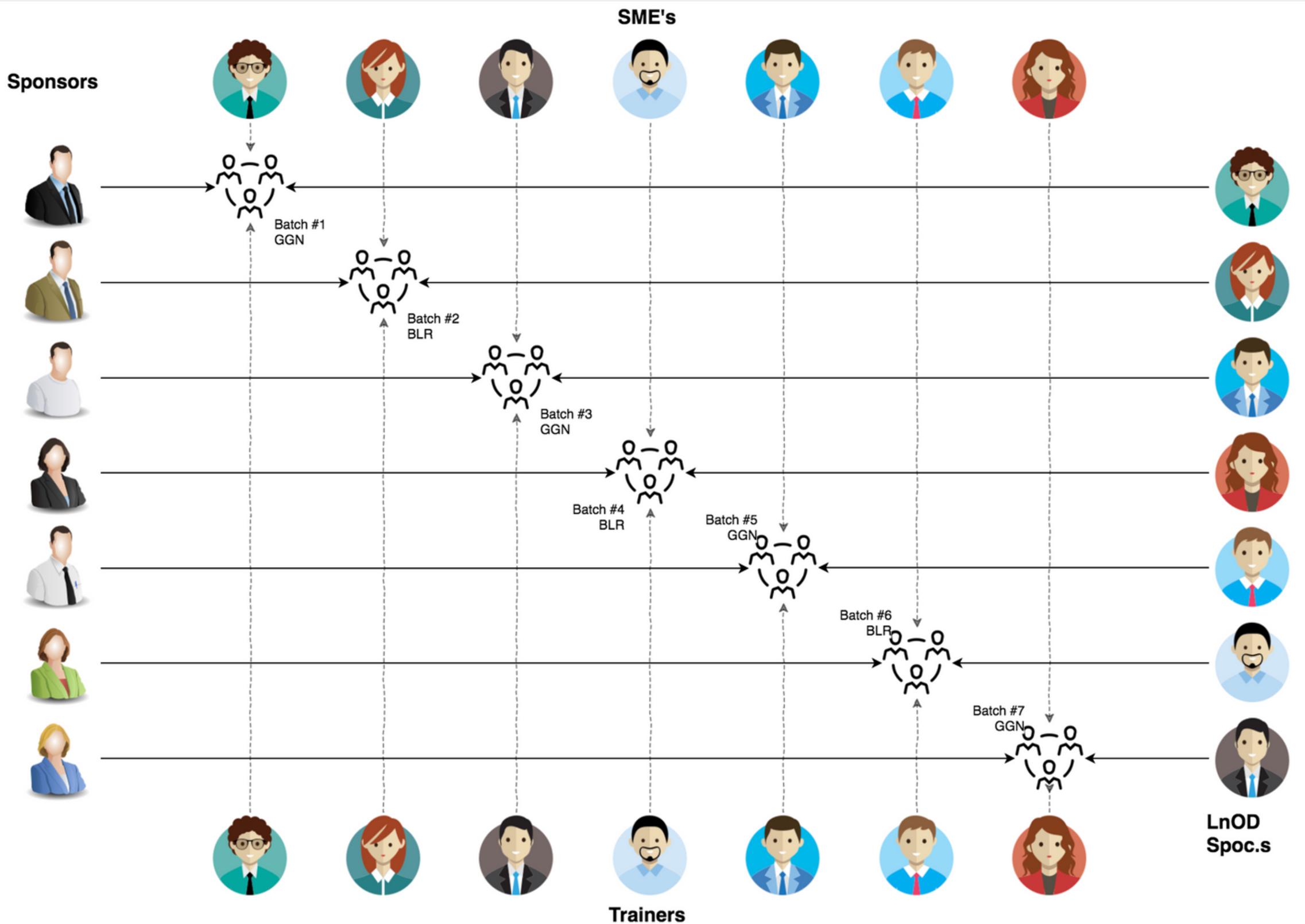
- Showcase new Java 17 features
- Preparing documentation and project reports
- Final deployment and demo preparation

Stakeholders

Sponsors
Flipkart

SME's
Amith Balyan

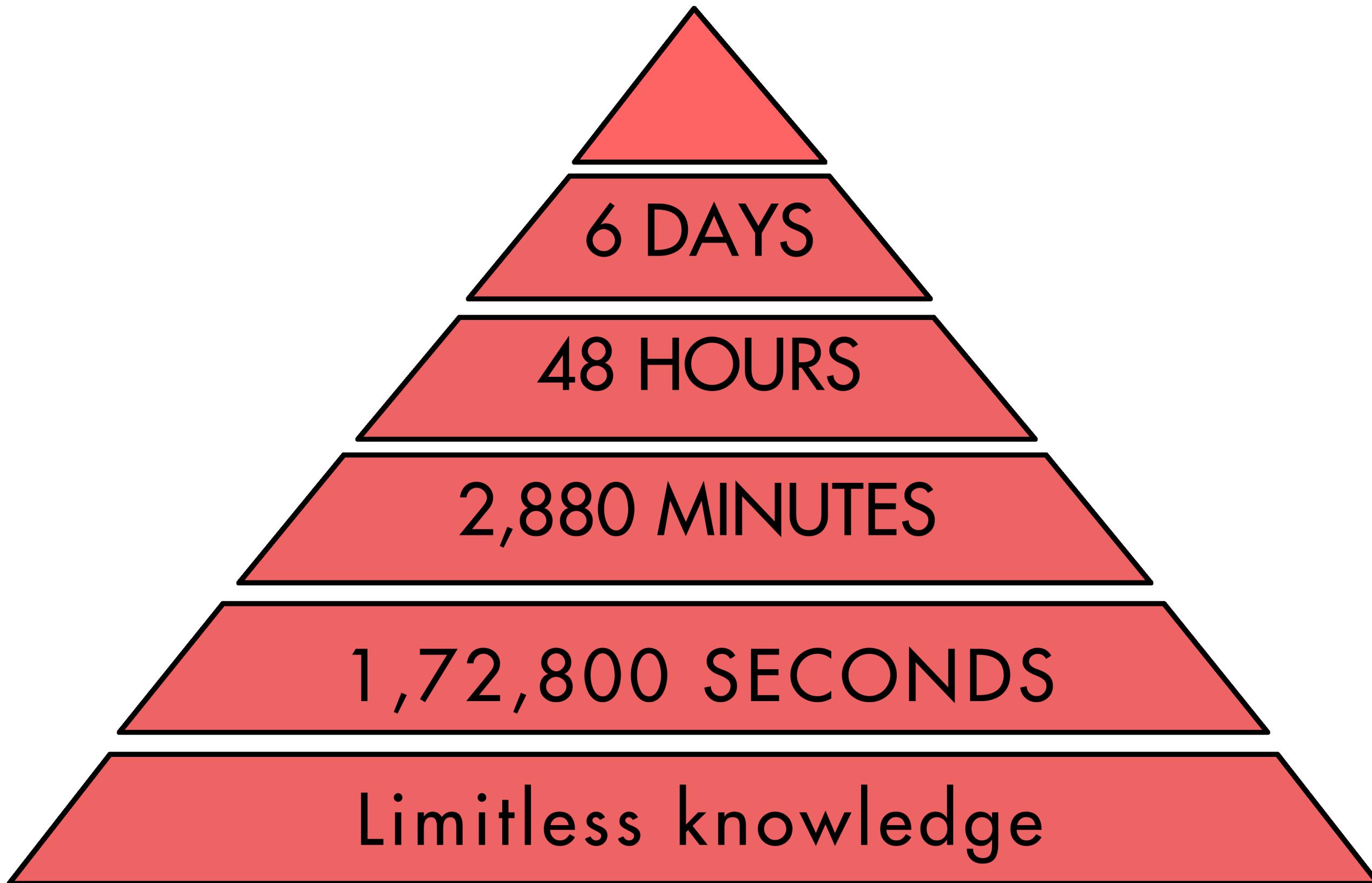
Coordinators
Raksha Dubey
Anushka Khanna
Chaitra Harti





Our next starts **NOW**

6days of Training + Project demo



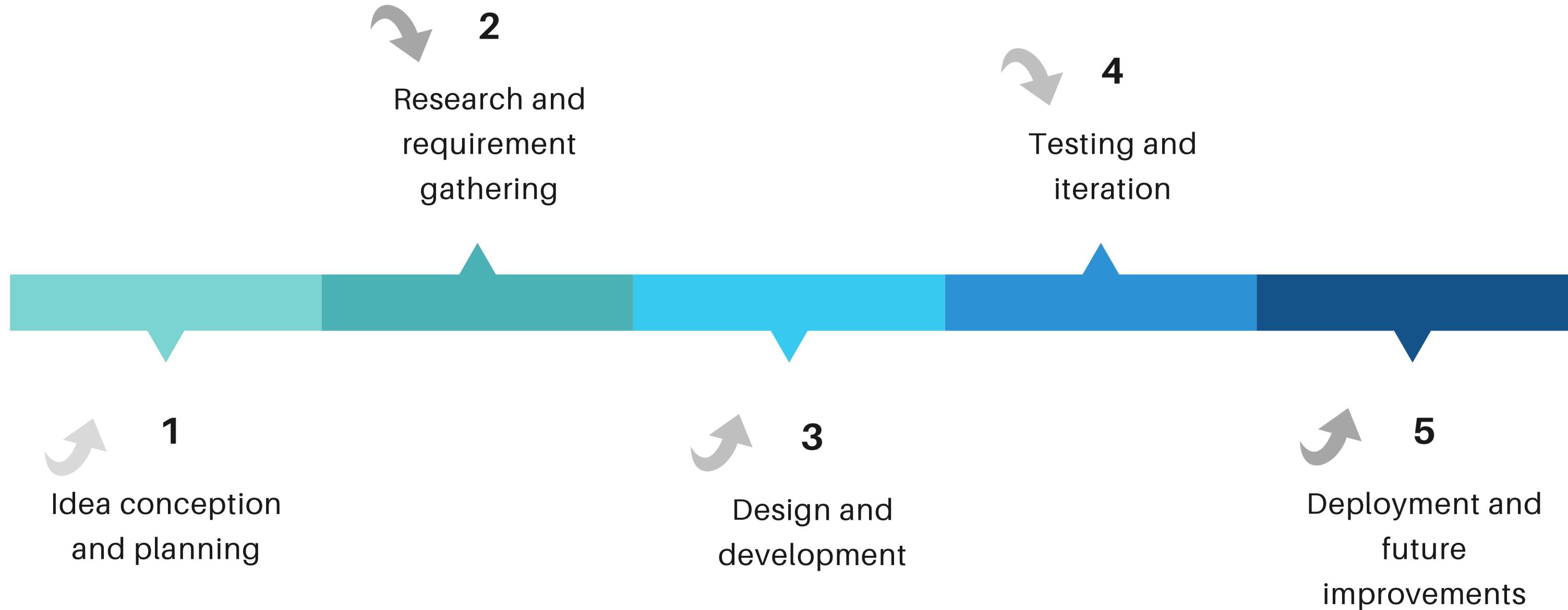
Agenda

- 01 Our Journey
- 02 Our Team
- 03 Project Goals
- 04 Engineering Practices
- 05 Tech Stack
- 06 Development
- 07 Challenges & Learnings
- 08 Demo
- 09 Questions



Our Journey





Our Team





Abhishek



Ayush



Sunny



Shrish



Pratik

Responsibility

Abhishek	Client Application, Database Design, Class Diagram, Integration
Ayush	GymOwner Implementation, Activity Diagram
Pratik	GymCustomer Implementation, Use-Case Diagram
Shrish	GymAdmin Implementation, Activity Diagram
Sunny	User Implementation, Use-Case Diagram

Project Goals



Our Vision

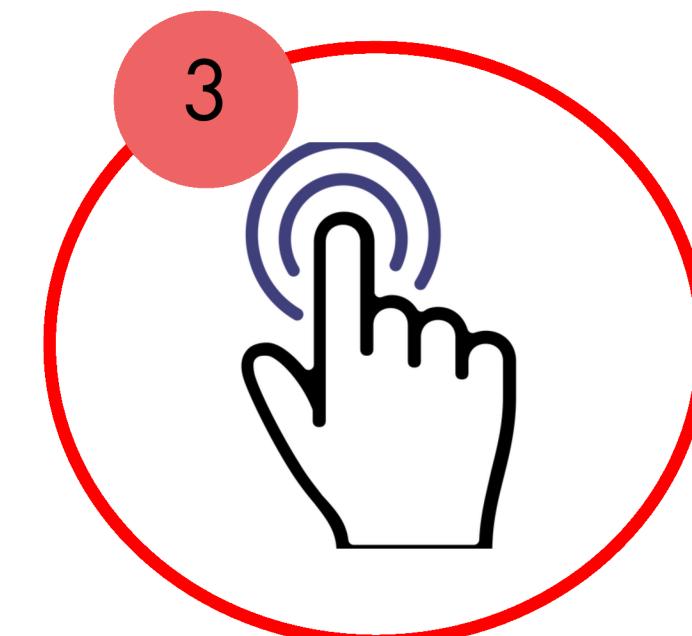
- To build a robust, scalable, and user-friendly gym management system
- Leverage modern Java features for better performance and maintainability
- Follow best engineering practices for software development
- Deliver a fully functional system with high code quality and documentation



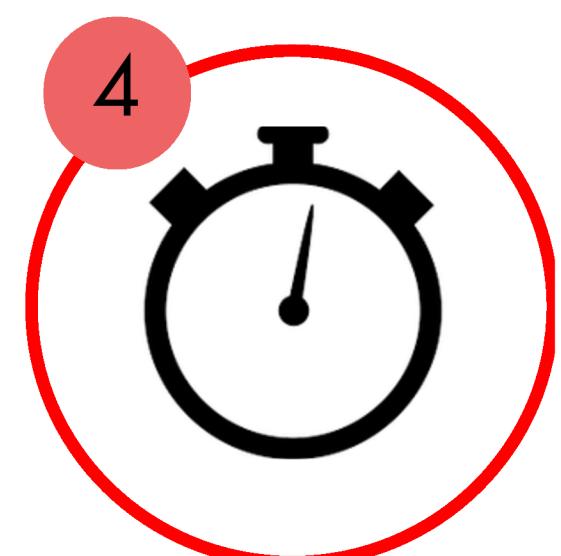
Quality



Security

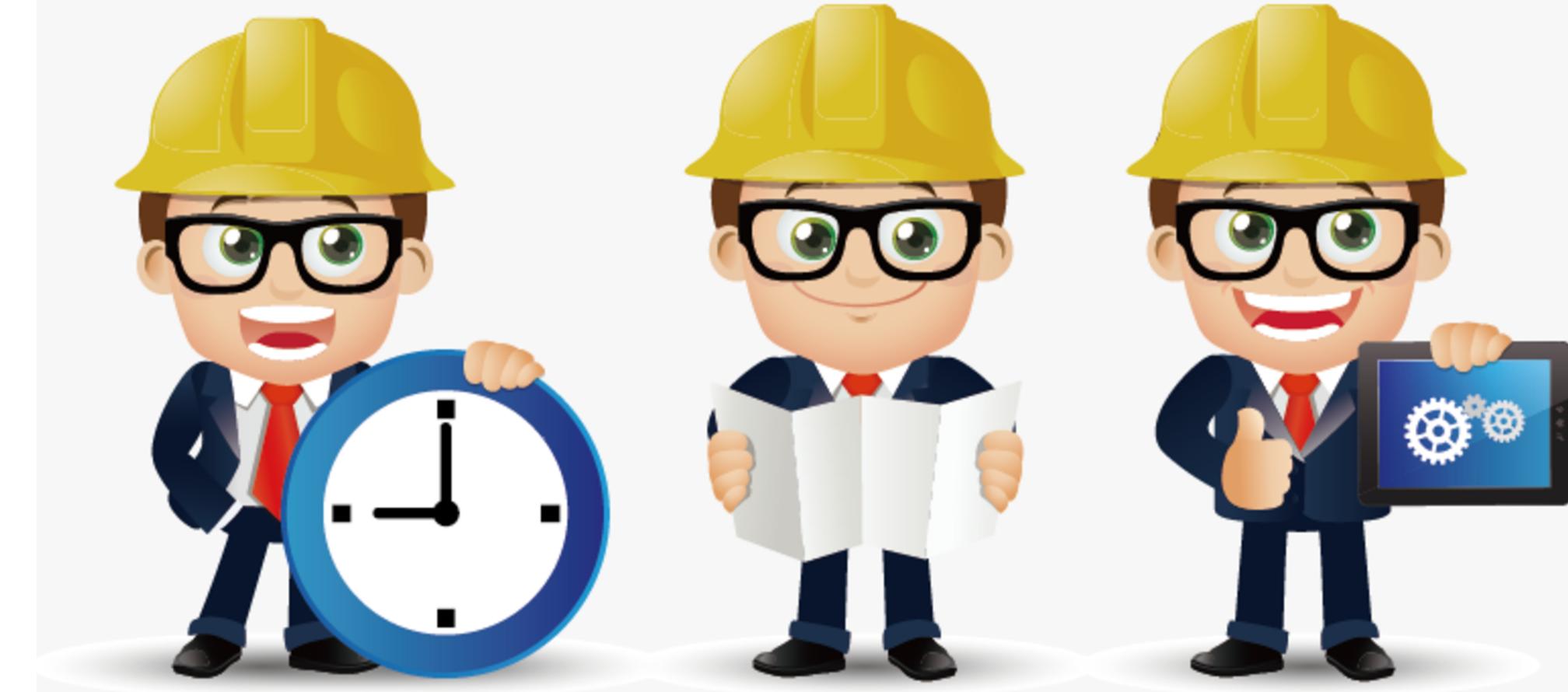


Interactivity



Speed

Engineering Practices



Separation of Concerns: This design principle ensures that the application's different concerns, such as UI, business logic, and data management, are handled by separate layers, making the system easier to maintain, test, and scale.

Git Version Control: Git is used for tracking changes in the project, allowing multiple developers to collaborate efficiently, manage different versions, and roll back to previous code states if necessary.

Microservices Architecture: In this approach, the application is broken down into smaller, independent services that handle specific tasks, allowing for better scalability, flexibility, and easier maintenance of each component.

Tech Stack



Application

Core Language



Infra

Colaboration Tool



Github

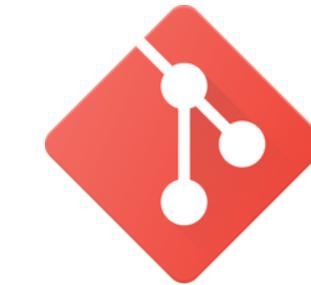
Data

SQL Database



Tools

VCS



Git

IDE



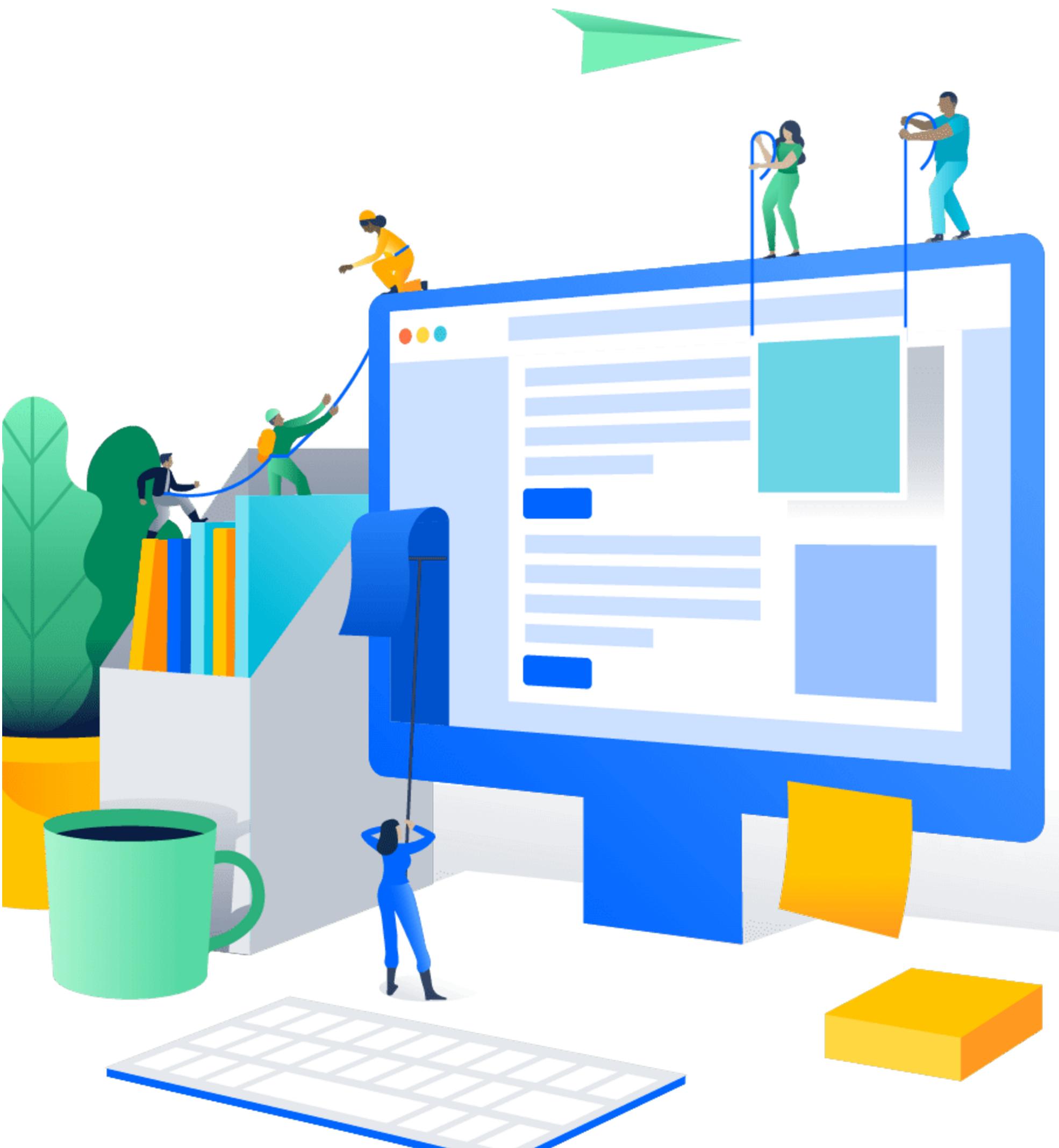
Spring Tool Suit 4

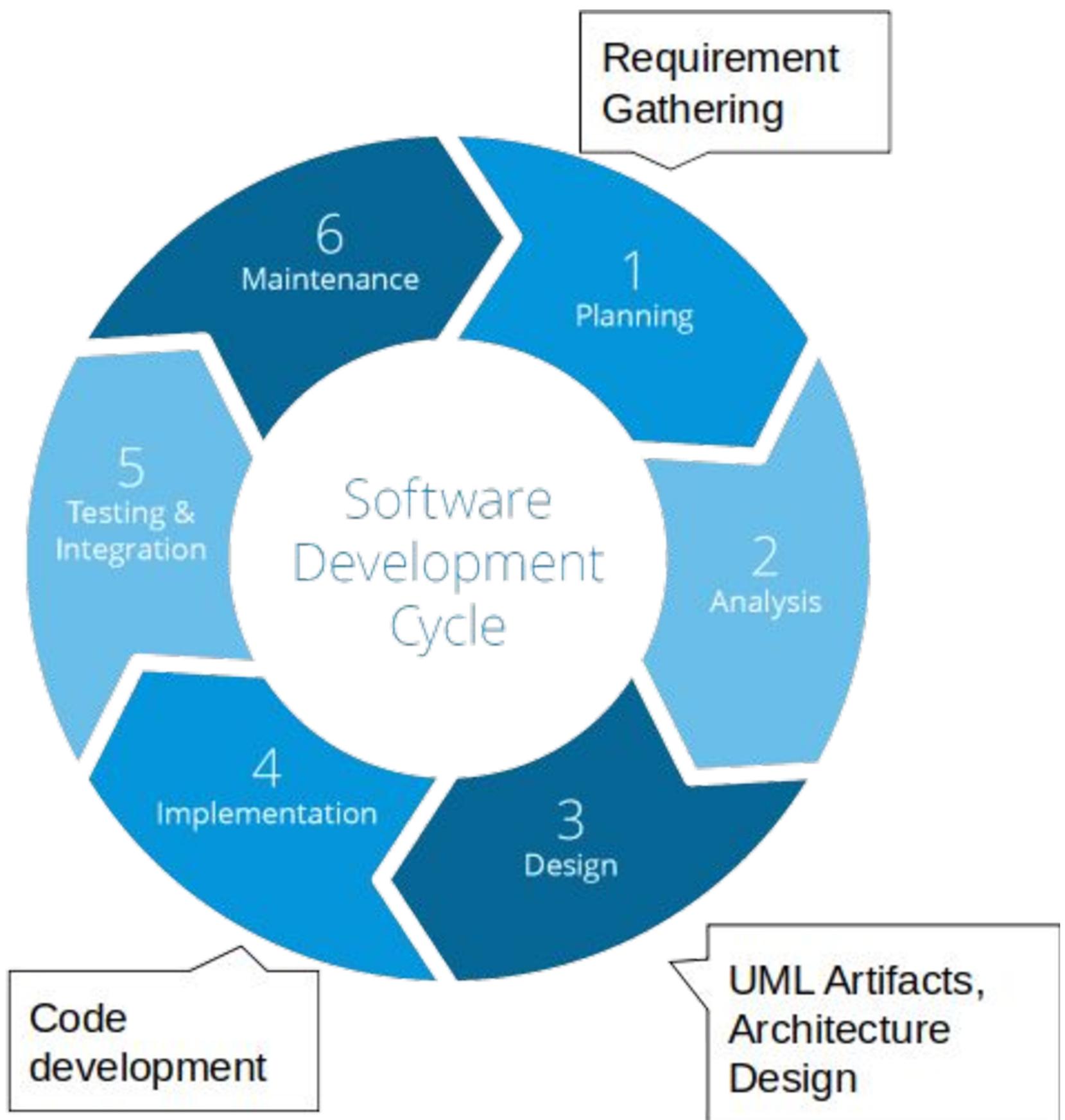
Visual Database
Design Tool



MySQL Workbench

Development



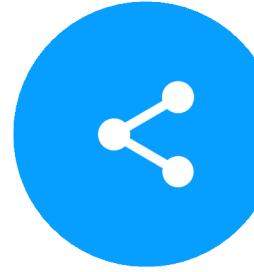


Challenges



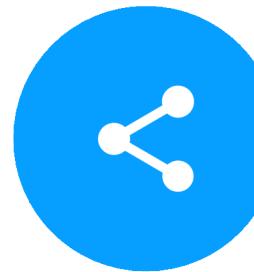
Understanding the Problem Statement

Clearly defining and understanding the problem statement is crucial for aligning the team's efforts and ensuring that the project stays on track.



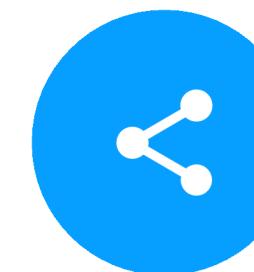
Database Connectivity Issues

Ensuring smooth interactions between the application and MySQL.



Collaboration Issues (Merge Conflicts in Git)

Coordinating development efforts across team members and resolving merge conflicts in Git efficiently to maintain a smooth workflow.



Efficient Exception Handling

Handling various runtime exceptions effectively.

UML Diagrams

Demo



Questions





Thank you