



Flipkart

Where did we start?

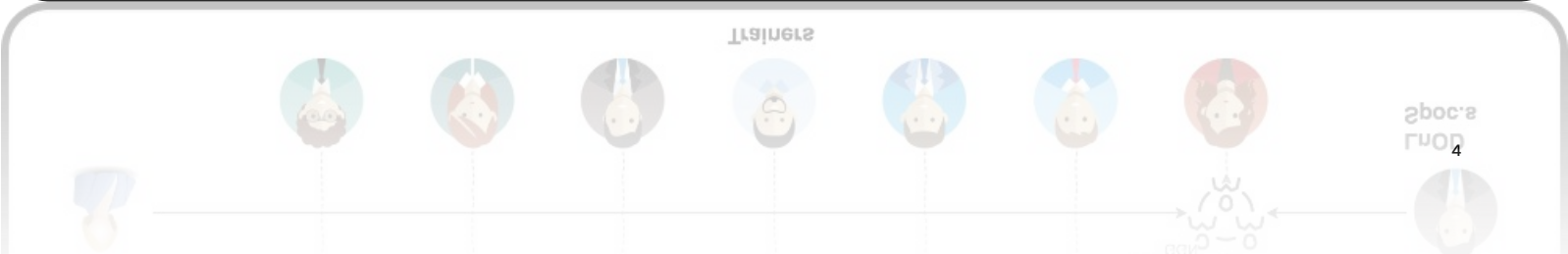
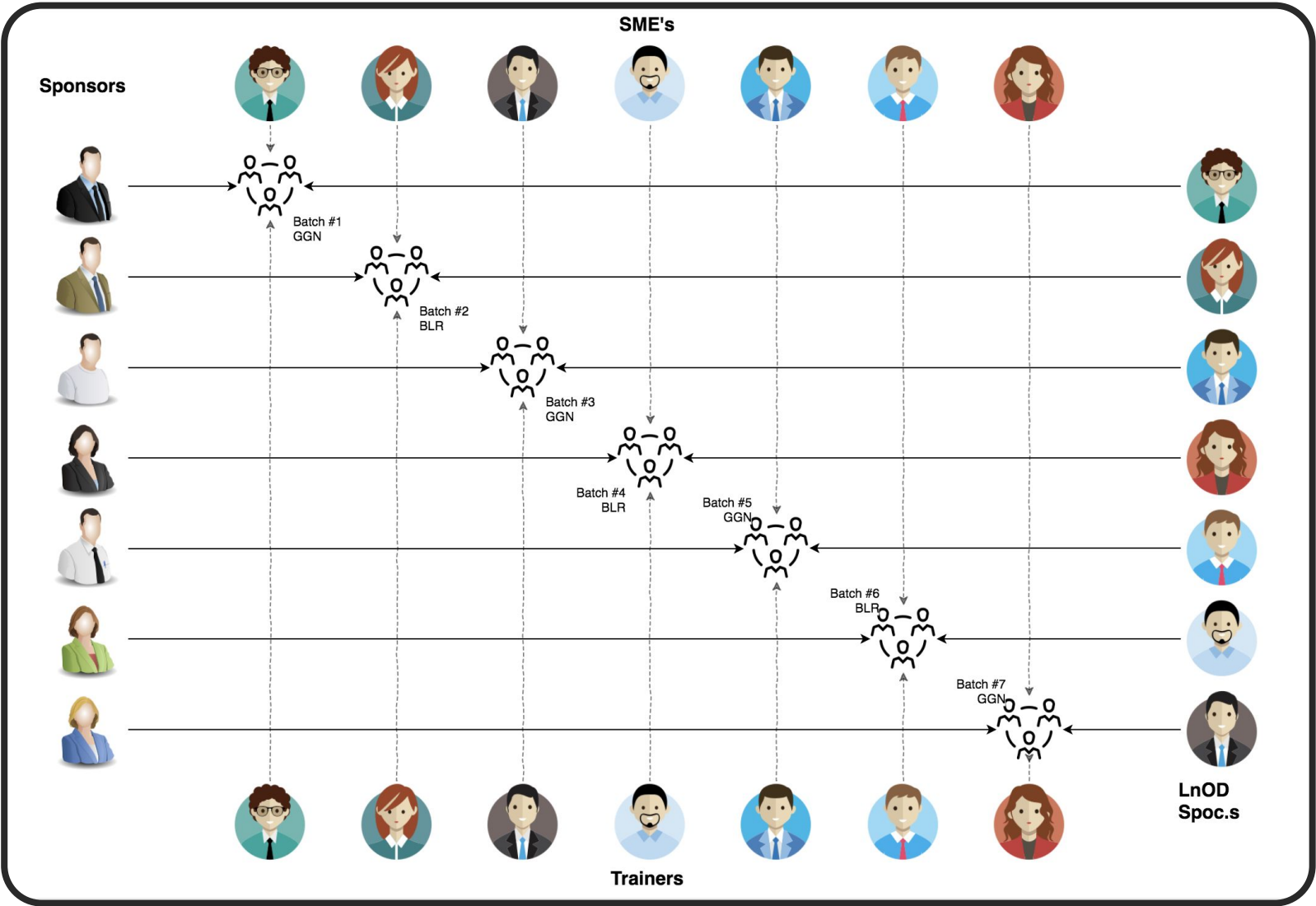
When we started, our team's and individual knowledge was limited to the smaller projects we'd completed in college. On the first day, we only knew the basics of Git and how Java works.

Framework for 1 Week

- 1 week plan
- Everyday Discussion about topics / Technologies / doubt clearance.
 - Git, UML, Java (classes, interfaces, exception handling, Java 17 features), Database Interaction (MySQL, JDBC), and Dropwizard.
- Every day with SME/Trainer discussion of Project progress & transformation based on UML & Technologies

Stakeholders

- 1. Sponsors
 - 1. Flipkart
- 2. SME
 - 1. Mr. Amit Kumar
- 3. Trainers
 - 1. Ms. Anushka khanna



A large white circle is centered on a solid red background. Inside the circle, the text "Our next starts NOW" is written in white.

Our next starts **NOW**

1 WEEK OF TRAINING + PROJECT DEMO



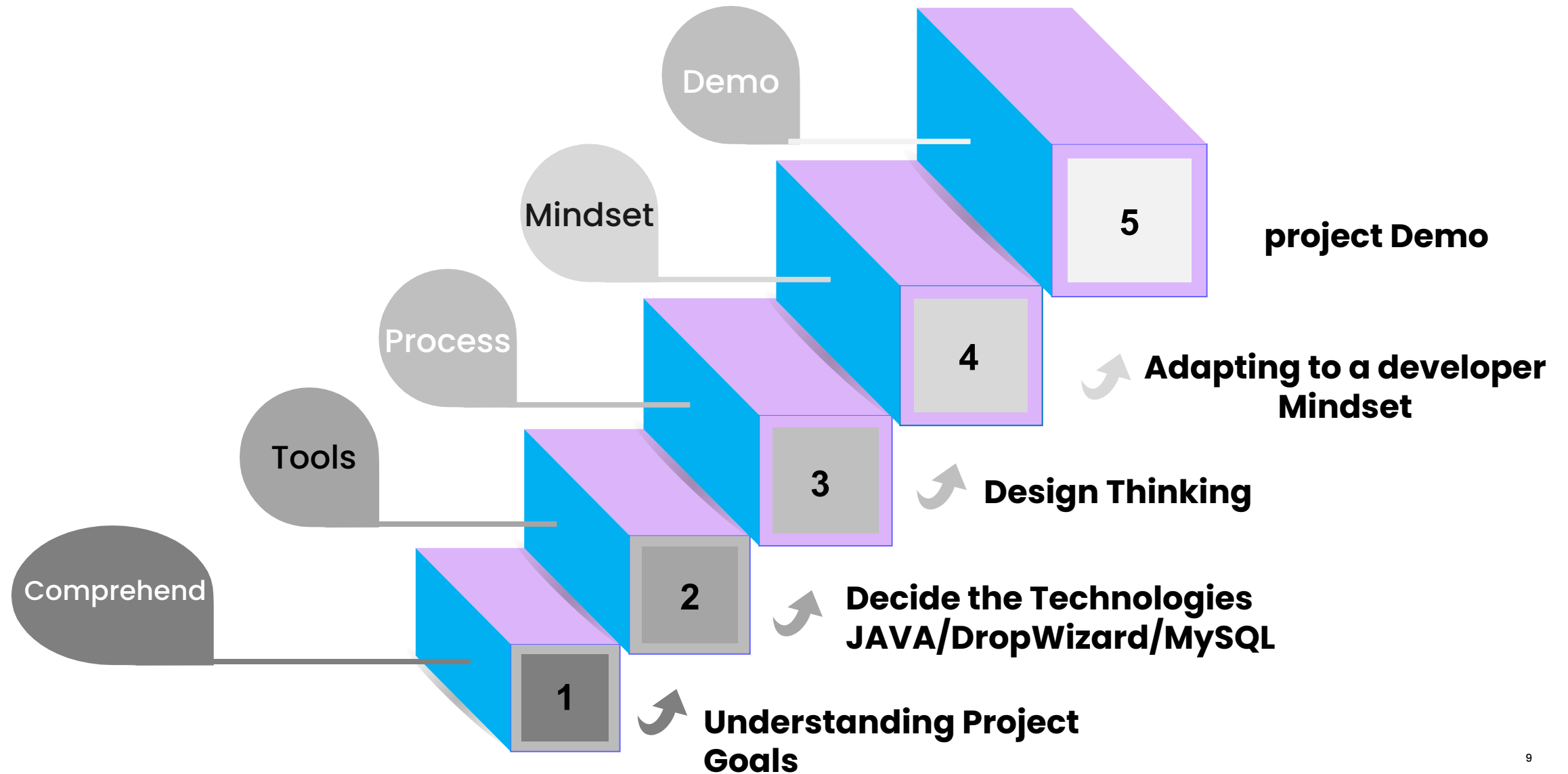
Agenda

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



Our Journey





Our Team



GROUP-D



**Amisha
Kumari**

Team Lead



**Tanish
Chugh**

Mr. Smarty



**Sravyasri
Mortha**

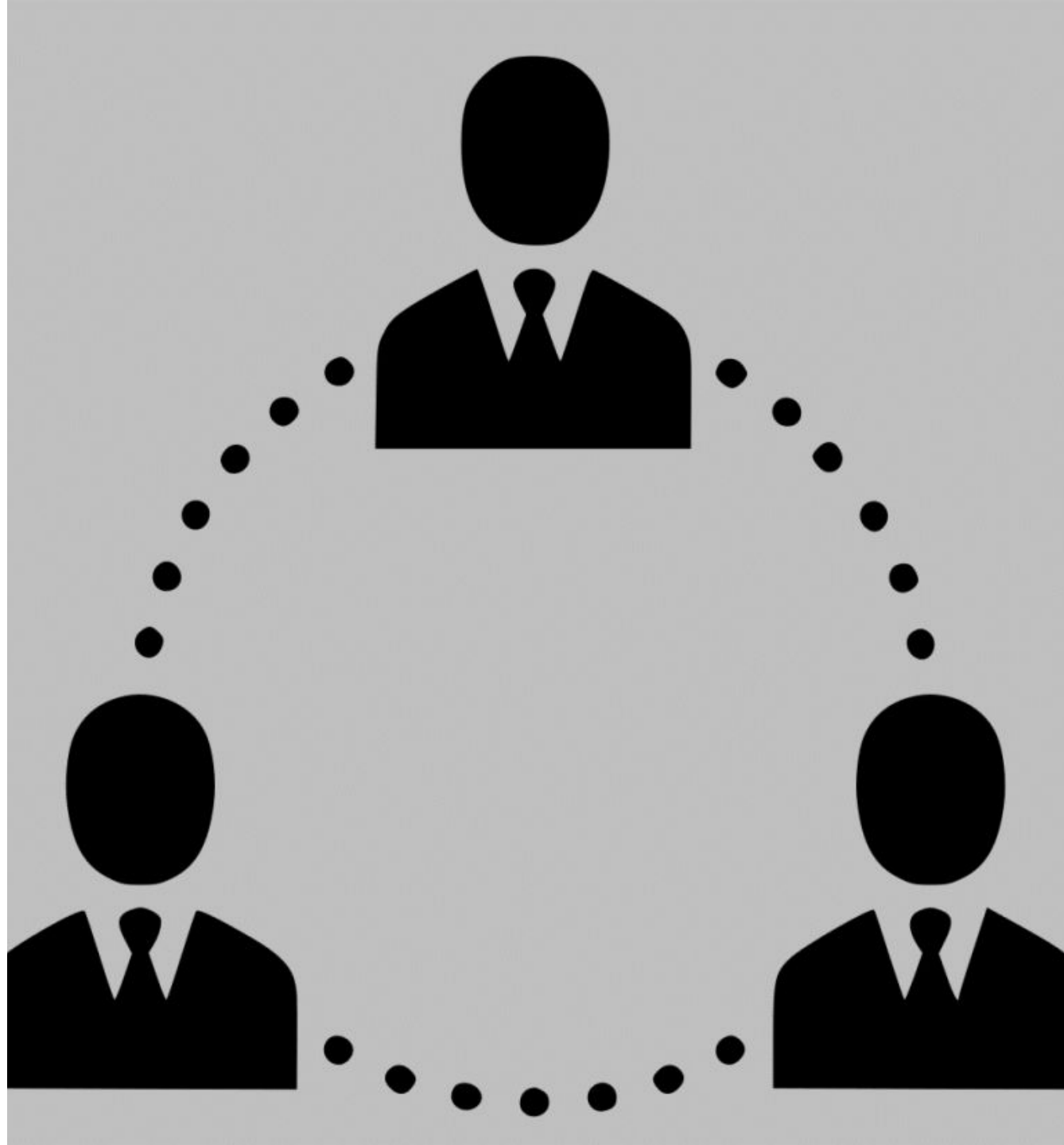
Ms. Dedicated



**Podugu
Deepika**

Ms. Hardworking

Team Structure

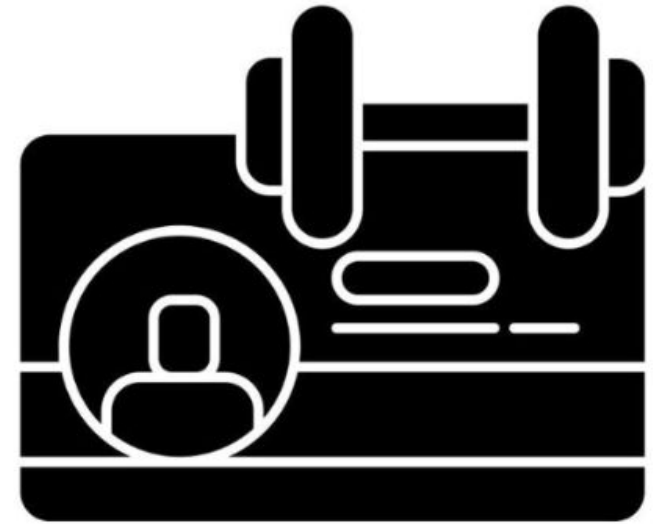


Project Goals

```
=====
Welcome To FlipFit
=====
```

Type:

- 1 -> Login
- 2 -> Registration of Customer
- 3 -> Registration of Gym Owner
- 4 -> Exit



Our Vision



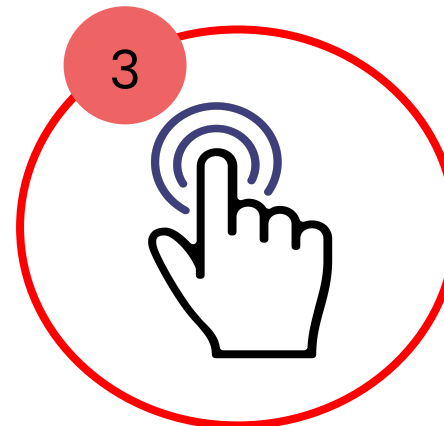
Our objective was to design a user satisfied FlipFit, application for Flipkart's fitness venture. The core challenge was to enable users to register, view gym availability (specifically for multiple centers and fixed-capacity hourly slots), and efficiently book workout sessions, all while ensuring accurate booking management and preventing double bookings.



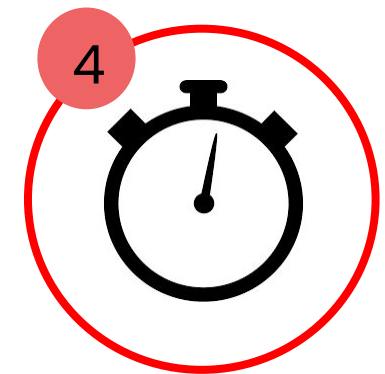
Quality



Security



Interactivity



Speed

Timeline for 1 Week

Day 1

- Install necessary tools and establish team roles.
- Discuss problem statement, goals, and solutions.
- Plan project milestones and deliverables.

Day 2

- Introduction to Git: Setup repository, branching, merging.
- Low-Level Design (LLD): Use Case, Class, Activity diagrams.
- Define application structure: Bean, Business, Client packages.

Day 3

- Explore JDK 17 features: Sealed classes, pattern matching.
- Divide application into packages: Bean, Business, Client.
- Start coding foundational components.

Day 4

- Connect MySQL database to FlipFit App.
- Implement DAO for database interaction.
- Develop business logic and login functionality.

Day 5

- Proceeded to MySql server and Workbench.
- Begin web-based application development.
- Improved the business logics.

Day 6

- Explored and implemented DropWizard.
- Prepared presentation.
- Explored High Level Design and its applications in web technologies.

Engineering Practices

1. Version Control (Git/GitHub): Used Git for version control, feature branching, and regular commits.

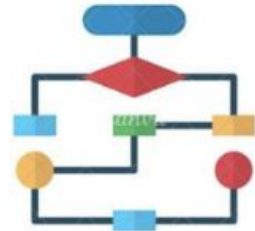
- Benefits: Facilitates collaboration, tracks changes, and enables easy rollback.

2. UML Diagrams: Standardized on UML for system design, ensure clarity in diagrams, and use tools for consistency.

- Benefits: Aligns team understanding, supports documentation, and aids in communicating design.

3. Design Principles (SOLID): Applied SOLID principles (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) for robust, maintainable code.

- Benefits: Improves code quality, scalability, and facilitates easier maintenance and extension.



4. Business Modules and Packaging: Organized code into cohesive modules aligned with business domains, use appropriate packaging structures.

- Benefits: Enhances modularity, reduces dependencies, and improves code organization and reusability.

5. Documentation and Knowledge Sharing: Maintained comprehensive documentation and Conducted internal team meetings before working on any feature

- Benefits: promotes knowledge sharing, and serves as a reference for future development and maintenance.

Tech Stack

Backend

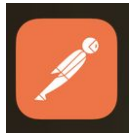
Core Language



Development
Tools And Testing



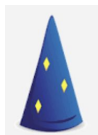
IntelliJ Idea



Postman



Git



Dropwizard

Data

SQL Database



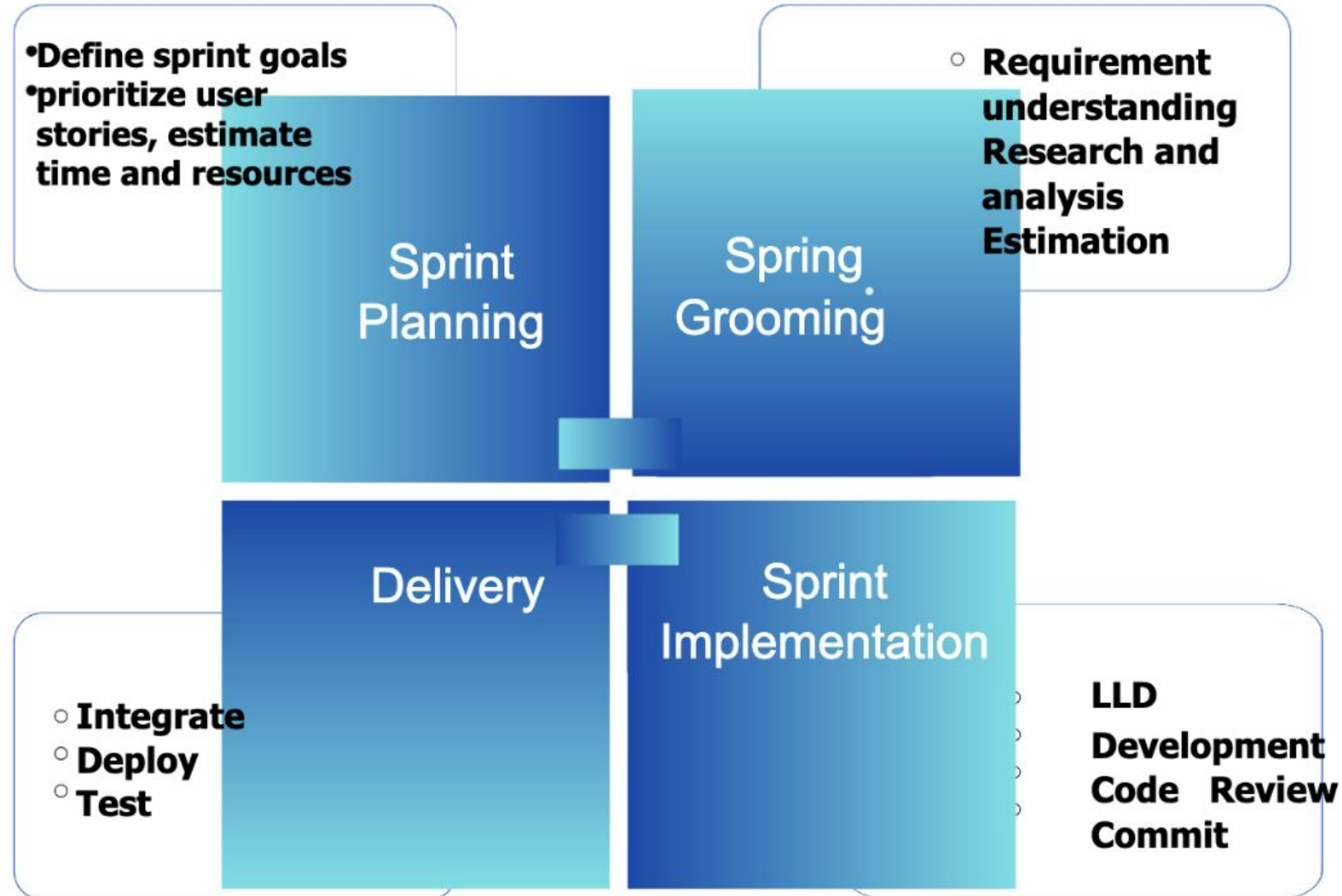
Administration
tool



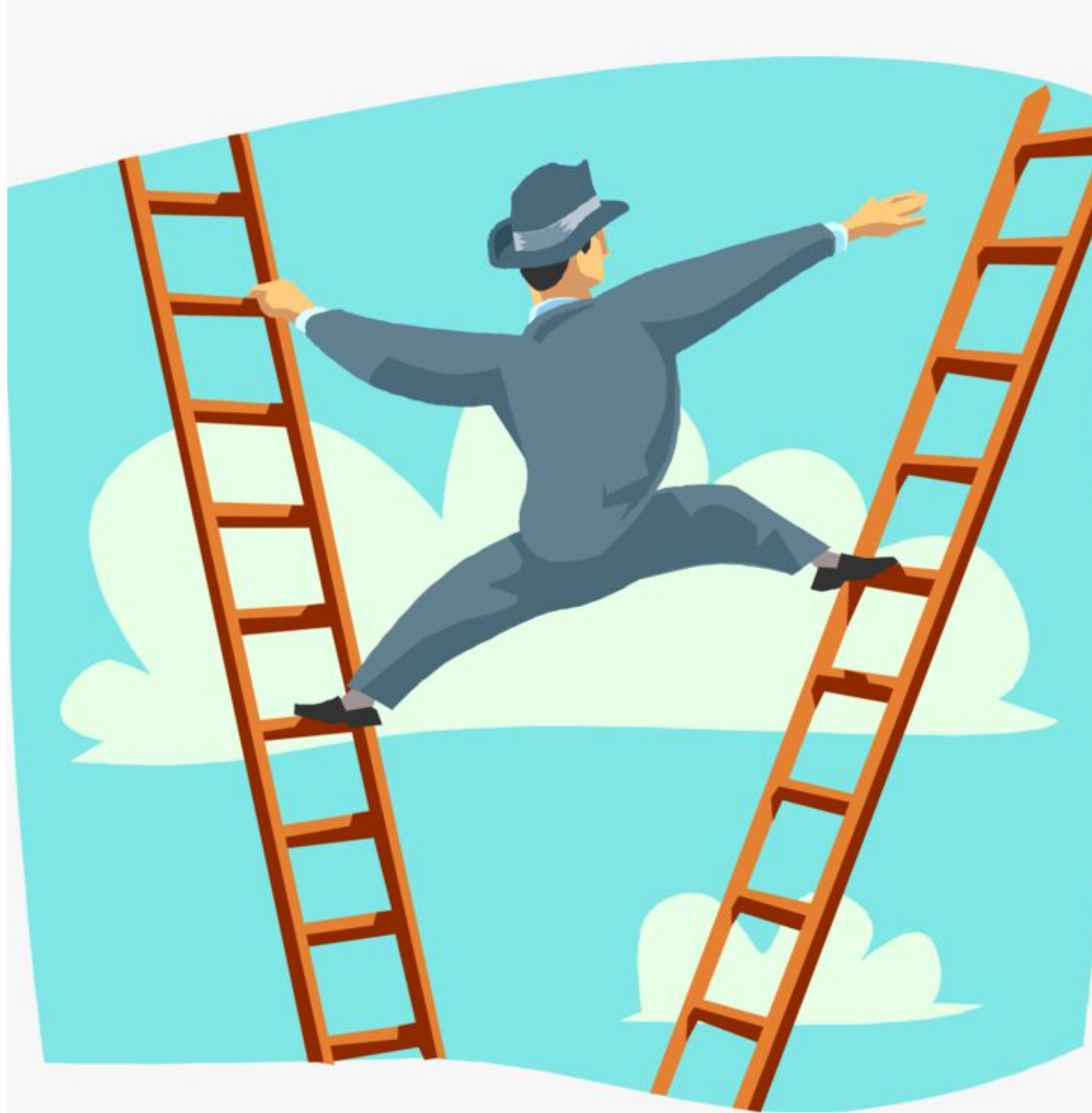
MySQL workbench



Development



Challenges & Learnings



UML Diagram - LLD Structure

Learning:

- Enhanced understanding of software design principles, standard notation usage
- Clear Communication
- Aligning diagrams with business module requirements
- Overview Java Language
- Working with DropWizard to create REST application
- Improved learning of Git/Github
- Debugged together

Challenges:

- Collaborating in a diverse team of seven with varying educational backgrounds, aligning on a single process flow and holistic system view.
- Configuring MySQL
- Achieving tight deadlines

Demo



Questions





Thank you