



LARANA, INC.

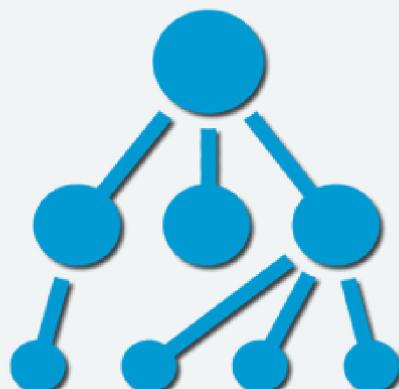
SOCIAL MEDIA APPLICATION

CONTENT

- 01** SAGIMBAYEV NURBEK
- 02** AUYEZKHANOV ALAN
- 03** TUMABAYEV ALI
- 04** OMAROV ALISHER

INTRODUCTION

- Our social media application aims to provide a user-friendly platform for people to connect, share, and stay updated with their friends, family, and communities.
- Key features include creating and sharing posts, following other users, receiving updates in a personalized news feed, and interacting with posts through likes, comments, and shares.
- The application is developed using Java and follows an object-oriented approach, incorporating various design patterns for modularity, extensibility, and maintainability.



BUSINESS PLAN AND PROJECT GOALS

- **Target Market and Audience:** Our primary target market is individuals aged 18-35 who are active on social media platforms and seek a more personalized and engaging experience.
- **Competitive Landscape:** While there are well-established players like Facebook, Twitter, and Instagram, our application aims to differentiate itself by offering a more tailored and community-driven experience, with a focus on privacy and user control.
- **Unique Value Proposition:** Our application emphasizes user privacy, customizability, and community-building features, allowing users to create and join niche interest groups, and share content in a more focused and relevant manner.
- **Objectives and Success Metrics:** Our key objectives are to acquire a user base of 100,000 active users within the first year, achieve a high user engagement rate (measured by daily active users and time spent on the app), and establish a strong brand presence in the social media space. Success will be measured through user acquisition, engagement, and retention metrics, as well as revenue generation through premium features and targeted advertising.

ARCHITECTURE AND DESIGN PATTERNS

- **Architectural Overview:** Our application follows a microservices architecture, with separate services for user management, post management, and notifications. This approach promotes scalability, modularity, and independent deployment of each service.

Design Patterns:

- **Observer Pattern:** Used to notify followers when a user creates a new post. The `NewsFeedObserver` class implements the `Observer` interface and updates the user's news feed when notified by the `PostSubject` class.
- **Facade Pattern:** The `SocialMediaFacade` class provides a simplified interface to the underlying services (`UserService`, `PostService`, and `NotificationService`), making it easier to interact with the application's core functionality.
- **Factory Method Pattern:** Used to create different types of posts (e.g., text posts, image posts) through the `PostFactory` interface and concrete implementations like `TextPostFactory` and `ImagePostFactory`.

MAIN COMPONENTS

User Management

User Class:

- Represents a user in the application, with properties like name, email, password, news feed, followers, and following lists.
- Methods:
- **addPostToNewsFeed(Post post)**: Adds a post to the user's news feed.
- **follow(User user)**: Allows the current user to follow another user.
- **unfollow(User user)**: Allows the current user to unfollow another user.
- Getters and setters for user properties.

User Authentication:

- The AuthenticationService class handles user registration and authentication.
- Users need to be authenticated before performing any actions in the application.

MAIN COMPONENTS

Post Management

Post Class:

- Represents a post in the application, with properties like content, author, and timestamp.
- Methods:
 - `getContent()`: Returns the post's content.
 - `getAuthor()`: Returns the post's author (a User object).
 - `getTimestamp()`: Returns the post's creation timestamp.
 - Getters and setters for post properties.

Creating Posts:

- Users can create different types of posts, such as text posts or image posts.
- The PostFactory interface and its concrete implementations (`TextPostFactory`, `ImagePostFactory`) handle the creation of different post types.

Displaying News Feed:

- The user's news feed is a collection of posts from the users they are following.
- The NewsFeedObserver class implements the Observer interface and updates the user's news feed when a new post is created.

MAIN COMPONENTS

Notification Service

- **Notifying Followers:**
 - When a user creates a new post, their followers are notified about the new post.
 - The NotificationService class contains the notifyFollowers(User author, Post newPost) method, which iterates over the author's followers and sends them a notification about the new post.
 - This component integrates with the User and Post classes to access the necessary data.

MAIN COMPONENTS

Facade

- **Simplifying the Application's Interface:**
 - The SocialMediaFacade class acts as a facade, providing a simplified interface to the underlying services (UserService, PostService, and NotificationService).
 - Method:
 - **createPost(User author, String content):** Allows creating a new post by providing the author and content. This method internally creates the post, notifies followers, and handles any other necessary operations.

FUTURE ENHANCEMENTS

Integration with Databases or External Services:

- Currently, the application stores data in-memory. Integrating with a relational database (e.g., MySQL, PostgreSQL) or a NoSQL database (e.g., MongoDB) would allow for persistent data storage and better scalability.
- Integrating with external services like cloud storage (e.g., Amazon S3) would enable storing and serving media files like images and videos more efficiently.

User Interface Development:

- The current implementation focuses on the backend logic and data structures. Developing a user-friendly and responsive user interface (e.g., web, mobile app) would greatly enhance the user experience and adoption of the application.

Performance Optimizations:

- As the user base and data grow, performance optimizations may be required, such as caching frequently accessed data, implementing load balancing, and optimizing database queries.

Additional Features:

- Implementing features like commenting on posts, liking posts, sharing posts, creating groups or communities, and real-time updates could further enhance the application's functionality and user engagement.

CONCLUSION

- Our social media application aims to provide a personalized and engaging experience for users by leveraging modern architectural patterns and design principles.
- Key strengths include user privacy, customizability, community-building features, and a modular and scalable architecture.
- With a strong business plan, a focus on user engagement, and a roadmap for future enhancements, our application has the potential to become a successful player in the social media space.
- [Thank the audience and open the floor for questions]

This detailed presentation covers the introduction, business plan, architecture, main components, code examples, future enhancements, and a conclusion. It provides a comprehensive overview of the social media application, its goals, technical implementation, and potential for growth and improvement.

**THANK'S FOR
ATTENTION**

