User Guide: Capabilities and Purpose of the Application

1. Overview of the Application

The application is a comprehensive library management system crafted to simplify and enhance the operations of library management. Designed with an intuitive graphical user interface (GUI) built using JavaFX, it provides users with an efficient platform to manage books, users, and related activities. The backend is powered by PostgreSQL, ensuring a robust and reliable database management system. By addressing the core needs of libraries, the application helps reduce administrative overhead and offers tools to maintain accurate and accessible records.

The purpose of this system is to bridge the gap between traditional manual processes and modern automated solutions. By offering real-time interactions with the database, it facilitates seamless operations, allowing librarians and administrators to focus on enhancing the user experience rather than managing operational complexities.

---

2. Features and Functionalities

User Authentication

The application includes a robust user authentication mechanism. This feature validates the credentials of individuals attempting to access the system. Passwords are securely stored using Argon2 hashing, which ensures a high level of security against unauthorized access. Users are required to enter a valid email and password combination, and the system gracefully handles errors such as incorrect credentials or failed login attempts.

Book Management

Librarians can efficiently manage book records through an intuitive interface. The book management functionality includes:

- Adding new book entries to the database, ensuring all essential details are captured.
- Editing existing book records, such as titles, authors, genres, and ISBN numbers.
- Removing outdated or incorrect records.
- Searching for books based on criteria like title, author, or genre, reducing the time spent on manual catalog navigation.

User Management

Administrators have access to advanced user management features. These include:

- Adding new users to the system and assigning them appropriate roles.
- Editing user profiles to update contact information or access levels.
- Removing inactive or unauthorized user accounts.
- Viewing detailed user activity logs to ensure transparency and security.

Reporting and Logs

The application generates detailed reports on library inventory and user activity. Logs capture critical actions such as login attempts, book modifications, and data exports. These logs provide an audit trail for administrators and help in diagnosing issues effectively.

---

3. Technical Framework and Architecture

The application is built with a modular architecture, leveraging modern technologies for enhanced performance and scalability.

Frontend

- Built using JavaFX, the frontend provides a responsive and user-friendly interface. It uses FXML files to define layouts and ensures a clear separation between design and logic.

Backend

- The backend is implemented in Java, following a layered architecture. Controllers link the user interface with business logic, while services like AuthService and LibService manage core operations. Repositories, such as LibRepository, handle all interactions with the PostgreSQL database.

Security

- Passwords are securely hashed using Argon2, providing resilience against brute-force attacks.
- Input validation mechanisms safeguard against SQL injection and unauthorized access.
- Centralized exception handling ensures consistent error management and improves user experience.

---

4. Steps to Get Started

System Requirements

- Java 17 or higher.
- PostgreSQL server running on localhost with the database.
- Maven for managing dependencies and building the project.

Setup Instructions

1. Clone the repository from the provided GitHub link.
2. Open the project in IntelliJ IDEA.
3. Update the application.properties file with PostgreSQL credentials.
4. Build the project using Maven.
5. Run the Main class to launch the application.

Initial Configuration

Ensure that user credentials exist in the database. If not, insert sample data into the bds.person table using Argon2-generated password hashes. This ensures seamless access during the first run.

---

5. Addressing Common Issues

Login Failures

- Verify that the PostgreSQL service is running and accessible.
- Check if the email and password exist in the database.
- Review the log file (logs/bds-java-logs.log) for detailed error messages.

Database Connectivity Problems

- Confirm the database URL, username, and password in application.properties.
- Ensure no network restrictions are preventing access to the database.

UI Bugs

- Review FXML files to ensure proper binding with controllers.
- Check the associated stylesheets (modena.css or myStyle.css) for missing or broken elements.

---

6. Additional Features and Workflow

User Activity Tracking

The application provides robust tracking of user activities within the system. This feature records key actions such as book searches, edits, and deletions, ensuring full transparency. Administrators can access these records to review usage patterns or investigate potential misuse.

Custom Notifications

Notifications are an integral part of maintaining smooth operations. Users can receive alerts regarding overdue books, system updates, or changes in their account status. Notifications can be configured for email or displayed directly in the user interface.

Data Backup and Restoration

The system supports regular data backup to prevent loss of information. Administrators can schedule backups and initiate restorations using a dedicated management panel. These features ensure business continuity in case of system failures.

Advanced Search Filters

The search functionality allows users to filter results by multiple parameters such as publication date, language, and availability. These filters streamline the process of finding specific records in large databases.

---

7. Potential Enhancements for Future Versions

Mobile Support

Extending the application to mobile platforms (Android and iOS) would broaden its accessibility.

Advanced Analytics

Incorporating visualization tools and predictive analytics could provide deeper insights into library operations.

Localization

Adding support for multiple languages would make the application accessible to a global audience.

Cloud Integration

Cloud-based storage would enhance scalability and facilitate remote access for distributed teams.

---

8. Conclusion

This library management system offers a secure, efficient, and user-friendly solution for managing books and users. With its intuitive interface, robust backend, and extensive features, it addresses the needs of modern libraries effectively. The modular design ensures scalability, making it adaptable for future enhancements such as mobile integration and advanced analytics.

Overall, the system empowers library staff and administrators with tools to improve efficiency and foster better service delivery. Whether managing thousands of books or catering to a diverse user base, the application stands as a reliable ally in modern library management.