BAHIRDAR UNIVERSITY

Bahir Dar Institute of Technology

Faculty of Computing

Departments of Software Engineering

PROJECT TITLE: **Museum Management system**

1. TEKLEEYESUS MUNYE	1404492
2. WENDOSEN GETANHE	1404092
3. WUBAMLAK GIRUM	1404373
4. TEMESGEN BIRHANU	1405788
5. YOSEPH BIRHANU	1404237
6. ZELALEM FISEHA	1405490
7. MULUGETA AGAFARI	1404409

Submitted to: Mr. Hagos T.

BAHIR DAR, ETHIOPIA

Table Content

1. Introduction

The Museum Management System serves as a comprehensive solution for museum organizations to efficiently manage their collections, streamline administrative tasks, and enhance visitor experiences. By leveraging modern technologies such as JavaFX and FXML, this system empowers museum staff to perform the following key functions:

1. **Item Management:**
   Curators can seamlessly add, update, and delete items within the museum's collection.
   Detailed information about each item, including its id, name, and exhibition history,description can be recorded and accessed.
2. **Staff Management:**
   Admins maintain staff profiles, ensuring accurate records of staff data, details, and contact information.
   Staff members can collaborate effectively, improving overall museum operations.
3. **Visitor Management:**
   The system captures visitor data, including entry times, contact info, and Payment.
   Curators and administrators gain insights into visitor demographics and preferences.
4. **User Authentication and Authorization:**
   A secure login system ensures that only authorized personnel can access sensitive information.
   User roles (admin, curator, visitor) dictate permissions and functionalities.
5. **Registration and Login Page:**
   New users can register their accounts, providing essential details.
   The login page authenticates users and grants access to relevant features.

2. System Architecture

2.1 High-Level Architecture

The museum management system, built with JavaFX, includes a user-friendly interface, data model, and business logic. It handles collections, visitors, and administrative tasks while ensuring security and integration with external systems."Highlight the separation of concerns (e.g., UI, business logic, data storage.

2.2 Technology Stack

List of technologies and frameworks used:

- ❖ Frontend: JavaFX, FXML
- ❖ Backend: Java (or any other language)
- ❖ Database:XAMPP(mysql)

3. User Roles and Permissions

3.1 Admin

Responsibilities: Managing users, system settings, and staff profiles.
Permissions: Full access to all functionalities.

3.2 Curator

Responsibilities: Item management (add, update, delete).
Permissions: Limited to item-related actions.

3.3 Visitor

Permissions: Read-only access.

4. User Interface (UI) Details

4.1 Login Page

The login page for your museum management system features a straightforward interface. Users can log in using their username and password. Additionally, there’s a prominent register button for new users to create an account. Upon clicking the register button, users can provide necessary details to set up their museum-related account.

4.2 Dashboard

Your museum management system’s dashboard provides essential information for curators. It includes the total income, staff count, visitor summary, and total items in the collection. The total income encompasses revenue from various sources such as ticket sales, memberships, and donations. The staff count reflects the current workforce, including curators, administrative staff, security personnel, and maintenance crew. The visitor summary highlights the overall footfall, with comparisons to previous periods. Lastly, the total items summary quantifies the museum’s diverse collection, which may span paintings, sculptures, historical artifacts, and more.

4.3 Manage Item Page

- **Addition of Items:**

  Curators and administrators can input details for new items, such as title, description, category, and acquisition date.
  The system validates input and assigns a unique identifier (e.g., accession number) to each item.

- **Updating Existing Items:**
  Users can modify item information, including descriptions, ownership history, and associated metadata.
  Changes trigger updates across relevant records (e.g., exhibition catalogs, visitor guides).
- **Deletion of Items:**
  Authorized personnel can remove items from the collection.
  The system ensures data integrity by handling dependencies (e.g., removing references in loan records).
- **Search and Filter Functionality:**
  A robust search interface allows users to find specific items based on keywords, categories, or other criteria.
  Filters enable targeted views (e.g., displaying only paintings or artifacts from a particular era).

4.4 Manage Visitor page

- **Addition of Visitors**:
  Curators and administrators can input visitor details, including names, contact information, and purpose of visit.
  The system records the date and time of each visitor entry.

- **Updating Existing Visitor Records:**
  Authorized users can modify visitor information (e.g., update contact details or reason for the visit).
  Changes are reflected across all relevant records.

- **Deletion of Visitor Records:**

  Staff can remove visitor entries (e.g., when a visitor checks out).
  The system ensures data consistency and handles dependencies.
- **Search and Filter Functionality:**
  Implement robust search capabilities to find specific visitors based on names, dates, or other criteria.
  Filters allow targeted views (e.g., displaying current visitors only).

4.5 Manage Staff Page

- **Adding Staff Members:**
  Administrators can input details such as names, roles, contact information, and employment start dates.
  The system assigns a unique staff ID to each new entry.
- **Updating Staff Records:**
  Authorized users can modify existing staff information (e.g., job titles, phone numbers, or work schedules).
  Changes are reflected across relevant records.
- **Deleting Staff Entries:**
  When staff members leave or need removal, administrators can delete their records.
  The system ensures data consistency and handles dependencies (e.g., payroll records).
- **Search and Filtering:**
  Implement robust search functionality to find specific staff members based on names, roles, or other criteria.
  Filters allow focused views (e.g., displaying only active staff or specific departments).

4.6 .Profile page

- **Default Profile Page:**
  The default profile page displays user information such as name, contact details, and profile picture.Users can view their existing data but cannot edit it directly.
- **Custom Profile Page (Edit):**
  A customized profile page allows users to edit their details.

5. User Registration Workflow

1. **User Initiation:**
   A new user accesses the registration page.
   They click on the “Register” button.
2. **Input Details:**
   The user provides necessary information, such as name, email address, username and password.The system validates input (e.g., checks if the email format is correct).
   If there are errors (e.g., duplicate email), appropriate error messages are displayed.

3. **Account Creation:**
   Upon successful validation, the system creates a user account.

4. **Confirmation and Welcome:**
   The user receives a confirmation alert message of register completion and the login page displays.They can now log in using their credentials.

6. Requirements

The system relies on Java for its backend and JavaFX for the frontend.
Ensure that the following JDK specifications are met:
JDK Version: JDK 21
JavaFX: Verify that JavaFX is included in your JDK distribution.

Software Dependencies :
Ensure the following software components are installed:
Java IDE: Use an Integrated Development Environment (IDE) NetBeans.

JavaFX SDK: Install the JavaFX SDK if it's not bundled with your JDK.

Project Configuration steps:

1st Step: Extract file
2nd Step: open "class_project" javafx project in IDE
3rd Step: Make sure JDK 21 is Available in system

Now Connecting Database

4th Step: Open a browser and go to URL "http://localhost/phpmyadmin/"
5th Step: Then, click on databases tab
6th Step: Create database naming "museumdb" and then click on import tab
7th Step: Click on browse file and select "museumdb.sql" file which is inside main folder
8th Step: Click on go.

After Creating Database,

9th Step: Build and Run the project
Insert the username='admin' and password ="admin" to login to system or Register and create new Account.