

**Object Oriented Programming**

**Mini Project Report on**

**Cattle Management System**

**SUBMITTED**

**BY**

Adi Narayan R. S. 33 230905238

Gautham Binod 32 230905235

Renet Susil Johns Jacob 19 230905141

Section B

**Under the Guidance of:**

**Mrs. Sujithra T**

**Department of Computer Science and Engineering**

**Manipal Institute of Technology, Manipal, Karnataka – 576104**

October 2024

**Introduction**

The **Cattle Management System** is a robust digital solution designed to modernize and optimize livestock management for dairy farms and cattle ranches. Built with JavaFX, this application meets the essential need for streamlined herd management, health tracking, and production monitoring in the agricultural industry.

In today’s fast-paced farming environment, managing cattle farms with traditional pen-and-paper methods is inefficient and error-prone. Our primary aim was to simplify farmers' work by providing an easy-to-use software solution to manage all their cattle-related tasks.

**Problem Statement**

Key Objectives:

1. Farmer Registration and Login
2. Cattle Registration
3. Insurance Policies and Premium Management
4. Insurance Claim Process
5. Claim review and Approval process
6. Cattle Ownership and Transfer Tracking
7. Audit Logging for Insurance Claim

**Implementation Details**

1. Class Structure & Design Pattern:

The core component includes singleton pattern implementation for data management, scene management which allows switching between different scenes and user authentication which implements role-based access control.

The key features contains multiuser login support, data and scene management.

1. Creating login scene

The first entry point of the application is the login page that contains the user name and password fields. If the user hasn’t previously registered, the user can click the “New User register” Button which will show a new page where the user can register his name, password and his farming location.

Users enjoy a streamlined experience starting from the initial farmer login tab, which features a clean, organized interface with direct access to input fields, clear information description. In addition to this, the admin can view the details of everyone who has registered by logging into the application using the username as “Admin” and password as “AdminCSEB”.

1. Cattle management

This process consists of a registration form and a cattle view table. The registration form collects cattle information and validates input data. It also organises form layout.

The cattle table view displays the Cattle age, breed, weight and vaccination history.

Every cattle has a unique ID that is generated and stored in the system

Overall, the features include input validation, data processing, and success or error notifications. It also provides editing capabilities to update details of existing cattle and view the updated information.

1. Insurance claiming process

We have listed the following policies:

* New India Assurance
* United India Insurance
* Oriental Insurance
* Livestock Insurance Scheme (LIS)
* Integrated Rural Development Program (IRDP) Insurance
* National Livestock Mission
* Rashtriya Gokul Mission insurance provisions
* Dairy Entrepreneurship Development Scheme

Proper information is also given right next to the checkboxes to help the user claim the right insurance for their need.Based on this, the insurance premium is calculated

1. Creation of Audit Log

The key features of the audit log include:

* Formatted timestamps to show when activity has occurred in real time along with the type of action that has occurred
* Manual refresh button to refresh the page according to the user’s convenience
* An organized layout with clear separation between the header and the content
* Sortable columns which the user can sort according to their convenience

There also separate tabs for insurance policies and policy management in which the user can check the type of insurance they have opted for and their respective policies.

1. Creation of Admin Scene

This interface provides a centralised admin control panel with easy access to all administrative functions. It also has access to the Audit tab to keep tabs on users who have used the application at certain timestamps.

The logout button is always visible at the top and it returns the user to the login screen. For security purposes, the admin flag is reset on logout.

For the admin to manage the insurance claim content, a claim display table is created which displays all the claims in a structured manner. The table has sortable columns allows selection capability to update the claims. It also includes automatic date formatting.

This tab also contains the encrypt and decrypt functions which converts all the passwords to SHA-256 and stores it in an array. The decrypt function returns the user name, password and farm location of all users upon invocation.

For the admins to manage the farmers content, the page includes a layout with a list of all farmers, and they can also simultaneously view a particular famers cattle detail. They also have the cattle’s ownership controls. To validate the farmer-cattle information, the admin must select the cattle and the new owner. The new owner cannot be the same as the old owner. An automatic feedback message is generated after this process is completed.

1. Generation of checksum

The checksum is generated by combining critical data fields from a cattle record, including the unique identifier (UID), farmer ID, breed, age, weight, vaccination records, and insurance policy ID. This composite string is passed to a specific method, which applies a hashing algorithm to generate a unique checksum for the record.

1. Checksum Creation:

Each time a new cattle record is created or updated, the Checksum() method is called. This method then concatenates with another function called showWelcomeScene() which contains essential data fields, ensuring that any modification to these fields will result in a different checksum value. The hash generated by PasswordHashing.java acts as a digital fingerprint for the data’s current state.

2. Checksum Verification:

To confirm the data’s integrity, the system utilizes the checkSum() method. This method recalculates the checksum based on the current field values and compares it with the previously stored checksum. If the recalculated checksum matches the stored one, it indicates that the data has remained unchanged; otherwise, a discrepancy signals possible data corruption or unauthorized modification.

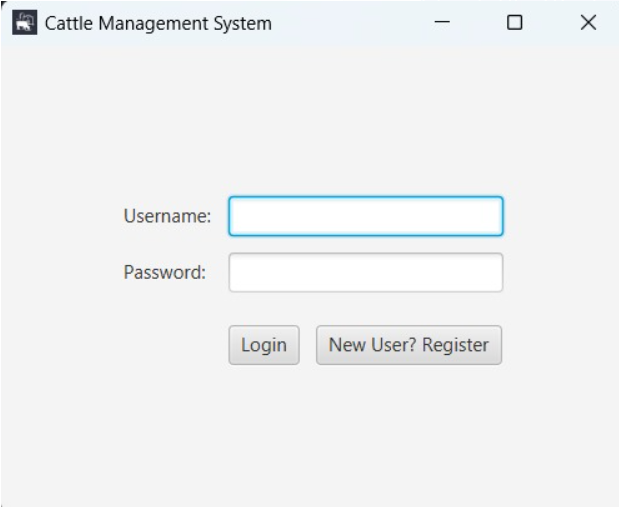
Benefits of checksum include:

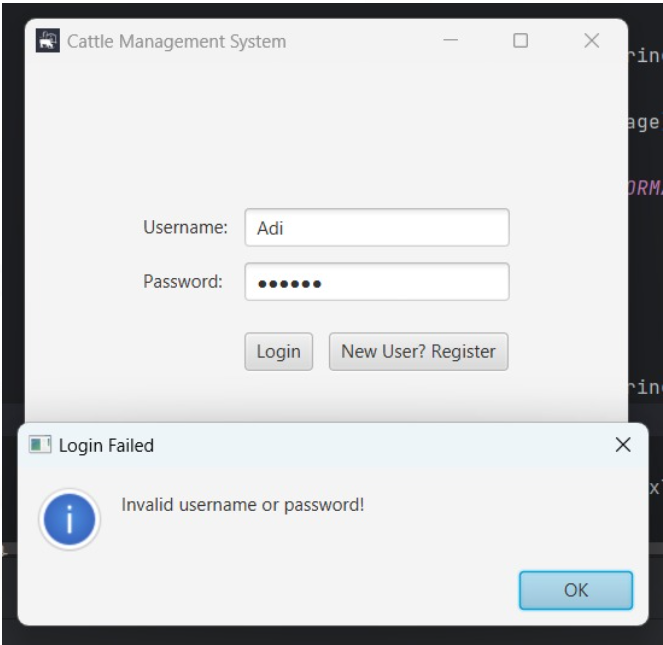
1. Tamper Detection: Any unauthorized alterations to critical fields will generate a different checksum, allowing the system to detect tampering or accidental changes promptly.

2. Data Consistency: The system ensures that stored records remain consistent, particularly important when tracking animal ownership, updating health records, or managing insurance claims.

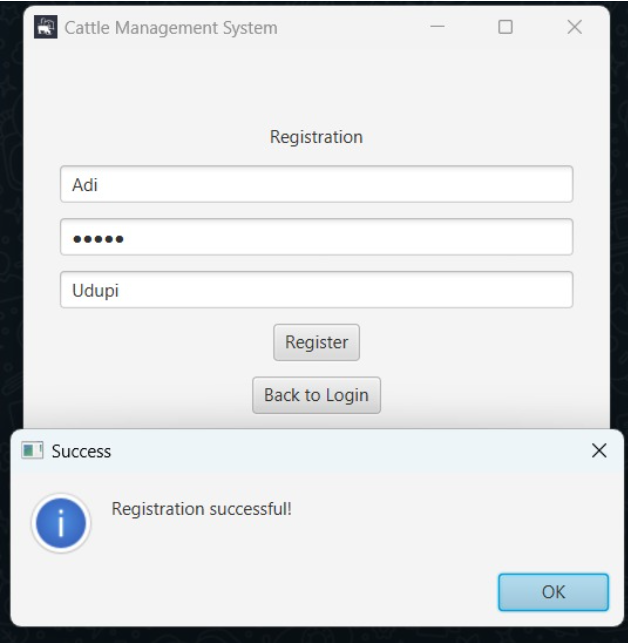
**Results**

Here are the output results we have received on running our JavaFX program:

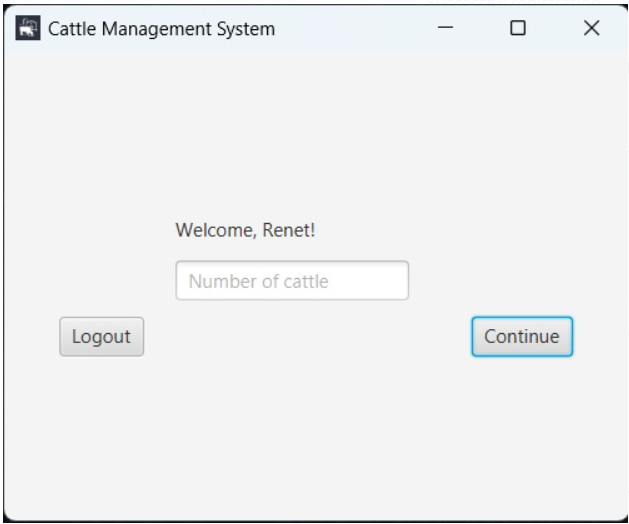




Login without Registration Error message



Successful registration

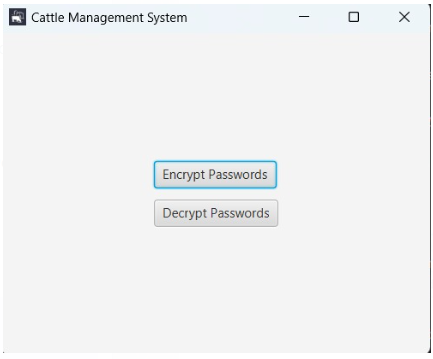


Upon login

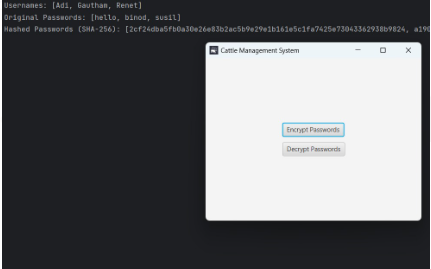
A screenshot of a computer

Description automatically generated

Cattle registration



Encryption and Decryption Buttons

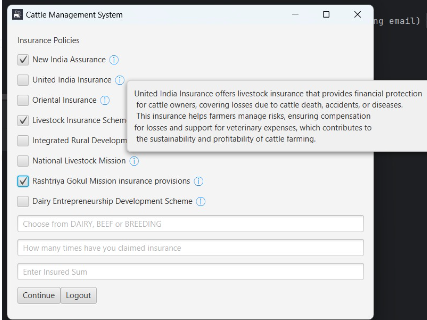


Encryption

A screenshot of a computer

Description automatically generated

Decryption



Insurance Policy Claim desk and Premium Calculation tab

**Conclusion**

The JavaFX-based Cattle Management System marks a major step forward in agricultural resource management technology. This system effectively tackles the primary challenges of modern cattle farming while offering a reliable, user-friendly interface for daily operations.

**References**

* Geek For Geeks
* Oracle Help Centre
* Free Code Camp
* Bro Code