

Core System Functions

main(String[] args)

Entry point of the application

- Loads user data from file
- Creates default admin if none exists
- Displays main menu in a loop
- Handles user selection (login/register/exit)

loadUserData()

Loads all users from text file into memory

File Format:

username|password|fullName|isAdmin|landSize

saveUserData()

Saves all users from memory to text file

Converts each User object to pipe-delimited string and writes to file

Authentication Functions

login()

Authenticates users

- Takes username/password input
- Checks if username exists
- Verifies password matches
- Sets currentUser if successful

register()

Creates new farmer accounts

- Checks for duplicate usernames
- Collects password and full name
- Adds new User to HashMap

Farmer Functions

journey title Farmer Function Flow section Profile Management manageLandDetails(): 5: Farmer section Information displayCropPrices(): 4: Farmer viewCropSteps(): 3: Farmer section Subsidies manageSubsidies(): 5: Farmer applyForSubsidy(): 4: Farmer viewMyApplications(): 3: Farmer

manageLandDetails()

Store/update farm information

- Land size (acres)
- Location
- Soil type

displayCropPrices()

Shows current market prices

Formatted table of all crops and prices

viewCropSteps()

Displays growing instructions

Shows step-by-step guides for selected crops

Admin Functions

manageCropPrices()

Manage crop pricing data

- **updateCropPrice()**: Modify existing prices
- **addNewCrop()**: Add new crop to system
- **removeCrop()**: Delete crop from system

manageSubsidyData()

Manage subsidy programs

- **updateSubsidy()**: Edit subsidy details
- **addNewSubsidy()**: Create new program
- **removeSubsidy()**: Delete subsidy

viewAllUsers()

Display all registered farmers

Formatted table with user details

manageApplications()

Process subsidy applications

- **viewAllApplications()**: Shows pending/approved requests
- **updateApplicationStatus()**: Approve/reject applications

Utility Functions

getIntInput(min, max)

Validated number input

```
// Example implementation
private static int getIntInput(int min, int max) {
    while (true) {
        try {
            int input = Integer.parseInt(scanner.nextLine());
            if (input >= min && input <= max) return input;
            System.out.printf("Please enter between %d-%d: ", min, max);
        } catch (NumberFormatException e) {
            System.out.print("Numbers only please: ");
        }
    }
}
```

viewSubsidyDetails()

Show complete subsidy information

Displays detailed description from SUBSIDY_DETAILS array

applyForSubsidy()

Submit subsidy application

- Checks for existing applications
- Requires confirmation
- Updates user records

User Class Methods

classDiagram class User{ +authenticate() +setLandDetails() +addSubsidyApplication() +hasAppliedForSubsidy() +getUsername() +getLandSize() +// other getters }

authenticate()

Password verification

setLandDetails()

Stores farm information

addSubsidyApplication()

Adds to application list

hasAppliedForSubsidy()

Checks for duplicate applications