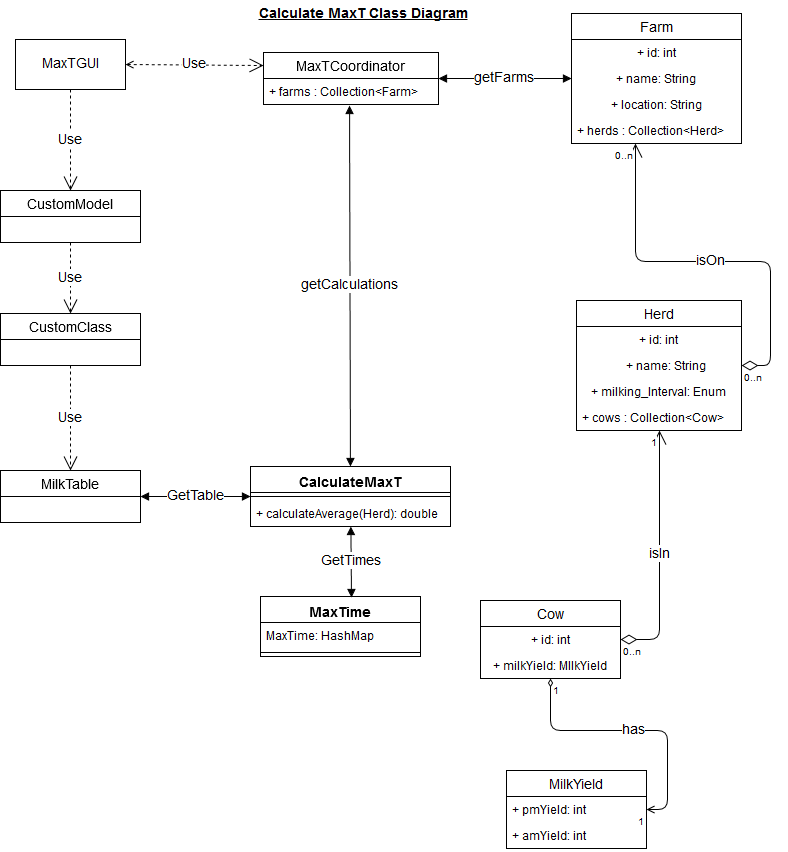
BIT 693 TMA4 Solution Document

# Use Cases

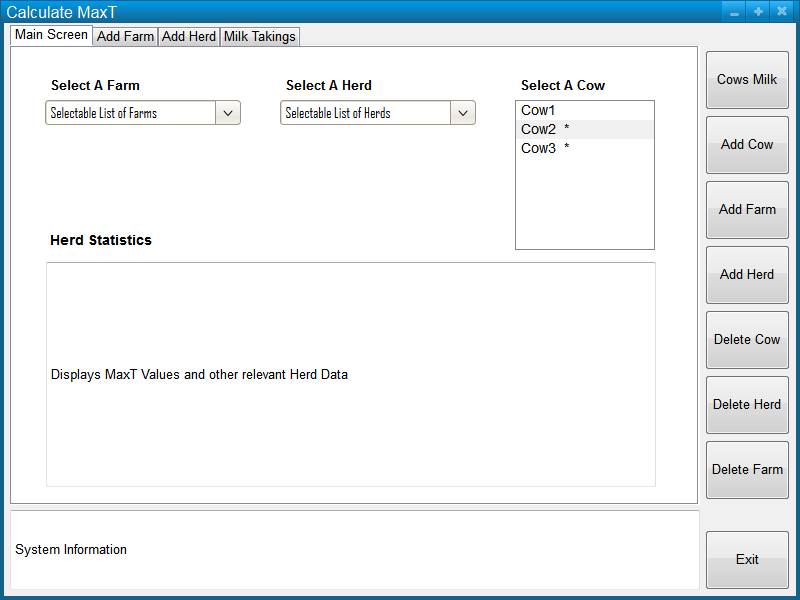
1. **Opens Application.** The user opens the application and is presented with a list of all farms and a list of herds for the first farm on the list with relevant MaxT values and statistics. If no farms exist the user is given an appropriate message. If no herds exist for the first farm on the list the user is given an appropriate message. (*Use case E.*)
2. **Create Farm.** The user enters name and location of farm. The system creates a new Farm with a unique identification number and add to list of farms displayed in the application.
3. **Create Herd.** The user selects a farm that the herd belongs to, enters a name to identify the herd and chooses a milking frequency for the herd. The system creates a new Herd with a unique identification number and adds to the list of herds associated with the farm.
4. **Create Cow.** The user selects a farm and a herd (that belongs to the farm) that the cow will belong to. The system creates a new Cow with a unique identification number and adds to the list of cows associated with the herd.
5. **Select a Farm.** The user selects a farm on the main window. The system presents a list of herds associated with the farm and relevant MaxT values and statistics. If no herds exist the user is given an appropriate message.
6. **Select a Herd.** The user selects a farm (*Use case E.*) then selects a herd from the presented list. The system presents a list of cows associated with the herd. If a cow has current milk taking values associated with it an indicator is displayed next to the cow. If no cows belong to the herd the user is given an appropriate message.
7. **Select a Cow.** The user selects a farm, (*Use case E.*), selects a herd (*Use case F.*), then selects a cow from the list. The system changes the display to a milk taking window presenting the current information held by the system for the selected cow.
8. **Add/Update/Delete Milk Taking.** The user enters or alters the information held for the morning and evening milking. The system stores the relevant data associated with the cow and displays an appropriate message. If the user does not enter all information or correct data, the system does not store the information and displays an appropriate message. If a user deletes the milk taking values the system removes associated values from the cow and displays an appropriate message.
9. **Calculate Values.** When **all** cows in a herd have milk takings associated with them, the system calculates the average milk yield per cow for the herd and determines the MaxT values for the morning and evening milking. This information is displayed on the main window when a user has selected a herd. (*Use case F.*)
10. **Delete Cow.** A user selects a cow (*Use case G.*) and can delete them from the main window (or milk taking window). The system deletes all information associated with the cow and displays an appropriate message.
11. **Delete Herd.** A user selects a Herd (*Use case F.*) and requests to delete. The system checks if there are cows associated with the herd. If there are no cows, the system deletes all information for the herd and displays an appropriate message. If there are cows associated with the herd, the system does not perform any deletion methods and displays an appropriate message to the user.
12. **Delete Farm.** The user selects a farm (*Use case E.*) and requests to delete. The system checks if there are any herds associated with the farm. If there are no herds associated with the farm the system deletes all information for the farm and displays an appropriate message. If there are herds associate with the farm, the system does not perform any deletion methods and displays an appropriate message to the user.

# Class Diagram: Final for stage 3



# Storyboards

## The Main Screen

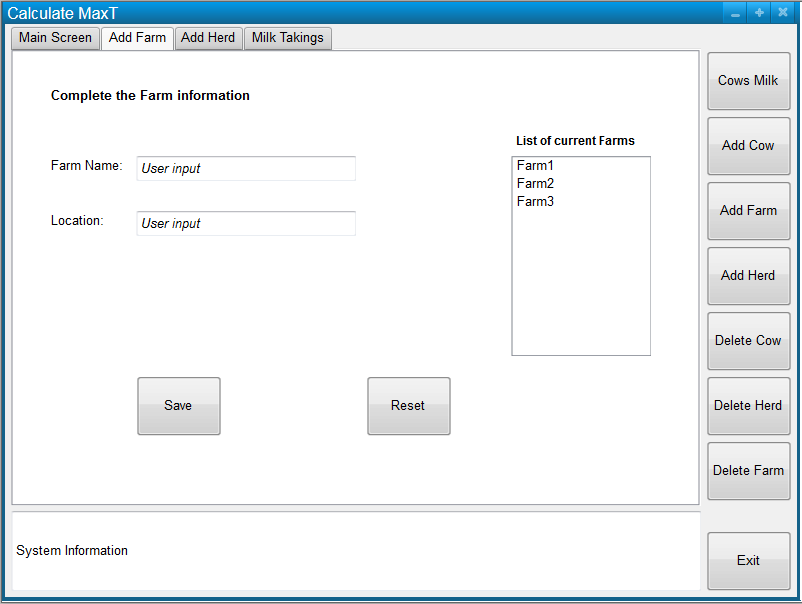


click



* On opening the application, the user is presented with the main screen.
* Tabs and buttons are disabled until the right data is available, e.g. Add Herd is not selectable until a Farm has been created.
* If the system criteria is not met, an error message will be displayed in the system information text box.

## The Add Farm Screen

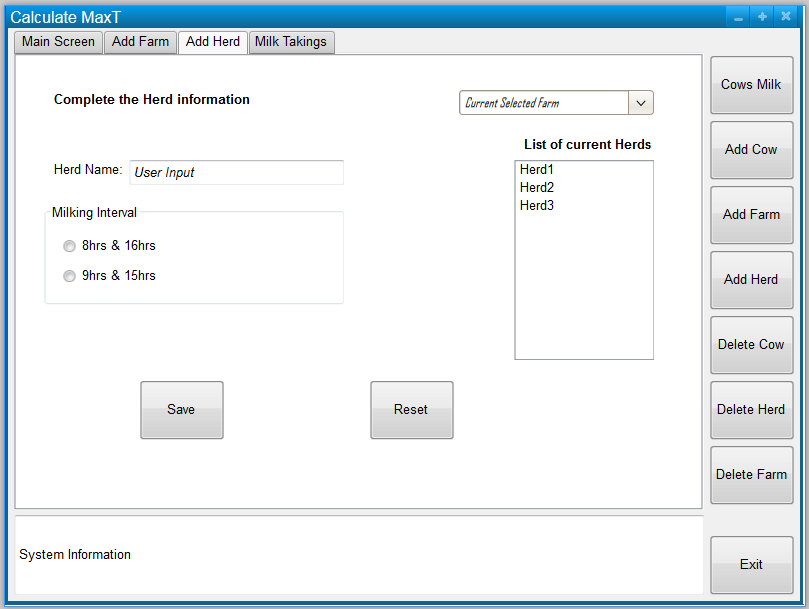


click



* The user has selected “Add Farm” either by tab or the right-hand side button.
* On entering details, the user can “Save” the new farm, the system will add the new farm to the list of current farms and display a message to the user.
* The list of current farms is displayed so the user is aware of other farm names in the system.
* If the user wishes to start again, without saving, a “Reset” button is available.
* The entry fields are cleared after a successful “Save” and a new farm can be created.
* If a user changes screens, by tab or button, any unsaved data will be lost.

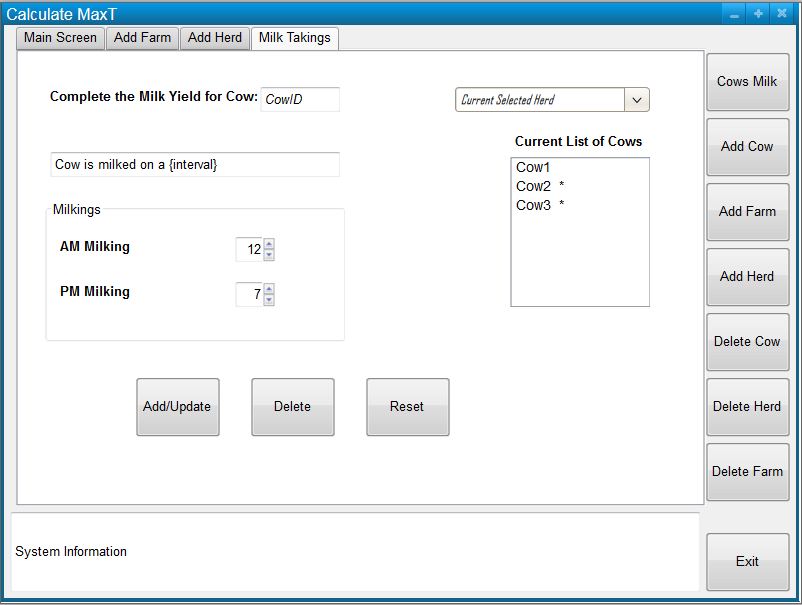
## The Add Herd Screen





* The user selects “Add Herd” from tab or right-hand side buttons.
* If a farm has been selected on the main screen, this is set as the “Current Selected Farm” and a list of current herds for the farm is displayed.
* The user can change the selected farm if they desire.
* On entering all valid data, the user may “Save” the herd, the system displays an appropriate message and adds the herd to the list of current herds.
* All entries are cleared and the user may enter a new herd.
* If a user makes a mistake, they can “Reset” the information, the system will not store any information.
* If a user changes screens, by tab or button, all unsaved data will be lost.

## The Milk Takings Screen



* The user selects “Milk Takings” from the tab or “Cows Milk” from the right-hand buttons, or double clicks on a “CowID” in the Cow list on the main screen and is taken to this screen.
* The cowID and milking interval of the cow is displayed.
* The user can enter the daily milk yields for that cow.
* A user can Add or Update the milk taking information, which the system will store and highlight the Cow in the list with an indicator “\*”.
* If a user makes a mistake they can “Reset” the information to original values.
* If the user wants to delete the milk taking value for the current cow the “Delete” button will activate this. The system will remove the milk taking values from the current cow and remove the indicator “\*” from the cow in the current list.
* A user may also “Delete Cow” from this page.
* If a user changes screens, by tab or button, all unsaved data will be lost.

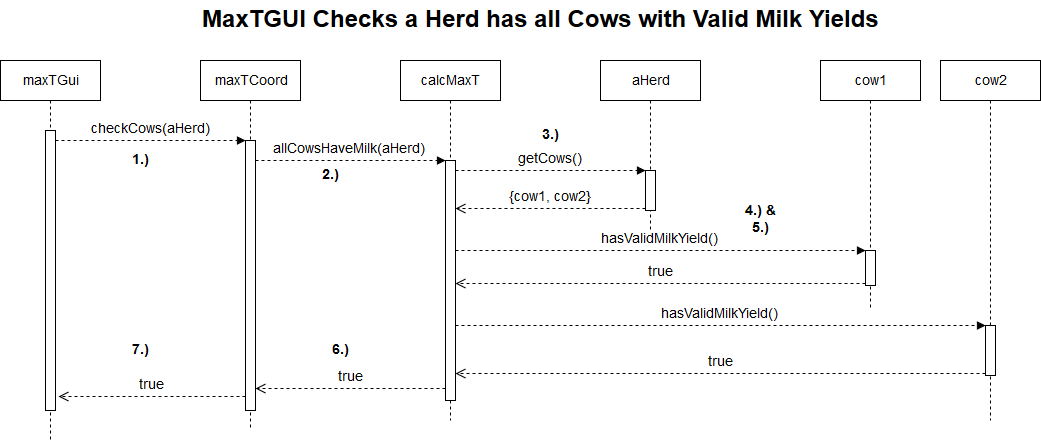
# Test Cases: Stage 2

|  |  |
| --- | --- |
| **Class MaxTCoord Test 5**  Tests for an empty herd from a Farm object |  |
| **Method**  getHerds(Farm aFarm) | Receiver and message  maxTCoord.getHerds(aFarm) |
| **Fixture and input**  A MaxTCoord object maxTCoord  a Farm object aFarm with empty herds | Expected result  An empty instance of Collection<Herd> is returned |
| **Class MaxTCoord Test 3**  Tests a Cow is added to a Herd |  |
| **Method**  addCow(Herd aHerd, Farm aFarm) | Receiver and message  maxTCoord.addCow(herd, farm) |
| **Fixture and input**  A MaxTCoord object maxTCoord  A Farm object farm  A Herd object herd in the Collection<Herd> herds associated with farm | Expected result  Returns true |
| **Class MaxTCoord Test 6**  Tests a Farm is deleted from the system |  |
| **Method**  deleteAFarm(Farm aFarm) | Receiver and message  maxTCoord.deleteAFarm(farm) |
| **Fixture and input**  A MaxTCoord object maxTCoord  A Farm object farm | Expected result  Returns true |

# Test Cases: Stage 3

|  |  |
| --- | --- |
| **Class CalculateMaxT testAllCowshaveMilk**  Tests all cows in a herd have valid milk yields |  |
| **Method**  checkHerd(Herd aHerd) | Receiver and message  instance.checkHerd(aHerd) |
| **Fixture and input**  A CalculateMaxT object instance  A Farm object aFarm  A Herd object aHerd  A Cow object cow1 with a valid MilkYield object  A Cow object cow2 with a valid MilkYield object | Expected result  true |
| **Class CalculateMaxT testHerdMilkAverage**  Test the average milk yield of the cows in a herd |  |
| **Method**  herdMilkAverage(Herd aHerd) | Receiver and message  instance.herdMilkAverage(aHerd) |
| **Fixture and input**  A CalculateMaxT object instance  A Farm object aFarm  A Herd object aHerd  A Cow object cow1 with a valid MilkYield object totalling 20  A Cow object cow2 with a valid MilkYield object totalling 22 | Expected result  A primitive type int which equals 21 |
| **Class CalculateMaxT testAmMilkMaxTTime**  Tests the morning milking maxT time returned for a herd |  |
| **Method**  amMaxTTime(Herd aHerd) | Receiver and message  instance.amMaxTTime(aHerd) |
| **Fixture and input**  A CalculateMaxT object instance  A Farm object aFarm  A Herd object aHerd  A Cow object cow1 with a valid MilkYield object totalling 20  A Cow object cow2 with a valid MilkYield object totalling 22 | **Expected result**  String object which equals “07:57” |

# Sequence Diagram



Part of **Use Case I**: **Calculate Values.** When **all** cows in a herd have milk takings associated with them….

Scenario, the user has selected aHerd on the user interface maxTGui, aHerd has 2 cows in it cow1 & cow2, both have valid milk yields.

**Step 1**. maxTGui sends a message checkCows(aHerd) to the coordinator maxTCoord and waits for a response true or false.

**Step 2.** maxTCoord sends a message allCowshaveMilk(aHerd) to the calculation coordinator calcMaxT and waits for a response true or false.

**Step 3**. calcMaxT send a message to getCows() to the Herd object aHerd and wait for a response of a Collection of Cow objects, which it receives back as cow1 and cow2.

**Step 4 & 5.** calcMaxT sends a message hasValidMilkYield() to each cow object and waits for a response true or false, both results from cow1 and cow2 are true.

**Step 6.** calcMaxT has received all true responses from the Cow objects, so calcMaxT responds to maxTcoord with a true result.

**Step 7.** maxTCoord has received a true result from calcMaxT, so responds to maxTGui with a true result also.