

AN INTERACTIVE SYSTEM FOR COW HERD PERFORMANCE

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Naqib Hossain

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Program:
Software Engineering

July 2019

Fargo, North Dakota

North Dakota State University
Graduate School

Title

AN INTERACTIVE SYSTEM FOR COW HERD PERFORMANCE

By

Naqib Hossain

The Supervisory Committee certifies that this *disquisition* complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Kendall E. Nygard

Chair

Dr. Oksana Myronovych

Dr. Mohiuddin Quadir

Approved:

August 1, 2019

Date

Dr. Kendall E. Nygard

Department Chair

ABSTRACT

This paper explores the application-development phases for the Cow Herd Appraisal of Performance System (CHAPS). In 1985, North Dakota State University's Dickinson Research Extension Center and the North Dakota Beef Cattle Improvement Association (NDBCIA) developed CHAPS as a management tool for cow producers. This was phase 1 development for the CHAPS application. CHAPS was initially built as a Microsoft Disk Operating System (MS-DOS) program. In phase 2, CHAPS was converted from MS-DOS to a desktop application. The desktop application was named CHAPS 2000. The desktop application lacks modern features, produces error-prone reports, etc. In 2013, North Dakota State University's Dickinson Research Extension Center and the NDBCIA decided to redevelop CHAPS. The proposed name was CHAPS 2020. This paper illustrates how CHAPS 2020 was designed, shows how the limitations of CHAPS 2000 were resolved, and describes the development processes for phase 3 in order to facilitate new and modern requirements.

ACKNOWLEDGMENTS

I want to acknowledge the following institutions and people:

- Dr. Kendall E. Nygard
- North Dakota State University's (NDSU) Dickinson Research Extension Center
- North Dakota Beef Cattle Improvement Association (NDBCIA)
- NDSU Department of Computer Science
- NDSU Department of Animal Science
- Professor Joseph Latimer and Professor Kris Ringwall
- Talukdar Asgar, Justin Riendeau, Deepak Dawar, Minhaz Chowdhury, Kelan Riley, and Priyanka Dawar
- Philip Berg, Keith Helmuth, and Madonna Tibor for their past development of the CHAPS software

DEDICATION

The disquisition is dedicated to my father, Md. Aflatun Hossain; my mother, Monowara Akhter;
and my wife, Naziath Farzana Karim.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	iv
DEDICATION	v
LIST OF TABLES	ix
LIST OF FIGURES	x
1. INTRODUCTION	1
2. BACKGROUND	3
3. REQUIREMENTS	5
3.1. Unique Identification.....	5
3.2. Report Generation	5
3.3. Application Interface	5
3.4. Application Platform	6
3.5. Architecture Design.....	6
3.6. Technology Stack.....	6
4. CHAPS	7
4.1. CHAPS II	7
4.1.1. The Calf Program.....	7
4.1.2. The Yearling Program.....	10
4.1.3. Observation.....	12
4.2. CHAPS 2000	12
4.2.1. File	14
4.2.2. Forms	15
4.2.3. Reports	15
4.2.4. Imports	16

4.2.5.	Exports	17
4.2.6.	Setup	18
4.2.7.	Utilities.....	19
4.2.8.	Help.....	19
4.2.9.	Observation.....	20
4.3.	CHAPS 2020	20
4.3.1.	Herd Setup	21
4.3.2.	Setting.....	24
4.3.3.	Bulls	27
4.3.4.	Cows	31
4.3.5.	Calves.....	34
4.3.6.	Reports	42
4.3.7.	Observation.....	44
4.4.	Comparisons Among CHAPS II, CHAPS 2000, and CHAPS 2020	45
5.	CHAPS 2020 TECHNOLOGY STACK	46
5.1.	Architecture	46
5.1.1.	System Architecture	46
5.1.2.	Application Architecture: AJAX	48
5.2.	Use Case Diagram.....	51
5.3.	Sequence Diagram.....	52
5.4.	Frontend.....	53
5.4.1.	HTML	53
5.4.2.	CSS	54
5.4.3.	Bootstrap.....	54
5.4.4.	Responsive Web Design	55

5.4.5.	JavaScript.....	55
5.4.6.	jQuery	56
5.5.	Backend	56
5.5.1.	Server	57
5.5.2.	Server-Side Language	57
5.5.3.	Database	58
5.6.	Testing	59
5.7.	Deployment	59
6.	REQUIREMENT FULFILLMENT.....	61
6.1.	Unique Identification.....	61
6.2.	Report Generation	61
6.3.	Application Interface	61
6.4.	Application Platform	62
6.5.	Architecture Design.....	62
6.6.	Technology Stack.....	63
7.	CONCLUSION.....	64
8.	FUTURE WORK.....	65
	REFERENCES	66

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1: Comparison of CHAPS II, CHAPS 2000, and CHAPS 2020.....	45

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1: Initial Menu.....	8
2: Main Menu.....	8
3: Herd Information.....	9
4: Calves Menu.....	9
5: Add Calf Page.....	10
6: Main Menu.....	11
7: Update Yearling.....	11
8: CHAPS 2000 Welcome Page.....	13
9: The First Page.....	13
10: Menu Item: File.....	14
11: Menu Item – Forms.....	15
12: Menu Item: Reports.....	16
13: Menu Item: Imports.....	17
14: Menu Item: Exports.....	17
15: Menu Item: Setup.....	18
16: Menu Item: Utilities.....	19
17: Menu Item: Help.....	19
18: CHAPS Flowchart.....	21
19: Herd Setup.....	22
20: Herd Creation.....	23
21: Herd Selection.....	23
22: Settings Page.....	24
23: Miscellaneous Fields.....	25

24: Bull Turn Out Date.	25
25: Single Breed.....	26
26: Crossbreed Button.....	27
27: Bulls.....	28
28: Sire Profile.....	29
29: Bulls EPD	30
30: Sire EPD.	30
31: Cows.	31
32: Cow Profile.....	32
33: Cows EPD.....	33
34: Cow EPD Creation.....	33
35: Birth Stage.	35
36: Calf Creation	36
37: Weaning Stage.	37
38: Weaning Creation.	38
39: Background Stage.	39
40: Background Stage.	39
41: Feedlot Form	40
42: Replacement page.	41
43: Replacement Creation	42
44: Herd Analysis Report.....	43
45: Progeny Report.	44
46: Culled Reference List.	44
47: Standalone System Architecture & Request Flow.....	46
48: Central System Architecture.	47

49: Classic vs. AJAX Web Architecture.....	49
50: Latency between classical vs Ajax web architecture.	50
51: Use Case Diagram of CHAPS 2020.	52
52: A Sequence Diagram of CHAPS 2020	53

1. INTRODUCTION

At the rise of computer technology during the 1980s, North Dakota State University's (NDSU) Dickinson Research Extension Center and the North Dakota Beef Cattle Improvement Association (NDBCIA) realized that they needed a computerized system for cow producers to keep herd records for future reference and analysis. As a result, the NDSU Dickinson Research Extension Center and the NDBCIA developed the Cow Herd Appraisal Performance Software (CHAPS) in 1985. CHAPS was reviewed and approved by the National Cattlemen's Beef Association following guidelines set by the Beef Improvement Federation (K. A. Ringwall, Berg, & Boggs, 1992; K. M. Ringwall, 2004). CHAPS is a data-intensive system for collecting, storing, and evaluating beef production as well as an overall herd-management tool.

CHAPS was initially built as an MS-DOS program (J. M. Ramsay, Hanna, & Ringwall, 2014). The first CHAPS should have been named CHAPS I, but after several reviews and updates, it was arbitrarily named CHAPS II. From here on, the first version of CHAPS is referred to as CHAPS II. In 2000, with the modernization of technology, the NDSU Dickinson Research Extension Center and the NDBCIA created a modern version of CHAPS, naming it CHAPS 2000. CHAPS 2000 was a desktop database application which was developed in the Visual Basic programming language with Microsoft Access 97 (phase 2). The NDSU Dickinson Research Extension Center and the NDBCIA noticed that CHAPS 2000 was not delivering the desired output and performance due to a lack of modern computer technology. To improve CHAPS 2000 and to facilitate the new, modern application requirements, they proposed a new version of CHAPS in 2013. This work was considered as phase 3 for the CHAPS program, and later, the name was changed to CHAPS 2020.

CHAPS 2020 maintains the basics of CHAPS 2000, including producer data entry, with a new interface that allows easier movement among the different stages of cow-calf production. Additionally, an internal identification field has been implemented to uniquely identify each sire, cow, and calf throughout the CHAPS 2020 application. CHAPS 2020 users who have multiple herds can store data for all herds in a single database. The rewrite of CHAPS 2000 improves the usability and makes data more accessible. This enhancement of CHAPS will help cow producers and NDSU Dickinson Research Extension Center professionals understand the trends with beef production in order to manage their beef herds effectively (J. Ramsay, Tisor, Hanna, Hossain, et al., 2017).

2. BACKGROUND

The most critical problem for CHAPS 2000 was that it did not have any unique identifiers for sires, cows, and/or any cattle. This problem led to other issues. CHAPS 2000 did not have the ability to store information for multiple herds which belonged to a single producer. For the same reason, it was not possible to get the animals' ancestry and to keep track of individual animals, which is crucial for this application because each animal, at some point in its life, will be the father or mother of a new calf. This issue was a possible debacle for researching the cattle's genealogy.

The process of getting the Herd Analysis Report is very difficult in CHAPS 2000. Each herd's data are stored in the Microsoft Access 97 database separately. Then, a Herd Analysis is calculated by using the functions of CHAPS 2000. Once the data are stored by producers or NDSU data-management specialists enter the information into CHAPS, the function generates a "Herd Report." NDSU data-management specialists manually transfer the herd-year benchmarks into an Microsoft Excel 2010 spreadsheet, identifying each CHAPS herd and year (J. M. Ramsay et al., 2014). The manual data transfer is prone to errors, including transcriptional errors such as transposed numbers and missing data, which can affect subsequent data analysis (Blasi, 2006; Hong et al., 2013).

The CHAPS 2000 interface is quite static and is not compatible with different monitor sizes. The increasing range of monitor sizes makes user interface more critical because most CHAPS 2000 users are not tech-savvy. User interfaces are not synchronized with the application's life cycle and are not self-explanatory. People must have enough training to do operations before using the program.

CHAPS 2000 is a Windows Operating System dependent platform. Many producers use the Macintosh Operating System now, so it is not possible for them to use the program on their machines.

The producers, or herd owners, live in rural areas where the internet is intermittently available. Therefore, it is a challenge for engineers and developers to design a system in such a way that it supports both online and offline processes, regardless of internet availability.

CHAPS 2020 is being developed by graduate students from the Computer Science Department at North Dakota State University as a part of their assistantship. The entire development heavily depends on the available graduate students and their skillsets, but graduate students leave when they finish their degree. It is challenging for developers and engineers to create CHAPS 2020 with technology that does not fade in the future and that requires a smaller learning curve.

3. REQUIREMENTS

An idea is the seed of an application system, but the idea is guarded with well-defined requirements. The requirements must be agreed on by the system's customers and users before it can be built (Westfall, 2005). It is very difficult to determine what features an application needs before it is made. The following sections outline the requirements which will be implemented in the future.

3.1. Unique Identification

The CHAPS 2020 database design introduces unique identifiers for sires, cows, and any cattle. The program gives a single producer the ability to store information for multiple herds. It is also possible to obtain the animals' ancestry, from generation to generation, and to research the cattle's genealogy.

3.2. Report Generation

A process to obtain the Herd Analysis Report from the standalone program without internet access and an NDSU data-management specialist is implemented. The multi-layer process for getting a report in CHAPS 2000 is replaced with a single-layer process.

3.3. Application Interface

The plan is to make the application browser based so that it comes from a box with every operating system's installation available. The browsers are compatible with different monitor sizes. The interfaces are designed to be responsive so that the applications runs smoothly,

regardless of the screen size. The user interfaces are also synchronized with the application's life cycle and are self-explanatory, so people require less training to master the new application.

3.4. Application Platform

To achieve the application's cross-platform capability, the developers have used JAVA by Sun Microsystems. The code's slogan is "Write once, run anywhere" (Blom, Book, Gruhn, Hrushchak, & Köhler, 2008).

3.5. Architecture Design

The application requires an architecture design that supports both online and offline operations, regardless of internet availability. Offline functionality is necessary because many CHAPS customers live in rural areas where internet availability is not always possible.

3.6. Technology Stack

The goal was to utilize technologies in such a way that the onboarding developers and engineers can grasp the development with a smaller learning curve. That reason led earlier designers to choose basic technologies over the advanced one of a stack. For example, if previous developers had chosen a specific framework, then it would have narrowed the choices for available graduate students with the same skillset.

4. CHAPS

To this point, CHAPS has three computerized versions. Each iteration has a different and unique setup, development process, user interface, and way of doing things. This chapter describes each CHAPS version's setup, interfaces, and procedures. The description of CHAPS II and CHAPS 2000 is brief due to lack of their documentation and resources. The description of CHAPS 2020 is more complete.

4.1. CHAPS II

CHAPS II required a PC-DOS or MS-DOS operating system (version 2.0 or higher) for installation on an IBM-PC or compatible microcomputer with a minimum of 256K RAM (system memory) (Kris Alan Ringwall & Berg, 1990). The program required a minimum of two 360K floppy disks, a single 720K or greater floppy disk, or a system with one floppy and one hard disk. K. A. Ringwall and Berg (Kris Alan Ringwall & Berg, 1990) described a double 360K disk and hard-disk system. CHAPS II is a console window-based application, and navigation within the application is strictly through menus on the console. CHAPS II consists of two programs: the calf program and the yearling program.

4.1.1. The Calf Program

Figure 1 shows the console which appears when the application is started. Upon selection of C), the cow program appears. The cow program evaluates the cow herd's performance. This program primarily works to gather the data. It neither processes any data nor generates a report. Below, I discuss four pages of the cow program.

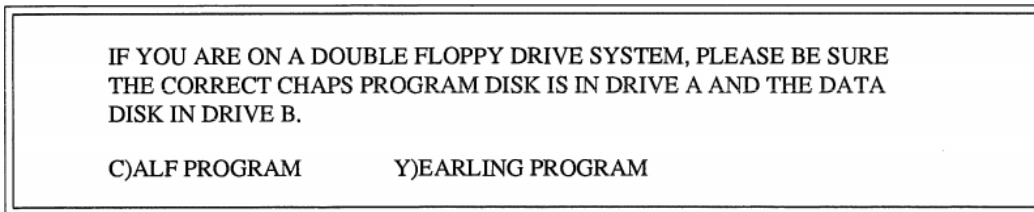


Figure 1: Initial Menu.

Figure 2 illustrates the CHAPS II menu. The top-left-corner text gives the herd name, which is BIF01, and the used data drive, which is B. The number in the top-right corner states the page's reference number. The menu items in the middle of the page are self-explanatory.

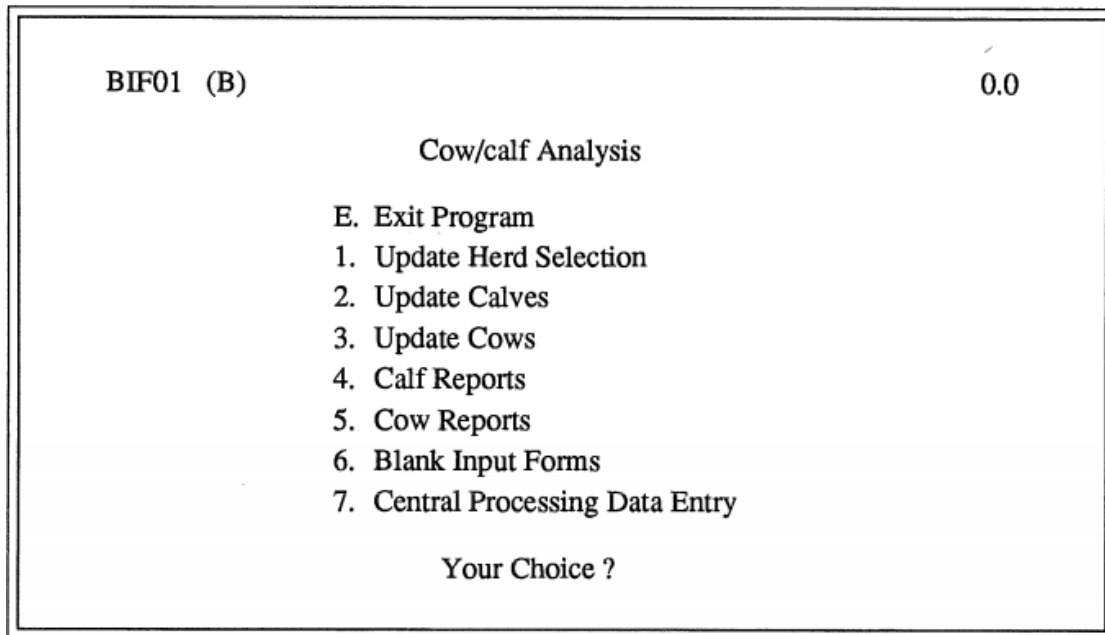


Figure 2: Main Menu.

Figure 3 is the result of choosing option 1 from the main menu shown in Figure 2. This page also has the previous page's reference on the top-right corner. The owner can edit the herd's information on this page. The form, page's body, and fields are self-explanatory.

BIF01 (B)	1.0
Update Herd Selection	
Herd ID: BIF01	
Name: North Dakota BCIA	
Address: North Dakota State University R&E Center	
City/St/Zp: Hettinger, ND 58639	
County: Adams	
Data Drive: B (A,B,C)	
<← NEXT ENTRY ← BACKSPACE <ESC> EXIT <TAB> END OF PAGE	

Figure 3: Herd Information.

Figure 4 is the result of choosing option 2 from Figure 2's main menu. This page also has the previous page's reference in the top-right corner. This page is for the calf menu, implying that everything related to calves can be done from this page.

BIF01 (B)	2.0
Update Calves	
0. Return to Previous Menu	
1. Add a Calf	
2. Edit an Existing Calf	
3. Delete a Calf	
4. Print All Input Data for This Herd	
5. Add Previous Years Calf Ratios	
6. Delete All Blank Calves from File	
Your Choice ?	

Figure 4: Calves Menu.

Figure 5 is the last page discussed here. The top-right corner's reference number indicates that this page originated from Figure 4's menu item 1. This form is utilized to enter a calf into

the system. Not all form fields are self-explanatory. The reader is referred to Ringwall and Berg (Kris Alan Ringwall & Berg, 1990) for a more descriptive explanation about the form's fields.

BIF01 (B)	2.1
Add a Calf	
Last Calf ID#: 8906	
Calf ID#: _____	Actual Wt: _____
Cow ID#: _____	Date Weighed: _____
Cow Age in Yrs: —	Management Code: -
Cow Breed: _____	Frame Score: _____
Sire of Cow ID#: _____	Cont. Group: -
Calf Sex: -	Grade: —
Sire ID#: _____	Misc. Code 1: _____
Sire Breed: _____	
Calf Birthdate: _____	
Birth Weight: —	
Calving Ease #: -	
<← NEXT ENTRY <← BACKSPACE <ESC> EXIT <TAB> END OF PAGE	

Figure 5: Add Calf Page.

4.1.2. The Yearling Program

When selecting Y) from Figure 1, the yearling program's main menu appears on the screen. The yearling program evaluates the calves' performance during a time period approximately between weaning and one year. This program is automatically the next part of CHAPS II. The yearling program utilizes the data which were entered into the cow program. The calf data from the cow program are transferred to the yearling program to calculate and to generate necessary reports. Two pages from the yearling program will be discussed below.

BIF01	(B)	0.0
Yearling Program		
E. Exit Program		
1. Update Herd Selection		
2. Update Yearlings		
3. Yearling Reports		
4. Blank Input Forms		
Your Choice ?		

Figure 6: Main Menu.

The interfaces for the yearling program are consistent with the cow program. Figure 6 shows the yearling program's main menu. This program gets the calf's transferred data and processes them. When necessary, the user can also print the desired report.

BIF01	(B)	2.0
Update Yearlings		
0. Return to Previous Menu		
1. Add a Yearling		
2. Edit an Existing Yearling		
3. Delete a Yearling		
4. Print All Input Data for This Herd		
5. Edit an Entire Herd for a Given Year		
6. Transfer Calf Data to Yearling File		
Your Choice ?		

Figure 7: Update Yearling.

Figure 7 shows the page to update the yearling menu. This menu is generated from Figure 6 by selecting menu item 2. From this page, the user can go further to edit or to add yearling data, and to print all the information.

4.1.3. Observation

The CHAPS II developers were consistent in terms of the user interface, giving the application more visual clarity. This application was console based, so it was not easy to navigate through pages by clicking on the menu like we do with modern desktop/web applications. Therefore, the referencing mechanism for page and menu items was very helpful for users. It's easy to get lost in a console-based application, but the application was well written for users to get back on track when necessary.

Although the CHAPS software was separated into two parts, the cow program and yearling program, during its early days, the later CHAPS 2000 version was combined into one part. From CHAPS 2000 and on, a combined version of 2 parts are used. Unlike CHAPS II, the later CHAPS versions have moved to more user-friendly interfaces, avoiding unnecessarily complex situations. The technology advancements also helped the later CHAPS versions to maintain the integrity of stored data in a single database which is always available throughout the application.

4.2. CHAPS 2000

CHAPS 2000 requires an operating system of at least Microsoft Windows 2000 (Windows 2000) installed with all minimum requirements. CHAPS 2000 is a desktop application that was written in the Visual Basic programming language, and Microsoft Access 97 (J. Ramsay, Tisor, Hanna, & Ringwall, 2017) is used as the database to store data. The computer must have a CD ROM because CHAPS 2000 comes with a CD that needs to be installed. An executable icon with the text "chaps" appears on the desktop after installing CHAPS 2000. It is

surprising that CHAPS 2000 still works with Windows 10, Microsoft's newest operating system, with the limitations described in Chapter 2.

There is no adequate information about the requirements for CHAPS 2000 to run on the computer. The previously mentioned information is the best obtained from other papers, and the rest is from people's experience with the application throughout the years.

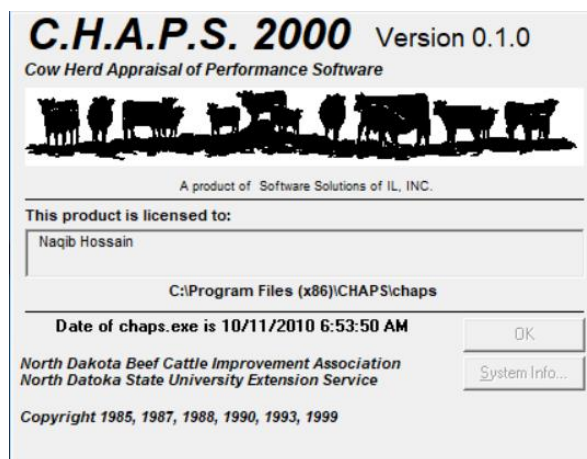


Figure 8: CHAPS 2000 Welcome Page.

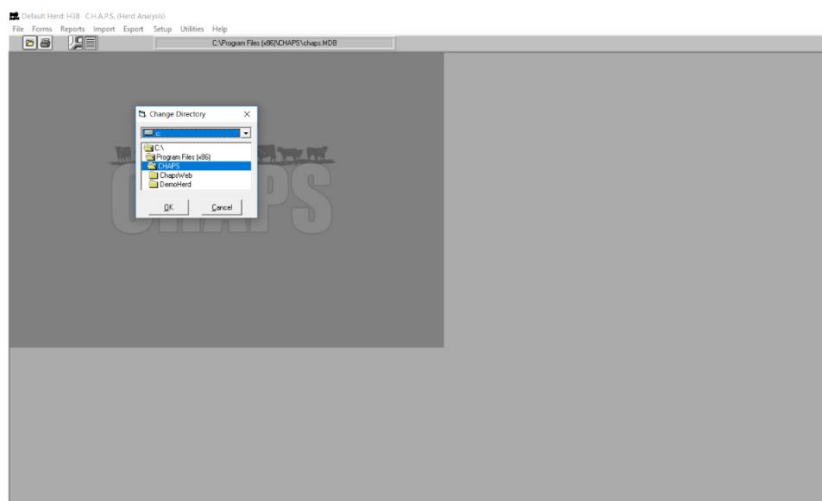


Figure 9: The First Page.

Figure 8 appears for few seconds when the “chaps” executable icon is run, and then, Figure 9 comes up as the first page of the application. The first page has a mini window which displays a pre-selected CHAPS directory. A user needs to click the OK button to go further and to start the normal operation of CHAPS 2000. A top-level menu is also visible in Figure 9; these items are briefly discussed in the following sections because the paper’s principal goal is to explain CHAPS 2020 in detail.

4.2.1. File

Figure 10 shows that File menu has four options: Open, Print Screen, Printer Setup, and Exit. Only the Open menu item is discussed. The Open item also has 3 sub items: Calves, Cows, and Sires; their corresponding windows are indicated by matching red numbers. Different animal types have specific attributes which are entered to store in the system.

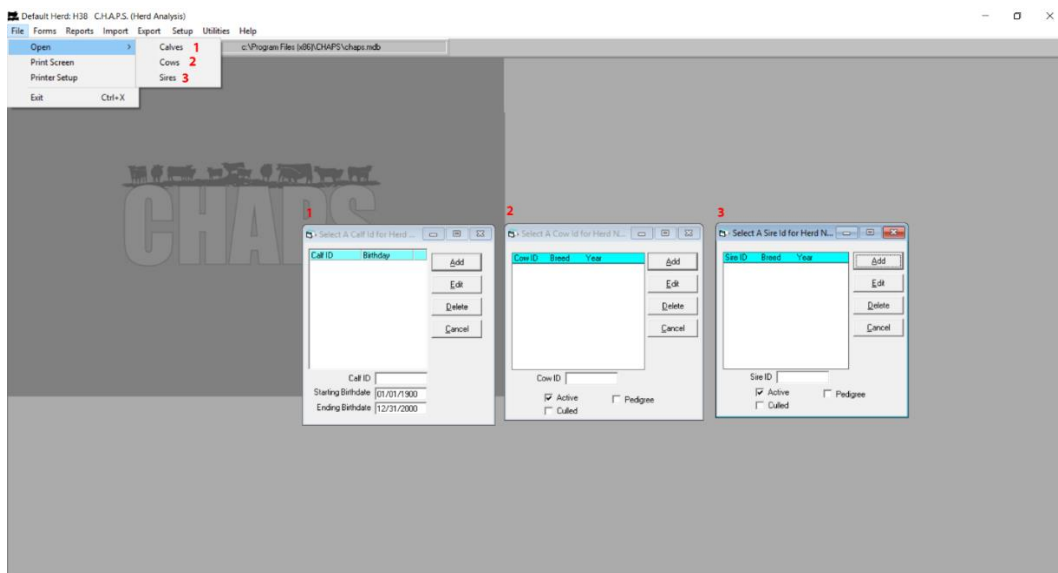


Figure 10: Menu Item: File.

4.2.2. Forms

Figure 11 shows the window with different options when clicking the Forms menu item.

Different pdf shows up upon selection of options and Ok button.

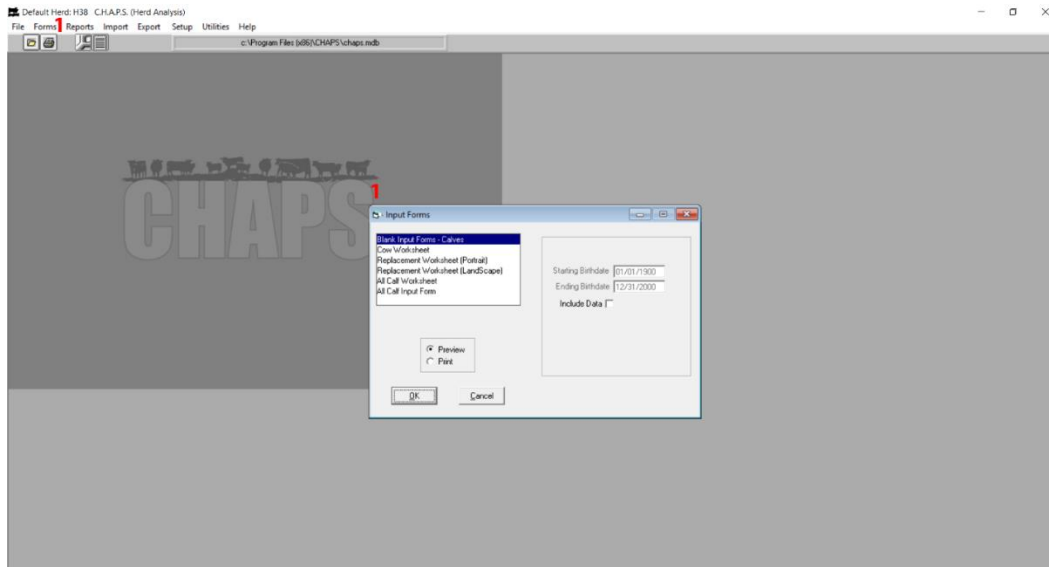


Figure 11: Menu Item – Forms.

4.2.3. Reports

Figure 12 shows the menu item which generates different report types. The entire application can basically be described in two ways: storing animal information and generating reports based on that information. The corresponding windows are marked with red numbers for easy inference. Each report has options for Start Birthdate or End Birthdate to be selected prior to report generation. The report can be presented in two ways: Preview and Print.

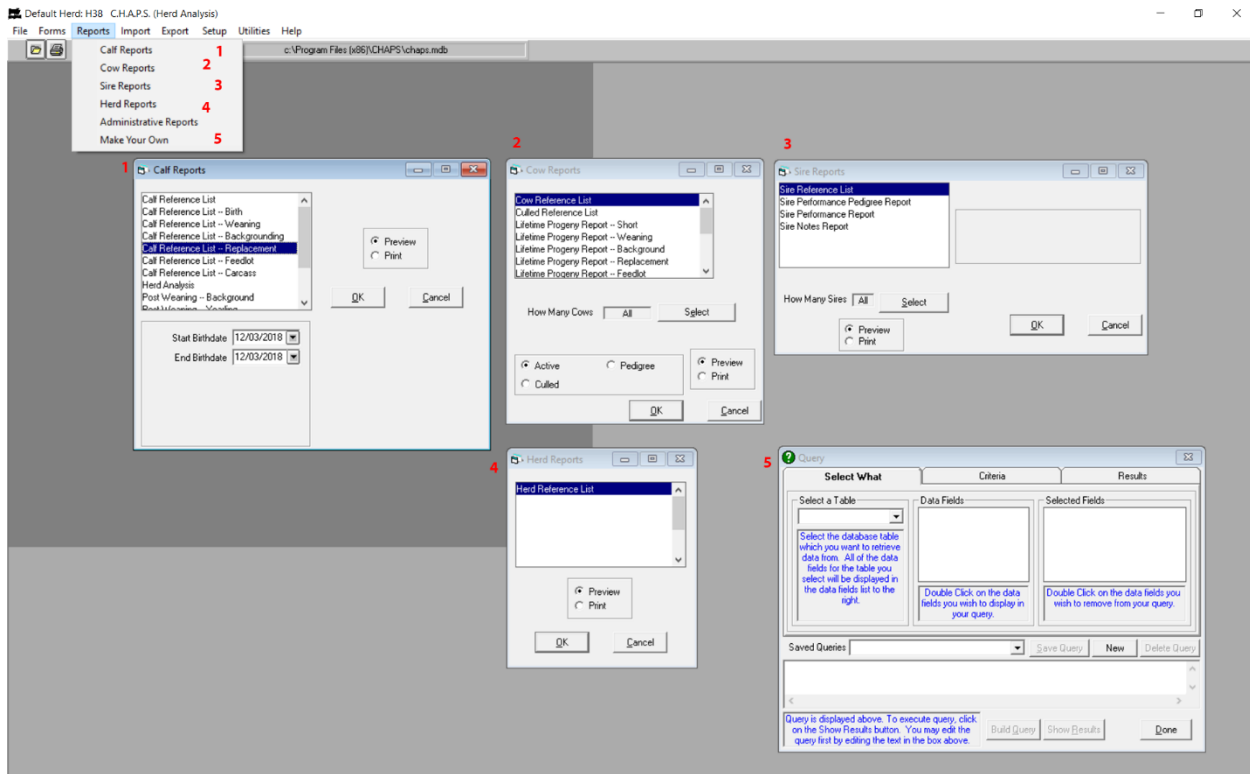


Figure 12: Menu Item: Reports.

4.2.4. Imports

The Import menu in Figure 13 has two options: DOS to Windows Conversion and Import Chaps Data. CHAPS 2000 has the ability to import data from the CHAPS II application database, but the new application, CHAPS 2020, does not have this feature because CHAPS II is now in no use. Both windows are marked with red numbers to indicate their correspondence to the menu. Each window has self-explanatory options from which to choose.

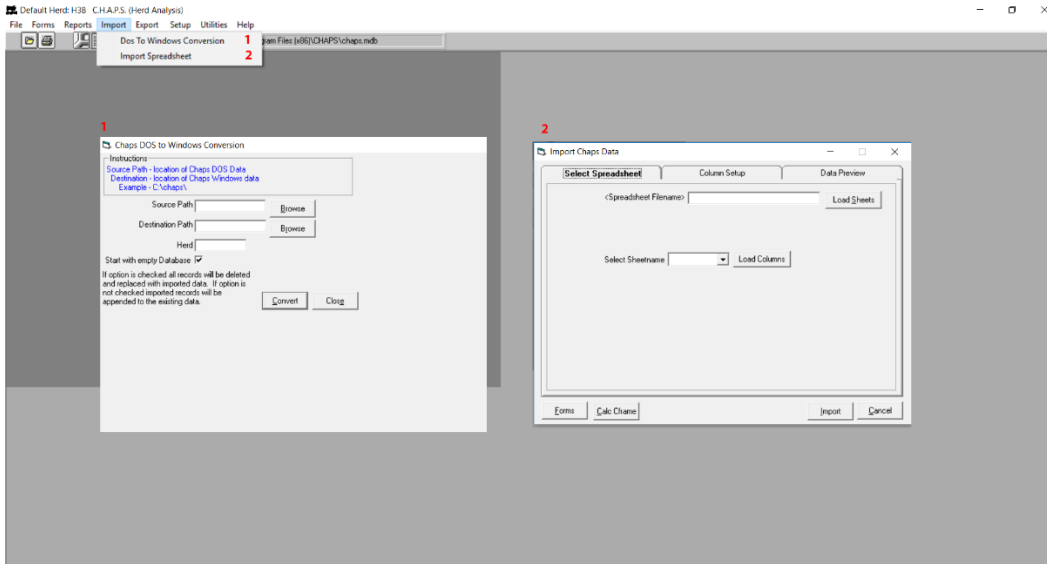


Figure 13: Menu Item: Imports.

4.2.5. Exports

Figure 14 shows a dialogue box containing a long message which explains how to export a report in this application. There is no easy process to export the report; rather, by following processes, the export is possible.

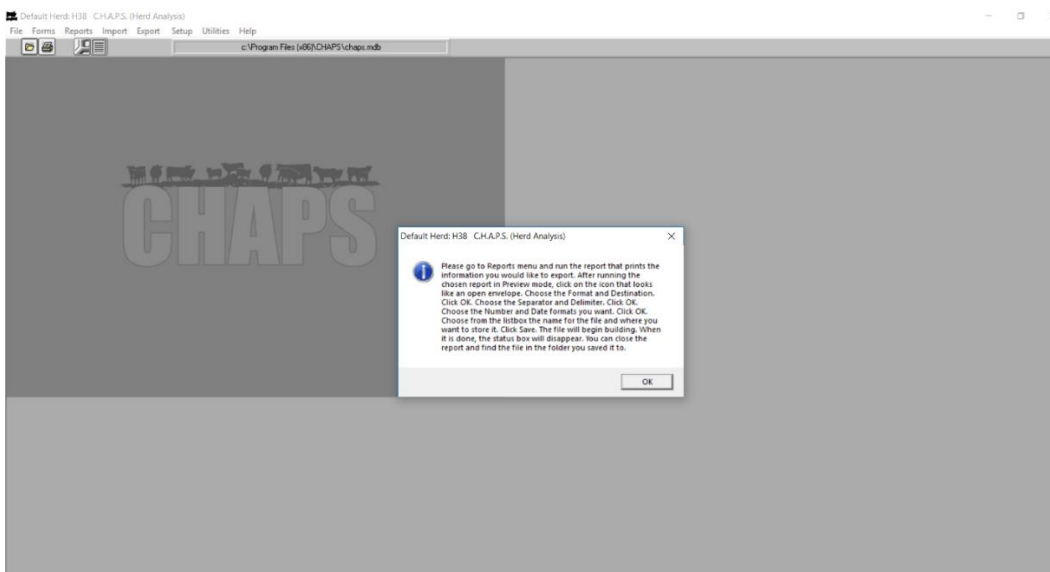


Figure 14: Menu Item: Exports.

4.2.6. Setup

Figure 15 shows the Setup menu's sub-menu items, illustrating different options for the application to change and to manipulate its properties. For example, a new herd can be generated, or an existing one can be edited. The reports generation needs some predefined values to run calculations while generating reports; SPA Default is the place to put those values. Also, Preferences is where miscellaneous fields can be added for different forms to be appeared within forms. The windows are marked with red numbers which correspond to the menu items.

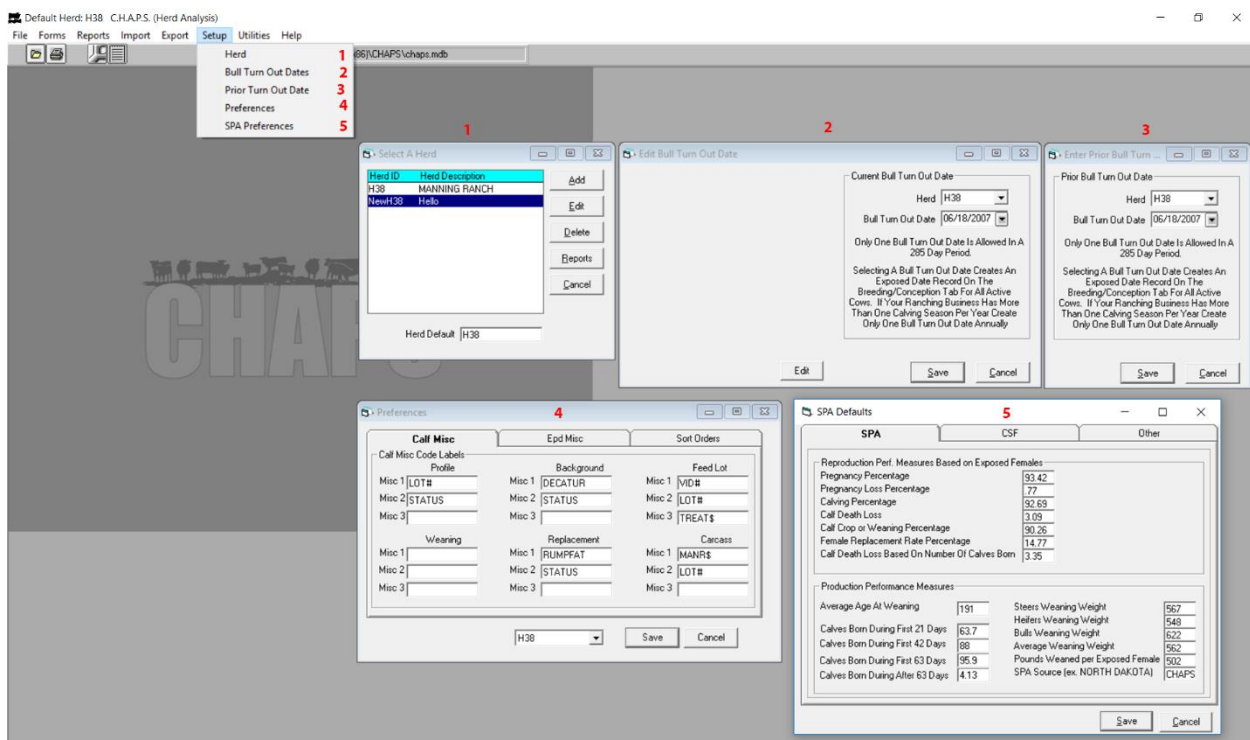


Figure 15: Menu Item: Setup.

4.2.7. Utilities

Figure 16 shows that the Utilities menu has options, Update Notes, Database, Dictionaries, and Copy Data, which can be utilized at various times.

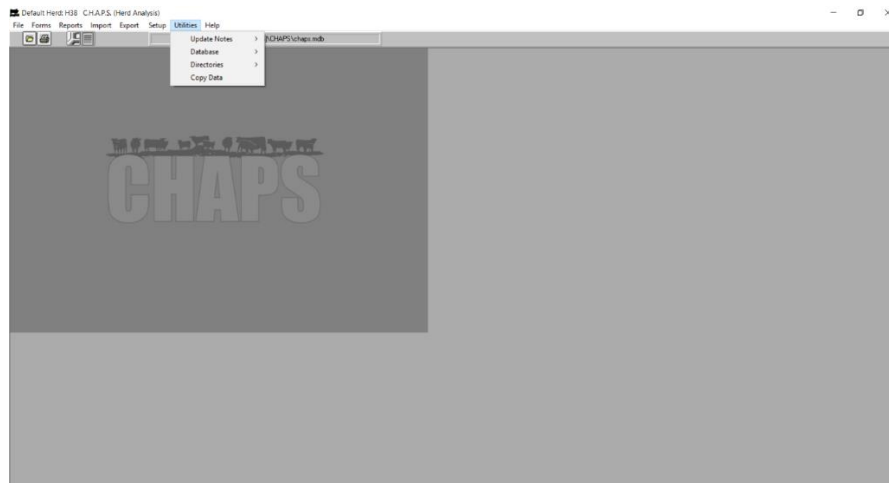


Figure 16: Menu Item: Utilities.

4.2.8. Help

Figure 17 shows some options which assist the user to get started with CHAPS 2000. Some information about the specific instance of CHAPS 2000 is also seen.

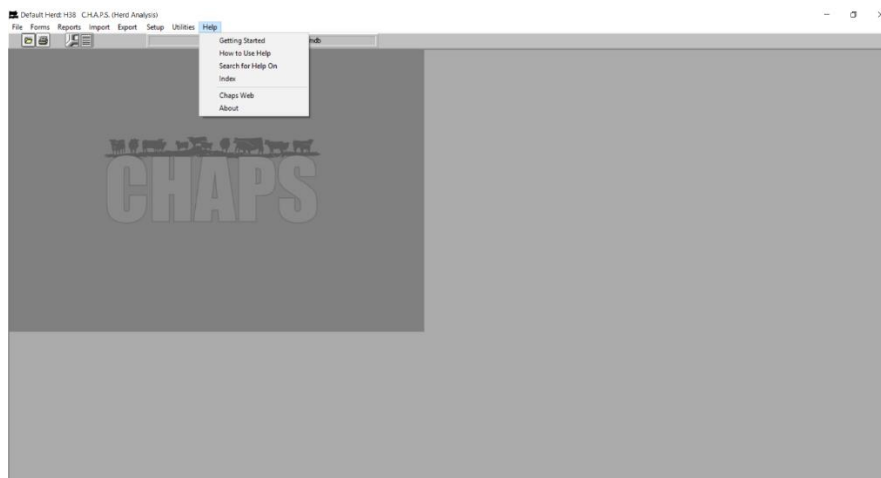


Figure 17: Menu Item: Help.

4.2.9. Observation

The CHAPS 2000 writers were consistent in terms of the user interface, giving the application more visual clarity. The CHAPS 2000 is a desktop application and it does not change with the size of the monitor or output device. The computer industry's advancement with monitors of varying sizes created an issue with the CHAPS 2000 interface. CHAPS 2000 is highly evolved from CHAPS II in terms of the interface, features, and technology used. Unlike CHAPS II which was divided into two segments, CHAPS 2000 segmented the application into different menus and sub menus for easy navigation and user interactivity.

The CHAPS 2000 developers also showed promise in terms of the application's database. They used Microsoft Access 97 which was one of the latest ones. CHAPS 2000 no longer needed an external device, such as a floppy disk or CD, to store data; rather, the animal data could be stored on the computer's internal hard drive. This approach greatly lessened the chance of data loss and showed the adaptability of new technology during the second phase for CHAPS development.

4.3. CHAPS 2020

The development of CHAPS 2020 started with designing and implementing the database. There was no adequate documentation from CHAPS 2000 to create a database for CHAPS 2020. The previous development team had to inspect CHAPS 2000's Access database and Visual Basic code line by line in order to figure out the program's the normal operation. Creating the database for CHAPS 2020 was nearly stopped in early 2016. At that time, I joined the team.

One prime requirement and specification for this application was to create an optimal user interface for customers. Much discussion went on, and the team decided to create a user

interface that would imitate the CHAPS operation flowchart which is shown in Figure 18. The team created a frontend that amplifies that overall operation of CHAPS.

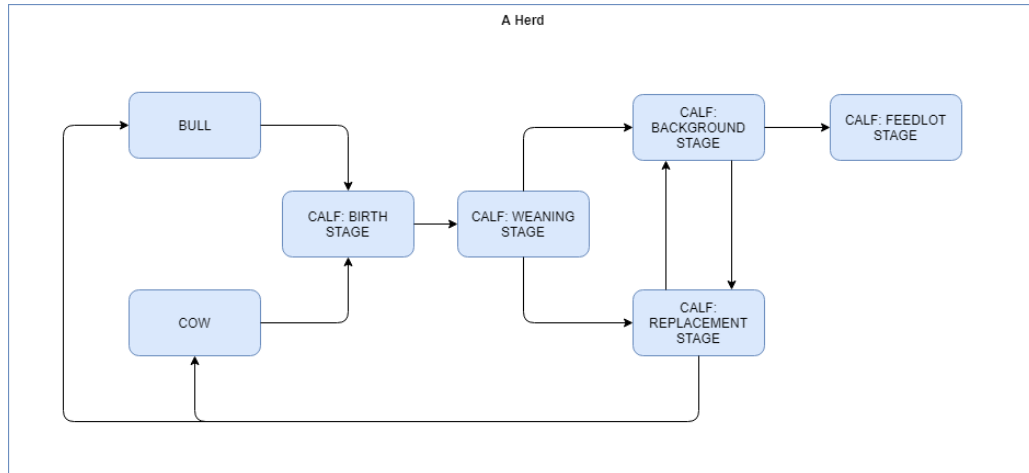


Figure 18: CHAPS Flowchart.

Figure 18 shows that herd operation starts with the Bull and Cow sections. A calf is born from a cow and a bull, creating the calf's birth stage. At a certain point, the calf is weaned. Then, the calf goes to either the Background or the Replacement stage. The Background and Replacement stages can switch for the cows or bulls. An animal in the Background stage can go to the Replacement stage or to the Feedlot stage. The animal's last stage is the Feedlot stage. Animals in the Replacement Stage can be cows or bulls; the animals can also switch stages to be in the Background. The operation outlined above is explained, in detail, in the next sections.

4.3.1. Herd Setup

The first page of CHAPS 2020, Herd Setup, displays a few statistics for an existing, pre-selected herd. The team used the "TestHerd1" herd while describing different processes and operations for CHAPS 2020. Figure 19 illustrates that "TestHerd1" has 22 bulls, 31 cows, etc. If

an empty herd is selected, these statistics will be 0 (zero). This page also tells about the CHAPS 2020 version and its release date.

CHAPS Herd ID: TestHerd1

HERD SETUP SETTINGS BULLS COWS CALVES REPORTS ▾

WELCOME TO CHAPS

Version: 0.1 (Beta)
Release Date: 06/10/2019

The Current Herd ID: TestHerd1

Bulls: 22	Cows: 31		
Calves			
Birth: 38	Weaning: 36	Background: 4	Replacement: 12

Select a Different Herd

- OR -

Setup a New Herd

* Indicates required field
© 2018 CHAPS

Figure 19: Herd Setup.

There are two buttons, “Select a Different Herd” and “Setup a New Herd,” right below the herd’s statistics in Figure 19. If the “Setup a New Herd” button is selected, a form shows up as a popup box as illustrated in Figure 20. The red asterisk (*) in the Herd ID field indicates that this field is required. The form also has optional fields for customers to utilize.

Insert a New Herd ×

Herd ID*	County
<input type="text"/>	<input type="text"/>
Herd Description	District
<input type="text"/>	<input type="text"/>
Address	Region
<input type="text"/>	<input type="text"/>
City	Comments 1
<input type="text"/>	<input type="text"/>
State	Comments 2
<input type="text" value="Alabama"/>	<input type="text"/>
Zip	Premise ID
<input type="text"/>	<input type="text"/>

Figure 20: Herd Creation.

Figure 21 appears when the “Select a Different Herd” button is chosen. This popup box has a list of herds with a radio button. The user needs to choose any one and submit it.

Select a Different Herd ×

TestHerd1
 TestHerd2

Figure 21: Herd Selection.

4.3.2. Setting

Certain settings needs to be configured before starting the use the application. Figure 22 shows three sections to consider. Later operations and calculations depend on these settings: Miscellaneous Fields, Breed List Customization, and Bull Turn Out Date. The following sections go into more detail.

CHAPS Herd ID: TestHerd1 HERD SETUP SETTINGS BULLS COWS CALVES REPORTS ▾

SETTINGS

Miscellaneous Fields

Sire Form Sire EPD Form Cow Form Cow EPD Form Calf-Birth Form
Calf-Weaning Form Calf-Background Form Calf-Replacement Form

Bull Turn Out Date

08-08-2017
08-17-2017
Insert bull turn out date

Breed List Customization

Single Breed Cross Breed

* Indicates required field
© 2018 CHAPS

Figure 22: Settings Page.

The Miscellaneous Fields section on the Settings page contains eight form buttons: Sire, Sire EPD, Cow, Cow EPD, Calf Birth, Calf Wean, Calf Background, and Calf Replacement. Each button has a popup box with eight fields. If a customer enters text in a field, the field shows up with the given text in the corresponding form. Initially, the fields were dynamic, but due to database-design limitations, the plan was discarded. Later, enough fields, eight in this case, were created so that the customer could satisfy his or her requirements. Figure 23 shows a form with eight empty fields for the Miscellaneous Fields.

Update Miscellaneous Fields ×

<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Figure 23: Miscellaneous Fields.

Another section on the Settings page is Bull Turn Out Date. It is a single-date field form which is shown in Figure 24. This date is very important. A Herd Analysis Report, which is described, in detail, later, cannot be generated without this date.

Insert bull turn out date ×

Choose bull turn out date*

Figure 24: Bull Turn Out Date.

Another section of the Settings page is Breed List Customization. It has two buttons: Single Breed and Cross Breed. Figure 25 shows the form for the Single Breed button. A breed can be selected from the predefined checkboxes. The selected breed is used for later forms.

Select ONE Breed (at a time) ×

<input type="checkbox"/> American Aberdeen (LO)	<input type="checkbox"/> Blonde D'aquitaine (BD)	<input type="checkbox"/> Dutch Belted (DL)	<input type="checkbox"/> Pinzgauer (PZ)
<input type="checkbox"/> American Akaushi (AA)	<input type="checkbox"/> Bos indicus influenced (BX)	<input type="checkbox"/> Galloway (GA)	<input type="checkbox"/> Ranger (RA)
<input type="checkbox"/> American Breed (AE)	<input type="checkbox"/> Brahman (BR)	<input type="checkbox"/> Gelbvieh (GV)	<input type="checkbox"/> Red Angus (AR)
<input type="checkbox"/> American Shorthorn (SH)	<input type="checkbox"/> Brangus (BN)	<input type="checkbox"/> Hereford (Horned) (HH)	<input type="checkbox"/> Red Brangus (RB)
<input type="checkbox"/> Amerifax (AM)	<input type="checkbox"/> Braunvieh (BU)	<input type="checkbox"/> Hereford (Polled) (HP)	<input type="checkbox"/> Red Poll (RP)
<input type="checkbox"/> Angus (AN)	<input type="checkbox"/> British White (BW)	<input type="checkbox"/> Highland (Scotch) (SH)	<input type="checkbox"/> Romagnola (RN)
<input type="checkbox"/> Ankina (AK)	<input type="checkbox"/> Brown Swiss (SB)	<input type="checkbox"/> Holstein (HO)	<input type="checkbox"/> Romosinuano (RS)
<input type="checkbox"/> Ankole-Watusi (AW)	<input type="checkbox"/> Buelingo (BQ)	<input type="checkbox"/> Jersey (JE)	<input type="checkbox"/> Salers (SA)
<input type="checkbox"/> Barzona (BA)	<input type="checkbox"/> Canadienne (CN)	<input type="checkbox"/> Limousin (LM)	<input type="checkbox"/> Santa Gertrudis (SG)
<input type="checkbox"/> Beef Crossbreeds (XB)	<input type="checkbox"/> Charolais (CH)	<input type="checkbox"/> Lincoln Red (LR)	<input type="checkbox"/> Senepol (SE)
<input type="checkbox"/> Beef Friesian (BF)	<input type="checkbox"/> Chianiana (CA)	<input type="checkbox"/> Maine-Anjou (MA)	<input type="checkbox"/> Simmental (SM)
	<input type="checkbox"/> Marchigiana (MA)		<input type="checkbox"/> South Devon (DS)

Figure 25: Single Breed.

Figure 26 shows the form when the Cross Breed button is selected. A crossbreed is created by selecting two different breeds for the animal's Sire, male parent, and Dam, female parent. Customers can create a crossbreed of their own and can use it later in the operation.

Select ONLY TWO (one from each group) for a joint breed -at a time! ✕

Sire

<input type="checkbox"/> American Aberdeen (LO)	<input type="checkbox"/> Brahman (BR)	<input type="checkbox"/> Hereford (Horned) (HH)	<input type="checkbox"/> Salers (SA)
<input type="checkbox"/> American Shorthorn (SH)	<input type="checkbox"/> Brangus (BN)	<input type="checkbox"/> Hereford (Polled) (HP)	<input type="checkbox"/> Santa Gertrudis (SG)
<input type="checkbox"/> Angus (AN)	<input type="checkbox"/> Charolais (CH)	<input type="checkbox"/> Highland (Scotch) (SH)	<input type="checkbox"/> Simmental (SM)
<input type="checkbox"/> Beef Crossbreeds (XB)	<input type="checkbox"/> Chianiana (CA)	<input type="checkbox"/> Limousin (LM)	<input type="checkbox"/> Tarentaise (TA)
	<input type="checkbox"/> Dexter (DR)	<input type="checkbox"/> Red Angus (AR)	<input type="checkbox"/> Texas Longhorn (TL)
	<input type="checkbox"/> Gelbvieh (GV)		

Dam

<input type="checkbox"/> American Aberdeen (LO)	<input type="checkbox"/> Brahman (BR)	<input type="checkbox"/> Hereford (Horned) (HH)	<input type="checkbox"/> Salers (SA)
<input type="checkbox"/> American Shorthorn (SH)	<input type="checkbox"/> Brangus (BN)	<input type="checkbox"/> Hereford (Polled) (HP)	<input type="checkbox"/> Santa Gertrudis (SG)
<input type="checkbox"/> Angus (AN)	<input type="checkbox"/> Charolais (CH)	<input type="checkbox"/> Highland (Scotch) (SH)	<input type="checkbox"/> Simmental (SM)
<input type="checkbox"/> Beef	<input type="checkbox"/> Chianiana (CA)	<input type="checkbox"/> Limousin	<input type="checkbox"/> Tarentaise (TA)

Figure 26: Crossbreed Button.

4.3.3. Bulls

The Bulls page is the first page to start with a herd animal. Figure 27 shows the page's basic structure. The page is separated into two sections: Profile and EPD (Expected Progeny Difference). The Profile section is described here. There is ambiguity with the page named Bull while treating the animal as a sire. As a general term, Bull is the page's name but sire, a breeding-specific male that fathers a calf, is utilized for adding, editing, and printing the animal's information. This page has the sire's listing which can be printed with the Print Sire button. There are three sire types: Active, Culled, and Pedigree; these types can be used to filter the sires' listing. The next paragraph describes the page in more detail.

CHAPS Herd ID: TestHerd1 HERD SETUP SETTINGS **BULLS** COWS CALVES REPORTS

BULLS

Profile

EPD

Add Sire

Active
 Culled
 Pedigree

Print Sire

Show 10 entries
Search:

Sire ID	Dam of Sire	Sire of Sire	Birth Date	Breed	Source	Status	Culled Reason	Reg. Name	Reg. No
1246			12-11-2018	MP	Raised	Active			
21			12-12-2018	an	Raised	Active			
46				NAM	Raised	Active			
72		21		NA	Raised	Active			
76	54	A4444		MAN	Raised	Active			
890	A123	B290		MNO	Raised	Active			
98	Unknown	A122		KLM	Raised	Active			
A124				Ab	Raised	Active			
A125	C1	A234	12-11-2018	AN	Raised	Active			
A234				wf5	Raised	Active			

Showing 1 to 10 of 22 entries

Previous
1
2
3
Next

Figure 27: Bulls.

The form in Figure 28 pops up when the Add Sire button is clicked from the Bulls page. A sire can be added by providing all the necessary information. The sire can also be added from the calf's Replacement stage; in that case, the calf's ID should be entered into the Sire Calf ID at Birth field, and all necessary information is be auto populated. The Weather field is an example of a Miscellaneous field. The same form appears when edit of that animal's information.

Add Sire Profile
✕

Sire ID*	Sire Calf ID at Birth	Sire Birth Date
<input type="text"/>	<input type="text"/>	<input type="text" value="MM-DD-YYYY"/>
Sire Breed*	Sire of Sire	Dam of Sire
<input type="text" value="Search Sire Breed"/>	<input type="text" value="Search Sire ID"/>	<input type="text" value="Search Dam ID"/>
Registration #	Registration Name	Electronic ID #
<input type="text"/>	<input type="text"/>	<input type="text"/>
Date Entered Herd	Source	Status
<input type="text" value="MM-DD-YYYY"/>	<input checked="" type="radio"/> Raised <input type="radio"/> Purchased <input type="radio"/> Natural Service <input type="radio"/> A.I.	<input checked="" type="radio"/> Active <input type="radio"/> Culled <input type="radio"/> Pedigree
Weather		
<input type="text"/>		
Profile Notes		
<input style="width: 100%; height: 50px;" type="text"/>		

Figure 28: Sire Profile.

Figure 29 shows the section of Expected Progeny Difference (EPD) from the Bulls page. On the EPD page, a vast amount of data about different traits is collected and can be used to improve the beef cattle's genetics. The EPD listing for an animal can be printed with the Print Sire EPD button.

CHAPS Herd ID: TestHerd1 HERD SETUP SETTINGS **BULLS** COWS CALVES REPORTS ▾

BULLS EPD

Profile EPD

Add Sire EPD **Print Sire EPD**

Show 10 entries Search:

Sire ID	Birth Weight EPD	Birth Weight Accuracy	Wean Weight EPD	Wean Weight Accuracy	Maternal Milk EPD	Maternal Milk Accuracy	Yearling Weight EPD	Yearling Weight Accuracy	Total Maternal EPD	Total Maternal Accuracy
No data available in table										

Showing 0 to 0 of 0 entries

Previous Next

* Indicates required field
© 2018 CHAPS

Figure 29: Bulls EPD

Figure 30 shows how the Bulls EDP is created. A sire is selected from the Profile section of the Bulls page, and all available attributes are put in the form. The same form appears with values when editing is done.

Add Sire EPD ×

Sire ID*

Birth Weight EPD <input type="text"/>	Wean Weight EPD <input type="text"/>	Total Maternal EPD <input type="text"/>
Birth Weight Accuracy <input type="text"/>	Wean Weight Accuracy <input type="text"/>	Total Maternal Accuracy <input type="text"/>
Yearling Weight EPD <input type="text"/>	Maternal Milk EPD <input type="text"/>	
Yearling Weight Accuracy <input type="text"/>	Maternal Milk Accuracy <input type="text"/>	

CANCEL **SUBMIT**

Figure 30: Sire EPD.

4.3.4. Cows

The Cows page is the next page after the Bulls page. Figure 31 shows the basic structure for the Cows page. The page is separated into two sections: Profile and EPD. The Profile section is described here. This page has the cow's listing which can be printed with the Print Cow button. There are three cow types: Active, Culled, and Pedigree; these types can be used to filter the cow's listing.

Cow ID	Dam of Cow	Sire of Cow	Birth Date	Breed	Source	Status	Culled Reason	Reg. Name	Reg. No
098		Unknown	12-05-2018	LO	Raised	Active			
345	098	Unknown	12-05-2017	TH7	Raised	Active			
554		21	12-04-2018	MAN	Raised	Active			
78		21	12-02-2014	MNB	Raised	Active			
87		21	12-02-2014	KLO	Raised	Active			
980	AB45	1245	12-05-2018	MN	Raised	Active			
A123	345	21	07-30-2018	MAN	Raised	Active			
A234	AB45	A897	07-02-2018	MN	Raised	Active			
A24		1245	07-30-2018	NA	Raised	Active			
A589		1245	07-02-2018	KLN	Raised	Active			

Figure 31: Cows.

The form in Figure 32 pops up when the Add Cow button is clicked from the Cows page. A cow can be added by providing all the necessary information. A cow can also be added from the calf's Replacement stage; in that case, the calf's ID should be entered into the Cow Calf ID at Birth field, and all necessary information is auto populated. The same form appears when editing the animal's data.

Add Cow Profile
✕

Cow ID*

Cow Calf ID at Birth

Cow Birth Date*

Pick the Cow Breed

Cow Breed*

Sire of Cow*

Dam of Cow

Registration #

Registration Name

Electronic ID #

Date Entered Herd

Source

Raised
 Purchased

Status

Active
 Culled
 Pedigree

Profile Notes

Figure 32: Cow Profile.

Figure 33 shows the Expected Progeny Difference (EPD) section from the Cows page. On the EPD page, a vast amount of data about different traits is gathered for the cow and can be used to improve the beef cattle's genetics. The animal's EPD listing can be printed with Print Cow EPD button.

CHAPS Herd ID: TestHerd1 HERD SETUP SETTINGS BULLS **COWS** CALVES REPORTS ▾

COWS EPD

Profile
EPD

Add Cow EPD
Print Cow EPD

Show entries
Search:

Cow ID ▲	Birth Weight EPD	Birth Weight Accuracy	Wean Weight EPD	Wean Weight Accuracy	Maternal Milk EPD	Maternal Milk Accuracy	Yearling Weight EPD	Yearling Weight Accuracy	Total Maternal EPD	Total Maternal Accuracy
No data available in table										

Showing 0 to 0 of 0 entries

Previous
Next

* Indicates required field
© 2018 CHAPS

Figure 33: Cows EPD.

Figure 34 shows how the cow EDP is created. A cow is selected from the Profile section of the Cows page, and all available attributes are put in the form. The same form appears with values if editing is done.

Add Cow EPD
✕

Cow ID*

Birth Weight EPD

Wean Weight EPD

Total Maternal EPD

Birth Weight Accuracy

Wean Weight Accuracy

Total Maternal Accuracy

Yearling Weight EPD

Maternal Milk EPD

Yearling Weight Accuracy

Maternal Milk Accuracy

CANCEL
SUBMIT

Figure 34: Cow EPD Creation

4.3.5. Calves

The Calves page comes after the Bulls and Cows pages and before the Reports page. This page has four different stages for the calf: Birth, Weaning, Background, and Replacement. Each stage is discussed, in detail, in the following sections.

4.3.5.1. Birth Stage

Figure 35 shows the Birth Stage page, the stage where cattle are before they are weaned, and this period ideally lasts 7-8 months. The correct weaning time depends not only the cow's age, but also on the cow's condition. This page lists all calves and can be printed with the Print Calves button. A calf is first added to the Birth Stage; subsequently, it goes to later stages.

The listing for a calf in the Birth Stage has a button called Wean it. A calf in the Birth Stage can be weaned directly without going to the Weaning Stage. In fact, this feature repeats the Weaning Information creation; therefore, the form used is the same as the one for Weaning Stage. A weaned calf is marked as "Weaned" in the Birth Stage.

CALVES

Birth
Weaning
Background
Replacement

Add Calf
Print Calves

Show entries
Search:

Calf ID	Dam ID	Sire ID	Birth Date	Birth Weight	Breed	Calving Ease	Sex	Cow Age	Reg Name	Reg No	Elec ID	Action
98	554	21	05-09-2018		anMAN	3	Bull	0				Wean it
A001	C1	B1	02-18-2016		AAAA	1	Bull	1				WEANED
A002	C2	B290	02-18-2016		BBBB	1	Heifer	1				WEANED
A003	C3	B3	02-18-2016		CCCC	1	Steer	1				WEANED
A004	C1	B290	02-18-2016		BBAA	1	Miscellaneous	1				WEANED
A005	C1	B3	02-18-2016		CCAA	1	Heifer	1				WEANED
A007	C2	B1	02-18-2016		AABB	1	Steer	1				WEANED
A008	C1	B3	02-18-2016		CCAA	1	Bull	1				WEANED
A0088	C2	B290	03-11-2009		BBBB	4	Bull	0				WEANED
A009	C1	B3	02-18-2016		CCAA	1	Heifer	1				WEANED

Showing 1 to 10 of 38 entries

Previous
1
2
3
4
Next

Figure 35: Birth Stage.

The form to create a calf is shown in Figure 36. There are two fields, Dam ID and Sire ID, in the form which populates the list from the Cows and Bulls pages. A calf cannot be added unless its male and female parents have been added to the Cows and Bulls pages.

ADD CALF
✕

Calf ID*

This is a required field

Birth Date*

Dam ID*

Calving Ease*

Sire ID*

Registration #

Calf Breed

Registration Name

Sex*

Electronic ID

Birth Weight

Birth Notes:

* Indicates required field

Figure 36: Calf Creation.

4.3.5.2. Weaning Stage

A calf is ideally weaned after 7-8 months when the conditions are good for it. Weaning is the process of acclimating the calf to a food source that is not its mother's milk. Figure 37 shows the listing for weaning information; this material can be printed with the Print Weaning Info button.

Each listing contains three buttons to perform different processes to sell, move to background, or move to replacement from weaning. If any processes have been performed, then

rest two buttons are disabled, and other one(s) is (are) marked. The operation for the move to background button (Move to BK) is a repetition of the Background Stage information creation, and the same form is used. The same thing goes for the move to replacement (Move to RP) button which imitates the Replacement Information's creation feature.

CHAPS Herd ID: TestHerd1 HERD SETUP SETTINGS BULLS COWS **CALVES** REPORTS ▾

CALVES

Birth **Weaning** Background Replacement

Add Weaning Info **Print Weaning Info**

Show 10 entries Search:

Calf ID	Date Weighed	Weight	Hip Height	Frame Score	Adjusted 205 Day Weight	Contemp. Group	Manage Code	Muscle Grade	Actions
A001	11-02-2016						1 - Months Creep Fed	M1 - Medium Frame, Thickness 1	sell it Move to BK Move to RP
A002	11-02-2016						1 - Months Creep Fed	Unknown	sell it Move to BK Move to RP
A003	04-06-2017	78.9			78.45		K - Foster Calf - Twin Calf	Unknown	sell it Move to BK Move to RP
A004	10-13-2017	5			53.18		A - Cow did not calve but retained in herd	Unknown	Disabled Disabled Disabled
A005	10-13-2017						X - Incomplete Record	Unknown	Disabled MOVED TO BK Disabled
A007	10-13-2017						X - Incomplete Record	Unknown	Disabled MOVED TO BK Disabled
A008	10-13-2017	100	100	28.9	85.48		8 - Misc Private Codes	Unknown	sell it Move to BK Move to RP
A0088	10-11-2018	56.5					6 - Months Creep Fed	Unknown	sell it Move to BK Move to RP

Figure 37: Weaning Stage.

The form used to enter the calf into its weaning stage is shown in Figure 38. The Weight and Adjusted 205 Day Weight fields are very important; these fields directly depend on three other fields: Calf ID, Manage Code, and Date Weighed.

Add Weaning Information
✕

Calf ID*	Manage Code	Contemp. Group
<input type="text" value="Search calf ID"/>	<input type="text"/>	<input type="text"/>
Muscle Grade	Lot No.	Status
<input type="text" value="Unknown"/>	<input type="text"/>	<input type="text"/>
Date Weighed*	Weight	Adjusted 205 Day
<input type="text" value="MM-DD-YYYY"/>	<input type="text" value="Decimal permissible"/>	<input type="text" value="00"/>
Date Measured	Hip Height	Frame Score
<input type="text" value="MM-DD-YYYY"/>	<input type="text" value="e.g 30.20"/>	<input type="text" value="e.g 30.20"/>

Notes

* Indicates required field

Figure 38: Weaning Creation.

4.3.5.3. Background Stage

This stage is the intermediate time between the Weaning and Feedlot Stages. Figure 39 shows the calf listing in the Background Stage. There are also three stages within the Background Stage: Receiving, Intermediate, and Final. The information can be printed with the Print Background Info button. In the future, there will be another button in same row in order to move a calf from Background to Replacement.

CHAPS Herd ID: TestHerd1 HERD SETUP SETTINGS BULLS COWS **CALVES** REPORTS

CALVES

Birth
Weaning
Background
Replacement

Add Background Info
Print Background Info

Show 10 entries Search:

Calf ID	Entry Date	Weight	Hip Height	Frame Score	Stage	Actions
A001	06-05-2019				Receiving	Move To FL
A005	02-01-2017	254	45	4.17	Receiving	MOVED
A005	02-02-2017	254	45	4.16	Intermediate	MOVED
A005	12-06-2017	254	45	2.39	Final	MOVED
A007	01-03-2017	254	100	31.05	Receiving	MOVED

Showing 1 to 5 of 5 entries Previous 1 Next

© 2018 CHAPS

Figure 39: Background Stage.

The form to Add Background Information is shown in Figure 40. This stage needs to be selected while entering data for the Background Stage. The Calf ID list comes from the immediate Weaning Stage.

Add Background Information ✕

Calf ID*

Stage*

Background Date*

Contemp. Group

Status

Weight

Hip Height

Frame Score

Notes

* Indicates required field

CANCEL
SUBMIT

Figure 40: Background Stage.

Each listing has a button named Move to FL (feedlot) in the Background Stage. After selecting Move to FL, the popup box shown in Figure 41 appears. Feedlot form also needs some economically viable attributes that could improve future beef production.

Add Feedlot Information ✕

Calf ID* <input type="text" value="A001"/>	Stage* <input type="text" value="Select One"/>	Feedlot Date* <input type="text" value="MM-DD-YYYY"/>
Weight <input type="text"/>	Hip Height <input type="text"/>	Frame Score <input type="text"/>
Back Fat <input type="text"/>	Rib Eye <input type="text"/>	Marbling <input type="text"/>
Status <input type="text"/>	Condition <input type="text"/>	

Notes

* Indicates required field

Figure 41: Feedlot Form.

4.3.5.4. Replacement Stage

The Replacement Stage is where calves are kept for future production. If the calves are fertile, then they are moved to the Bulls or Cows Sages, depending on their gender. If the calves are not fertile, then they are moved to the Background Stage and eventually to the Feedlot Stage. The forms to transfer a calf from Replacement Stage to the Bulls or Cows tables are the same as for the Bulls and Cows pages.

Figure 42 shows a listing for calves in the Replacement Stage. The listing can be printed with the Print Replacement Info button. In the future, there will be another button in same row to move a calf from the Replacement stage to the Background Stage.

CHAPS Herd ID: TestHerd1 HERD SETUP SETTINGS BULLS COWS **CALVES** REPORTS -

CALVES

Birth
Weaning
Background
Replacement

Add Replacement Info
Print Replacement Info

Show entries
Search:

Calf ID	Entry Date	Weight	Back Fat	Frame Score	Hip Height	Pelvic Area	Stage	Rib Eye	Condition	Actions
A006	02-01-2017	523		108.42	254		Receiving			Make Cow
A006	08-01-2017	523		109.91	254		Intermediate			Make Cow
A009	01-01-2017						Final			Make Cow
A009	01-01-2017			100	100		Final			Make Cow
A009	01-02-2017						Receiving			Make Cow
A009	01-02-2017						Intermediate			Make Cow
A009	02-01-2017	523		108.42	254		Final			Make Cow
A010	04-01-2018	125		31.64	100		Receiving			Make Cow
A011	01-04-2017	250		31.79	100		Receiving			Make Cow
A011	07-05-2017	250		31.01	100		Intermediate			Make Cow

Showing 1 to 10 of 12 entries

Previous
1
2
Next

Figure 42: Replacement page.

The form to add a calf in the Replacement Stage is shown in Figure 43. A calf needs the Calf ID, Stage, and Replacement Date fields to be added to the Replacement Stage.

Add Replacement Information



Calf ID*	Stage*	Status
<input type="text" value="Search calf ID"/>	<input type="text" value="Select One"/>	<input type="text"/>
Replacement Date*	Scrotum Date	Pelvic Date
<input type="text" value="MM-DD-YYYY"/>	<input type="text" value="MM-DD-YYYY"/>	<input type="text" value="MM-DD-YYYY"/>
Weight	Hip Height	Frame Score
<input type="text"/>	<input type="text"/>	<input type="text"/>
Back Fat	Rib Eye	Marbling
<input type="text"/>	<input type="text"/>	<input type="text"/>
Scrotum Circum	Condition	Pelvic Area
<input type="text"/>	<input type="text"/>	<input type="text"/>
Notes		
<input type="text"/>		

* Indicates required field

CANCEL **SUBMIT**

Figure 43: Replacement Creation.

4.3.6. Reports

The Reports page is one of the most important pages in CHAPS 2020. A report generated from CHAPS 2020 should match the reports from CHAPS 2000. CHAPS 2020 will have additional reports.

4.3.6.1. Herd Analysis Report

The Herd Analysis Report is the most detailed and important report in CHAPS 2020. As Figure 44 shows, this report requires the Bull Turn Out Date which was added to the Settings page. Depending on the Bull Turnout Date, the year would be selected automatically. This Herd Analysis Report is done yearly.

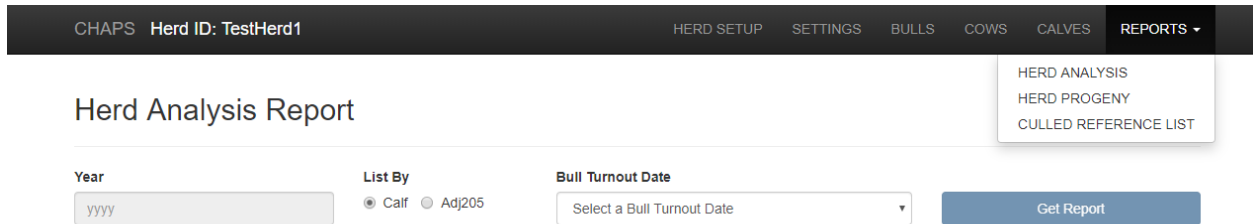


Figure 44: Herd Analysis Report.

The Herd Analysis Report is actually a combination of many sub-reports. The Herd Analysis Report from CHAPS 2020 creates an additional sub-report: Misc Calves Calf List. The rest of the sub-reports are the Bull Calves Calf List, Bull Calves Group Averages, Bull Calves Sire Averages, Bull Calves Cow Breed Averages, Heifer Calves Calf List, Heifer Calves Group Averages, Heifer Calves Sire Averages, Heifer Calves Cow Breed Averages, Steer Calves Calf List, Steer Calves Group Averages, Steer Calves Sire Averages, Steer Calves Cow Breed Averages, Sire Summary, Herd Summary, Herd Comparison, and NCBA-IRM-SPA Summary of Reproduction and Production Performance.

4.3.6.2. Progeny Report

The Progeny Report lists an animal's descendants throughout the years. There are two types of Progeny Report, long and short, which are visible in Figure 45. The short Progeny Report

shows some of the mother's attributes for her children. The long Progeny Report gives each animal's attributes along with the mother's necessary traits.

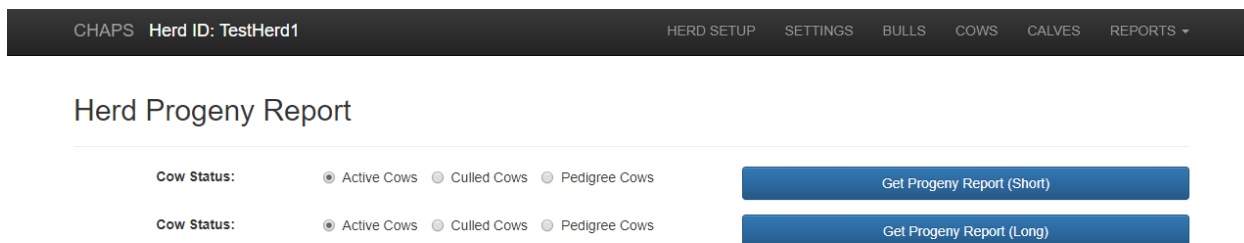


Figure 45: Progeny Report.

4.3.6.3. Culled Reference List

This report is new in CHAPS 2020. This report lists all the culled animals within the given date range as shown in Figure 46.

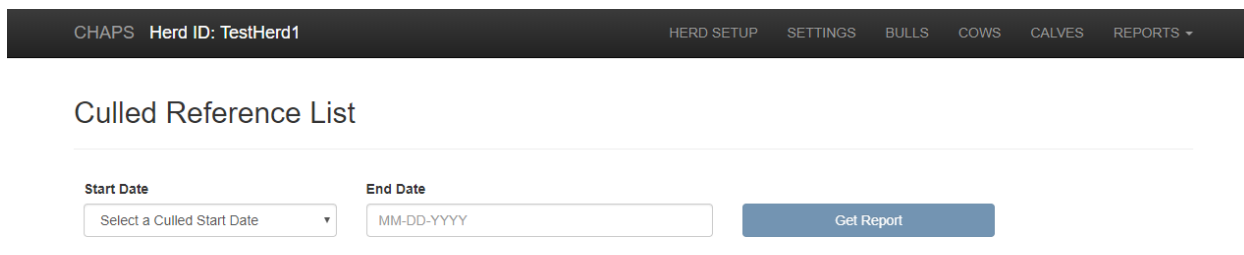


Figure 46: Culled Reference List.

4.3.7. Observation

The development of CHAPS 2020 is consistent in terms of the user interface, giving the application more clarity visually. The program is a browser application, and the user interface is responsive so that the frontend changes depending on the monitor or output device's size. The CHAPS 2000 user interface is static, and the sections are ambiguously segmented, leaving the customer with difficulties. The user interface for CHAPS 2020 is designed in such a way that it

follows the normal CHAPS operations and processes; therefore, the customer is more comfortable.

CHAPS 2020 is built with today’s modern technology. The application is operating-system independent. Java 7 is used as the server-side language, and MySQL 5.7 is used for the database, both of which were the latest options available at the time of development. The technology stack for CHAPS 2020 is discussed, in detail, in Chapter 5.

4.4. Comparisons Among CHAPS II, CHAPS 2000, and CHAPS 2020

Different aspects of CHAPS were shown in this chapter. Key findings were observed. I tried to summarize the key differences and inconsistencies for CHAPS II, CHAPS 2000, and CHAPS 2020. That information is presented in Table 1.

Table 1: Comparison of CHAPS II, CHAPS 2000, and CHAPS 2020.

Operating System	MS-DOS	Windows 2000	Windows 10
User Interface	Console based	Static	Responsive
Programming Language	No Information	Visual Basic	Java 7 and JavaScript
Type	Console	Desktop	Browser based
Platform	Windows and MacIntosh	Windows	Platform Independent
Feature Navigation	Static	Dynamic but not natural	Dynamic and natural
Database	Floppy Disk	Microsoft Access 97	MySQL 5.7
Development Phase	Phase I	Phase II	Phase II

5. CHAPS 2020 TECHNOLOGY STACK

CHAPS 2020 was designed as a standalone, native, browser-based web application. Thus, web technologies were utilized. The team used the most basic technologies to build the application. The following sections describe the technology stacks used for different parts of the application.

5.1. Architecture

An application certainly needs architecture to represent the system's common high-level abstraction and to maintain consistent communication among the application's stakeholders (Clements & Northrop, 1996). This architecture designing phase is the common ground where stakeholders are attached to a certain problem or solution. The CHAPS 2020 architecture is discussed in the following sections.

5.1.1. System Architecture

There are two kinds of system architecture for CHAPS 2020. One is a standalone system architecture, and the other one is a central system architecture.

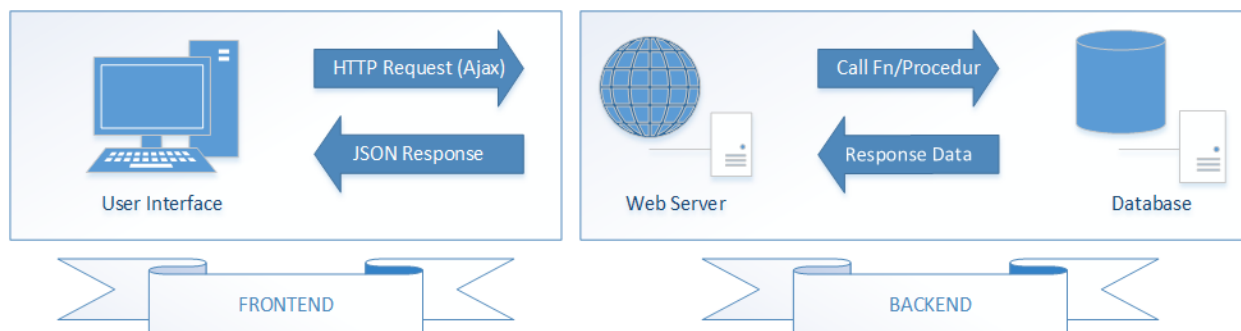


Figure 47: Standalone System Architecture & Request Flow.

The standalone system architecture, shown in Figure 47, only consists of the frontend and the backend. The frontend has many technologies inside it. This information is covered in Section 5.4. The backend also utilizes different technologies which are covered in Section 5.5. In a standalone system, a request which is generated from an HTML page goes to the JavaScript page. The JavaScript page processes the data and gets them ready for Asynchronous JavaScript and XML (AJAX) to forward the request to Java Servlet Page (JSP). JSP works as middle-ware for CHAPS 2020; JSP's sole activity is to dispatch data to the database or to catch data from the database. JSP dispatches the request to the database. After receiving a request from the JSP, the database does its operations and responds when necessary. Unlike the traditional web-application model, the team has placed all of the CHAPS 2020 functionality into MySQL-stored procedures and functions. Each technology is covered in the following sections.

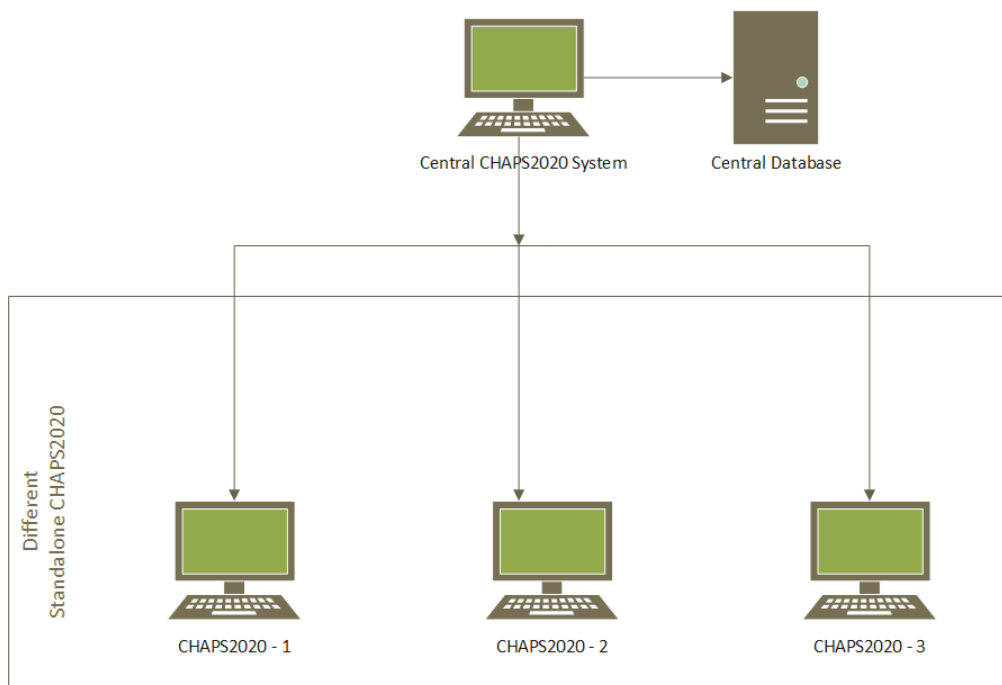


Figure 48: Central System Architecture.

Each standalone CHAPS 2020 program will be installed on the individual herd owner's machine. All herd information has to feed into a central database so that data specialists from North Dakota State University's Extension center can analyze all the data. The central application and the database will be at the NDSU Extension center. The central system's architecture is shown in Figure 48. The development of CHAPS 2020 did not reach the point where the central database can be implemented. The team has discussed the idea with the CHAPS 2020 stakeholders. The idea is that the owner of each standalone CHAPS 2020 program will push his or her herd information to the central database when the herd's data insertion is complete and the internet is available. Future developers will design and develop the exact technology for the central system.

5.1.2. Application Architecture: AJAX

The classic web-application architecture is a synchronous process and works as follows. The client triggers an HTTP request to the web server. The web server processes the necessary operations, communicates with the database, talks to various legacy systems, etc., depending on the type of request. The server generates the HTML page and returns it to the client. The architecture does what is supposed to do while holding the client idle until the client gets a response. The classic web-application model creates a huge user-experience caveat and a longer user-perceived latency.

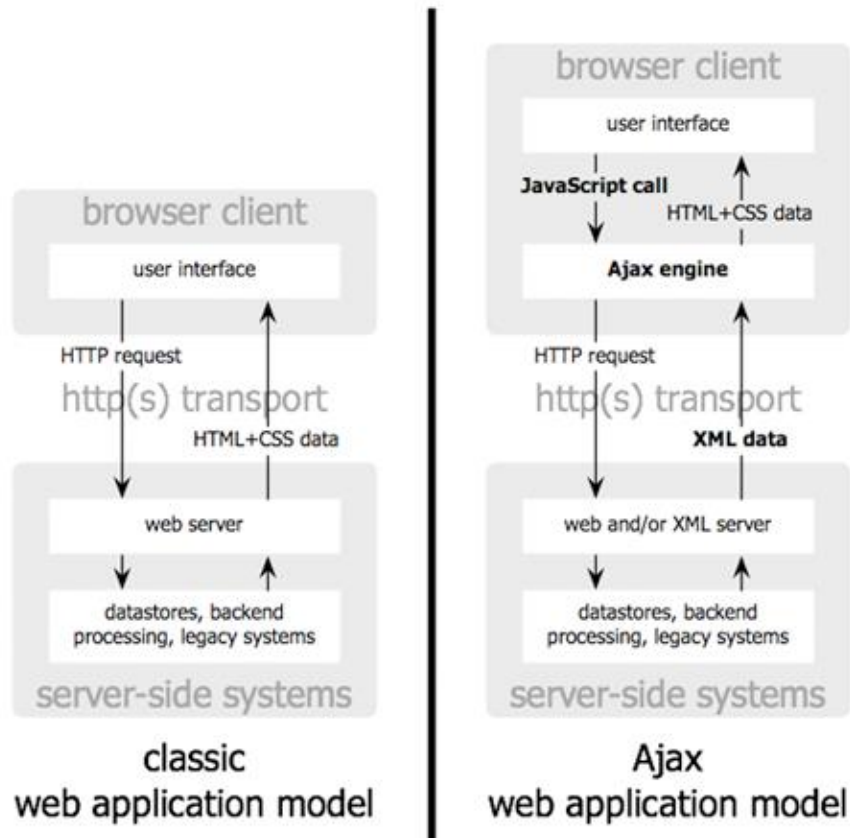


Figure 49: Classic vs. AJAX Web Architecture.

AJAX stands for Asynchronous JavaScript and XML; it is a dynamic and interactive web technology which provides the feel of a desktop application. With the AJAX architecture, an HTTP request cannot be dispatched directly to the web server; rather, the request must go through the AJAX engine. AJAX can send requests and receive responses in various formats, including JavaScript Object Notation (JSON), XML, HTML, and text files. Asynchronous character is the main advantage of AJAX, which leads to communicating with the web server, manipulating data, and updating a page or part of it without refreshing the page. CHAPS 2020 benefits greatly from this AJAX characteristic. The entire classic and AJAX web architecture is illustrated in Figure 49.

Teo et al. (Teo, Oh, Liu, & Wei, 2003) conducted comprehensive research about user interactions with commercial web applications, and their results suggested that, if we can increase the interactivity of a web application, it enhances the satisfaction level for that web application. Increasing the user's interactivity was the driving factor behind choosing AJAX as web-application architecture for CHAPS 2020. AJAX also reduces the traffic's travel time between the client-side and server-side communication, and the response time is much faster than with traditional web technologies (Paulson, 2005). The team used AJAX for the entire application; no other communication technology was utilized for client- and server-side communication.

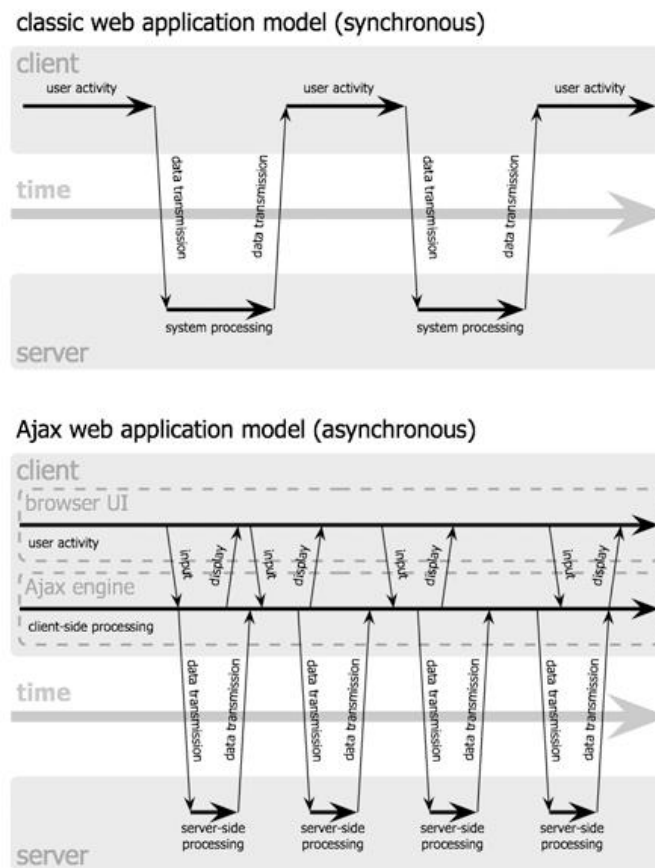


Figure 50: Latency between classical vs Ajax web architecture.

Figure 50 shows an illustration of the latency with the classic and AJAX web models. The figure's upper part illustrates that user activity stops while the server is processing functionality or communicating with the database. With a lot of processing, the client has to sit idle for a longer period of time until the server responds. The team observed this problem in CHAPS 2000 while creating a report for a data-extensive herd. There is a different scenario in lower part of the figure. User activity is always active on the client-side regardless of the server-side operation. The user is always live in the browser no matter what the herd size is.

5.2. Use Case Diagram

A Use Case Diagram is a type of Unified Modeling Language (UML). It helps to identify a system's actors and the use cases for those actors. It gives application clarity and a top-level view about what an actor can and cannot do, and it also shows which features are extended from which feature. In CHAPS 2020, the actor is a customer; the program has different use cases which are shown in Figure 51.

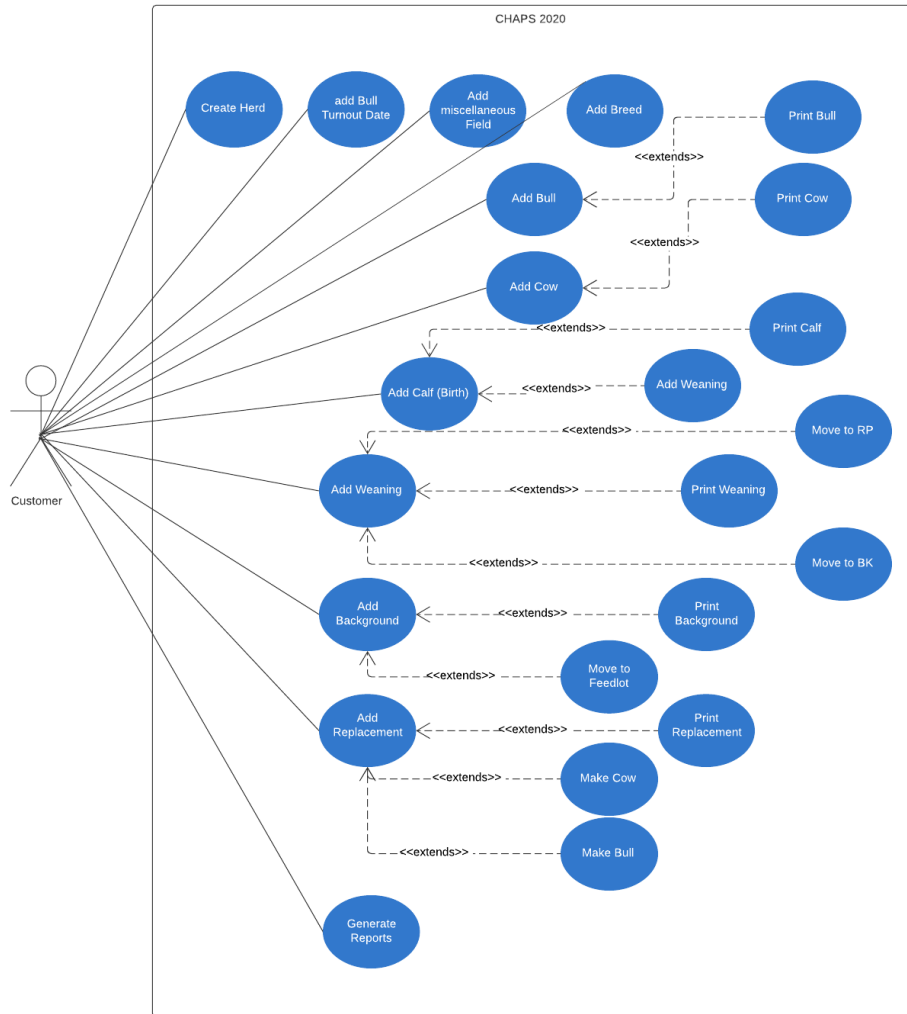


Figure 51: Use Case Diagram of CHAPS 2020.

5.3. Sequence Diagram

A sequence diagram is a useful medium to showcase the interaction logic between objects within the system and with time order. The actor in Figure 52 is the customer, and this figure is an example of saving information for the system. The data-entering process is same across CHAPS 2020. Therefore, there is only an example of a sequence diagram to avoid repetitive work.

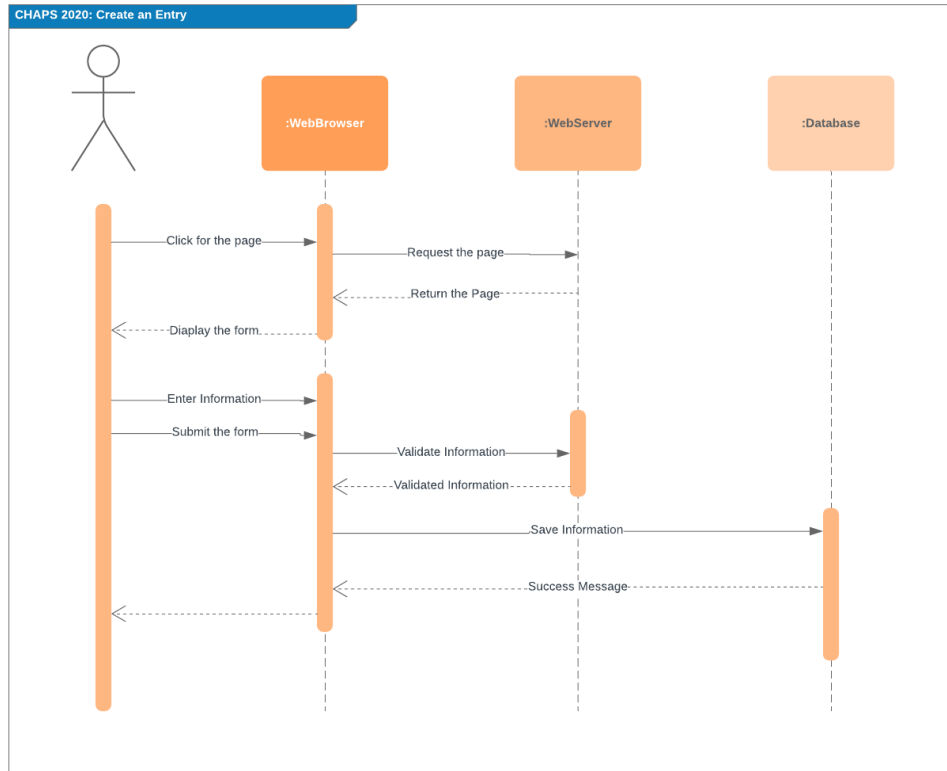


Figure 52: A Sequence Diagram of CHAPS 2020

5.4. Frontend

A good application needs a beautiful, clean, and consistent interface. No matter how complex the business solutions are, programmers need to create an easily demonstrated frontend; otherwise, customers will not be attached, and the entire purpose may fail. The frontend interface may seem easy compared to the backend's complexity, but there many great examples of failure due to a lack of investment for the user interface.

5.4.1. HTML

HTML is short for Hypertext Markup Language; it is a standard markup language to create web pages for web applications. Web pages are constructed with the building blocks of HTML semantic elements which are described by tags and which are written by using angle brackets around the tags. Web browsers read HTML pages from local source of the from the

application server into multimedia web pages. Browsers do not display the HTML tags but use the tags to interpret the page's content in a structure they tags are coded. HTML alone does not do much except to display the content as coded in different tags. Using Cascading Style Sheets (CSS) and JavaScript brings style and interactivity to the pages, attributes which are discussed in the following sections. The team has used the semantic HTML tags all over the application pages.

5.4.2. CSS

CSS is short for Cascading Style Sheets and is used to provide some life for the barren HTML tags. CSS designs the layout for the HTML page. HTML deals with the page's structure, and CSS deals with the tag's style. CSS controls the appearance of multiple HTML pages by including CSS with it. CSS can be used in three different ways on a page. Inline CSS is used inside an HTML tag and only affects the concerned tag. Inline CSS's capacity is limited to that tag where it is used. Internal CSS is placed right on the page with the affected interface. Internal CSS's scope is the page where it is placed. External CSS is placed externally, and it has an application-wide scope to influence. The team has used all three CSS types in the application, but most of the utilized styles are external.

5.4.3. Bootstrap

Bootstrap is a CSS framework that is widely used by designers. It helps to avoid the CSS boilerplate code. Consistency is the fundamental principle of its introduction, and it ensures style consistency throughout the web pages, regardless of the developer. The development speed is another reason for using Bootstrap. One does not need to know every style by him or herself; the

developer can use Bootstrap's predefined design templates and classes, putting those classes exactly where they fit. The biggest advantage of using Bootstrap is that it has a predefined class for a responsive design, which the team used to achieve the responsive web design. The next section describes the responsive web design and its benefits.

5.4.4. Responsive Web Design

Responsive Web Design is also referred to as RWD in the industry; it is a design approach which enables a web page to automatically scale its content and elements, depending on the orientation or size of the screen or the device type on which it is being seen. The device could be an oversized desktop monitor or a laptop monitor with multiple orientations. RWD eliminates the unnecessary resizing, scrolling, zooming, or panning that occurs with sites that have not been optimized for different resolutions.

The team has used Responsive Web Design (RWD) approaches for CHAPS 2020 in order to facilitate the application's view on desktops or laptops with different monitor sizes and orientations. RWD design fulfills the Application Interface from the Requirements section. A very good example is using CHAPS 2020 on a Microsoft Surface device which has multiple orientations. The application interfaces will resize once the orientation changes.

5.4.5. JavaScript

JavaScript is a commonly used a client-side scripting or programming language which implements interactive and complex things on web pages. If HTML and CSS are considered as the skeleton and body, respectively, then JavaScript is the "life" of that body. HTML, CSS, and JavaScript are the triad of almost every page on the web. When a user requests an HTML page

that has JavaScript, the script is sent to the browser, and it's up to the browser to do something with that script.

CHAPS 2020 depends heavily on JavaScript (Version XX) and, more specifically, its popular library, JQuery, which is discussed in the next section. The team has achieved most of the frontend interactivity and functionality using JavaScript. Frontend functionality includes choosing a herd, opening and closing a modal, entering and displaying form data, displaying data in the list client-side form validation, etc.

5.4.6. jQuery

jQuery is a JavaScript library, and it was introduced to ease the writing code over JavaScript's native syntaxes. JavaScript has a reputation of being cumbersome for developers, inconsistent for browser support, difficult to debug, and having intimidating syntax for newcomers. All the things that make JavaScript seem difficult are what make it so powerful. Although it does not appeal to novice developers to gauge it, jQuery has much more acceptance (Lengstorf, 2010).

Using jQuery reduces the development time by writing small code, handling cross-browser differences, manipulating Document Object Model (DOM) easily, and most importantly making the AJAX call much simpler. The team members did not use native JavaScript syntax in the application; they depended on jQuery syntax for fast implementation.

5.5. Backend

The request was for CHAPS 2020 to be platform independent. Platform independence is a vague idea because it does not describe what the platform is. "Platform" could mean the

operating systems, browsers, or devices. The CHAPS 2020 requirement was to make the application operating-system independent. The early engineers opted to choose Java instead of other cross-platform programming languages. It was a wise decision because Java's slogan indicates platform independence: "Write once, run anywhere" (Blom et al., 2008, p. 206).

5.5.1. Server

The team used an Apache Tomcat server, an open-source, Java servlet container which implements several Java Enterprise Edition specifications, such as Java Servlet, JSP, and WebSocket, and facilitates an environment where Java code can run. Apache Tomcat runs a Java Virtual Machine (JVM) where every request that comes from a browser is processed. Apache Tomcat 7 was used for CHAPS 2020.

5.5.2. Server-Side Language

The development started with Java 7 which was the latest version 2013. Java is a server-side programming language. There are many technologies, based on Java, to perform specific operations. Because CHAPS 2020 is web and browser based, the team used Java Serverlet Page (JSP). The JSP technology simplifies web development with Java by generating web pages dynamically. JSP is a self-sufficient technology to develop any Java-based web application, but in the industry, JSP is used with servlets. A servlet is a simply a class which responds to a server request, typically HTTP (Hypertext Transfer Protocol) ones. With a typical Java web application, the servlet handles the request, and JSP displays pages in the browser.

However, CHAPS 2020 used JSP in different way than the industry does. The team did not use JSP as a means for web pages; rather, it established a connection to the database and

processed the data in it. From the frontend, a request is made to the JSP file directly, and the JSP file processes the requested or responding data from database to the frontend, respectively. In the next section, the database and how it works in CHAPS 2020 are discussed.

5.5.3. Database

In the industry, it is often said that the database is the soul of an application. This fact is very true for CHAPS 2020. The team has chosen a MySQL (version 5.7) database for CHAPS 2020 because it is basically an open-source Relational Database Management System (RDBMS) to manage CHAPS 2020's relational database. One major advantage of using this database is that it is cross-platform compatible. Because CHAPS 2020 is a data-extensive application, it needs to have a database that supports large-scale data. MySQL can support almost 50 million data records (Chervenak, Palavalli, Bharathi, Kesselman, & Schwartzkopf, 2004). The team has used MySQL Community Edition for CHAPS 2020 as a server machine which works as a service after installation.

The critical problem for CHAPS 2000 was not having unique identifiers for sires, cows, or any cattle. The unique identifier was introduced into the cattle table as the CHAPS ID during the CHAPS 2020 database development. The CHAPS ID is referenced to other tables in the database to track its transformation. Some attributes are calculated at run time in CHAPS 2020, and the CHAPS ID is used during those calculations.

The team has leveraged the power of stored procedures and functions to increase the application's performance. Once the procedures and functions are created, they are stored in the database. MySQL compiles them on demand, and after compilation, MySQL puts the data into a cache for a single connection. If the application uses the same stored procedure multiple times

during a single connection, then MySQL utilizes the cached version. CHAPS 2020 calls a single stored procedure multiple times within one connection while creating the Herd Analysis Report, and CHAPS 2020 has a plan for entering multiple data sets in a single action. For all these cases, using the stored procedure and function will definitely help the application run faster.

5.6. Testing

Software testing is an essential activity for software engineering. Testing helps to identify any errors or missing requirements in the application. Testing ensures the end product's quality to satisfy the customer's requirement. There are many testing techniques to utilize, and I will discuss the tests done during phase III of the CHAPS development.

Beta testing is a type of acceptance testing that is manually done without the help of an automated tool. Beta testing is also known as the "field testing" which takes place in a real environment. The beta-testing environment creates the customers' ideal conditions during production. The beta test is run by customer groups who would actually use the application. For CHAPS 2020, the team has chosen two consumer types to test the application: customers who have used CHAPS 2000 and new customers for CHAPS 2020. When this master's paper was written, CHAPS 2020 was being beta tested.

5.7. Deployment

Herd owners installed CHAPS 2000 by using a CD. CHAPS 2020 stakeholders decided to not only allow installation with a DVD, but also to have a downloadable file. Because the development is not finished, the complete installation process has not been implemented. For the time being, manual installation is used. I download and install Apache Tomcat and MySQL Community Server on the device. Then, I run SQL scripts to create tables, stored procedures, and

stored functions. Finally, I place the application's Web Application Resource (WAR) file on the Tomcat server.

6. REQUIREMENT FULFILLMENT

At this point, it is very important to verify that all the requirements have been fulfilled before the development of CHAPS 2020 concludes. The following sections explain how all the requirements have been satisfied.

6.1. Unique Identification

The critical problem for CHAPS 2000 was not having unique identifiers for sires, cows, or cattle. The unique identifier was introduced in the cattle table as the CHAPS ID during the CHAPS 2020 database development. The CHAPS ID is referenced to other tables in the database in order to track the animal's transformation. Some attributes are calculated at run time in CHAPS 2020, and the CHAPS ID is used during those calculations.

6.2. Report Generation

The application is built as a standalone program and works without the internet. Therefore, Herd Analysis Reports can be generated from the program without internet access and an NDSU data-management specialist. Only a few clicks are required to generate reports, indicating the application's simplicity.

6.3. Application Interface

Responsive Web Design is used with CHAPS 2020 in order to facilitate the application's appearance on desktops or laptops with different monitor sizes and orientations. RWD fulfills the versatility of the CHAPS 2020 application interface. A very good example is using CHAPS 2020

on a Microsoft Surface device which has multiple orientations. The application interface resizes once the orientation changes.

6.4. Application Platform

To achieve the application's cross-platform capability, the team used Sun Microsystems' Java; the Java slogan is "Write once, run anywhere" (Blom et al., 2008). However, CHAPS 2020 utilized JSP in a different way than the industry. The team did not use JSP as a mean for web pages; rather, the team established a connection to the database and processed the data in the database. From the frontend, the team requested a JSP file directly, and the JSP file processed the requesting or responding data from the database and to the frontend, respectively. For frontend operations, the team used JavaScript which is compatible with any operating system. By using Java and JavaScript, the team achieved compatibility for CHAPS 2020 across operating systems.

6.5. Architecture Design

Each standalone CHAPS 2020 program will be installed on the individual herd owner's machine. All installations will be operational independently. The central system's architecture is shown in Figure 48. The CHAPS 2020 development did not reach a point where the central database could be implemented. The idea is that the owner of each standalone CHAPS 2020 program will push his or her herd information to the central database when the herd's data insertion is complete and when internet access is available. Future developers will design and create the central system's exact technology for the central system.

6.6. Technology Stack

Technologies which require a smaller learning curve are necessary. That goal led earlier designers to choose basic technologies over the advanced one of a technology stack. For example, if the previous developers had chosen a specific framework, then it would narrow the choices of available graduate students who have the same skillset. Also, the utilized technologies were open source.

7. CONCLUSION

CHAPS 2020 responded to all application requirements during different parts of its development. Properly developing the CHAPS 2020 database helps to track each animal even if a single customer has multiple herds. CHAPS 2020 can now generate the existing reports from CHAPS 2000, in addition to other reports, with or without internet availability. The biggest achievement of CHAPS 2020 is its user interface. The program is browser based and is compatible with screens of varying shapes and sizes. CHAPS 2020 has a smaller learning curve, greatly benefiting customers who are not technology savvy.

It was challenging to design CHAPS 2020 in such a way that it incorporated platform independence. Another hurdle was coming up with an architecture that works with or without an internet connection. The development of CHAPS 2020 satisfied the architecture and platform requirements. The technology used for the phase III development of CHAPS was given great consideration so that future developers can lessen the learning curve. When this master's paper was written, CHAPS 2020 was in the beta version for testing.

8. FUTURE WORK

The phase III development of CHAPS 2020 is not complete, but the major portion of it is done. The deletion of any herd, sire, cow, or calf at different stages is not done, but that process was defined during the database's design. The deletion feature will not affect the existing database when the feature is implemented. Also, the transformation of a calf from the Background stage to the Replacement stage or from the Replacement stage to the Background stage is not done.

The application has been fully tested with the Chromium browser, and part of the additional work is to test the compatibility of other major browsers so that the application works well. The central database system has not been developed; the database will be created after the CHAPS 2020 development is completed. Also, importing a database from CHAPS 2000 and exporting data to CHAPS 2020 are not possible. Finally, the installation process for CHAPS 2020 is not quite complete, but it is under development.

REFERENCES

- Blasi, D. A. (2006). Record systems for beef stocker production enterprises. *The Veterinary Clinics of North America. Food Animal Practice*, 22(2), 475–492.
- Blom, S., Book, M., Gruhn, V., Hrushchak, R., & Köhler, A. (2008). Write once, run anywhere a survey of mobile runtime environments. *Grid and Pervasive Computing Workshops, 2008. GPC Workshops ' 08. The 3rd International Conference On*, 132–137. IEEE.
- Chervenak, A. L., Palavalli, N., Bharathi, S., Kesselman, C., & Schwartzkopf, R. (2004). Performance and scalability of a replica location service. *IEEE*, 182–191. <https://doi.org/10.1109/HPDC.2004.27>
- Clements, P. C., & Northrop, L. M. (1996). *Software Architecture: An Executive Overview*. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst.
- Hong, M. K. H., Yao, H. H. I., Pedersen, J. S., Peters, J. S., Costello, A. J., Murphy, D. G., ... Corcoran, N. M. (2013). Error rates in a clinical data repository: lessons from the transition to electronic data transfer—a descriptive study. *BMJ Open*, 3(5), e002406.
- Lengstorf, J. (2010). Introducing jQuery. In *Pro PHP and jQuery* (pp. 3–23). Springer.
- Paulson, L. D. (2005). Building rich web applications with Ajax. *Computer*, 38(10), 14–17.
- Ramsay, J. M., Hanna, L. L. H., & Ringwall, K. A. (2014). Enhancing the identification of error checking in the Cow Herd Appraisal Performance Software (CHAPS) benchmarks using SAS procedures. *2014 North Dakota Beef Report*, 33.
- Ramsay, J., Tisor, L., Hanna, L. H., Hossain, N., Latimer, J., & Ringwall, K. (2017). CHAPS 2020: Rewriting the Cow Herd Appraisal Performance Software (CHAPS). *2017 North Dakota Beef Report*, 17.
- Ramsay, J., Tisor, L., Hanna, L. H., & Ringwall, K. (2017). Cow Herd Appraisal Performance Software (CHAPS): 15 years of beef production benchmarks. *2017 North Dakota Beef Report*, 21.
- Ringwall, K. M. (2004). *BeefTalk 215: CHAPS: basic information and decision-making power*. Dickinson Research Extension Center.
- Ringwall, K A, Berg, P. M., & Boggs, D. L. (1992). A computer program for appraising and increasing productivity in beef cattle. *Veterinary Medicine*.
- Ringwall, Kris Alan, & Berg, P. M. (1990). CHAPS II: cow herd appraisal performance software. *Dickinson Research Extension Center*.
- Teo, H.-H., Oh, L.-B., Liu, C., & Wei, K.-K. (2003). An empirical study of the effects of interactivity on web user attitude. *International Journal of Human-Computer Studies*, 58(3), 281–305.
- Westfall, L. (2005). Software requirements engineering: what, why, who, when, and how. *Software Quality Professional*, 7(4), 17.