****

**NORTHWEST STUDENT SUCCESS CENTER REPORTING APPLICATION**

**System Documentation and Final Report**

**Team: Nightmare**

**Prepared By:**

**Abhiram Madugula S534093**

**Akhila Gandra S533621**

**Divyaharshini Bheemireddy S533899**

**Haritha Atmakuri S534049**

**Razeena Rao Nemarugommula S533903 Suresh Chandra Peddinti S533575**

**Table of Contents**

|  |  |
| --- | --- |
| 1. **Introduction** |  |
| 1. Problem Statement | 2 |
| 1. Proposed Solution | 2 |
| 1. **Requirement Artifacts** |  |
| 1. Requirements Gathering | 2 |
| 1. Functional Requirements | 2 |
| 1. Non-Functional Requirements | 3 |
| 1. Use Cases | 4 |
| 1. Specification Requirements | 5 |
| 1. **Analysis and Design Artifacts** |  |
| 1. Software styles/Architectures | 6 |
| 1. Analysis Model (Database Diagram) | 6 |
| 1. Design Model of the System | 6 |
| 1. **Implementation and Testing Artifacts** |  |
| 1. Testing Methodologies and Evaluations | 7 |
| 1. **Deployment Artifacts for Users and Clients** |  |
| 1. User’s Guide for users and clients | 8 |
| 1. Installation Guide | 26 |
| 1. Execution Guide | 27 |
| 1. Project | 27 |
| 1. **Project Management Artifacts** |  |
| 1. Conclusion | 28 |
| 1. Risk List, Pros and Cons | 28 |
| 1. Summary of requirements completed | 28 |
| 1. Summary of requirements not completed | 28 |
| 1. Conclusion | 28 |

1. **Introduction**
2. **Problem Statement:**

In order to feed the cattle with proper amount of nutrients needed for their growth, using minimal cost of ingredients.

1. **Proposed Solution:**

The purpose of Ration Balancing app is to provide the animal the amount of feed that will supply the proper amount and proportions of nutrients needed for an animal to perform a specific purpose such as growth, maintenance. It makes the user life easy by just calculating the amount of feed when user gives specific requirement of nutrients and ingredients as an input.

**II. Requirement Artifacts**

**1. Requirements gathering:**

According to our client, we have been able to gather the functional and non-functional requirements that the application must have.

Any requirement which describes the behavior of the system i.e. what the system should do is considered as a functional requirement.

Any requirement which describes how the system performs and what are its limitations is considered as a non-functional requirement. Some examples of non-functional requirements are as following:

* Security
* Availability
* Confidentiality
* Integrity
* Reliability
* Efficiency

**2. Functional requirements:**

**Must have:**

1. User must have to select the weight of the cattle.
2. User must have nutrient requirements of different categories of diary animals.
3. User must have specified the ration range (min, max) of nutrients.
4. User must be able to edit the range (min, max) of nutrients.
5. User must have ingredients list and growth requirements of each cattle.
6. User must have to select the pound in ration (lb in ration) of each ingredient selected.
7. User must have to be notified whether the total in ration is in between the min and max values of nutrients.

**Should have:**

1. User should have in detailed formulae to calculate rations required to feed the cattle.
2. User should have to know the availability of ingredient specified.
3. User should be able to save the data of ingredients selected even after navigating to the other page.

**Could have:**

1. User could have provided statistics for each animal on the daily basis.
2. User could have listed any additions or supplements to add.
3. User could have the history of amount of animal fed for each cattle.
4. User could have saved the selected ingredients to feed the cattle such that it maintains the same diet for a particular period of time.

**Won’t have or Would:**

1. User would have determined the feed for the cattle based on the cost. This is not implemented in the current version and will be doing in the next version.
2. User would have the analysis of amount of feed for each cattle on a daily basis.

**3.Non-functional requirements**

**3.1 Security**

1. Newingredients with new nutrient composition can be added only by authorized users.

**3.2 Availability**

1. The application may not be available for users to access during maintenance.
2. The application may not be available for users to access in case of any technical issues.

**3.3 Confidentiality**

1. Admin account passwords shall be encrypted.
2. Admin account passwords shall never be displayed in the application.

**3.4 Integrity**

1. The application shall not alter the data from the user input once stored into the database.

**3.5. Reliability**

1. The selected ingredient proportion of nutrients can be rolled back to the original proportion from the database if any operation fails.

**3.6. Efficiency**

1. The results are shown with the ingredient weights within less than 5 seconds.
2. The nutrients proportion for the selected cattle category will be retrieved and displayed in less than 5 seconds.

**4. Use Cases:**

### **Use Case Diagram**

### 

**Use Case Scenarios**

1. **Cattle Settings**

* Primary Actor: Admin/User
* Users Story: Admin/User should select the growth requirement and cattle weight

1. **Nutrient Values**

* Primary Actor: Admin/User
* Users Story: By default, Admin/User can see nutrient values if he/she wants to change can increase or decrease by 0.5 range. Admin/user can even change directly in the textbox.

1. **Select Ingredients**

* Primary Actor: User
* User needs to select the ingredients, In split view controller user can see the ingredients selected in the other half of the page and enters the cost as it is a mandatory field.
* User after entering the cost of the ingredients clicks on the calculator button and result will be displayed.

**IV. Add Ingredients Page:**

* Primary Actor: Admin
* Admin has the capability to add new ingredients to the ingredients database.
* Admin should be able to log in to the add new ingredients page with his credentials and enter the mandatory nutrients percentage in order to add the new ingredient.
* Admin can see the added ingredient in the select Ration Balance page where the ingredients list will be displayed.

|  |  |
| --- | --- |
|  |  |

**5. Specification requirements**

1. **Platforms:**

* iOS
* Backendless

1. **Technologies:**

* Swift
* JavaScript

1. **Additional Libraries:**

* Swift Libraries

1. **Hosting Strategy:**

* Xcode 10.0 or Above
* Apple Ipad(5th Generation Preferable).
* Backendless Database.

1. **Windows/Linux Server Requirements:**

* RAM: 4 GB
* Storage: 50 GB
* Processor Type: Multi core
* No. Of Processor: 4

1. **Database Servers:**

* RAM: 4 GB
* Storage: 50 GB
* Processor Type: Multi core
* No. Of Processor: 4

**III. Analysis & Design Artifact**

**1. Software styles/architectures:**

The software style architecture is the process of steps that are taken place during the running of the application. For any application it follows the planning, designing and constructing phases. It is like a blueprint for the system, developing project and set the tasks that are necessary to be executed by design teams.

It is the process of converting software characteristics such as flexibility, scalability, reusability and security into a structured solution that meets technical expectations. Software characteristics describe the requirements and expectations of a software in technical levels.

It makes easy communication between team and client, able to make decisions early about design and allows reuse of design components between the project.

**2. Analysis model:**

There are different Analysis models and we are using Flow-based Analysis Model. This model provides an indication on how objects are transferred by a set of processing functions.

We are using the relational database in storing the data into the database. We have used the Entity-Relationship model in the lucid chart to represent our database.

**3. Design model of the system:**

The design model is mainly used in the implementation and testing phase. It is a model describing the achievement which serves as abstraction for the implementation and its source code. It helps in the documenting the software design system. The software architect is responsible for the integrity of the design model. When it comes to the use-case model, design model is the best option. It primarily sets the architecture of the design model. The architecture fulfills the design model of its purpose, deployment views and its logical process.

|  |  |
| --- | --- |
|  |  |

**E-R Diagram:**

A screenshot of a cell phone

Description automatically generated

**IV. Implementation and Testing Artifacts**

**1. Testing methodologies and evaluations:**

The application is tested with different use cases manually and client has tested different corner cases.

A screenshot of a cell phone

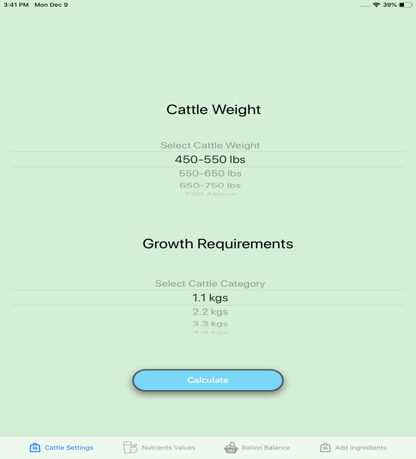
Description automatically generated

Please refer the attached document for list of Testcases

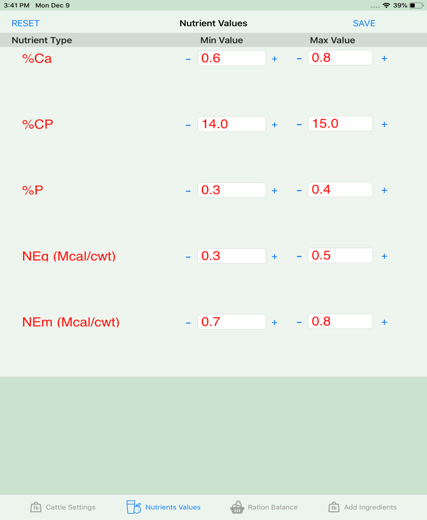
1. **Deployment Artifacts for Users and Clients**
2. **User’s Guide for users and clients**

**1) Cattle Settings Page: -** This is the landing page of our application. Here the user will select the cattle weight and its corresponding growth requirement for which we must calculate the ration balancing for the cattle.

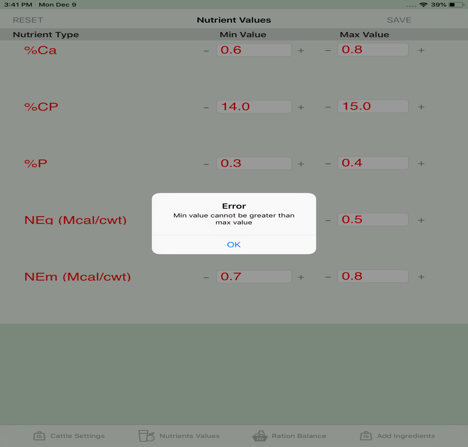
User must click on the calculate button in order to proceed forward. Without selecting any of the cattle weight or the growth requirement, it will throw an alert message saying, “Please select the cattle category”.

****

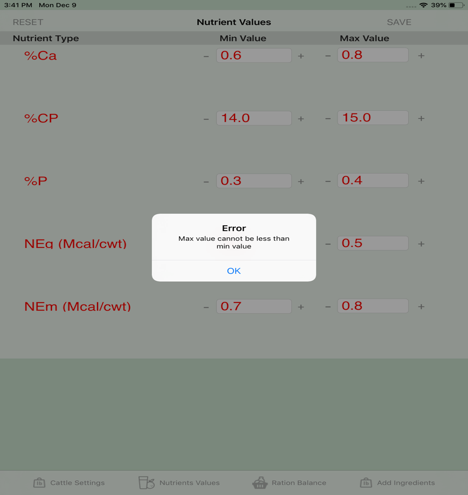
**2a) Nutrient Values Page:** - When user clicks on calculate button in the cattle settings page, it navigates to the Nutrient Values page. Here, the default nutrient values for a cattle weight and growth requirements combination is displayed from the database. The user can edit the nutrient values either directly in the text box or by clicking on ‘+’ or ‘-’ buttons (the value increases or decreases by 0.5).

****

**2b) Incrementing Nutrient Value: -** When the user tries to increase the min value greater than max value, it displays an error message saying, “Min value cannot be greater than max value”.

****

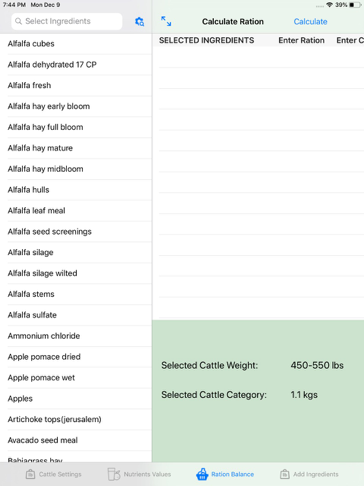
**2c) Decrementing Nutrient Value: -**When the user tries to decrease the max value less than the min value, it displays an error message saying, “Max value cannot be less than min value”.

****

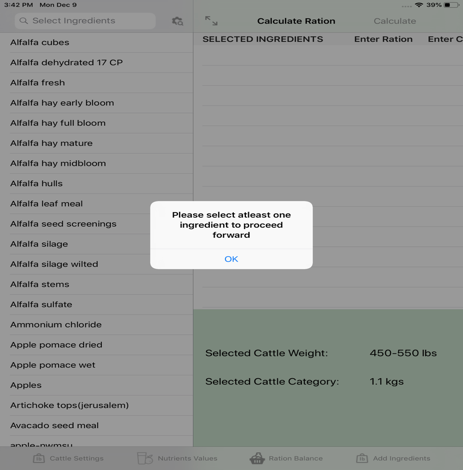
**2d) Reset Nutrient Values: -** The user can change min and max values of the nutrients based on the requirements. The values that are changed are for temporary purpose and user can get back to their previous values on clicking on the RESET button.

****

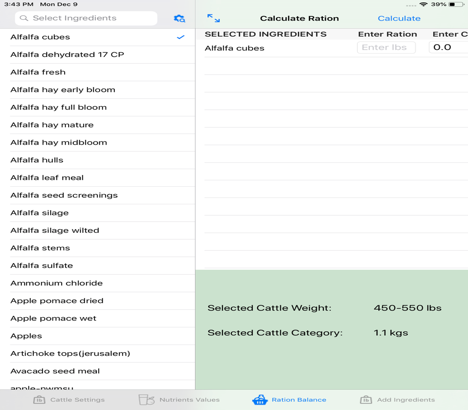
**3a) Ration Balance: -** This is the page where user will be balancing the amount of ration that he needs to fill for the cattle. In this page the user has to select the ingredients that he has to feed for the cattle. This page gives an option of selecting the ingredients out of 279 ingredients. The right side of the page displays the selected ingredients and the bottom of the right side displays the selected cattle weight and the cattle category.

****

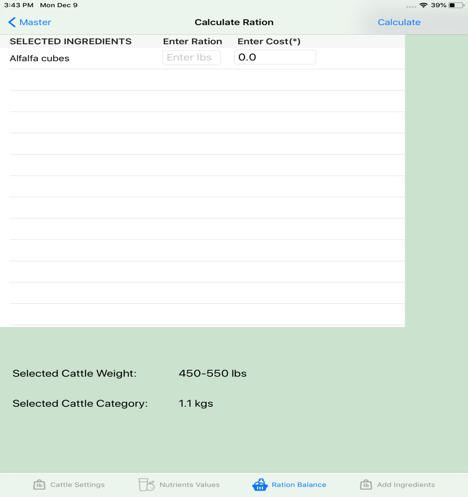
**3b) Select Ingredient Error Message: -** When user tries to click on the calculate button without selecting the ingredient, it will throw an error message saying that “Please select at least one ingredient to proceed forward”. The user must select at least one ingredient to make the calculation for the ration balancing.

****

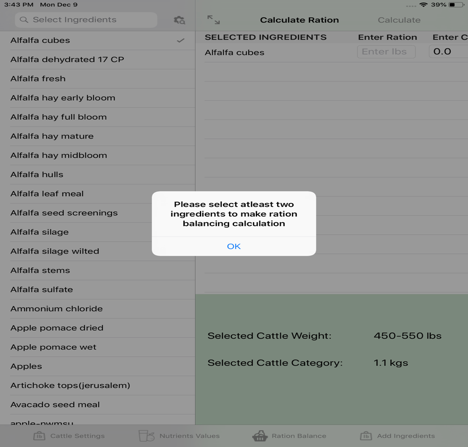
**3c) Selected Ingredient: -** When user clicks on a ingredient, the ingredient will be shown immediately on the right side of the page which is the ingredient that we have selected. On the left side, we can see a checkmark, which tells that it is the selected ingredient.

****

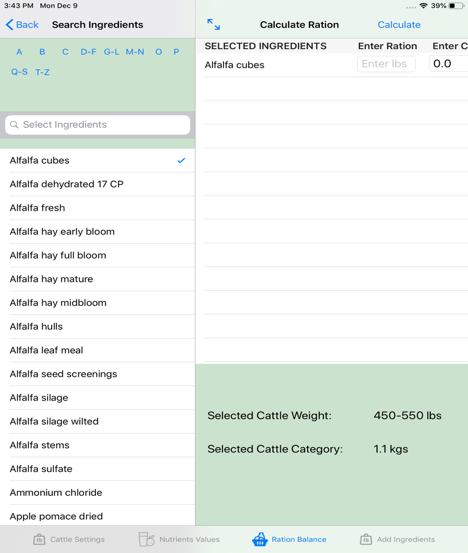
**3d) Selected Ingredient:** To see the selected ingredient in detail, the user can expand the page by clicking on the expand button on the top corner of the right-hand side of a page. User can navigate back to actual page by clicking on the back button named Master.

****

**3e) Ration Balance Calculation Error:** In order to make the ration balancing calculation, we must select two ingredients since there is no point in making the ration balance calculation with a single ingredient.

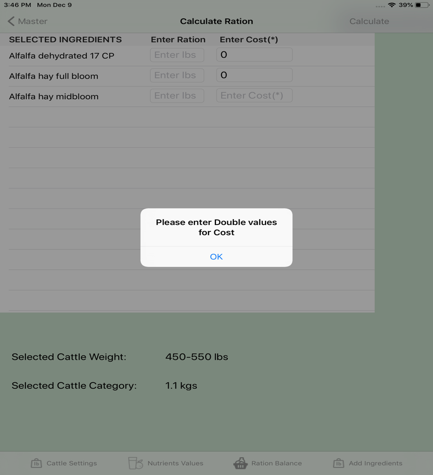
****

**3f) Advanced Search:** The user can make an advance search based on the alphabetical order which makes an easy way for searching a ingredient.

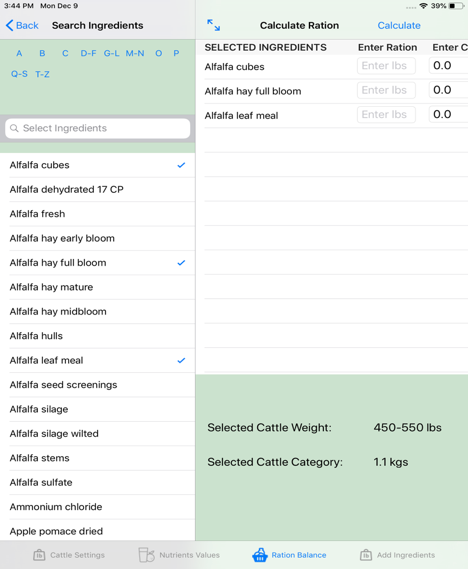
****

**3g) Non-Double values Error:** When user tries to click on the calculate button without entering the proper input for the cost, it throws an error saying that “Please enter double/integer values for the cost. It even throws the same error, if we try to enter the values that are neither integer or double values.

It even throws the error, if we don’t enter any values for the cost. The cost values are mandatory fields to enter. Without entering all the cost values, it doesn’t allow us to proceed forward.

****

**3h) Ration Balance with selected Ingredients:** This is how it looks like after selecting all the ingredients and with the cost values.

****

**3i) Result Page for the Ration Balance:** This is the result page of the ration balancing. It is divided into two halves named top part and the bottom part.

The top part consists of the results with ration in pounds of each ingredient that we must feed for the cattle. It also gives us the proportion in the percentage as well.

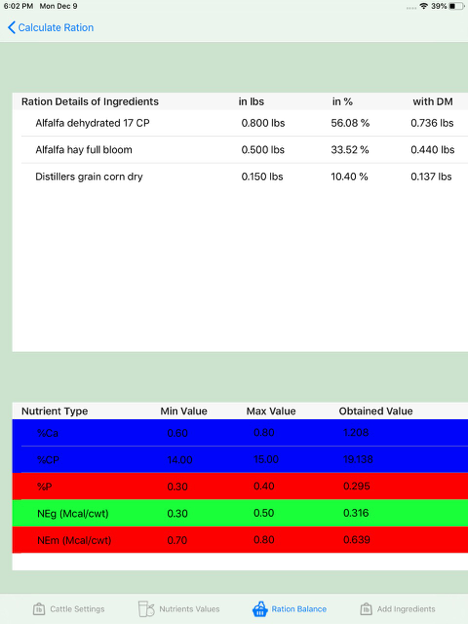
The below part of the page displays us the result, whether the given cost and the weight in pounds are within the min and the max value of the nutrients for the selected ingredients.

There are different colors that are displayed based on the output values.

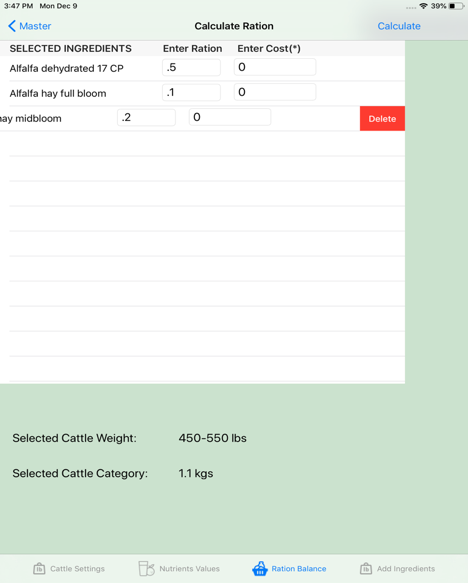
**1. Red: -** The red color indicates that the values are below the range and is danger to the cattle.

**2. Green:** - The green value indicates that the values are in the range and is good for the cattle.

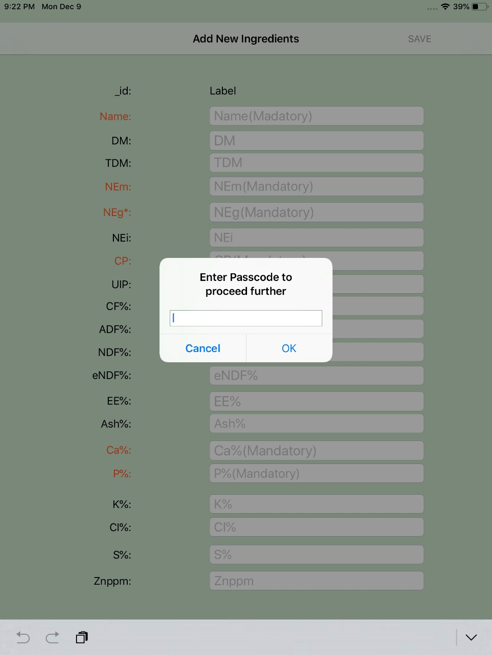
**3. Purple: -** The purple color indicates that the values are out of range and is unto to the user whether to proceed with the given proportions or not.

****

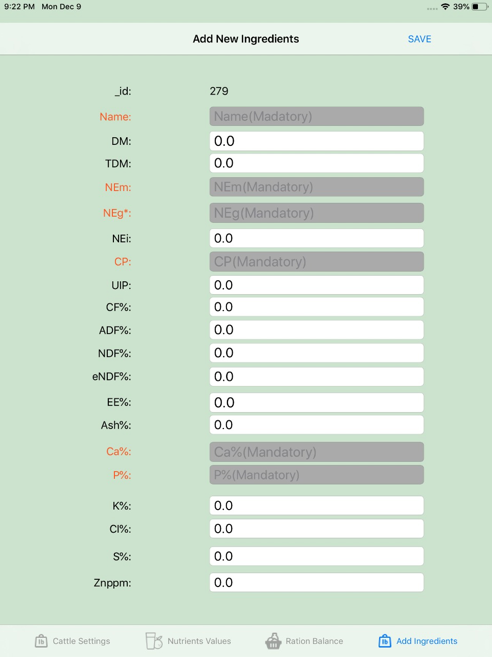
**3j) Delete Selected Ingredient:** The user can delete the selected ingredient by swiping the particular ingredient to left side which shows the delete button. By clicking on the delete it deletes the selected ingredient from the list.

****

**4a) Add New Ingredients**: This is the add ingredients page where only admin can add ingredients to the ingredients data which are stored in the database with the valid log in credentials.

****

**4b) Add New Ingredients:** When the Admin successfully logs into the add new ingredients page, should be able to add the ingredients.

****

1. **Installation guide**

**Pre-requisites**

Setup the deployment environment to have latest version of **Xcode 10.0 or Above**. Should be aware of the database named Backendless. If needed the user need to install pods for the Backendless in the project folder.

**Step 1**

The application is developed in the Xcode. It can be used well in the iPad Devices. The application is connected to the database named “Backendless”. As we have lot of ingredient values the data is stored in the organized manner which is easy to understand about the data in the database.

Please refer the following link to know about the Backendless.

<https://backendless.com/>

**Step 2**

Initially you need to create an account for the Backendless and then create a new user/schema in database for this application. Make sure the new user/schema have enough permissions to perform DML and DDL operations.

You can also use the database we have already created by logging into the “Backendless” with below credentials.

You need to login as with Gmail credentials. Here are the credentials.

Gmail id:- [rationbalancing@gmail.com](mailto:rationbalancing@gmail.com)

Password:- Gdp2019

**Step 3**

Application source code is available in the following Bitbucket repository. Clone or download the repository on to Xcode Platform

<https://github.com/Haritha-Atmakuri/RationBalancing>

Open the cloned repository location in command prompt or terminal and run `git clone https://github.com/Haritha-Atmakuri/RationBalancing.git ` to get the project in your intended location.

1. **Execution guide**

**Step 1**

Open the Database. Check whether the data is present in the database or not.

**Step 2**

Open your source location for the project in terminal or command prompt and run ` git clone https://github.com/Haritha-Atmakuri/RationBalancing.git *`* to clone your project into a folder.

**Step 3**

When running the application please select the device type as iPad 5th Generation and then run the application.

1. **Project URL**

Local application run URL:

<https://drive.google.com/file/d/1PEvgqH1iGYgahVCxZBHEagPlV2Xd5JvS/view?usp=sharing>

Project Repository URL: <https://github.com/Haritha-Atmakuri/RationBalancing.git>

1. **Project Management Artifacts**

**1. Conclusion**

We can conclude that the application is running smoothly. Based on cattle weight and growth requirement provided by the user, it displays the nutrient values accordingly and calculates the appropriate amount of feed required for the cattle. Our application is also capable of increasing the nutrient values and the Admin can add ingredients by using his/her log in credentials. The performance of the application has been tested and appropriate error messages are displayed for incorrect password and other areas in the application as required.

**2. Risk List, Pros and Cons**

**Risk List:**

* If the User without any knowledge about the feed, gives inappropriate nutrient values can be a risk to the cattle.
* Security of the application ensures that unauthorized admins cannot access the system.

**Pros:**

Our application calculates based on the cost and provides the feed in minimum cost.

* Admin is able to login using proper credentials, Only Admin has the capability of adding new ingredients.
* Unauthorized credentials are not permitted access into the application.
* Performance is high because of tabbed application as the data is navigated to every page.
* Application works efficiently according to user’s requirements

**Cons:**

* If the User selects more than ten Ingredients, as the algorithm connects to the server it takes more time to calculate the amount required for cattle.
* User should have the knowledge about the feed and nutrient content of ingredients.
* Our application doesn’t give nutrients in a specified range provided by the user in all cases.

**3. Summary of Requirements Completed**

1. Cattle Settings Page
2. Nutrient Values Page
3. Ration Balance Page
4. Add Ingredients Page
5. Log in for Admin
6. Result Page showing nutrients range
7. Advanced search for Ingredients
8. Backendless for storing data

**4. Summary of Requirements Not Completed**

1. Help page.
2. Filtering based on the cost to calculate the required amount of feed is not possible in cases where some of the ingredient’s weights are predefined.

**5. Conclusion**

We, as a team, have combined our efforts to develop this application. We would like to finally conclude by acknowledging our mentor, Dr Scott Bell for his effort and patience in helping us and guiding us through the project and development process. We would also like to thank our Client, McGee Marcus from the School of Agriculture Sciences, for giving us the opportunity to provide him with an application as such.