Software Requirements Specification

for

Goat Observation and Assessment Technology (G.O.A.T)

Version 1.0 approved.

Prepared by: Mean Green Solutions (Albert Kileo, Jack Follett, Jesus Chavez)

Clean Chickens and Co. LLC

Sponsor info: Rebecca Wierschke

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Jack Follett Albert Kileo Jesus Chavez	10/04/24	Initial creation	1.0

Team name: Mean Green Solutions

Members: Albert Kileo, Jack Follett, Jesus Chavez

Project name: Clean Chickens

Sponsor info: Rebecca Wierschke

1. Introduction

1.1 Purpose

The purpose of this project is to create an automation system that takes in information related to goat meat. The project will consist of goat pictures and goat measurements. This project is to be created By Jack Follett, Albert Kileo, and Jesus Chavez. This project is for Rebecca Wierschke.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority. This project document will be written for the simplest use needed. >

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type. This project is intended for Rebecca Wierschke. >

1.4 Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. The project goal is to create an automated system that will allow our client Rebecca Wierschke to keep track of any goats in terms of size, scope and value. >

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.TBD We have yet to be required to cite any references. >

2. Overall Description

2.1 Product Perspective

This software is being developed to fill the gap in the butchery industry to automatically grade and price live goats before they get slaughtered. There is already a system in place for animals such as chicken, cow, and pig, but nothing for goats. This system will use multiple cameras to take pictures of a goat, use image recognition, and image processing to grade the goat based on certain qualities. Once it is graded, a price is assigned, and stored in a database.

2.2 Product Functions

- Interface with multiple cameras to take pictures of a goat.
- Be able to run effective image analysis to grade and price a goat accurately.
- Take all this information and put it in a database for recordkeeping.

2.3 User Classes and Characteristics

The most important user to build this project around is mainly the only user that will be using it, and that will be the worker that is looking over the process. The user's ability to use a computer is assumed to be adept enough to comprehend a basic and primitive user interface and do minimum interaction with the system. This worker will be interacting with this system for possibly every goat, so they will be using this system a lot.

2.4 Operating Environment

This system will operate in a presumed dirty, industrial environment. It's possible that hundreds of goats every day will passthrough the general area, leaving dirt, saliva, and possible feces behind. It is important that the main computer be setup in a room that is away from the main staging area, to make sure it stays clean. It is possible that the camera lenses might need to be cleaned on a regular basis (at the end of each day) to make sure smudges and other dirt don't distort the image.

2.5 Design and Implementation Constraints

- There aren't many design constraints here. Any language can be used for this process, as long as it is able to perform image recognition, and interface with cameras. The most probable pick right now is Python.
- Cameras that are chosen must be within budget and must be able to be interfaced with automatically using computer code and a wired connection.
- Any additional software or libraries used must be able to be licensed to be used in an enterprise environment.

2.6 User Documentation

Documentation must be provided to the end user for basic setup and usage of the software. The documentation must be able to guide a user that has used technology regularly to become proficient at using the software quickly. A basic troubleshooting guide may be included as well.

3. External Interface Requirements

3.1 User Interfaces

The main user interface will be very simple, and include just the basic controls. The user must manually input the goats breeder and barcode number (unless if this information can be scanned.) After this, the user will be prompted with a TAKE PICTURE button, that will take a picture with all cameras at the same time. The screen will then show all pictures taken, and provide a grade and price. The user then has 3 options. SUBMIT, RE-TAKE, or DELETE. Once submit is clicked, all of the information is sent off to the database. If re-take is selected, pictures are taken again and info is re-calculated. If delete is selected, a “are you sure?” box will pop up and then warn the user of data loss, then start from the beginning. A basic user interface will be created for the database, that will look like a spreadsheet similar to Microsoft Excel. Users will be able to sort by breeder, or find a specific goat, and look at the information attached to it.

3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

3.3 Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.4 Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

4. System Features

4.1 Image Gathering

4.1.1 Description and Priority

Priority: HIGH
Description: This part of the software will interface with cameras and take pictures all at the same time. Once all pictures are taken, they are sent back to the main software that will do image processing on this.

4.1.2 Stimulus/Response Sequences

The user will be prompted with a TAKE PICTURE button, that will take a picture with all cameras at the same time. The screen will then show all pictures taken, and provide a grade and price. The user then has 3 options. SUBMIT, RE-TAKE, or DELETE. Once submit is clicked, all the information is sent off to the database. If re-take is selected, pictures are taken again, and info is re-calculated. If delete is selected, a “are you sure?” box will pop up and then warn the user of data loss, then start from the biggening.

4.1.3 Functional Requirements

REQ-1G: Cameras must be able to take high quality pictures of the goats, high enough quality to run image processing on the resulting image.

REQ-2G: Cameras must be able to be interfaced with by a computer using code. If an error happens with a connection to a camera, and a picture can’t be taken, an error is displayed where the image would be that says “ERORR – Camera was not able to take picture. Please check the connection to the camera, and ensure it’s turned on.”

4.2 Image Recognition

4.2.1 Description and Priority

Priority: HIGH
Description: This is the beef of the software that will possibly use AI or other image recognition technology to recognize certain aspects of the goat and assign a price and grade to a goat based on certain qualities.

4.2.2 Stimulus/Response Sequences

The system will mostly do this in the background with little to no input by the user. This will run right after the software receives images from the cameras.

4.2.3 Functional Requirements

REQ-1R: Must be able to accurately provide a price and grade in at most 30 seconds.

REQ-2R: Must be able to learn and understand the small intricacies between the different grades of goat and assign values properly.

4.3 Database Management

4.3.1 Description and Priority

Priority: HIGH
Description: This part has been requested on behalf of Clean Chickens and Co. All information gathered (pictures taken, the grade and price assigned, the weight of the goat, etc.) will all be assigned to a database. The primary ID being the Goats barcode number. This database will be used for recordkeeping, and to notice trends.

4.3.2 Stimulus/Response Sequences

After the user clicks "SUBMIT" and confirming the results are accurate, the information gets sent off to the database. The user can then go into the database and sort it by any of the variables, and look at any of the data relating to the goat.

4.3.3 Functional Requirements

REQ-1D: All inputs into the database must be valid.
REQ-2D: Sufficient storage must be on-site to store all data for the foreseeable future. If a drive runs out of space, an error message will be displayed. If the drive it is saved to has less than 15% space remaining, a warning message will appear after each goat is sent to the database.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system should be able to process data for at least 100 goats per day, with an expected expansion to 26,000 goats per year within five years. It must be scalable to handle a potential increase in capacity.
- Image processing and grading must be completed within 30 seconds per goat, to ensure operational efficiency during high throughput.

5.2 Safety Requirements

- *Ensure proper safety measures are implemented for data privacy, particularly concerning breeder information and health data.*
- Any physical systems (such as cameras) should be secured to avoid accidents or injuries when operated near livestock.

5.3 Security Requirements

- User authentication must be implemented to restrict access to the grading and database system, ensuring only authorized personnel can view or modify data.
- All data transmitted between hardware components (e.g., cameras) and the software must use encrypted communication to prevent unauthorized access.

5.4 Software Quality Attributes

- **Reliability**: The software should maintain a 99% uptime, with minimal disruptions.
- **Scalability**: It should be designed to handle up to 100 goats per day and scale to accommodate more than 26,000 goats annually.
- **Maintainability**: The codebase should be modular and documented to allow easy updates; especially as grading algorithms improve.
- **Usability**: The user interface must be intuitive for farm workers with limited technological experience.

5.5 Business Rules

- The system should follow a consistent grading system (e.g., sections 1-3) and must be integrated with the existing custom meat system for tracking and origin documentation purposes.
- Pricing of goats must be calculated based on grading and weight, with traceability to the original breeder.

6. Other Requirements

- **Database Requirements**: The system will utilize a SQL-based database for maintaining goat records, including grading, weight, breeder, and any subsequent health or recall-related information. The database should support advanced search and filtering options for efficient information retrieval.
- **Legal Requirements**: Compliance with food safety regulations is mandatory, including traceability standards for meat origin in case of recalls. Data must be securely stored following local data protection laws.
- **Internationalization Requirements**: The system should support units commonly used in the U.S., such as pounds for weight and USD for pricing, but should be adaptable for other measurement systems as required in future markets.
- **Reuse Objectives**: Parts of the system, such as the image processing module, should be designed for reuse in similar projects, potentially with other livestock (e.g., sheep).

Appendix A: Glossary

- **G.O.A.T**: Goat Observation and Assessment Technology.
- **UNT**: University of North Texas.
- **SRS**: Software Requirements Specification.
- **Grading System**: Classification of goats (1 to 3).
- **Breeder**: Original owner of the goat.
- **Living Price**: Price assigned to a goat based on grading.
- **Mean Green Solutions**: Project team.
- **Clean Chickens and Co. LLC**: Client requesting the project.
- **Image Recognition**: Technology for grading goats.
- **Database**: Storage for goat data.

Appendix B: Analysis Models

- **Data Flow Diagram**: Shows data flow from cameras to the database.
- **Class Diagram**: Includes classes like "Goat," "Grader," "Database."
- **Entity-Relationship Diagram**: Displays relationships between "Goat," "Breeder," "Grade."
- **State-Transition Diagram**: Depicts goat states in the system.

Appendix C: To Be Determined List

1. **Budget Allocation**: Final budget breakdown (TBD 10/15/24).
2. **Legal Requirements**: Confirm regulations (TBD 10/31/24).
3. **Stakeholder Availability**: Review meeting dates (TBD 11/05/24).
4. **Data Storage Capacity**: Storage needs (TBD 11/20/24).
5. **Performance Metrics**: Image processing targets (TBD 12/01/24).