

# Requirements Document

*Client: Zeehondencentrum Pieterburen*

ANDREI MIHALACHI *s3491862*  
ANNIKA MOELLER *s3585832*  
BRIAN DE JAGER *s4104579*  
MIKE LUCAS *s3519120*  
ROAN ROSEMA *s4109449*

Lecturer: Dr. Andrea Capiluppi  
Teaching Assistant: C.M. Rosiu  
June 14, 2021

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Actors</b>	<b>2</b>
<b>3</b>	<b>Use Cases</b>	<b>3</b>
3.1	UC1: Upload .wav audio file (Requirement 4.1.1) . . . . .	3
3.2	UC2: Export .csv file (Requirement 4.2.2) . . . . .	5
3.3	UC3: Upload .csv file (Requirement 4.2.4) . . . . .	7
3.4	UC4: Add a column to table/data (Requirement 4.2.5) . . . .	9
3.5	UC5: Delete a column or row from table/data (Requirement 4.2.6 4.3.1) . . . . .	11
3.6	UC6: View statistics/information about data (Requirement 4.2.7 4.3.4) . . . . .	13
<b>4</b>	<b>Requirements</b>	<b>15</b>
4.1	Critical Functional Requirements . . . . .	15
4.2	Important Functional Requirements . . . . .	15
4.3	Useful Functional Requirements . . . . .	16
4.4	Non-Functional Requirements . . . . .	16
4.5	Won't do . . . . .	17
<b>5</b>	<b>Use case diagram</b>	<b>18</b>
<b>6</b>	<b>Requirements Traceability Matrix</b>	<b>19</b>
<b>7</b>	<b>Customer Contact</b>	<b>20</b>
7.1	Customer Meeting Log . . . . .	20
7.2	Customer Correspondence . . . . .	20
<b>8</b>	<b>Change Log</b>	<b>24</b>

# 1 Introduction

The vets who work with the rescued seals at the Zeehondencentrum Pieterburen use several methods to assess the health of the seals once they have been brought to the sanctuary. The goal of this application is to create a tool that will make use of artificial intelligence technology to aid them with this process. Wavealyze is a Windows application that uses two convolutional neural networks (CNN) to analyze the health level of a seal based on the sounds it makes. The user can upload a sound file of a seal making a noise, which is analysed by the CNNs. The first CNN outputs a yes or a no, indicating whether or not there is a whistle in the seal's lungs. The second one indicates the severity level of the rhonchus (a different, lower type of whistle caused by obstruction of the lungs) from 1 to 3.

The main application screen is a table, where the user can upload sound files. The rows indicate the name of each sound file and display the corresponding health scores for the left and right lungs in separate columns. The user also has the option to add columns for further relevant annotations (e.g temperature, bodyweight, general behaviour, etc).

The data generated by the CNN should be portable, so the application will allow the user to download the generated table as a .csv file. Furthermore, the application will also allow users to upload a previously generated .csv file, view it in the application window, and continue editing the table.

# 2 Actors

- **Target user:** This application is being developed for the vets who work at the seal sanctuary of Pieterburen.
- Other seal sanctuaries.

## **3 Use Cases**

In the upcoming sections we elaborate on the most important use cases of our program.

### **3.1 UC1: Upload .wav audio file (Requirement 4.1.1)**

#### **Corresponding user story**

As a user, I want to upload audio files to let the system analyze it.

#### **Rationale/Context**

The veterinarians at Zeehondencentrum Pieterburen want to analyze the survival prospects of their seals. This can get done by results returned by the system. The user being able to upload audio files is therefore one of the main use cases we should deal with.

#### **Frequency of Occurrence**

A few times a month (could vary depending on when analysis is done)

#### **Primary Actor**

The veterinarians at Pieterburen are the primary actors for uploading audio files.

#### **Preconditions**

The user has access to the system (i.e the application is downloaded on the computer they are using).

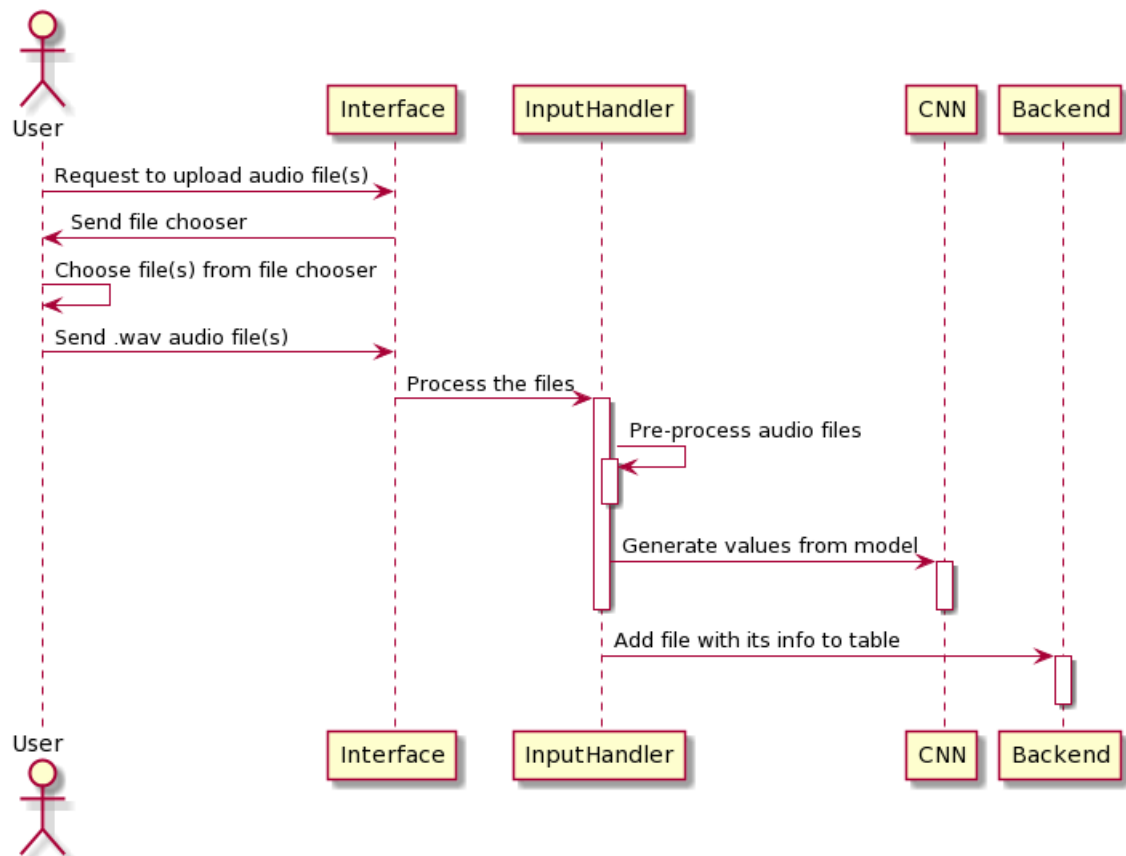
#### **Postconditions**

The uploaded audio file is added to the back end with the result generated from the CNN model.

### Main Success Scenario:

1. The User asks the system to upload an audio file.
2. The System gives the User a file chooser.
3. The User chooses a .wav audio file and sends it to the System.
4. The System pre-processes the audio file.
5. The System uses the CNN model to generate the likelihood of the seal surviving.
6. The System sends the gathered information to the back end of the System.

### Sequence Diagram



## **3.2 UC2: Export .csv file (Requirement 4.2.2)**

### **Corresponding user story**

As a user, I want to export/download a .csv representation of the processed audio files.

### **Rationale/Context**

When audio files have been processed and the data of these files have been stored in the system, the user might want to download this data and work on another set of data or to use the data set on a different computer. The user should therefore be able to export this into a .csv file. The user can continue working on that data set when opening the downloaded .csv file into the system.

### **Frequency of Occurrence**

A few times a month.

### **Primary Actor**

The veterinarians at Pieterburen are the primary actors.

### **Preconditions**

The user has access to the system and at least one audio file has been uploaded to the system.

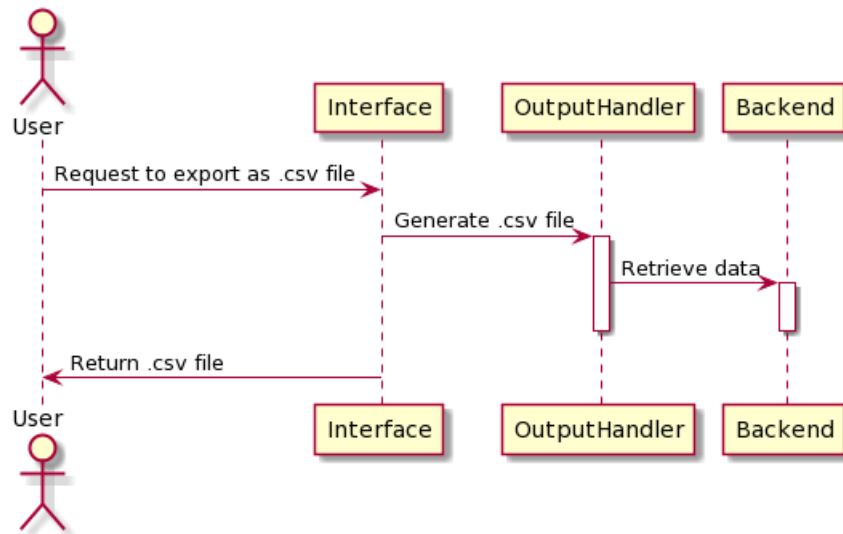
### **Postconditions**

The data of the audio files have been exported to the computer as a .csv file.

### **Main Success Scenario:**

1. The User asks the System to export the data
2. The System generates the .csv file
3. The System returns the .csv file to the User

## Sequence Diagram



### **3.3 UC3: Upload .csv file (Requirement 4.2.4)**

#### **Corresponding user story**

As a user, I want to upload a .csv representation of the processed audio files.

#### **Rationale/Context**

The user might want to upload an already existing .csv file to the program. For example, when a user wants to add information about additional analyzed .wav files to an already existing .csv file, a user might want to upload it. Or if a user has forgotten to add one missing sound file, and finds out after already started working on a new session. Also, a user might want to alter data inside the table of an older session. Therefore, uploading a .csv file is a use case needed to be incorporated.

#### **Frequency of Occurrence**

A few times a month.

#### **Primary Actor**

The veterinarians at Pieterburen are the primary actors.

#### **Preconditions**

The user has access to the system and a .csv file formatted conform the required format of the application.

#### **Postconditions**

The application opens a session with data from a already existing .csv file

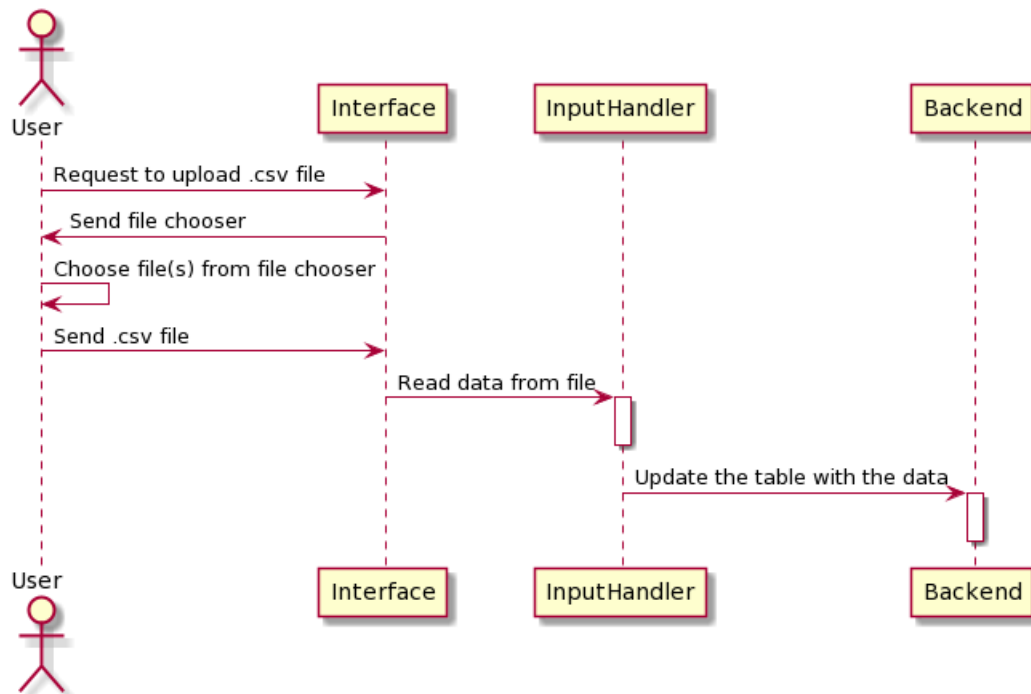
#### **Main Success Scenario:**

1. The User asks the System to upload the csv file
2. The System gives the User a file chooser
3. The User chooses a .csv file and sends it to the System



4. The System checks if data in .csv file is structured conform the structure of the program
5. The System opens a session with data from existing .csv file

### Sequence Diagram



### **3.4 UC4: Add a column to table/data (Requirement 4.2.5)**

#### **Corresponding user story**

As a user, I want to add columns to keep track of more information

#### **Rationale/Context**

The user might want to write down and store additional pieces of information which is not included by the application. For this reason, the user wants to add one or more columns with headers that satisfy their need of reserving specific information in an organized fashion.

#### **Frequency of Occurrence**

A few times a month.

#### **Primary Actor**

The veterinarians at Pieterburen are the primary actors.

#### **Preconditions**

The user has access to the system and a session is currently running.

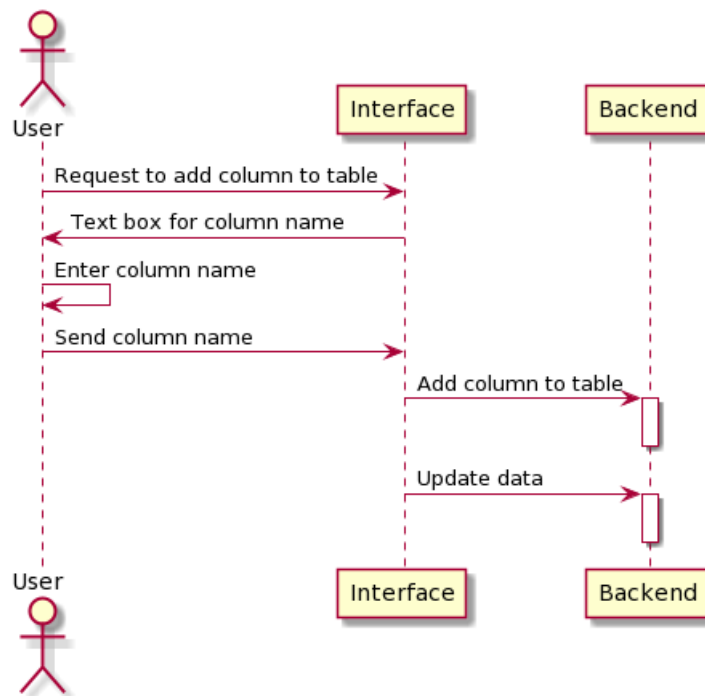
#### **Postconditions**

The application is running a session with one or more columns in the data manually added by the user.

#### **Main Success Scenario:**

1. The User asks the System to add a column to the data
2. The System asks the user for the name of this column
3. The System adds a new column to the table and updates GUI
4. The System updates the data in the back end to keep track of this additional data

## Sequence Diagram



### **3.5 UC5: Delete a column or row from table/data (Requirement 4.2.6 4.3.1)**

#### **Corresponding user story**

As a user, I want to delete columns or rows dispose of unnecessary data

#### **Rationale/Context**

The user might run into the following scenarios. he/she has come to the conclusion that some data is no longer needed. Or, a user has uploaded incorrect files by accident. Or a user wishes to reorganize a file. For these reasons, the user would want to delete rows or columns from the table/data.

#### **Frequency of Occurrence**

A few times a month.

#### **Primary Actor**

The veterinarians at Pieterburen are the primary actors.

#### **Preconditions**

The user has access to the system and a session is currently running, the table/data has a minimum of either one row or one column, and the user has this row/column selected.

#### **Postconditions**

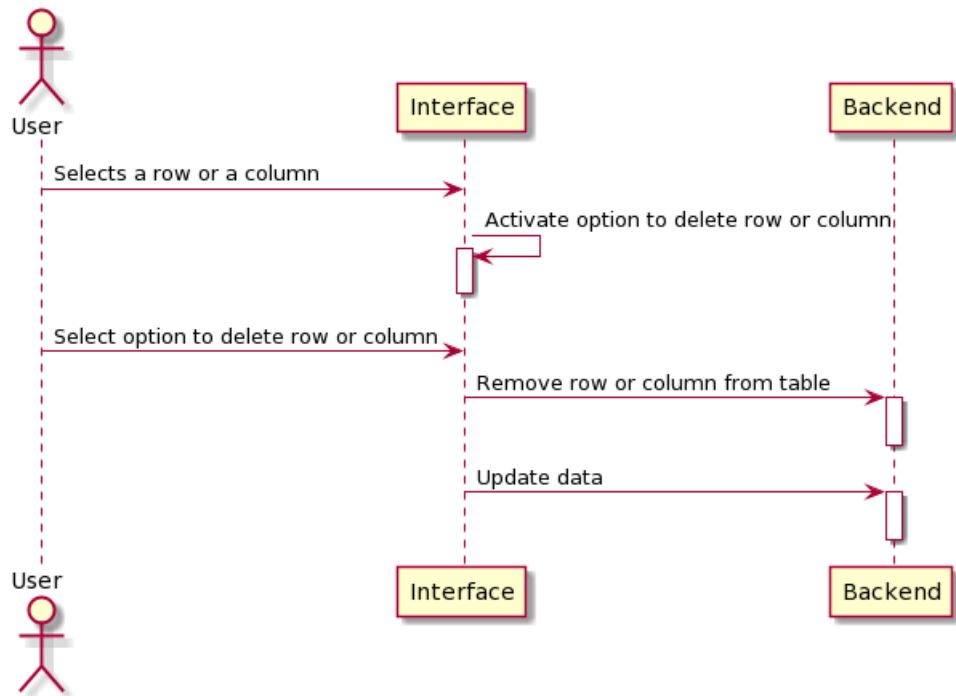
The application is running a session and the table/data present in the post-condition now has the previously selected row or column removed.

#### **Main Success Scenario:**

1. The User selects a row or a column
2. The System makes the option to delete a row or column available
3. The User selects to option to delete a row or column

4. The System removes this row or column from the table and updates GUI
5. The System updates the data in the back end to remove selected row or column

### Sequence Diagram



### **3.6 UC6: View statistics/information about data (Requirement 4.2.7 4.3.4)**

#### **Corresponding user story**

As a user, I want view statistical data and information as spectrogram images.

#### **Rationale/Context**

The user might want to have a clear overview of statistical information about the data. The table/data might be too large to easily acquire this information. Or a user might want to analyse the spectrogram generated by the model per audio file. For these reasons, a statistical view should be present that offers an overview to the user.

#### **Frequency of Occurrence**

A few times a month.

#### **Primary Actor**

The veterinarians at Pieterburen are the primary actors.

#### **Preconditions**

The user has access to the system and a session is currently running, the table/data has a minimum of one valid uploaded sound file and/or values with information generated by the model.

#### **Postconditions**

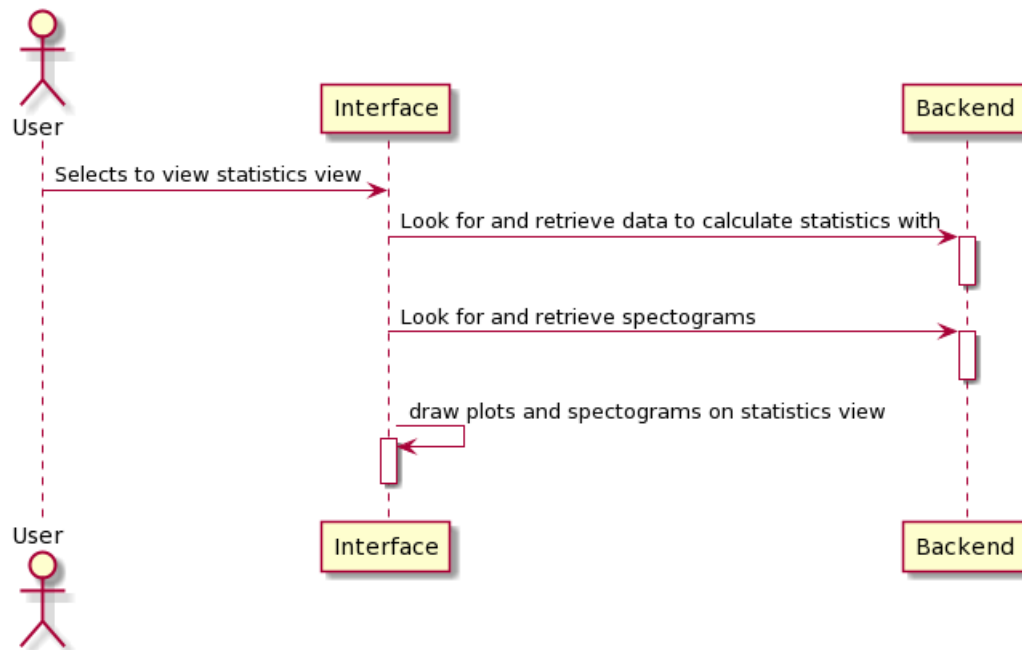
The application is running a session and has updated its statistics view according to the data present in the table/data.

#### **Main Success Scenario:**

1. The User switches to the statistics view
2. The System checks for information in the data to calculate statistics for

3. The System checks for spectrograms generated when sound files are uploaded
4. The System draws plots and spectrograms to the statistics view

### Sequence Diagram



## 4 Requirements

### 4.1 Critical Functional Requirements

4.1.1 Upload one or more .wav audio files at a time (**UC1** at section 3.1)

- The user should be able to upload a sound file (in .wav format) to the application. The 'upload' button should lead to a pop-up menu that allows the user to select only .wav files from their computer.

4.1.2 Pre-process the uploaded audio file(s) (**UC1** at section 3.1)

- The audio files must be pre-processed such that they are in a format that the CNN can read. They should be resampled using a sampling rate of 4000Hz, and use 40 Mel-Frequency-Cepstral-Coefficients (MFCCs) should be retrieved.

4.1.3 Apply the CNN model to the processed audio file(s) (**UC1** at section 3.1)

- The pre-processed sound file must be passed to the neural network so that it can generate the seals' health score.

4.1.4 Display the output of the CNN (**UC1** at section 3.1)

- The application must show the output of the CNN (i.e the health score) in a table format.

### 4.2 Important Functional Requirements

4.2.1 Add information to a cell in the table.

4.2.2 Export the data gathered within the application's table to a .csv file. (**UC2** at section 3.2)

4.2.3 Upload and edit a previously generated .csv file. (**UC3** at section 3.3)

4.2.4 Add columns to the table. (**UC4** at section 3.4)

4.2.5 Delete columns from the table. (**UC5** at section 3.5)



4.2.6 Delete row from the table. (**UC5** at section 3.5)

4.2.7 Locally store the data of the current session.

4.2.8 Create a new empty table

### **4.3 Useful Functional Requirements**

4.3.1 Put left and right lung of a seal in the same row of the table

4.3.2 Generate bar plots based on the result of the CNN model (**UC6** at section 3.6)

4.3.3 Generate additional bar plots by selecting columns added by the user.

4.3.4 View spectrogram(s) of uploaded sound files. (**UC6** at section 3.6)

4.3.5 Export spectrogram of uploaded sound files.

4.3.6 Open a recently saved table.

### **4.4 Non-Functional Requirements**

4.4.1 Run on a Windows computer

- The application must work on Windows.

4.4.2 Intuitive and practical

- The user interface should be clear and easy to use.

4.4.3 Professional aesthetic

- The appearance of the program should be approved by the client.

4.4.4 Make the program modular

- The program should be adaptable to different values gotten from the CNN models

## **4.5 Won't do**

4.5.1 Adapt this application to any other operating system other than Windows

4.5.2 Implement a homescreen for the application

## 5 Use case diagram

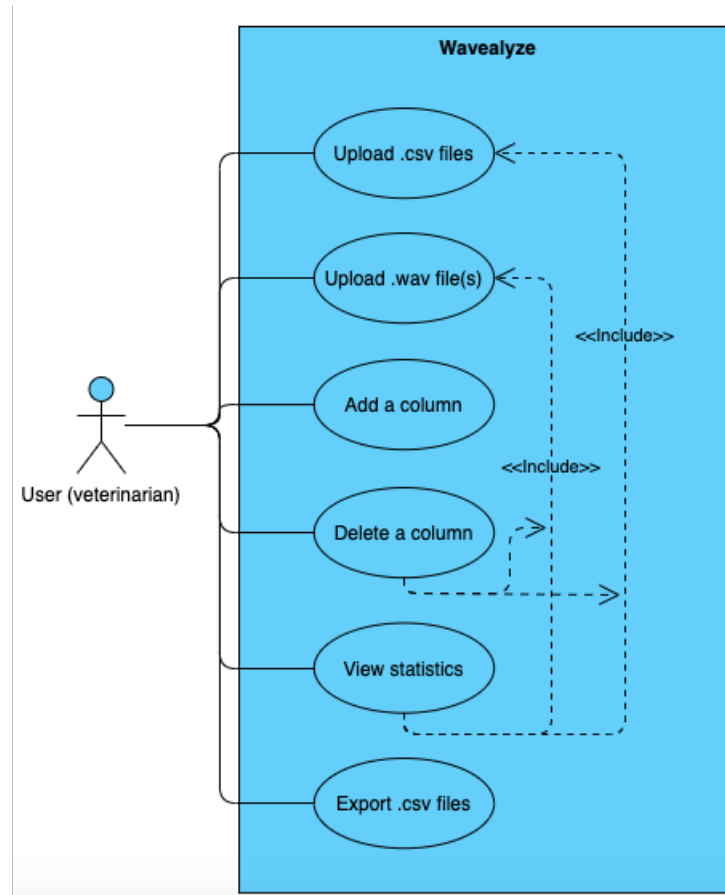


Figure 1: Use case diagram

This use case diagram shows 'include' relationships between use cases. We elected not to use 'extend' relationships, since almost any use case could be performed after another - there is no real flow between the use cases. The only conditional use cases are 'View statistics' and 'Delete a column'. In order to do either of these, the user must already have a table with information on their screen, so they must have completed either 'Upload .wav file(s)' or 'Upload .csv file'.

## 6 Requirements Traceability Matrix

Requirement ID	Module	Test	Passed
4.1.1 Upload one or more .wav audio files at a time	ParseWav.py FileUploadAction.py	test_getFilesWav()	✓
4.1.2 Pre-process the uploaded audio file(s)	feature_extractor2.py feature_extractor3.py	test_feature_extractors()	✓
4.1.3 Apply the CNN model to the processed audio file(s)	feature_extractor2.py feature_extractor3.py	test_feature_extractors()	✓
4.1.4 Display the output of the CNN	BackEnd.py	test_update()	✓
4.2.1 Add information to a cell in the table	BackEnd.py	test_update()	✓
4.2.2 Export the data gathered within the application's table to a .csv file	ExportToCsv.py FileUploadAction.py	test_CSVExport()	✓
4.2.3 Upload and edit a previously generated .csv file	OpenAction.py BackEnd.py TableView.py	test_CSVUpload() test_clear() test_makeEmptyDataFrame() test_openTable()	✓
4.2.4 Add columns to the table	AddColumnAction.py	test_addColumn()	✓
4.2.5 Delete columns from the table	DeleteColumnAction.py	test_deleteColumn()	✓
4.2.6 Delete rows from the table	DeleteRowAction.py	test_deleteRow()	✓
4.2.7 Locally store the data of the current session			
4.2.8 Create a new empty table	BackEnd.py TableView.py	test_clear() test_makeEmptyDataFrame()	✓
4.3.1 Put left and right lung of a seal in the same row of the table	BackEnd.py	test_update()	✓
4.3.2 Generate bar plots based on the result of the CNN model	StatisticsView.py	Testable by running application	
4.3.3 Generate additional bar plots by selecting columns added by user	StatisticsView.py	Testable by running application	
4.3.4 View spectrogram(s) of uploaded sound files	StatisticsView.py	Testable by running application	
4.3.5 Export spectrogram of uploaded sound files	StatisticsView.py	Testable by running application	
4.3.6 Open a recently saved table	BackEnd.py TableView.py	test_clear() test_openTable()	✓

## 7 Customer Contact

### 7.1 Customer Meeting Log

**First meeting (11/02/2021):** The first meeting with the client was held thorough google meet. At this meeting we got the opportunity to introduce ourselves and discuss in more detail what the requirements of the application were. We went into detail through a few questions about said requirements. From these questions we extracted each type of requirement.

**Second meeting (26/03/2021):** A functional application was already in place. We went through some functionality features with the client and we presented a general layout of the application. The client was satisfied with our progress although the color scheme was too bright. The model that classifies the sound of the osculations was not yet connected to the application which remained to be the next update.

**Third meeting (06/04/2021):** This meeting was on the day of the intermediate presentation. We asked whether we can remove the current home screen since it did not add any functionality. Instead we would just have a splash screen that pops up before the table. The client agreed with this decision.

### 7.2 Customer Correspondence

To client, 17-02-2021

Dear Estefania ,

We have started working on a mock-up of the application design , but we still require some clarifications about what the client actually wants. The additional functionalities besides using the CNN to analyse the .wav files are still not fully clear to us.

The project proposal asked us to create an application that creates an overview of the output of the CNN. If we extend the program to include other information about the seals such as temperature, breed etc , it seems like the most appropriate type of application for that would be a sort of animal shelter manager program (example attached below). However , this is quite out of the scope of what the project proposal mentioned. What we understood from what we discussed during the meeting was that the program should first analyse the sound files , then move to a window that has an excel-style table where the output of the CNN is displayed in

one column, and the rest of the columns are available for the vets to enter other information about the seals.

We are not sure if this is the most optimal way for the program to meet the functional requirements of the vets, so we thought that it might be useful to schedule a call with one of them, since they will be the ones actually using the application. It would be very useful for us to be able to talk to a vet and find out which functionalities and what kind of UX would be the most beneficial for them. Would it be possible for you to give us a contact from the sanctuary, or would it be more convenient if we organised this ourselves?

Could you also send us the example .wav files and CNN output when possible?

Kind regards,  
Andrei, Annika, Brian, Mike, Roan

## From client, 17-02-2021

Dear all,

Please, keep Cristian Rosiu in the loop since he is working on the CNN model himself — he will be able to give you more info about the input file, the model, and the output.

What you have sent below is not what I am expecting, I would say that it is too complicated/complex for what it is really needed.

Let me rephrase what we discussed during our meeting:

- Input: Audio files — it would be optimal if the files are read from a folder and process (so the vets just need to include the new files in a specific folder)

The way I imagine the tool is with a table where for each row ( animals/ samples analysed ) we have different annotations (columns)

- Software should process and "keep track" of the info related to the animals:
  - 1) process the input audio file and fill in the correspondent annotation column
  - 2) keep track of the process data — as a table where the info from previously process files is stored
  - 3) give the option to the vets to include some extra annotations ( e.g. annotation1 (output of the CNN model), annotation2 (e.g. temperature), annotation3 (e.g. blood pressure), annotation4, annotation5).
- \* ). users should be able to download the info of table and upload it (in case the tool is moved from one pc to another or if they want to do a back up of the info) — you can store this in .csv, or however you prefer .
- ) Available menu for the users: "Run Analysis", Annotations (include or delete annotation column), "Download Data" and Statistics — if you think of more functionalities please add them
- Statistics will show number of animals / samples per value in the different annotation categories (e.g. after the classification into dead or alive , how many of each) — you can use piecharts or histograms to show statistics.

Is this somehow clearer?

As for meeting the vets, they do not know about UX. These functionalities are the basic ones that I already discussed with them. However, if you find that you will be able to include extra ones, then yes, it would be a good idea to meet them.

Best regards,  
Estefan a

## **To client, 30-04-2021**

Dear Estefania,

While working on the Seals application, we came up with a few ideas and questions which we would like to get your opinion on when you have the time.

1. As mentioned in the presentation, we realised that the home screen was redundant in its functionality, and thought that it would be best to move the functionality for uploading sound files and .csv files to the table view. As such, all functionalities will be handled in the table view, and this will be the main screen.
2. We thought it would be nice to add a short splash screen when the app opens. Currently we have the Pieterburen logo as the splash screen, but is there anything you'd prefer instead?
3. We currently have an auto-save functionality so that in case the user closes the program accidentally, their work will still be saved. There are two options for the design of the app in this case: either opening this auto-saved work in the table view immediately upon reopening the program, or allowing the user to go to an 'Open Recent' button and choose to open the last saved session. Our group was leaning towards the 'Open Recent' button, but we would like to ask what you think would be the most appropriate choice.
4. Cristian mentioned something about putting the data for the left and right lung of the seal in the same row of the table, but we are still unclear on how this would be done. Do you think this is worth doing and if so how would it look?
5. Finally, could you give us some more details about what you expect from the statistics page? Perhaps an example of what you would like it to look like, and how it should appear in the application (i.e to the side of the table in the same window, in a different pop-up window, continuously updated as data is added, a one-time generated graph etc).

Thank you for your time and looking forward to hearing from you, Seals group

## **From client, 03-05-2021**

Dear Seals group,

- > 1. As mentioned in the presentation, we realised that the home screen was redundant in its functionality, and thought that it would be best to move the functionality for uploading sound files and .csv files to the table view. As such, all functionalities will be handled in the table view, and this will be the main screen. Is this okay with you?

Yes !

>

> 2. We thought it would be nice to add a short splash screen when the app opens. Currently we have the Pieterburen logo as the splash screen, but is there anything you'd prefer instead?

I would use an image of a seal (without copyright). So it makes the software broader in case other seal centres use it. Do you find this feasible?

>

> 3. We currently have an auto-save functionality so that in case the user closes the program accidentally, their work will still be saved. There are two options for the design of the app in this case: either opening this auto-saved work in the table view immediately upon reopening the program, or allowing the user to go to an 'Open Recent' button and choose to open the last saved session. Our group was leaning towards the 'Open Recent' button, but we would like to ask what you think would be the most appropriate choice. Another option is to auto-save the file that the user is working on directly to the .csv file, instead of a separate session as we are doing it now. Let us know what you think.

I would not have more than one session saved – just to make it easier for them. That can become messier.

If they want to save/store a version, they can always download it as csv.

What would be good to have is to be able to work with different tables (i.e. csvs) – there they could select with which one to work. This could help in the case they work with data from different centres for instance.

> 4. Cristian mentioned something about putting the data for the left and right lung of the seal in the same row of the table, but we are still unclear on how this would be done. Do you think this is worth doing and if so how would it look?

I do agree with him.

Each row describes the output per id (seal). This would be the same as replicating the excel file they were working with.

Ex:

Row 1 would go for seal 1

Col 1: id seal,

Col 2: output lung L for classification task 1, Col 3: output lung L for classification task 2, Col 4: output lung L for classification task 3

Col 5: output lung R for classification task 1, Col 6: output lung R for classification task 2, Col 7: output lung R for classification task 3

Is this somehow clear?

> 5. Finally, could you give us some more details about what you expect from the statistics page? Perhaps an example of what you would like it to look like, and how it should appear in the application (i.e. to the side of the table in the same window, in a different pop-up window, continuously updated as data is added, a one-time generated graph etc).

Here I give you the freedom to choose because any of the following cases would be good:

– You can either continuously update it or give the user the chance to



- create it.
- It would be good to either have a different pop-up window or just download the graphs / tables with the data statistics.
  - As for the statistics: Table with samples per given label – Here @Cristian could help because he created such classes when describing the dataset in his reports

Please, let me know if you need further information or feedback!

Cheers,  
Estefan a

## 8 Change Log

Who	When	Which section	What
Annika	10-02-21	The document	Created the document
Brian	11-02-21	Requirements	First draft of requirements
Annika & Andrei	12-02-21	Requirements	Updated requirements
Annika	12-02-21	Introduction	First draft
Brian & Roan	16-03-21	Use Cases	Added UC1 and UC2
Brian	29-03-21	Requirements	Added new requirements
Roan	29-03-21	The document	Small restructure + target users updated
Mike	30-03-21	Use Cases	Added UC3 and UC4
Brian	04-05-21	Requirements	Added requirement identification
Roan & Brian	05-05-21	Requirements + UCs	Start of feature tracing/referencing
Roan	06-05-21	Use Cases	Added Sequence Diagrams to first 4 UCs
Andrei	09-05-21	Requirements + CL	Visual update in change log + new UC
Andrei	10-05-21	Customer meeting log	Created and updated the meeting log
Mike	11-05-21	Customer Correspondence	Added email correspondence
Mike	22-05-21	Use Cases	Added UC5 and UC6 and their diagrams
Mike	31-05-21	Requirements	Updated, removed and added requirements
Brian & Mike	08-06-21	Requirements	Rephrasements + added requirements
Roan	11-06-21	Requirements	Added 4.2.8
Roan	11-06-21	RTM	Added some tests to RTM
Annika & Andrei	11-06-21	Requirements Traceability Matrix	Filled in the table
Annika	14-06-21	Use case diagram	Made this section
Mike & Brian	14-06-21	Requirements Traceability Matrix	Entered modules
Brian & Mike	14-06-21	Document	Restructure and proof reading