Graduation Assignment:

Expanding & Improving the Ons Wondzorg Wound Recognition Software

Student (author)	Walter von Gabain (406391)
Academy	ACT, HBO-IT
Course	Graduation/Graduation Project HBO-ICT (2020-2021)
Company	Nedap Healthcare B.V.
Company supervisor	Michiel Klitsie
Graduation Coordinator	Anthony van den Berg
Date	28-6-2021

This page has been left blank intentionally

Table 1: Document History

Version	Modifications	Date
0.1	**Created** the general skeleton with chapters. **Added** All relevant info regarding the assignment from the PoA & Background research document. **Finished** the concept graduation report.	29 – 05 – 2021
1.0	**Expanded & corrected** the content based on feedback. **Added** A warning to this page due to graphic imagery. UML explaining the starting / current situation and more figures. Several figures explaining the old & new situation. The Google Survey in the Appendix. More improvements regarding software design & implementation. **Finished** the final version of the graduation report.	13 – 06 – 2021

Table 2: Distribution History

Version	Distributed to	Comment	Date
0.1	Anthony van den Berg Michiel Klitsie	Review of concept graduation report.	29 – 05 – 2021
1.0	Blackboard hand-in point	Final hand-in according to deadline.	13 – 06 – 2021
1.1	Saxion Post-it Nedap Datascience dep. Nedap Vesalius dep. Nedap Ons Wondzorg dep.	Uploading to the thesis paper database. & Distributing it to the relevant Nedap departments.	28 – 06 – 2021

Warning:
This document contains graphic imagery. There are a number of wounds visible in the chapter 5. The Product – Realization. These images have been documented with the intent of showcasing the main functionality of the mobile application.

Table of Contents

Glossary & Abbreviations

Table 3: Glossary and Abbreviations

Name	Abbreviation	Definition
Machine Learning Algorithm	MLA	A program using math and logic in order to adjust itself as they are exposed to more data. Using this method, it can be taught to create predictions.
Woundcare Consultant Society	WCS	A wound care consultancy agency that created the WCS model. The WCS model is a wound care treatment model that is applied to topological wound care treatments across The Netherlands.
Tensor Flow Lite	TFLite	A machine learning library that is for on-device inference of MLA models such as the Vesalius MLA.
Human-Machine Interfacing	HMI	A methodology that focuses on design & implementation of technical solutions with emphasis on usability and user experience.
Low Fidelity	Lo-Fi	Simple, low-fidelity concepts.
High Fidelity	Hi-Fi	Elaborate, high-fidelity concepts.
Ons (Wondzorg) Platform	-	A Nedap Healthcare SaaS platform & product line including the wound care mobile application.
User Experience	UX	The user experience encompasses the aspects of the end-user's interaction with the company, software and its services.
User Interface	UI	The space in which interactions between humans and machines occur.
Software as a service	SaaS	Software that is licensed and sold on a subscription basis.

Preface

This software engineering graduation assignment was conducted for Saxion Hogeschool and has been issued by the Nedap Healthcare business unit in order to advance the Ons Woundzorg wound recognition software.

I approached Nedap B.V. while searching for graduation opportunities and left my resume via their website whereafter a close & good friend, Mattijs Kuhlmann, recommended me to contact the Nedap Healtcare unit as well. This led to me being chosen to advance and improve the features of the Vesalius mobile machine learning solution that utilizes smart vision to assess wound images.

This report consists of an introduction, project description, methodology and background research concerning the Ons Wondzorg wound recognition software as well as the derived requirements necessary to advance and improve the Vesalius wound recognition software. The report finishes with a conclusion regarding the progression of the project and the implementations as well as future recommendations.

The purpose of this report is to show what approach has been taken in order to advance the Vesalius wound recognition software, which results have been made as well as what could be worked on further. The conclusion shows a quick overview on what has been achieved using a table combined with elaboration.

If this generates more interest regarding the subject of the report, then chapter 5: The Product Realization shows further elaboration regarding the implementation phase. Furthermore, if there is interest in the overall design choices made then chapter 3: Background Research & chapter 4: The Product Design will justify the choices made throughout the process.

I started this graduation assignment with a lot of excitement regarding the application of my specialization, human-machine interfacing where I wanted to find out if this process can be applied to design, test & validate an optimal form of interfacing between a professional healthcare user and a machine learning model. Now that the assignment is concluded, I have a different perspective towards the design, development and implementation of machine learning solutions and the application of my specialization.

I've never looked into deep learning and would have stayed more or less ignorant on the subject was it not for this assignment. While working with the talented data scientist, ms. Nijman, it became clear that deep learning solutions are becoming useful and applicable fast and she inspired me to delve deeper into that branch of computer science.

The fact that the pandemic was fully ongoing, a lockdown was in place and only remote work could be done, had an impact on the realization of this project. Furthermore, since I became a father on the same day that the academic year started, there were days that it was tough to focus solely on the assignment and its realization. For this reason, I am not fully satisfied with the results but have learned tremendously from the process and circumstances

I would like to thank the supervisors of this graduation assignment, mr. Klitsie for giving me the opportunity to work on this graduation assignment and creating a welcoming environment throughout the project. I also would like to thank Anthony van den Berg for the support, motivation and professional guidance given.

Finally, I would like to thank my good friends Mattijs Kuhlmann for recommending me this opportunity and working alongside him for the projects' duration and Mihail Ciobu for the recommendation to use the Pomodoro time management technique together so that work-time was used in a focused and efficient manner. This graduation assignment lasted five months and took place between the 8th of February till the 5th of July.

Abstract

The Nedap Healthcare business unit is developing a mobile wound recognition solution named Vesalius using Kotlin, Python and tools / libraries such as Anaconda and TensorFlow Lite in order to utilize machine learning to determine the boundary of a wound as bitmap based on its picture.

This solution is aimed to become a SaaS product within the Ons Wondzorg mobile platform so that healthcare professionals can benefit from the wound assessments made by the Vesalius machine learning algorithm using the mobile solution.

The Vesalius MLA is not ready for commercial use yet due to the fact that it has to be trained further in order for it to give the most accurate predictions possible. In order to provide datasets for the MLA to learn from, a pipeline was constructed by the previous graduation students that allowed for the assessing of wound images on the mobile Kotlin solution using TensorFlow Lite after which the user is able to correct the predicted wound boundary mask and upload it to the MLA back-end running using Anaconda.

However, to further the advancement of the Vesalius wound recognition software, the thesis student was asked to expand the user interface and improve the user experience so that healthcare professionals could eventually make use of the Vesalius software and gather user-corrected datasets for the MLA to improve on.

Extensive background research has been conducted by asking critical questions to six stakeholders regarding the starting situation and the required improvements. Furthermore, the students' specialization, human-machine interfacing, has been applied in order to research, design, test and validate the optimal form of UX / UI which resulted in a Hi-Fi Figma prototype that got tested by five survey & prototype participants.

The UX / UI of the mobile solution was expanded so that the solution could create additional masks according to the wound care classification methodology which together with the wound image itself would become the dataset that the Vesalius MLA will learn from and increase its assessment accuracy. The Vesalius Python back-end has been adapted to facilitate the sending of the new dataset.

Lastly, changes were made to the UI so that it would conform to the Ons Wondzorg interface theme and iconography.

1. Introduction

This chapter briefly introduces the company Nedap N.V and their business unit Nedap Healthcare which issued this assignment. The introduction continues with the current situation as well as the overall assignment.

1.1 The Company

Nedap Healthcare is a business-unit within Nedap N.V. which specializes in technological development aimed at the healthcare system. The Nedap Healthcare unit was established in 1999 and currently has 200 employees providing medical technical solutions to over a thousand healthcare providers and over three-hundred thousand healthcare professionals.

Nedap Healthcare started when a solution was required regarding the way healthcare was registered in The Netherlands. The main issue was the fact that everything had to be registered and maintained on paper and a digital system would be a more suitable solution for homecare planners.

The Nedap Healthcare unit as well as their solutions have been expanding and specializing, where solutions for healthcare providers, general practitioners as well as medical specialists can utilize Nedap Healthcare's solutions.

The Ons Wondzorg division currently launched their revised wound care module in the last quartile (Q1) of this year (2021). This module helps healthcare professionals with the documentation of the wound care trajectory as well as the ability to input data on the go using the available Wondzorg android application.

1.2 The Starting Situation

Nedap's data science team has developed a MLA named Vesalius. Vesalius makes it possible to create an assessment that distinguishes between wound pixels and non-wound pixels of a wound image and can thereby create a wound mask displaying the boundaries of a wound on top the image. This wound mask is a separate layer that is generated by a TensorFlow Lite library which creates a bitmap that states for each pixel if it is part of a wound or not.

Since the creation of the Vesalius MLA, many students have had the chance to improve on this MLA and the implementation thereof.

The previous Saxion software engineering graduate students have made a front-end mobile implementation for Vesalius in which the MLA can assess pictures made on-the-go using the mobile camera. These pictures can be assessed by the user and the predicted wound area can be adapted by the healthcare professional.

Figure 1: Wound Correction Activity - Starting Situation

Furthermore, the previous students have made a processing pipeline from the MLA codebase through an API into the front-end solution and back. Using this approach, the MLA can process the data sets that have been made on the front-end solution and learn from the user-corrected data. Every so often the MLA will then generate a new Vesalius model that learned from the user-corrected feedback. The front-end solution makes sure that it always has the most recent Vesalius MLA model so that the assessments are always made by the most up-to-date version of Vesalius.

In its current state, the solution can take pictures of the wound and assess them through the Vesalius model stored locally on the device. The user can correct the Vesalius assessment by expanding and reducing the wound matrix (mask) and send it to the Vesalius back-end together with the corrected mask where they will be stored without any form of data management.

1.3 The Assignment

Improving UX / UI of the Vesalius Kotlin Application

The main problem is that the assessed pictures taken using the application have to be re-assessed and possibly edited by medical professionals in order to make sure the artificial neural network can be trained in the medical field to give accurate predictions and therefore be certified as a marketable healthcare product. This user-correction front-end application is not designed with UX / UI in mind since the focus was previously solely on creating the data pipeline.

Expanded User Correction Features

A secondary problem is the re-learnability of the Vesalius MLA due to the fact that at the current moment, Vesalius only assesses the size of wounds and should ideally also take a range of wound colours, primarily: red, yellow, black & blue (WCS colour model plus undefined), into consideration. This means that the wound images have to be distinguished on more than just the wound-size mask, in order to train on wound.

At the current moment there are two students, including the author of this document, working on the improvement of the Vesalius solution. The author is working on the mobile implementation of the solution. Ms. Nijman works on the improvement of the MLA to see if it's possible for the MLA to discern colors of the wound according to the WCS model.

Hashing of User Data for Client Obfuscation

Due to privacy and security concerns, these user-corrected images have to have their client data hashed and stored anonymously and in an orderly fashion so that the Vesalius MLA can still track the images per set, which belong to one person, made sure that said images cannot be retraced to the original client.

Sending & Storage of the User Corrected Data

In the starting situation, the Android application allows the user to correct, and send one dataset consisting of the wound picture plus the wound mask. This needs to be reworked in order to facilitate the changes that are going to be made by the data science intern ms. Nijman as well as the changes made in the mobile application. The new data set consists of the wound image plus four masks.

2. Process Description

This chapter describes the overall process used to complete the assignment. The chapter starts with describing the various methodologies used, it continues with the tools applied and the way code is documented. Lastly, this chapter describes the project phases that it went through.

2.1 Methodologies

Agile Approach & Scrum

The project was conducted according to the Agile methodology, specifically using the Scrum framework. The process and advancement of the project were planned, executed and documented following the Scrum phases and tools.

The Agile approach was chosen as methodology due to the fact that this is the most fitting project management methodology in this context because it allows for a dynamic approach to development where tasks, in the form of user stories, get selected from a product backlog for development sprints. These user stories then go through a cycle of implementation, testing & reviewing and could end up back in development at any time testing failed or the development was insufficient.

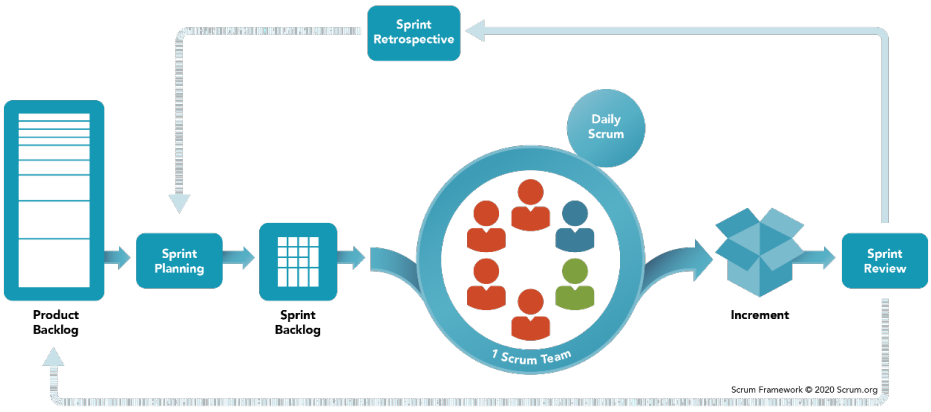


Figure 2: Agile Scrum Process

Alternative methodologies considered were Kanban instead of Scrum but the decision was made to work according to Scrum sprints so that the development process could be tracked and assessed per sprint as seen in appendix H: Example Sprint Log.

The Ons Wondzorg team has a daily Scrum at 11:45 every work day which the graduate student attended and participated. In these stand-ups there is a moment to state what has been done, what is being worked on and what is planned for the rest of the day as well as if there are any blocking issues which the team could help with. These daily stand-ups offered a chance to demo progress to the company supervisor & other Ons Wondzorg colleagues.

Besides the stand-ups there were weekly progression meetings with the company supervisor and product-owner, mr. Klitsie. In these weekly meetings the student informed the company supervisor in a more in-depth manner regarding the advancement of the project. The sprint backlog got reviewed and updated according to the priority of the user stories as stated by the company supervisor.

User Stories

In order for the Agile Scrum method to work, user stories were devised based on the requirements found throughout the background research phase. These user stories were given story points that reflected the amount of effort and uncertainty required for them to be completed. The list of user stories devised are documented in appendix: E: User Stories. At the start of every sprint, the product owner, mr. Klitsie and the student had a sprint review and kick-off where the previous sprint was discussed and the next sprint was planned. When planning the sprint, user stories were selected for the sprint backlog from the overall backlog.

Pomodoro Time-Management

A focused work approach was found using Pomodoro time-slots technique. This technique consists of 25 minutes of focused work with breaks of 5 minutes in between. Every 4 Pomodoro time-slots a break of 20 minutes was allowed.

Some Pomodoro time-slots were continued without breaks, mainly during programming & implementation sessions, but the Pomodoro technique allowed for more precise focus and attention to specific tasks and therefore helped immensely with velocity and motivation. This ability to focus and gain motivation increased with every successful task completed.

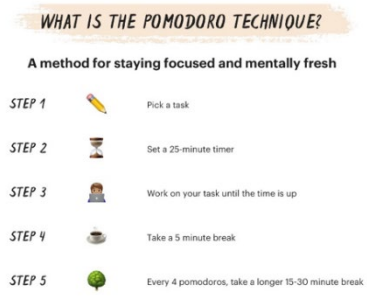


Figure 3: Pomodoro Technique

This methodology was not initially planned to be used, but helped immensely with the lack of progression happening during the early-middle period of the project.

HMI & Design Thinking

Human-Machine Interfacing is the specialization completed by the graduate student before starting the graduate assignment. HMI puts focus on the users' ability to optimally interact with technology as well as what tools and methods optimize user-oriented design. Certain tools and methods used during the specialization that were applied to this assignment are: Figma and Design Thinking.

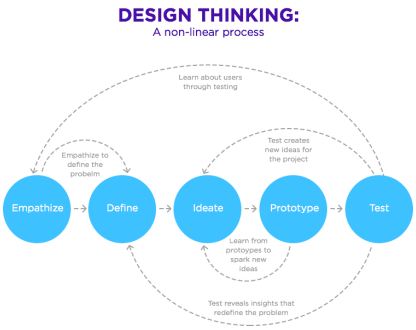


Figure 4: Design Thinking Process

Design Thinking is a methodology that is made for user-oriented design and implementation and therefore fit in the context of this assignment. The Design Thinking methodology was chosen due to its strengths regarding design and implementation in combination with the Agile methodology because both these methods use a structure that can progress dynamically. An alternate option would have been to take a more rudimentary approach to designing and prototyping but one major requirement of this project was to take the users into account. Therefore, the UX / UI got designed and prototyped using Design Thinking with the end-users participating and validating the created prototype.

Design Thinking and Figma were applied to the overall design phase near the end of the background research phase. Empathizing and Defining the solution were not completely relevant to this project due to the fact that the Ons Wondzorg department already had a concise idea regarding the solution and what the users were looking for. For this reason, the Design Thinking methodology started at the Ideation phase with empathizing and defining phase of the solution done as part of the background research and stakeholder interviews conducted.

MoSCoW (Must, Should, Could, Would)

MoSCoW was used to prioritize the requirements in order to gain understanding regarding which requirements where the most important to keep focus on and start implementing. This method is relatively simple and can be of great assistance when many requirements are set. During a sprint kick-off, MoSCoW can help greatly with identifying critical stories.

2.2 Code Documentation

Markdown

Markdown is a lightweight markup language made to create formatted text using a plain-text editor. Markdown documents (.md) exist in the various codebases to elaborate on the codebase, the use of the code as well as the way it should be deployed. During and after the implementation phase, the markdown document will be expanded where ever possible with information regarding the advancement of the project such as necessary additional deployment information or general information necessary to work with the adapted codebase.

Markdown documents were present in the codebases of the Vesalius solution and explained the vital information necessary for further development. For this reason, the documents were expanded upon with the additional information regarding the improvements made by the student.

2.3 Project Phases

Initial Planning & Kick-off

The project starts with the plan of approach as well as initial back-ground research. The plan of approach lays emphasis on the deliverables of the project, the technical & general risk analysis as well as the planning of the project itself while the back-ground research starts off with initial stakeholder meetings and some codebase analysis. The planning that was laid out in the plan of approach consists of a Gantt chart and a general table of phases and their approximate deadlines.

After the plan of approach is approved by the supervisors of the graduation project, a Jira Scrum board is created and the phases laid out in the planning were added as epics.

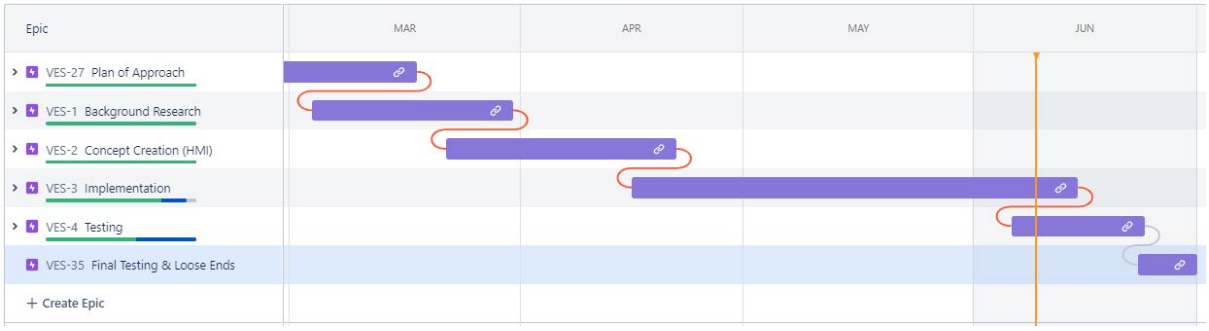


Figure 5: Jira Epics

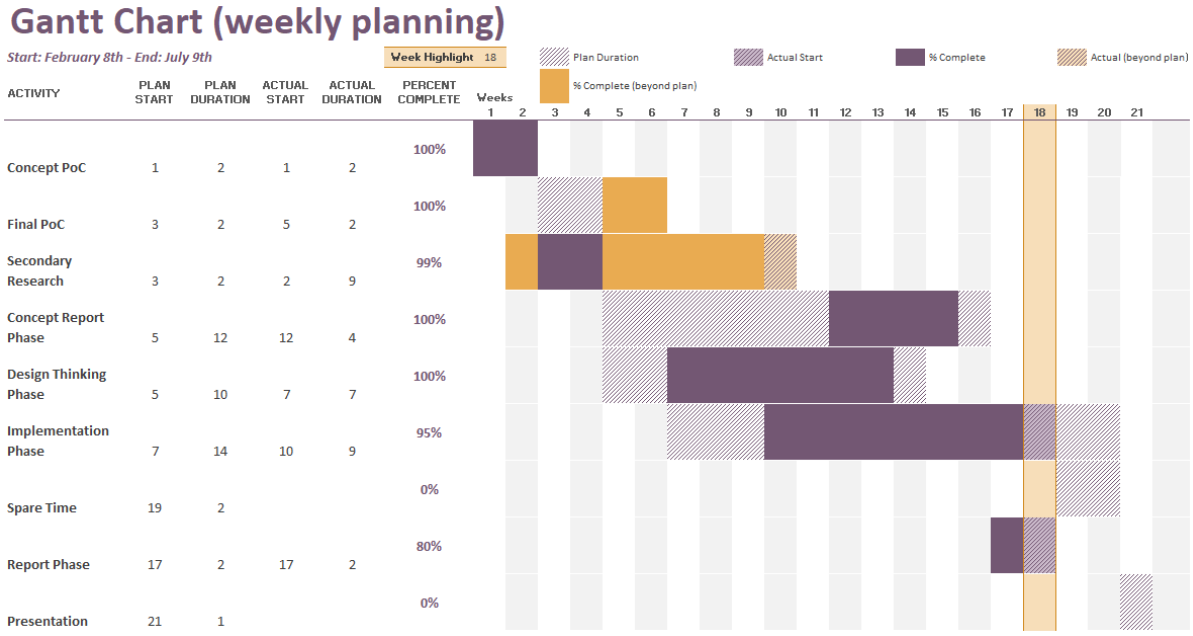


Figure 6: Gantt Chart in Excel

Research Phase

After the plan of approach is finished and accepted by the supervisors, the research phase starts. In this phase emphasis is put on the research questions devised during the plan of approach. Using the various stakeholder interviews with Nedap Healthcare employees as well as their clients the research questions relevant to the stakeholders will be answered.

The research questions that are remaining require further background research. By reviewing the Vesalius solution and its repositories using IDEs and documentation available as well as a competitor analysis these research questions will be answered.

Finally, the HMI – Design Thinking methodology was used in order to gain insight into the optimal form of UX / UI for the front-end solution. This methodology requires a Hi-Fi prototype designed based on the stakeholder interviews and UX requirements. This Hi-Fi prototype will afterwards be tested and validated by stakeholders & clients of the Ons Wondzorg platform.

Implementation Phase

The implementation phase starts after the research phase is mostly completed and the gathered requirements are prioritized according to the MoSCoW methodology so that it is easier to see which requirements have priority. Afterwards these requirements are used to devise user stories which are documented in the background research report and added to the Jira backlog.

The user stories chosen for development are chosen based on their prioritization and assigned story points based on the perceived difficulty and uncertainty according to the Fibonacci sequence. Besides user stories, the sprint board contains tasks that are important to keep track of and complete, such as planning meetings with stakeholders and deadlines for documents.

Testing & Code Review

Testing & code review starts after the implementation phase is completed or the time constraints require the advancement of the project in order to finish. In this phase, the user stories and required features that are implemented will be tested using Java unit tests & test cases whenever applicable.

Kotlin unit tests will help making sure that the implemented functions work as intended and the test cases will make sure that the implemented user-stories are functioning as they are designed. Furthermore, pylinting & pytesting will help make sure that the implemented additions in the back-end solution functions as intended and is implemented using proper syntax.

Lastly, performance testing & benchmarking will take place to verify that the additions to the Vesalius solution do not diminish the performance of the front-end application.

Project Hand-in and Presentation

In order to finalize the project and get graded, the relevant documents, repositories and final graduation report will be handed in. Afterwards, further testing & code review will be done and a presentation will be prepared regarding the project in its totality. The final presentation will take place in the last week of the period.

3. Background Research

This chapter describes the research phase conducted during the assignment. It starts with a problem analysis regarding the current situation and the stakeholders involved. The chapter continues with background research into the research questions devised at the start of the assignment. Using the conducted background research, a number of requirements are formulated and documented in chapter 4. The last subchapter describes the start of the HMI: Design Thinking methodology.

3.1 Problem Analysis

A research phase is necessary in order to get to the root of the problem and define the scope of the solution. In this phase the scope of the assignment will be defined as precisely as possible using stakeholder interviews, research questions and the requirements gathered using said stakeholder interviews.

Stakeholder interviews have been conducted with many specialists both from Nedap B.V. and with organizational users that are clients of Nedap Healthcare. These stakeholders have answered questions that gave insight into the starting situation, the desired situation as well as their views on what was important to keep in mind or add to the project in terms of scope or requirement.

Some stakeholders were self-evident, such as the product owner mr. Klitsie and the previous graduate students who worked on the Vesalius solution and are now employed at Nedap Healthcare. Other stakeholders were introduced during the assignment, such as the organizational users, the data-scientist intern mr. Nijman and the data privacy officer mr. Glintmeijer.

Some stakeholders were contacted once, such as the privacy officer, and others were contacted weekly or even more often than that. Meetings with ms. Nijman, the machine learning intern, happened multiple times a week since our projects aligned and progressed somewhat interdependently. Ms. Nijman was asked to expand and improve the Vesalius MLA so that it could distinguish more than just the wound boundaries. The improved MLA should be able to distinguish wound colors besides the original assessments so that the new dataset includes four additional wound masks for each WCS wound color plus an undefined color.

For this reason, the starting situation regarding the front-end solution had to be expanded by creating multiple masks identically to what the new Vesalius MLA expects as results so that the user can assess and validate more than only the wound boundary mask. In order to allow the user to go through this process optimally, a front-end UX & UI had to be designed and validated. Lastly, the datasets that are send from the mobile application through the back-end needed to be adapted in order to accept multiple masks. This meant that the previous graduate student, Mathijs Hudepohl, who worked on the data pipeline between the mobile solution and the Vesalius MLA could offer a lot more insight into this problem. The privacy offer, Michel Glintmeijer, also required the solution to obfuscate the client information that came along with the data sets since this information is not allowed to be stored in this manner with connotations to the owners of the wounds.

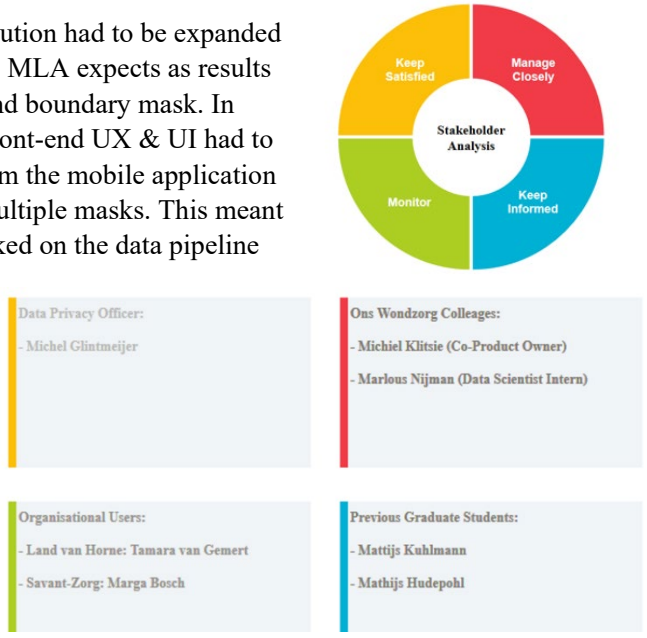


Figure 7: Stakeholder Analysis

3.2 Research Questions

The main research question was: **"What is the optimal way to display and edit the affected areas of a wound using masks in inside a wound boundary as well as the colors red, yellow and black according to the WCS model."**

3.2.1 What is the current situation?

The starting situation consisted of a number of repositories developed in Kotlin and Python. The Android front-end application was developed in Kotlin while the Android back-end & Vesalius back-end are both Python repositories.

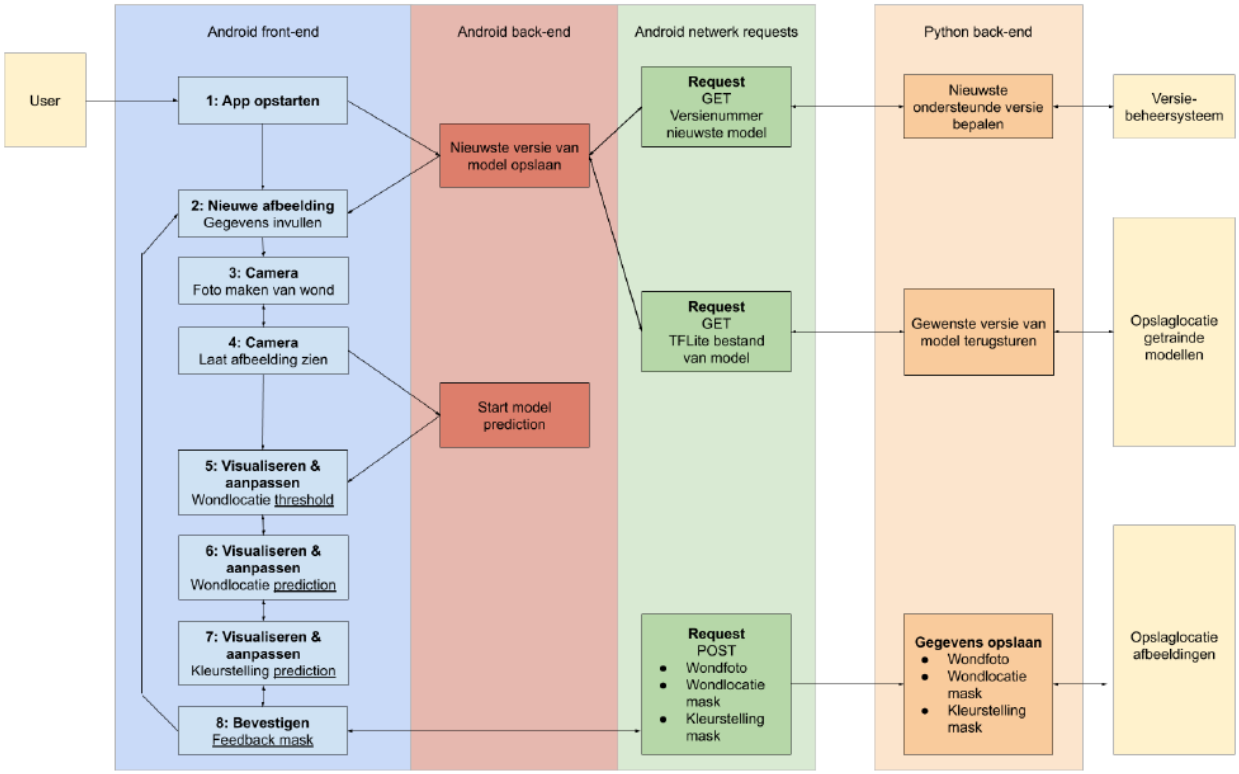


Figure 8: The Vesalius Solution Overview - Starting Situation

3.2.1.1 How does the current UX function?

The user starts by opening the application, where the application will determine if the user has:

1) The most recent version of the Vesalius model.
2) Internet so that the back-end can be asked to retrieve the latest version.

If there is no internet available, the application still functions with the latest model available. If there is no model at all then it will ask to connect to the internet.

The user continues by filling in basic details and makes a picture of the wound to be assessed by the model. Afterwards the user can adapt the wound location threshold in order to get the best result possible and lastly, the user can confirm and send the mask containing the assessed image as well as user feedback through the back-end into the image storage.

If there is no internet available while the user attempts to send assessments, the assessments will be saved nonetheless and can be send at a later time.

3.2.1.2 What functionalities are in the Vesalius Android Front-end?

In this layer of the solution, the wound image gets taken and a wound boundary mask gets determined by the local Vesalius TFLite model using the TFLite library implemented in the *TFLiteClassifier* – utility class.

There are multiple activities present in the mobile solution, such as the *takeWoundPhoto* activity and the *enterInfo* activity. These activities are necessary for the creation of a correctly annotated dataset which is done by entering a client and organization ID as well as the ability to utilize the android camera to provide an image for the TFLite model to determine a wound boundary mask.

The primary activity of the front-end solution contains the logic necessary to adapt the bitmap of the wound area determined by the local Vesalius model. This activity is the *adjustWoundArea* activity. In order for this model to predict more accurately, many user-corrected assessments will have to be stored as an annotated dataset so that they can be re-processed by the Vesalius MLA running on the Nedap datascience server.

When the user takes an image and verifies that it is a wound image that has to be determined, the *STARTING_PREDICTION* state is initiated as depicted below.

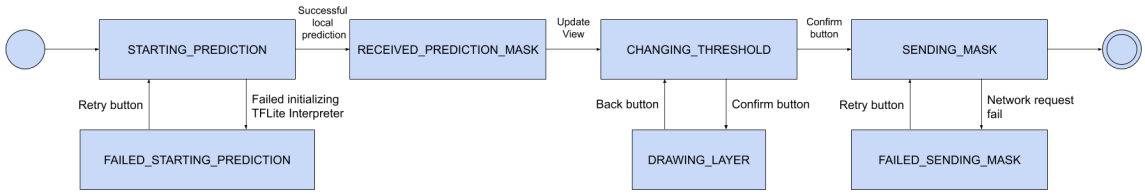


Figure 9: Android State Diagram - Starting Situation

After the prediction is successful, the next step is for the user to assess the correct threshold of the wound image, the default threshold is 0.5 but the accurate threshold has to be determined using user assessments. This threshold determines the probability of pixels belonging to the wound boundary mask. Afterwards the user can proceed to the next phase of the wound boundary mask assessment where the user will use the android *onTouch()* event to edit the wound mask boundary itself by drawing and / or deleting parts of the mask.

The resulting corrected wound boundary mask, alongside the original image get send and stored on the Python back-end as new dataset for re-trainability purposes of the Vesalius MLA.

3.2.1.3 What functionalities are in the Vesalius: Android back-end?

The Android back-end consists of an API-interface implemented in the Kotlin codebase as well as an API written in Python that handles the HTTP calls from the mobile application.

The Android back-end received the wound image plus wound boundary mask as bit stream after which it converts this data into a wound JPEG and mask JPEG that are then annotated using the ClientID / OrganisationID given in the front-end solution and stored as part of a new dataset for the Vesalius MLA to process and learn from.

3.2.1.4 What is a wound image mask that the MLA model generates and what is it used for?

The wound image and wound image mask are two values within the dataset that is used for a Machine Learning Algorithm such as Vesalius to learn from in order to predict more accurately. A MLA like Vesalius uses thousands of accurate values in order to learn how to predict accurately.

The wound boundary mask allows for distinction between wound and non-wound in an image. It is important to track the development of a wound in regards of size as well as wound color because this indicates the wound trajectory according to the stakeholder meeting with the wound care professionals.

The wound trajectory determines what type of professional care and procedure is required to help the wound heal or if it is healing at all. According to the stakeholder interview with the wound care specialists, a MLA will potentially be able to determine the wound size more effectively than actual nurses due to the fact that X amount of nurses will likely have different determinations regarding the size and coloration of the wound.

The old Vesalius MLA that was utilized in the Vesalius solution used a binary prediction algorithm that determined the chance of which a pixel was part of the wound or not. The user therefore had to adjust a threshold value that is between 0 and 1 in order to indicate the chance of which a pixel is part of the wound. As shown in figure 10, this threshold value is depicted as a slider in the UI and can be adjusted by the user.

The new situation does not use a threshold value however, because it uses a different prediction model. This is explained more elaborately in the chapter 4.3: Front-end.

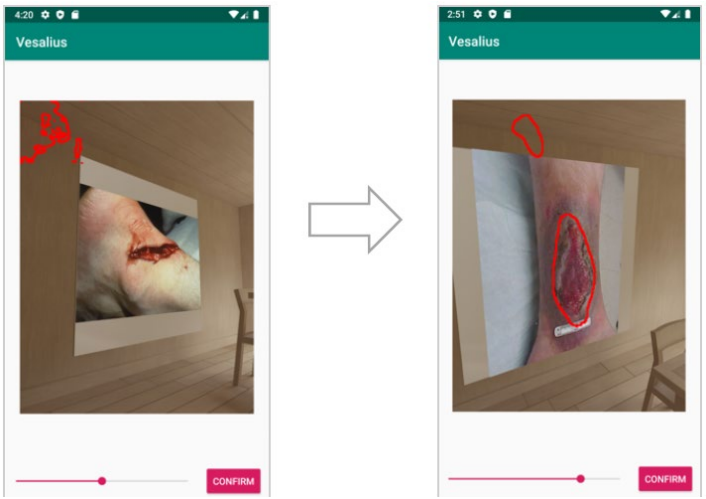


Figure 10: Wound Boundary Mask Threshold Adjustment - Starting Situation

3.2.1.5 Why does the MLA require user-corrected wound masks from the front-end application?

The Vesalius MLA requires values that are generated the TFLite Vesalius model and are afterwards validated and if necessary, corrected by users so that the MLA can learn from these user-corrected values and learn how to predict more accurately until it works as intended and can be released for commercial use.

Using a MLA to determine the wound boundary and wound colors according to the WCS model as labels as well as the percentage of wound color in relationship to the wound boundary has a lot of potential for the medical field because wound-care trajectories can be determined and documented on a level of pixel-perfect accuracy. This is important because having a wound image together with these labels that fit together pixel-perfectly will sum together into a level of accuracy a human assessment is unlikely to accomplish.

This level of wound assessment and documentation can potentially safe a lot of time from nurses by simply making a picture of a wound instead of measuring it as well as reduce the error-margin of human assessment due to the fact that a MLA can be trained to become more accurate by re-training it on datasets.

3.2.1.6 How does the Vesalius: MLA function in relationship to the mobile solution?

The Vesalius MLA is trained and deployed on the Vesalius python back-end. It is continuously re-trained on user-assessed / user-corrected data that is send from the mobile solution and stored on the Vesalius python repository.

Using the TFLite interpreter library and a version of the TFLite Vesalius MLA model, which is generated using the TFLite API on the Vesalius back-end, the Android solution is able to assess wound images locally and create a bitmap of wound pixel values that encompass the wound.

In figure 11 the functionality of the TFLite library is shown. TFLite is utilized in a server – client structure within the Vesalius Solution. The first layer is the Vesalius Python back-end where new models are trained and published. These models are saved and converted to a TFLite Flatbuffer for client-side use. When the mobile solution is started, the latest version of the TFLite flatbuffer is fetched and stored locally so that it can always be used whether the user has internet or not.

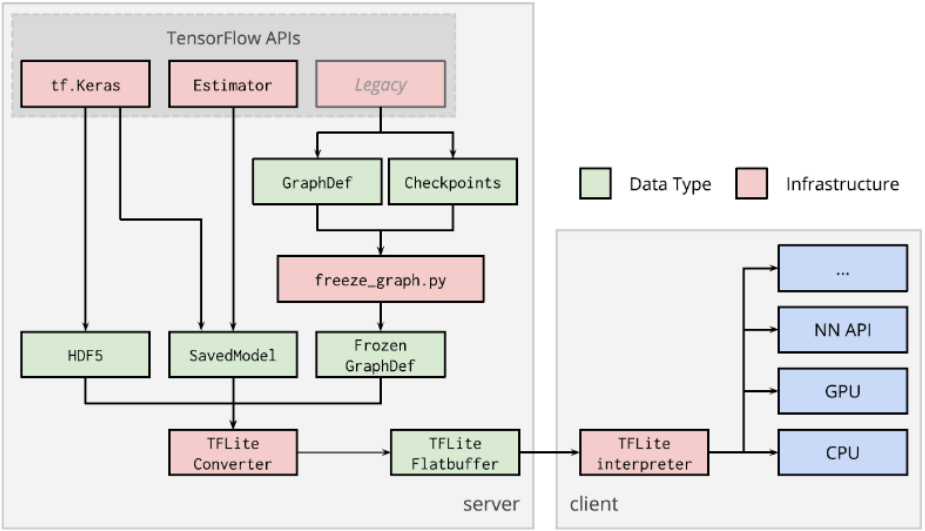


Figure 11: TensorFlow Client Server Architecture

3.2.2 Achieving the Desired Situation

3.2.2.1 Which camera & image editing strategies can be used to allow the user to correct and validate the MLA's wound assessments?

The previously implemented solution worked by adapting the wound boundary bitmap using the Android *onclick()* function which calls the *matrixutility* class so that the neighboring pixels of the matrix are expanded or retracted. This strategy was reproduced and expanded on multiple extra labels according to the WCS model. As described by wound care specialists who are future users and stakeholders of this assignment, there will have to be extra wound layers based on the WCS colors plus the color for an undefined wound. These extra layers should therefore be implemented as separate bitmaps that can be adapted similarly to the original wound boundary mask.

It is important to mention that the improved strategy for user correction and validation requires the ability to distinguish different bitmaps so that they don't overlay and interfere with each other so that every assessment is pixel perfect and can be assessed by the MLA for improved accuracy. The assessments are pixel perfect in the sense that every picture value is claimed by one label only and therefore wound colors will never be able to overlap or interfere with each other so that the Vesalius MLA does not have to re-train on pictures with overlapping labels.

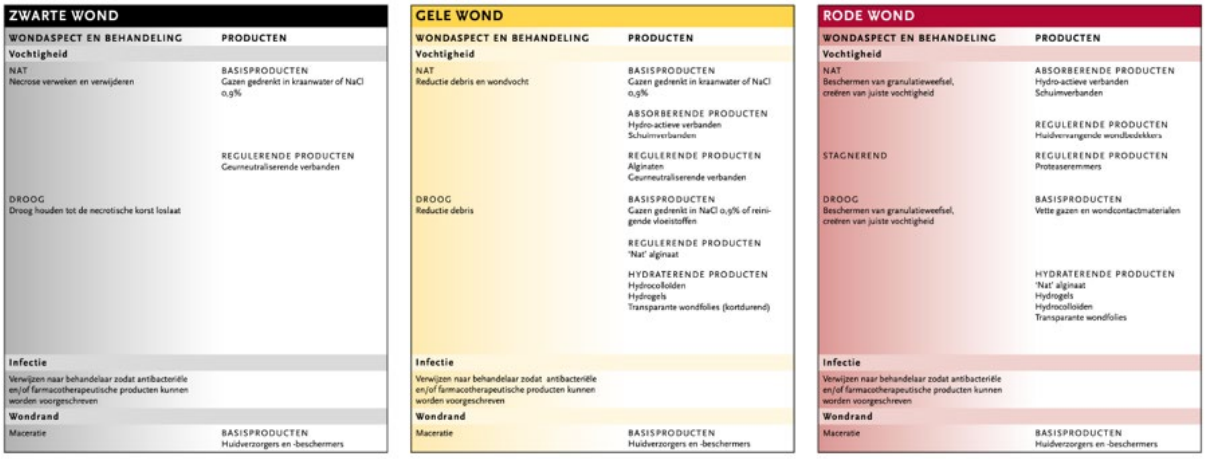
These additional bitmaps are also important for the data scientist intern and stakeholder; ms. Nijman. Ms. Nijman stated that her improved Vesalius MLA will be able to determine wound images for more than the wound boundary only but the wound colors as well. This means that in order to use this new MLA as model, the wound colors will have to be loaded as bitmaps alongside the original bitmap. For this reason, the solution benefits from having one wound image plus five wound masks as one data set.

3.2.2.2 What is the WCS model how should it be applied to the solution?

The WCS model is a wound treatment model made to measure the wound care trajectory of wounds according to the size of the wound and three colors. These three colors are black, yellow and red. Black is classified as necrotic (dead) tissue, yellow is classified as exudated (infected) tissue and red means granulated (healing) tissue. This model is used by nurses and wound care specialists in order to assess the healing trajectory of wounds so that treatments can be assigned based on the state and trajectory.

Stakeholder interviews with professional wound care specialists who are also users of the Ons Wondzorg platform have stated that not all wound tissue types can be assigned within the WCS classification guidelines and requested that a fourth color label should be introduced to the Vesalius solution in order to assign an undefined wound tissue type that can occur due to e.g., burn wounds.

The stakeholders have mentioned that due to the fact that nurses documenting these wound patterns according to the WCS methodology are usually not consistent in their approach and document various results concluding in a tough to assess trajectory. If AI can successfully and accurately predict these wound patterns so that they can be logged onto a follow-up over time where it will result in a more consistent and accurate wound healing process.



ZWARTE WOND	
WONDASPECT EN BEHANDELING	PRODUCTEN
Vochtigheid	
NAT Necrose verweken en verwijderen	BASISPRODUCTEN Gazen gedrenkt in kraanwater of NaCl 0,9%
	REGULERENDE PRODUCTEN Geurneutraliserende verbanden
DROOG Droog houden tot de necrotische korst loslaat	
Infectie	
Verwijzen naar behandelaar zodat antibacteriële en/of farmacotherapeutische producten kunnen worden voorgeschreven	
Wondrand	
Maceratie	BASISPRODUCTEN Huidverzorgers en -beschermers

GELE WOND	
WONDASPECT EN BEHANDELING	PRODUCTEN
Vochtigheid	
NAT Reductie debris en wondvocht	BASISPRODUCTEN Gazen gedrenkt in kraanwater of NaCl 0,9%
	ABSORBERENDE PRODUCTEN Hydro-actieve verbanden Schuimverbanden
	REGULERENDE PRODUCTEN Alginaten Geurneutraliserende verbanden
DROOG Reductie debris	BASISPRODUCTEN Gazen gedrenkt in NaCl 0,9% of reinigende vloeistoffen
	REGULERENDE PRODUCTEN 'Nat' alginaat
	HYDRATERENDE PRODUCTEN Hydrocolloïden Hydrogels Transparante wondfolies (kortdurend)
Infectie	
Verwijzen naar behandelaar zodat antibacteriële en/of farmacotherapeutische producten kunnen worden voorgeschreven	
Wondrand	
Maceratie	BASISPRODUCTEN Huidverzorgers en -beschermers

RODE WOND	
WONDASPECT EN BEHANDELING	PRODUCTEN
Vochtigheid	
NAT Beschermen van granulatieweefsel, creëren van juiste vochtigheid	ABSORBERENDE PRODUCTEN Hydro-actieve verbanden Schuimverbanden
	REGULERENDE PRODUCTEN Huidvervangende wondbedekkers
STAGNEREND	REGULERENDE PRODUCTEN Proteaseremmers
DROOG Beschermen van granulatieweefsel, creëren van juiste vochtigheid	BASISPRODUCTEN Vette gazen en wondcontactmaterialen
	HYDRATERENDE PRODUCTEN 'Nat' alginaat Hydrogels Hydrocolloïden Transparante wondfolies
Infectie	
Verwijzen naar behandelaar zodat antibacteriële en/of farmacotherapeutische producten kunnen worden voorgeschreven	
Wondrand	
Maceratie	BASISPRODUCTEN Huidverzorgers en -beschermers

Figure 12: WCS Wound Classification Method

3.2.2.3 What data regarding the wound image and its masks does the Vesalius MLA require for the purpose of re-trainability?

According to the stakeholder interviews held with ms. Nijman, the old MLA required a dataset consisting of a wound image and a wound mask indicating the boundary of said wound alone. Ms. Nijman trained a new MLA that takes images in a different dimension 224 x 224 instead of the original 256 x 256. Besides the difference in resolution, the new MLA requires 4 additional masks for the colors of the wound. These 4 new masks are required in order for the MLA to distinguish wound coloration according to the WCS model. These wound colors are; black, yellow, red and blue. Blue is not a wound color however; this color is meant to define an unclarified wound color outside of the WCS model. This extra unclarified color was added because of the stakeholder interview held with the wound care specialists and future users of the Vesalius solution, who stated that not all wounds fall into the WCS category such as burn wounds which often have different coloration due to heat damage.

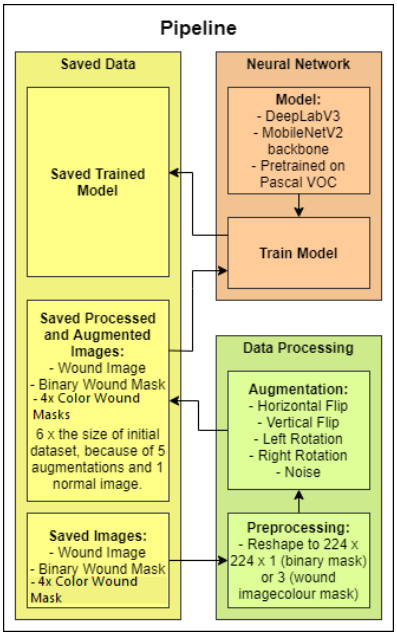


Figure 13: Vesalius MLA Re-trainability Pipeline

3.2.2.4 Where does the user-corrected Vesalius wound-image assessments have to be stored and in what format in order for Vesalius to learn from them?

The Vesalius wound label assessments are determined on the first layer of the solution using the TFLite library and the local TFLite model. The bitmap that is generated by the model is validated and / or corrected by the user in the front-end solution whereafter it gets send to the mobile back-end where it gets transformed into JPEGs. These JPEGs are then uploaded and stored on the Vesalius back-end solution in separate folders based on the client and company ID entered by the user in the mobile application.

3.2.2.5 What security & user privacy measurements have to be taken in order for the user-corrected wound images to be stored securely?

In the starting scenario as stated by the stakeholder Mathijs Hudepohl, the data send between layers of the solution are encrypted using Nedap environment keys therefore the data is transported throughout the pipeline securely. This can be maintained throughout the improvements made on the Vesalius solution.

However, it is important to note that the data which is send securely is not private or obscured in any way. For this reason, a stakeholder interview was held with the privacy officer of Nedap, Michel Glintmeijer. Mr. Glintmeijer filled out a privacy assessment form based on the Vesalius solution and the way it handles information which concluded that the solution needs to obfuscate the datasets containing the ClientID because these datasets are not allowed to be stored in a way that they can be derived back to the owner of the wound image.

For this reason, the solution needs to hash the ClientID of the datasets before getting stored on the Vesalius server so that every dataset can be congregated into separate folders per Client per Organization so that the images can be derived back to the owner of the wound image without revealing the owners' identity.

3.2.2.6 What does the user require regarding the UX / UI design in order to achieve the most optimal usability of the front-end application?

After multiple stakeholder interviews with mr. Klitsie the product owner, wound-care experts & developers such as the previous graduates working on the Vesalius project, a UX scenario can be designed to get a sense of the possible implementable solution. Using the competitor analysis, stakeholder interviews and background research into the state of the current solution, the following scenario got written and verified by the relevant stakeholders.

1) Login through the Ons platform and continue to add or correct an image to the Ons Wondzorg Rapportage or upload a new image to the wound rapport.

2) On the Ons Wondzorg add-wound camera activity, switch to the smart-camera functionality.
   OR click on wound-detail view in the timeline activity.

3) Take a picture of a wound so that Vesalius can assess it based on its size and WCS colors.
   OR assess a picture from the wound-detail view timeline.

4) Give the user image-based feedback laid-over the camera screen so that the user makes the best possible image of the wound.
AND A colored crosshair to help center the wound and provide feedback for the user regarding the optimal positioning of the camera for a wound image.

5) Display the assessment incl. most important details (tissue color %) and prompt if it is correct *No?*
Start correcting Vesalius assessment using drawing tools for the boundary label and WCS model labels (plus other for undetermined tissue).

6) Finalizing the assessment and user feedback.
Finish and "upload" to Ons Wondzorg Rapportage.
Add the image and relevant wound data to the wond rapport.

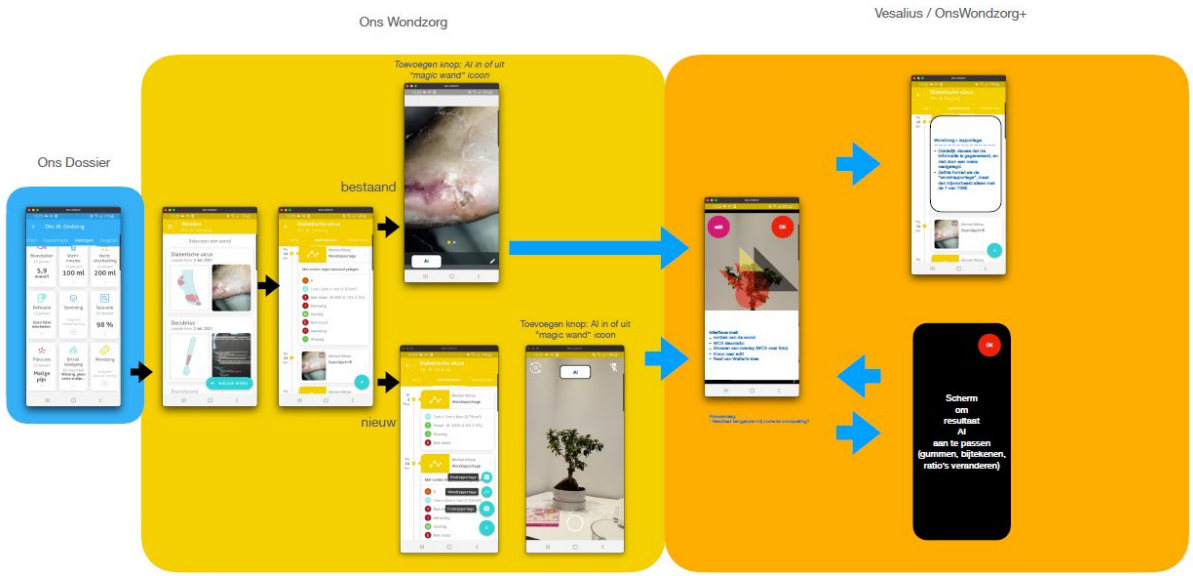


Figure 14: Ons Wondzorg Vesalius Solution Integration UX

This image shows the flow from the Ons Wondzorg platform (yellow) into the Vesalius component (orange) and how it is supposed to function seamlessly. There will have to be two routes into the component. The first route is through the existing wound rapport where you can prompt the Vesalius AI to assess existing pictures and the second route is through the Camera button in which you can switch to "Ons Wondzorg +" module and it will assess a picture that is made on the spot.

3.3 HMI - Design Thinking

The mobile solution required UI / UX design which means that an interface should be developed with the users' intents in mind. This was achieved using the Design Thinking methodology and its phases starting at the empathizing phase. Any phases before the empathizing phase are not relevant for the current situation due to the fact that the idea and technicalities regarding the Vesalius solution where already thought out thoroughly by the product owner, mr. Klitsie.

In the **Empathizing & Defining** phase, background research was conducted regarding the optimal usability of the mobile solution. This form of research was done using code review into the front-end solution to see the current UX / UI and the features implemented as well as stakeholder interviews with potential users and the product owners.

Afterwards comes the **Ideation** phase where the users' requirements for the interface and design were taken into account in order to come up with possible solutions. Inspiration can be taken from similar projects and situations when a solution is straight forward, but in the case of a smart wound vision assessment application there are not many examples to look into. Fortunately, access was given to the Ekare wound assessment application by the product owner, mr. Klitsie. Therefore, during the ideation phase a competitor analysis was conducted in which the existing wound assessment application was analyzed for its good, bad and ugly aspects.

After ideation comes the **Prototyping** phase in which designs get drawn up according to HMI design principles such as the Gestalt principles. The Gestalt principles help with the creation of the most useful and easiest to interpret user interfaces by considering the way information is most naturally presented. Prototyping can be done using Lo-Fi designs of general concepts and A/B tests as well as in-depth Hi-Fi prototypes that can be interacted with somewhat. The decision was made to focus on a Hi-Fi prototype due to time constraints. This prototype is presented and described further in chapter 4.2: HMI Prototyping.

Finally, there is the **testing** phase. This phase is meant to validate the design decisions and is also done in chapter 4 using an interactable Figma prototype, discourse and a Google survey.

Competitor Analysis: Ekare Insight Europe

The competitor analysis was conducted by inspecting every aspect regarding the UX / UI of the Ekare mobile solution and sorting them as good, bad and ugly. This gave further insight into what was expected of a mobile solution utilizing a MLA to allow users to assess and validate wound images.

Some of the highlights found were the fact that the camera gives feedback to the user regarding the correct distance to take pictures and combined with the crosshair it helps immensly to know how the camera has to be positioned for wound pictures.

An example of bad UX is that the most important activity editing the wound determination is not user-friendly because the placement of buttons are not thought-out and their functionality did not work with the users' intent in mind. Instead the features seemed to be implemented from a developers' point of view.

The analysis concluded that from a UX / UI point of view, the application is a step in the right direction but requires further improvements as well which are great.

Figure 15: Ekare Wound Correction Activity

4. The Product - Design

The design chapter starts with the requirements that were gathered and continues with the HMI: prototyping phase in which the stakeholder requirements are used to create a prototype as well as technical designs which describe the desired technical situation that was implemented.

4.1 Requirements

Requirements were drafted using the stakeholder interviews and background research conducted. These requirements are numbered and tagged with either an F or NF where F stands for functional and NF stands for non-functional. These requirements were then prioritized according to the MoSCoW method.

Table 4: Gathered Requirements Prioritized According to MoSCoW

F / NF	Prioritization	Requirement
F.1	Must	Allow for user-correction of the wound image mask using image editing tools (i.e., draw, remove, resize & delete) in order to shape the wound area and to correct the Vesalius MLA wound assessment masks.
F.2	Must	Allow for user-correction of wound image masks according to the WCS-model plus an extra color for undetermined wound tissue in such a way that the bitmaps do not overlap.
F.3	Must	Edit and Send user-corrected wound masks to the Vesalius back-end.
F.4	Must	Store the MLA's determined pictures and their masks in a file structure that congregates wound pictures based on hashed ClientID's on the Vesalius MLA back-end storage.
F.5	Must	Facilitate a newly trained TFLite model that can determine five masks instead of one. (boundary + WCS + other)
F.6	Must	Adhere to the theme and iconography of the Ons Wondzorg SaaS platform so that it fits seamlessly into the platform as module.
F.7	Must	The Vesalius module must become a part of the Ons Wondzorg SaaS platform so that nurses can opt-in to use the module and gather data for the Vesalius MLA to re-train on.
F.8	Should	Show the combined percentages of wound colors present in the current edit view.
F.9	Should	Obfuscate client picture metadata using hashing so that the owner of the wound image cannot be derived.
NF.1	Must	Load the new MLA model within a reasonable timeframe. (<2s)
NF.2	Must	Generate the wound mask label results within a reasonable timeframe. (<2s)
NF.3	Must	Send the results to the MLA back-end storage within a reasonable timeframe. (<2s)

4.2 HMI: Prototyping

Using the previously researched UX and competitor analysis as well as the skills taught during the specialization HMI, a front-end UI / UX design was made.

The main intent of the Hi-Fi prototype is to test and validate the interactions and features so that the implementation of the technical solution can be done with a clear intent.

This prototype focused mainly on the ability to use the MLA wound assessment model by the click of a button or the snap of a picture whereafter the user can verify the wound assessment and use tools easily in order to correct it by navigating the individual WCS colors plus white for alternative tissue marked as "other".

The Figma prototype as shown in the figure below contains the most important functionalities that were required for the solution in order to expand the assessment of a wound image using multiple labels. It starts at the left-bottom yellow activity which insinuates that we are still in the Ons Wondzorg platform wound review whereafter the user can start the Wondzorg+ (Vesalius) module as depicted in orange. The middle bottom activity shows the assessment made by Vesalius and by clicking edit, the user can start the editing activity so that the wound color labels can be adjusted.

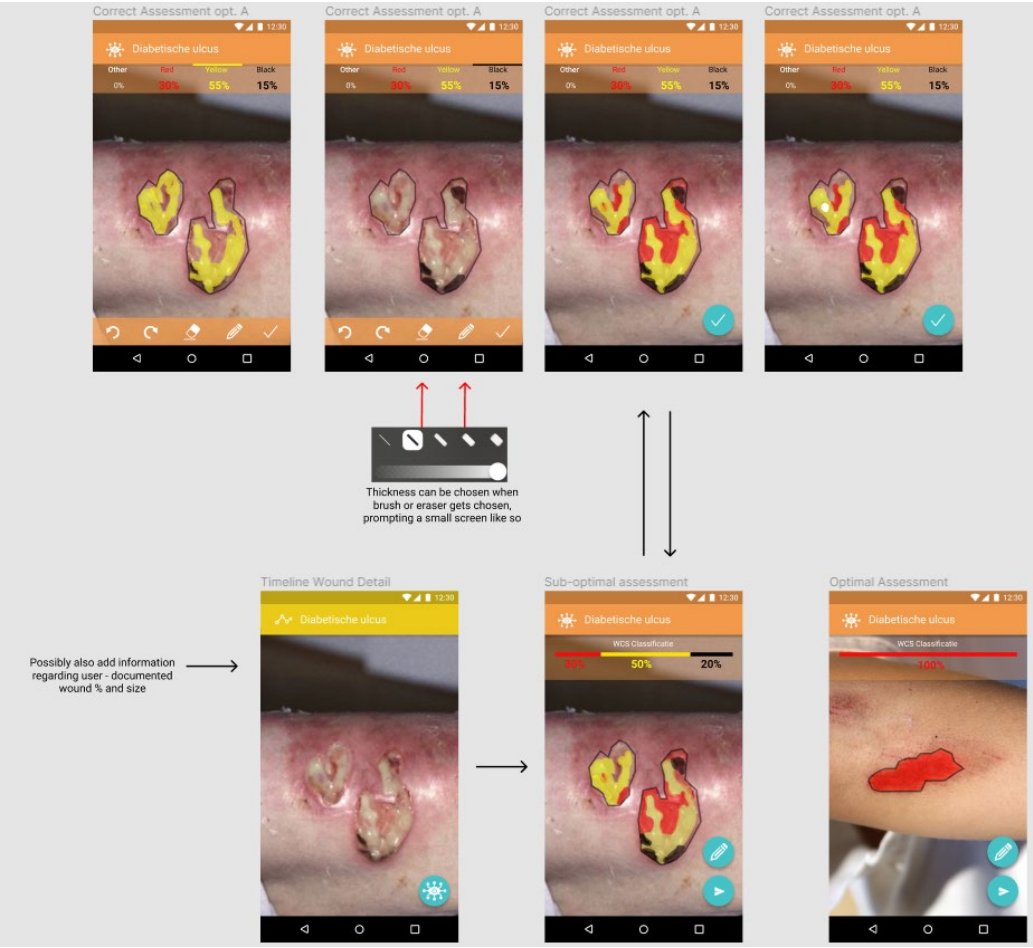


Figure 16: Figma Hi-Fi Prototype of the Vesalius UX / UI

4.3 Technical Design

The schematic below offers a high-level overview of the new situation. The steps taken in the Vesalius Android front-end are identical to the starting situation except for the second to last step. Compared to the starting situation, in the improved situation, the user can edit multiple wound labels as separate, colored masks in the mobile solution. Instead of solely assessing the wound boundary, the WCS tissue colors plus an extra color for undefined tissue can be adapted and uploaded to the Vesalius Back-end.

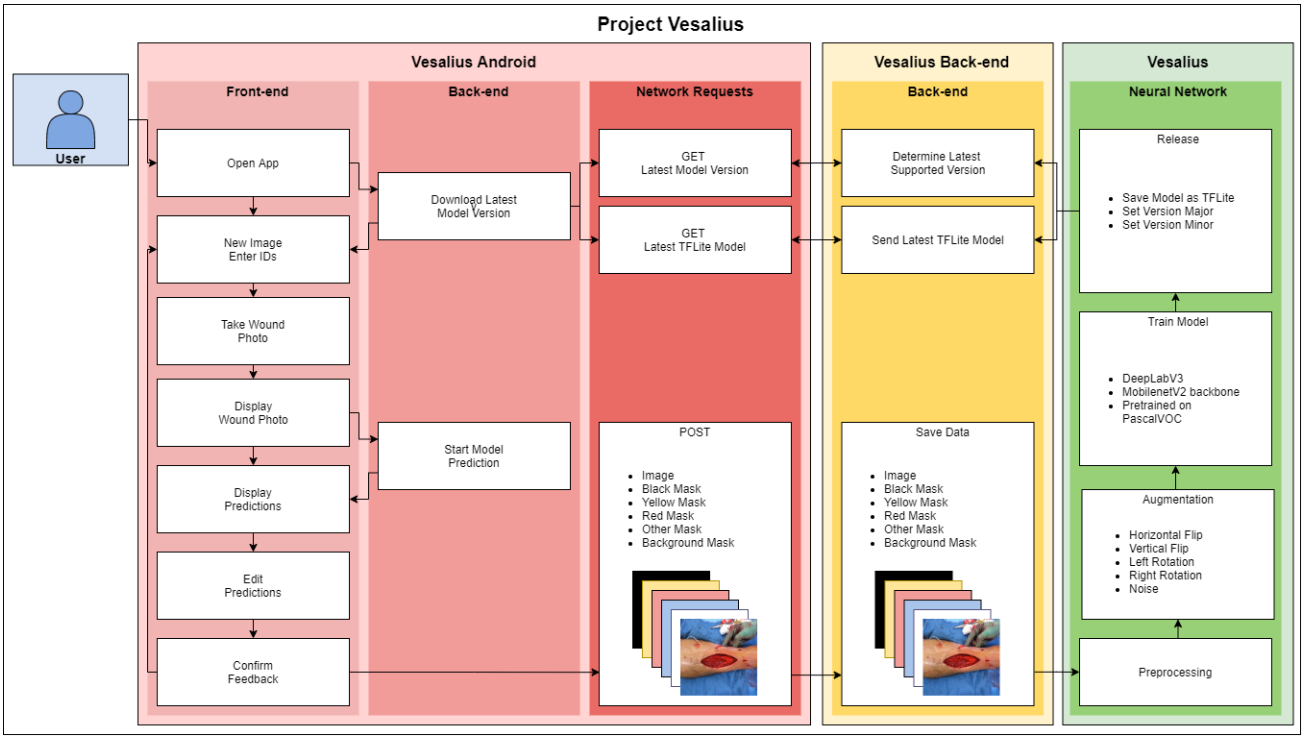


Figure 17: The Vesalius Solution Overview - New Situation

The API has therefore also been adapted in order to upload the new dataset consisting of multiple labels and the wound image. In the back-end most of the pipeline is similar except for the way the data is send from the front-end is adapted in order to facilitate multiple masks. The new POST request can be seen in the third column in the schematic, the network requests. The new POST information consists of a more elaborate dataset compared to the starting situation.

The principal challenge was getting to understand the current solution and how the modules, libraries and utility classes function together. The starting situation was an elaborate software stack build by multiple thesis students that each improved different parts of this system. Due to the fact that ms. Nijman worked alongside and often together with the student, certain implementations and adaptations to the Vesalius Android and back-end solution had to be made after the research and software design phase as completed because both of our projects progressed in parallel and changes on one side of the solution meant there had to be adjustments on the other end as well.

The complexity of the solution came from the fact that many changes had to be made across the stack of technologies in order to facilitate a new and differently trained MLA model. Combined with the overhaul of the front-end interface and the change / addition of new features such as the ability to expand and decrease separate masks without them clashing.

Some additional libraries were considered, such as the change from the "fotoapparat" library to the Ons Wondzorg in-house camera library so that platform integration would be realized more easily. And a python hashing library so that the data stored on the back-end could be obfuscated. The same library was adapted for the use of the new TFLite model that produces five labels and works according to instead of one, more on that in the next chapter.

Front-end

The improved front-end solution functions similarly to the starting situation in regards to activity sequence and functionalities outside of the *editwoundactivity*. The main purpose of this assignment was to expand the *editwoundactivity* and allow it create multiple masks that define the colors of the wound as well as its boundary. Besides the previous point it was also important to overhaul the Vesalius mobile application UX / UI according to HMI principles and stakeholder requirements.

With these points in mind, a design was drawn up and tested for its UI / UX using the Design Thinking Testing phase where organizational users and stakeholders were able to interact with the Hi-Fi prototype and leave feedback or validate the UI / UX. The results were promising and therefore have been implemented as similarly to the Hi-Fi prototype as possible.

The front-end adjustments have been made using Kotlin since that is the chosen programming language for the Ons Wondzorg mobile platform. The repository was pulled from the private GitHub repo and a specific branch was made.

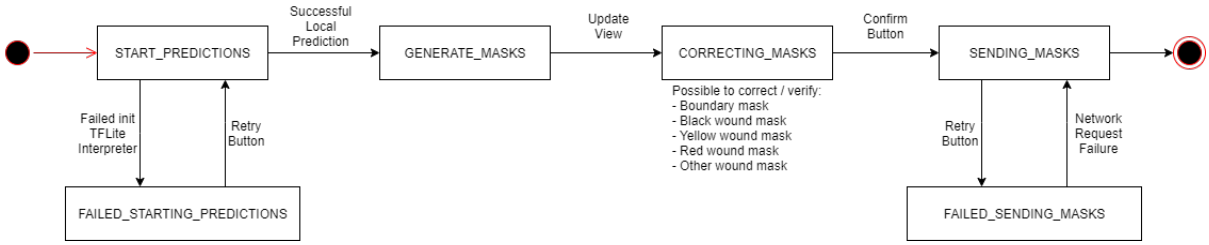


Figure 18: Android State Diagram - New Situation

The new Vesalius MLA is trained using: https://github.com/bonlime/keras-deeplab-v3-plus which is a multi-class classification model instead of the binary classification model. This meant that the front-end solution needed to adjust its mask predictions from a threshold value between 0 and 1 to an argmax value that is based on the resolution of the image which get compared with the output of the model per pixel to determine the *thresholdByteArray*.

```
Old:
thresholdByteArray[index] = (probabilityMatrix[x][y][0] * 100).toInt().toByte()

New:
val argmax = 1.0 / (imageDimensions.height * imageDimensions.width)
if (probabilityMatrix[x][y][0] >= argmax) ByteArray[index] = 1
```

So instead of multiplying the probability matrix determination value times 100 and converting it, the new solution compares the calculated argmax value (based on the resolution of the prediction) to the probability matrix and assigns a 1 to the byteArray if the argmax value is bigger or equals the predicted value.

API & Back-end

Adjustments had to be made to the python back-end in order for the data pipeline to function since there have been changes to the Vesalius MLA made by the stakeholder ms. Nijman. Since the front-end solution now creates multiple wound masks and therefore the dataset is expanded, the POST request had to be adapted similarly as seen in appendix C: API Endpoints.

Furthermore, the storage of the datasets on the Vesalius back-end had to be adjusted as well due to the fact that the original solution dropped folders with the ClientID and Organization per assessment which became cluttered quickly. For this reason, the wound images and labels had to be congregated per client and organization so that it is easy to distinguish between them when the wound images are checked manually for quality assurance. This means that the images will be checked whether or not there are any tattoos, faces or words that can be deduced back to the owner of the wound. By congregating the data into organizations and clients (that are obfuscated) it is easier to distinguish which organization produces subpar data and can be excluded from the dataset that is used to retrain the Vesalius MLA.

Lastly, the client related information had to be obfuscated as requested by the stakeholder & privacy and security officer. This is done using hashing on the ClientID so that it cannot be derived back to the original client.

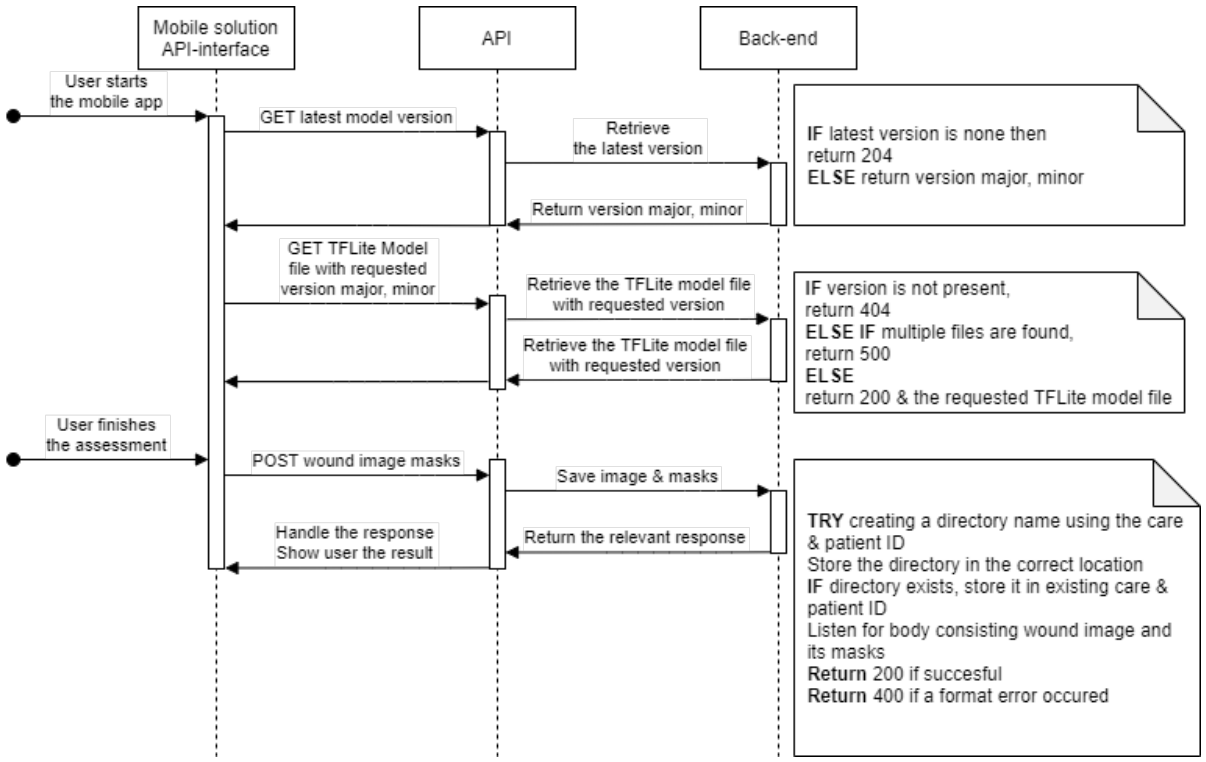


Figure 19: Sequence Diagram of the Server - Client Communication

5. The Product - Realization

This chapter contains the realization phase of the project. It starts with the HMI testing & validation of the prototype as well as the implemented front-end & back-end functionality.

5.1 HMI: Testing & Validation

After the Figma prototype was finished, the next step was to test and validate the main features, it was shared on the Ons Wondzorg Slack channel as Google Survey. In this channel, the main stakeholders of the solution assessed the prototype and gave some feedback regarding the iconography and lay-out.

The main arguments made were to make sure that the final implementation used material design so that it adheres to the Ons Wondzorg platform and that the features used editing tools as described in the requirements so that the editing of wound pictures could be done as user-friendly as possible.

Survey Results

The survey concluded with n=5 where n is a diverse group of people including the organizational users, stakeholders and other HMI specialists. The overall results were positive regarding the Hi-Fi prototype and its UX / UI. According to the answered questions it was somewhat clear how the module is activated within the Ons Wondzorg mobile platform and it was very clear how the Vesalius solution determines multiple masks that can be assessed and edited by the user.

Overall, the users were satisfied with the design and mentioned it was a worthy addition to the Ons Wondzorg platform. An interesting mention was the green crosshair that was added to the AI camera capture activity and it was suggested to add a text that mentions explicitly that this was going to be an AI assessment.

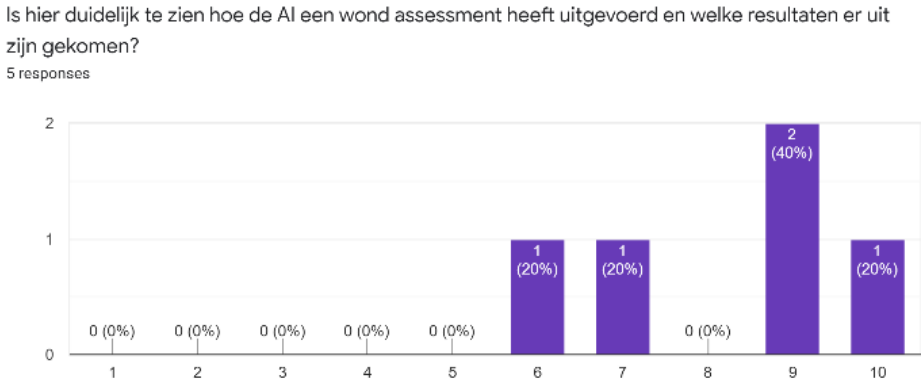


Figure 20: A Survey Result Regarding the Effectiveness of the Hi-Fi Prototype

5.2 Front-end Application

Multiple Masks according to WCS Model

The main objective of the assignment was about the ability to expand on the existing functionality. The main requirement being that the user should be able to assess the wound image on more aspects than its boundary. The new solution is therefore able to instantiate and draw wound color masks according to the WCS model besides adapting solely the wound boundary mask.

This is implemented based on the original bitmap functionality which only allowed for one mask to be drawn on top of the image in the *AdjustWoundArea* module. The new solution instantiates five buttons that represent the five masks that have to be assessed and corrected. Switching between the masks using the buttons will select which bitmap is to be edited using the editing tools present in the bottom bar as seen in the figures 21 & 22 on the next page.

Extra logic in the *AdjustWoundAreaView* was necessary for the new bitmaps to keep track of each other in the way that:

- The wound boundary bitmap is always the ultimate boundary of any wound color bitmap so that the wound color bitmaps cannot exceed the wound boundary. This bitmap is currently drawn in green.
    - The wound color bitmaps cannot draw onto each other and will always have to share the pixels within the wound boundary bitmap. These bitmaps have the colors; black, yellow, red & blue respectively.

- When the reset button is pressed on the wound boundary bitmap, it defaults back to the Vesalius wound boundary assessment and the wound color bitmaps get deleted. Essentially starting from the beginning.
    - When the reset button is pressed on a wound color bitmap, that specific bitmap is deleted and the rest of the assessed wound image remains intact.

UI / UX Overhaul According to Ons Wondzorg

Following the design of the Hi-Fi prototype, the overall UI has been overhauled to conform to the Ons Wondzorg wound care platform as stated by the product owner. This consisted mainly of XML color scheme changes, Material Design iconography changes as well as the addition to Dutch strings for automatic translation in Dutch versions of the android OS.

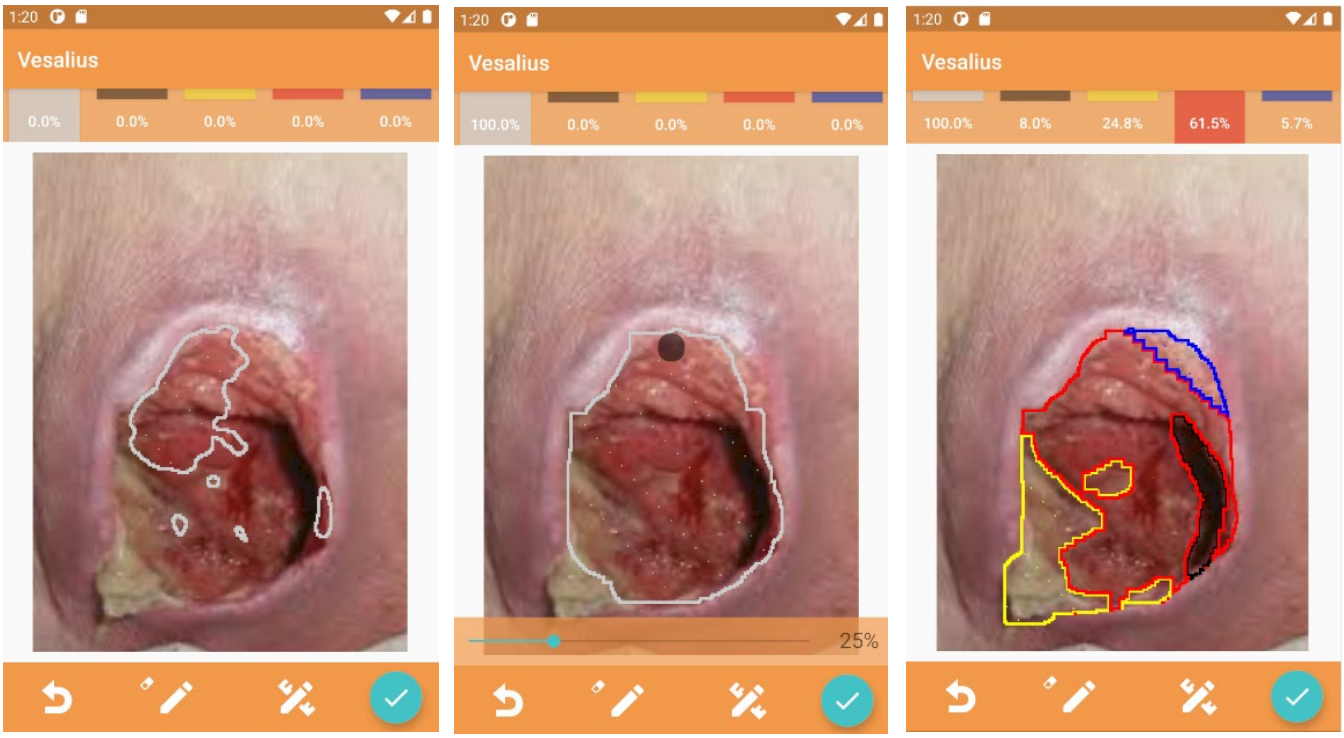


Figure 21: Vesalius Front-end – Implemented Wound Mask Correction UI & Multi-Mask Functionality

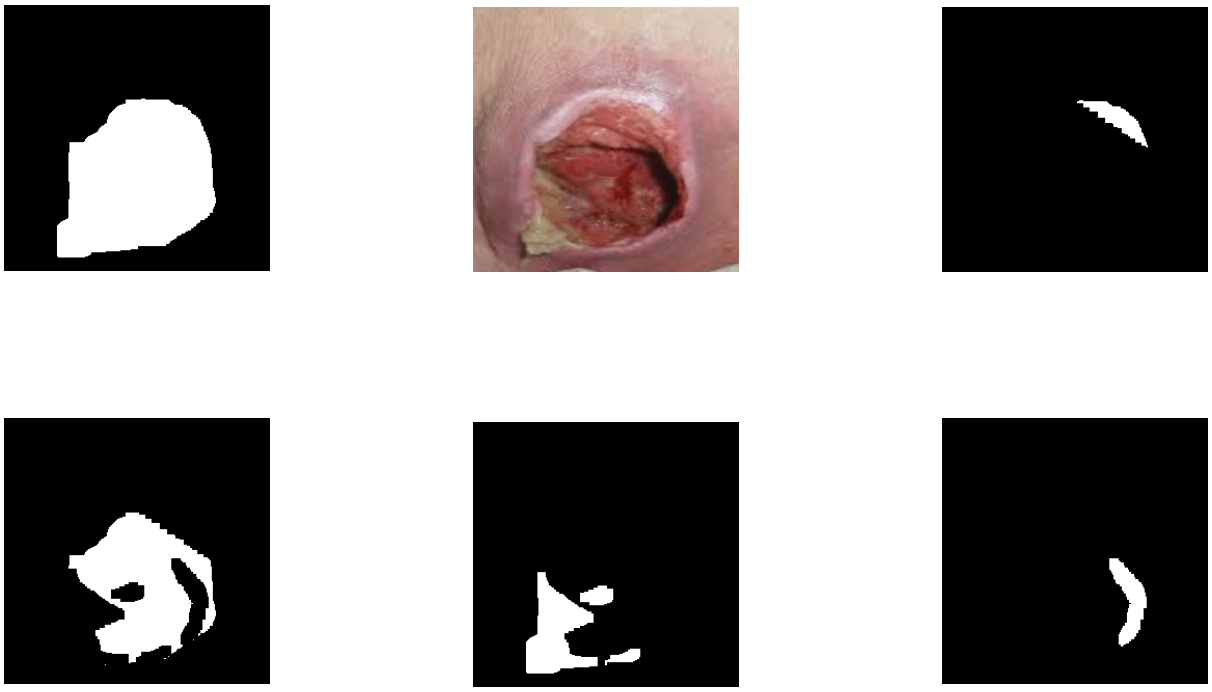


Figure 22: The Five Masks Plus Wound Image Scaled for Vesalius Re-trainability Purposes Fetched from the Vesalius back-end

5.3 Back-end Application

Adjusting the Vesalius Mobile API Interface

The currently implemented Kotlin data-classes and API interface had to be adapted in order for the new dataset to be uploaded to the Vesalius server. The new labels produced by the current Vesalius MLA do not need certain attributes such as the wound threshold as mentioned by ms. Nijman, the threshold was a value that was also given by the user to determine the correctness of the assessment according percentual value per pixel as depicted on the previous page. The new Vesalius model has been pre-trained to already work according to the optimal threshold value.

Furthermore, the new wound data-class has multiple wound masks that determine the boundary and the colors of the wound. Therefore, the new data upstream had to account for these changes which were made in the “AdjustWoundArea” module.

Adjusting the Python Back-end

The Vesalius Python back-end containing the code which transforms the data upstream containing the dataset into the correct format of JPEGs had to be adjusted as well. Instead of expecting a POST containing one user adjusted mask plus image, it would now receive a dataset with the aforementioned dataset containing multiple masks and the image. This has been realized by expanding the API call, “post wound image mask” so that it receives, converts and stores the wound image masks in the correct folder until

The size of the JPEGs had to be adjusted as well, due to the fact that the new Vesalius MLA model trains on images of 224 x 224 resolution instead of 256 x 256.

6. Results

This chapter shows the table of requirements and current status whether they are completed or not.

Table 5: Requirement Results

F / NF	Completed	Requirement
F.1	Yes	Allow for user-correction of the wound image mask using image editing tools (i.e., draw, remove, resize & delete) in order to shape the wound area and to correct the Vesalius MLA wound assessment masks.
F.2	Yes	Allow for user-correction of wound image masks according to the WCS-model plus an extra color for undetermined wound tissue in such a way that the bitmaps do not overlap.
F.3	Yes	Edit and Send user-corrected wound masks to the Vesalius back-end.
F.4	Yes	Store the MLA's determined pictures and their masks in a file structure that congregates wound pictures based on hashed ClientID's on the Vesalius MLA back-end storage.
F.5	No	Facilitate a newly trained TFLite model that can determine five masks instead of one. (boundary + WCS + other)
F.6	Yes	Adhere to the theme and iconography of the Ons Wondzorg SaaS platform so that it fits seamlessly into the platform as module.
F.7	No	The Vesalius module must become a part of the Ons Wondzorg SaaS platform so that nurses can opt-in to use the module and gather data for the Vesalius MLA to re-train on.
F.8	Yes	Show the combined percentages of wound colors present in the current edit view.
F.9	No	Obfuscate client picture metadata using hashing so that the owner of the wound image cannot be derived.
NF.1	No	Load the new MLA model within a reasonable timeframe. (<2s)
NF.2	No	Generate the wound mask label results within a reasonable timeframe. (<2s)
NF.3	No	Send the results to the MLA back-end storage within a reasonable timeframe. (<2s)

7. Conclusion & Recommendations

This chapter consists of the conclusion which starts with the requirements table and what is implemented thus far as well as further elaboration on the concluding situation. The chapter also has future recommendations which describes possible improvements and other relevant information for the Vesalius wound recognition software.

7.1 Conclusion

The **main** goal of this graduation assignment was to expand the Vesalius mobile solution so that it could allow for users to correct a dataset consisting of multiple wound labels instead of the single wound boundary label alone. These multiple labels required logic in order for them to create a wound dataset consisting of an outer boundary plus four inner colors according to the WCS model and an additional color for undefined wound area.

A **secondary** goal was to improve on the UX / UI of the Vesalius mobile solution so that it conforms to the Ons Wondzorg platform theme as well as create a design through HMI & Design Thinking that allows for users to interact with the Vesalius application with relative ease.

Both of these goals have been achieved with emphasis on the main goal. Getting to understand the various layers of the solution during the research phase as well as conducting a HMI – design thinking approach required precious time that would otherwise result in one or more extra sprint cycles. Nevertheless, without this design thinking approach, ms. Nijman and the graduation student wouldn't have gotten the user feedback to add another label on top of the WCS model as well as several suggestions and improvements regarding the lay-out, design and UX of the mobile solution from the stakeholders as explained in appendix K: Organizational Users and in appendix I: Competitor Analysis.

More work was planned for the Vesalius solution, such as component integration in the Ons Wondzorg platform and the integration testing of the new TFLite model and the three non-functional requirements but personal circumstances and the situation surrounding the assignment have made it challenging to fulfill the implementation to its current extent.

In the week of the deadline of this report, ms. Nijman put the new multi-class TFLite model on the Vesalius back-end which got deployed on the 9th of June. This model got pulled onto the mobile device but is not finished being implemented, as shown in the last paragraph and code-snippet on Chapter 4.3: Front-end.

For this reason, Requirement F.5 and NF. 1 to 3 are still in progress and will be completed in the week following the deadline of the report. F.7 and F.9 have been dropped in agreement with the product owner and company supervisor, mr. Klitsie.

7.2 Future Recommendations

Camera Activity User Feedback

In order for the Vesalius MLA to create accurate predictions of the wound boundary and its colors, a lot of user-assessed data will have to be collected in order to re-train the Vesalius MLA. This data will have to adhere to specific standards in order for the MLA to become more accurate. One such standard is the fact that each pixel can only belong to one mask so that they don't overlap, which is achieved in this project. Another standard is the fact that ideally each picture should be taken from the same angle and distance from the wound as all others in a dataset as depicted in the competitor analysis figure 29.

Vesalius Tutorial

Whenever a user starts the smart vision assessment component via the Ons Platform, the user will ideally be able to use the feature seamlessly and intuitively. This is however often not the case and most applications which introduce new features do so with a guide of some sort.

As seen in the Ekare competitor analysis, the features of the wound assessment solution get introduced briefly using some images and an explanation. Ideally the Vesalius wound assessment solution will contain at least an introduction as such or preferably a guide through the assessment activity with the various options available. This is already being worked on in the extensive Ons E-learning course being developed by Nedap Healthcare.

Vesalius Front-end Ons Wondzorg Integration

With expansive work done on the Vesalius mobile application, allowing it to adjust multiple masks as well as an improved Vesalius machine learning algorithm and extended back-end functionalities the solution is coming along nicely. A next step for the Vesalius project would be to integrate the solution into the Ons Wondzorg platform where the emphasis has to be put on extensive research into the Ons Wondzorg mobile application and how to connect the relevant data from this application into the Vesalius application.

Internal Vesalius Wiki

When a project such as Vesalius is launched or improved, it would be wise to create some form of internal wiki where each employee, graduate student or intern can document their findings and improvements in a structured centralized manner.

This wiki can contain extensive information regarding relevant reports made by the students, the various repositories and justifications as to why they were built as well as the way to deploy them on various environments used by the students for local development & testing.

A wiki could help to onboard new students quicker and more effectively due to the fact that it centralizes and maintains the necessary information required to continue the development of a solution.

7.3 Final Remarks

As the project is being finalized and a presentation is being prepared, there are noticeable improvements made to the Vesalius solution by both the graduation student and the data-scientist intern ms. Nijman. The results produced will be a great addition to the already existing solution and it is only a matter of time until the Vesalius wound assessment feature becomes a viable product to be added to the Ons Wondzorg platform so that it may help a lot of healthcare professionals save time and gain a consistent follow-up over time.

8. Bibliography

[1] Ekare, "About Ekare," Ekare, 1 04 2021. [Online]. Available: https://ekare.ai/about-ekare/. [Accessed 1 04 2021].

[2] W. v. Gabain, "Figma," Figma, 1 04 2021. [Online]. Available: https://www.figma.com/proto/Jju0Hrnz7ttNJsKBeJW2Vs/Vesalius-Component?node-id=120%3A24&scaling=min-zoom&page-id=0%3A1. [Accessed 1 04 2021].

[3] W. C. Society, "WCS classificatie model," WCS, 01 01 2018. [Online]. Available: https://www.wcs.nl/over-wcs/wcs-classificatiemodel/. [Accessed 19 04 2021].

[4] W. v. Gabain, "Survey Vesalius component user validatie," Nedap B.V., 21 04 2021. [Online]. Available: https://forms.gle/oe5SH5CkRKMSQKCJ9. [Accessed 21 04 2021].

[5] I. O. f. Standardization, "https://www.iso.org/," ISO, 01 01 2021. [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010. [Accessed 21 04 2021].

[6] Guru99, "How to Write Test Cases: Sample Template with Examples," Guru99, 01 01 2021. [Online]. Available: https://www.guru99.com/test-case.html. [Accessed 30 04 2021].

[7] TensorFlow, "Image classification," TensorFlow, 22 02 2021. [Online]. Available: https://www.tensorflow.org/lite/examples/image_classification. [Accessed 30 04 2021].

[8] Material.io, "System Icons," Material.io, 01 01 2021. [Online]. Available: https://material.io/design/iconography. [Accessed 03 05 2021].

[9] A. Fard, "The Ultimate Guide to Stakeholder Interviews," adamfard, 1 January 2020. [Online]. Available: https://adamfard.com/blog/stakeholder-interviews. [Accessed 29 April 2021].

[10] İ. Bilgilerimiz, "Design Thinking," TOFAS Akademi, 01 01 2021. [Online]. Available: https://tofasakademi.com/design-thinking-2019/. [Accessed 20 05 2021].

[11] M. Hudepohl, "Post-it," Saxion Hogeschool, 01 06 2020. [Online]. Available: https://postit.saxion.nl/resolver/getfile/B8C4C080-0DB6-402A-98188912E8C4A64A. [Accessed 06 05 2021].

[12] M. Kuhlmann, "Post-it," Saxion Hogeschool, 01 07 2019. [Online]. Available: https://resolver.saxion.nl/display_details/F2C15F97-8CEE-4EFF-B9A36D4BB0308C40. [Accessed 06 05 2021].

[13] J. Ross, "The Pomodoro Technique | A Productivity Guide," Medium, 21 June 2016. [Online]. Available: https://medium.com/manager-mint/the-pomodoro-technique-a-productivity-guide-908c73619e9. [Accessed 01 05 2021].

[14] D. N. &. J. Nielsen, "The Definition of User Experience (UX)," Nielsen Norman Group, 01 01 2021. [Online]. Available: https://www.nngroup.com/articles/definition-user-experience/. [Accessed 28 05 2021].

[15] Scrum.org, "What Is Scrum?," Scrum.org, 01 01 2021. [Online]. Available: https://www.scrum.org/resources/what-is-scrum. [Accessed 09 06 2021].

[16] C. Chapman, "Exploring the Gestalt Principles of Design," Toptal, 01 01 2021. [Online]. Available: https://www.toptal.com/designers/ui/gestalt-principles-of-design. [Accessed 10 06 2021].

9. Appendix

A. List of Figures

B. List of Tables

C. API Endpoints of the New Situation

Table 6: Latest Model Version

Endpoint	**/ latest_model_version**
Description	This endpoint is used to retrieve the latest model version for the mobile solution. The given major and minor arguments are hardcoded in the Kotlin solution to depict which version of the model the solution is currently supporting.
Request Type	GET
Request Data	Major (integer) - The major version supported in the app which is currently set to 1. Minor (integer) - The minor version supported in the app which is currently set to 0.
Result	Success: Sends the information regarding the latest version of the given major and minor available as major, minor & patch. Error: If no models are found according to the requested major and minor then a 204 code is sent.

Table 7: Retrieve Specific Model

Endpoint	**/ model**
Description	This endpoint is used to retrieve the actual trained TFLite model as requested by a major, minor and patch.
Request Type	GET
Request Data	Major (integer) – Major version of the to-be requested model. Minor (integer) – Minor version of the to-be requested model. Patch (integer) – Patch version of the to-be requested model.
Result	Succesful: A 200 code with the TFLite model attached. Error: An error code as described in the API.yml and a description of the issue.

Table 8: Send Wound Image and Determined Masks

Endpoint	**/ wound_image_masks**
Description	This endpoint is used to send and collect the wound image and its masks. It contains the organization & client information as well as the determined wound image and its mask data.
Request Type	POST
Request Data	Body: • Healthcare organization and client ID. • The size of the made picture. • The determined masks converted to base64 string.
Result	Succesful: “OK” and a status code 200. Error: A description of the issue as described in the API.yml and a status code 400.

D. Tools & Frameworks

Technical Tools

Table 9: Technical Tools

Name	Use
Android Studio	This IDE is built on Jetbrains' IntelliJ specifically for Google's Android development and has integrated emulator tool that can run multiple emulators and benchmark their performance running the front-end solution.
IntelliJ	This IDE is designed for Java development. IntelliJ was used briefly to generate UML of the current solution during the background research. IntelliJ Pro has useful features when it comes to Java development and it was surprising to see that the feature that allows for UML generation would also apply to Kotlin code as well as it did.
Visual Studio Code	This IDE was used for the back-end repositories because it contains a lot of useful features for Python development since VSC contains a Python coding extension. Together with the build in terminal and optional importable extensions such as Pylint and Python Preview makes VSC a great tool for this scenario.
Git(Hub)	Git was used for code version control and code review, this was done through the use of GitHub for the reason that this is where Nedap keeps their repositories such as Vesalius (MLA), Vesalius Android (front-end) & Vesalius Android back-end.
SCRCPY	Free and open-source screen mirroring application that allows control of android devices from a Windows, MacOS or Linux computer.
Conda	An open-source, cross-platform, package management system and environment management system used to create Python applications.
Figma	Figma is a collaborative interface design tool that is able to edit vector graphics as well as prototype designs. Figma is built to be used cooperatively via a web-based interface where designs can also be shared and prototyped with ease. A Ons Wondzorg Figma design was shared with the Graduate Student by the Nedap design team. This template was adapted in order to create the interactive Vesalius application prototype.

Project Tools

Table 10: Project Tools

Name	Use
Jira	Jira is an online work management tool used to track product development in an Agile framework with additional features that make it possible to see all sorts of statistics regarding sprint completion and burn-down charts. The sprints consisting of user stories and overall tasks required to finish the graduation assignment were managed and documented using this tool.
Gantt-Chart	A Gantt chart is a type of bar chart that illustrates a projects' schedule. A Gantt chart helps to oversee the general progress of a project in relationship to the individual phases, tasks and deliverables. The Gantt chart used for this project was taken from the Excel Gantt chart template in which every phase of the project can be added per horizontal row. Each phase can afterwards be defined based on expected duration, expected start, expected finish and during the project's advancement these values can be compared to the actual duration, actual start and actual finish. The overall progress of the project can be assessed and adapted quickly using this planning method.

Documentation Tools

Table 11: Documentation Tools

Name	Use
MS Office	The Microsoft Office suite was used for many different cases and situations. Word, Excel and PowerPoint were used for documentation, planning and presentation.
Google Drive	This tool was used in order to keep the vital documentation such as the research report and graduation document backed-up. Using a cloud-based storage solution, the documentation is easily accessible from multiple machines in case the PC started to malfunction in any way.

Communication Tools

Table 12: Communication Tools

Name	Use
Outlook	A fair amount of communication between the graduate student and Nedap / Saxion has been done through Outlook. E-mails were send to schedule meetings with stakeholders and colleagues as well as to communicate questions with the Saxion supervisor.
Zoom	This audio/video communication tool was used on a daily basis when communicating with the Nedap Healthcare colleagues for the daily stand-ups, stake-holder meetings, weekly progression meetings and more.
MS Teams	This audio/video communication tool was used for daily meetings with another classmate, Mihail Ciobu, for pomodoro sessions and general discussions regarding the graduation assignment as well as weekly stand-ups with other graduate students and the coach.
Slack	Slack is a business communication platform that offers features such as persistent chat rooms organized by topics, private groups and direct messaging.
Discourse	Discourse was used in order to communicate with Nedap Healthcare's clients (users) regarding stakeholder interviews and design thinking user testing & validation.
Google Surveys	This tool was used to create an easy to read, quick to fill out survey regarding the prototype in the HMI – design thinking phase. The survey was made and published on Discourse in order to test & validate the Figma prototype.

E. User Stories

Table 13: User Stories

#	Story
1	As a **user**, I want to be able to use **image editing tools** (i.e., draw, remove, resize & delete) in order to **edit the Vesalius wound boundary mask** as accurately as possible.
2	As a **user**, I want to be able to use **image editing tools** (i.e., draw, remove, resize & delete) in order to **edit the Vesalius wound assessment color masks** according to the WCS model as accurately as possible.
3	As a **user**, I want to be able to **add a fourth color** using editing tools to the mask image in order to **indicate wound tissue** that cannot be classified according to the WCS model.
4	As a **user**, I want to be able **to see the wound-related information** of the Vesalius assessment, such as the **WCS color % in the UI** so that the wound and its colors can be assessed more clearly.
5	As **product owner**, I want the Vesalius assessments to be **stored on the Vesalius back-end** in a congregated form according to organization & client IDs so that they can be **used for re-trainability** purposes.
6	As **product owner**, I want the Vesalius assessments to be **hashed** so that the client-related metadata is obfuscated and therefore **cannot be derived back** to its original owner according to privacy law.
7	As a **product owner**, I want the Vesalius component to be **integrated into the Ons Wondzorg platform** and function according to the platform software related standards regarding its authentication and authorization of users.
8	As a **product owner**, I want the Vesalius component to **conform to the Ons Wondzorg UI** theme and UX so that it can be used **intuitively and fits inside the platform**.
9	As a **user**, I want the Vesalius Ons Wondzorg component to be as **user-friendly as possible** so that it can be used during **quick wound check-up routines** without taking much time.
10	As a **user**, I want the image editing and correction tools to **include a way to zoom into a wound** picture so that the **corrections can be made in a more detailed view**.
11	As a **user**, I want to be able to **slide the transparency of the color layers** so that it can be **compared to the wound photo beneath the layer**.
12	As a **user**, I want the Vesalius component **to use the same iconography** as the main Ons Wondzorg mobile platform so that there are **no inconsistencies** between the Vesalius solution and the main platform.

F. Overall Code Statistics

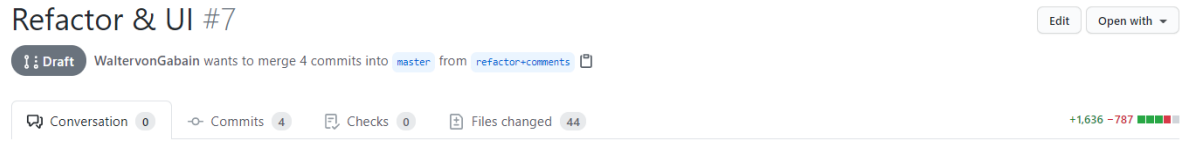


Figure 23: First Branch - Refactor & UI

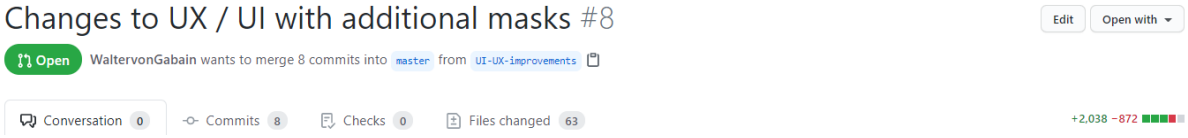


Figure 24: Second Branch - Changes to UI / UX and Additional Mask Functionality

G. Example Test Case

Test Case ID	US_2	**Test Case Description**	Test the testcase		
Created By	Walter	**Reviewed By**	None	**Version**	1
Tester's Name	Walter	**Date Tested**	May 25, 2021	**Test Case (Pass/Fail/Not**	Pass

S #	**Prerequisites:**
1	Access to Front-end Application
2A	A downloaded model on the device / emulator
2B	Access to the internet to download a model

S #	**Test Data**
1	A wound picture, but can be tested without
2	
3	

Test Scenario Verify on entering valid userid and password, the customer can login

Step #	**Step Details**	**Expected Results**	**Actual Results**	**Pass / Fail / Not executed / Suspended**
1	Start the Vesalius front-end application	Application should load client input screen	As Expected	Pass
2	Input any user details	Application should continue to camera view	As Expected	Pass
3	Use the preloaded emulator image OR capture a wound picture	Image loads successfully	As Expected	Pass
4	After the wound assessment is completed, edit the wound boundary mask	Mask can be drawn, removed, reset.	As Expected	Pass
5	Repeat wound assessment steps per layer of the masks. (red, yellow, black & blue)	The WCS color layers cannot override the boundaries of the wound image NOR can they override the boundaries of eachother.	As Expected	Pass

Figure 25: Example Test Case of User Story Two

H. HMI: Design Thinking – Testing & Validation

After the Figma prototype was finished, it was shared on the Ons Wondzorg Slack channel so that it could test and validate the main features. In this channel, the main stakeholders of the solution assessed the prototype and gave some feedback regarding the iconography and lay-out.

The main arguments were to make sure that the final implementation used material design and that the features used photoshop – like editing tools so that the editing of wound pictures could be done as user-friendly as possible.

9.1.1 Survey

The survey (n=5) itself was added onto the Nedap Discourse platform where Nedap Healthcare and their client-base can communicate about the ongoing development as well as possible improvement suggestions made by the users of the Ons Wondzorg platform.

Five people filled out the survey, one of which was the stakeholder mr. Klitsie and another one was a fellow HMI student. This means that three Ons Wondzorg users have given their feedback regarding the prototype.

9.1.2 Results

According to figure 26, it was somewhat clear to the users that the prototype switched from the Ons Wondzorg platform to the Vesalius AI assessment module.

According to figure 27, it was very clear how the component assesses a picture and on what basis it judges the boundaries and wound colors.

Overall, the users were satisfied with the design and mentioned it was a worthy addition to the Ons Wondzorg platform. An interesting mention was the green crosshair that was added to the AI camera capture activity and it was suggested to add a text that mentions explicitly that this was going to be an AI assessment.

9.1.3 Conclusion

The results show that overall, the Figma prototype demonstrates the features reasonably well but it some aspects were still unclear.

Users were satisfied with the way the component made use of a crosshair, similarly to the Ekare application but users were not fully satisfied of the general lay-out as designed.

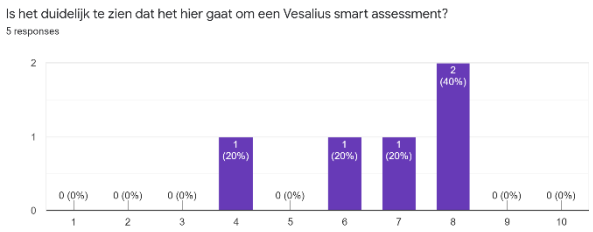


Figure 26: Question Regarding the Wound Recognition UI

Figure 28: Final Remarks from Testers

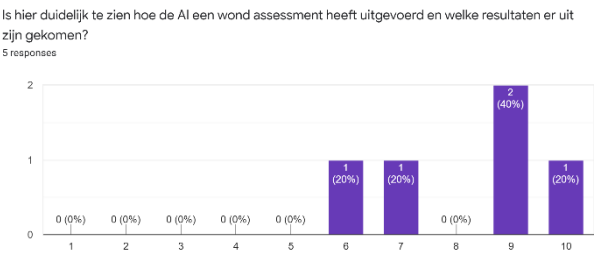


Figure 27: Question Regarding the Results of the Assessment

I. Competitor Analysis: Ekare Wound Assessment Solution

The Good UX

A Help center that contains the information / settings hub so that the user can seek all options and information in one orderly view.
When the main view appears, there are two ways to enter the settings menu, using the button and swiping from the left edge to right.

Optional rudimentary tutorial that explains the features using some text and an image.
A vital feature for the general user because this is a new technology that requires a basic level of guidance so that the user can get accustomed to the flow of the application.
Besides this, tutorials and guides in mobile applications can be made very intuitive and concise so that they will generally optimize usability without losing a significant amount of time.

Is able to estimate depth of a wound when using two types of cameras on the mobile device.
One very interesting feature that the Ekare wound-assessment application offers is the ability to measure depth of wounds. The usual method to analyze the depth is through a calibration-sticker but the Ekare application also offers the ability to compare the data from two cameras to calibrate the depth of the wound.

Immediately gives feedback regarding the picture, such as;

- Being too close / far.
- Obfuscating the background using black pixels.
- A colored crosshair & warning letters.

These are all examples of human-machine interaction done right because they offer a natural form of feedback to the user that require no contextual knowledge to interpret and can therefore be used very intuitively.

Intuitive wound assessment design showing the most important information in a thought-out design and also allows for some user correction.
The view in which the user can assess and correct the determined wound image and its data is very clear and has a good oversight.

Figure 29: Ekare – Too Close To the Wound

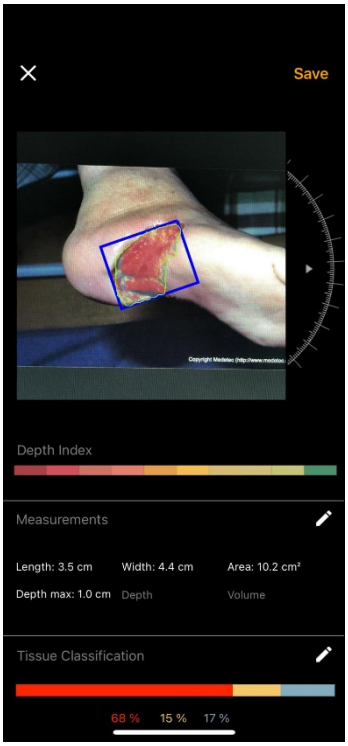


Figure 30: Ekare - Wound Assessment Activity

The Bad UX

Deletes password when correcting a wrong attempt.
Whenever a password is inserted in the login page, if it is wrong and the user attempts to correct it, it will delete the attempt and the user will have to enter the whole password again.

Depth index might be incorrect.
A clear statement cannot be made due to the fact that the Depth Index doesn't show exactly what the color scale means.
The wound is obviously deepest in the areas that the Index shows as green. Does this functionality work as intended? It cannot be determined due to the fact that the depth index has no context besides the color scale.

Reduce wound-size functionality is vague.
Instead of precisely correcting a certain spot that the ANN missed, every time the user draws, the correction remains visible which unintentionally makes the drawing look rather chaotic.
Combined with the fact that the undo / redo features do not work, this is the worst part of UX present in the Ekare wound-assessment application.

There is no back button, only an X (exit).
This is another part of the UX that shows that the developers did not put enough thought into the process. What if the user wants to remake a picture?

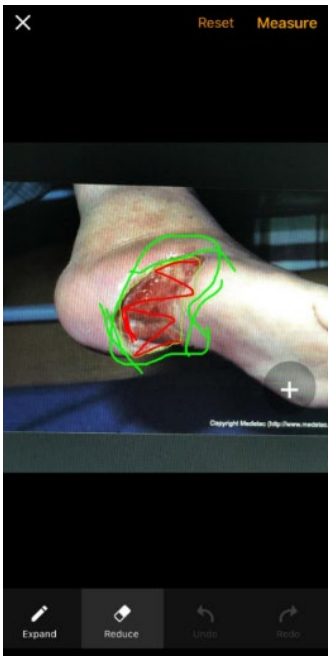


Figure 31: Ekare - Correcting a Wound Assessment

The Terrible UX

Difficult to assess what should be drawn exactly due to reduce & expand features.
As depicted in image X, the user-corrected areas get re-drawn by the application but the lines drawn by the user remain and make it a bit of a mess to correct the assessment properly.

The reset / measure buttons are text boxes instead of buttons.
This display of interaction stands out from the other features in the wrong way due to the fact that they are small, text boxes that do not look like they are part of a UI.
Furthermore, in order for the user to go through the procedure effortlessly, all buttons should be near the users' thumb (ideally) and on the Ekare user-correction page all buttons are far away from the right thumb.

Undo / Redo button are not functional, then why display them?
When attempting to make many corrections, they become ugly and the user would like to try over. For some reason, this feature is greyed out and unavailable during the testing of the application.

Conclusion

The application shows interesting features features. It is impressive to see the dual-camera depth measurement feature in action. This implementation is not relevant, but it might be worth a future-recommendation since more and more smart-phones brands are adding dual-camera abilities to their devices.

Besides the aforementioned remarks, the rest of the mobile solution shows room for improvement. The functionalities seem to be implemented in order to showcase the main features of Ekare, but are not as user friendly as they could be.

J. Example Sprint Log

Sprint 1

Set-up

I've aimed at starting sprint one on the 12th of April but due to personal circumstances and the overall progression of the research & the HMI process, the start of the sprint has been postponed to the 26th of April. This Sprint will focus on getting velocity going by starting to work on the Vesalius Kotlin application and the ability to use editing tools to correct the Vesalius assessment and its wound mask.

Besides this, the background document will be concluded with user testing and validation of the Figma prototype. While sprint one officially starts the 26th, I am still able to start working from tomorrow (22nd) and onwards due to the fact that background research is pretty much concluded at this point.

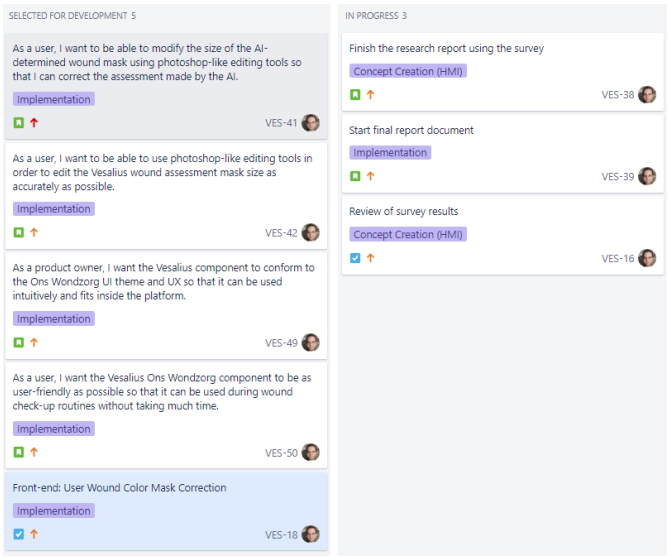


Figure 32: Sprint Backlog

Start

Before the second sprint started, I wanted some more insight into the Kotlin application due to the fact that there were multiple modules, libraries and utility classes that worked together. The overall composition of the front-end solution was somewhat messy. I started by refactoring the file structure and moved the main activity outside of the "*woundadjustment*" module, as it should be.

Next up, I attempted to generate UML using various tools that Android Studio had available or that could be imported, such as PlantUML. Unfortunately, these plug-ins either didn't work or did not produce the results as intended. A solution I found for this was to import the Front-end solution into IntelliJ pro (using my students' license) and use the UML generation feature there which produced class diagrams that were easy to read and allowed for great insight.

Figure 33: Written TODO's

Conclusion

I started the sprint with writing TODO's throughout the solution with personal findings and possible methods & classes to start in. The moment I wanted to launch the front-end solution on a pixel device it had issues importing the TFLite MLA model because of the way it was implemented. A solution I found to to this was to refactor the manifest, dependencies and more so that it would circumvent these warnings.

After some extensive programming sessions, I was able to add buttons that generate extra masks based on the wound circumference that can be edited as their own boundaries so that these can depict the colors within the wound according to the WCS model.

Lastly, I looked into the Android back-end repository as well as the Vesalius back-end repo, because these only work interpedently. I attempted to host these repositories locally but had no success due to the way they were implemented. I decided to leave this for the next sprint where I will seek further clarification and tips from the stakeholders by contacting them and setting up a meeting.

In this sprint I was able to complete all the selected user stories where two will still be user tested before the stories of sprint 2 will be completed.

K. Stakeholder Interviews

Product owner: Michiel Klitsie

Mr. Klitsie is the Ons Wondzorg product owner who is very enthousiastic about the implementation of the Vesalius smart vision wound assessment component into the Ons Wondzorg platform.

Mr. Klitsie is the graduate student's coach and one of the main stakeholders in this thesis project. Mr. Klitsie stated that there are a number of crucial steps left to be taken in order for the Vesalius ANN to be used as an Ons Wondzorg module.

The main concern being the fact that the front-end application is not user friendly since it is made to test the back-end and deep learning features of the module. Secondly, research will have to be conducted regarding the back-end capabilities of the application as well as its ability to store the user-corrected Vesalius wound assessments so that the Vesalius ANN can learn from these user-corrected images opposed to only wound images themselves.

Mr. Klitsie mentioned that it is important to test the solution on two emulators and a physical device in order to try out the front-end solution in all possible circumstances available.

Another suggestion was to check the iPad Adobe photoshop app to see how they allow the user to interact with picture editing tools.
Furthermore, it is important to keep the layer priority from black wound tissue to yellow wound tissue to red tissue and lastly, blue for unspecified tissue. This way the relevance of the tissue is prioritized.

OpenEHR & Vesalius Graduates: Mattijs Kuhlmann & Mathijs Hudepohl

In its current state of Vesalius, there is no UX-friendly front-end module created. M. Kuhlmann and M. Hudepohl, whom worked on this project created a Java-based front-end interface that utilizes the phone camera to make pictures which are analyzed by the ANN module locally and afterwards get stored on the back-end in a chaotic fashion. This implementation choice was made due to the fact that their assignment was to make sure that the data flow was put in place properly.

Furthermore, the focus of this assignment is mostly on the UX / UI aspects of the Vesalius module and to make sure that the pictures and assessments made by the user and the ANN are stored and processed correctly by the back-end / ANN.

<u>The main tasks are:</u>

- To do research into the state of the back-end and what is the best way to store professionally gathered user corrections.
- To do research into how it teaches the ANN to create models which can be used locally to determine the wounds.
- To do research into the best UX / UI flow so that the user can use the Vesalius component as effortlessly as possible.
- Create technical designs of how to store the corrected assessments.
- Create a front-end component (module) that warns the user that he makes sure that whatever feedback he gives to the module is made with intent and careful consideration.
- Facilitate correct storage and processing of the user feedback so that the ANN can learn correct and new wound patterns.

Data Science Student: Marlous Nijman

Ms. Nijman is currently studying Artificial Intelligence at the Radboud University. She Is doing a part-time internship (3 days a week) at the Nedap Data Science department, where she is going to re-work the learnability of the Vesalius ANN.

Since ms. Nijman and I work on the same solution but on a different scope, it is important to note that we have to remain in contact in order to make sure that both our adaptations fit together in the end.

For this reason, we have decided to meet weekly in order to keep each other up-to-date and also make agreements on what will have to be adapted from each other's end.

Furthermore, we discussed the current state of the ANN and the solution currently in place, and we have agreed that the data storage location will have to be somewhat reworked in order for the user-corrected data to be stored properly so that Vesalius can use the stored images to learn.

Ms. Nijman conducted an expert interview regarding the security of the Vesalius project and the privacy of the clients involved. After the privacy assessment made by the data privacy officer was conducted, ms. Nijman and mr. van Eenennaam discussed a solution that will remove the metadata of the images as well as hash the client ID so that the pictures can still be grouped by the person who they are of, but cannot be retraced to the actual client.

"Ik heb zojuist met Martijn van Eenennaam overlegd over de clientID.
Wat er nu gedaan wordt is dat we een lijst krijgen met clientIDs die we nodig hebben om de foto's te kunnen downloaden.
Voordat de foto's gedownload worden wordt er nu een one-way hash gebruikt om de clientID en unieke wond foto ID anoniem te maken.
De gedownloade foto's zullen dan als volgt heten: "clientIDhash_uniqueImageHash.jpg", waarbij beide hashes niet ter herleiden zijn naar de patiënt.
Verder is ook extra informatie van de foto's zoals de locatie waar deze is genomen verwijderd."

Ms. Nijman and I came to the agreement that the front-end application will have to send one mask for every color (red, yellow, black & white) alongside the original wound-image to the storage location for every wound assessed and validated by the user. This validated assessment will be stored as a file with the wound name preferably inside a file of the rehashed ClientID so that the owner of the wound is obfuscated but the wounds can still be organized according to the owner.

Nearing the end of our assignments, we have put a lot of effort together (peer programming) in order to execute a number of important changes to the Vesalius solution. Firstly, a new back-end and API were deployed containing the functionality necessary to retrieve the new model trained using multi-class classification. With the help of Mathijs Hudepohl, this back-end was deployed successfully and the new models were retrievable on the Vesalius mobile solution.

Data Privacy Officer: Michel Glintmeijer

Due to the fact that this technical solution will handle sensitive medical data of users, it is vital to make sure that the requirements regarding security are discussed and documented.

For this reason, mr. Glintmeijer, the privacy officer of Nedap N.V. went through the official "DPIA" procedure regarding the Vesalius ANN and its solution.

"Hierbij de samenvatting van de uitkomst van de quickcheck:
Het project brengt een heel klein risico met zich mee.
Dat risico is afhankelijk van het moment waarop het clientID wordt vervangen door een niet te herleiden ID.
Het moment waarop en hoe dat plaatsvindt wordt nog nader uitgezocht en teruggekoppeld.
Verder zijn er dermate maatregelen genomen dat het project zondermeer kan worden uitgevoerd, omdat de persoonsgegevens in de uitvoering niet meer aanwezig zijn bij de wondfoto's.
Deze uitkomst is bepaald op basis van de quickscan en daarom is ook besloten om geen DPIA uit te voeren."

Organizational Users: Various Healthcare Organisations

The users of the Ons Wondzorg mobile platform are the final and arguably most important stakeholders. These users will eventually decide whether or not they want to utilize this feature of the Ons Wondzorg platform and therefore need to have the ability to review the process and product during its development. Using the Discourse platform an amount of volunteer organizational users will be selected to review and test prototypes according to HMI heuristics so that the android component functionality and UX can be validated.

Land van Horne: Tamara van Gemert

The first organization that I got in contact with is named; Land van Horne. The manager, Tamara van Gemert, is a wound-care specialist who directs healthcare teams. These teams offer healthcare at home using the Ons platform for their healthcare documentation.

A main topic of discussion was the fact that the color of tissue is a great start for the ANN to determine, but does not neccessarely say everything about the type of wound because there are multiple types of red and yellow tissue. For the ANN to be truly accurate on a medical level, it has to determine tissue types instead of tissue colors. Furthermore, Tamara is very interested in testing out prototypes of the application and will gladly help test the UX / UI of the Vesalius component when a suitable prototype is produced.

Savant-Zorg: Marga Bosch

The second organization I met was Savant-Zorg were I spoke to ms. Bosch, besides the aforementioned aspects in the previous interview (4.6.1) it is important to distinguish fire induced wounds to normal wounds because they are more difficult to determine based on color alone.

Furthermore, ms. Bosch mentioned that the application should have a way to annotate the wound colors so that the user can add additional data that they discovered and link it to the color of the wound. Ms. Bosch was also very interested in participating in the UX / UI user survey and prototype validation.