Assistive Assistant

AI-driven Vision Impairment Assistant

Dominik Haider
K11920664

Pascal Pilz
K12111234

Table of Contents

Goal of the System

The goal of the system can be summarized using the initial project idea proposal:

Q1. What is the domain of your system?

Assistive technology.

Q2. Who would be the users of your system?

Visually impaired people, specifically colorblind people.

Q3. What is the main goal of your system?

Support impaired people in their daily lives by making interactions with colored objects easier. More concretely, the user takes an image and gives a color and a class name (e.g., “red apple”), and the program will communicate to the user where in the image the object is located.

Q4. How would your system achieve its goal?

By identifying (classifying) objects on supplied images via object detection using bounding boxes or image segmentation using object masks and then determining their dominant color.

Q5. Which type of AI/ML strategy would be required/useful and why?

Computer Vision is used to identify objects, and other techniques—also AI or, more simply, pixel sampling/averaging—are used to determine object color.

Requirements

Functional Requirements – what the system should do

[Business Requirements]

Req01: AA **shall** segment, classify and identify the colour of an object of interest.

Req02: AA **shall** integrate an auditive user-interface.

[UI/UX]

Req03: When the AA app opens **the** camera **shall** immediately turn on.

Req04: If auditive output is enabled **when** the camera is ready **the** AA **shall** speak to the user (e.g. "ready").

Req05: When the processing has finished**,** AA **shall** display the result (bounding box, label and colour[as text] of the object of interest) on the screen.

Req06: When the processing has finished**,** AA **shall** give an auditive signal, whereby a sound is played according to the status (success / failure).

Req07: If auditive output is enabled **when** the processing has finished**,** AA **shall** give an auditive description of the result.

[Reliability]

Req08: If the camera is active **when** the user single taps on the screen **then** AA **shall** initiate segmentation with the tap-location as area of primary interest.

Req09: When the status is 'success'**, the** AA **shall** display the results for at least 7 seconds, before the user can issue another request.

Req10: When the user starts the app**,** AA **shall** be ready within 3 seconds.

Req11: When the segmentation succeeded**,** object classification **shall** start.

Req12: When the object classification is finished**,** colour identification **shall** be initiated.

Req13: When segmentation found more than one segment**,** AA **shall** cancel the process and tell the user to re-try.

[Trustworthiness]

Req14: When the colour identification returns an inconclusive result**,** AA **shall** inform the user of this fact.

Req15: When segmentation succeeded**,** AA **shall** highlight the object's bounding box.

Req16: When the object classification has finished**,** AA **shall** display the label.

Req17: If live-mode is active **when** the result of the previous user-request has been shown for 10 seconds**,** AA **shall** reopen the camera.

Req18: If previous user-request has been processed **when** user double-taps on the device**,** AA **shall** reopen the camera.

Req19: If auditive control is enabled **when** the user says "colourise" **then** AA **shall** start the process.

Non-Functional Requirements – how the system should do it

[External Interfaces]

NfReq01: If vibration motors are present **when** the camera is ready **the** AA **shall** give a distinct haptic response ("haptic-standard-patterns").

NfReq02: If vibration motors are present **when** the processing has finished**,** AA **shall** give a haptic response to indicating the status (success / failure).

NfReq03: When the result is displayed on the screen **the** AA **shall** temporarily increase the screen's brightness for 5 seconds.

NfReq04: If the result is displayed **when** the user double taps on the screen **the** AA **shall** give an auditive description of the result.

NfReq05: When the processing has finished, AA **shall** initiate a vibration signal for 1 second.

NfReq06: The auditive description **shall** be well pronounced and easily understandable.

NfReq07: The UI/UX **shall** follow best practices of font legibility for visual impairment, e.g., use high contrast and easily recognizable typeface (e.g. Arial)

NfReq08: The AA **shall** provide a simple and clean user interface with sufficient large UI-Elements.

[Attributes]

NfReq09: The AA **shall** be free-to-use and open-source.

NfReq10: If a result has been produced **when** the user asks for explanation **the** AA **shall** be able to show the corresponding bounding box or segmentation mask.

NfReq11: The object detector of the AA **shall** be a pretrained model.

[Performance]

NfReq12: The AA **shall** be able to detect (segment, classify and colourise) most everyday items.

NfReq13: The AA **shall** have a success rate of at least 90%.

NfReq14: The entire process (from taking a photo until the result is ready) **shall** take less than 6 seconds.

NfReq15: The object classification **shall** take less than 3 seconds.

NfReq16: The colour identification **shall** take less than 2 seconds.

[Design Constraints]

NfReq17: The AA **shall** be able to function on a reasonably modern mobile device.

NfReq18: The AA **shall** be able to function locally, on-device.

NfReq19: The AA **shall** utilize “haptic-standard-patterns” for haptic responses.

NfReq20: The AA **shall** utilize “standard-sounds” for audio signals.

NfReq21: The AA **shall** only support processing on user-request, not continuously process live video feed.

Use Case Descriptions

The color of the heading indicates which use cases are implemented in the prototype:
green means it is implemented, **red** means it is not implemented.

Main Use Cases

UC 1: Segmentation Only

ID	UC1
Description	Segmentation of the area of interest
Actors	User, segmentation-model
Stakeholders:	Everyone within camera-view(GDPR), Developer
Pre-Conditions	camera access granted
Success end condition:	A single segment of the area of interest was retrieved
Failure end condition:	Segmentation of too low quality (e.g. more than one segment found)

Main Success Scenario		**Linked UCs**
1	User opens the app, app opens the camera and shows the user a live feed,	
2	The app informs the user that the app is ready, by speaking to him and a haptic prompt	SUC2, SUC4
3	User takes a photo, app shows that photo, user marks the area of interest on the photo	
4	The app feeds the photo and the area of interest to the image segmentation-model	
5	The segmentation-model segments the area, finds the most prominent object, returns that	
6	The augmentator overlays the photo with the segmentation result and displays it	SUC1
7	The app gives a 'success'-haptic response to the user as well as an according auditive one	SUC4, SUC2
8	The app blocks further user requests for at least 7 seconds, since status is 'success'	

Alternative Scenarios

4.A1	The segmentation-model found more than a single segment, in the marked area of interest	
4.A2	The app gives an 'error'-haptic feedback	SUC4
4.A2	The app cancels the process and prompts the user to issue the request again	

Exception Scenario

1.A1	The app has no access to the camera	
1.A2	The app has not started within 3 seconds	

UC 2: Classification (incl. Segmentation) Only

ID	UC2
Description	Classify object(s) in area of interest
Actors	User, classification-model, segmentation-model
Stakeholders:	Everyone within camera-view(GDPR), Developer
Pre-Conditions	camera access granted
Success end condition:	Objects were identified(=labelled), with sufficient certainty
Failure end condition:	Unidentifiable objects resp. threshold between classes too low

Main Success Scenario		**Linked UCs**
1	User opens the app, app opens the camera and shows the user a live feed to the user	
2	The app informs the user that the app is ready, by speaking to him and a haptic prompt	SUC2, SUC4
3	User takes a photo, app shows that photo, user marks the area of interest on the photo	
4	The app feeds the photo and the area of interest to the image segmentation-model	
5	The segmentation-model segments the area, finds the most prominent object, returns that	
6	The app takes the segmentation result and feeds it into the classification-model	
7	The classification-model classifies the object, returns the class	
8	The augmentator overlays the photo with the found segment and the class label and displays it	SUC1
9	The app gives a 'success'-haptic response to the user as well as an according auditive one	SUC4, SUC2
10	The app blocks further user requests for at least 7 seconds, since status is 'success'	

Alternative Scenarios

4.A1	The segmentation-model segmented the image, but there is not a single most prominent object	
4.A2	The app gives an 'error'-haptic feedback	SUC4
4.A3	The app cancels the process and prompts the user to issue the request again	

6.A1	The classification-model was unable to classify the object	
6.A2	The app overlayes the segmentation result over the photo without the class	

Exception Scenario

1.A1	The app has no access to the camera	
1.A2	The app has not started within 3 seconds	
6.A1	The object classification takes more than 3 seconds	

UC 3: Colour Identification per Auditive Control

ID	UC3
Description	Identify the colour (scheme) per auditive control(consequently emphasis on auditive output)
Actors	User, colour-identifier, classification-model, segmentation-model
Stakeholders:	Developer, Everyone within earshot, Everyone within camera-view(GDPR)
Pre-Conditions	camera access granted
Success end condition:	The colour (schema) is correctly identified
Failure end condition:	Inconclusive result, colour (scheme) not identified

Main Success Scenario **Linked UCs**

1	User opens the app, app opens the camera and shows the user a live feed to the user	
2	The app informs the user that the app is ready, by speaking to him and a haptic prompt	SUC2, SUC4
3	User issues an auditive command to the app to 'colourise' the most prominent items in the camera's view	SUC5
4	The app crops the photo by 1/3, so that the borders are thrown away and sets the area of interest to the center	
5	The app give the cropped photo and the area of interest to the image segmentation-model	
6	The segmentation-model segments the area, finds the most prominent object, returns that	
7	The app takes the segmentation result and feeds it into the classification-model	
8	The classification-model classifies the things in the found segment, returns the class	
9	The app feeds the retrieved segment into the colour-identifier	
10	The colour-identifier identifies the color of the segment and returns the found colors	
11	The system informs the user of successful processing, through a haptic feedback	SUC4
12	The app uses the speaker of the user's device to give an auditive description of the classes and colours	SUC2

Alternative Scenarios

5.A1	The segmentation-model was unable to find a segment of sufficient quality	
5.A2	The app gives an 'error'-haptic feedback, as well as an according auditive one	SUC4, SUC2
5.A3	The app informs the user to adjust the camera and issue a new request	

7.A1	The classification-model was unable to classify the object	
7.A2	The app simply refers in the auditive description to the coloured thing as 'items' instead of their classes	

9.A1	The colour-identifier was unable to identify the colour	
9.A2	The app simply tells the user that no colour could be identified	

Exception Scenario

1.A1	The app has no access to the camera	
2.A1	The user's device has no microphone	
3.A1	The user's device has no speaker	
7.A1	The object classification takes more than 3 seconds	
9.A1	The color identification takes more than 2 seconds	

UC 4: Colour Identification

ID	UC3
Description	Identify the colour (scheme)
Actors	User, colour-identifier, classification-model, segmentation-model
Stakeholders:	Developer Everyone within camera-view(GDPR)
Pre-Conditions	camera access granted
Success end condition:	The colour (schema) is correctly identified
Failure end condition:	Inconclusive result, colour (scheme) not identified

Main Success Scenario		**Linked UCs**
1	User opens the app, app opens the camera and shows the user a live feed to the user	
2	The app informs the user that the app is ready, by speaking to him and a haptic prompt	SUC2, SUC4
3	User takes a photo, app shows that photo, user marks the area of interest on the photo	
4	The app feeds the photo and the area of interest to the image segmentation-model	
5	The segmentation-model segments the area, finds the most prominent object, returns that	
6	The app takes the segmentation result and feeds it into the classification-model	
7	The classification-model classifies the object, returns the class	
8	The app feeds the retrieved segment into the colour-identifier	
9	The colour-identifier identifies the color of the segmented object, returns the colour	
10	The augmentator augments the photo with the segment, the class label and the colour name	SUC1
11	The app gives a 'success'-haptic response to the user as well as an according auditive one	SUC4, SUC2
12	The app blocks further user requests for at least 7 seconds, since status is 'success'	

Alternative Scenarios

3.A1	The segmentation-model segmented the image, but there is not a single most prominent object	
3.A1	The app informs the user that image was invalid and asks the user to take a new photo	

6.A1	The classification-model was unable to classify the object	
6.A2	The app overlayes the segmentation result over the photo without the class	

8.A1	The colour-identifier was unable to identify the colour	
8.A2	The app overlayes the seg. result over the photo with the class but wihout naming the colour	

Exception Scenario

1.A1	The app has no access to the camera	
7.A1	The object classification takes more than 3 seconds	
9.A1	The color identification takes more than 2 seconds	

UC 5: Live-Feed

ID	UC5
Description	Continourly annotate Live-Feed as fast as possible
Actors	User, segmentation-model, classification-model, colour-identifier
Stakeholders:	Every natural person(GDPR), Developer
Pre-Conditions	-
Success end condition:	-
Failure end condition:	-

Main Success Scenario **Linked UCs**

1	User opens the app, app opens the camera and shows the user a live feed to the user	
2	The app takes a photo, crops it by 1/3 and scales it down to 1/2	
3	The app feeds this photo into the segmentation model	
4	The segmentation-model segments the area, finds the most prominent object, returns that	
5a	The app displays the found segment to the user(bounding-box)	SUC1
5b	Meanwhile, the app feeds the segment into the classification model	
5c	Simultaneously, the app also feeds the segment into the color identifier	
6	The classification-model classifies the object, returns the class	
7	The colour-identifier identifies the color of the segmented object, returns the colour	
8	The app displays the label of the found class	SUC1
9	The app displays the name of the found colour	SUC1

Alternative Scenarios

3.A1	The segmentation-model was unable to find a segment of sufficient quality	
3.A2	The app does not display a bounding-box, just an unaugmented live-feed	
5b.A1	The classification-model was unable to classify the object	
5b.A2	The app displays 'unknown' as class label	

5c.A1	The colour-identifier was unable to identify the colour	
5c.A2	The app does show '?' as colour name	

Exception Scenario

1.A1	The app has no access to the camera	

Supporting Use Cases

SUC 1: Augmentate

ID	SUC1
Description	Augment visual input, to inform user of processing steps
Actors	Augmentator(= non-ai subsystem)
Stakeholders:	User
Pre-Conditions	Given are the information to show and a photograph resp. a video-frame
Success end condition:	
Failure end condition:	

Main Success Scenario		**Linked UCs**
1	The augmentator extends the image with the information given	
2	The augmentator returns this augmented image	

Alternative Scenarios

Exception Scenario

1.A1	Provided information is stale(may happen in live-feed conditions)	
1.A2	The system cancels the augmentation and prompts user to issue a new request	

SUC 2: Auditive Output

ID	SUC2
Description	Information is given to the user in an auditive way
Actors	Text-to-Speech module, User
Stakeholders:	OS, Everyone within earshot
Pre-Conditions	Option 'auditive output' is enabled and speaker access granted
Success end condition:	-
Failure end condition:	Output device was not ready or no audio-data to play

Main Success Scenario		**Linked UCs**
1	The system prepares the output device(try to get handle)	
2	The Text-to-Speech module turns text into natural-sounding speech	SUC3
3	The system plays generated audio to user	

Alternative Scenarios

Exception Scenario

1.A1	The system is no able to obtain control of speaker(e.g. used by another app)	
1.A2	The system cancels the speech output	

1.B1	The OS reports low volume to the system	
1.B2	The system temporarily increases volumne, to make sure user clearly hears what is being said	

2.A1	The Text-to-Speech module returns an empty audio	
2.A2	The system skips the playback	

SUC 3: Text to Speech

ID	SUC3
Description	Turn text into natural-sounding speech
Actors	Text-to-Speech module
Stakeholders:	User, OS, User's device
Pre-Conditions	-
Success end condition:	Generated speech is returned
Failure end condition:	Generation of speech was unsuccessful

Main Success Scenario **Linked UCs**

1	The Text-to-Speech module generates speech from supplied text	
2	The Text-to-Speech module evaluates generated speech	
3	The system returns the audio	

Alternative Scenarios

Exception Scenario

1.A1	The Text-to-Speech module takes more than 5 seconds to generate speech	
1.A2	The system returns empty audio(=NULL)	

2.A1	Quality of the generated audio is considered to be insufficient, by the Text-to-Speech module	
2.A2	The system returns empty audio(=NULL)	

SUC 4: Haptic Feedback

ID	SUC4
Description	Provides feedback to the user in a haptic way
Actors	Hapticator(= non-ai subsystem)
Stakeholders:	OS, User, User's device(vibration motors)
Pre-Conditions	Option 'haptic output' is enabled
Success end condition:	-
Failure end condition:	-

Main Success Scenario **Linked UCs**

1	The Hapticator gets handle to vibration motors (=making them ready) from OS	
2	The Hapticator sends situation-dependent haptic effect to vibration motors	

Alternative Scenarios

Exception Scenario

1.A1	The Hapticator is unable to get access to vibration motors	
1.A2	The system remains unaffected by this and continuous with its routine	

1.B1	Intended (complex) effect (e.g. haptic-level 'rich haptics') not supported by user's device	
1.B2	Hapticator must fallback to 'clear haptics'	

SUC 5: Auditive Control

ID	SUC5
Description	Enables the user to control the app per spoken commands
Actors	User, Speech-to-Text module
Stakeholders:	OS, Everyone within earshot, User's device
Pre-Conditions	Option 'auditive control' is enabled and microphone access granted
Success end condition:	Command returned to the system
Failure end condition:	Input device was not ready

Main Success Scenario		**Linked UCs**
1	The system prepares the input device(try to get handle)	
2	User talks to the device(record audio)	
3	The Speech-to-Text module turns speech into text	SUC6
4	The system lookups text for registered commands	
5	Found command is enqueued by the system	

Alternative Scenarios

Exception Scenario

1.A1	The microphone can not be obtained by the Speech-to-Text module	
1.A2	The system prompts the user of this fact, and cancels	
3.A1	The Speech-to-Text module was unsuccessful in transcribing text	
3.A2	The system prompts the user to repeat the voice input	

4.A1	The system could not identify a valid command	
4.A2	The system prompts the user to repeat the voice input	

SUC 6: Speech to Text

ID	SUC6
Description	Transcribe text from speech
Actors	Speech-to-Text module
Stakeholders:	User, OS, User's device
Pre-Conditions	-
Success end condition:	Transcribed text is returned
Failure end condition:	Text transcription was unsuccessful

Main Success Scenario		**Linked UCs**
1	The Speech-to-Text module transcribes the text	
2	The system returns the transcribed text	

Alternative Scenarios

Exception Scenario

1.A1	Speech-to-Text module takes more than 6 seconds to transcribe text	
1.A2	The system returns an empty text	

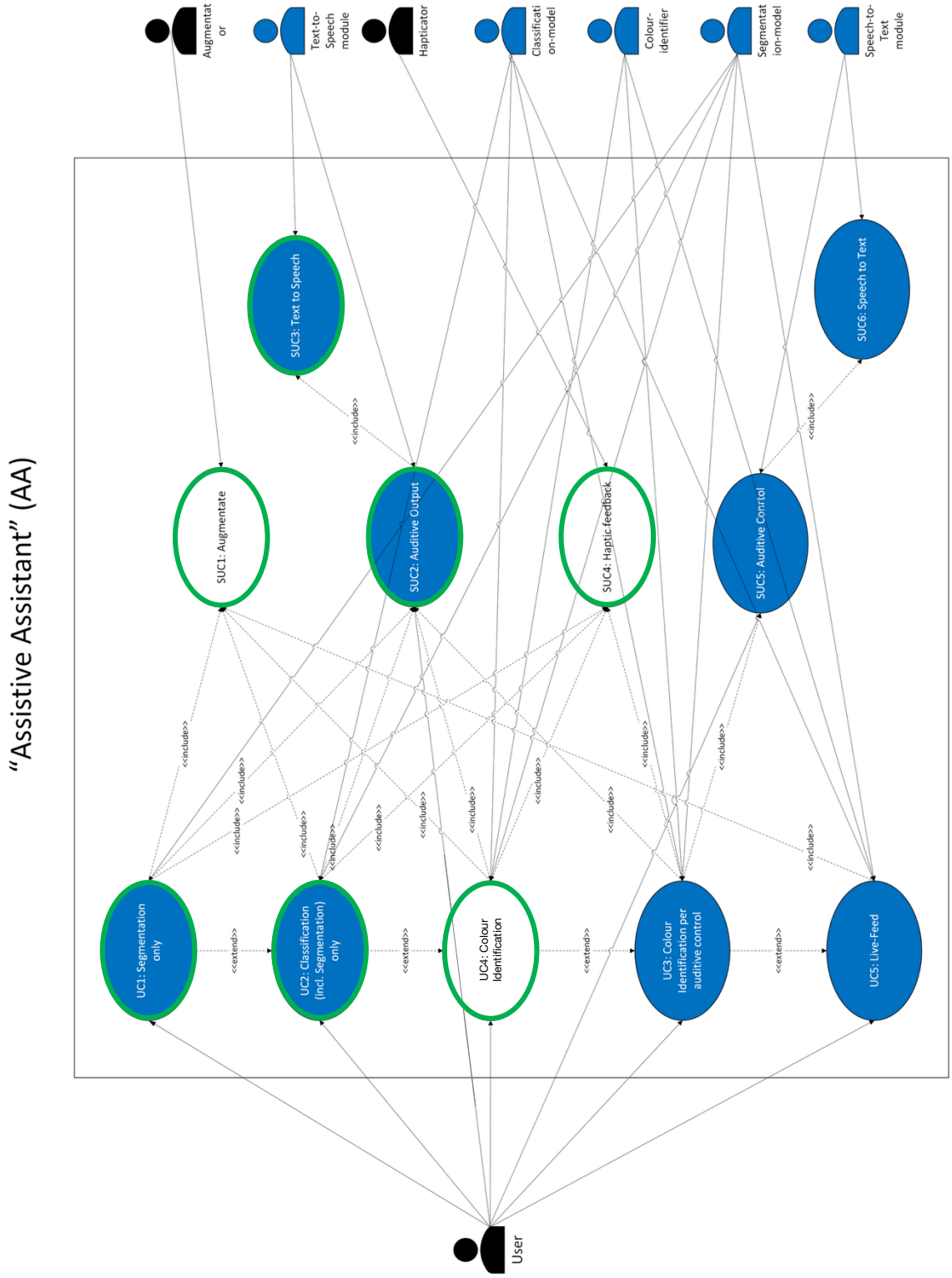
Traceability matrix

	UC1	UC2	UC3	UC4	UC5	SUC1	SUC2	SUC3	SUC4	SUC5	SUC6
Req1	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req2	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req3	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req4	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
Req5	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
Req6	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req7	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
Req8	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req9	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req10	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req11	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req12	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req13	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req14	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req15	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
Req16	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
Req17	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req18	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
Req19	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
NfReq1	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
NfReq2	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
NfReq3	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq4	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq5	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
NfReq6	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
NfReq7	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq8	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq9	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
NfReq10	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq11	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq12	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq13	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq14	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq15	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq16	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq17	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq18	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
NfReq19	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
NfReq20	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
NfReq21	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

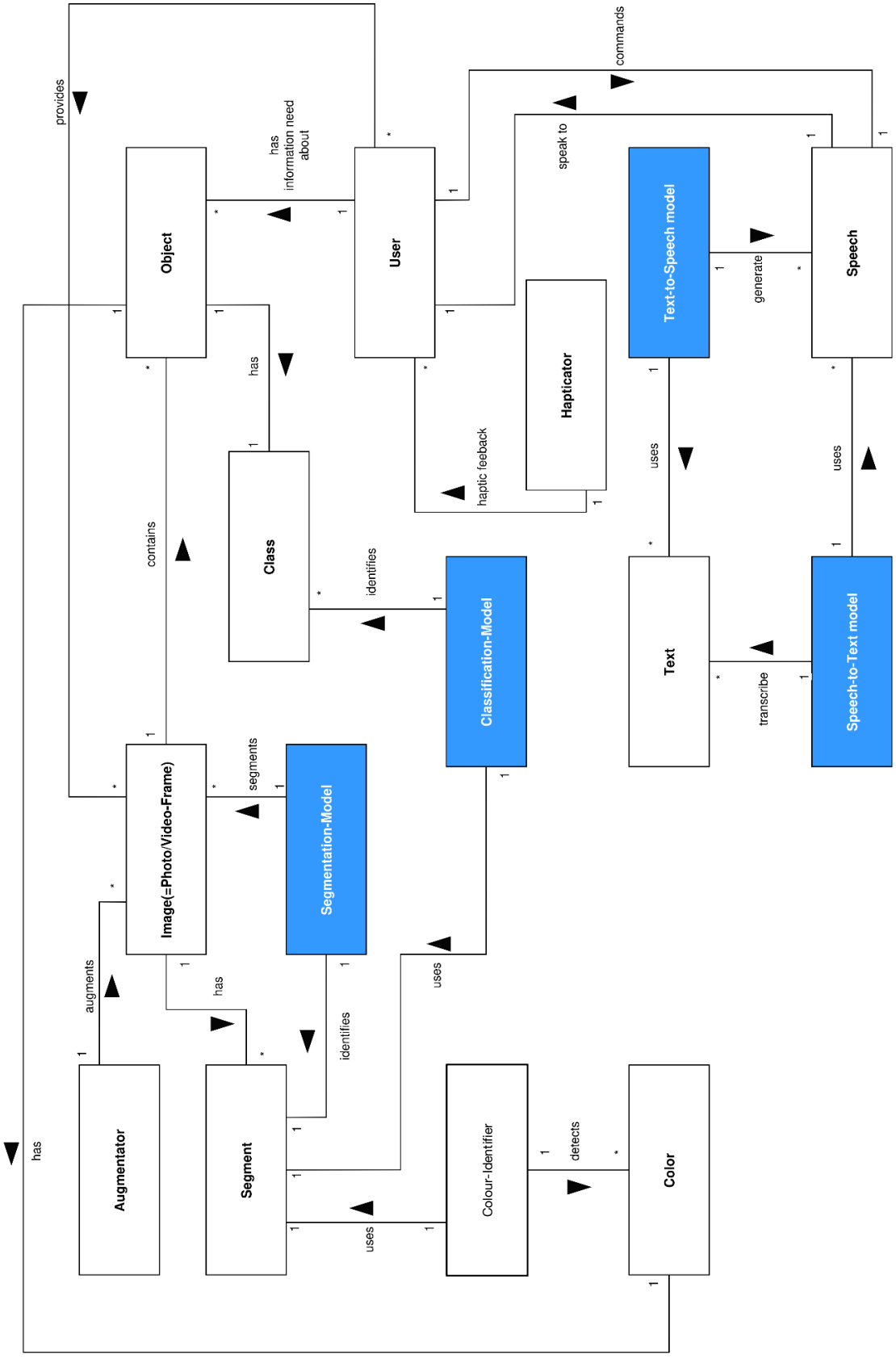
Diagrams

Use Case Diagram

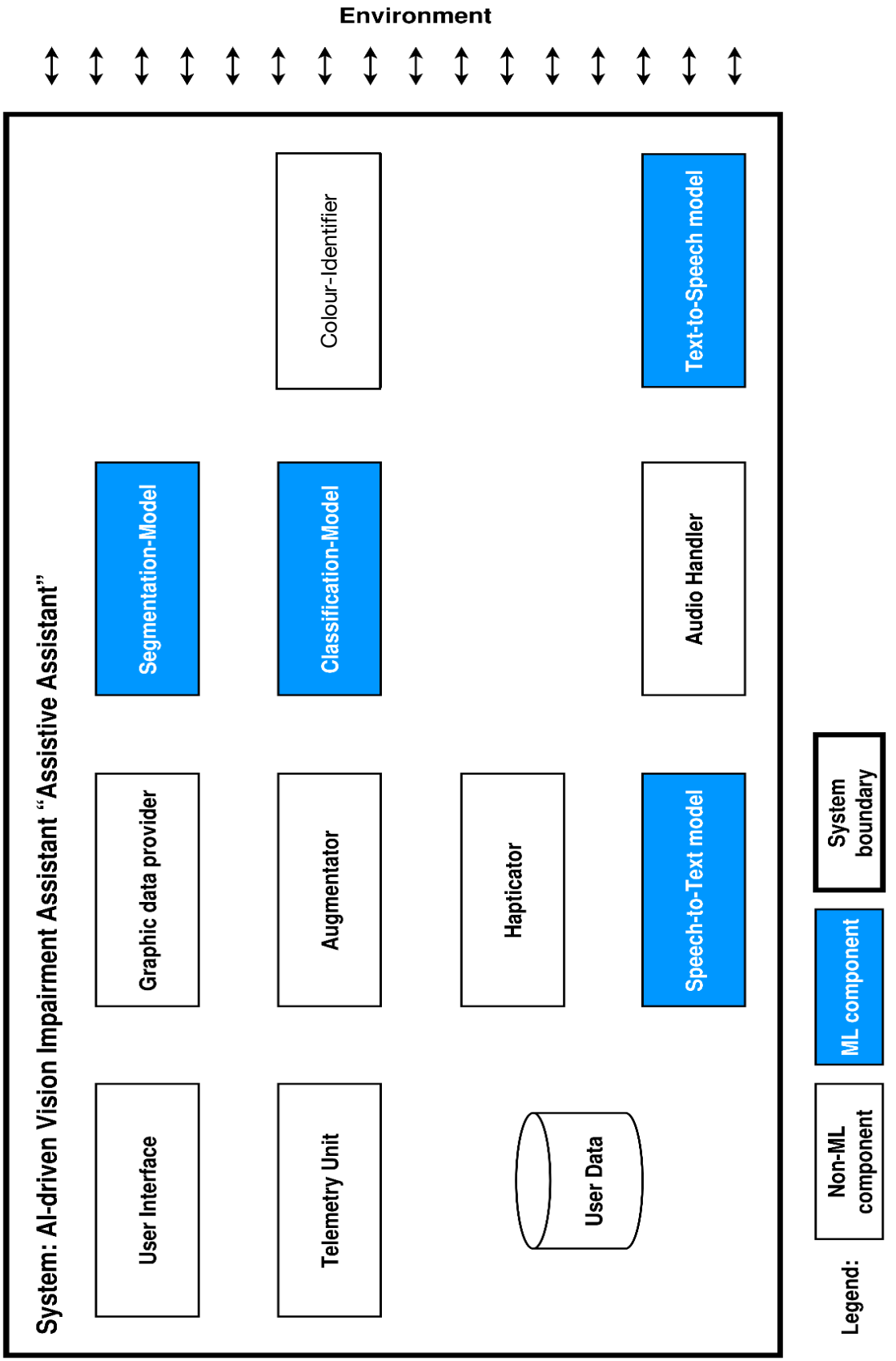
Again, a **green** outline indicates that this use case was implemented in the prototype.



Domain Model



Architecture Diagram



Components Description

User Interface

Main User Interaction Component, it provides the user with a graphical user interface as well as an auditory one. The latter
Req02, Req03, Req08, Req09, Req10, Req17, Req18
NfReq04, NfReq07, NfReq08, NfReq10

Telemetry Unit

Observes each submodule's respective subsystem.
For AI-related components, it is additionally responsible for error recovery as well as making sure the AI models comply with set time demands.
Req13, Req14
NfReq14, NfReq15, NfReq16

Graphic data provider

This component provides access to the camera on the smartphone as well as to the file system. The latter is needed to use the app on saved images.
Further, it provides the raw video feed, which can either be directly consumed via the live-feed functionality, or by taking a specific frame (= photograph)
Req03, Req08

Audio Handler

Handles the auditory communication, hence input and output.
Precisely, the access to the microphone and speaker, the playback, and the recording of the user's spoken command.
Req02, Req04, Req06, Req07, Req14, Req19
NfReq04, NfReq20

Augmentator

Responsible for communicating the results in a visual way.
For instance, this subsystem enriches the graphic input data (photograph/frame) with results (bounding box, class, colour) from the AI models.
Req05, Req14, Req15, Req16
NfReq03, NfReq10

Hapticator

The Hapticator is concerned with providing haptic feedback to the user.
Hence, it implements the “haptic-standard-patterns" and the OS calls to access the vibration motors of the smartphone.
Req14
NfReq01, NfReq02, NfReq05, NfReq19

User Data

Handles application-specific settings as well as facilitates a browsable history of previously successful processed photographs. Live-feed sessions are not recorded.
Req04, Req07, Req19

Segmentation-Model

Offline-Model, which segments the area of interest, which is either provided by the user or estimated by the application (see also Assignment3::UsesCases)
Req11
NfReq11, NfReq12, NfReq13, NfReq15, NfReq18

Classification-Model

Offline-Model, which provides a reliable classification of everyday items
Req11, Req12
NfReq11, NfReq12, NfReq13, NfReq15, NfReq18

Colour-Identifier

An offline module that identifies colour (schemes).
Req12, Req14
NfReq12, NfReq13, NfReq16, NfReq18

Text-to-Speech model

State-of-the-art model, which generates a well-pronounced and easily understandable speech from the results(text).
Req04, Req07
NfReq06

Speech-to-Text model

Enables auditory input by transforming the user’s spoken command to text.
Req02, Req19

Design Questions

1. How can we ensure that the largest possible number of **devices can run the AA**?
    - Running the models locally -> not dependent on internet connection
    - Small models made for mobile computing -> most devices can run it
    - Down-sampling to 224x224 resolution -> compatibility with most cameras, lower compute cost

2. Does the **AA run locally** or via a cloud service?
    - The AA runs locally to ensure maximum reliability. For this to work, we need to use a very efficient segmentation model.

3. What **types of objects** do we support?
    - The current objective is to support objects covered in common image datasets, such as CIFAR-100, ImageNet, MS COCO. This means most everyday items. (For the sake of prototyping, we might have to restrict ourselves to, for example, just fruits.)

4. Do we **collect user/usage data**? If so, is there a chance it could **leak**?
    - We do not collect any data. Yet, a history is provided that is stored on the user's device. The app is designed as an "offline" one.

5. What design elements can we use in the UI to **make the AA accessible** to visually impaired people?
    - Easily recognizable typefaces (e.g. Arial), simple UI-elements of sufficient size, high contrast, adequate brightness, and usage of colours which are not easily confused by the various kinds of colour-blindness.

6. Should we **allow the users to customize** their experience in terms of accessibility features?
    - The users should be able to customize their experience. This is important since not every person suffers from the same impairments. (For the sake of prototyping, we will omit extensive customization.)

7. How does the **AA interact with other systems** (OS, ...)? Does that require future maintenance?
    - Since OS-calls are used, backward compatibility as well as access rights may be an issue, due to the rapid development cycle of mobile operating systems. Hence, to guarantee the functionality in the long term, at least a yearly update will be necessary.

8. What **metric** (accuracy, precision, recall, F1, ...) is important when it comes to **object detection and classification** with regards to the AA?
    - This depends on how we weigh different kinds of errors, *False Positives* vs *False Negatives*. In our application, it seems that *False Negatives* should be avoided, since

it could lead the user to assume facts about their environment that aren't true. E.g., imagine a fruit that is poisonous if not ripe, and the could be detected by colour. Therefore, we choose to use precision (aka positive predictive value), since it captures how many predictions were True Positives out of all positive predictions:
precision = true positive / (true positive + false positive)

9. Do we want to use a **certain type of AI model** (architecture, developer)?
    - We use a model suitable for mobile computing. Furthermore, we use a model that is openly available and is freely usable for our purposes.

10. Do we need to **update the AI models**?
    - Once a system with satisfying performance is found, it doesn't need to be updated. We don't exclude possible future updates, but maximum compatibility must be ensured.

11. Do we use an **AI technique or a simpler approach** for the **Colour-Identifier**?
    - This simple task can be done by simple methods, with no need for a CNN or similarly computationally costly techniques.

12. Are there **safety risks** associated with the use of the AA?
    - Misclassification of certain items could be dangerous, for example, imagine a fruit that is poisonous when unripe, and its ripeness can be determined by its colour; therefore, a misclassification of our system could lead to direct harm.
    - Other lesser risks can include misclassification of items and, therefore, inconvenience, for example, a sweet potato classified as a regular potato.

13. Do we include a **confidence score** when showing our results to the user?
    - This depends on the AI models we use. Not all AI models are able to accurately communicate their confidence. The values given by a neural network are often interpreted as probabilities, but they do not reflect real-world probabilities. Depending on the system, a confidence score can be very misleading; therefore, I suggest that we omit such a score for now.