An Interactive Data Collection and Labeling Web-Tool with Meta-Learning

Requirements Specification

Tarek Gaddour
Atalay Donat
Ömer Erdinç Yagmurlu
Mustafa Enes Batur
Murat Kurnaz

December 3, 2020

Contents

1 Product Overview **4**

2 Purpose **5**
2.1 Core criteria 5
2.2 Optional criteria 6
2.3 Exclusion criteria 6

3 Usage **7**
3.1 Area of Application 7
3.2 **Use Case Examples** 7
3.3 Target Groups 8
3.4 Operating Conditions 8

4 Operating Environment **9**
4.1 Software 9
4.1.1 Server Appliance 9
4.1.2 Client Devices 9
4.2 Hardware 9
4.2.1 Server Appliance 9
4.2.2 Client Devices 9

5 Functional Requirements **10**
5.1 Main Functions 10
5.1.1 Web Client 10
5.1.2 Server API 12
5.1.3 Data Processing 12
5.2 Extending Functions 12

6 Data **13**

7 Nonfunctional Requirements **14**
7.1 Usability 14
7.2 Swiftness 14
7.3 Maintainability 14
7.4 Portability 14
7.5 Security 14
7.6 Scalability 15

8 System Models **16**
8.1 Desktop System Model 16
8.2 Mobile System Model 17

9 User Interface **18**
9.1 Desktop . . . 18
9.2 Mobile . . . 18
9.3 Screenshots . . . 19

10 Quality Assessment Criteria **24**

11 Test Cases and Test Scenarios **25**
11.1 Test Cases . . . 25
11.1.1 Main Test Cases . . . 25
11.1.2 Extending Test Cases . . . 26
11.2 Test Scenarios . . . 26

12 Development Environment **27**
12.1 Operating Systems . . . 27
12.2 IDEs, Editors . . . 27
12.3 Versioning . . . 27
12.4 Integration . . . 27
12.5 Miscallaneous . . . 27

1 Product Overview

An Interactive Data Collection and Labeling Web-Tool with Meta-Learning is a web-tool enabling its users to create, edit and manage machine learning models on different sensor data. An Interactive Data Collection and Labeling Web-Tool with Meta-Learning consists of two primary components:

1. A server application that handles model training and storage, and makes these models available through a REST API. Supports multiple models and multiple users.

2. Desktop and mobile client applications to manage and interact with the model, e.g. adding new data points, identifying given data points and setting parameters for model training. A model can be utilized on different clients at the same time. Clients have different capabilites depending on the platform, mobile clients allow input of new data points in form of raw sensor data, while desktop clients allow management features.

Both components together allow for a streamlined and easy-to-use machine learning experience for both the unexperienced and the expert.

2 Purpose

2.1 Core criteria

- The web pages are the graphical user interfaces (GUI) for users.
- Users register a new account or login with their existing accounts to use the tool. The users are presented with their own set of workspaces after logging in or a prompt to create a new workspace if they do not have one.
- When creating a new workspace, the user chooses a name for their new workspace and the sensors that they wish to use when recording samples and classifying actions.
- The desktop client allows creating new labels and renaming the existing ones for the actions to be recorded in the workspace. It also shows the amount of recorded samples with the label.
- The desktop client presents a QR code specific to the selected workspace. The code connects the mobile client to the same workspace when scanned.
- The mobile web client allows recording of data samples in form of raw sensor data. The user chooses the label for which they wish to record. The recording parameters like sampling rate, countdown and recording duration can be configured.
- The mobile web client gives a real time feedback of sensor data to the user during the recording period.
- Recorded data samples are stored in a database and displayed chronologically on the desktop web client. The user can view the metadata of the sample by clicking on it. The user can also view the data as a time graph on which he can label the relevant time frames. The user can remove or relabel samples.
- The desktop web client presents the user options/features to process the data set in the workspace. The results are stored in turn and are available to the user.
- Trained and available models are listed on the desktop web client. The user can view the metadata of each model, e.g. parameters and data samples used for the training or performance metrics by clicking on the model.
- Each model is assigned a link which the user can use on a mobile device to classify action with the respective model. This identification happens in real time. The

user can view the link as well as the QR code with the same link embedded to ease usage.

2.2 Optional criteria

- The desktop web client serves a "Stay Signed In" functionality.
- Other data capturing devices are supported, e.g. Arduino.
- The desktop web client displays a status sign if a data collecting device is currently connected to the workspace, e.g. a green sign if connected and a red sign otherwise.
- The mobile web client can define triggers if something is detected, e.g. play a sound.
- Workspaces can be deleted together with the related data samples.
- Data samples can be transferred between compatible workspaces.

2.3 Exclusion criteria

- The mobile web client does not have a QR scanner.
- The user can not change the sensors used in a workspace.
- Each web page is designed for either a desktop device or a mobile device. Trying to display a web page in an unsupported device has undefined behavior.

3 Usage

3.1 Area of Application

The application is for collecting and labeling sensor data, training a model from the collected data and serving the stored resulting model to the user to classify actions in real time.

3.2 Use Case Examples

Frank has sparse knowledge of machine learning. He just discovered “Explorer”. On the website that he views using his laptop, he creates a new account. After logging in, he creates a new workspace by selecting sensors to sample data (e.g., accelerometer). He sets up some simple labels (e.g. swipe right, swipe left). After opening the QR code of the workspace, he can select the action to perform on his phone from the available labels. He can also set the duration for a recorded sample and a countdown that is shown before recording. He performs some simple swipe gestures by moving his phone mid-air. The recorded data is labeled and is pushed to the laptop where he can see the data coming in. With a single click he can build a machine learning model that is immediately available on his smartphone to classify gestures.

Alice is an engineer at a washing machine company. Alice has been observing that HCSOB washing machines with clogged circulating pumps show an unusual pattern of movement during the washing process (error reference 404). The washing machine is moving in a specific rhythm, which Alice recognizes when she is at the customer’s home. However, Alice would rather make a diagnosis without having to go to the customer. Then she remembers the program "Explorer" which she can use to easily develop machine learning models. With the help of the smartphone acceleration sensors in her cell phone, she and her colleagues record the movement patterns at some of her repair sites. After she has collected enough data, she can use the Explorer program to automatically train and deploy a machine learning model with one click. When a new case comes in, Alice just sends the customer a URL to a website of the Explorer. The customer places his smartphone on the washing machine and the model can determine directly "on the edge" whether error 404 is present. This saves Alice money and time. Alice can order the parts for repair and perform the repair with a single visit.

3.3 Target Groups

The target group of this application is people that wants to create a machine learning model from mobile sensor data. That includes:

- Inexperienced people (e.g. Frank in the first use case), who are new to machine learning and want an easy tool to discover machine learning and create models that they can use.

- Experienced engineers (e.g Alice in the second use case), that have knowledge of machine learning but need a fast tool to collect data and create a model.

We assume a rather technical audience who has basic knowledge of how to use a web browser in a desktop and mobile client. **(Isn't this useless?)**

3.4 Operating Conditions

The tool is a web application that can be run on any modern browser. A web browser must be installed on the user device to view the webpage. The application can be used from anywhere with a decent network connection.

Service Duration: 24 hours a day

4 Operating Environment

Database and the server appliance both run on a server, whereas different web browsers may be used for the clients. A Docker image may be provided for the server appliance.

4.1 Software

4.1.1 Server Appliance

- MongoDB support
- Recent versions of Node.js and Python

4.1.2 Client Devices

- A modern web browser (Chrome 87+, Firefox 83+) supporting:
  - Sensor APIs
  - Web Storage API

4.2 Hardware

4.2.1 Server Appliance

The server has to be fast enough to support all clients. This depends on the expected number of clients. Most computations will be done on the server. **TOO ARBITRARY?**

4.2.2 Client Devices

- **TBD**
- **TBD**

5 Functional Requirements

5.1 Main Functions

These functions must be implemented in order to fulfill the core criteria.

5.1.1 Web Client

The web client supports both desktop and mobile modes. The functionality that will be displayed is determined by the device information of the browser.

Desktop Web Client

/**F010**/ Show a welcome page with the registration and login panel

/**F020**/ List workspaces of the logged in user

/**F030**/ Allow creating workspaces given the name and sensors to be used

/**F040**/ List the available sensors for the recordings

/**F050**/ Allow renaming workspaces

/**F060**/ Show a workspace panel when a workspace is selected

/**F070**/ Allow creating labels for the actions to be recorded on the workspace

/**F080**/ List labels with their sample count

/**F090**/ Allow renaming labels

/**F100**/ Allow deleting labels which in turn deletes the data samples with the selected label

/**F110**/ Display a link and the QR code with the same link embedded to be used for recording data

/**F120**/ Display the collected data samples chronologically on the workspace panel

/**F130**/ Display the metadata recording and allow selecting the relevant timeframes of the sample

/**F140**/ Allow relabeling and deleting samples

/**F150**/ List the trained models

/**F160**/ Display the used parameters and data samples of the selected model

/**F170**/ Display the performance metrics of the selected model

/**F180**/ Allow selecting the possible options on the data on the management panel

/**F190**/ Request the processing of the data according to the selected options

Mobile Web Client

/**F200**/ Show a configuration page with available labels

/**F210**/ Allow configuring the countdown duration until the recording starts

/**F220**/ Allow configuring the recording duration

/**F230**/ Allow configuring the sampling rate of the sensors

/**F240**/ Show a button to initiate the recording

/**F250**/ Show a countdown page

/**F260**/ Display the current configuration on the countdown page

/**F270**/ Show a recording page

/**F280**/ Display the sensor data in real-time as curve graphs

/**F290**/ Show a recording completed page

/**F300**/ Allow discarding the last recording

/**F310**/ Send the sensor data to the server

/**F320**/ Allow another recording with the same configurations

/**F330**/ Allow editing configurations for the next recording

5.1.2 Server API

/**F340**/ Serve authentication services

/**F350**/ Serve workspace information for a user

/**F360**/ Serve data set and recording link for a workspace

/**F370**/ Create, rename and delete labels on a workspace

/**F380**/ Serve label information to the mobile client

/**F390**/ Accept data from the mobile client

/**F400**/ Initiate the configured model training

/**F410**/ Serve the model information on a workspace

/**F420**/ Rename and delete models on a workspace

5.1.3 Data Processing

/**F430**/ **TBD, after workshop**

5.2 Extending Functions

/**F440**/ Allow listing samples by label on the workspace panel

6 Data

1. Source code
2. Documentation
3. Login information
4. Workspace information
5. Webpage HTML files
6. Webpage style sheets
7. Webpage texture files
8. Webpage sound files
9. Data samples
10. Trained models

7 Nonfunctional Requirements

7.1 Usability

- /N010/ User does not need to register to use the web-tool.
- /N020/ User should be able to record a new sample in less then 3 interactions.
- /N030/ User should be able to train and deploy the machine learning model (/FXXX/) with a single click.

7.2 Swiftness

- **lazy load components?**
- /N040/ Sth about page load speed - might be too restrictive, cannot assess yet.

7.3 Maintainability

- /N050/ Codebase is well documented to ease bug fixes.

7.4 Portability

- /N060/ Desktop web client /FXXX/ should be functioning on browsers newer than Chrome version XX, Firefox version XX, ...
- /N061/ Mobile web client /FXXX/ should be functioning on browsers newer than Chrome version XX, Firefox version XX. ...

7.5 Security

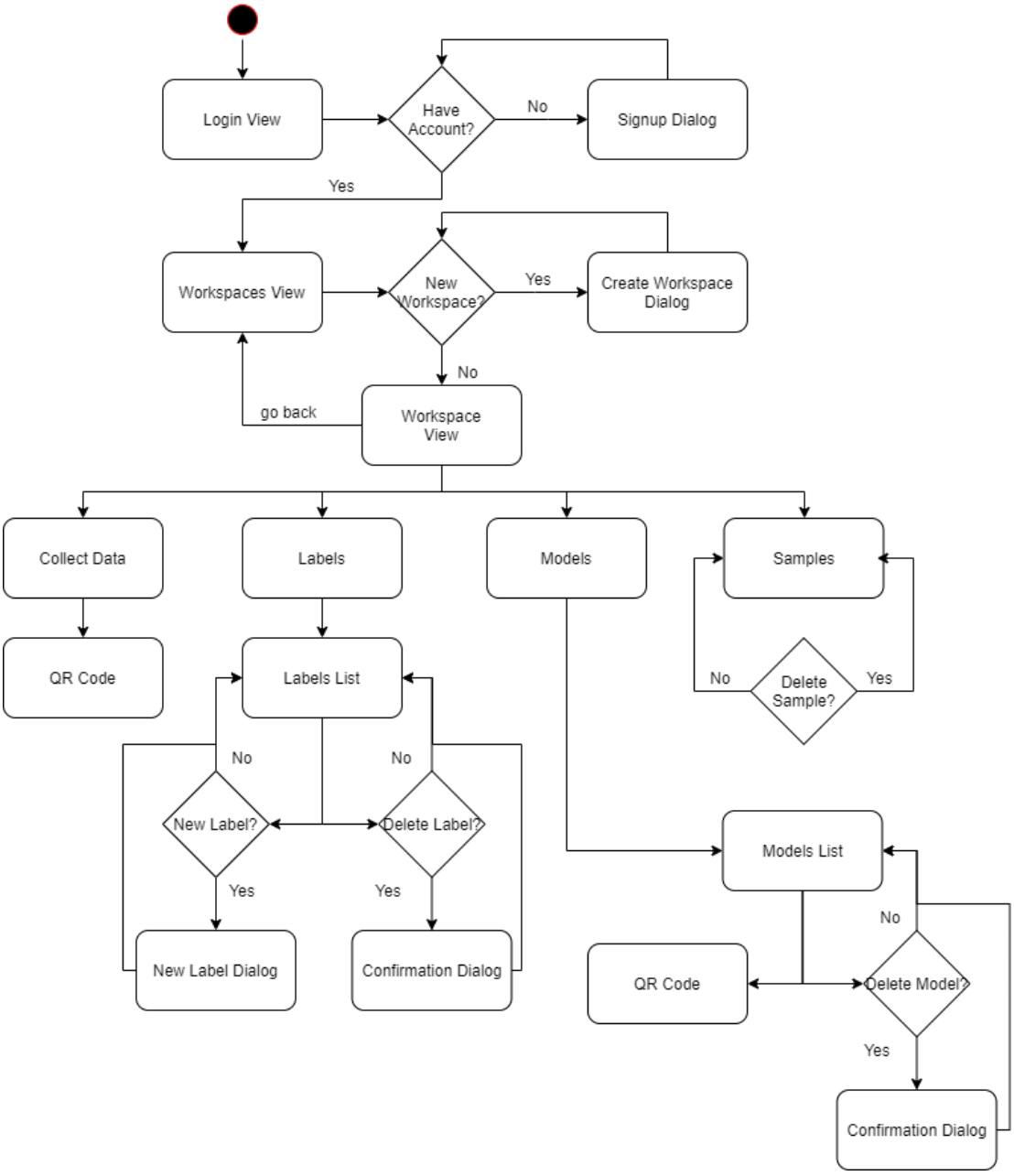
- /N070/ A workspace is inaccessible to other users without the QR code or the link to the workspace.

7.6 Scalability

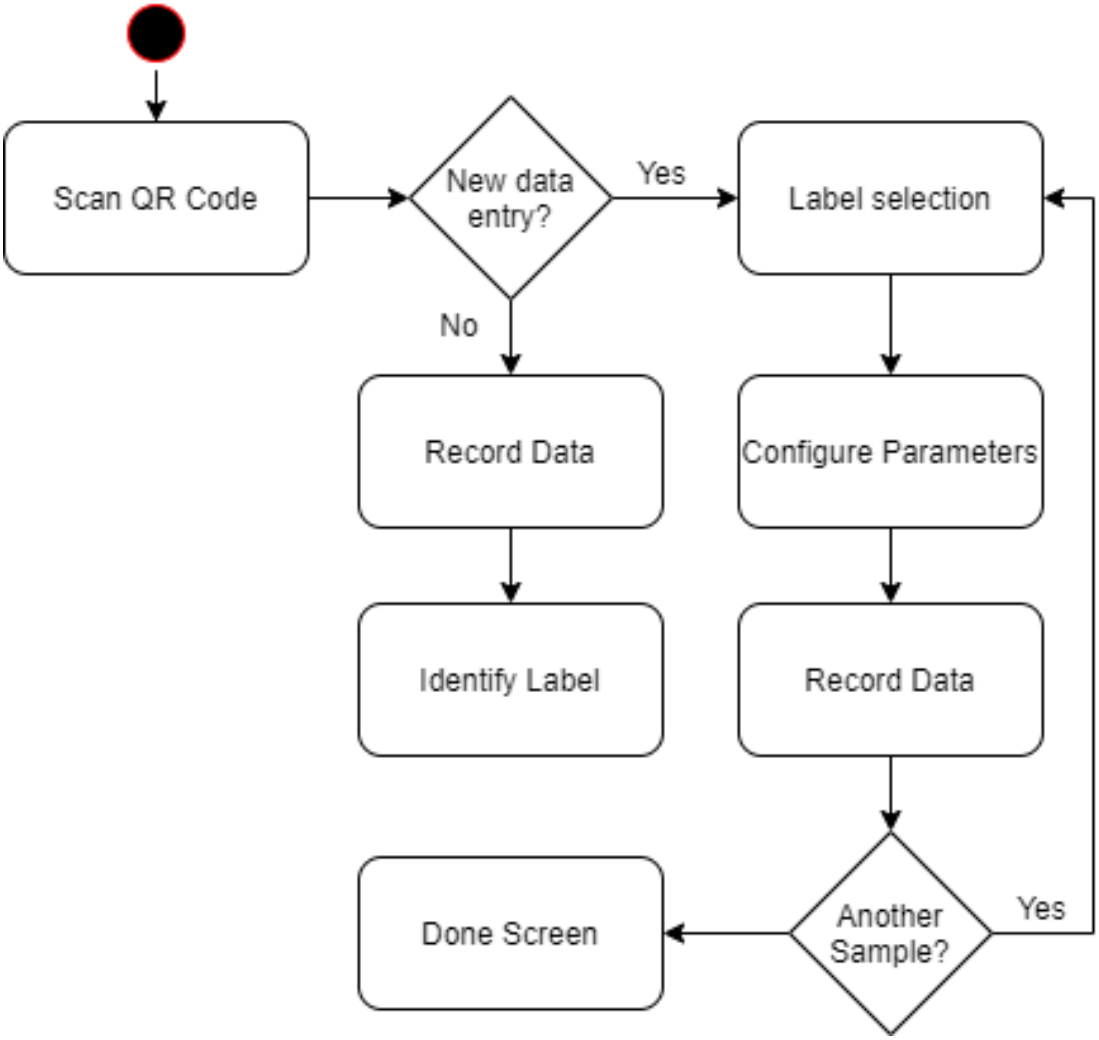
- /N090/ A maximum of 10.000 ?? workspaces with a maximum of 10 ?? users at the same time is supported.

8 System Models

8.1 Desktop System Model



8.2 Mobile System Model



9 User Interface

The user interface of both the desktop and mobile client will be designed so that a user without a lot of technical knowledge of machine learning can easily make use of this web-tool. The user interface is also great for experienced engineers as it presents a simple and fast way to create models and start classifying data right away.

9.1 Desktop

The users are first met with a simple login/registration panel. Then the current workspaces are listed, from which the users can choose to work with. The users can also create a new workspace, in which they name it and choose the labels it will have. In the workspace overview page, the collected data samples are listed on the left, and on the right there are buttons to view the labels and models. The users can also easily select hyperparameters and train and deploy a model in this page with a single click.A link and a QR code, which a mobile web client can use to start collecting data to this workspace, will also be available with a single click. Users can also see the data sample overview, labels overview and the models overview. Each model will provide its own link and QR code, which a mobile client can use to start classifying actions with this mode and which again will be provided with a single click.

9.2 Mobile

When a QR Code is scanned or a link is visited, the users are met with a list of labels to choose from. When a label is selected, users are allowed to configure the recording parameters before pressing the button to start recording. After that, the countdown runs and then the recording page is displayed. In the end, it is stated that the recording is done. The users can repeat the recording process by pressing a redo button in this page or discard the previous recording.

Details of the mobile web client can be seen in the screenshots.

9.3 Screenshots

Figure 1: Login/Registration Panel

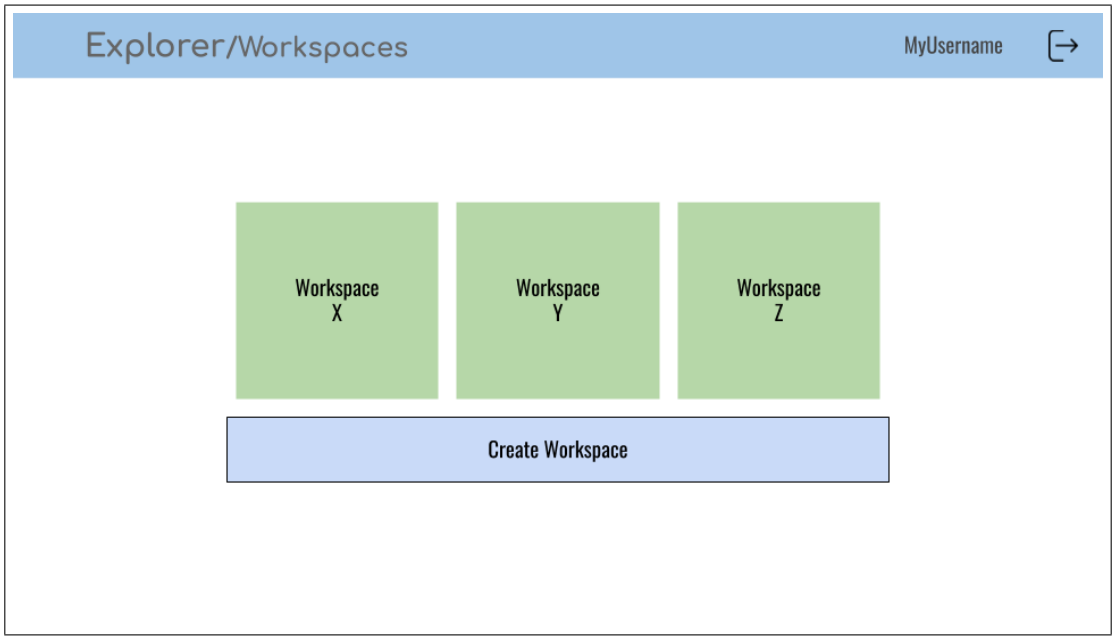


Figure 2: List of current workspaces

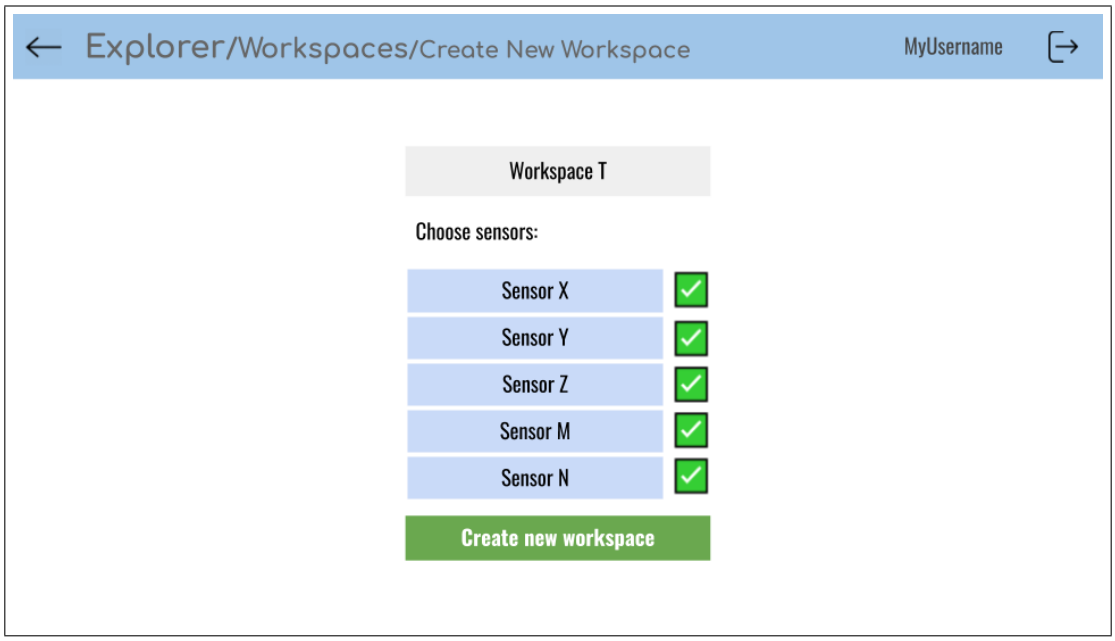


Figure 3: Create a new workspace

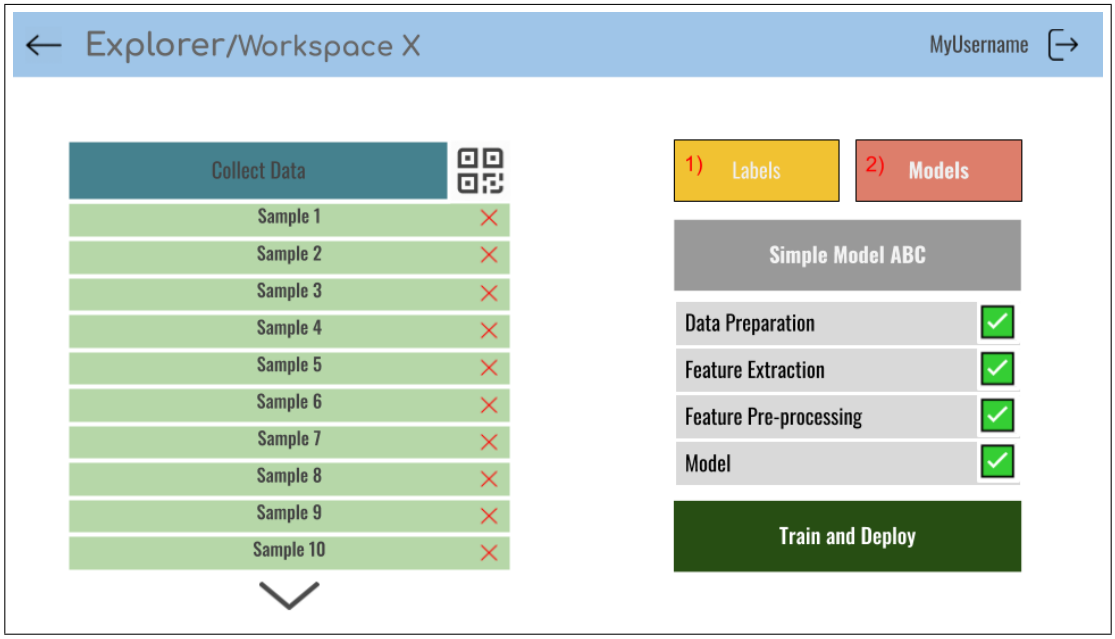


Figure 4: Workspace overview

Figure 5: QR Code/Link of workspace to collect data

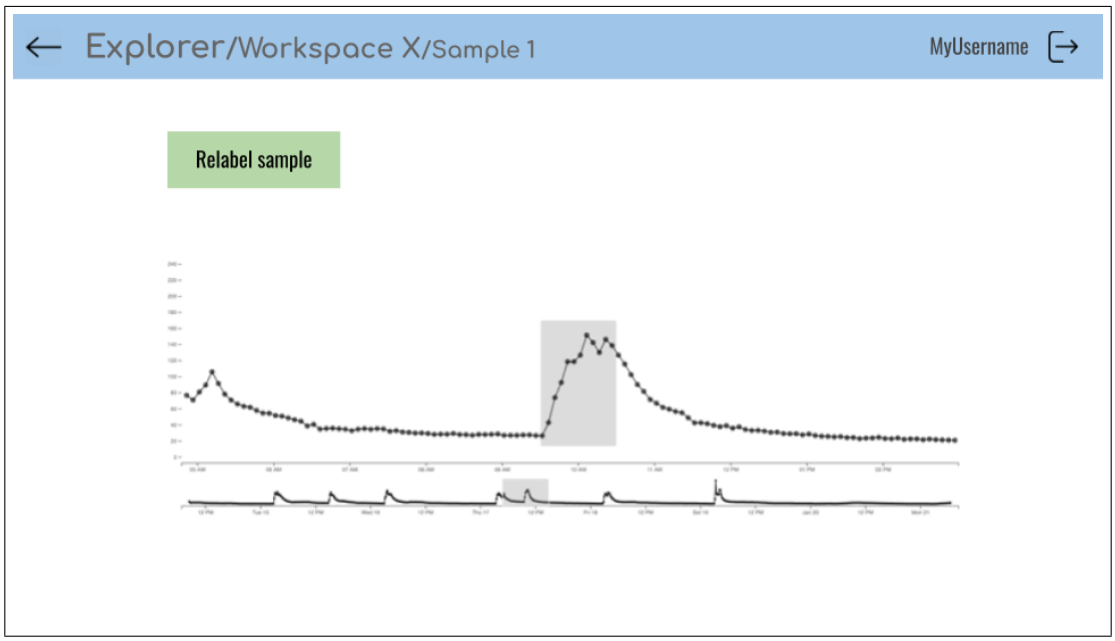


Figure 6: Data sample overview

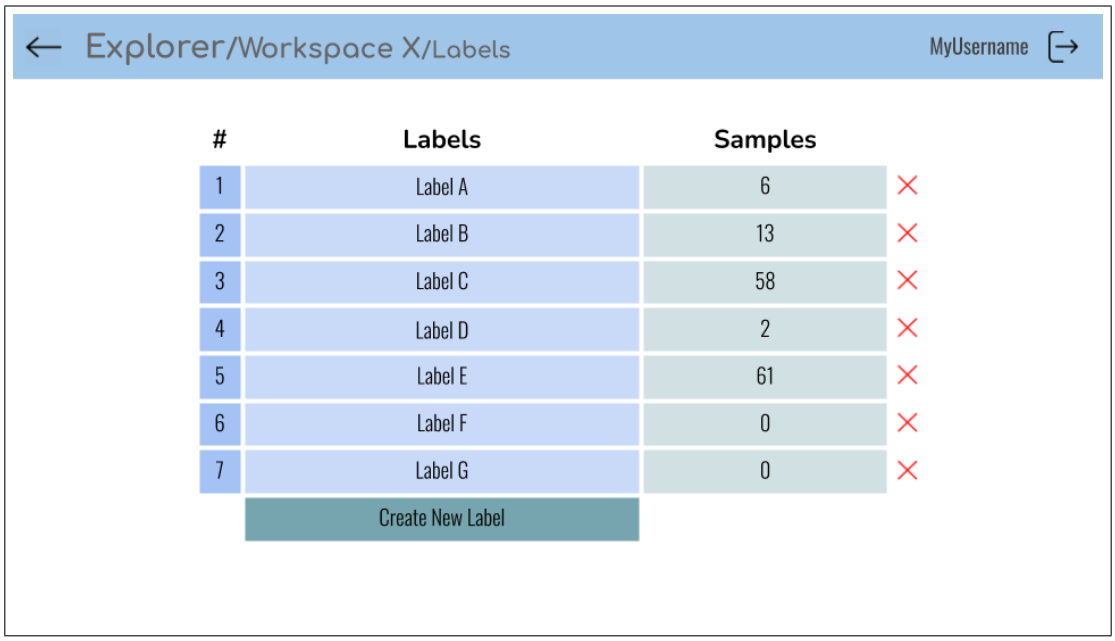


Figure 7: Labels overview

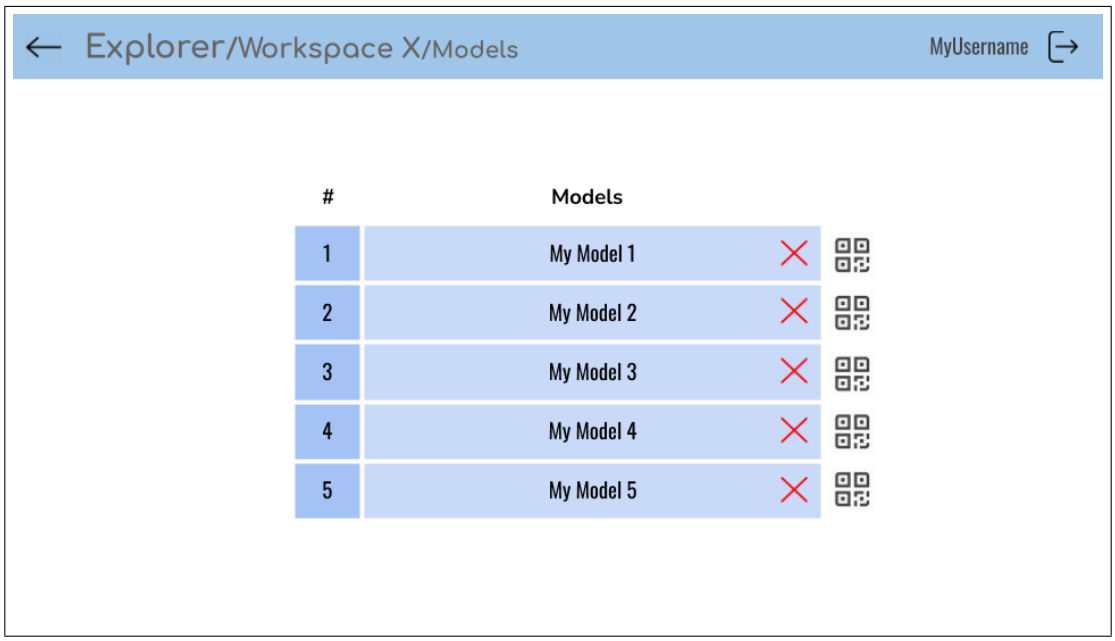


Figure 8: Trained models overview

Figure 9: QR Code/Link of trained model to classify data

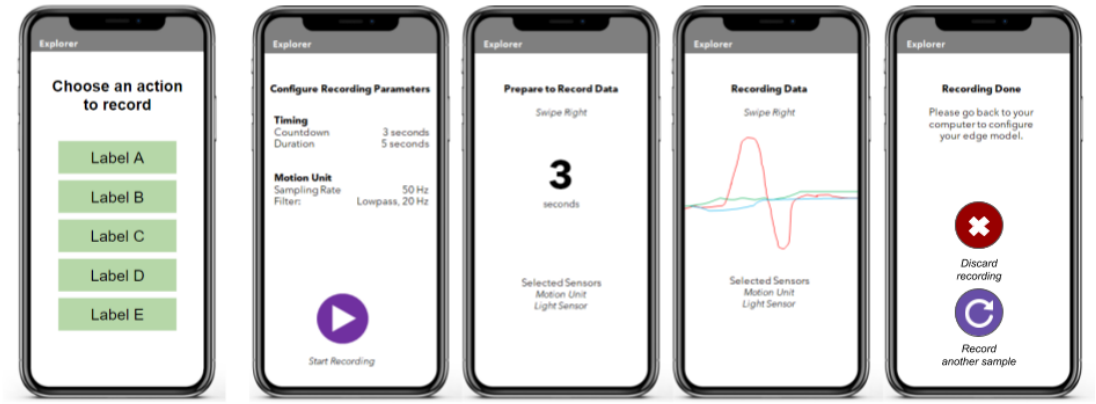


Figure 10: Data collection in mobile

10 Quality Assessment Criteria

11 Test Cases and Test Scenarios

TBD

11.1 Test Cases

11.1.1 Main Test Cases

These test cases are necessary for the web-tool to work properly and succeed in serving basic requests.

/**T010**/ Access web page (/F010/)

- /**T011**/ via Google Chrome,
- /**T012**/ via Mozilla Firefox.

/**T020**/ Create a new user. 010, 330

/**T030**/ Login with existing credentials. 010, 330

/**T040**/ Login with invalid credentials. 010, 330

/**T050**/ Open an already existing workspace. 020, 040, 060, 340

/**T060**/ Create a new workspace. 030

/**T070**/ Rename the workspace. 050

/**T080**/ Create new label. 070

/**T090**/ Rename label. 090

/**T100**/ Delete label. 100

/**T110**/ Delete sample. 140

/**T120**/ Scan QR code. 110

/**T130**/ Open the workspace link. 110

/**T140**/ Overview collected data. 120

/**T150**/ Overview an individual data sample. 130

/**T160**/ Overview trained models. 150

/**T170**/ Overview an individual model. 160

/**T180**/ Check performance metrics of a model. 170

/**T190**/ 180 ???

/**T200**/ 190 ???

/**T210**/

/**T220**/ Choose a label for the recording. 200

/**T230**/ Select sensors to use for recording. 200

/**T240**/ Set the duration for the countdown. 210

/**T250**/ Set the duration of the recording. 220

/**T260**/ Set the sampling rate. 230

/**T270**/ Record with specified configuration. 240, 250, 260, 290, 300

/**T280**/ Overview sensor data whilst recording. 270, 280

/**T290**/ Record again with same configuration. 310

/**T300**/ Record again with different configuration. 320

11.1.2 Extending Test Cases

11.2 Test Scenarios

TBD

12 Development Environment

12.1 Operating Systems

- Windows 10 with WSL1 / WSL2
- Ubuntu 20.04.1 LTS

12.2 IDEs, Editors

- VS Code

12.3 Versioning

- git
- Github for hosting

12.4 Integration

- **(we'll need a CI server, TECO has one?) FIXME**

12.5 Miscallaneous

- LaTeX for documents
- 3D Paint for mockups
- VS Code Live Share for communication
- Discord for communication

And many more to be added later...