BC-PREDICT Technical Report

Andrew Jerrison1 and Matthew Machin2

*1 Research IT, IT Services, University of Manchester, UK*

*2 Centre for Health Informatics, University of Manchester, UK*

May 2021

Classification: Public

# Summary

This report is designed to be read in conjunction with the source code of the PROCAS2 Collator software published on the University of Manchester’s FigShare site. It is a record of the software and procedures created by Research IT and used in the BC-PREDICT project. It also records how the software interacts with other 3rd party supplied software.

The report also highlights areas that were originally required but not used, and any improvements that could be made in future development of this software.

Table of Contents

[Summary 2](#_Toc75764661)

[1. Introduction 3](#_Toc75764662)

[2. Procedure 3](#_Toc75764663)

[2.1. BC-Predict Process 3](#_Toc75764664)

[2.2. BC-Predict Software 7](#_Toc75764665)

[2.2.1. Collator 7](#_Toc75764666)

[2.2.2. Receive Consent 10](#_Toc75764667)

[2.2.3. Receive Volpara Data 10](#_Toc75764668)

[2.2.4. Receive Questionnaire Answers and Risk Letter 10](#_Toc75764669)

[2.2.5. Volpara Consenting Tool 11](#_Toc75764670)

[2.2.6. Volpara Delete Stale Images 11](#_Toc75764671)

[2.2.7. Volpara Delete Non-invited Patients 11](#_Toc75764672)

[3. Technical Findings 11](#_Toc75764673)

[3.1. What worked well 11](#_Toc75764674)

[3.2. Common issues 11](#_Toc75764675)

[3.3. What could be improved 12](#_Toc75764676)

[3.4. Unused functionality 12](#_Toc75764677)

[4. Conclusion 13](#_Toc75764678)

# Introduction

BC-Predict was part of the PROCAS2 breast cancer research programme at Manchester University NHS Foundation Trust. Within this programme, mammogram patients were provided with a risk score of developing cancer within the next 10 years. The aim of the research was to study whether this would make patients more likely to take preventative action, including attending a family history clinic.

The Centre for Health Informatics at the University of Manchester were asked to provide some software for the project, and Research IT (part of IT Services) supplied some software developer support.

The project already had two suppliers of software: Volpara Solutions ([website](https://www.volparahealth.com/)) and CRA Health ([website](https://crahealth.com/)) Volpara’s software ran on the scanners in the hospitals, intercepting the images for the patients invited to the study, and then calculating the breast density score. CRA Health supplied a questionnaire web application that asked the end user about certain aspects of their own health history and family health history. It then used the results along with the density score from Volpara to calculate an individual risk score and generate a risk letter.

The PROCAS2-Collator software was the software that stood in the middle and coordinated the workflow of the research project. The main piece of software from a user’s point of view is a web application, and there are a number of smaller serverless functions that coordinate the communication between the three applications and the storage.

Microsoft Azure was used to host the application and storage.

# Procedure

## BC-Predict Process

The BC-Predict research workflow was operated by members of the research team based at Manchester University NHS Foundation Trust. All women who matched the inclusion and exclusion criteria for the research were invited to the study, by letter and sometimes additionally in person on the day of their appointment.

The research team informed the Collator software that they were invited by uploading a spreadsheet with the information. At this stage just a minimum of the NHS number was required (as that is used as the identifier in the scanners). The invitee used CRA’s software to consent to join the study, and once that consent occurred a number of automatic and manual processes initiated. First of all, the images from the scanner were processed by the Volpara breast density calculator, and the results sent to both Collator and CRA. If the images were not immediately available (i.e. the patient consented to the study sometime before their appointment) the process waited until they arrived from the scanners.

The CRA software combined the result from Volpara and the answers to the questions asked by CRA, to produce a risk score and the text of a letter to send to the patient to highlight the risk score. This risk score represented the likelihood of developing breast cancer over the next 10 years. The score and letter text were sent to Collator.

Within Collator the patient’s name and address were added, and then merged with the letter. The letter was exported then sent to the patient and their GP (general practitioner).

**Fig 2.1: Overview of BC-PREDICT data flow. Numbers show the order of progress through the system.**



**Fig 2.2: BC-PREDICT data flow in more detail.**

|  |
| --- |
|  |
| **Key for figure above:**1. **List of people attending screening downloaded from NBS (spreadsheet download).**
2. **Plaintext NHS numbers of those invited to participate uploaded to Collator (spreadsheet upload).**

**Fields: NHS Number, date of birth, date of first offered appointment.**1. **Hashed NHS numbers of those invited to participate uploaded to CRA (spreadsheet upload).**

**Fields: Hashed NHS number, date of birth, date of first appointment, study number**1. **Consent information sent from CRA to Collator (automated via Azure ServiceBus queue\*).**

**Fields: Hashed NHS number, PDF consent form**1. **Consent forms sent from Collator to cloud storage (automated via Azure Storage SDK\*\*).**
2. **Invitation/Consent information sent from Collator to Volpara (automated via Azure ServiceBus queue\*).**

**Invitation fields: Hashed NHS number.****Consent fields: Hashed NHS number, date of consent.** 1. **Full identifiable details uploaded to Collator, once consented (spreadsheet upload).**

**Fields: Title, first name, last name, 4 x address fields, postcode, GP name, 4 x GP address fields, GP postcode, screening number, screening site**1. **Screening data sent from Volpara to Collator (automated via Azure ServiceBus queue\*).**

**Fields: All DICOM fields (see separate attachment as an example). No personal identifying data, hashed NHS number used as identifier.**1. **Images sent from Volpara to cloud storage (automated via Azure Storage SDK\*\*).**
2. **Density data sent from Volpara to CRA (automated via Azure ServiceBus queue\*).**

**Fields: hashed NHS number, breast density percentage, Volpara density grade.**1. **PROCAS2 staff ask CRA to generate risk letter and score after validating the Volpara data (spreadsheet upload).**

**Fields: hashed NHS number**1. **CRA sends Collator the questionnaire results, risk score and letter (automated via Azure ServiceBus queue\*).**

**Fields: Message in HL7 ORU^R01 format, example attached. Identified by hashed NHS ID. Responses to all questions in survey, risk score calculation results and risk letter content.**1. **Risk letters downloaded from Collator (document download).**
2. **Patient outcomes downloaded from NBSS and family health clinic records (spreadsheet download) using NHS number as an identifier.**
3. **Outcomes uploaded into Collator (Spreadsheet upload or manual entry).**

**(\*) An Azure ServiceBus queue is a channel on which messages can be securely transmitted between applications.****(\*\*) Azure Storage SDK is a software development kit to allow software applications to securely interact with Azure’s cloud storage.** |

The diagrams in Fig 2.1 and Fig 2.2 show the different data flows between the component parts of the BC-PREDICT process. Note that this is the idealised order of data flow, the order of the flows connected to CRA and Volpara could in theory occur differently, for instance a patient might have consented to join the study after their mammogram appointment, however the process initiation and completion was always the same.

The researchers could use Collator to view the progress of patients through the system, i.e. see when their results were available and when to export their letter. They also had access to a number of reports that effectively allowed them to export the database, in an anonymised fashion, to allow them to study the data.

At the end of the study the database, consent forms and images were archived in a suitable storage area, either at the NHS Trust (consent forms and database) or the University.

## BC-Predict Software

This section provides an overview of the software written by Research IT for the BC-Predict project.

Microsoft Azure was used as the cloud platform to host the applications and storage. Cloud was the preferred platform of the researchers, and Azure had the added advantage at the time of being certified for NHS use. Both Volpara and CRA already used that platform so alterations for hosting environment were kept at a minimum.

In Collator, the underlying data was stored in a relational database called Azure SQL, which is based on SQL Server, modified for use in the cloud. The cloud storage for images and other data was an Azure blob store. CRA Health and Volpara also use Azure for storage and SQL purposes.

### Collator

Collator was a web application that allowed the researchers to view and control the workflow for the BC-Predict project.

Login to the software was controlled by an administrator giving access via an NHS or University email address only. The user had to create their own password, which needed to comply with the University’s password policy at the time.

The entry screen presented the user with a dashboard allowing them to see the state of the patient progress through the study at that time. An example of the dashboard can be seen in fig 2.3.

**Fig 2.3:** **Dashboard**



#### Patient Upload

The screens most frequently used were grouped under the Participant and Letters menus. A spreadsheet of newly invited patients (NHS numbers) was uploaded using the Upload New Patients option. Once consent was received (see 2.2.2) the name and address could be added using the Update Patient Details screen, once again using a spreadsheet upload. In both screens a spreadsheet upload option was preferred by the researchers because it was simpler to copy the information from the National Breast Screening Service (NBSS) using that method. As spreadsheet uploads are notoriously prone to formatting errors this meant that a lot of validation was required around each of the fields in the spreadsheets, to maintain data integrity. An example spreadsheet upload screen can be seen in fig 2.4.

**Fig 2.4: Example of a spreadsheet upload: Upload New Participants**



When a new patient was created an output file that could be uploaded to CRA was created, so that the patient could use their NHS number to consent and fill in the questionnaire. The NHS number itself was not directly stored in CRA’s database – Collator encrypted it and CRA checked the patient’s entered NHS number against that encryption.

#### Patient View and Edit

The Patient View and Edit screen (Fig 2.5 and 2.5) was used to look at and modify details of specific participants in the study. Data such as study outcomes was entered here, and it was a place to view the Volpara data and questionnaire responses. All changes to the data made by users was recorded in an audit trail.

**Fig 2.5: Participant Edit screen**



**Fig 2.6: Participant View screen**



#### Letter Generation

Letter text was imported into CRA in a manual process. The patients first had to be checked that all their Volpara details were correct (see 2.2.3), and then marked for letter request. A text file of NHS numbers could then be output from Collator and uploaded to CRA. This would trigger the sending of letter and risk information to Collator (see 2.2.4). Names and addresses would then be added and the letters exported using the Export Letters menu option.

#### Additional Functionality

Additional functionality in Collator included a number of reports to allow researchers to quickly access certain data of interest, and a database download function which exported all the data into a spreadsheet (anonymised) so that they could create their own queries should they need to. There was also another spreadsheet upload function that allowed researchers to update study outcome information in batch, and a data entry screen to allow entry of histology information. A couple of screens which allowed us to simulate incoming and outgoing consent messages in the pilot phase of the project were deemed useful enough to keep in the main phase version of the software (see 3.2).

### Receive Consent

The Receive Consent program was a small Azure Webjob, which was published as part of the Collator web application. Webjobs are functionally separate from the web application, and effectively run as small batch programs on the same server. In this case the webjob monitored an inbound Azure ServiceBus queue for incoming messages. These messages were sent by the CRA software when a patient consented to join the study. They were short text files containing the hashed NHS number, encoded PDF consent form and the date of consent. When the webjob received this message it marked the patient as consented, stored the consent form in Azure storage, then sent a message via another ServiceBus queue to the Volpara Consenting Tool (2.2.5) to say start the image processing for this patient.

### Receive Volpara Data

The Receive Volpara Data program was an Azure Webjob, which was published as part of the Collator web application. In this case the webjob monitored an inbound Azure ServiceBus queue for incoming messages. These messages were sent by the Volpara software when an image was processed and the breast density calculated. They were long DICOM files containing the hashed NHS number, reference to the image name, and lots of data from the scanners. When the webjob received this message it stored the data in the Collator database.

### Receive Questionnaire Answers and Risk Letter

The Receive Survey Details program was an Azure Webjob, which was published as part of the Collator web application. In this case the webjob monitored an inbound Azure ServiceBus queue for incoming messages. These messages were sent by the CRA software when the patient had completed the questionnaire and the researcher had asked for the risk letter to be generated. Usually the researcher would wait until the Volpara data (2.2.3) had arrived before asking for the letter. They were long HL7 files containing the hashed NHS number, the questionnaire answers, and the risk letter text. When the webjob received this message it stored the data in the Collator database and marked the patient as ready for letter export.

### Volpara Consenting Tool

The Volpara Consenting Tool program was an Azure Function, originally written by Volpara but modified slightly by Research IT. Functions are functionally separate from the web application. In this case the Function monitored an inbound Azure ServiceBus queue for incoming messages. These messages were sent by the Collator software (2.2.2) when the patient had consented to join the study. They were short text files containing the hashed NHS number. When the Function received this message it would search for any images for that patient, and if found they would be transmitted to Volpara for processing, then moved to the Collator permanent image store. If not found the message would be requeued to be tried again 24 hours later, with a maximum of 42 attempts (6 weeks).

### Volpara Delete Stale Images

The Volpara Delete Stale Images program was an Azure Function, originally written by Volpara but modified slightly by Research IT. In this case the Function was on a timer to run every 24 hours. The Function would search for any images in the incoming folder older than 6 weeks old, and if found they would be deleted. This was necessary, as the images for invited patients were held for 6 weeks, in case they consented to join the study during that time. After that window they needed to be deleted.

### Volpara Delete Non-invited Patients

The Volpara Delete Non-invited Patients Images program was an Azure Function, originally written by Volpara but modified slightly by Research IT. In this case the Function was monitored every image that was transmitted to the Collator incoming folder. The Function would delete any images if the patient had not been invited to the study.

# Technical Findings

## What worked well

1. The link between CRA and Collator worked very well from a technical point of view. During the entire 3 year period of operation there were no errors attributable to the communication link and only 1 error of any kind, which was due to a problem in the CRA database.
2. The dashboard was popular as it was very easy to get a list of patients for which the researchers needed to perform an action such as export a letter or follow up because they’d consented but not attended their appointment.
3. The letter generation and reporting mechanism worked without issue, so data and other outputs could be obtained easily.

## Common issues

1. Incorrect details of invited patients could be uploaded to Collator by error. This especially caused a problem if the wrong NHS number was used, as Collator/Volpara would not be expecting their images, so would not record the data. There was no real software method of remedying that; in order to automate the process of transcribing the NHS number from NBSS to the upload spreadsheet NBSS would need a public API that we could use. However that did not exist.
2. For various reasons the Volpara software often missed processing images. Since this process was triggered by the participant consenting to join the study the researchers often had to manually send through the images again, and simulate the participant consenting using one of the screens in Collator. The consent date on the consent form was left untouched however, since that was the official record of when the consent happened.
3. The Volpara Functions originally wrote to a physical log file, rather than using Azure’s logging. This resulted in lots of race conditions and crashes. The Functions were modified to only use Azure Functions.

## What could be improved

1. The login functionality could be upgraded to 2-factor authentication (2FA). At the time of writing the software no institution-approved 2FA system existed. The University has since approved Duo, however that is only available to those with a University account, and the users of Collator would not have that. Google Authenticator would be the obvious choice for an alternative, provided that approval could be received, and the research staff had the means to receive the generated code (something not always possible within the research locations).
2. Access to the Collator software could be limited to within institution VPNs only. At the time there were no administrators with the necessary knowledge to tell us how to do that.
3. In the end the checks on the Volpara data did not often take place. This negates the need for a manual intervention in the process for requesting letters and so the step could have been removed from Collator at least, to reduce administrative overhead.
4. Although Azure SQL databases are encrypted by default, and Collator’s database admin access restricted to a single IP address, it might have also been a good idea to encrypt some identifying data. This wouldn’t be one-way hashes (like the NHS number or passwords) but it would increase the obfuscation. The problem this would create would be a slower browse/search/letter export time as Collator would need to unencrypt everything on the fly, plus it is likely that if someone had access to the database then they’d also have access to the source code, which would negate any security benefits anyway.
5. Approaches to software security have progressed since the Collator software was developed. If it is used for a future project it would make sense to undertake a security threat modelling exercise to ensure that all current security approaches are incorporated.

## Unused functionality

A fair amount of functionality was requested in Collator but not used:

1. None of the Histology edit screen was used in practice, so it is not known how well this would have worked for the researchers.
2. It is thought that the spreadsheet upload of study outcome information was not used, as the manual per-patient edit view was preferred for this task.
3. It is not thought that the screens for viewing incoming Volpara data were used, as this was of more interest as a spreadsheet report. The same could be said of questionnaire responses. These were quite complicated screens that probably didn’t need to be developed.
4. The researchers did not use the Site, Error reports or User edit functionality in Collator even though they had access to do so; this was left to the software engineer to view and maintain so was probably over-engineered in the end, as they had been designed for a non-technical user.

# Conclusion

The Collator software was a success operationally. It fulfilled all the researchers’ requirements and there were no bugfixes required after the initial pilot stage. The project needed a piece of software to coordinate the workflow and collate the incoming data. Quite a bit of the developed functionality was not used, which might well inform future projects; they needn’t have spent so much money on the development time. At the time they had the money though, so added in a few ‘would like to haves’.

Since Collator was written there have been legislative changes and new technologies developed that would affect the software. Any future use of this software should consider upgrading to 2 factor authentication of technology, and further auditing of data taking into account not just editing but also viewing. Refinement of the research protocol learning from the lessons of BC-Predict might well result in more automation which would reduce the amount of time researchers needed to spend operating the software when manually updating information.

There was more administrative overhead than expected, both technical and for the project. This was partly down to having to use a cloud provider and managing invoices and notifications from Azure. All through the project there were issues with the Volpara software not working as expected, and this often meant lengthy investigations into the Collator logs to try and eliminate issues. Similarly, although the CRA software worked well it was very expensive to run, requiring 4 large virtual machines, 2 of which were continuously on, plus other expensive infrastructure such as a Redis installation.

Overall Collator was successful technically and the end users (the research team) were very positive about its usability and design. The project itself was successful in its aims and has produced a number of academic papers.