Rigshospitalet

PET & Cyklotronenheden

Technical documentation and risk assessment for Tracershop

Christoffer Vilstrup Jensen
MSc. Computer Science

Introduction

This document describes the electronic web shop, Tracershop, used for ordering and release of radioactive tracers for clinical procedures produced at Rigshospitalet's cyclotron unit. Since radioactive tracers are medicinal products, Tracershop must fulfil some documentation requirements stated by GMP Volume 4 annex 11, which this document fulfil.
Rigshospitalet delivers radioactive tracers to hospitals and scientific institutions around Copenhagen, that would not be able to perform various PET scans without these tracers. Therefore Tracershop should be considered a important piece of software.
Tracershop was released in 2004 and is used to this day. Due to it's age a new system have been developed with the intent to replace the old system. This document contains a short description of the old system, then a deeper explication in of some techniques and technologies that the system is using to obtain a "Better" software product.
Since annex 11 is dated, EMA has released a concept paper on annex 11[1] containing various comments on the guidelines presented in annex 11. These comments have also been considered when applicable.
The document have been review with Jacob Madsen and Helle Østergren Bak as Process owners, and Christoffer Vilstrup Jensen as developer.

User Requirements

The requirements of the software, that tracershop fulfils are:

- A user can order radioactive tracer.
- A user can review any order they have made, and view batch numbers of any product that they have received. They cannot alter confirmed or released orders.
- A user cannot view or alter order, which doesn't belong to a related customer.
- A certified user can release a radioactive tracer.
- A released order has a batch number and a record of who released it.
- Non authenticated users cannot alter or view information in tracershop.
- A released tracer must display if it's intended for veterinary usage or human usage.

Terminology

Tracershop handles two different types of orders. Activity based orders and Injection based orders.
An **activity order** is defined as: An order, where a customer orders an amount of MBq radioactive tracer at a predetermined time slot known as a "deliver time". It's the user responsibility to account for radioactive decay between injection time and delivery time.
An **injection order** is an order with a number of injections with a predefined amount of activity, and it's Tracershops responsibility to account for any radioactive decay between production time and injection time.
Users er limited which injection tracers they can order.

Tracershop - Current system

The software ecosystem is centered around a MySQL 5.1 database running on a openSUSE 11.2 distribution, which contains all the records and logs about the production of tracers. This database is back up on an external machine every day at midnight.

Tracershop is a web site hosted with Zope 2-2.13 web interface at http://pet.rh.dk allowing the users to systematically write to the main database. These messages are logged.
The website is deprecated and no longer receive updates. The system contains an dispenser, that records how much radioactive material was dispensed, allowing accurate tracking of activity in vials and minimizing human error. This data is pushed via a script to tracershop.
An overview of the current system can be seen in figure 1.

The system is hosted on internal server owned by The Department of clinical physiology nuclear medicine. This is not desired because hosting server is a core task for CIMT and not for the clinical departments.

[1] https://www.ema.europa.eu/en/documents/regulatory-procedural-guideline/concept-paper-revision-annex-11-guidelines-good-manufacturing-practice-medicinal-products_en.pdf

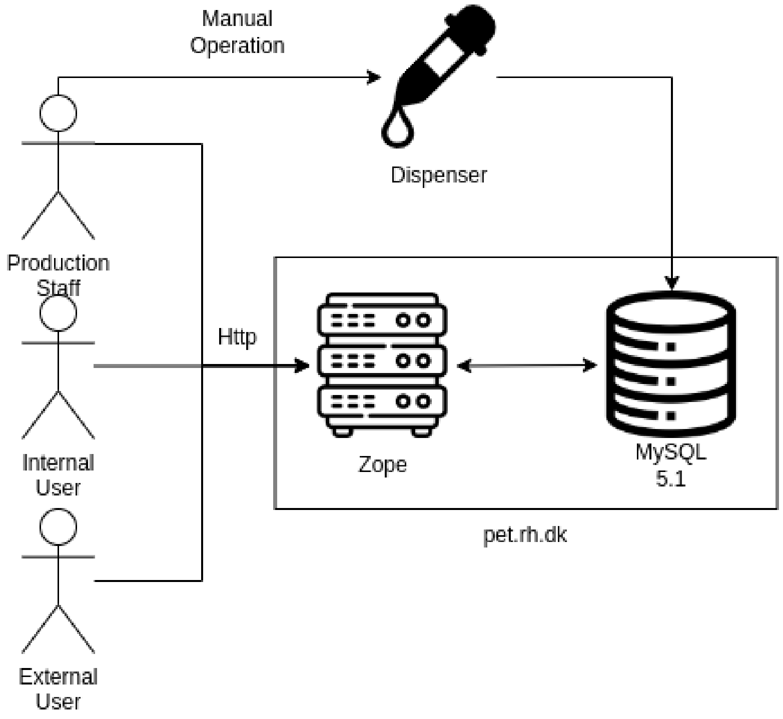


Figure 1: Current Tracershop system

Database layout

The database contains the following tables:

1. Log - Likely Zope related. No longer in use as the last entry was in 2010-03-18.
2. MiscData - Likely Zope related. Likely a temporary value container.
3. Roles - Zope related, defines different user roles in the Tracershop program.
4. Sessions - Likely Zope related, defines active user sessions.
5. Tokens - Likely Zope related, Likely defines the length of active sessions.
6. TracerCustomer - Defines which user have access to order which injection Tracer.
7. Tracers - Catalog of tracer available in tracershop.
8. UserRoles - Relates user to a Role from the Role table.
9. Users - All users able to be authenticated in tracershop
10. VAL - Record of a vial with tracer, produced by the dispenser.
11. VAL2 - Not in use.
12. blockDeliverDate - Extraordinary dates where tracershop is closed, such a holydays.
13. deliverTimes - Weekly points in time specifying when a customer can place an activity order.
14. isotopes - Catalog of radioactive isotope used in the tracers.
15. orders - List of activity orders.
16. productionTimes - Weekly points in time, where a production should happen. Entries are also referred as a run.
17. productions - Production of tracers.
18. storage - Old mails, assumed not in use.

19. t_orders - List of injection orders.

The database is not utilizing the foreign key restriction, meaning that the relations are ensured at application level and not the database level, and as such only reaches the first level of database normalization. This low level of normalization allow a number of error to be present in the database:

- Reference errors - When a field references another table without the foreign key restriction, then the reference can point an entry that doesn't exists, which is a error, therefore any change to an ID, must by application logic update the rest of database.
- Transitive functional dependencies may not respected. As an example: An activity order contains both the amount of radioactive material was ordered and how much tracer should be produced while taking into account the overhead percentage of the customer. If the ordered amount is edited then the database integrity relies on application logic to update overhead amount.

It's considered an industry standard to keep databases in third normal form, which require the database to be designed in such a way, that these errors cannot occur.

The new Tracershop

The idealized setup can be seen in figure 2, it's two virtual computers hosted by CIMT running Red hat 9 https://www.redhat.com/en:

- tracershop.unix.regionh.top.local - Default server for common usage.
- tracershop-dev.unix-regionh.top.local - Development server, intended for backup usage and development.

The servers run the following programs:

- Apache - open source web server, https://www.apache.org
- Daphne - Websocket protocol server, developed for django channels https://github.com/django/daphne
- Dispenser script - This is script responsible for fetching data from the dispenser, however this might not necessary run on machine tracershop.unix.regionh.top.local, since it's dependant on file events, however ideally it runs on the host machine.
- MariaDB - Open source Mysql database - https://www.mariadb.org
- Redis Database - Open source in memory database - https://www.redis.io
- Supervisor - Monitor program for Daphne http://www.supervisord.org

The web server daphne hosts is an open source django / channels web server found at https://github.com/demiguard/Tracershop, this program will be refed to as "the new Tracershop".
The new tracershop uses common software packages for web servers:

- React - Frontend Javascript library developed by facebook. https://reactjs.org
- Django - Backend Python web server library https://djangoproject.com
- Channels - Django extension for using websockets. https://channels.readthedocs.io./en/stable

These packages are well supported and all in active development, so their developers will continue to provide security updates.

Message handling

The new Tracershop utilize websockets instead of purely http communication, because websockets allows the server to push updates to the users unprompted, because the protocol uses a persistance connection, while http does not use a persistent connection. This technology allow the communication protocol seen in figure

1. A user causes an update, that needs processing from the server.
2. A message is send by the server
3. The message is processed by the server

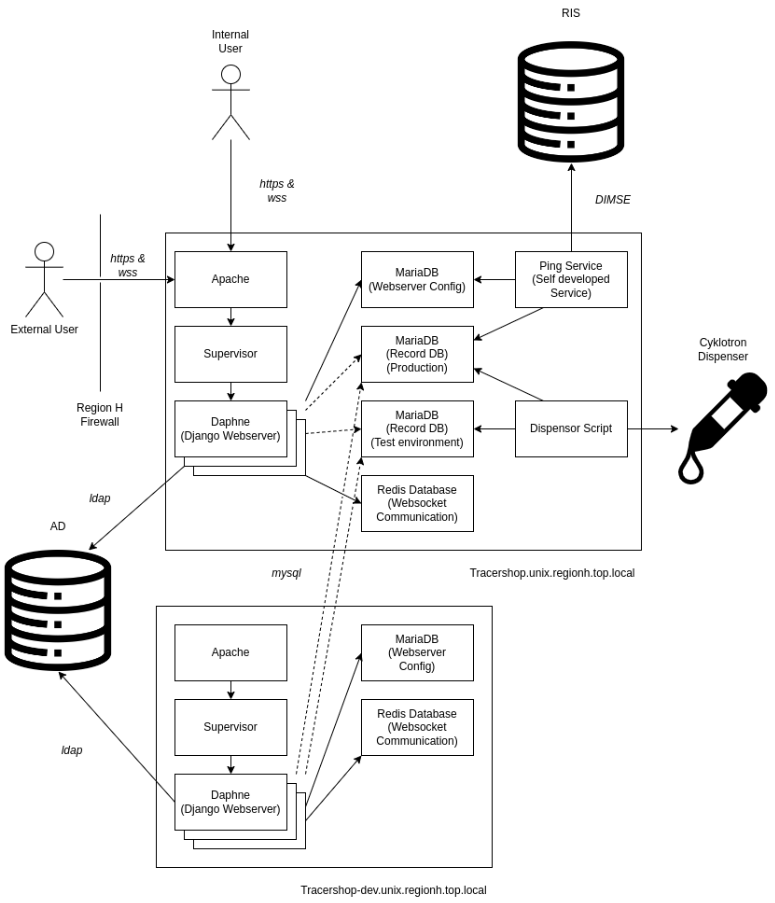

RIS
Internal User
https & wss
DIMSE
External User
https & wss
Apache
Supervisor
Daphne (Django Webserver)
MariaDB (Webserver Config)
MariaDB (Record DB) (Production)
MariaDB (Record DB) (Test environment)
Redis Database (Websocket Communication)
Ping Service (Self developed Service)
Dispensor Script
Cyklotron Dispenser
Region H Firewall
ldap
AD
mysql
Tracershop.unix.regionh.top.local
Apache
Supervisor
Daphne (Django Webserver)
MariaDB (Webserver Config)
Redis Database (Websocket Communication)
ldap
Tracershop-dev.unix.regionh.top.local

Figure 2: The desired system

4. the server sends a responses, that the message have been processed. If the message type require an update websites state, then the server informs all clients by broadcasting the message.

5. The server rerenders the website with the modified state in mind.

To ensure that each message is valid, and that user is authorized to send that message, all message are processed in a three step plan seen in figure 4

1. Validate message - The server determines if the message has all the needed fields and that the fields contains values of the expected type. It also ensures that sender doesn't have an outdated version of the frontend client. If validation fails, there's no update to the database.

2. This checks that the sender is authorized to send, this message type. If authentication fails, there's no update to the database.

3. Message processing, this step performs the message.

New login system

The new tracershop system uses an external authentication system, provided by the Capital region called BAM ID, managed by CIMT. All members of staff working in the capital region have a BAM ID login.
This login have various security features build in, such as automatic deprecation of passwords, password reset, deactivation of inactive accounts, and minimum complexity requirements to passwords. CIMT also handles reestablishment forgotten passwords. Secondary tracershop doesn't need store passwords and instead store tokens that the system can authenticate against CIMTs systems. That was if the system is compromised, no data about the users passwords is leaked.
All users in the new tracershop are personal and have a role. These roles are divided into 3 catagories:

- Shop - Users ordering tracers

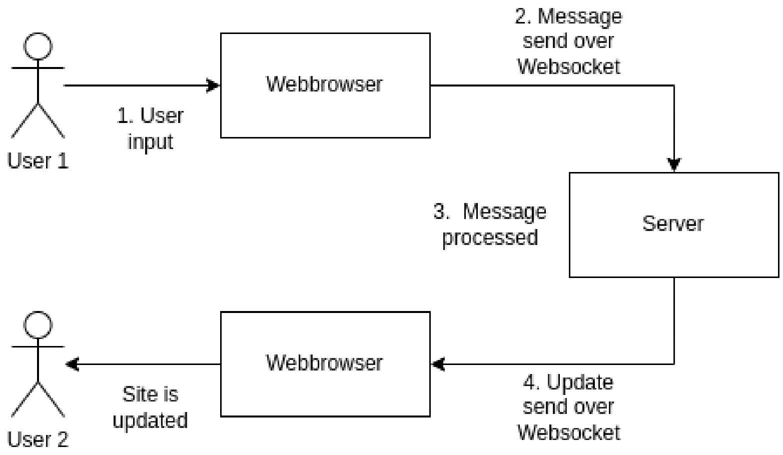


Figure 3: Overview of the message handling.

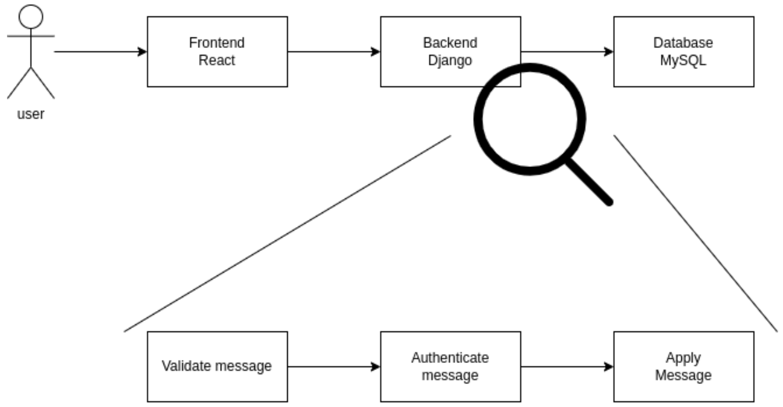


Figure 4: Message handling by the backend

- Production - Users producing tracers
- Admin - Users administrating the site, otherwise detached from both ordering and production of tracers

Users from different catagories uses a different interface. This is because the task in tracershop performed by users in different catagories are fundamentally different.
A user acquires a role by contacting the local active directory manager, requesting the role. The manager can then determine if the role that is requested is appropriate and accept or deny the request based on their judgement.
All the roles are as follows:

- Shop - External - The user isn't an employee of the Capital region and doesn't have BAM ID. They can view their assigned customer's orders and order tracer authorized to the customer. Each of these user are connected to an external customer. The user is managed by an production admin.
- Shop - Internal - This is an employee of the Capital region and therefore have a BAM ID. A user is associated with one or more internal customers. This user is managed by a "Shop Superuser" user.
- Shop - Superuser - This is an internal shop user with additional rights to modify the automatic generated order based on a RIS booking. They can grant an internal user access to the same customers that they moderate.
- Production - User - The user can view orders from all customers and release tracer of authorized types.
- Production - Admin - Everything the production user can. Grant release rights to other production users, create new tracers, allow tracers to ordered by customers, and manage external user.
- Site Admin - The equivalent of a root user. The user can mimic all the functionality of other roles. The user isn't intended to use tracershop only to support it.

A shop user is associated a number of customers. These associations are managed by the various admins and superusers of the site. By default user isn't associated with any customers, and therefore cannot view any orders until they have been associated with a customer, then they can view and order on behalf of that customer.

Logging

Annex 11 is very light on it's requirement of logging, therefore we refer to the comments in the concept paper. The raw logs are only accessible by system administrators, however information contained in the logs may be visible in tracershop. Each log entry have a time stamp, and a record of the user which invoked the user statement. An anonymous User is given to any log message triggered by a user, who is not logged in. The new tracershop contains the following logs:

- Audit log: The audit log contains significant action with regards to GMP. The log is permanent, but not backed up. The following actions are log in the audit trail:
  - The system automatically assign a user group to a user.
  - The system automatically removes a user group from a user.
  - A production user / admin manually adds a vial, data of the vial is logged.
  - A production user / admin manually edits an existing vial, both old and new data is logged.
  - A production user / admin frees an order.
  - A production admin grants releasing rights to a production user.

  There's a number of actions which where considered for the audit log but, were left out due to reducing cluttering of the audit log. These are:
  - A shop user creates an order
  - A production user accepts an order
  - A production user moves an activity order to another activity delivery.
  - The system creates a ghost order due to movement of an activity order
  - A user edits an unconfirmed or accepted order
  - The system creates a vial from the dispenser script.
  - A user logs in, the IP of the machine is also included in this log message.
  - A user fails to log in, due to incorrect password. The IP of the machine attempting to log in is also included.
- Operational Log: General operational messages - Indented for debugging. Weekly Rotating log, up to 4 logs are kept
- SQL log: Log of SQL command to the underlying database - Indented for debugging. Weekly Rotating log, up to logs are kept

Testing and Development

Tracershop is a program is developed under the strategy of "continuous test-driven development". This means that the development team will continue to develop new features at the request of users, assuming the resources are available. To ensure correctness of new features and that the changes doesn't break any previous functionality of tracershop, test programs are written. These program execute a predetermined feature or functionality of tracershop, then compare the actual state and return value to the expected state and return values and flags any disparity.

Currently there is 139 tests for the backed server and 121 tests for the frontend. However these are numbers are likely to increase as development time is increases. A small high light of these tests cases are seen below, showing core functionality of tracershop.

- `websocket.tests.test_Consumer_end_2_end.test_GreatState`
  Ensures that when a user connects they receive a correct image of the database.
- `websocket.tests.test_Consumer_end_2_end.test_free_vial_dependant_orders`
  Ensures that an activity order is freed correctly.

- `lib.tests.tests_pdfGeneration.test_PDF_Order`
  Ensures that the generated pdf matching the Activity Order. This test require manual verification since pdf files are not an easily parsed format.

- `lib.tests.tests_pdfGeneration.test_PDFInjectionOrder`
  Ensures that the generated pdf matching the Injection Order. This test require manual verification since pdf files are not an easily parsed format.

- `frontend.src.tests.components.ProductionPages.ActivityTable.White Box Order Rendering`
  Ensures that order data is displayed correctly.

Risk assessment

Tests cannot by definition only showcase that the program works in a predefined "idealized" environment nor do they deal with incorrect data. To combat the problems uncovered by the automated testing. There has been made a risk assessment below.
A risk assessment is a collections of risks, where each risk consists of:

- A description of the risk.
- A highlight of the current system
- A likelihood of how likely an incident is to occur.
- A damage estimate of an incident.
- A plan of action if an incident happen.
- How the new system reduces this risk

This risk assessment doesn't include production related risks such a dropped vial, or tracers not passing quality assurance. The dispenser and the program transferring data have not modified by the new Tracershop, and therefore derives its validity from previous risk assessments. See document number.: PVP-PROD-Tracershop-001-14.01 in D4, for the latest validation of Tracershop.

- Loss of a server.

Description - A hosting server might become unavailable for a number a reasons: A foreign threat might encrypt the entire server, hardware failure, or a critical files might become corrupted due to aging hard drive. This is not an exhaustive list of reasons for server loss however other factors have minimal likelihood. It's beneficial to have plan of action in the event of an incident caused by an unknown risk.

Currently - Currently the service runs on old hardware, insecure protocols and outdated software, which raises the likelihood of an incident occurring.

Likelihood - Low to medium, Likelihood is increasing with age of the system.

Damages - High - Without the server, it would be impossible to create electronic records of produced tracers.

Plan - Repairing the server hardware might be possible and could restore the system to a pre-incident state, within 1-5 working days. If that's not possible to repair a new server is required. Assuming that the system is not deployed on CIMTs services and a spare server is not available, procuring a new server would take between 4-12 weeks with a 1 week installation period afterwards.
Any data not in the backup would be lost.

New System - The system is hosted by CIMT, in a virtualized environment, allowing for easy cloning and therefore the recovery period. A backup server is also available for usage.

- Database is brought into an invalid state or incorrect state.

Description - Due to the fact that it's the application responsibility to ensure correctness of the database, this is a risk of incident.
If the record database is in an invalid state, undefined behavior might occur because some data is invalid.
Consider an example where a tracer is deleted. All orders with that tracer can no longer be determined to be of that tracer.
A common source of errors have been duplicate acceptance messages, this have causes orders appear released. In the new tracershop, both accepting and releasing orders are idempotence operations.

Currently - If the database is in an invalid state, the site is unavailable.

Likelihood - Low - The web services doesn't allow, the user to perform arbitrary SQL queries, only predefined queries that take the database from one valid state to another valid state. It's difficult to ensure all possible user inputs, so it's possible that a user query might bring the database into an invalid state. An incident could occur if a system administrator creates a query that brings the database is into an invalid state.

Damages - None to Low - Restoring the database is an easy task for a system administrator, worst case would be to revert to a backup. If the invalid state persist unnoticed for more 1 day the backup will be over written and the damage made permanent.

Plan - Upon notice a system administrator would enter the database and create a query to revert the database into a valid state. All queries made by tracershop to the record database is logged, which can greatly help system administrator to undo the damage. If the system administrator unable to restore the data using a query, then the plan switches to rollback to a backup of the record database, causing 1 days worth of data loss.
Note that direct queries to the database is not logged to the audit log, therefore their actions are untraceable, similarly their edits might not record the previous data, that is required to be logged.

New system - A new database have been designed which is on 5 normal form, eliminating reference and transitive errors, making it impossible to bring the state in a invalid state. A page for unexpected errors has been created, giving some hints to what the error might.

- Incorrect user input

Description - Tracershop have a number of fields, where the user must write some data in. Most critically this includes batch numberBecause it's humans that write this data, the system is subject to human error. It's incredible difficult to prevent this as there's nothing inherently wrong with incorrect data user input. This also includes whenever a user forgets to update a piece of data.

Currently - No warnings are given when attempting to release an order from another day.

Likelihood - Very high. Humans use this program, no further elaboration required.

Damages - None to High - Not all data in tracershop is critical, however if incorrect batch number is written to the database and the error is not noticed, then that could put patient safety at risk. If the incorrect data is auxillary, then there's no damages.

Plan - If the data is committed, a system administrator must edit the data, if not the data can be edited in tracershop.

New Systems - Additional warning has been made for commons errors, like freeing orders from the wrong date. See figure 5.

- The dispenser script stops working

Description - The script that pushes data is highly fragile. The exporting of data is not a build in feature of the dispenser, instead it writes to a local database. By monitoring file events the operating system can call a script which uses a predefined offset into this binary file. Then the script reads the data and export it to the database via SQL. The Author of this script no longer works at Rigshospitalet. Configuration of the script is also not known at time of writing, that might cause difficultly in moving the script, however moving would not prevent current operation.

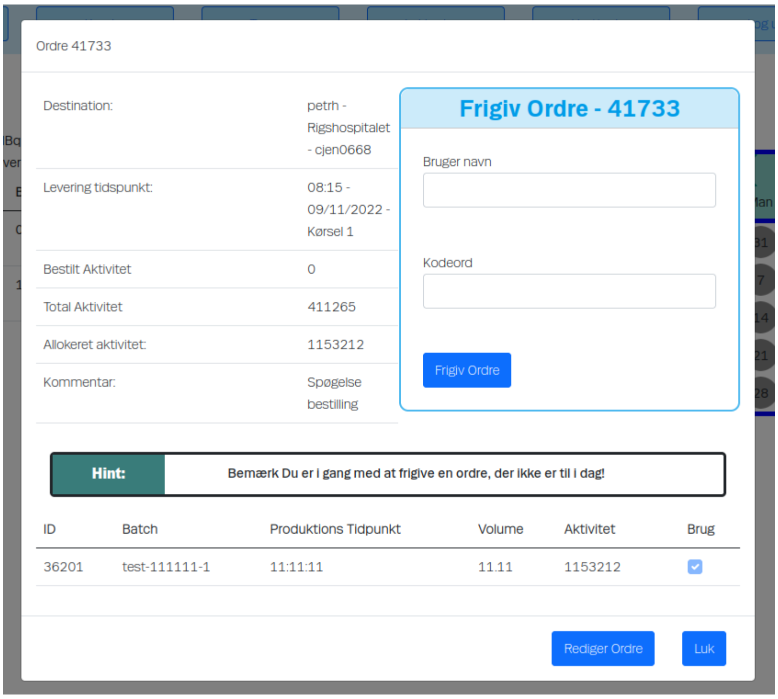
Likelihood - Low to medium - An update to the dispensers database would render the offset incorrect. There might be other sources of errors that might cause the offset to wrong, and in such a case recovery might be difficult.

Damages - Low - The service would be very difficult to repair, due to the author of the program no longer working at Rigshospitalet, and is a forking perl script. However the loss of the service would not incorrect or invalid data by itself. Only an additional source of error and additional human labor is required.

Plan - Users would be forced to manually input data, that normally would have been transferred automatically and correctly.

- Email server is unavailable

Description - The email service is external, thus may for any reason be unavailable for an undefined amount of time.



Ordre 41733

Destination:	petrh - Rigshospitalet - cjen0668
Levering tidspunkt:	08:15 - 09/11/2022 - Kørsel 1
Bestilt Aktivitet	0
Total Aktivitet	411265
Allokeret aktivitet:	1153212
Kommentar:	Spøgelse bestilling

Frigiv Ordre - 41733

Bruger navn

Kodeord

Hint: Bemærk Du er i gang med at frigive en ordre, der ikke er til i dag!

ID	Batch	Produktions Tidpunkt	Volume	Aktivitet	Brug
36201	test-111111-1	11:11:11	11.11	1153212	☑

Figure 5: Modal for releasing an order

Likelihood - Unknown - The author have so limited experience with the available of the mail server, that a valid likelihood estimate is difficult to make.

Damages - None to low - Customers are required to wait for confirmation that the tracer passed quality assurance mandated by GMP, so if the email server is down, the customer can not be notified through email.

Plan - Delivery bills can be send via mails, phone calls and other communication.

New system - The users can download delivery bills in new system, emails are deprecated and will shut down.

- Malicious usage of tracershop

Description - A user of tracershop attempt to sabotage the site. This risk also includes impersonation of staff members.

Currently - All communication is unencrypted, the service is vulnerable to SQL injections.

Likelyhood - Minimal - Users both staff and customers are verified users, which minimizes the risk of malicious usage.

Damages - Low - Even in the event of that a user some how manages to destroy the record database, the backup is stored externally and thus not subject to any attack by the malicious user.

Plan - All account are Personal. Critical actions such as freeing orders are audit logged. Allowing system administrator to identify the malicious user. There's a detection of SQL injections before the execution of each query, if it detects an injection it discards the query.

New system - Security whole with SQL injections are patched. Communication is send over encrypted channels.

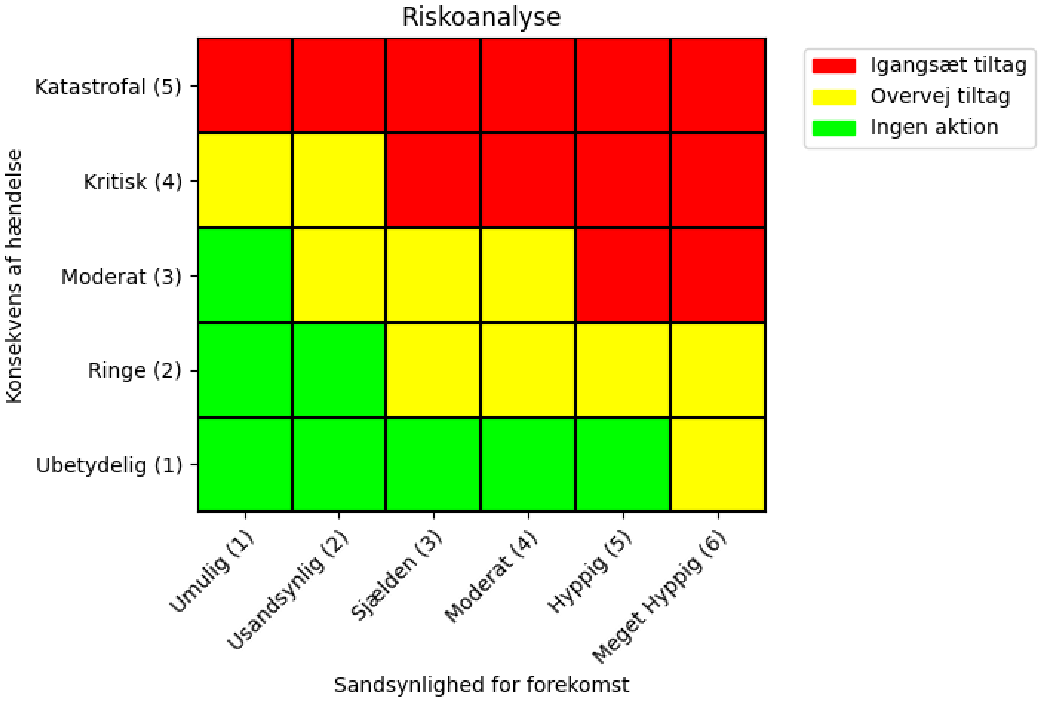
Conclusion

The new system contains substantial upgrades to user experience, IT security, and functionality. It fulfil the requirements stated by GMP Volume 4 Annex 11.

Glossary

CIMT	Center for IT and Medico technologies. Responsible department for IT in the Capital Region of Denmark.
EMA	European medicines Agency - www.ema.europa.eu.
GMP	Good manufacturing practice.
incorrect data	Data that doesn't reflect reality. Examples include a misspelled batch number or an activity entry which does not match actual activity in the vial.
invalid data	Nonsensical data that could never be valid. Examples include a reference to an nonexistent object or Negative halflife or activity.

Appendix


Riskoanalyse
Igangsæt tiltag
Overvej tiltag
Ingen aktion
Konsekvens af hændelse
Katastrofal (5)
Kritisk (4)
Moderat (3)
Ringe (2)
Ubetydelig (1)
Umulig (1)
Usandsynlig (2)
Sjælden (3)
Moderat (4)
Hyppig (5)
Meget Hyppig (6)
Sandsynlighed for forekomst