

1. Home	2
1.1 KV-Redback	4
1.1.1 Project Vision	6
1.1.1.1 General Requirements	7
1.1.1.2 Scope	8
1.1.2 Project Structure	9
1.1.2.1 Elicitation	10
1.1.2.1.1 Motivational Modelling	11
1.1.2.1.2 Personas	12
1.1.2.2 Architecture	13
1.1.2.2.1 Development View	14
1.1.2.2.2 Logical View	16
1.1.2.2.3 Physical View	23
1.1.2.3 User Requirements	25
1.1.2.3.1 Epic 1 - A production ready system	26
1.1.2.3.2 Epic 2 - A secure system	31
1.1.2.3.3 Product backlog	35
1.1.2.3.4 Prototypes	36
1.1.3 Project Plan	37
1.1.3.1 Sprint 2 Plan	38
1.1.3.2 Sprint 3 Plan	42
1.1.3.3 Software Quality Assurance Plan	45
1.1.3.4 Retrospectives	47
1.1.3.4.1 Sprint 1 Retrospective	48
1.1.3.4.2 Sprint 2 Retrospective	49
1.1.3.4.3 Sprint 3 Retrospective	50
1.1.4 Quality Management	51
1.1.4.1 Coding Standards	53
1.1.4.2 Version Control	56
1.1.4.3 Testing	58
1.1.4.3.1 Acceptance Testing	59
1.1.4.3.2 Unit Testing	64
1.1.4.4 Code Review Checklist	85
1.1.4.4.1 Code Review for Sprint 2	87
1.1.4.4.2 Code Review for Sprint 3	88
1.1.5 Development	89
1.1.5.1 Development Tools	90
1.1.6 Decisions	91
1.1.6.1 Two Team Cooperation Method	92
1.1.6.2 Architecture Decisions	93
1.1.7 Presentation	94
1.1.8 Draft	95
1.1.8.1 Digital Prototype	96
1.1.8.2 Requirements Elicitation (UI)	100
1.1.8.3 Testing Plan	102
1.1.8.4 Research	106

Home

Project Overview

Name: Software application for the ACDS kV Level 1b audit program (code: KV)

Background:

The ACDS conducts dosimetry audits at clinics across Australia and New Zealand. Each audit generates some data which is currently stored in two working Excel spreadsheets. One spreadsheet is used to collect data during the audit at a specific clinic. Once the audit is complete, this data is added to the second spreadsheet so that all the data from all past audits are in one place. The aggregated data is used for further comparative analysis and calculating results (graphs etc). We would like both spreadsheets to be replaced with one application – preferably with some sort of GUI so that we can use it easily.



Our Client

Dr. Sabeena Beveridge (Research ROMP)

Australian Radiation Protection and Nuclear Safety Agency

Australian Clinical Dosimetry Service

School of Computing and Information Systems

The University of Melbourne

COMP 90082 Software Project

Road Map

Sprint 1 - Inception	March 27, 2022
Sprint 2 - Development	May 1, 2022
Sprint 3 - Development	May 29, 2022
Sprint 4 - Product	June 12, 2022

Quick Links

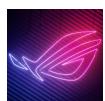
- [Slack Chanel](#) - The channel to contact with client in a quick way.

Project Team

Project Staff				
Subject Coordinator	Project Supervisor	Chartered Scientist	Medical Physicist	
				(Placeholder)
Dr. Eduardo Oliveira	Lin Li	Dr. Sabeena Beveridge	Mr. Fayz Kadeer	(Placeholder)
KV-Redback				
Product Owner	Scrum Master	Frontend lead	Backend lead	(Placeholder)

				
Xinyu Zeng	Caleb Wang	Fangtai Dong	Yipeng Li	
Product Owner	Scrum Master	Tech Lead		
				
Ye Yuan	Zihao Zhang	Yahui Wen	Yangpeng Lai	

Recent space activity



Fangtai DONG

Motivational Modelling updated 2 minutes ago • [view change](#)



Xinyu Zeng

Deployment Plan updated 7 minutes ago • [view change](#)

Epic 1 - A production ready system updated 8 minutes ago • [view change](#)

Epic 2 - A secure system updated 8 minutes ago • [view change](#)

Product backlog updated 9 minutes ago • [view change](#)

Space contributors

- Fangtai DONG (2 minutes ago)
- Xinyu Zeng (7 minutes ago)
- Haoran Zhao (13 minutes ago)
- CALEB Wang (17 minutes ago)
- Ye Yuan (58 days ago)
- ...

KV-Redback

About Team

We are agile team with five member to develop the project of Software application for the ACDS.

Dev Tool



Project Status

SPRINT 1	Inception on March 27, 2022
SPRINT 2	Development on May 1, 2022
SPRINT 3	Development on May 29, 2022
SPRINT 4	Product on June 12, 2022

Role and Responsibilities

Title	Role	Responsibilities
Product Owner  Xinyu Zeng xinyu. zeng1@student. unimelb.edu.au	The product owner ensures the team delivers the most value to the client. The product owner have a strong understanding of the project, which is used to prioritise user stories in upcoming sprints.	<ul style="list-style-type: none">Maintain close ties with the client and seek their validation and input.Communicate with the client - all emails, meeting invites, etc.Guide the team to reach the client.Set up meeting schedule.
Scrum Master  CALEB Wang caleb. wang@student. unimelb.edu.au	The scrum master ensures the Scrum framework is followed. The scrum master coaches the team rather than leading them - scrum masters are servant leaders.	<ul style="list-style-type: none">To ensure that they oversee how the project is tracking, and work to unblock developers to continue meeting objectives.
Frontend lead  Fangtai DONG fangtai. dong@student. unimelb.edu.au	The frontend leader oversees the frontend development of the project. The frontend leader plays a key role in any decision-making about frontend architecture.	<ul style="list-style-type: none">Set up and configure the frontend codebase.
Backend lead  Yipeng Li yipeng.li@student. unimelb.edu.au	The backend leader oversees the backend development of the project. The backend leader plays a key role in any decision-making about backend architecture.	<ul style="list-style-type: none">Set up and configure the backend codebase.

Quality Assurance  Haoran Zhao haoran. zha01@student. unimelb.edu.au	The quality assurance monitors testing other initiatives to ensure the system is built to a high quality.	<ul style="list-style-type: none"> • Create a testing plan and ensure the testing objectives are met.
--	---	--

Blog stream



| KV-Redback Team delivered in Sprint 1 !

Xinyu Zeng posted on Mar 29, 2022

Our team finished the Sprint 1 - design part and here is the checklist as following! Sprint 1 Assessment Checklist This checklist helps you double check your work for Sprint 1. Background description, client goals, motivation [x] Project overview, background and goals were created. [x] DO-BE-FEEL list and GOAL MODEL were created. [x] The goal model is consistent with the client understanding of the problem and with DO-BE-FEEL list....

[Edit](#) ·

- [redback](#)

Recently updated

[S3] 05/18 Supervisor Meeting

25 May, 2022 • created by Haoran Zhao

[S3] 05/18 Team Stand up Meeting

25 May, 2022 • created by Haoran Zhao

[S1] 03/24 Meeting notes with Sabeena

17 May, 2022 • updated by Haoran Zhao • view change

[S1] 03/24 Tech Question with Software Specialist

17 May, 2022 • updated by Haoran Zhao • view change

[S3] 05/11 Supervisor Meeting

17 May, 2022 • updated by Haoran Zhao • view change

[S3] 05/04 Supervisor Meeting

17 May, 2022 • updated by Haoran Zhao • view change

[S3] 05/04 Team Stand up Meeting

17 May, 2022 • updated by Haoran Zhao • view change

[S3] 05/11 Team Stand up Meeting

17 May, 2022 • updated by Haoran Zhao • view change

[S1] 03/24 Meeting notes with Sabeena

13 May, 2022 • updated by Xinyu Zeng • view change

[S1] 03/24 Tech Question with Software Specialist

13 May, 2022 • updated by Xinyu Zeng • view change

| Project Vision

Project Overview

Background

This project is in collaboration with the **Australian Clinical Dosimetry Service (ACDS)**, an organization responsible for checking, testing, and verifying ionizing radiation therapy and treatment machines across Australia and New Zealand.

The ACDS conducts audit on treatment machines to check that the radiation dose patients receive for their cancer treatment is within an acceptable margin. They provide clinics with an official report whether they can continue to use the treatment machines to treat patients.

This project will be dealing with audits on low-energy ionizing machines (VIOLET – Voltage Integration Over Low Energy Therapy) that are used to treat mainly skin cancers, see their [website](#) for more details. Each audit generates some data which is currently stored in two working Excel spreadsheets. One spreadsheet is used to collect data during the audit at a specific clinic. Once the audit is complete, this data is added to the second spreadsheet so that all the data from all past audits are in one place. The aggregated data is used for further comparative analysis and calculating results.

Goal

The goal of this project is to create and design a software application, with a database, to manage and calculate the physics data obtained for KV radiotherapy audits performed by the Australian Clinical Dosimetry Service (ACDS) i.e., to convert the Excel spreadsheet into a software application and database where the inputs from the facilities are stored securely on a database, the background calculations and interpolations that are essential for the audit inputs are “hidden” from the user, and a user interface allows the user to input measurements and record experimental data during audits. With this software application, audit information can be more efficiently managed, and with the audit results stored in one place, i.e., the database server, analysis can be performed more effectively, potentially increasing productivity.

Reference from previous project VIOLET.

Quick Link

- [General Requirements](#)
- [Scope](#)

General Requirements

- [Nk calculation] Based on the information provided by the clinic, the Nk value under different Chambers is calculated and stored in the database.
- [Mu/rho calculation] Using the Mu/rho values from the lookup table and the given measurements, interpolate the measured Mu/rho value.
- [Bw calculation] With the given measurements and the 3-D lookup table of Bw, do multiple interpolations for both Aluminium and Copper HVL and obtain the average of Bw, if exists.
- [Closed Cone Corr calculation] For closed Cones, the correction factor will be calculated automatically after the Bw values are generated.
- [Pstem] Users can have the Pstem correction factor when a Plane-Parallel Chamber is used for the measurements.
- [Notification] The user is able to receive notifications about the calculations being performed in the background, for example, when the extrapolation happens.
- [Form Injection] Based on the information provided by the clinic, the audit can input, edit and delete data from page rather than Excel.
- [Security Purpose] role management for different role accessing the different authorized data

Scope

This project focuses primarily on developing a scheduling system to facilitate interactive activities between the moderator and school representatives.

Project Deliverables

The final product of the project should be a web application developed using Vue and Spring Boot. Project deliverables will include

- the progressed versions of Project Management Plan,
- individual reflection from all team members,
- a copy of the source code created during the development process,
- a copy of the final running version of the web application.

Project deliverables will be submitted to the project supervisor at the end of the semester.

Scope

- This project includes the elicitation of requirements from the clients, planning, designing the architecture, implementing, and testing of a system to replace the existing Excel based database of the ACDS allowing them to more efficiently search for and analyze past audits, and keep their audit records safe and secure on a database server instead of having numerous Excel files stored in their machine
- The ACDS staff using this system can choose a pre-audit Excel file collected from the facilities and save the information to the database. Once saved on the database, the auditors can input the measurement and experimental values to the system, and view the result of the calculations with respect to the information saved on the database. The ACDS staff can use the system to search for facility information and past audit results more efficiently than browsing or searching through numerous Excel files.
- All necessary project artefacts, such as the requirements, design and architecture, quality assurance plan, source code, deployment manual, and user manual will be released to the clients for them to be guided on how to deploy, use, extend, and maintain the system.
- Role access management.

Out of Scope

- This project does not include ongoing operations and maintenance of the system after it has been released to the client. The generation of a pdf report and the architectural design to implement the said feature are beyond the scope of this project. In addition, since the development team will not be given access to the internal resources of the client due to their security protocols as part of the government, the deployment of the application on-site will not be covered by this project. As an alternative, a deployment manual containing the step-by-step procedure to install and deploy the application on-site will be provided to the client.

| Project Structure

| Elicitation

- Motivational Modelling
- Personas

| Architecture

- Development View
- Logical View
- Physical View

| User Requirements

- Epic 1 - A production ready system
- Epic 2 - A secure system
- Product backlog
- Prototypes

| Elicitation

The purpose of this section is to understand more about requirements elicitation, the broad spectrum of tasks involved to help teams derive their requirements.

Table of Contents

[Motivational Modelling](#)

[Personas](#)

Motivational Modelling

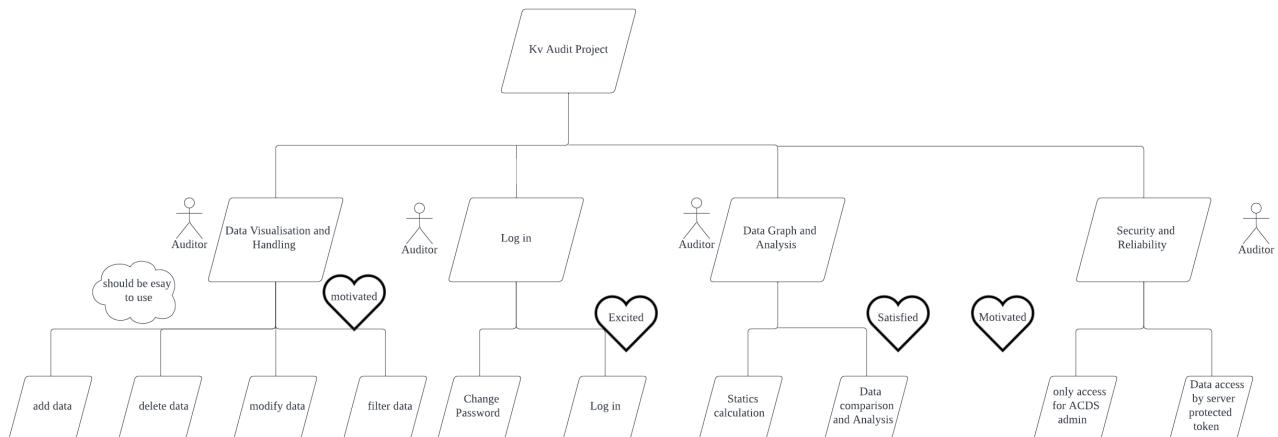
Revision History

Date	Version	Description	Author
2022/04/30	V1.1	Add more details on security and reliability based on client feedback.	Fangtai DONG
2022/03/28	V1.0	Initial Draft	Fangtai DONG

DO/BE/FEEL/WHO List

Do (Function Goal)		Be (Quality Goal)	Feel (Emotional Goal)	Who (Role)
Data Visualisation and Handling	Auditor can use the system to add, delete modify and check clinical dosimetry data.	Easy to use.	Motivated	Auditor
Data Graph and Analysis	Presentation and comparison of analysis results and statistical data.	Computing successfully	Satisfied	Auditor
Data Upload	Generation of final audit result summary report.	Automatically input excel data.	Convenient	Auditor
Summary	Excel Data Uploading and Parsing.	Good to have.	Bonus	Auditor
Business Workflow	Induction page which makes users know more about company auditing process and what can be achieve from the system	Optional.	Fun	Auditor
Report Sending	Sending form and report to specific administrator email address, and good to have functions of review and signature.	Extensive.	Motivated	Auditor

Hierarchical Diagram (Goal Model)



- Lucidchart: https://lucid.app/lucidchart/d1a6c34e-c40b-48bd-a6e5-f0e102d4a193/edit?invitationId=inv_68a830d4-1a43-4216-a19e-1887bda92f39

Personas

Revision History

Date	Version	Description	Author
2022/04/10	V1.1	Delete one persona based on client feedback	Haoran Zhao
2022/03/28	V1.0	Initial Draft	Haoran Zhao

There are one personas for this project based on client requirement.

One is the auditor who is a normal employee of the Clinical Audit Company.

NAME

Eric White



Motivations

Eric is an auditor of a Clinical Audit Company. His responsibility is to audit the clinic. Recently, the company developed a new data system to improve the working efficiency. Eric needs to upload the Excel file of the audit data to the system and get a summary report. Also, Eric can read previous data and reports from the website dashboard so that he can make a comparison.

Goals

- Upload Excel files to the website
- Review previous audit data
- Get a summary report of uploaded data

Frustrations

- The system is lack of responsiveness
- The previous audit data is invisible or incorrect
- No graphs in the summary report

Skills

Data Collection & Analysis

Programming

This persona was built in uxpressia.com

| Architecture

Software architecture is the fundamental organisation of the system under development. It includes all components, their interactions, the environment in which they operate, and the principles used to design the software.

Table of Contents

[Development View](#)

[Logical View](#)

[Physical View](#)

Development View

This view is represented by the package diagram and illustrates a system from a programmer's perspective and is concerned with software management. To demonstrate the development view, our team described the architectural goals and constraints, as well as system diagrams, and API descriptions.

- Architectural goals and constraints
- API references
- System diagram

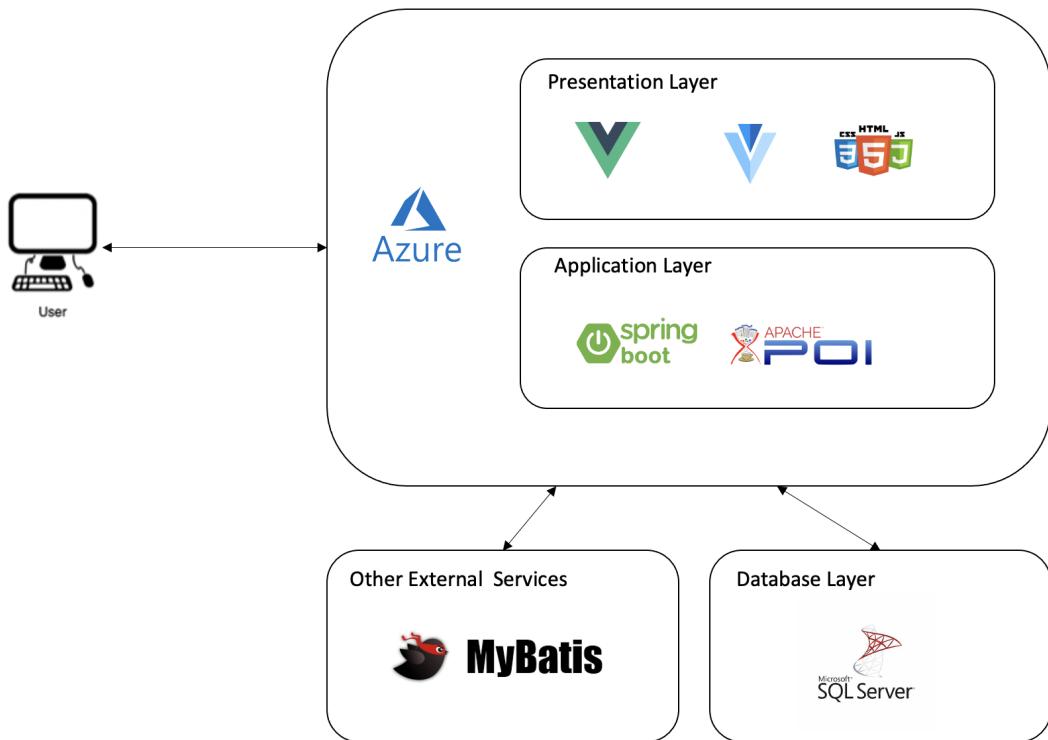
Architectural goals and constraints

Requirement	What	Why	How
Authentication	The system should be able to verify the user's identity.	To protect data from being read or modified by unauthorized users.	<ul style="list-style-type: none">• Login is required before users can access any data.
Confidentiality	The system should ensure that only authenticated people can access sensitive information.	To protect data from malicious use, and to allow users to upload data to the system with confidence.	<ul style="list-style-type: none">• Use HTTPS to encrypt data transmission on web pages.• Use SSL for encryption in database connections.• Use password hash (e.g., password SALT) and/or JWT for user identification and authentication.• Implement user privilege for function and data access control.
Data persistence	The system should ensure that data is persisted and should not be lost even if the server is restarted.	To prevent loss of sensitive data.	<ul style="list-style-type: none">• Use a database to store data (Microsoft SQL Server).
Extensibility	The system should support future expansion and addition of new features.	To allow future changes in business requirements and further development.	<ul style="list-style-type: none">• There should be low coupling between system components.• Front-end, back-end and database should all be able to run independently.

API references

- [Spring boot documentation](#)
- [Microsoft SQL server 2019 documentation](#)
- [Vuety Documentation](#)
- [Vue3 Documentation](#)
- [MyBatis Documentation](#)
- [Apache POI Documentation](#)

System diagram



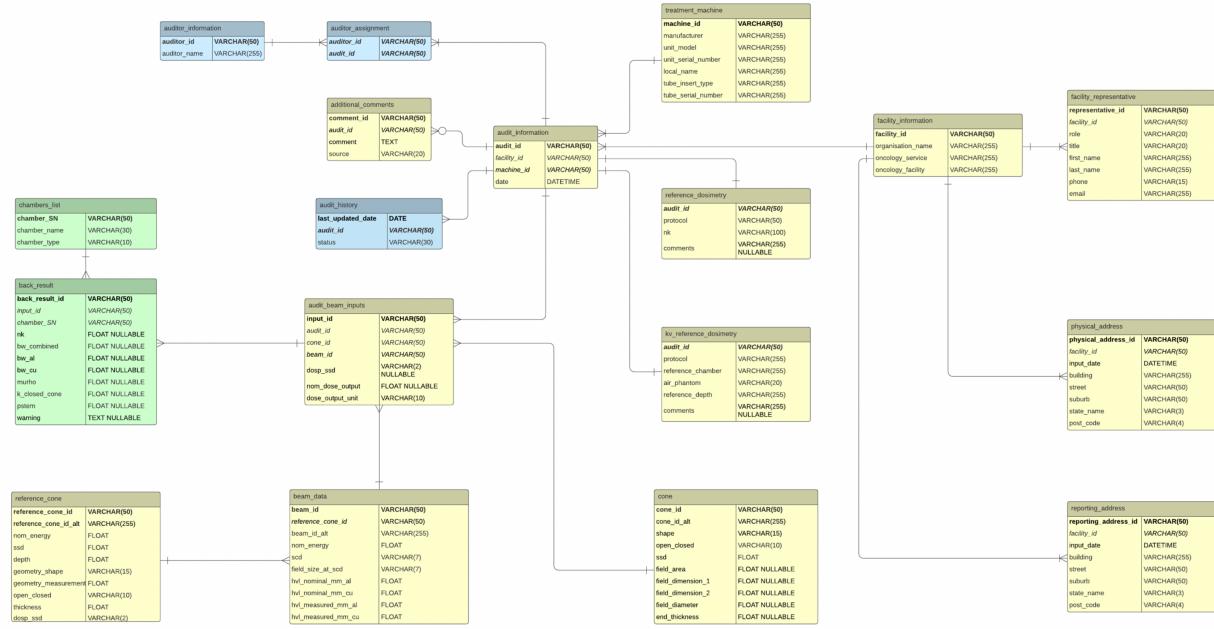
Logical View

Database Model

Entity-Relationship Diagram

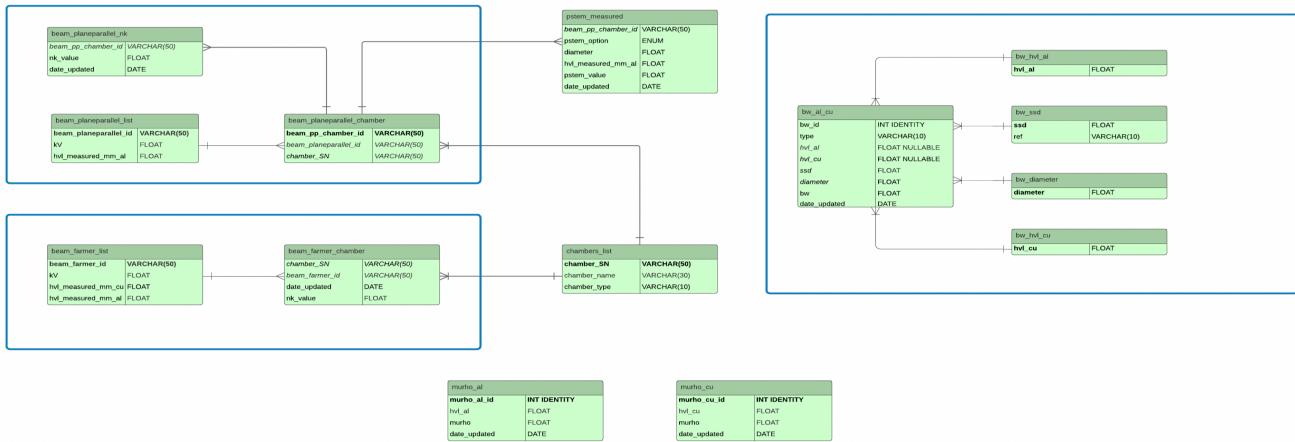
I. `Identification` Database Design

This is the design of the main database that will contain all the information collected from the facility in the 'Identification' sheet.



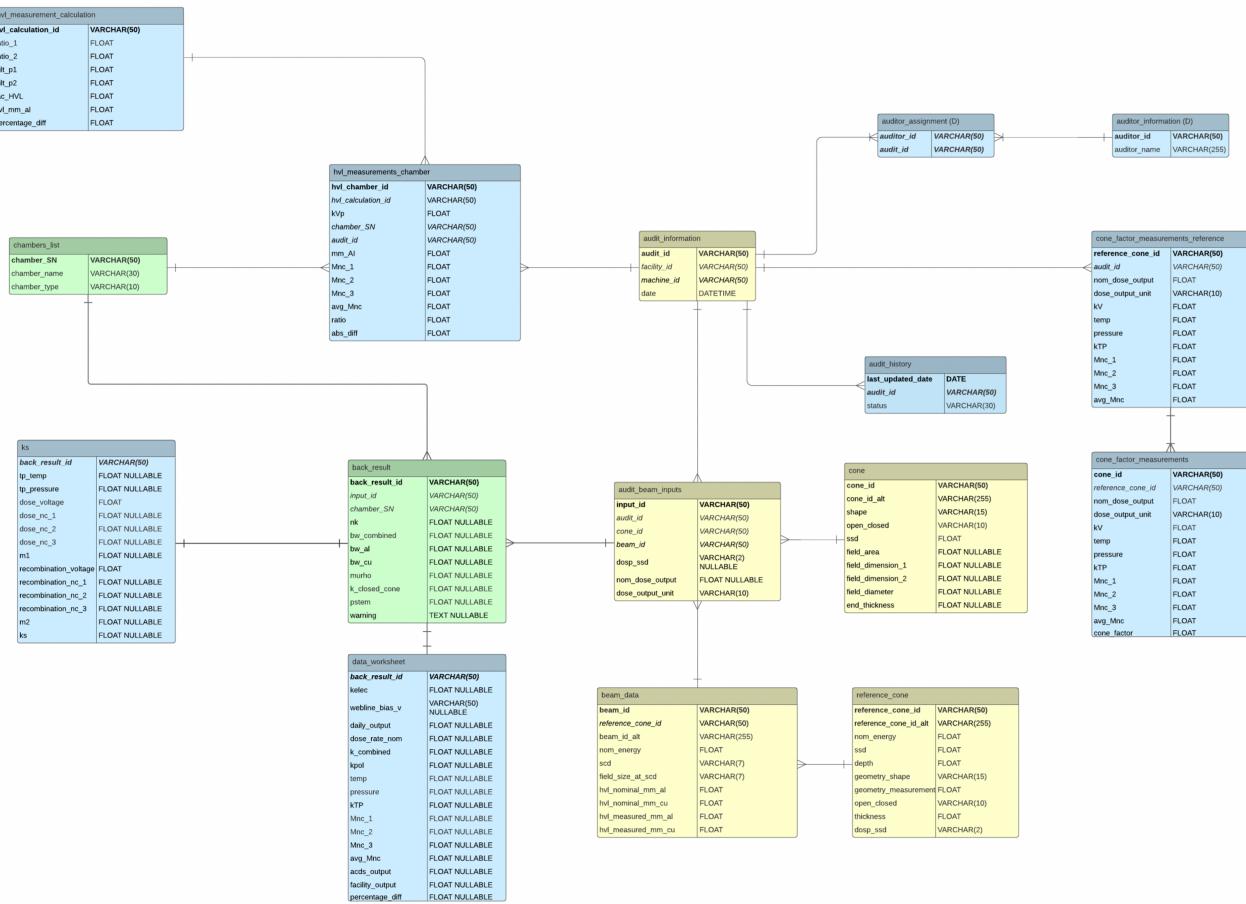
II. Lookup Tables Database Design

This is the design of the database for the lookup tables that will be used by the backend to perform the calculations.



III. Audit results Database Design

This is the initial design of the database for the audit results calculated from the Ks, Data Worksheet, HVL, and Cone Factors tab. This design requires more analysis and refinement to satisfy the requirements to save the audit results before implementing an API to integrate these tables to the frontend of the web application.



Data Dictionary

The tables, fields belonging to each table, their data type, nullable property, and brief descriptions of each field are listed on the following dictionary. This dictionary is designed to be a supplement document for the above Entity-Relationship diagrams. Along with the ERD, this can be used as reference for accessing the database.

Table name	Keys	Field name	Datatype	Mandatory	Description
additional_comments	PK	comment_id	varchar(50)	YES	ID of the comments provided by the facility.
additional_comments	FK	audit_id	varchar(50)	YES	Reference to the ACDS audit information.
additional_comments		comment	text (2147483647)	YES	Comment provided by the facility.
additional_comments		source	varchar(20)	YES	The source table of the comment.
audit_beam_inputs	PK	input_id	varchar(50)	YES	ID of the audit inputs.
audit_beam_inputs	FK	audit_id	varchar(50)	YES	Reference to the ACDS audit information.
audit_beam_inputs	FK	cone_id	varchar(50)	YES	Reference to the cones subject to audit.
audit_beam_inputs	FK	beam_id	varchar(50)	YES	Reference to the beam data.
audit_beam_inputs		dosp_ssd	varchar(2)	NO	Dosage output specification point at SSD, i.e., the point at which the output is defined.
audit_beam_inputs		nom_dose_output	float(53)	NO	Nominal dose output.
audit_beam_inputs		dose_output_unit	varchar(10)	NO	Nominal doses output unit used by the facility.
audit_history	FK,PK	audit_id	varchar(50)	YES	Reference to the ACDS audit information.
audit_history		status	varchar(30)	YES	Status of the audit case (Finished/Progressing).
audit_history		last_updated_date	date()	NO	Date the status of the audit case is last updated.
audit_information	PK	audit_id	varchar(50)	YES	ACDS Audit Number having the format: ACDS-kV-5XXX.

audit_information	FK	facility_id	varchar(50)	YES	Reference to the facility information.
audit_information	FK	machine_id	varchar(50)	YES	Reference to the treatment machine information.
audit_information		date	date(0)	YES	Date the ACDS audit case is published.
auditor_assignment	FK	auditor_id	varchar(50)	YES	Reference to the auditor information.
auditor_assignment	FK	audit_id	varchar(50)	YES	Reference to the ACDS audit information.
auditor_information	PK	auditor_id	varchar(50)	YES	ID of the auditor-in-charge.
auditor_information		auditor_name	varchar(255)	YES	Name of the auditor-in-charge.
back_result	PK	back_result_id	varchar(50)	YES	ID of the results calculated in the background.
back_result	FK	input_id	varchar(50)	YES	Reference linking the results to the audit beam inputs.
back_result	FK	chamber_SN	varchar(50)	YES	Reference to the chamber serial number used in the audit.
back_result		nk	float(53)	NO	Interpolated Nk value.
back_result		bw_combined	float(53)	NO	Combined backscatter factor (average of Bw Al and Bw Cu if both values exist) value.
back_result		bw_al	float(53)	NO	Interpolated Bw Al value.
back_result		bw_cu	float(53)	NO	Interpolated Bw Cu value.
back_result		murho	float(53)	NO	Combined mass energy absorption coefficient (average of mu/rho Al and mu/rho Cu if both values exist) value.
back_result		k_closed_cone	float(53)	NO	Closed cone correction (k_closed cone) value.
back_result		pstem	float(53)	NO	Correction factor (Pstem) value for a Plane-parallel chamber.
back_result		warning	text (2147483647)	NO	Optional warning message.
beam_data	PK	beam_id	varchar(50)	YES	ID of the input beam data.
beam_data	FK	reference_cone_id	varchar(50)	YES	Reference to the reference cone dosimetry conditions informationd.
beam_data		beam_id_alt	varchar(255)	YES	Beam ID used by the facility in the `Identification` sheet.
beam_data		nom_energy	float(53)	YES	Nominal beam energy (kVp).
beam_data		scd	varchar(7)	YES	Souce-to-chamber distance (cm) used for HVL measurement.
beam_data		field_size_at_scd	varchar(7)	YES	Approximate field size (cm) used for HVL measurement.
beam_data		hvl_nominal_mm_al	float(53)	NO	Nominal HVL Al (mm Al) value.
beam_data		hvl_nominal_mm_cu	float(53)	NO	Nominal HVL Cu (mm Cu) value.
beam_data		hvl_measured_mm_al	float(53)	NO	Measured HVL Al (mm Al) value.
beam_data		hvl_measured_mm_cu	float(53)	NO	Measured HVL Cu (mm Cu) value.
beam_farmer_chamber	FK	beam_farmer_id	varchar(50)	NO	Reference to the list of farmer-type beams.
beam_farmer_chamber	FK	chamber_SN	varchar(50)	YES	Farmer-type chamber serial number
beam_farmer_chamber		date_updated	date(0)	YES	Date the farmer-type lookup table is updated.
beam_farmer_chamber		nk_value	float(53)	YES	Nk value corresponding to the Nominal kVp and Measured HVL values.
beam_farmer_list	PK	beam_farmer_id	varchar(50)	YES	ID of the farmer-type beam.
beam_farmer_list		kV	float(53)	YES	Nominal kVp value of the farmer-type beam.
beam_farmer_list		hvl_measured_mm_cu	float(53)	YES	Measured HVL Cu value of the farmer-type beam.
beam_farmer_list		hvl_measured_mm_al	float(53)	YES	Measured HVL Al value of the farmer-type beam.
beam_planeparallel_chamber	PK	beam_pp_chamber_id	varchar(50)	YES	ID of the beam plane-parallel chamber.
beam_planeparallel_chamber	FK	beam_planeparallel_el_id	varchar(50)	YES	Reference to the list of plane-parallel beams.
beam_planeparallel_chamber	FK	chamber_SN	varchar(50)	YES	Plane-parallel chamber serial number.

beam_planeparallel_list	PK	beam_planeparallel_id	varchar(50)	YES	ID of the plane-parallel beam.
beam_planeparallel_list		kV	float(53)	YES	kVp value of the plane-parallel beam.
beam_planeparallel_list		hvl_measured_mm_al	float(53)	YES	HVL Al value of the plane-parallel beam.
beam_planeparallel_nk	FK	beam_pp_chamber_id	varchar(50)	YES	Reference to the beam Plane-parallel chamber list.
beam_planeparallel_nk		nk_value	float(53)	YES	Nk value corresponding to the kVp and HVL values.
beam_planeparallel_nk		date_updated	date(0)	YES	Date the beam Plane-parallel lookup table is updated.
bw_al_cu	PK	bw_id	int(10)	YES	ID of the backscatter factor (Bw).
bw_al_cu		type	varchar(10)	YES	Type of backscatter factor 'al' or 'cu' (Al/Cu).
bw_al_cu	FK	hvl_cu	float(53)	NO	HVL Cu (mm Cu) parameter of the Bw Cu lookup table.
bw_al_cu	FK	hvl_al	float(53)	NO	HVL Al (mm Al) parameter of the Bw Al lookup table.
bw_al_cu	FK	ssd	float(53)	YES	Measured source-to-surface distance (cm).
bw_al_cu	FK	diameter	float(53)	YES	Field size diameter (cm) parameter of the Bw lookup table.
bw_al_cu		bw	float(53)	YES	Bw value corresponding to the HVL, SSD, and diameter parameter values.
bw_al_cu		date_updated	date(0)	YES	Date the Bw lookup table is updated.
bw_diameter	PK	diameter	float(53)	YES	Field size diameter parameter of the Bw Al/Cu lookup table.
bw_hvl_al	PK	hvl_al	float(53)	YES	HVL Al parameter of the Bw Al lookup table.
bw_hvl_cu	PK	hvl_cu	float(53)	YES	HVL Cu parameter of the Bw Cu lookup table.
bw_ssd	PK	ssd	float(53)	YES	Source-to-surface distance parameter of the Bw Al/Cu lookup table.
bw_ssd		ref	varchar(10)	YES	Whether the ssd value is for 'al' only or for 'both' al and cu.
chambers_list	PK	chamber_SN	varchar(50)	YES	Chamber serial number.
chambers_list		chamber_name	varchar(30)	YES	Chamber name.
chambers_list		chamber_type	varchar(10)	YES	Chamber type 'farmer' or 'pp' (Farmer-type/Plane-parallel).
cone	PK	cone_id	varchar(50)	YES	ID of the cone subject to audit.
cone		cone_id_alt	varchar(255)	YES	Cone ID used by the facility in the `Identification` sheet.
cone		shape	varchar(15)	YES	Cone shape (circular/square/rectangular).
cone		open_closed	varchar(10)	YES	Cone type (open/closed).
cone		ssd	float(53)	YES	Measured source-to-surface distance (cm).
cone		field_area	float(53)	NO	Field area (cm ²) of a square cone.
cone		field_dimension_1	float(53)	NO	Field dimension 1 of a rectangular cone.
cone		field_dimension_2	float(53)	NO	Field dimension 2 of a rectangular cone.
cone		field_diameter	float(53)	NO	Field size diameter (cm) of a circular cone.
cone		end_thickness	float(53)	NO	End thickness of a closed cone.
data_worksheet	FK,PK	back_result_id	varchar(50)	YES	Reference to the results calculated in the backend (saved in the back_result table).
data_worksheet		kelec	float(53)	NO	The `kelec` value used in the calculation of ACDS Output.
data_worksheet		webleine_bias_v	varchar(50)	NO	The bias voltage of a chamber.
data_worksheet		daily_output	float(53)	NO	Audit input value to `Daily Output` for the calculation of Dose rate.
data_worksheet		dose_rate_nom	float(53)	NO	Nominal dose output.
data_worksheet		k_combined	float(53)	NO	Combined correction factor = (murho)*Bw*k_closedcone
data_worksheet		kpol	float(53)	NO	Audit input value to `kpol` used in the calculation of ACDS Output.
data_worksheet		temp	float(53)	NO	Audit input temperature (C) value for the calculation of kTP.
data_worksheet		pressure	float(53)	NO	Audit input pressure (kPa) value for the calculation of kTP.
data_worksheet		kTP	float(53)	NO	Calculated kTP value used in the calculation of ACDS Output.
data_worksheet		Mnc_1	float(53)	NO	First M (nC) input.
data_worksheet		Mnc_2	float(53)	NO	Second M (nC) input.

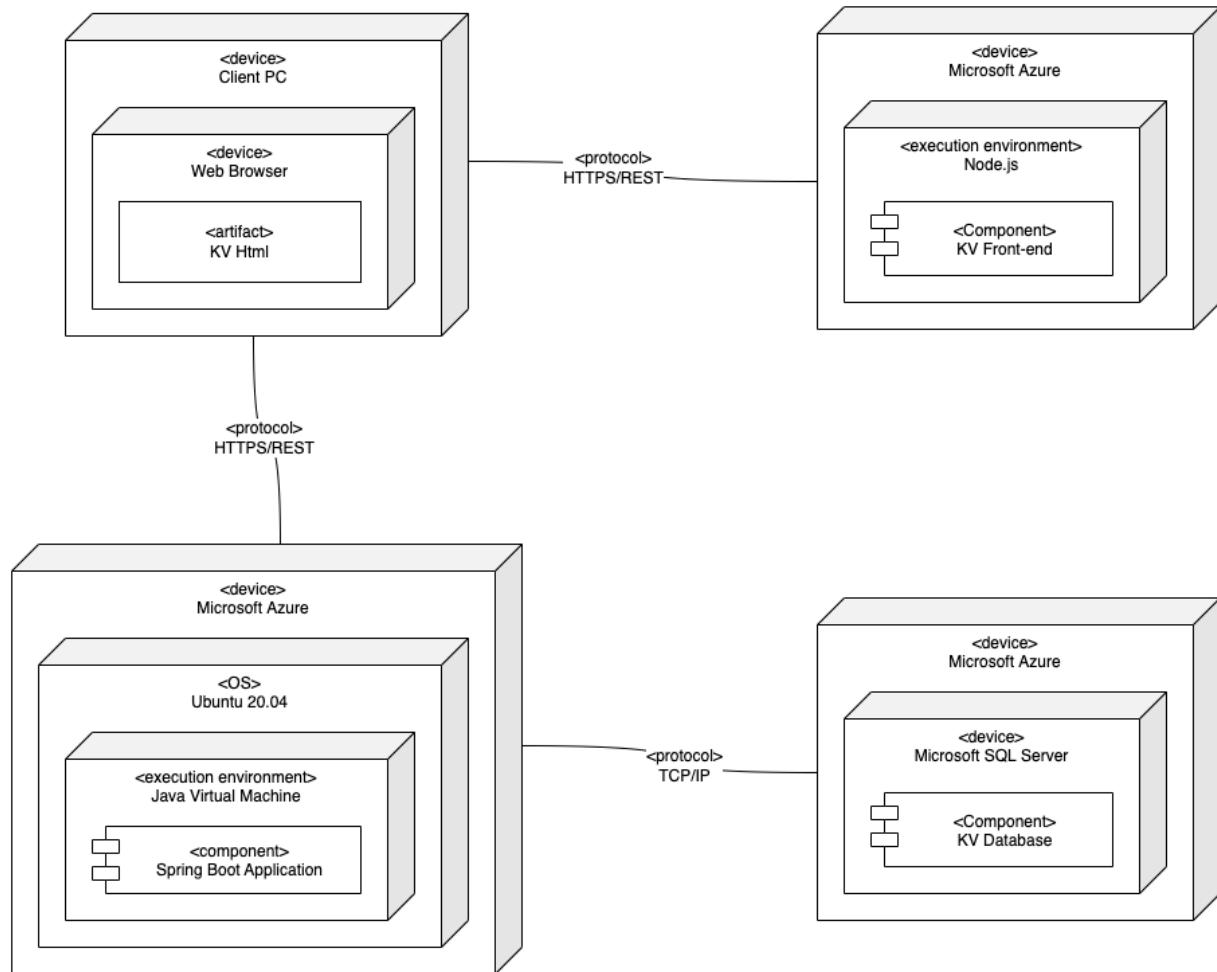
data_worksheet		Mnc_3	float(53)	NO	Third (last) M (nC) input.
data_worksheet		avg_Mnc	float(53)	NO	Average value of the M (nc) inputs.
data_worksheet		acds_output	float(53)	NO	Calculated ACDS Output (cGy/MU).
data_worksheet		facility_output	float(53)	NO	Facility Output (cGy/MU) provided in the Identification sheet.
data_worksheet		percentage_diff	float(53)	NO	Percentage difference between the ACDS and Facility Outputs.
facility_information	PK	facility_id	varchar(50)	YES	ID of the facility subject to audit.
facility_information		organisation_name	varchar(255)	YES	Radiation Oncology Organisation.
facility_information		oncology_service	varchar(255)	YES	Radiation Oncology Service.
facility_information		oncology_facility	varchar(255)	YES	Radiation Oncology Facility.
facilityRepresentative	PK	representative_id	varchar(50)	YES	ID of the facility representative.
facilityRepresentative	FK	facility_id	varchar(50)	YES	Reference to the facility information.
facilityRepresentative		role	varchar(20)	YES	Role of the facility representative.
facilityRepresentative		title	varchar(20)	YES	Title to address the facility representative.
facilityRepresentative		first_name	varchar(255)	YES	First name of the facility representative.
facilityRepresentative		last_name	varchar(255)	YES	Last name of the facility representative.
facilityRepresentative		phone	varchar(15)	YES	Phone number of the facility representative.
facilityRepresentative		email	varchar(255)	YES	Email address of the facility representative.
ks	FK,PK	back_result_id	varchar(50)	YES	Reference to the results calculated in the backend (saved in the back_result table).
ks		tp_temp	float(53)	NO	Input temperature (C) during dose measurement.
ks		tp_pressure	float(53)	NO	Input pressure (kPa) during dose measurement.
ks		dose_voltage	float(53)	YES	Input voltage during dose measurement.
ks		dose_nc_1	float(53)	NO	First M (nC) input during dose measurement.
ks		dose_nc_2	float(53)	NO	Second M (nC) input during dose measurement.
ks		dose_nc_3	float(53)	NO	Third (last) M (nC) input during dose measurement.
ks		m1	float(53)	NO	Average value of the M (nc) inputs during dose measurement.
ks		recombination_voltage	float(53)	YES	Calculated voltage during recombination measurement.
ks		recombination_nc_1	float(53)	NO	First M (nC) input during recombination measurement.
ks		recombination_nc_2	float(53)	NO	Second M (nC) input during recombination measurement
ks		recombination_nc_3	float(53)	NO	Third (last) M (nC) input during recombination measurement
ks		m2	float(53)	NO	Average value of the M (nc) inputs during recombination measurement.
ks		ks	float(53)	NO	Calculated Ks value.
kv_reference_dosimetry	FK,PK	audit_id	varchar(50)	YES	Reference to the ACDS audit information.
kv_reference_dosimetry		protocol	varchar(255)	YES	kV reference dosimetry protocol used for the audit.
kv_reference_dosimetry		reference_chamber	varchar(255)	YES	Reference chamber used in audit.
kv_reference_dosimetry		air_phantom	varchar(20)	YES	in-air or in-phantom.
kv_reference_dosimetry		reference_depth	varchar(255)	YES	Reference depth used in audit.
kv_reference_dosimetry		comments	varchar(255)	NO	Additional information for the kV reference dosimetry used in audit.
murho_al	PK	murho_al_id	int(10)	YES	ID of the mu/rho AI lookup table.
murho_al		hvl_al	float(53)	YES	HVL AI parameter of the mu/rho AI lookup table.
murho_al		murho	float(53)	YES	Mass energy coefficient (mu/rho) corresponding to the HVL AI parameter value.
murho_al		date_updated	date(0)	YES	Date the mu/rho AI lookup table is updated.

murho_cu	PK	murho_cu_id	int(10)	YES	ID of the mu/rho Cu lookup table.
murho_cu		hvl_cu	float(53)	YES	HVL Cu parameter of the mu/rho Cu lookup table.
murho_cu		murho	float(53)	YES	Mass energy coefficient (mu/rho) corresponding to the HVL Cu parameter value.
murho_cu		date_updated	date(0)	YES	Date the mu/rho Cu lookup table is updated.
physical_address	PK	physical_address_id	varchar(50)	YES	ID of the physical addresss information of the facility.
physical_address	FK	facility_id	varchar(50)	YES	Reference to the facility information.
physical_address		input_date	datetime(3)	YES	Date the physical address of the facility is entered into the database.
physical_address		building	varchar(255)	YES	Building location of the facility.
physical_address		street	varchar(50)	YES	Street address of the facility.
physical_address		suburb	varchar(50)	YES	Suburb address of the facility.
physical_address		state_name	varchar(3)	YES	State address of the facility.
physical_address		post_code	varchar(4)	YES	Post code number of the facility.
pstem_measured	FK	beam_pp_chamber_id	varchar(50)	YES	Reference to the beam Plane-parallel chamber list.
pstem_measured		pstem_option	varchar(10)	YES	Pstem type (unity/measured).
pstem_measured		diameter	float(53)	YES	Field size diameter (cm) parameter of the Pstem lookup table.
pstem_measured		hvl_measured_mm_m_al	float(53)	YES	HVL Al value of the plane-parallel beam.
pstem_measured		pstem_value	float(53)	YES	Pstem value corresponding to the beam's HVL, and diameter parameter values.
pstem_measured		date_updated	date(0)	YES	Date the Pstem table is updated.
reference_cone	PK	reference_cone_id	varchar(50)	YES	ID of the reference cone dosimetry condition.
reference_cone		reference_cone_id_alt	varchar(255)	YES	Reference cone dosimetry condition beam ID used by the facility in the 'Identification' sheet.
reference_cone		nom_energy	float(53)	YES	Nominal energy (kVp) produced by the beam.
reference_cone		ssd	float(53)	YES	Measured source-to-surface distance (cm).
reference_cone		depth	float(53)	YES	Measured depth z_ref (cm).
reference_cone		geometry_shape	varchar(15)	YES	Cone geometry shape.
reference_cone		geometry_measurement	float(53)	YES	Cone geometry area (cm/cm2).
reference_cone		open_closed	varchar(10)	YES	Cone type (open/closed).
reference_cone		thickness	float(53)	YES	End thickness of a closed cone (mm).
reference_cone		dosp_ssd	varchar(2)	YES	Dosage output specification point at SSD, i.e., the point at which the output is defined.
reference_dosimetry	FK,PK	audit_id	varchar(50)	YES	Reference to the ACDS audit information.
reference_dosimetry		protocol	varchar(50)	YES	Reference dosimetry protocol used in audit.
reference_dosimetry		nk	varchar(100)	YES	How Nk value is determined.
reference_dosimetry		comments	varchar(255)	NO	Additional information on the reference dosimetry.
reporting_address	PK	reporting_address_id	varchar(50)	YES	ID of the reporting addresss information of the facility.
reporting_address	FK	facility_id	varchar(50)	YES	Reference to the facility information.
reporting_address		input_date	datetime(3)	YES	Date the reporting address of the facility is entered into the database.
reporting_address		building	varchar(255)	YES	Building location of the facility.
reporting_address		street	varchar(50)	YES	Street address of the facility.
reporting_address		suburb	varchar(50)	YES	Suburb address of the facility.
reporting_address		state_name	varchar(3)	YES	State address of the facility.
reporting_address		post_code	varchar(4)	YES	Post code number of the facility.
treatment_machine	PK	machine_id	varchar(50)	YES	ID of the facility's treatment machine subject to audit.
treatment_machine		manufacturer	varchar(255)	YES	Manufacturer of the kV unit.
treatment_machine		unit_model	varchar(255)	YES	Model of the kV unit.

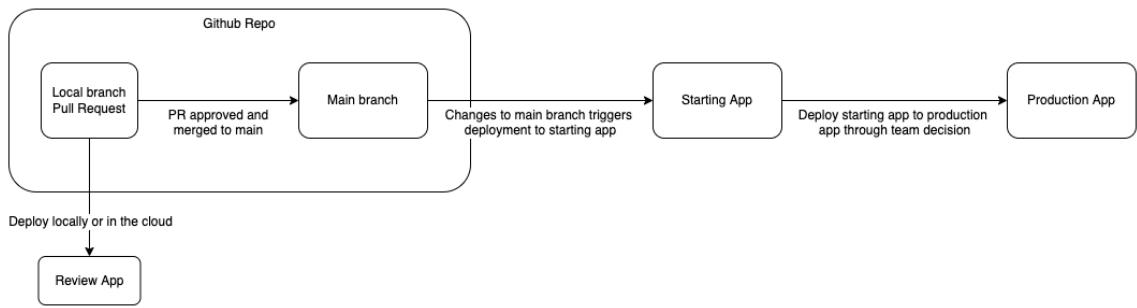
treatment_machine		unit_serial_number	varchar(255)	YES	Serial number of the kV unit.
treatment_machine		local_name	varchar(255)	YES	Local name of the kV unit.
treatment_machine		tube_insert_type	varchar(255)	YES	Tube insert type of the kV unit.
treatment_machine		tube_serial_number	varchar(255)	YES	Serial number of the tube.

Physical View

Deployment Diagram



Development Pipeline



| User Requirements

[Add Product requirements](#)

Title	Designer	Developers	Document owner	Document status	Epic	QA	Target release
Epic 1 - A production ready system	Fangtai DONG	Fangtai DONG CALEB Wang Yipeng Li	Xinyu Zeng	COMPLETED	A production ready system	Haoran Zhao Xinyu Zeng CALEB Wang	1.1
Epic 2 - A secure system	Fangtai DONG	Fangtai DONG CALEB Wang Yipeng Li	Xinyu Zeng	COMPLETED	A secure system	Haoran Zhao Xinyu Zeng CALEB Wang	2
Product backlog	Fangtai DONG	CALEB Wang Fangtai DONG Yipeng Li	Xinyu Zeng	COMPLETED	1 and 2	Haoran Zhao Xinyu Zeng	v1.1

Epic 1 - A production ready system

Versions

VersionID	Description	Date
1.0	Initialise a user story table based on the current understanding of requirements, goal model and persona.	2022-03-21
1.1	Review and edit based on the current situation, and moved item #5 API documentation to #9, so that the ID in this document and Sprint 2 Plan is consistent. A burndown chart analysis is attached to indicate the team's performance in the sprint 2, according to Sprint 2 Plan it is equivalent to the completion of Epic 1. Reviewer: Caleb	2022-04-27

Target release	1.1
Epic	A production ready system
Document status	COMPLETED
Document owner	Xinyu Zeng
Designer	Fangtai DONG
Developers	Fangtai DONG CALEB Wang Yipeng Li
QA	Haoran Zhao Xinyu Zeng CALEB Wang

Goals

- As the epic name suggests, the client want this system to be production ready.
- Fix bugs in frontend, backend, and database if necessary.
- Add new functionality if required.

Background and strategic fit

The previous teams gave the system prototype, and the client was pleased. Although the client also suggests that the system is not good enough for production due to security concerns. Besides that, the client also pointed to improving the system's usability. The team addressed that issue at the client meeting on 17/03/2022 and decided to improve the system's robustness (in the frontend and backend) and fix the bugs.

Please note that the team decided to rewrite the entire project due to the unknown bugs left-over from the previous project. However, we also assumed that some part of the system is reusable (such as the Database). As a result, the balanced workload for a single student team.

Assumptions

- The team was notified by the client during the meeting on 17/03/2022 that the handover project had some unknown left-over bugs. Thus, the team derived user stories from the previous project regarding this matter.
- The team assumed that the database design can fulfill the most of the requirement. Thus, the team does NOT put all DB related user stories in this document, the team **will focus more on the frontend and backend development**.

Requirements

Note that below acceptance criteria unspecified with the test results are all **Accept cases**.

#	Title	User Story	Importance	Acceptance Criteria	Notes

1	Frontend and Backend communication	As an user, I want the front-end website can communicate with the back-end server so that my calculations behind the scenes can be sent to the front-end website.	Must have	<p>Scenario: The front-end program can send the requests to the server while the server can send back the calculation results as responses.</p> <p>Given:</p> <ul style="list-style-type: none"> The user browser prepared a request. <p>When:</p> <ul style="list-style-type: none"> The user browser send a (REST/HTTP) request to the server. <p>Then:</p> <ul style="list-style-type: none"> The server should calculate the result. The result is returned back to the user's browser. <ul style="list-style-type: none"> The left-over backend use cases and objective approved by the client: <ul style="list-style-type: none"> [Nk calculation] Based on the information provided by the clinic, the Nk value under different Chambers is calculated and stored in the database. [Mu/rho calculation] Using the Mu/rho values from the lookup table and the given measurements, interpolate the measured Mu/rho value. [Bw calculation] With the given measurements and the 3-D lookup table of Bw, do multiple interpolations for both Aluminium and Copper HVL and obtain the average of Bw, if exists. [Closed Cone Corr calculation] For closed Cones, the correction factor will be calculated automatically after the Bw values are generated. [Pstem] Users can have the Pstem correction factor when a Plane-Parallel Chamber is used for the measurements. [Notification] The user is able to receive notifications about the calculations being performed in the background, for example, when the extrapolation happens. [Connect with the database] The backend should be able to query the DB or store values in DB whenever needed.
2	Backend and Database communication	As an user, I want the backend server can connect with the database so that my calculation results can be stored in the database for further use.	Must have	<p>Scenario: The calculation results can be stored in the database when the server connects to the database and inserts the data.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. <p>When:</p> <ul style="list-style-type: none"> The server update the data into DB instance. <p>Then:</p> <ul style="list-style-type: none"> The data is immediately available when the update transaction has been completed. <ul style="list-style-type: none"> Note that this use case also applies to the following DB operations: <ul style="list-style-type: none"> WRITE a new record UPDATE a record DELETE a record Also note that we assume the DB has mature and <i>bug-free</i> DBMS that the backend developers should not worry about the DBMS. That is, the group assumed that the ACID property of the DBMS always held.
3	Backend provides previous input	As an user, I want the backend server can query all the required input data from the database so that I don't need to type any additional input as before	Must have	<p>Scenario: The required input data can be queried from the DB when the server is calculating the results.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. The user made inputs for the calculation. <p>When:</p> <ul style="list-style-type: none"> The server starts calculation process and updating the data into DB instance. <p>Then:</p> <ul style="list-style-type: none"> The previous inputs should all become visible on the frontend.

4	Error notification	<p>As an user, I want the backend server can return some important variables or error messages(e.g. out of range/extrapolation) to the frontend website during the calculation so that I can track the calculation process and find the problems easily and be aware of potential sources of discrepancies.</p>	Good to have	<p>Scenario: The user can view the important variables or error messages during the backend calculation.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. The user made inputs for the calculation. <p>When:</p> <ul style="list-style-type: none"> The server starts calculation process and updating the data into DB instance. <p>Then:</p> <ul style="list-style-type: none"> The important variables or error messages should all become visible. 	<ul style="list-style-type: none"> Note: this user story applies to both frontend and backend.
5	Calculation service (backend)	<p>As an user, I want all of the data* is correctly interpolated, calculated and updated, so that I can use them for audit process.</p> <p>*please see the note on the RHS for detail.</p>	Must to have	<p>Scenario: The user wants to upload Bw, Mu/rho, Closed Cone Corr (CCC), Nk, Pstem and gets correct results.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. <p>When:</p> <ul style="list-style-type: none"> The user made inputs for the calculation. The calculation and interpolation involves: <ol style="list-style-type: none"> Bw Mu/rho Closed Cone Corr (CCC) Pstem Nk The input consists of Excel, and/or frontend HTML input. <p>Then:</p> <ul style="list-style-type: none"> The calculation result should be correctly respond to the frontend and stored on the Database. 	<p>Please refer the calculation flowcharts provided by the previous team.</p> <p>The calculation can be explained in 3 phases:</p> <ol style="list-style-type: none"> 1. Interpolation (not visible to the user) 2. Calculation (not visible to the user) 3. Update to database (Retrievable by the backend + frontend)
6	Data visibility	<p>As an user, I want to see all audit cases on the system website.</p>	Must have	<p>Scenario: The user wants to view all the audit cases on the web application.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. <p>When:</p> <ul style="list-style-type: none"> The user is logged in with User account <p>Then:</p> <ul style="list-style-type: none"> All cases with their audit ID, date, clinics and status are visible to the frontend. 	<ul style="list-style-type: none"> From this item, below are all frontend requirements For this item, notice that the audit case refers to the following items: <ol style="list-style-type: none"> Audit date Audit number (ID) Status (Ongoing, Completed, etc) KS table HVL table Data Worksheet Table Cone Factors Table (If applicable)

7	Search data	As an user, I want to search the specific audit case by date, audit number, etc.	Good to have	<p>Scenario: The user wants to view specific audit case(s) on the web application.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. The user is logged in with User account <p>When:</p> <ul style="list-style-type: none"> User type audit ID/date /clinics/status in the search box (or filter tool bar) <p>Then:</p> <ul style="list-style-type: none"> The corresponding audit case(s) will be visible. 	<ul style="list-style-type: none"> In this case, the team assumed equality/partial equality search.
8	Calculation service (frontend)	As an user, I want all of the data* is correctly interpolated, calculated and updated , so that I can use them for audit process. *please see the note on the RHS for detail.	Must have	<p>Scenario: The user wants to upload Bw, Mu/rho, Closed Cone Corr (CCC), Nk, Pstem and gets correct results.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. <p>When:</p> <ul style="list-style-type: none"> The user made inputs for the calculation. <ul style="list-style-type: none"> The input consists of Excel, and/or frontend HTML input. <p>Then:</p> <ul style="list-style-type: none"> The calculation result should be correctly received from the backend and visible on the frontend. 	<p>Please refer the calculation flowcharts provided by the previous team.</p> <p>The calculation can be explained in 3 phases:</p> <ol style="list-style-type: none"> Interpolation (not visible to the user) Calculation (not visible to the user) Update to database (Retrievable by the backend + frontend) <p>Note that this item applies to the frontend, which focus on the user input handling + server respond handling.</p>
9	API	As an user, I want to have the API documentation, so that I can refer to the documentation for any further modification.	Must have	<p>Scenario: The user can have the API documentations after the development.</p> <p>Given:</p> <ul style="list-style-type: none"> The entire software system that's ready to use <p>When:</p> <ul style="list-style-type: none"> The user wants to improve /understand this system in technical perspective. <p>Then:</p> <ul style="list-style-type: none"> The API documentation should be ready for reference by the handover. 	<ul style="list-style-type: none"> A general requirement. This requirement should be fulfilled by Sprint 4 (handover)

User interaction and design

#	User interaction	Design
1	Frontend and Backend communication	User can input the data they want on the interface, frontend collect the data by <form> and transmit them to backend server as POST request for handling.
2	Backend and Database communication	Calculation and handling the users' GET request from interface.
3	Backend provides previous input	Calculation.
4	Error notification	Front-End validation can be fulfilled by <dialog> or alert JavaScript.

5	Data visibility	Table for data visibility and use some RESTful API try to realize the data graph visualization
6	Search and filter	Search bar or filter flag setting.
7	Calculation service (frontend)	CRUD component setting.

Questions

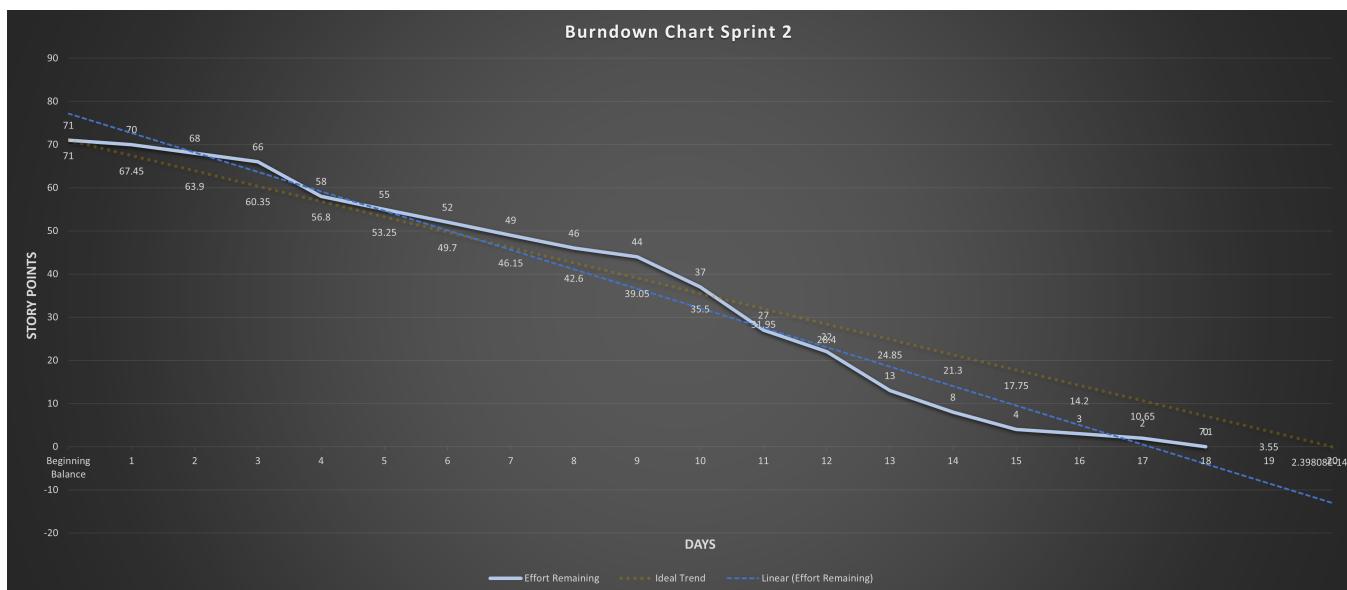
Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome
How users can better use this software	<i>API documentation</i>

Not Doing

- Migrate SQL database to non-SQL database, but it is ok to migrate between SQL databases.
- Deploymnet + Network configuration on ARPANSA.

Effort evaluation and burndown chart



As the team finalized most of the Epic 1 tasks in sprint 2, the burndown chart above indicates the team's performance. The effort before day 10 was a bit underperformed because the team spent time on researching and understanding the codebase provided by the previous team and the client. However, the team overperformed after day 10 because everyone was get used to the development and thus we finalized the Epic 1 earlier than the deadline by 2 days.

Epic 2 - A secure system

Versions

VersionID	Description	Date
1.0	Initialise a user story table based on the current understanding of requirements, goal model and persona.	2022-03-21
1.1	Review and edit some requirements based on the client feedback that this system should be a single user system (the relevant 2-level user stories are removed, and strikethrough towards to some points regarding this functionality), edit the roles based on current situation. Reviewer: Caleb	2022-04-23

Target release	2
Epic	A secure system
Document status	COMPLETED
Document owner	Xinyu Zeng
Designer	Fangtai DONG
Developers	Fangtai DONG CALEB Wang Yipeng Li
QA	Haoran Zhao Xinyu Zeng CALEB Wang

Goals

- The web application should only allow authorized user to access.
- Unauthorized user should only see the login page. Nothing else.
- Only a **single user** is needed in this system.

Background and strategic fit

The previous teams gave the system prototype, and the client was pleased. Although the client also suggests that the system is not good enough for production, especially the security features.

For example, the user can get into the dashboard page by typing the server address (IP/URL) in the web browser.

Assumptions

- The team assumed that "Security" is mainly about authentication from the clients' perspective.
- The team assumed that some standard security practices in software engineering should be commenced even the client did not mention that explicitly during the meeting on 17/03/2022.
- The team assumed that the deployment model was in the public cloud. However, the solution should be flexible in any deployment model.
- Based on the above point, the team further assume that the webserver is reachable for all public. Further communication with the client may change this assumption.

Requirements

Note that below acceptance criteria unspecified with the test results are all **Accept cases**.

#	Title	User Story	Importance	Acceptance criteria	Notes

1	User portal	As a user, I want to have a web portal exclusively for me to use so that I can examine and edit any database content.	Must have	<p>Given:</p> <ul style="list-style-type: none"> The user is located at Log in page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input username, password, then hit log in button. <p>Then:</p> <ul style="list-style-type: none"> The user will be redirected to the Audit cases page. <p>(Reject case)</p> <p>Given:</p> <ul style="list-style-type: none"> The user is located at Log in page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input INCORRECT username, password, then hit log in button. <p>Then:</p> <ul style="list-style-type: none"> The user will stay at Log in page with error messages. 	
2	User password change	As a user, I want to be able to change the password so that I can use my own secret to access the system.	Must have	<p>Given:</p> <ul style="list-style-type: none"> The user is located at Audit cases/Lookup page, and accessed the user panel pop up. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input valid password, then hit change password button. <p>Then:</p> <ul style="list-style-type: none"> The user will be informed that the change password operation succeeded. Then the user will be redirected to the Log in page. <p>(Reject case)</p> <p>Given:</p> <ul style="list-style-type: none"> The user is located at Audit cases/Lookup page, and accessed the user panel pop up. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input invalid password, then hit change password button. <p>Then:</p> <ul style="list-style-type: none"> The user will be informed that the change password operation failed. 	The client stated that this item should be the highest priority because it is a MUST for Government agency that change their password for a set of time period (~6 months)

3	User log out	As a user, I want to be able to log out the website so that nobody can see my account on my machine	Must have	<p>Given:</p> <ul style="list-style-type: none"> The user is logged in with User account. The user is located at Audit cases/Lookup page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input hit log out button. <p>Then:</p> <ul style="list-style-type: none"> The user will be redirected to the Log in page. 	
4	Offline running	As a user, I want to be able to use the web server in my local machine so that the security of the system is maximized.	Good to have	<p>Given:</p> <ul style="list-style-type: none"> A completed solution (backend, frontend, database) All solutions can run standalone as a Docker service. <p>When:</p> <ul style="list-style-type: none"> Client access the web server in the local machine. <p>Then:</p> <ul style="list-style-type: none"> Every functionality should be available in the web server. 	<p>Note that this item could incur some syncing problems, and the team assumed that the offline version would never communicate to the public cloud instance.</p> <p>The syncing would be done by the client side.</p>
5	Functions used by authenticated person	As a user, I want to make sure that all services and functionality provided by the web server is serving the authenticated individual/entity, so that the system is secure.	Must have	<p>Given:</p> <ul style="list-style-type: none"> The user is logged in with User account. The user is located at Audit cases/Lookup page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user made any input in identification /lookup/change password function. <p>Then:</p> <ul style="list-style-type: none"> The user will be prompted/redirected accordingly. (The functions are served normally) <p>(Reject case)</p> <p>Given:</p> <ul style="list-style-type: none"> The user is logged in with User account. The user is located at Audit cases/Lookup page. A good and healthy web server. An expired user token. <p>When:</p> <ul style="list-style-type: none"> The user made any input in identification /lookup/change password function. <p>Then:</p> <ul style="list-style-type: none"> The user will be prompted/redirected to the log in page. (The user input will never works) 	

User interaction and design

#	User interaction	Design
1	Login access	login page – The data entered by the user can be compared with the credentials on the database.
2	Logout	log out button – let users/ jump out from dashboard to login page.
3	Change password	any page – let user change password, and then will be redirected to login page if successful.

Questions

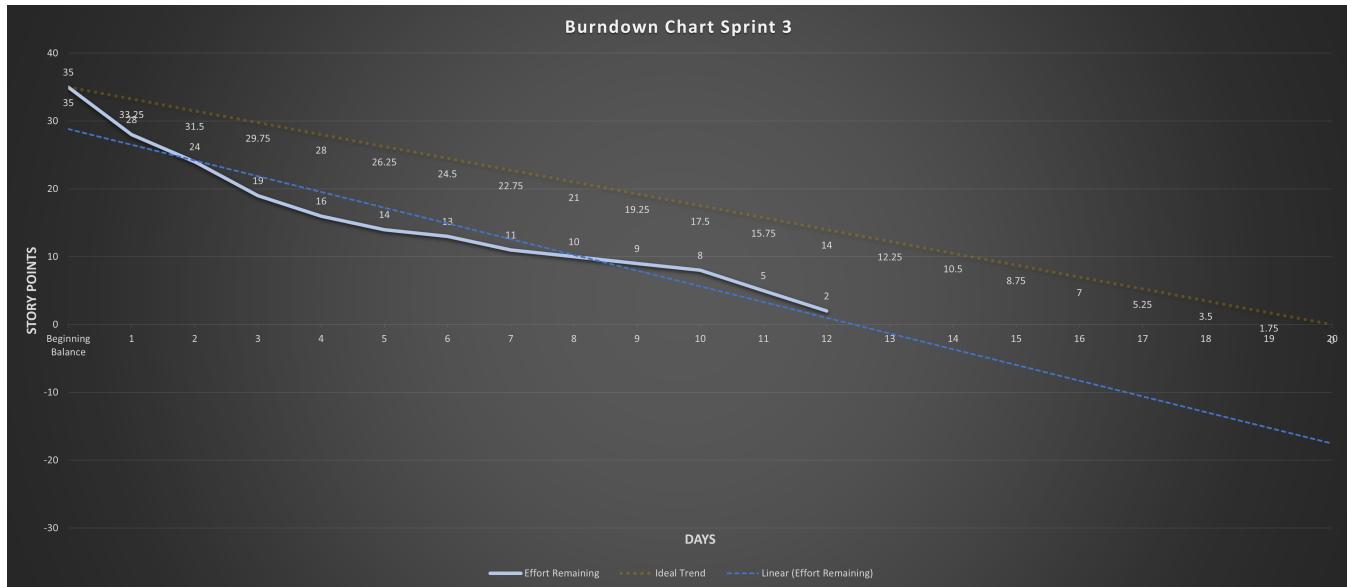
Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome
How to make sure that only one single user is using the product	Use of single account, and only one active key.

Not Doing

- The team is not responsible for managing the data integrity of both offline/online solutions. (if offline solution is applicable).

Effort evaluation and burndown chart



As the team finalized all of the Epic 2 tasks in sprint 3, the burndown chart above indicates the team's performance. The team overperformed during the development and all user stories were completed at day 12, while it is significant that keep workload along with the ideal trend, but the team spent more time on verification and debugging for better quality.

Product backlog

Date	Version ID	Description
21-03-2022	v1.0	Document initialization
01-05-2022	v1.1	Review and edit based on current situation and client feedback. Reviewer: Caleb
Target release	v1.1	
Epic	1 and 2	
Document status	COMPLETED	
Document owner	Xinyu Zeng	
Designer	Fangtai DONG	
Developers	CALEB Wang Fangtai DONG Yipeng Li	
QA	Haoran Zhao Xinyu Zeng	

Background and strategic fit

Declare the functionality of the project. Note that this document defines the basic requirement of the system, and will be defined in lower granularity in the epic documents: [Epic 1 - A production ready system](#) and [Epic 2 - A secure system](#).

Assumptions

- Migrate SQL database to non-SQL database, but it is ok to migrate between SQL databases.
- Deployments + Network configuration on ARPANSA.
- The team is not responsible for managing the data integrity of both offline/online solutions. (if offline solution is applicable).

Requirements

#	Title	Importance	Notes
1	User portal	Must have	<ul style="list-style-type: none"> • Including user login/logout/password change, and Jwt mechanism.
2	Offline running	Good to have	
3	Calculation	Must have	<p>The calculation and interpolation involves:</p> <ol style="list-style-type: none"> 1. Bw 2. Mu/rho 3. Closed Cone Corr (CCC) 4. Pstem 5. Nk <p>Also note that the calculation is mainly about interpolation and extrapolation.</p>
4	Excel processing	Must have	Process lookup/identification excels on the backend.
5	Data visibility	Must have	Frontend process/represents all audit cases, calculations, and accepting new user inputs

User interaction and design

please refer to [Architecture](#) and [User Requirements](#)

Prototypes

Revision History

Date	Version	Description	Author
2022/03/28	V1.0	Initial Draft	Fangtai DONG
2022/04/10	V1.1	Add user logic.	Fangtai DONG

See more details on [here](#).

| Project Plan

Development Process

- Standups - The team standups will be conducted over Zoom for 2~3 times a week. The meeting time may be vary.
- Sprint planning - Before each sprint, we will decide how much work we can do during the sprint, and specify the goal of the sprint.
- There are 2 sprints for project development. Each sprint lasting for 4 weeks, and for the first sprint, we plan to complete as much of the critical features in [Epic 1 - A production ready system](#) as possible. Other lower priority major and minor features are planned to be completed on the second sprint. After each sprint, we will check our progress, measure the team's actual velocity, and evaluate if the remaining user stories in [Epic 2 - A secure system](#) in the sprint backlog can be completed on the second sprint. We will also check in with the client to confirm if there are changes in the priorities of their requirements.

Development Status

For updates on the development status, refer to the [Trello board](#) which includes:

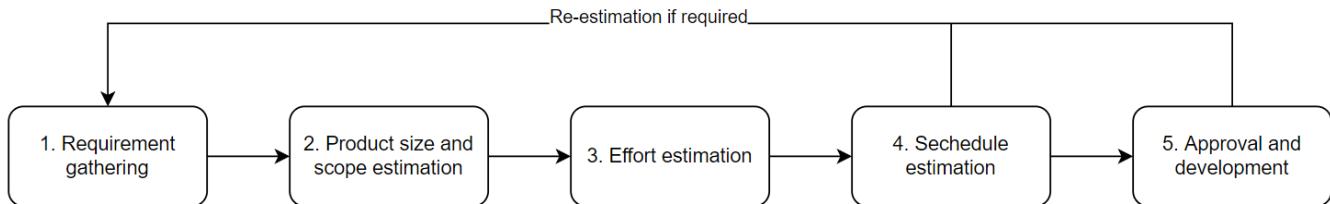
- Product backlog
- Sprint backlog, doing, done tasks

Estimation process

The project estimation could be summarized into 5 steps as the figure below:

1. **Requirement gathering:** Gathers requirement via client meetings.
2. **Product size and scope estimation:** After requirement gathering, the high level product backlog is generated. In this case, the backlog should be verified and approved by the client.
3. **Effort estimation:** The effort estimation comes from the break-down product backlog, and will be used for later sprint backlog. The user stories will be estimated by:
 - a. Task dependency: the more dependencies the task have, the higher effort is required (because higher cost of testing, cost of refactoring the code, etc).
 - b. Task difficulty: if the task requires a combination of 3rd party library, or the task is complex in terms of code length. Then it is a difficult task and deserve more effort.
 - c. Test difficulty: if the task requires a series of testing cases, then the task requires more effort because it is harder to implement.
 - d. Story Point estimation: based on above 3 points, if all 3 points are valid for a task, then it is considered to have ~21 story points. The rest story points estimation will be depending on the developer's estimation.
4. **Schedule estimation:** The schedule in this project is mainly about 2 development sprints, where the highest priority tasks will be scheduled as early as possible. So that the second development sprint (sprint 3) could help the team to do the clean up and catch up if necessary. Note that there are 4 levels of priority used in this project:
 - a. Must have
 - b. Should have
 - c. Could have (not used)
 - d. Won't have (not used)
5. **Approval and development:** Once the team works out the effort and priority, the epic documents and sprint plan will be handover to the client for verification and approval. Once approved, the development starts.

6. **Note:** If necessary, the re-estimation could also be available during schedule or development stage.



Sprint 2 Plan

VersionID	Description	Date
1.0	Document initialization.	2022-03-21
1.1	Document review based on current situation. The task 1 was incorrectly estimated and it should be 21 because of the task/user story dependency, SP for task 3 is increased to 13 due to complexity. Reviewer: Caleb	2022-04-27

Sprint 2 Goal

- Complete ALL user stories in [Epic 1 - A production ready system](#).
- The sign of the completion of Epic 1 represents all leftover bugs and logic errors are fixed/mitigated, including:
 - Performs all types of calculations on the backend.
 - A complete communication between frontend and backend as well as the database.

Planned user stories and acceptance tests

Note that all the user stories below refer to [Epic 1 - A production ready system](#), the story points are estimated along the Fibonacci series (1, 2, 3, 5, 8, 13, 21...)

Also note that this sprint is focus on the system functionality development without implementing user account, so that **all acceptance test result items would be indicated as Accept**.

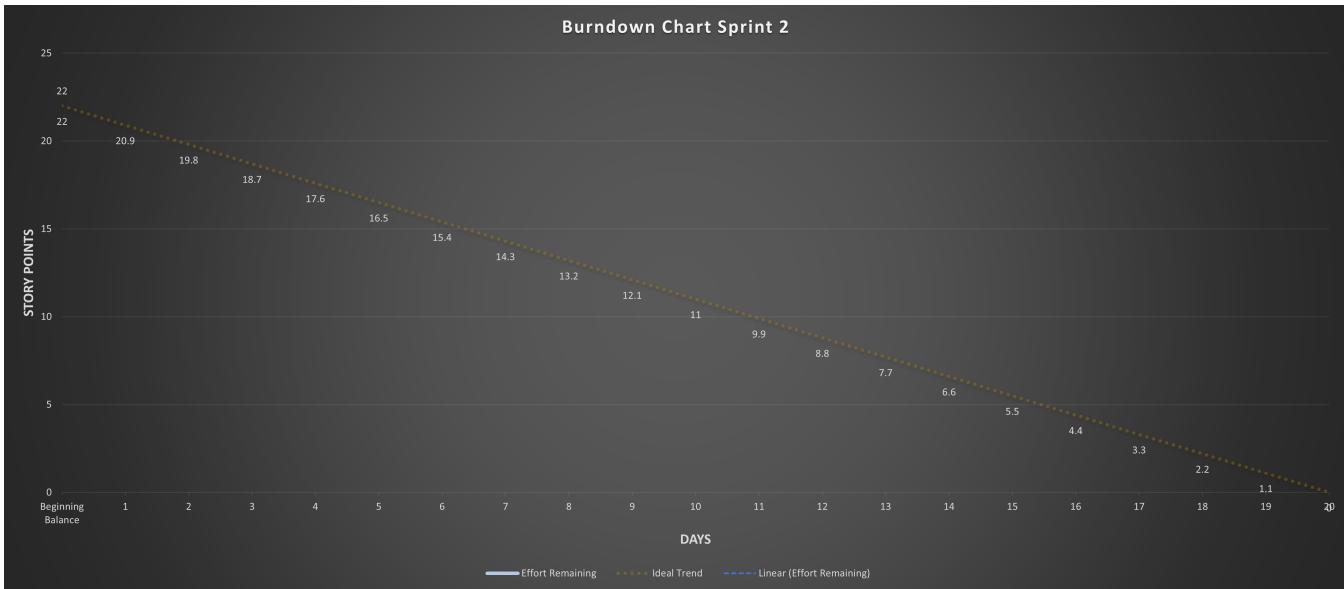
ID	User stories	Acceptance criteria	Priority	Story points
1	As an user, I want the front-end website can communicate with the back-end server so that my calculations behind the scenes can be sent to the front-end website.	<p>Scenario: The front-end program can send the requests to the server while the server can send back the calculation results as responses.</p> <p>Given:</p> <ul style="list-style-type: none"> • The user browser prepared a request. <p>When:</p> <ul style="list-style-type: none"> • The user browser send a (REST /HTTP) request to the server. <p>Then:</p> <ul style="list-style-type: none"> • The server should calculate the result. • The result is returned back to the user's browser. 	High	21
2	As an user, I want the backend server can connect with the database so that my calculation results can be stored in the database for further use.	<p>Scenario: The calculation results can be stored in the database when the server connects to the database and inserts the data.</p> <p>Given:</p> <ul style="list-style-type: none"> • The database and the server instance has healthy connection. <p>When:</p> <ul style="list-style-type: none"> • The server update the data into DB instance. <p>Then:</p> <ul style="list-style-type: none"> • The data is immediately available when the update transaction has been completed. 	High	2

3	As an user, I want the backend server can query all the required input data from the database so that I don't need to type any additional input as before	<p>Scenario: The required input data can be queried from the DB when the server is calculating the results.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. The user made inputs for the calculation. <p>When:</p> <ul style="list-style-type: none"> The server starts calculation process and updating the data into DB instance. <p>Then:</p> <ul style="list-style-type: none"> The previous inputs should all become visible on the frontend. 	High	13
4	As an user, I want the backend server can return some important variables or error messages(e.g. out of range/extrapolation) to the frontend website during the calculation so that I can track the calculation process and find the problems easily and be aware of potential sources of discrepancies.	<p>Scenario: The user can view the important variables or error messages during the backend calculation.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. The user made inputs for the calculation. <p>When:</p> <ul style="list-style-type: none"> The server starts calculation process and updating the data into DB instance. <p>Then:</p> <ul style="list-style-type: none"> The important variables or error messages should all become visible. 	High	2
5	As an user, I want all of the data* is correctly interpolated, calculated and updated , so that I can use them for audit process.	<p>Scenario: The user wants to upload Bw, Mu/rho, Closed Cone Corr (CCC), Nk, Pstem and gets correct results.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. <p>When:</p> <ul style="list-style-type: none"> The user made inputs for the calculation. <ul style="list-style-type: none"> The calculation and interpolation involves: <ol style="list-style-type: none"> Bw Mu/rho Closed Cone Corr (CCC) Pstem Nk The input consists of Excel, and frontend HTML input. <p>Then:</p> <ul style="list-style-type: none"> The calculation result should be correctly respond to the frontend and stored on the Database. 	High	21

6	As an user, I want to see all audit cases on the system website.	<p>Scenario: The user wants to view all the audit cases on the web application.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. <p>When:</p> <ul style="list-style-type: none"> The user is logged in with User account <p>Then:</p> <ul style="list-style-type: none"> All cases with their audit ID, date, clinics and status are visible to the frontend. 	High	2
7	As an user, I want to search the specific audit case by date, audit number, etc.	<p>Scenario: The user wants to view specific audit case(s) on the web application.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. The user is logged in with User account <p>When:</p> <ul style="list-style-type: none"> User type audit ID/date/clinics/status in the search box (or filter tool bar) <p>Then:</p> <ul style="list-style-type: none"> The corresponding audit case(s) will be visible. 	Mid	5
8	As an user, I want all of the data is correctly interpolated, calculated and updated , so that I can use them for audit process.	<p>Scenario: The user wants to upload Bw, Mu/rho, Closed Cone Corr (CCC), Nk, Pstem and gets correct results.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance has healthy connection. <p>When:</p> <ul style="list-style-type: none"> The user made inputs for the calculation. <ul style="list-style-type: none"> The input consists of Excel, and /or frontend HTML input. <p>Then:</p> <ul style="list-style-type: none"> The calculation result should be correctly received from the backend and visible on the frontend. 	High	5

Burndown chart

The burndown chart has only the ideal line, which suggests the ideal workload per day. Notice that this chart includes 20 days, meaning 4 weeks of development (4 weeks * 5 working days). The completed version will be evaluated in [Epic 1 - A production ready system](#).



Sprint 3 Plan

VersionID	Description	Date
1.0	Document initialization.	2022-03-21
1.1	Review and edit some requirements based on the change of Epic 2 - A secure system . Reviewer: Caleb	2022-04-23
2.0	Review and added Epic 2 link under the burndown chart because of the project completion.	2022-05-28

Sprint 3 Goal

- Complete ALL user stories in [Epic 2 - A secure system](#)
- The sign of the completion of Epic 2 represents the security feature as well as general software engineering practice has been conducted. It represents:
 - The code is well structured, and any memory operation is safe.
 - An account system that blocks unauthorized access.

Planned user stories and acceptance tests

Note that all the user stories below refer to [Epic 2 - A secure system](#), the story points are estimated along the Fibonacci series (1, 2, 3, 5, 8, 13, 21...)

Also note that below user stories are subject to change in later communication with the client.

The Acceptance test in this sprint contains both Accept/Reject.

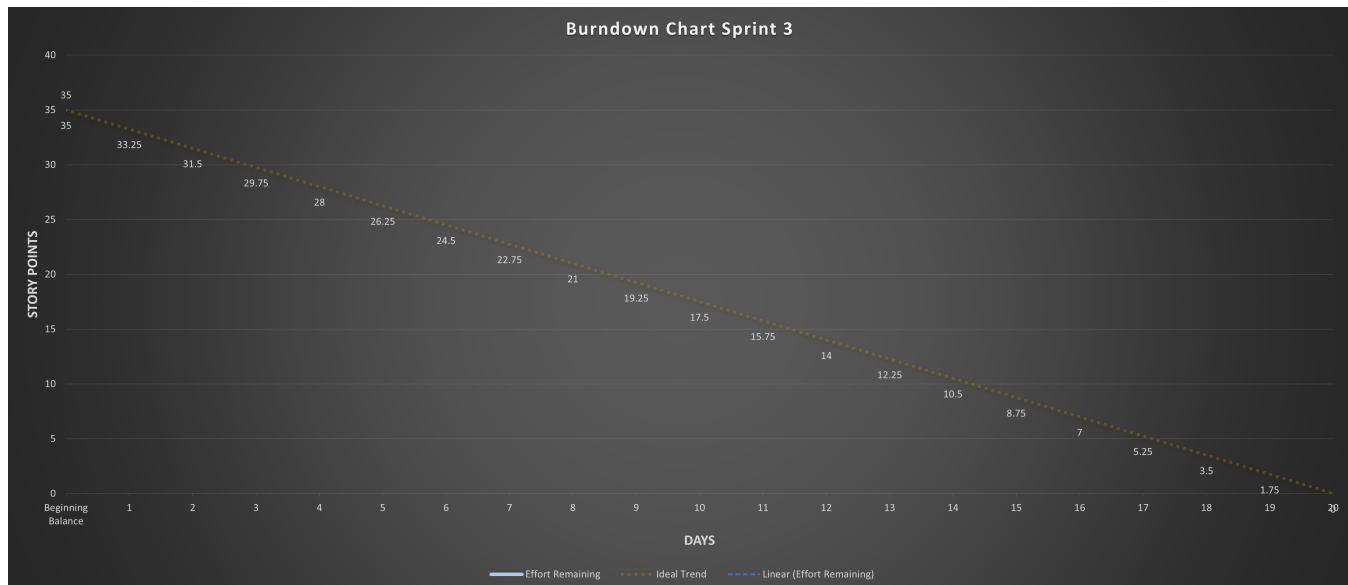
ID	User stories	Acceptance criteria	Test Result (Accept /Reject)	Priority	Story points
1	As a user, I want to be able to login to the website so that I can use the functions provided on the server.	<p>Given:</p> <ul style="list-style-type: none"> • The user is located at Log in page. • A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> • The user input username, password, then hit log in button. <p>Then:</p> <ul style="list-style-type: none"> • The user will be redirected to the Audit cases page. <p>Given:</p> <ul style="list-style-type: none"> • The user is located at Log in page. • A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> • The user input INCORRECT username, password, then hit log in button. <p>Then:</p> <ul style="list-style-type: none"> • The user will stay at Log in page with error messages. 	Accept	High	5

2	<p>As a user, I want to be able to change the password so that I can use my own secret to access the system.</p>	<p>Given:</p> <ul style="list-style-type: none"> The user is located at Audit cases /Lookup page, and accessed the user panel pop up. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input valid password, then hit change password button. <p>Then:</p> <ul style="list-style-type: none"> The user will be informed that the change password operation <i>successted</i> Then the user will be redirected to the Log in page. 	Accept	High	3
		<p>Given:</p> <ul style="list-style-type: none"> The user is located at Audit cases /Lookup page, and accessed the user panel pop up. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input invalid password, then hit change password button. <p>Then:</p> <ul style="list-style-type: none"> The user will be informed that the change password operation <i>failed</i>. 	Reject		
3	<p>As a user, I want to be able to log out the website so that nobody can see my account on my machine</p>	<p>Given:</p> <ul style="list-style-type: none"> The user is logged in with User account. The user is located at Audit cases /Lookup page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input hit log out button. <p>Then:</p> <ul style="list-style-type: none"> The user will be redirected to the Log in page. 	Accept	High	1
4	<p>As a user, I want to make sure that all services and functionality provided by the web server is serving the authenticated individual/entity, so that the system is secure.</p>	<p>Given:</p> <ul style="list-style-type: none"> The user is logged in with User account. The user is located at Audit cases /Lookup page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user made any input in identification/lookup/change password function. <p>Then:</p> <ul style="list-style-type: none"> The user will be prompted/redirected accordingly. (The functions are served normally) 	Accept	High	13

	<p>Given:</p> <ul style="list-style-type: none"> The user is logged in with User account. The user is located at Audit cases /Lookup page. A good and healthy web server. An expired user token. <p>When:</p> <ul style="list-style-type: none"> The user made any input in identification/lookup/change password function. <p>Then:</p> <ul style="list-style-type: none"> The user will be prompted/redirected to the log in page. (The user input will never works) 	Reject		
5	<p>As a user, I want to be able to use the web server in my local machine so that the security of the system is maximized.</p> <p>Given:</p> <ul style="list-style-type: none"> A completed solution (backend, frontend, database) All solutions can run standalone as a Docker service. <p>When:</p> <ul style="list-style-type: none"> Client access the web server in the local machine. <p>Then:</p> <ul style="list-style-type: none"> Every functionality should be available in the web server. 	Accept	Medium	13

Burndown chart

As for the current stage, the team has not yet started the sprint 3. The current burndown chart has only the ideal line, which suggests the ideal workload per day. Notice that this chart includes 20 days, meaning 4 weeks of development (4 weeks * 5 working days). The completed version will be evaluated in [Epic 2 - A secure system](#)



| Software Quality Assurance Plan

Abbreviation	SQP
Epic	Epic 1 - A production ready system Epic 2 - A secure system
Implement	Quality Management
Document status	FINAL
Document owner	Quality Assurance Lead Haoran Zhao

Purpose

Software quality assurance consists of a means of monitoring the software engineering processes and methods used to ensure quality. This encompasses the entire software development process, and is organised into goals, commitments, abilities, activities, measurements, and verification. It is a supporting process that has to provide the independent assurance in which all the work products, activities, and processes comply with the predefined plans. The detail implementation of QA, including coding standards, version control policy, testing plan, and code review is available at | [Quality Management](#).

Goals

This document aims to answer the following questions:

- What is our software quality process?
- When should we update our completed tasks to done?
- What is the criteria to accept tasks as part of our release?

Roles

Role	Responsibility
Moderator	Leads inspections Schedules meetings Controls meetings
Author	Creates or maintains the product being inspected
Reader	Describes the sections of work product to the team as they proceed
Recorder	Classifies and records defects and issues raised
Reviewer	Attempts to find errors in the product

Methods

Walkthrough

- A walkthrough is a peer group review of software documents
- Involves several people
- Each person plays a defined role
- Typically, a walkthrough involves at least one person (usually an author) whose job is to introduce the documents to the rest of the group
- In the context of requirements documents, reviews are for consensus and walkthroughs are for training

Reviews

- Manual process
- Multiple readers
- Checks for anomalies and omissions
- Representatives of stakeholders should participate in a review

Detection Methods

Ad Hoc

- Informal, non-systematic detection technique
- No explicit assignment of reviewer responsibility
- Relies on knowledge and experience of inspector

Checklist

- Very popular method
- Reuse of "lessons learned"
- Defines reviewer responsibilities
- Suggests ways for reviewers to identify defects

Scenario

- Defect specific procedures
- Used to detect particular classes of defects
- Each reviewer executes a single scenario
- Multiple reviewers are needed to achieve broad coverage of the document
- Each review has a distinct responsibility

Quality Metrics

Features present within tools which can aid us in measuring quality of our software engineering process.

Trello

- Estimation
- Sprint planning
- Scrum board columns
- Reports
- Issues (types)
- Links (toolset)

Confluence

- Specification

Github

- Branches
- Merges
- Tests

JUnit

- Integration
- Tests

| Retrospectives

[Add Retrospective](#)

Title	Date	Participants
Sprint 3 Retrospective	28 May 2022	Xinyu Zeng CALEB Wang Yipeng Li Fangtai DONG Haoran Zhao
Sprint 2 Retrospective	01 May 2022	CALEB Wang Xinyu Zeng Haoran Zhao Fangtai DONGYipeng Li
Sprint 1 Retrospective	15 Apr 2022	Xinyu Zeng Haoran Zhao CALEB Wang Yipeng Li Fangtai DONG

Sprint 1 Retrospective

Date	15 Apr 2022
Participants	Xinyu Zeng Haoran Zhao CALEB Wang Yipeng Li Fangtai DONG

Retrospective

What did we do well?

- Efficiency
 - We were able to meet the Sprint goal of developing the basic functionality of the system.
 - All team members completed their assigned tasks before the deadline.
- Collaboration
 - Used online tools such as ZOOM and Trello to promote the smooth progress of the project.
 - Actively communicate with customers and discuss specific requirements of some modules and functions that are confusing
- Meetings
 - No absence from any team member during all meetings
 - Have weekly meetings with team members assigned the same tasks
 - All shown responsive during group meetings.
 - Meetings were highly structured and meeting agenda explicit.

What should we have done better?

- Task allocation
 - The assignment of tasks should be more reasonable, taking into account the time, energy and priority of tasks of each member.
 - Due to the high coupling of previous designs, the division of labor is not very balanced. In sprint 2 we optimize the division of labor by decoupling.
- ReadMe File
 - The ReadME section should be more detailed, introducing each function and the role of related parameters.
- Sprint 1 Delivery
 - Meeting notes should not be exported to repo since it is Internal Processes.
 - Do Be Feel list need more clear functional requirements, such as security, reliability.
 - More detailed descriptions of plan2&3 are needed in trello.

Actions

- Update the Personal based on client feedback Haoran Zhao
- More details on Development Sprint Plan CALEB Wang
- Update the Do Be Feel list based on feedback [Fangtai DONG](#)

Sprint 2 Retrospective

Date	01 May 2022
Participants	CALEB Wang Xinyu Zeng Haoran Zhao Fangtai DONGYipeng Li

Retrospective

What did we do well?

- Communication:
 - An active discussion is constantly maintained through Wechat Group.
 - An online meeting via Zoom is held when it is necessary
- Collaboration:
 - Successfully integrated the front-end and the backend with our API.
 - A healthy communication line is established despite of the time difference and other challenges / inhibitions. The team was able to help each other with problem-solving
- Velocity:
 - Deadline is always met on time [Epic 1 - A production ready system](#) according to the Burndown chart. Hence, the team was able to complete required features.

What should we have done better?

- Process
 - Engaged more with the client to understand and define the scope better. We could have setup a meeting with the client to confirm and make sure that we captured all the critical and priority requirements before planning each sprint and other lower priority features, such as user login.
- Product
 - Exception Handling was not detailed enough to be included in this current sprint due to the time limitation.
- Documentations
 - Not familiar enough with the structure of some required documentations at all.

Actions

- more details in Exception Handling.

Sprint 3 Retrospective

Date	28 May 2022
Participants	Xinyu Zeng CALEB Wang Yipeng Li Fangtai DONG Haoran Zhao

Retrospective

What did we do well?

- Efficiency
 - The performance of our team was outstanding. Most team members finish the allocated tasks prior to its deadline. And a few managed to even completed some requirements for sprint 4.
- Velocity:
 - Deadline is always met on time [Epic 2 - A secure system](#) according to the Burndown chart. Hence, the team was able to complete required features.
- Meeting
 - No absence from any team member during all meetings.
 - All shown responsive during group meetings.
 - Meetings were highly structured and meeting agenda explicit.

What should we have done better?

Learn from past mistakes

For every sprint planning, it is important to take into consideration the successful and well, not so successful, practices in past sprints

Leave some buffer time

It is useful to allot a little extra time in a sprint for unprecedeted situations. These situations may vary depending on your environment.

• Github

- Team still needs to play around more with the GitHub mechanism to ensure code maintainability.

Actions

- To upload the recording for presentation.

| Quality Management

Definitions

Quality

End-user's Perspective

- Typically, end-users judge the quality of a product by their interaction with it
- For users, a system has quality if it is fit for purpose, is reliable, has reasonable performance, is easy to learn and use, and helps the users in achieving their goals
- Sometimes, if the functionality is hard to learn but is extremely important and worth the trouble of learning, then users will still judge the system to have high quality
- These are termed external quality characteristics, because they are typically associated with the external behaviour of the system.

Developer's Perspective

- The developer's perspective typically also includes the number of faults that the system has, ease of modifying the system, ease of testing the system, ease of understanding the system design, re-usability of components, conformance to requirements, resource usage, and performance
- These are mainly internal quality characteristics, because they are concerned with the quality of the internal structure of the system

The quality of the process influences the quality of the product.

Ensuring product quality typically involves measuring and assessing the product and process.

Standards

A set of rules for ensuring quality

Types

- Product standards
- Process standards

Process

1. Quality Planning

The selection of appropriate procedures and standard from the framework, adopted for CoachingMate Analytics.

2. Quality Assurance

The establishment of a framework of organisational procedures and standards that lead to high-quality software.

3. Perform Quality Control

Ensuring that the software development team has followed the project quality procedures and standards.

Garvin's Quality Dimensions

- Aesthetics
- Conformance
- Durability
- Feature Quality
- Perception
- Performance Quality
- Reliability
- Serviceability



McCall's Quality Factors

Product Revision

Product Transition

Product Operation

- Portability
- Correctness

- Maintainability
- Flexibility
- Testability

- Reusability
- Interoperability

- Reliability
- Usability
- Integrity
- Efficiency

References

- Garvin D., "Competing on the Eight Dimensions of Quality," Harvard Business Review , November 1987, pp. 101–109.
- McCall, J. A., Richards, P. K., & Walters, G. F. (1977). Factors in software quality. volume i. concepts and definitions of software quality. GENERAL ELECTRIC CO SUNNYVALE CA.
- Pressman, R. S. (2005). Software engineering: a practitioner's approach. Palgrave Macmillan.

| Coding Standards

This section lists the standards employed and followed throughout development process.

- [General Front-end Guide](#)
- [Back-end Guide](#)
- [GitHub Guide](#)

General Front-end Guide

Naming:

- Names should be descriptive, but not excessively so.
- Variable and function names should be full words.
- Use camel case with a lowercase first letter.
- Constructors intended for use with new should have a capital first letter ([CamelCase](#)).

Commenting:

- Comments come before the code to which they refer.
- Comments should always be preceded by a blank line.
- Multi-line comments should be used for long comments.

Functions and variables:

- Each function should begin with a single comma-delimited var statement that declares any local variables necessary.
- All globals used within a file should be documented at the top of that file.

Syntax and Format:

- Indentation with tabs.
- No whitespace at the end of line or on blank lines.
- Lines should usually be no longer than 80 characters, and should not exceed 100 (counting tabs as 4 spaces).
- if/else/for/while/try blocks should always use braces, and always go on multiple lines.
- Always include extra spaces around elements and arguments.
- Use single-quotes for string literals.

Reference

The style guide for javascript from vue can be the guide for all the front end code, as it makes sense to follow above sessions with every language for which style guide isn't mentioned.

- [JavaScript Coding Standards](#)
- [HTML Coding Standards](#)
- [CSS Coding Standards](#)
- [Vue Coding Standards](#)

Back-end Guide

Since the back-end used by the KV ACDS platform is Spring Boot , the Spring Boot standards are followed.

- [Spring Boot Coding Standards](#)
- [Java Coding Standard](#)

GitHub Guide

- [Creating or Duplicating repository](#)

Command	Description
git init	Initialize a local Git repository
git clone ssh://git@github.com/[username]/[repository-name].git	Create a local copy of a remote repository

- [Basic snap shooting](#)

Command	Description
---------	-------------

<code>git status</code>	Check status
<code>git add [file-name.txt]</code>	Add a file to the staging area
<code>git add -A</code>	Add all new and changed files to the staging area
<code>git commit -m "[commit message]"</code>	Commit changes
<code>git rm -r [file-name.txt]</code>	Remove a file (or folder)

- **Branching and merging**

Command	Description
<code>git branch</code>	List branches (the asterisk denotes the current branch)
<code>git branch -a</code>	List all branches (local and remote)
<code>git branch [branch name]</code>	Create a new branch
<code>git branch -d [branch name]</code>	Delete a branch
<code>git push origin --delete [branchName]</code>	Delete a remote branch
<code>git checkout -b [branch name]</code>	Create a new branch and switch to it
<code>git checkout -b [branch name] origin/[branch name]</code>	Clone a remote branch and switch to it
<code>git checkout [branch name]</code>	Switch to a branch
<code>git checkout -</code>	Switch to the branch last checked out
<code>git checkout -- [file-name.txt]</code>	Discard changes to a file
<code>git merge [branch name]</code>	Merge a branch into the active branch
<code>git merge [source branch] [target branch]</code>	Merge a branch into a target branch
<code>git stash</code>	Stash changes in a dirty working directory
<code>git stash clear</code>	Remove all stashed entries

- **Sharing and Updating project**

Command	Description
<code>git push origin [branch name]</code>	Push a branch to your remote repository
<code>git push -u origin [branch name]</code>	Push changes to remote repository (and remember the branch)
<code>git push</code>	Push changes to remote repository (remembered branch)
<code>git push origin --delete [branch name]</code>	Delete a remote branch
<code>git pull</code>	Update local repository to the newest commit
<code>git pull origin [branch name]</code>	Pull changes from remote repository
<code>git remote add origin ssh://git@github.com/[username]/[repository-name].git</code>	Add a remote repository
<code>git remote set-url origin ssh://git@github.com/[username]/[repository-name].git</code>	Set a repository's origin branch to SSH

- **Inspection and Comparison**

Command	Description
Command	Description
<code>git log</code>	View changes
<code>git log --summary</code>	View changes (detailed)

```
git diff [source branch] [target branch] Preview changes before merging
```

- **References**

The above commands have been referred by a git hub account by joshnh.

Key Note: This document will be updated after complete negotiations and discussions are done by the team along with the Client and his technical team.

Currently,The team is involved in testing and validating the system produced by previous unimelb team, to verify if the system is reliable for extend-ability.

Based on the result of exploring the previous team's work, a decision will be made whether the team can build on top of system produced by previous team

or Should the team be building the entire system from scratch which would again cause the risk of increase of scope.

| Version Control

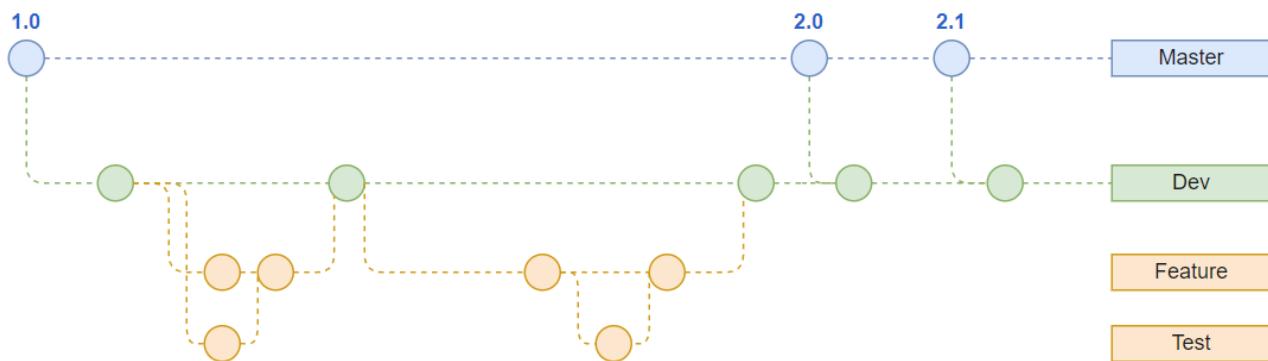
The code repositories for the application are hosted on [GitHub](#).

For all the repositories, we have chosen to use Git branching workflow, with branching and merging policies in place, alongside automated check described below.

Branching Policy

There are mainly 4 types of branches:

1. Master: The stable release, where the tag is created when other branches merged into main/master.
2. Dev: The development branch, which holds nightly release, it will become stable when the code is reviewed and thoroughly tested.
3. Feature: The feature branch(es), each feature branch should be spawned from the latest dev branch. The naming should be "{feature_name}-feature".
4. Test: The branches for testing purpose, there will be unit + integration testing in this project. For the unit testing, the naming should be "{feature_name}-feature-test". For the integration testing, the naming should be "Dev-test". Each testing branch will be finally merged with the corresponding feature/dev branch.



Commit Policy

It is important to do **atomic commit**, which means small and meaningful commit rather than big changes in one single commit. Commit messages also should include a meaningful message.

Example below is just for guidance on committing a change:

```
init readme.md
fix calculation for hvl
add feature to be able to search for audit cases
```

Merge Policy

When trying to merge a local branch to `main`, one reviewer [Yipeng Li](#) is required to review the pull request. The reviewer then will decide whether to leave a review comment or approve and merge the branch into `main`.

The pull request for a branch normally created once the user-story (or task) has been completed.

Below is the typical **workflow of creating a pull request**:

1. Latest code must be taken from `main` or in other words do a merge from main to local branch. This is to ensure that there is no merge conflicts when creating a pull request.
2. The developed feature must be tested to ensure the merging not introduce new bugs
3. Pull request is made for the specific branch
4. At least one reviewer must be assigned which that person will review the branch
5. All changes must be reviewed and accepted by the reviewer assigned
6. All checks from GitHub Actions must be passed

Code Review

Nomination of Reviewers: [Yipeng Li](#) and [CALEB Wang](#)

TimeFrames for review: Expected responding time is within a week after pull request

Some of the key aspects that a reviewer should consider:

1. **Code formatting:** check the code formatting for readability and ensure that proper naming conventions have been followed (e.g. camelCase or snake_case) and that lines are not too long
2. **DRY (Do not Repeat Yourself):** same code should not be repeated multiple times
3. **Constants and variables:** make sure there is no magic number and all constants and variables are used the way they should
4. **Secrets management:** sensitive information should not be written in any of the branch. Consider using environment variables or files that are distributed in share folder.

See more details information in [| Code Review Checklist](#).

| Testing

Usability Testing

1. Prototyping
2. Reviewing
3. Refining

Acceptance Testing

Revision History

Date	Version	Description	Author
2022/05/01	V1.0	Initial Draft	Haoran Zhao

Acceptance Criteria ID	Acceptance Test	Acceptance criteria	Test Result (Accept /Reject)	Comment
1	As a user, I want the front-end website can communicate with the back-end server so that my calculations behind the scenes can be sent to the front-end website.	<p>Scenario: The front-end program can send the requests to the server while the server can send back the calculation results as responses.</p> <p>Given:</p> <ul style="list-style-type: none"> The user browser prepared a request. <p>When:</p> <ul style="list-style-type: none"> The user browser sends a (REST/HTTP) request to the server. <p>Then:</p> <ul style="list-style-type: none"> The server should calculate the result. The result is returned back to the user's browser. 	Accept	
2	As a user, I want the backend server can connect with the database so that my calculation results can be stored in the database for further use.	<p>Scenario: The calculation results can be stored in the database when the server connects to the database and inserts the data.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance have a healthy connection. <p>When:</p> <ul style="list-style-type: none"> The server updates the data into the DB instance. <p>Then:</p> <ul style="list-style-type: none"> The data is immediately available when the update transaction has been completed. 	Accept	

3	<p>As a user, I want the backend server can query all the required input data from the database so that I don't need to type any additional input as before</p>	<p>Scenario: The required input data can be queried from the DB when the server is calculating the results.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance have a healthy connection. The user made inputs for the calculation. <p>When:</p> <ul style="list-style-type: none"> The server starts the calculation process and updates the data into the DB instance. <p>Then:</p> <ul style="list-style-type: none"> The previous inputs should all become visible on the frontend. 	Accept	
4	<p>As a user, I want the backend server can return some important variables or error messages(e.g. out of range/extrapolation) to the frontend website during the calculation so that I can track the calculation process and find the problems easily and be aware of potential sources of discrepancies.</p>	<p>Scenario: The user can view the important variables or error messages during the backend calculation.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance have a healthy connection. The user made inputs for the calculation. <p>When:</p> <ul style="list-style-type: none"> The server starts the calculation process and updates the data into the DB instance. <p>Then:</p> <ul style="list-style-type: none"> The important variables or error messages should all become visible. 	Accept	
5	<p>As a user, I want all of the data* to be correctly interpolated, calculated, and updated, so that I can use them for the audit process.</p>	<p>Scenario: The user wants to upload Bw, Mu/rho, Closed Cone Corr (CCC), Nk, Pstem and gets correct results.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance have a healthy connection. <p>When:</p> <ul style="list-style-type: none"> The user made inputs for the calculation. <ul style="list-style-type: none"> The calculation and interpolation involve: <ol style="list-style-type: none"> Bw Mu/rho Closed Cone Corr (CCC) Pstem Nk The input consists of Excel, and frontend HTML input. <p>Then:</p> <ul style="list-style-type: none"> The calculation result should be correctly responded to the frontend and stored in the Database. 	Accept	

6	As a user, I want to see all audit cases on the system website.	<p>Scenario: The user wants to view all the audit cases on the web application.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance have a healthy connection. <p>When:</p> <ul style="list-style-type: none"> The user is logged in with the User account <p>Then:</p> <ul style="list-style-type: none"> All cases with their audit ID, date, clinics and status are visible to the frontend. 	Accept	
7	As a user, I want to search the specific audit case by date, audit number, etc.	<p>Scenario: The user wants to view specific audit case(s) on the web application.</p> <p>Given:</p> <ul style="list-style-type: none"> The database and the server instance have a healthy connection. The user is logged in with the User account <p>When:</p> <ul style="list-style-type: none"> User type audit ID/date/clinics /status in the search box (or filter toolbar) <p>Then:</p> <ul style="list-style-type: none"> The corresponding audit case(s) will be visible. 	Accept	
8.1	As a user, I want to be able to log in to the website so that I can use the functions provided on the server.	<p>Given:</p> <ul style="list-style-type: none"> The user is located on the login page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user inputs username, and password, then hit the login button. <p>Then:</p> <ul style="list-style-type: none"> The user will be redirected to the Audit cases page. 	Accept	
8.2		<p>Given:</p> <ul style="list-style-type: none"> The user is located on the login page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user inputs the INCORRECT username, and password, then hit the login button. <p>Then:</p> <ul style="list-style-type: none"> The user will stay on the login page with error messages. 	Reject	

9.1	<p>As a user, I want to be able to change the password so that I can use my own secret to access the system.</p>	<p>Given:</p> <ul style="list-style-type: none"> The user is located on the login page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user inputs username, and password, then hit the login button. <p>Then:</p> <ul style="list-style-type: none"> The user will be redirected to the Audit cases page. 	Accept	
9.2		<p>Given:</p> <ul style="list-style-type: none"> The user is located on the login page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user inputs the INVALID username, and password, then hit the login button. <p>Then:</p> <ul style="list-style-type: none"> The user will be redirected to the Audit cases page. 	Reject	
10	<p>As a user, I want to be able to log out of the website so that nobody can see my account on my machine</p>	<p>Given:</p> <ul style="list-style-type: none"> The user is logged in with the User account. The user is located on the Audit cases page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user input hit the log out button. <p>Then:</p> <ul style="list-style-type: none"> The user will be redirected to the login page. 	Accept	
11	<p>As a user, I want to make sure that all services and functionality provided by the webserver are serving the authenticated individual/entity so that the system is secure.</p>	<p>Given:</p> <ul style="list-style-type: none"> The user is logged in with the User account. The user is located on the Audit cases page. A good and healthy web server. <p>When:</p> <ul style="list-style-type: none"> The user made any input in the identification/lookup/change password function. <p>Then:</p> <ul style="list-style-type: none"> The user will be prompted /redirected accordingly. (The functions are served normally) 	Accept	

11.1		<p>Given:</p> <ul style="list-style-type: none"> The user is logged in with the User account. The user is located on the Audit cases page. A good and healthy web server. An expired user token. <p>When:</p> <ul style="list-style-type: none"> The user made any input in the identification/lookup/change password function. <p>Then:</p> <ul style="list-style-type: none"> The user will be prompted /redirected to the login page. (The user input will never work) 	Reject	
12	As a user, I want to be able to use the webserver in my local machine so that the security of the system is maximized.	<p>Given:</p> <ul style="list-style-type: none"> A completed solution (backend, frontend, database) All solutions can run standalone as a Docker service. <p>When:</p> <ul style="list-style-type: none"> Client access the web server on the local machine. <p>Then:</p> <ul style="list-style-type: none"> Every functionality should be available on the webserver. 	Accept	

Unit Testing

- [Lookup](#)
 - [TC1: Bw functions](#)
 - TC 1.1: Read Bw lookup tables
 - TC 1.2: Read Bw Excel Sheet
 - TC 1.3: Save Bw lookup tables
 - [TC2: Farmer functions](#)
 - TC 2.1: Read farmer lookup tables
 - TC 2.2: Read Farmer Excel Sheet
 - TC 2.3: Save Farmer lookup tables
 - [TC3: MurhoAI functions](#)
 - TC 3.1: Read MurhoAI lookup tables
 - TC 3.2: Read MurhoAI Excel Sheet
 - TC 3.3: Save MurhoAI lookup tables
 - [TC4: MurhoCu functions](#)
 - TC 4.1: Read MurhoCu lookup tables
 - TC 4.2: Read MurhoCu Excel Sheet
 - TC 4.3: Save MurhoCu lookup tables
 - [TC5: Plane-parallel functions](#)
 - TC 5.1: Query Planeparallel lookup tables
 - TC 5.2: Read Planeparallel Excel Sheet
 - TC 5.3: Save Planeparallel lookup tables
 - [TC6: Pstem functions](#)
 - TC 6.1: Query Pstem lookup tables
 - TC 6.2: Read Pstem Excel Sheet
 - TC 6.3: Save Pstem lookup tables
- [Calculation](#)
 - TC 7: Get a list of BackResults
 - TC 8: Query the specific beam planeparallel Nk
 - TC 9: Query the latest beam planeparallel Nk
 - TC 10: Query the specific hvl
 - TC 11: Query the latest farmer lookup table
 - TC 12: Check whether there is data that has lower Hvl
 - TC 13: Check whether there is data that has higher Hvl
 - TC 14: Query the Input cone data list
 - TC 15: Select data from the farmer table
 - TC 16: Calculate Nk value
 - TC 17: Query the latest BwAIcu data
 - TC 18: get the target boundary
 - TC 19: Calculate Bw value
 - TC 20: Calculate Bw value with specified inputs
 - TC 21: Calculate Murho value
 - TC 22: Calculate Murho value with specified inputs
 - TC 23: Calculate ccc value
 - TC 24: Calculate pstem value
- [AuditController](#)
 - TC 25: upload identification excel
 - TC 26: Query audit cases
 - TC 27: Edit audit cases
 - TC 28: Delete audit case
 - TC 29: Query work datasheet1
 - TC 30: Query work datasheet2
 - TC 31: Query beam information
 - TC 32: Query data from the frontend table
 - TC 33: Save data into input_data_front_end table
 - TC 34: Query idenification
 - TC 35: Query warnings
- [AuditService](#)
 - TC 36: Read audit information in Excel
 - TC 37: Read reference cone in Excel
 - TC 38: Read reference cone comments in Excel
 - TC 39: Read cone in Excel
 - TC 40: Read beam quality in Excel
 - TC 41: Get the treatment machine information
 - TC 42: Get the facility information
 - TC 43: Get the physical address
 - TC 44: Get the reporting address
 - TC 45: Get the facility representative
 - TC 46: Get the audit information
 - TC 47: Get the cone
 - TC 48: Get the reference cone
 - TC 49: Get the beam data
 - TC 50: Get the beam inputs
 - TC 51: Get the reference dosimetry
 - TC 52: Get the kv reference dosimetry
 - TC 53: Get the audit history

Revision History

Date	Version	Description	Author
2022/05/28	V1.2	Add audit unit testing	Haoran Zhao
2022/05/01	V1.1	Add calculation unit testing	Haoran Zhao
2022/03/28	V1.0	Initial Draft	Haoran Zhao

Lookup

TC1: Bw functions

The database relationship:

- bw_al_cu
 - bw_hvl_al
 - bw_hvl_cu
 - bw_ssd
 - bw_diameter

TC 1.1: Read Bw lookup tables

Function Name:	queryBw
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the relevant data from the Bw lookup tables can be extracted with measured HVL (Al/Cu), measured SSD (cm), and field size diameter (cm) from tables on the database.
Pre-Condition:	<ul style="list-style-type: none"> • The user can extract the correct Bw value corresponding to SSD (cm), diameter (cm), and HVL Al/Cu (mm Al/Cu). • If there is no corresponded value, the function will catch an exception.
Read Successfully:	The relevant data of Bw lookup tables can be extracted correctly.
Read Unsuccessfully:	The relevant data of Bw lookup tables can not be extracted with the queryBw function.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 1.2: Read Bw Excel Sheet

Function Name:	readBw
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the relevant elements of Bw Excel can be read and stored in a list.
Pre-Condition:	<ul style="list-style-type: none"> • The Bw Excel contains all elements which the readBw function needs. • The data type of elements is as same as required.
Read Successfully:	All the required elements are read and stored in the list.
Read Unsuccessfully:	The application will catch an error. And all processes of this read function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 1.3: Save Bw lookup tables

Function Name:	saveBw
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the Bw lookup tables can be saved on tables of the database
Pre-Condition:	<ul style="list-style-type: none"> The relevant Bw information has been read successfully and stored in the required list. The database is connected.
Save Successfully:	The relevant Bw information is saved on the bw_al_cu table and other relevant tables of the database.
Save Unsuccessfully:	The relevant Bw information is not saved on the bw_al_cu table and other relevant tables of the database. And all processes of this save function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC2: Farmer functions

The database relationships:

- chambers_list
 - beam_farmer_list
 - beam_farmer_chamber

TC 2.1: Read farmer lookup tables

Function Name:	queryFarmer
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the relevant data from the Farmer lookup tables can be extracted from tables on the database
Pre-Condition:	<ul style="list-style-type: none"> The user can extract the correct Nk value corresponding to chamber SN, beam energy (kV), and measured HVL Al/Cu (mm Cu). If there is no corresponded value, the function will catch an exception.
Read Successfully:	The relevant data of Farmer lookup tables can be extracted correctly.
Read Unsuccessfully:	The relevant data of Farmer lookup tables can not be extracted correctly.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 2.2: Read Farmer Excel Sheet

Function Name:	readFarmer
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the relevant elements of Farmer Excel can be read and stored in a list.
Pre-Condition:	<ul style="list-style-type: none"> The Farmer Excel contains all elements which the readFarmer function needs. The data type of elements is as same as required.
Read Successfully:	All the required elements are read and stored in the list.
Read Unsuccessfully:	The application will catch an error. And all processes of this read function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 2.3: Save Farmer lookup tables

Function Name:	saveFarmer
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the Farmer lookup tables can be saved on tables of the database
Pre-Condition:	<ul style="list-style-type: none"> The relevant Farmer information has been read successfully and stored in the required list. The database is connected.
Save Successfully:	The relevant Farmer information is saved on the beam_farmer_chamber, beam_farmer_list, and chamber_list tables of the database.
Save Unsuccessfully:	The relevant Farmer information is not saved on the beam_farmer_chamber, beam_farmer_list, and chamber_list tables of the database. And all processes of this save function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC3: MurhoAI functions

The database relationship:

- murho_ai

TC 3.1: Read MurhoAI lookup tables

Function Name:	queryMurhoAI
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct mass-energy absorption coefficient for the Aluminum (murho AI) value is extracted using HVL AI (mm AI) value as input.
Pre-Condition:	<ul style="list-style-type: none"> The user can extract the correct murho value corresponding to HVL AI (mm AI). If there is no corresponded value, the function will catch an exception.
Read Successfully:	The relevant data of MurhoAI lookup tables can be extracted correctly.
Read Unsuccessfully:	The relevant data of MurhoAI lookup tables can not be extracted correctly.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 3.2: Read MurhoAI Excel Sheet

Function Name:	readMurhoAI
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the relevant elements of MurhoAI Excel can be read and stored in a list.
Pre-Condition:	<ul style="list-style-type: none"> The MurhoAI Excel contains all elements which the MurhoAI function needs. The data type of elements is as same as required.
Read Successfully:	All the required elements are read and stored in the list.

Read Unsuccessfully:	The application will catch an error. And all processes of this read function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 3.3: Save MurhoAI lookup tables

Function Name:	saveMurhoAI
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the MurhoAI lookup tables can be saved on tables of the database
Pre-Condition:	<ul style="list-style-type: none"> The relevant MurhoAI information has been read successfully and stored in the required list. The database is connected.
Save Successfully:	The relevant MurhoAI information is saved on the murho_al tables of the database.
Save Unsuccessfully:	The relevant MurhoAI information is not saved on the murho_al table of the database. And all processes of this save function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC4: MurhoCu functions

The database relationship:

- murho_cu

TC 4.1: Read MurhoCu lookup tables

Function Name:	queryMurhoCu
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct mass-energy absorption coefficient for Copper (murho Cu) value is extracted using HVL Cu (mm Cu) value as input
Pre-Condition:	<ul style="list-style-type: none"> The user can extract the correct murho value corresponding to HVL Cu (mm Cu). If there is no corresponded value, the function will catch an exception.
Read Successfully:	The relevant data of MurhoCu lookup tables can be extracted correctly.
Read Unsuccessfully:	The relevant data of MurhoCu lookup tables can not be extracted correctly.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 4.2: Read MurhoCu Excel Sheet

Function Name:	readMurhoCu
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the relevant elements of MurhoCu Excel can be read and stored in a list.
Pre-Condition:	<ul style="list-style-type: none"> The MurhoCu Excel contains all elements which the readMurhoCu function needs. The data type of elements is as same as required.
Read Successfully:	All the required elements are read and stored in the list.

Read Unsuccessfully:	The application will catch an error. And all processes of this read function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 4.3: Save MurhoCu lookup tables

Function Name:	saveMurhoCu
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the MurhoCu lookup tables can be saved on tables of the database
Pre-Condition:	<ul style="list-style-type: none"> The relevant MurhoCu information has been read successfully and stored in the required list. The database is connected.
Save Successfully:	The relevant MurhoCu information is saved on the murho_cu tables of the database.
Save Unsuccessfully:	The relevant MurhoCu information is not saved on the murho_cu table of the database. And all processes of this save function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC5: Plane-parallel functions

The database relationship:

- chambers_list
 - beam_planeparallel_list
 - beam_planeparallel_chamber

TC 5.1: Query Planeparallel lookup tables

Function Name:	queryPlaneparallel
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct Nk (mGy/nC) value is extracted using the kVp (KV), HVL Al (mm Al), HVL Cu (mm Cu), and chamber SN values as input.
Pre-Condition:	<ul style="list-style-type: none"> The user can extract the correct Nk value corresponding to chamber SN, beam energy (kV), and measured HVL (Al/Cu). The user can extract two records with corresponding HVL (Al/Cu) values when The user measured cannot HVL extract (Al/Cu) any value record does not exist in the lookup table (for interpolation/extrapolation by the backend).
Read Successfully:	The relevant data of Planeparallel lookup tables can be extracted correctly.
Read Unsuccessfully:	The relevant data of Planeparallel lookup tables can not be extracted with the queryPlaneparallel function.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 5.2: Read Planeparallel Excel Sheet

Function Name:	readPlaneparallel
Test Type:	Functional
Execution:	Automated

Objective:	Verify that all the elements of Planeparallel Excel can be read and stored in a list.
Pre-Condition:	<ul style="list-style-type: none"> The Planeparallel Excel contains all elements which the readPlaneparallel function needs. The data type of elements is as same as required.
Read Successfully:	All the required elements are read and stored in the list.
Read Unsuccessfully:	The application will catch an error. And all processes of this read function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 5.3: Save Planeparallel lookup tables	
Function Name:	savePlaneparallel
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the plane-parallel information is saved on the beam_plane_parallel_list table of the database.
Pre-Condition:	<ul style="list-style-type: none"> The relevant Planeparallel information has been read successfully and stored in the required list. The database is connected.
Save Successfully:	The relevant Planeparallel information is saved on the beam_planeparallel_list table of the database.
Save Unsuccessfully:	The relevant Planeparallel information is not saved on the beam_planeparallel_list and beam_planeparallel_chamber table of the database. And all processes of this save function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC6: Pstem functions

The database relationship:

- chambers_list
- beam_planeparallel_list
- beam_planeparallel_chamber
- pstem_measured

TC 6.1: Query Pstem lookup tables	
Fucntion Name:	queryPstem
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the relevant data from the Planeparallel lookup tables can be extracted with measured HVL (Al/Cu), measured SSD (cm), and field size diameter (cm) from tables on the database
Pre-Condition:	<ul style="list-style-type: none"> The user can extract the correct Bw value corresponding to SSD (cm), diameter (cm), and HVL Al (mm Al). The user can extract two or more records with corresponding SSD (cm), diameter (cm), and HVL Al (mm Al) values when SSD (cm), diameter (cm), and HVL Al (mm Al) values do not exist in the lookup table (for interpolation/extrapolation by the backend).

Read Successfully:	The relevant data of Bw lookup tables can be extracted correctly.
Read Unsuccessfully:	The relevant data of Bw lookup tables can not be extracted with the queryBw function.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 6.2: Read Pstem Excel Sheet

Function Name:	readPstem
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct Pstem value is extracted using the Plane-parallel chamber HVL (mm Al) values as input.
Pre-Condition:	<ul style="list-style-type: none"> The user can extract the correct Pstem value corresponding to HVL Al. The user can extract two or more records with corresponding field size (cm), and HVL (mm Al) do not exist in the lookup table (for interpolation/extrap)
Read Successfully:	All the required elements are read and stored in the list.
Read Unsuccessfully:	The application will catch an error. And all processes of this read function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 6.3: Save Pstem lookup tables

Function Name:	savePstem
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the Pstem information is saved on the pstem_measured table of the database.
Pre-Condition:	<ul style="list-style-type: none"> The Pstem information is added to the pstem_measured table on the database.
Save Successfully:	The relevant Pstem information is saved on the pstem_measured table of the database.
Save Unsuccessfully:	The relevant Pstem information is not saved on the pstem_measured table of the database. And all processes of this save function will be recalled.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

Calculation

TC 7: Get a list of BackResults

Function Name:	getBackResult
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the calculation of back results is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.

Calculate Successfully:	The returned backresult list contains the correct result.
Calculate Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 8: Query the specific beam planeparallel Nk

Function Name:	queryBoundaryBeamPlaneparallelNk
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct Beam planeparallel Nk value is extracted using the chamber ID, and date as input.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are valid.
Query Successfully:	The returned result contains the correct planeparallel Nk example.
Query Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 9: Query the latest beam planeparallel Nk

Function Name:	queryLatestBeamPlaneparallelNk
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the late Beam planeparallel Nk value is extracted using the date.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Query Successfully:	The returned result is the late planeparallel Nk example according to the input date.
Query Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 10: Query the specific hvl

Function Name:	queryBoundaryHvl
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct Hvl value is extracted using the beam farmer ID as input.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters is valid.
Query Successfully:	The returned result contains the correct Hvl example.
Query Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 11: Query the latest farmer lookup table

Function Name:	queryLatestFarmerLookupTable
Test Type:	Functional

Execution:	Automated
Objective:	Verify that the latest farmer lookup table is extracted using the date.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Query Successfully:	The returned result is the latest farmer lookup table example according to the input date.
Query Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 12: Check whether there is data that has lower Hvl

Function Name:	checkHvlLower
Test Type:	Functional
Execution:	Automated
Objective:	Check whether there is data that has lower hvl.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Check Successfully:	The function returns true.
Check Unsuccessfully:	The function returns false.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 13: Check whether there is data that has higher Hvl

Function Name:	checkHvlUpper
Test Type:	Functional
Execution:	Automated
Objective:	Check whether there is data that has higher hvl.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Check Successfully:	The function returns true.
Check Unsuccessfully:	The function returns false.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 14: Query the Input cone data list

Function Name:	queryInputConeDataList
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the Cone data is extracted by using the audit ID.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Query Successfully:	The returned result is the input cone data list as specified cone id.
Query Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 15: Select data from the farmer table

Function Name:	selectFromFarmer
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the specified data can be extracted from the farmer table as required.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Select Successfully:	The returned result contains the farmer data information as the required type.
Select Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 16: Calculate Nk value

Function Name:	calculateNkValue
Test Type:	Functional
Execution:	Automated
Objective:	Calculate the nk value from the beam data list.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Calculate Successfully:	The returned result is the correct calculation.
Calculate Unsuccessfully:	The function returns a null or incorrect calculation.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 17: Query the latest BwAlCu data

Function Name:	queryLatestBwAlCu
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the latest bwalcu table data is extracted using the SSD, diameter, and hvl.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Query Successfully:	The returned result is the latest bwalcu data example according to the inputs.
Query Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 18: get the target boundary

Function Name:	getRange
Test Type:	Functional
Execution:	Automated
Objective:	Verify the target boundary.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Get Successfully:	The returned result is the correct target's boundary.

Get Unsuccessfully:	The function returns null.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 19: Calculate Bw value

Function Name:	calculateBwValue
Test Type:	Functional
Execution:	Automated
Objective:	Calculate the Bw value from the beam data list and con data list
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Calculate Successfully:	The returned result is the correct calculation.
Calculate Unsuccessfully:	The function returns a null or incorrect calculation.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 20: Calculate Bw value with specified inputs

Function Name:	bwCalculation
Test Type:	Functional
Execution:	Automated
Objective:	Calculate the Bw value according to the given cone, beam, and type inputs.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Calculate Successfully:	The returned result is the correct calculation.
Calculate Unsuccessfully:	The function returns a null or incorrect calculation.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 21: Calculate Murho value

Function Name:	calculateMruhoValue
Test Type:	Functional
Execution:	Automated
Objective:	Calculate the Bw value by giving the beam data list as input.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Calculate Successfully:	The returned result is the correct calculation.
Calculate Unsuccessfully:	The function returns a null or incorrect calculation.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 22: Calculate Murho value with specified inputs

Function Name:	murhoCalculation
Test Type:	Functional
Execution:	Automated
Objective:	Calculate the Murho value by given beam data, murhoal list, and murhocu list inputs.

Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Calculate Successfully:	The returned result is the correct calculation.
Calculate Unsuccessfully:	The function returns a null or incorrect calculation.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 23: Calculate ccc value

Function Name:	calculateCccValue
Test Type:	Functional
Execution:	Automated
Objective:	Calculate the ccc value by giving the beam data list, cones list, and bw results list as inputs.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Calculate Successfully:	The returned result is the correct calculation.
Calculate Unsuccessfully:	The function returns a null or incorrect calculation.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 24: Calculate pstem value

Function Name:	calculatePstemValue
Test Type:	Functional
Execution:	Automated
Objective:	Calculate the pstem value by giving the beam data list, cones list, and audit beam list as inputs.
Pre-Condition:	<ul style="list-style-type: none"> The input parameter is valid.
Calculate Successfully:	The returned result is the correct calculation.
Calculate Unsuccessfully:	The function returns a null or incorrect calculation.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

AuditController

TC 25: upload identification excel

Function Name:	upload
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether identification excel can be uploaded and saved into the database.
Pre-Condition:	<ul style="list-style-type: none"> The uploaded excel file is correct
Calculate Successfully:	The relevant information is stored in the database.
Calculate Unsuccessfully:	The identification information does not be stored in the database or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 26: Query audit cases

Function Name:	queryAuditCases
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct audit cases are extracted using the audit ID, clinic, status, date, page, and page size as input.
Pre-Condition:	<ul style="list-style-type: none"> The user can extract the correct audit cases corresponding to the input values.
Query Successfully:	The relevant audit cases can be extracted.
Query Unsuccessfully:	The relevant audit cases can not be extracted or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 27: Edit audit cases

Function Name:	editAuditCase
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the audit case status can be edited by the audit ID.
Pre-Condition:	<ul style="list-style-type: none"> The input audit ID and audit status are correct.
Edit Successfully:	The specified audit case status will be edited as required.
Edit Unsuccessfully:	The specified audit case status is not edited as required or the function catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 28: Delete audit case

Function Name:	deleteAuditCase
Test Type:	Manual
Execution:	Automated
Objective:	Verify that the correct audit case can be deleted according to the audit ID.
Pre-Condition:	<ul style="list-style-type: none"> The input audit ID is correct.
Delete Successfully:	The relevant audit cases can be deleted.
Delete Unsuccessfully:	The relevant audit cases can not be deleted or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 29: Query work datasheet1

Function Name:	queryWorkDataSheet1
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct beams data and chamber result can be gotten according to the audit ID.
Pre-Condition:	<ul style="list-style-type: none"> The audit ID is correct.
Query Successfully:	The relevant work datasheet information can be extracted.

Query Unsuccessfully:	The relevant work datasheet information can not be extracted or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 30: Query work datasheet2

Function Name:	queryWorkDataSheet2
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct basic data, beams data, and chamber results can be gotten according to the audit ID.
Pre-Condition:	<ul style="list-style-type: none"> The audit ID is correct.
Query Successfully:	The relevant work datasheet information can be extracted.
Query Unsuccessfully:	The relevant work datasheet information can not be extracted or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 31: Query beam information

Function Name:	queryBeamInfo
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct beam data can be gotten according to the audit ID.
Pre-Condition:	<ul style="list-style-type: none"> The audit ID is correct.
Query Successfully:	The relevant beam information can be extracted.
Query Unsuccessfully:	The relevant beam information can not be extracted or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 32: Query data from the frontend table

Function Name:	queryDataFrontEnd
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct frontend data can be extracted using the audit ID.
Pre-Condition:	<ul style="list-style-type: none"> The audit ID is correct.
Query Successfully:	The relevant frontend data information can be extracted.
Query Unsuccessfully:	The relevant frontend data information can not be extracted or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 33: Save data into input_data_front_end table

Function Name:	saveDataFrontEnd
Test Type:	Functional
Execution:	Automated
Objective:	Verify that input data can be saved into the input_data_front_end table according to the audit ID.

Pre-Condition:	<ul style="list-style-type: none"> The audit ID is correct and the data type is correct.
Save Successfully:	The input data can be saved into the table correctly.
Save Unsuccessfully:	The input data can not be saved into the table correctly or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 34: Query identification

Function Name:	queryIdentification
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct identification information can be extracted using the audit ID
Pre-Condition:	<ul style="list-style-type: none"> The audit ID is correct.
Query Successfully:	The relevant identification information can be extracted.
Query Unsuccessfully:	The relevant identification information can not be extracted or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 35: Query warnings

Function Name:	queryWarnings
Test Type:	Functional
Execution:	Automated
Objective:	Verify that the correct warning information can be extracted using the audit ID
Pre-Condition:	<ul style="list-style-type: none"> The audit ID is correct.
Query Successfully:	The relevant warning information can be extracted.
Query Unsuccessfully:	The relevant warning information can not be extracted or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

AuditService

TC 36: Read audit information in Excel

Function Name:	readAuditInfo
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the audit information can be read correctly from the Excel
Pre-Condition:	<ul style="list-style-type: none"> Read the correct Excel file
Read Successfully:	The relevant data of audit information can be extracted correctly.
Read Unsuccessfully:	The relevant data of audit information does not be extracted correctly or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 37: Read reference cone in Excel

Function Name:	readRefCone
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the refCone information can be read correctly from the Excel
Pre-Condition:	<ul style="list-style-type: none">• Read the correct Excel file
Read Successfully:	The relevant data of refCone information can be extracted correctly.
Read Unsuccessfully:	The relevant data of refCone information does not be extracted correctly or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 38: Read reference cone comments in Excel

Function Name:	readRefConeComments
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the refCone comment information can be read correctly from the Excel
Pre-Condition:	<ul style="list-style-type: none">• Read the correct Excel file
Read Successfully:	The relevant data of refCone comment information can be extracted correctly.
Read Unsuccessfully:	The relevant data of refCone comment information does not be extracted correctly or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 39: Read cone in Excel

Function Name:	readCone
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the cone information can be read correctly from the Excel
Pre-Condition:	<ul style="list-style-type: none">• Read the correct Excel file
Read Successfully:	The relevant data of cone information can be extracted correctly.
Read Unsuccessfully:	The relevant data of cone information does not be extracted correctly or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 40: Read beam quality in Excel

Function Name:	readBeamQuality
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the beam quality information can be read correctly from the Excel
Pre-Condition:	<ul style="list-style-type: none">• Read the correct Excel file
Read Successfully:	The relevant data of beam quality information can be extracted correctly.

Read Unsuccessfully:	The relevant data of beam quality information does not be extracted correctly or catch an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 41: Get the treatment machine information

Function Name:	getTreatmentMachine
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the treatment machine information is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned treatment machine information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 42: Get the facility information

Function Name:	getFacilityInformation
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the facility information is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned facility information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 43: Get the physical address

Function Name:	getPhysicalAddress
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the physical address is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned physical address is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 44: Get the reporting address

Function Name:	getReportingAddress
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the Reporting address is correct or not.

Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned Reporting address is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 45: Get the facility representative

Function Name:	getFacilityRepresentative
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the facility Representative's information is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned facility Representative information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 46: Get the audit information

Function Name:	getAuditInformation
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the audit information is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned audit information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 47: Get the cone

Function Name:	getCone
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the cone information is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned cone information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 48: Get the reference cone

Function Name:	getReferenceCone
----------------	------------------

Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the reference cone information is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned reference cone information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 49: Get the beam data

Function Name:	getBeamData
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the beam data is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned beam data is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 50: Get the beam inputs

Function Name:	getAuditBeamInputs
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the audit beam inputs information is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned audit beam inputs information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 51: Get the reference dosimetry

Function Name:	getReferenceDosimetry
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the reference dosimetry information is correct or not.
Pre-Condition:	<ul style="list-style-type: none"> The input parameters are correct.
Get Successfully:	The returned reference dosimetry information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 52: Get the kv reference dosimetry

Function Name:	getKvReferenceDosimetry
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the kv reference dosimetry information is correct or not.
Pre-Condition:	<ul style="list-style-type: none">• The input parameters are correct.
Get Successfully:	The returned kv reference dosimetry information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

TC 53: Get the audit history

Function Name:	getAuditHistory
Test Type:	Functional
Execution:	Automated
Objective:	Verify whether the audit history information is correct or not.
Pre-Condition:	<ul style="list-style-type: none">• The input parameters are correct.
Get Successfully:	The returned audit history information is the correct result.
Get Unsuccessfully:	The function returns the wrong result or catches an error.
Time constraint:	Minimum 1 seconds; Maximum 3 seconds

| Code Review Checklist

[[Maintainability](#)] [[Error Handling](#)] [[Security](#)] [[Thread Safeness](#)] [[Resource Leaks](#)] [[Control Structures](#)] [[Reusability](#)] [[Reference](#)]

Maintainability

1. Does the code make sense?
 - Make an effort to understand what the code is supposed to do before performing a code review.
 - Require the developer to comment as much as necessary to make the code readable.
2. Does the code comply with the accepted Java Conventions?
3. Does the code comply with the accepted Best Practices?
 - See Conventions
4. Does the code comply with the accepted Comment Conventions?
 - All classes and methods should contain a descriptive JavaDoc comment.
 - All methods should contain brief comments describing unobvious code fragments.
 - All class files should contain a copyright header.
 - All class files should contain class comments, including author name.
 - All methods should contain comments that specify input parameters.
 - All methods should contain a comment that specifies possible return values.
 - Complex algorithms should be thoroughly commented.
 - Comment all variables that are not self-describing.
 - Static variables should describe why they are declared static.
 - Code that has been optimized or modified to "work around" an issue should be thoroughly commented, so as to avoid confusion and re-introduction of bugs.
 - Code that has been "commented out" should be explained or removed.
 - Code that needs to be reworked should have a TODO comment and a clear explanation of what needs to be done.
 - When in doubt, comment.
 - When you've commented too much, keep commenting.
 - When your wrists hurt from commenting too much, take a break ... and then comment more.

Error Handling

1. Does the code comply with the accepted Exception Handling Conventions.
 - We need to expand our notion of Exception Handling Conventions.
 - Some method in the call stack needs to handle the exception, so that we don't display that exception stacktrace to the end user.
2. Does the code make use of exception handling?
 - Exception handling should be consistent throughout the system.
3. Does the code simply catch exceptions and log them?
 - Code should handle exceptions, not just log them.
4. Does the code catch general exception (java.lang.Exception)?
 - Catching general exceptions is commonly regarded as "bad practice".
5. Does the code correctly impose conditions for "expected" values?
 - For instance, if a method returns null, does the code check for null?
 - The following code should check for null

```
Person person = Context.getPersonService().getPerson(personId);
person.getAddress().getStreet();
```

- What should be our policy for detecting null references?

6. Does the code test all error conditions of a method call?

- Make sure all possible values are tested.
- Make sure the JUnit test covers all possible values.

Security

1. Does the code appear to pose a security concern?
 - Passwords should not be stored in the code. In fact, we have adopted a policy in which we store passwords in runtime properties files.
 - Connect to other systems securely, *i.e.*, use HTTPS instead of HTTP where possible.
2. Service methods should have an @Authorize annotation on them
3. JSP pages should use the openmrs:require taglib

Thread Safeness

1. Does the code practice thread safeness?
 - If objects can be accessed by multiple threads at one time, code altering global variables (static variables) should be enclosed using a synchronization mechanism (synchronized).
 - In general, controllers / servlets should not use static variables.

- Use synchronization on the smallest unit of code possible. Using synchronization can cause a huge performance penalty, so you should limit its scope by synchronizing only the code that needs to be thread safe.
 - Write access to static variable should be synchronized, but not read access.
 - Even if servlets/controllers are thread-safe, multiple threads can access HttpSession attributes at the same time, so be careful when writing to the session.
 - Use the volatile keyword to warn that compiler that threads may change an instance or class variable - tells compiler not to cache values in register.
 - Release locks in the order they were obtained to avoid deadlock scenarios.
2. Does the code avoid deadlocks?
- I'm not entirely sure how to detect a deadlock, but we need to make sure we acquire/release locks in a manner that does not cause contention between threads. For instance, if Thread A acquires Lock #1, then Lock #2, then Thread B should not acquire Lock #2, then Lock #1.
 - Avoid calling synchronized methods within synchronized methods.

Resource Leaks

1. Does the code release resources?
 - Close files, database connections, HTTP connections, etc.
2. Does the code release resources more than once?
 - This will sometimes cause an exception to be thrown.
3. Does the code use the most efficient class when dealing with certain resources?
 - For instance, buffered input / output classes.

Control Structures

1. Does the code make use of infinite loops?
 - If so, please be sure that the end condition CAN and WILL be met.
2. Does the loop iterate the correct number of times?
 - Check initialization and end condition to make sure that the loop will be executed the correct number of times.

Reusability

1. Are all available libraries being used effectively?
2. Are available openmrs util methods known and used?
3. Is the code as generalized/abstracted as it could be?
4. Is the code a candidate for reusability?

If you see the same code being written more than once (or if you have copied-and-pasted code from another class), then this code is a candidate.

Reference

*The resources are from: <https://wiki.openmrs.org/display/docs/Code+Review+Checklist>

Code Review for Sprint 2

Code Review for Sprint 2		
Time: 13th/April/2022-1st/May/2022		
Maintainability	Does the code comply with the accepted Best Practices?	Yes
	Does the code comply with the accepted Comment Conventions?	Yes
	Does the code make sense?	Yes
Error handling	Does the code make use of exception handling?	Yes
	Does the code simply catch exceptions and log them?	Yes
	Does the code catch general exception?	Yes
	Does the code correctly impose conditions for "expected" values?	Yes
	Does the code comply with the accepted Exception Handling Conventions.	Yes
Security	Does the code appear to pose a security concern?	Yes
Thread Safeness	Does the code practice thread safeness?	Yes
	Does the code avoid deadlocks?	Yes
No Resource Leaks	Does the code release resources more than once?	Yes
	Does the code release resources?	Yes
	Does the code use the most efficient class when dealing with certain resources?	Yes
Control Structures	Does the code make use of infinite loops?	No
	Does the loop iterate the correct number of times?	Yes
Reusability	Are all available libraries being used effectively?	Yes
	Is the code as generalized/abstracted as it could be?	Yes
	Is the code a candidate for reusability?	Yes

Code Review for Sprint 3

Code Review for Sprint 3		
Time: 1st/May/2022-30th/May/2022		
Maintainability	Does the code comply with the accepted Best Practices?	Yes
	Does the code comply with the accepted Comment Conventions?	Yes
	Does the code make sense?	Yes
Error handling	Does the code make use of exception handling?	Yes
	Does the code simply catch exceptions and log them?	Yes
	Does the code catch general exception?	Yes
	Does the code correctly impose conditions for "expected" values?	Yes
	Does the code comply with the accepted Exception Handling Conventions.	Yes
Security	Does the code appear to pose a security concern?	Yes
Thread Safeness	Does the code practice thread safeness?	Yes
	Does the code avoid deadlocks?	Yes
No Resource Leaks	Does the code release resources more than once?	Yes
	Does the code release resources?	Yes
	Does the code use the most efficient class when dealing with certain resources?	Yes
Control Structures	Does the code make use of infinite loops?	No
	Does the loop iterate the correct number of times?	Yes
Reusability	Are all available libraries being used effectively?	Yes
	Is the code as generalized/abstracted as it could be?	Yes
	Is the code a candidate for reusability?	Yes

| Development

This page houses documentation related to the development aspect of Software application for the ACDS kV Level 1b audit program.

[Development Tools](#)

[Deployment Plan](#)

[Handover](#)

[User Guide](#)

Development Tools

Listed on this table are the tools and technology we have chosen to support our software development, testing, and to increase our collaboration efficiency.

[Collaboration Efficiency Tools] [Development Tools] [Testing Tools]

Collaboration Efficiency Tools	
Tools	Description
Confluence	We use Confluence to create, collaborate, and organize our project artefacts, ideas and plan to achieve our common goal.
Slack	We use Slack to communicate effectively as a team, coordinate with our supervisor, and initiate short discussions and arrange meetings with the clients. It is a space where all three groups can communicate and collaborate, and where each group have their own space to discuss among members and do a daily standup.
Zoom	For more formal and detailed discussions, such as meetings with the clients, we use Zoom to allow a face-to-face meeting that can be recorded and be viewed by all members, especially by those who were not able to attend.
CloudStor	We use CloudStor to share large files such as the recorded meetings, and files that may contain sensitive information not well-suited to be uploaded or shared on Confluence.
GitHub	Since all three groups will be working together to build one software application for the clients, all groups share one repository on GitHub, where each group can collaborate and develop the components of the software in their own branch.
LucidChart	This tool is utilised to collaboratively create diagrams (for example Entity Relationship Diagram) among team members.
draw.io	This tool is utilised to create diagrams such as the motivation model.
Trello	This tool is utilised to manage tasks allocation and supervise the progress and completion of tasks.

Development Tools	
Tools	Description
Microsoft Azure	<p>For development purposes, we are using Azure Virtual Machine to host the database and the application.</p> <p>Current specification of Azure Virtual Machine:</p> <ul style="list-style-type: none"> • Machine Type: Standard B2S (2 vCPUs, 4 GB memory) • Disk: 30GB • OS: Centos 7.9 • Region: Australia Southeast
Microsoft SQL Server	Due to security issues and as agreed with the client, we have decided to implement Microsoft SQL Server as the primary database. Accessing the database will be via SQL Server Management Studio (SSMS).
Java	Python will be the base language used by all teams to develop the application per the request of the clients. The APIs to link to the database will also be developed using this language. The backend part of the software will be on the Java framework Spring Boot .
HTML JavaScript CSS	As the requirement of building a web site as GUI, these programming language will be used.
Vue	The framework Vue CLI which will be used in building web pages.

Testing Tools	
Tools	Description
Automated Unit Testing	<p>Software will be tested with Java built-in 'JUnit' framework, which helps with creating and running unit tests.</p> <p>Automation of unit testing will be using GitHub Action as part of Continuous Integration workflow.</p>
Manual Acceptance Testing (User Acceptance Testing)	We will conduct manual acceptance tests before the end of each sprint.

Decisions

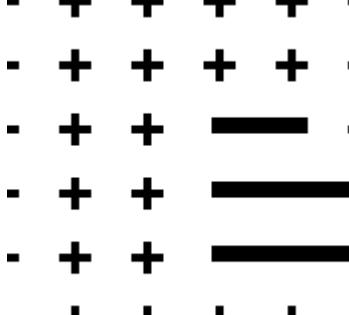
About

This page is dedicated to listing important decisions, as well as their outcomes, made by one or more stakeholders in the project.

Decisions

Record important project decisions and communicate them with your team.

[Create decision](#)



Decision log

Decision	Status	Stakeholders	Outcome	Due date	Owner
Architecture Decisions	DECIDED	@RedbackTeam	Front-end: Vuetify Back-end: Spring Boot	23 Mar 2022	@RedbackTeam
Two Team Cooperation Method	DECIDED	@RedbackTeam	Parallel Development	18 Mar 2022	@RedbackTeam

Two Team Cooperation Method

Status	DECIDED
Stakeholders	@RedbackTeam
Outcome	Parallel Development
Due date	18 Mar 2022
Owner	@RedbackTeam

Background

There are Redback and Boxjelly team in a group to develop the KV project. We have to decide what cooperation method between two teams.

Options

Methodology	Advantages	Disadvantage
Parallel Development	Technology stack of development team members to meet development needs Teams are not affected by each development stage	Increased content developed by each team
Division based on Technical Stack	Separate front-end and back-end development More focused on technology stack development	Require a middle-man to adult team's code Sprint must be submitted as a team
Division based on Function	More features can be developed Duplicate file content can be shared	Require a middle-man to adult team's code Sprint must be submitted as a team

Architecture Decisions

Status	DECIDED
Stakeholders	@RedbackTeam
Outcome	Front-end: Vuetify Back-end: Spring Boot
Due date	23 Mar 2022
Owner	@RedbackTeam

Background

Before the start of sprint 2, the technology stack and interaction method used by the front-end and back-end need to be decided.

Options for Front-end

Methodology	Advantages	Disadvantages
Element ui	Flexible configuration.	The previous template is complex. Third-party libraries is often required
Vuetify	Easy to setup. More familiar by team developer.	Debugging is difficult.

Options for Back-end

Methodology	Advantages	Disadvantages
Flask	Lightweight. Flexible configuration.	Simple function. Third-party libraries is often required.
Spring Boot	Powerful and handy. Easy setup.	Debugging is difficult.

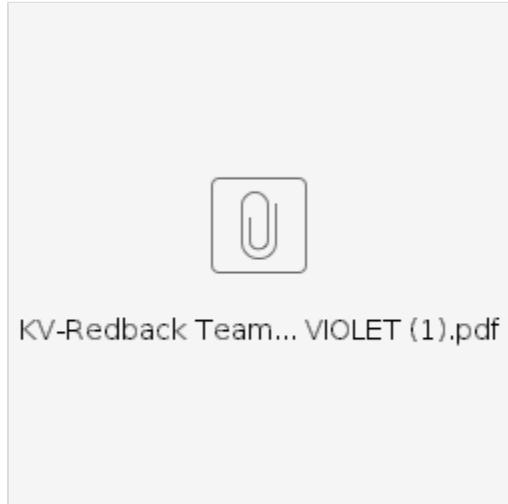
Action items

- Decide on the technology stack used by the backend. Yipeng Li
- Decide on the external libraries and versions used by the backend. Yipeng Li
- Initialize the backend project. Yipeng Li
- Decide on the data format for front-end and back-end interactions. Yipeng Li Fangtai DONG

| Presentation

This section is a collection of the application presentation, including the slides and recorded video.

Presentation Slides



Presentation Recording



CloudStor (59.7MB)

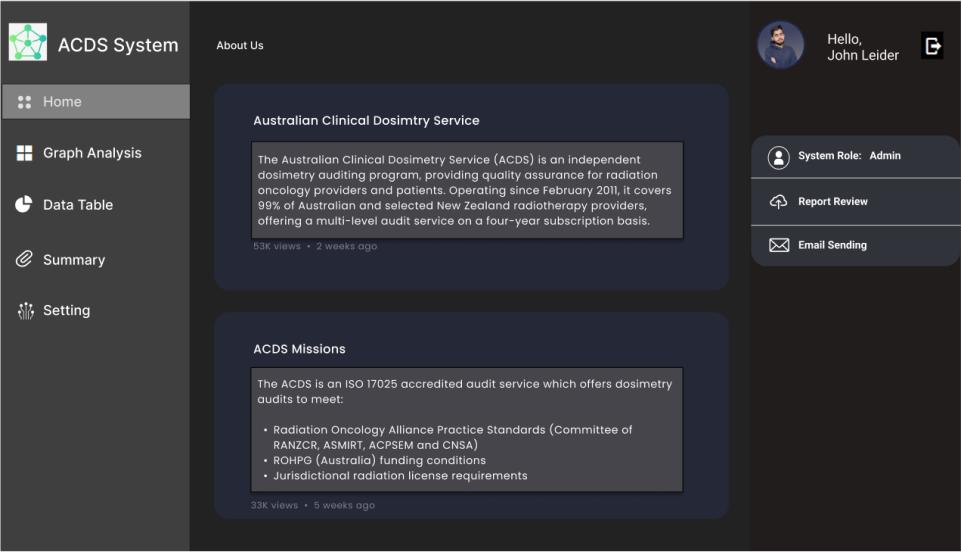
- com90082 final presentations KV-redback.mp4

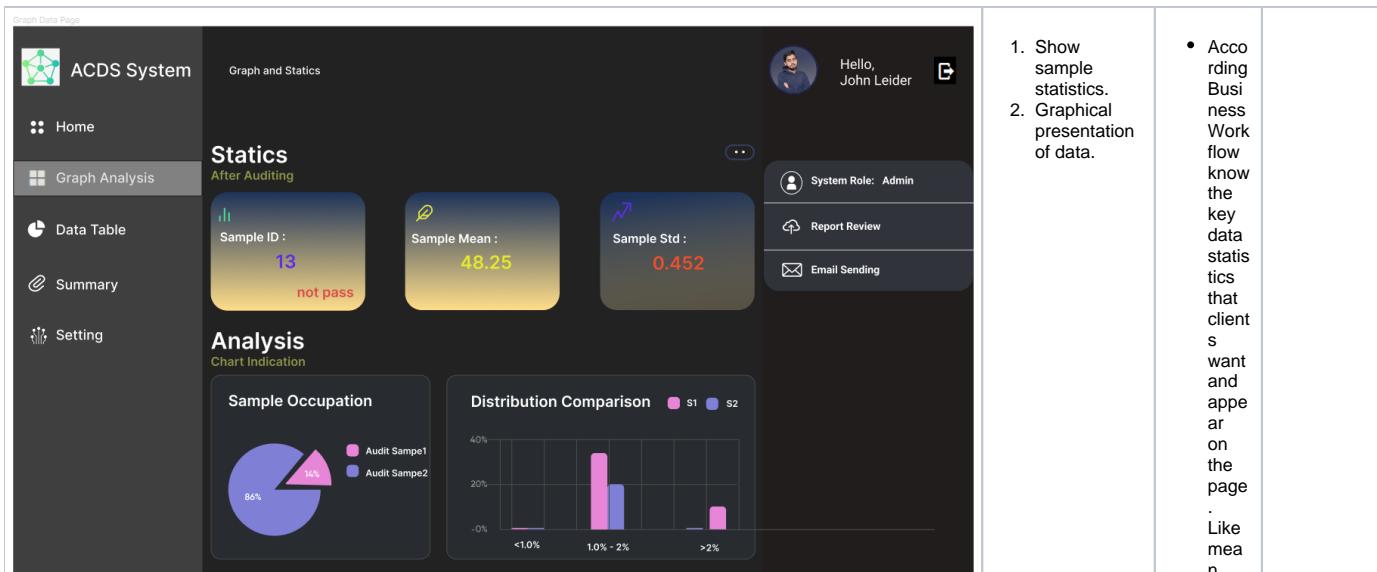
The password for this recording is **comp90082KV***

| Draft

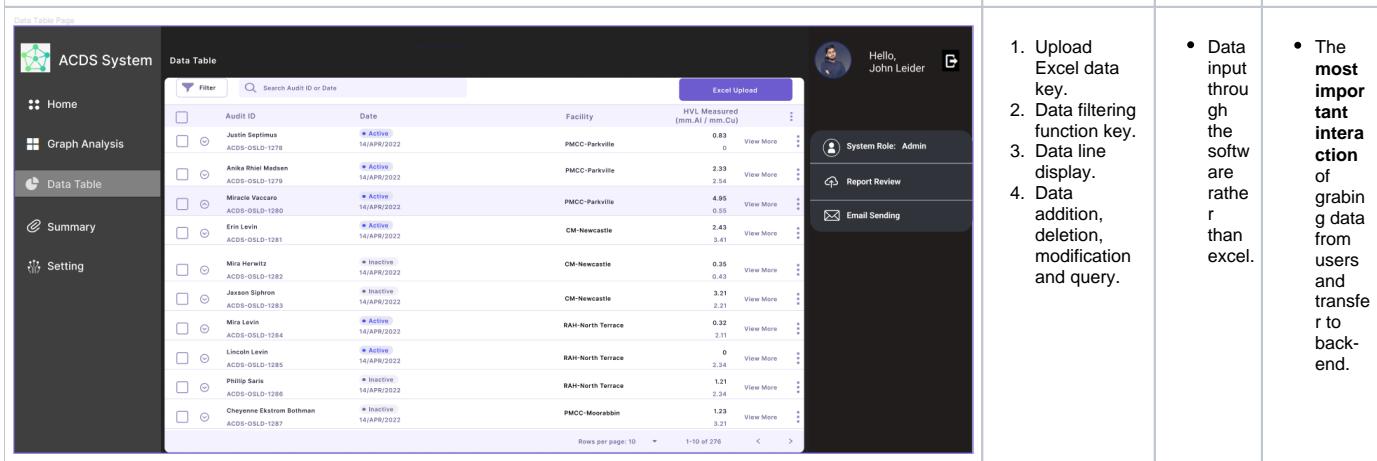
This section includes the draft of some pages and will not use in our project.

Digital Prototype

Application Page	Effect	Meet Clients Needs	Remarks
 <p>The screenshot shows the ACDS System application page. On the left is a sidebar with icons for Home, Graph Analysis, Data Table, Summary, and Setting. The main content area has a dark header "About Us". Below it is a card titled "Australian Clinical Dosimetry Service" with a description: "The Australian Clinical Dosimetry Service (ACDS) is an independent dosimetry auditing program, providing quality assurance for radiation oncology providers and patients. Operating since February 2011, it covers 99% of Australian and selected New Zealand radiotherapy providers, offering a multi-level audit service on a four-year subscription basis." It shows 53K views and was posted 2 weeks ago. Another card below it is titled "ACDS Missions" with a description: "The ACDS is an ISO 17025 accredited audit service which offers dosimetry audits to meet: Radiation Oncology Alliance Practice Standards (Committee of RANZCR, ASMI RT, ACPSEM and CNSA), ROHPG (Australia) funding conditions, Jurisdictional radiation license requirements." It shows 33K views and was posted 5 weeks ago.</p>	<ol style="list-style-type: none"> 1. Introduce company background. 2. Introduce the company's audit standards and compliance. 	<ul style="list-style-type: none"> • Visitors, Users or Admins know what ACDS business. • Know what industrial standards they need to meet. • Good to have business workflow shown on page. 	<ul style="list-style-type: none"> • Right sidebar can show role of what people login.



1. Show sample statistics.
 2. Graphical presentation of data.
- According Business Work flow know the key data statistics that clients want and appear on the page . Like mean and Std.
 - Use pie chart and bar chart to make data analysis between two samples or the pass samples and non-pass samples distribution.



1. Upload Excel data key.
 2. Data filtering function key.
 3. Data line display.
 4. Data addition, deletion, modification and query.
- The most important interaction of grabbing data from users and transfer to back-end.

Summary Page

QC DATA

Optimal - no action required (between QC limit 1 and 2)
Sub-optimal - check between QC limit 2 and 3)
Poor - Immediate action required (larger than QC limit 3)
Overly-optimal - check (less than QC limit 1)

limit multiplier for mean	0	0.4	0.8					
limit multiplier for std	0.7	1.3	1.5					
Range 1	All	40 - 80	80 - 120	120 - 160	160 - 200	200 - 240	240 - 280	280 - 320
QC Limit 1	-0.24%	0.02%	-0.73%	0.07%	-0.09%	-0.02%	0.48%	-0.57%
QC Limit 2	0.69%	0.67%	0.91%	0.69%	0.83%	0.47%	0.08%	0.46%
QC Limit 3	OK	OK	real offset	too good	OK	too good	low stats	low stats
	153	81	51	19	6	11	2	2

Report Review

Email Sending

Review Report

Email **Send**

Hello, John Leider

System Role: Admin

1. Results summary presentation.

2. Generate presentation Report.

3. Enter the acceptance email and click send summary report to the corresponding email.

- Summary generation
- Send summary report to specific email address.

Page-2

Create your account

Email

Username

Password

Confirm password

Remember me

Create your account

1. The user jumps to the registration page.

- New user registration.

LOGIN

Username

Password

Remember me

LOGIN

Forgot Password ?

Need an account? [SIGN UP](#)

1. User login interface and new user jump registration.
2. Given the fixed initial user password of the administrator, modify the password on the setting page inside the system. The administrator logs in with the set initial account password, and the input value matches the original account password of database management . If it is equal, it is allowed to enter

- Different roles login page.

Requirements Elicitation (UI)

ID	Client requirements	Features elicitation	Dev comprehension (priority 1-3: from highest to lowest) (Front-End perspective) for UI Design	Supplementary remarks
1	ARPANSA_FORM Data need to be incorporated into the software Application.	Responsive application		
2	Analyses it by means of graphs and statistics .	Data analysis page	Priority:1 Mean Std Compare	
3 (prev)	Excel spreadsheet upload and transfer to DB.	Data Column: Date; Audit ID: Facility: kV Unit ID: Farmer dose: PP Dose: Farmer Var(%): PP Var(%): Cone: Field Size: SSD: Beam/Filter: kVp: HVL(mm.Al) Nominal: HVL(mm.Cu) Nominal: HVL(mm Al) Measured: HVL(mm. Cu) Measured: Fac Dose: KE new: KE old: Sens standoff corr: OSDL standoff: ISLcorr: Oringinal ACDS dose: Above are basic calculative attributes	Priority: 1 Audit-service procedures. Table page with each row CRUD function Synchronize to database Each column ascending or descending sort (or single page database manipulating) Excel document uploading	SQLAlchemy -- backend-DB POST request Table html with form
4	what we want is to incorporate this "summary" spreadsheet into the app and actually GET RID OF THIS spreadsheet!	Calculate summary sheet and appear page.	Priority:1 Summary page	
5				
6	convert this and have the DB running off a single PC hard drive that the staff access?	Local DB --> python RSA encryption	Priority:2 Security lock --> encryption for DB --> block all Get Requests.	

7	Security is part of the project - I would like to know your suggestions? "security is a HUGE issue:"	SECURITY suggestions for this App 1: for Web --> protection of CSRF --> token mechanism request instead of cookies request 2: for Local --> RSA cyber encryption mechanism --> secret key setup 3: for Software --> roles management	Priority:1 Login page Register page Authentication	
8 (Prev.)	ACDS - 1: checking testing 2: verifying ionizing radiation therapy 3: verifying treatment machines	Introduction page	Priority:2 Introduction page(index page)	
9 (Prev.)	Check radiation dose patients receive for their cancer treatment is correct (<=2%)	Summary filter list for: a. clinics pass b. clinics not pass	Priority:1 Range display page	
10 (Prev.)	We independently check these machines at the clinics. We audit them.	Clinics upload data. Audit analysis by graph and summary. (with roles separate) Audit admin Audit users	Priority:3 Client (visitor role) excel (data) file upload for auditor analyzing page	
11 (Prev.)				
12 (Prev.)	Audit what? Low-energy ionizing radiation machines (for skin cancers) Theory on Half Value Layer	Machine selection page	Priority:3 extending function for machine categorization.	
13 (Prev.)	Detailed comments for each feature.	Code function description		
14 (Prev.)	The reason why these spreadsheets have so many cells is because in order for us to calculate and interpolate the values we need, we need to filter and find the information in Excel to use properly. So really – when it comes to just coding it, it's actually really simple! (or should be).	Multi-layers Filter. Or meta search		
15 (Prev.) (Bonuses)	After auditing clinic --> Audit Team then provide the clinic with an official report to say whether or not they are okay to continue treating patients. The report tab is used to generate our documentation and official report that we send to the clinic.	Generate calculated summary report(pdf) for clinics. Email reports to clinic.	Priority:1 Form(html) button for sending report by email.	
16 (Prev.) (Bonuses)	This has the results and calculations listed that we performed during the clinic's audit and also contains our signatures.	1. Report view online then 2. Signatures create.	Priority:2 Browse report and encrypt signature.	
17 (Prev.) (Bonuses)	Data take from database and front-end, then do calculations in back-end, then inject them to report. "It references or takes the data from the yellow and blue tabs and puts them into the report (remember the green tabs are all background calculations)."			

| Testing Plan

The testing plan section will describe how the tester will verify the system works as required.

There are two directions for testing: front-end testing and back-end testing.

Front-end Testing Plan

UC01 - If users already have an account, they can type the account username as well as the password to log in	
Test Type:	Functional
Execution:	Manually
Objective:	Users can log in to the website with the correct username and password
Pre-Condition:	User input their username and password in the boxes and correctly.
Login successfully:	The system verifies the identity of users from the user table in the Database. If users insert the correct username and password, they can access the home page.
Login unsuccessfully:	After the user clicks the login bottom, the error message pops out. Show "the username and password do not match" message.
Time constraint:	Minimum 20 min; Maximum 25 min

UC02 - If users do not have an account, they can sign up to create a new account	
Test Type:	Functional
Execution:	Manually
Objective:	Users can sign up for their own user accounts on the sign-up page.
Pre-Condition:	The user clicks the sign-up button. The user creates a new account and password. The information should be unique and follow the basic rules.
Sign-up successfully:	The user gets the success message. The system jumps back to the log-in page.
Sign-up unsuccessfully:	After the user clicks the sign-up button, the error message pops out. Show "sign up fails " message.
Time constraint:	Minimum 20 min; Maximum 25 min

UC03 - If users log in, the home page shows all cases and users can select them to view details	
Test Type:	Functional
Execution:	Manually
Objective:	Users can view all cases information on the home page.
Pre-Condition:	The user login to the system.
Show successfully:	The user can view all cases and related information on the home page. The system access all case information from the database
Show unsuccessfully:	After the user clicks the sign-up button, the error message pops out. Show "sign up fails " message.
Time constraint:	Minimum 20 min; Maximum 25 min

UC04 - Users can use select their preferred filter and use the input or poped box to manage their case list	
Test Type:	Functional
Execution:	Manually
Objective:	Users can filter and search any case on conditions.
Pre-Condition:	The user login the system. The home page shows all cases.

Select successfully:	The user selects or inputs the filter conditions in the boxes. The system sends a request to access the filtered data from the database.
Select unsuccessfully:	The system cannot send a request to get all case information.
Time constraint:	Minimum 20 min; Maximum 25 min

UC05 - Users can view the updated case list after they clicked the search button

Test Type:	Functional
Execution:	Manually
Objective:	Users can view selected cases after implementing the search function.
Pre-Condition:	The user login the system, The users select the filter condition and click the search button. The users click the edit button in the operation column.
View successfully:	The case table updates the data. All cases in the table follow the search conditions.
View unsuccessfully:	The case table does not update the data. All cases in the table do not follow the search conditions. No cases in the table.
Time constraint:	Minimum 20 min; Maximum 25 min

UC06 - Users can upload an excel file from local storage by clicking an upload button

Test Type:	Functional
Execution:	Manually
Objective:	Users can upload an excel file from their devices.
Pre-Condition:	The user login the system. The user opens the identification page and clicks the upload button.
Upload successfully:	The system opens a UI to upload any files to the devices. The website sends an FTP request to the backend or database to store the file.
Upload unsuccessfully:	The system cannot open a UI to upload any files to the devices. The website cannot send an FTP request to the backend or database to store the file.
Time constraint:	Minimum 20 min; Maximum 25 min

Back-end Testing Plan

UC07 - Calculate Nk for Farmer-type chamber

Test Type:	Functional
Execution:	Automated
Objective:	Input beam id and information. Output the Nk result based on Farmer-chamber measurements.
Pre-Condition:	Input must have HVL reading. The look-up table is stored in the database. Input HVL within the range specified by KVP in the look-up table (non-extrapolation). Nk values for chosen Farmer-type chamber exist in the look-up table.
Step:	Connect to the database. Select input (beam ID, HVLs, KVP) from the database. Import the look-up table from the database. Execute the program to obtain Nk corresponding to different Farmer-type chambers.
Calculation successfully:	Conduct a successful program execution with normal values and no exceptions.
Calculation unsuccessfully:	Execute the program normally but feed it with an invalid lookup table. Check that the correct exception is thrown and the program exits.
Time constraint:	Minimum 5 min; Maximum 15 min

UC08 - Calculate Nk for Plane-parallel chamber

Test Type:	Functional
------------	------------

Execution:	Automated
Objective:	Input beam ID and information, output the Nk result based on Plane-parallel chamber measurements.
Pre-Condition:	<p>Input must have HVL reading. The look-up table is stored in the database. Input HVL within the range specified by KVP in the look-up table (non-extrapolation). Nk values for chosen Plane-parallel chamber exist in the look-up table.</p>
Step:	<p>Connect to the database. Select input (beam ID, HVLs, KVP) from the database. Import the look-up table from the database. Execute the program to obtain Nk corresponding to different Plane-parallel chambers.</p>
Calculation successfully:	Conduct a successful execution of Nk calculated for Planeparallel chambers.
Calculation unsuccessfully:	Execute the program normally but feed it with an invalid lookup table. Check that the correct exception is thrown and the program exits.
Time constraint:	Minimum 5 min; Maximum 15 min

UC09 - Calculate Nk using extrapolation for Farmer-type chamber

Test Type:	Functional
Execution:	Automated
Objective:	Given an input HVL that is out of bounds, do extrapolation to this value for Farmer-type chamber only.
Pre-Condition:	The lookup table is complete and is stored in the database prior to program execution. For each input beam, the measured HVL is either greater than the largest HVL value in the lookup table or less than the smallest HVL value in the lookup table, given that the KVP is the same.
Step:	<p>Connect to the database. Import lookup table for interpolation/extrapolation. Input beam HVL that is out of bounds according to the lookup table. Execute the program to obtain Nk calculated by extrapolation.</p>
Calculation successfully:	Successfully calculate the Nk for Farmer-type chambers by using the extrapolation method.
Time constraint:	Minimum 5 min; Maximum 15 min

UC10 -View and retrieve results for Nk

Test Type:	Functional
Execution:	Automated
Objective:	After successful execution of Nk calculation, query the calculated Nk values from the database.
Pre-Condition:	The database is clean (have no previous calculation results stored). Input data is within the spec according to the lookup table.
Step:	<p>Connect to the database. Verify lookup table correctness. Select input (beams, HVLs, KVPs) from the database. Execute the program and verify no exception is thrown. Query the database for Nk results. Verify results correctness.</p>
Query successfully:	Perform a clean execution of the program and gather stored Nk results from the database. Verify that the result is correct.
Time constraint:	Minimum 10 min; Maximum 30 min

UC11 - Retrieve look-up table for Plane-parallel chamber

Test Type:	Functional
Execution:	Automated
Objective:	When calculating Nk for Plane-parallel chambers, retrieve the look-up table from the database (Look-up table may be subject to change).

Pre-Condition:	The lookup table for the Plane-parallel chamber is ready for retrieval.
Step:	<p>Connect to the database.</p> <p>Verify that the lookup table is intact and ready for retrieval.</p> <p>Add debugging clause at select_from_plane_parallel from nk_value.</p> <p>Select input data from the database.</p> <p>Execute the program normally and verify the debugging output.</p> <p>for successful retrieval of the lookup table.</p>
Retrieval successfully:	Successfully retrieve the lookup table for Plane-parallel chambers.
Time constraint:	Minimum 5 min; Maximum 15 min

UC12 - Calculate Bw when hvl, ssd, the diameter can be found on the lookup table	
Test Type:	Functional
Execution:	Automated
Objective:	Input hvl, ssd and diameter. Output the Bw result.
Pre-Condition:	The look-up table is stored in the database. Hvl, ssd, and diameter can be found in the lookup table.
Step:	<p>Connect to the database.</p> <p>Select input (hvl, ssd, diameter) from the database.</p> <p>Import the look-up table from the database.</p> <p>Execute the program to obtain Bw value.</p>
Calculation successfully:	Conduct a successful program execution with normal values and no exceptions.
Time constraint:	Minimum 5 min; Maximum 15 min

| Research

This section contains documentation regarding our understanding of the data involved - what it is, how we obtain it, and how we can derive from it. This is meant to be an extension to our requirements, and will be complementary to our development.