COMP9034-DevOps Team 4

Links
DevOps Risks Issue log, Rodmap

DevOps
Sprint

User Story 1.5

Figma

Lean Next js (watch first 20 min)

Testing Strategy (Sprint 2 updated)
Test Cases

Riscks/Issues log

Role

- Product Owner: Andy
- Architect : Taka
- Business Analyst (user story, how deep, how detail) (use case, diagram): Max (Leader), Alyssa
- Project Manager : Lola
- DevOps Engineer : Tan
- Scrum Master : Taka (developer/tester)
- Developers :
  - First stage- Front end design: Kong (Leader), Luck, Tan
    - UI: Min, Alyssa, Jiaqi, Kong (Luck, Tan, Taka)
    - Database, dataflow (function): Tan
- Testers : Jiaqi (Leader), Max

Roadmap

Sprint 0			
date: week1 tasks: ✓ 1. Allocated roles 2. Decide on development tools/languages	date: week2 tasks: ✓ 1. Create DevOps environment(Database, Task board, Git, Testing) - **Lola, Taka** 2. Learn the technology, Select Libraries, agree on coding conventions: **Developers** 3. (After user stories come out) - Research chemical Types- **Tan** Dangers, Variety?, - Locations -> Facilities(building) -> Lab(Room No.) - Regulations *** - Procedures *** 4. Create Risks and Issues log - The project (Group) **Lola** **https://docs.google.com/spreadsheets/d/14UlNox_ihK3XZPykbtP6jPH3qO4OKM8e_E3bKHUnzxA/edit?usp=sharing** 5. Add Epics and User Stories and tasks for Sprint Zero and Sprint 1 - **Alyssa, Max**	date: week 3 tasks: ✓ 1. Set up boards for Sprint Zero and Sprint 1 2. Recording template - Taka	date: week 4 tasks: 1. Determine testing strategies - Thursday 2. Create test plans for Sprint 1 3. Create UI projects and develop simple one page application to display Application details. **4. ER diagram (Database design, SQL code to create tables, data dictionary)-Taka, Developer team**

Sprint 1 **Duration: Now until Week 6**		
Epic 2: User Management and Security Epic 5: Administration		
date: week4 tasks: 1. UI design for Epic 2 - UI & Dev team 2. Data Dictionary - Architect	date: week 5 tasks: 1. Develop (build) Epic 2. 2. UI design for Epic 5 - UI & Dev team	date: week 6 tasks: 1. 2. 3.

& Dev team 3. ERD Design - Architect & Dev team	3. Develop Epic 5	

Sprint 2 **Duration: Week 7 - Week 8**	
Epic 1: Chemical Request and Approval process Epic 3: Ordering and Receiving Management Create the report structure - Architect	
date: week 7 tasks: 1. 2. 3.	date: week 8 tasks: 1. 2. 3.

Sprint 3 **Duration: Week 9 - Week 10**	
Epic 4: Storage and Inventory Management	
date: week 9 tasks: 1. 2. 3.	date: week 10 tasks: 1. 2. 3.

Sprint **Duration: Week 11 - Week 12**	
Report and Presentation	
date: week 11 tasks: 1. 2. 3.	date: week 12 tasks: 1. Presentation 2. Submit the report

Kong
Sprint 1:
Epic 2: User Management and Security
Epic 5: Administration
Duration: Now until Week 6.

Sprint 2:
Epic 1: Chemical Request and Approval process
Epic 3: Ordering and Receiving Management
Create the report structure - Architect
Duration: Week 7 - Week 8

Sprint 3:
Epic 4: Storage and Inventory Management
Duration: Week 9 - Week 10

Stack

Library:

Fullstack Framework: NextJS14(React, TypeScript)
CSS: Tailwind ShadCn
ORM: Drizzle
Auth: NextAuth 5(Okta, Google, Credentials=> only for dev)
Package manager: NPM
State Management: Zustand(only when the project seems complex)
Testing: Jest
Container: -

UI design: Figma
Version control: GitHub, Azure DevOps
Deployment: Vercel
Database: Vercel Postgres
Task Management: Azure DevOps
IDE: VScode

Login Username a, Password a
Development environment URL
https://comp-9034-dev-ops.vercel.app/login

Testing environment URL
https://comp-9034-dev-ops-test-takas-projects-2b6e7a41.vercel.app/login

Azure DevOps
https://dev.azure.com/COMP9034DevOpsTeam4/COMP9034DevOps/

Github Repo :
https://github.com/rhapvn/COMP9034DevOps

Role Description

No.	Roles	Definition	Tasks in Project	Suggestion of used technologies
1	**Architect**	Designs and oversees the implementation of DevOps strategies and processes, ensuring alignment with organizational goals and objectives.		Cloud Service Provider: **Azure** UML tools: **Draw.io**
2	**Business Analyst**	Business Analyst is up front and center in designing requirements based on both customers' needs and changing markets. You spend much of your time analyzing needs, designing requirements, and estimating time and costs associated with getting new software releases "out the door" and into the hands of the customers.		Requirements gathering, user stories, documentation: **Jira, Google Docs** Flowcharts and diagrams creation: **Draw.io** Prototyping tools: **Figma, Miro** Communication**: MS Teams, Slack**
3	**Project Manager**	DevOps Project Managers coordinate multiple teams and contributors, tracking timelines and dependencies to ensure a smooth project flow from development to production. They play a crucial role in aligning the project with the overall goals of the organization designing strategies to integrate various teams into a collaborative DevOps environment. The emphasis is on continuous delivery and integration, focusing on optimizing the process rather than a single end goal.		Project Management and tracking tools: **Jira** Project Planning and Scheduling tools: **Microsoft Project, MS Outlook** Team communication: **MS Teams, Slack** Documentation and Collaboration: **Google Docs** Budgeting tools: **Google Excel (optional)**
4	**DevOps Engineer**	Collaborates with developers and IT staff to manage code releases, merging the barriers between software development, quality assurance, and IT operations while keeping an eye on swift deliveries and deployment.		Source code repositories: **Github** Environment managements: **Azure (Must prepare DEV, TEST, PRODUCTION environments)** Monitoring and logging tools: **Grafana, Elastic** Continuous Integration/Continuous Deployment (CI/CD) tools: **Jenkins, CircleCI, Travis CI**
5	**Scrum Master**	The product owner submits a request and the development team (led by a Scrum Master) breaks it down into smaller pieces, also known as "sprints." Sprints consist of reiterative and collaborative development and testing procedures in a fast-paced environment and, ideally, create a more efficient product lifecycle. By using agile Scrum methodologies effectively, companies can produce a viable deliverable in two to four weeks. The Scrum		Sprints management, backlogs, and progress tracking: **Jira, Azure Devops** Online standups and meetings: **MS Teams**

		Master is at the center of it all by coordinating project activities with business objectives.		
6	**Developers**	Focuses on writing code and developing applications, while also incorporating DevOps practices into the development process.		Front-end technologies: **React, TailwindCSS, Javascript.** Backend technologies: **Nodejs, ExpressJs, TypeORM** Database: **MySQL** Dev Tools: **VSCode, Figma, Miro** Version Control System: **Github**
7	**Testers**	Testers always think about requirements that must be meet. Testers be focused on every change and their effect on developing products. In order to do this, requirements analysis done by testers. Testers find out test scope. QA team should be focused about quality in software development life-cycle. Testers be part of technical teams. Testers have to consider manual functional and non-functional testing, and should focus on their skills and experiences for apply test automation, and define test strategies.		Automated testing frameworks: **Selenium** API testing: **Postman** Bug Logging and tracking tools: **Jira** Manual Testing: **User Interface (Front-end website)**

GitHub Accounts

Taka: rhapvn
Keng: TanapongAUS
Kong: leekongdev
Luck: SmithPH
Alyssa: alyssaliuzz
Min: ChealseaYao
Max: muskrjwq
Jacky: JackyChn
Lola:Hsin89

Requirement

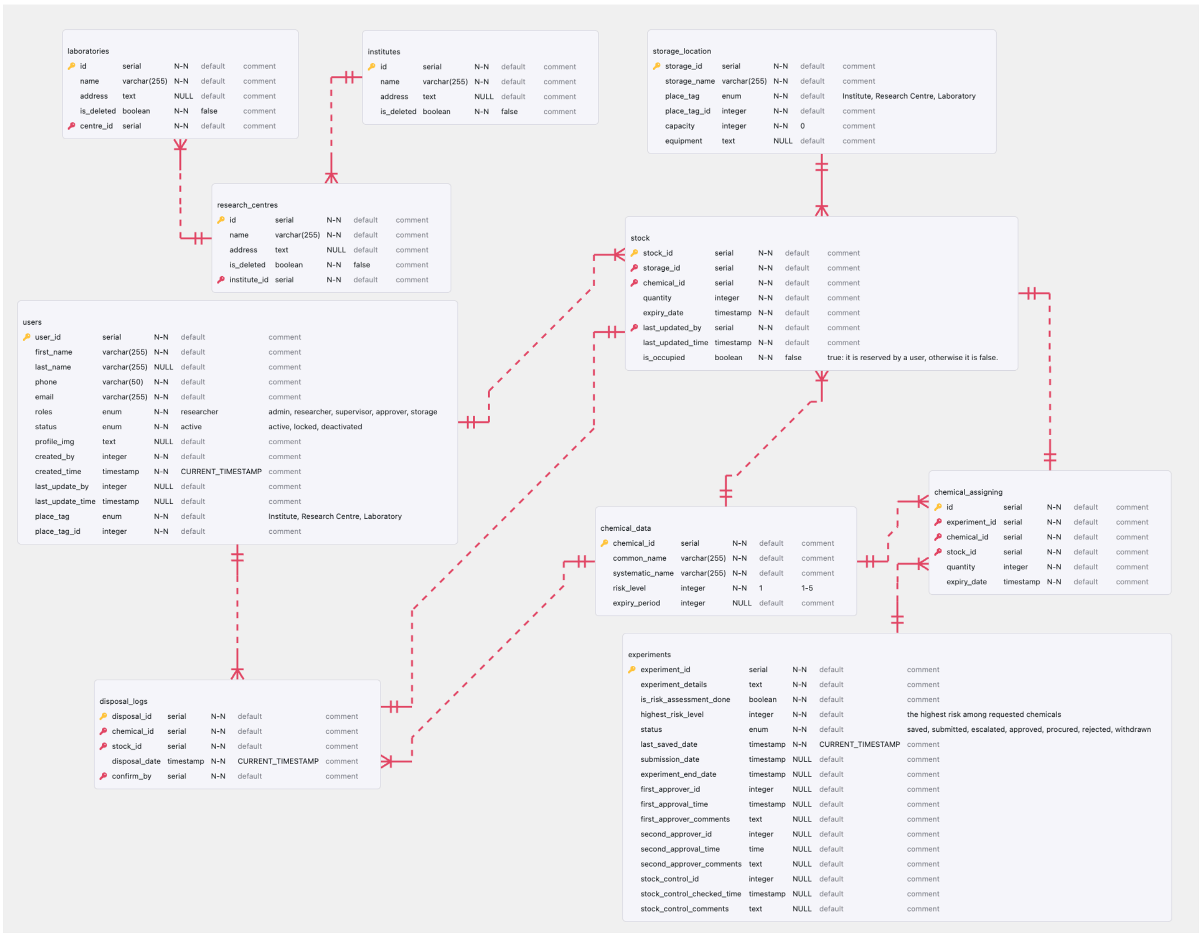
- experiment registration form (What exp, which chemical will be used?, risk assessment)
- Approval form
- Disposal Form
-
- level of risk category can be from 1 to 10 (e.g. from lv 5 and above may need approval from a supervisor)
- Stock class
- Database for the system
- Incident report including hazard identification
- Danger Identification system
- Authentication
- MFA can be mentioned when deploying the project (no need to develop)

Object
- User class
- Storage class
- Chemical class (including matching chemical system, Exp date)

risks log -> what might go wrong (e.g. lose a member of the team -> what to do about it?)
issues log -> things that do go wrong.
risks log can be 10, but issue log might be 0 or some.

Sprint planning - how long for each sprint?
sprint zero -> at least 3 weeks or more

SQL Table



laboratories

🔑	id	serial	N-N	default	comment
	name	varchar(255)	N-N	default	comment
	address	text	NULL	default	comment
	is_deleted	boolean	N-N	false	comment
🔑	centre_id	serial	N-N	default	comment

institutes

🔑	id	serial	N-N	default	comment
	name	varchar(255)	N-N	default	comment
	address	text	NULL	default	comment
	is_deleted	boolean	N-N	false	comment

storage_location

🔑	storage_id	serial	N-N	default	comment
	storage_name	varchar(255)	N-N	default	comment
	place_tag	enum	N-N	default	Institute, Research Centre, Laboratory
	place_tag_id	integer	N-N	default	comment
	capacity	integer	N-N	0	comment
	equipment	text	NULL	default	comment

research_centres

🔑	id	serial	N-N	default	comment
	name	varchar(255)	N-N	default	comment
	address	text	NULL	default	comment
	is_deleted	boolean	N-N	false	comment
🔑	institute_id	serial	N-N	default	comment

stock

🔑	stock_id	serial	N-N	default	comment
🔑	storage_id	serial	N-N	default	comment
🔑	chemical_id	serial	N-N	default	comment
	quantity	integer	N-N	default	comment
	expiry_date	timestamp	N-N	default	comment
🔑	last_updated_by	serial	N-N	default	comment
	last_updated_time	timestamp	N-N	default	comment
	is_occupied	boolean	N-N	false	true: it is reserved by a user, otherwise it is false.

users

🔑	user_id	serial	N-N	default	comment
	first_name	varchar(255)	N-N	default	comment
	last_name	varchar(255)	NULL	default	comment
	phone	varchar(50)	N-N	default	comment
	email	varchar(255)	N-N	default	comment
	roles	enum	N-N	researcher	admin, researcher, supervisor, approver, storage
	status	enum	N-N	active	active, locked, deactivated
	profile_img	text	NULL	default	comment
	created_by	integer	N-N	default	comment
	created_time	timestamp	N-N	CURRENT_TIMESTAMP	comment
	last_update_by	integer	NULL	default	comment
	last_update_time	timestamp	NULL	default	comment
	place_tag	enum	N-N	default	Institute, Research Centre, Laboratory
	place_tag_id	integer	N-N	default	comment

chemical_assigning

🔑	id	serial	N-N	default	comment
🔑	experiment_id	serial	N-N	default	comment
🔑	chemical_id	serial	N-N	default	comment
🔑	stock_id	serial	N-N	default	comment
	quantity	integer	N-N	default	comment
	expiry_date	timestamp	N-N	default	comment

chemical_data

🔑	chemical_id	serial	N-N	default	comment
	common_name	varchar(255)	N-N	default	comment
	systematic_name	varchar(255)	N-N	default	comment
	risk_level	integer	N-N	1	1-5
	expiry_period	integer	NULL	default	comment

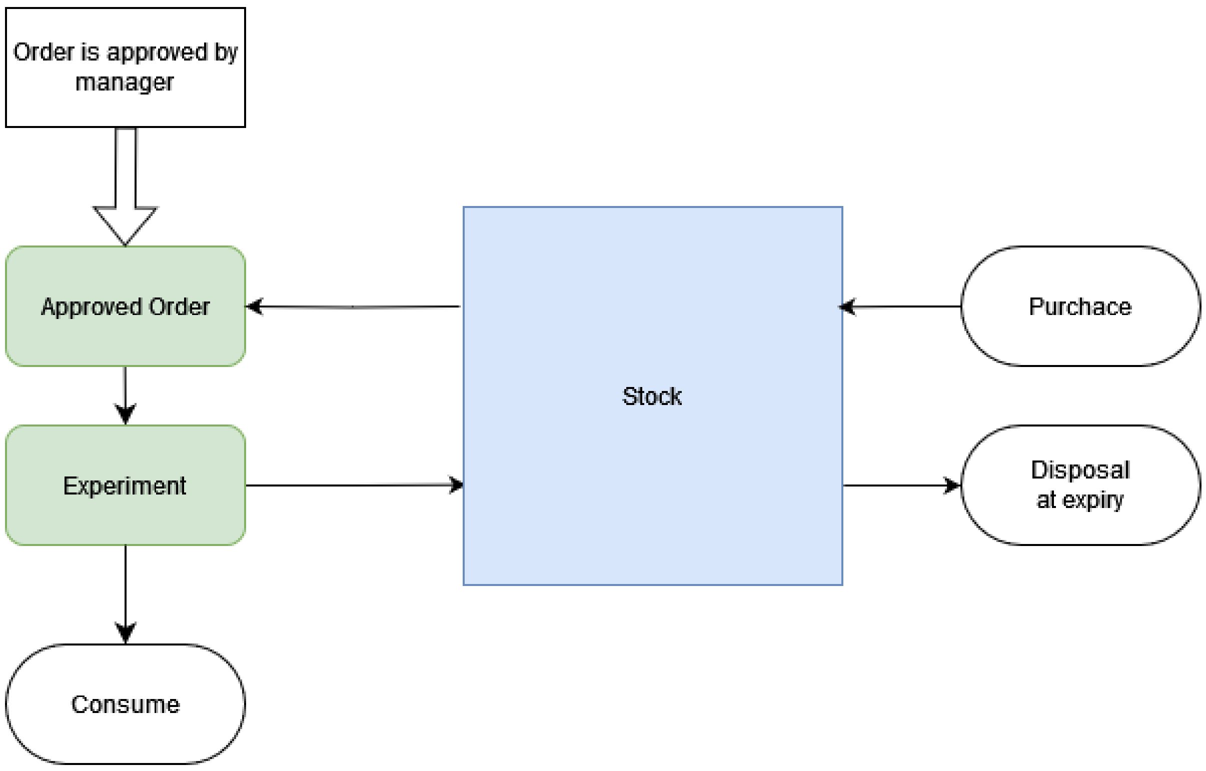
experiments

🔑	experiment_id	serial	N-N	default	comment
	experiment_details	text	N-N	default	comment
	is_risk_assessment_done	boolean	N-N	default	comment
	highest_risk_level	integer	N-N	default	the highest risk among requested chemicals
	status	enum	N-N	default	saved, submitted, escalated, approved, procured, rejected, withdrawn
	last_saved_date	timestamp	N-N	CURRENT_TIMESTAMP	comment
	submission_date	timestamp	NULL	default	comment
	experiment_end_date	timestamp	NULL	default	comment
	first_approver_id	integer	NULL	default	comment
	first_approval_time	timestamp	NULL	default	comment
	first_approver_comments	text	NULL	default	comment
	second_approver_id	integer	NULL	default	comment
	second_approval_time	time	NULL	default	comment
	second_approver_comments	text	NULL	default	comment
	stock_control_id	integer	NULL	default	comment
	stock_control_checked_time	timestamp	NULL	default	comment
	stock_control_comments	text	NULL	default	comment

disposal_logs

🔑	disposal_id	serial	N-N	default	comment
🔑	chemical_id	serial	N-N	default	comment
🔑	stock_id	serial	N-N	default	comment
	disposal_date	timestamp	N-N	CURRENT_TIMESTAMP	comment
🔑	confirm_by	serial	N-N	default	comment

Chemical Life Cycle

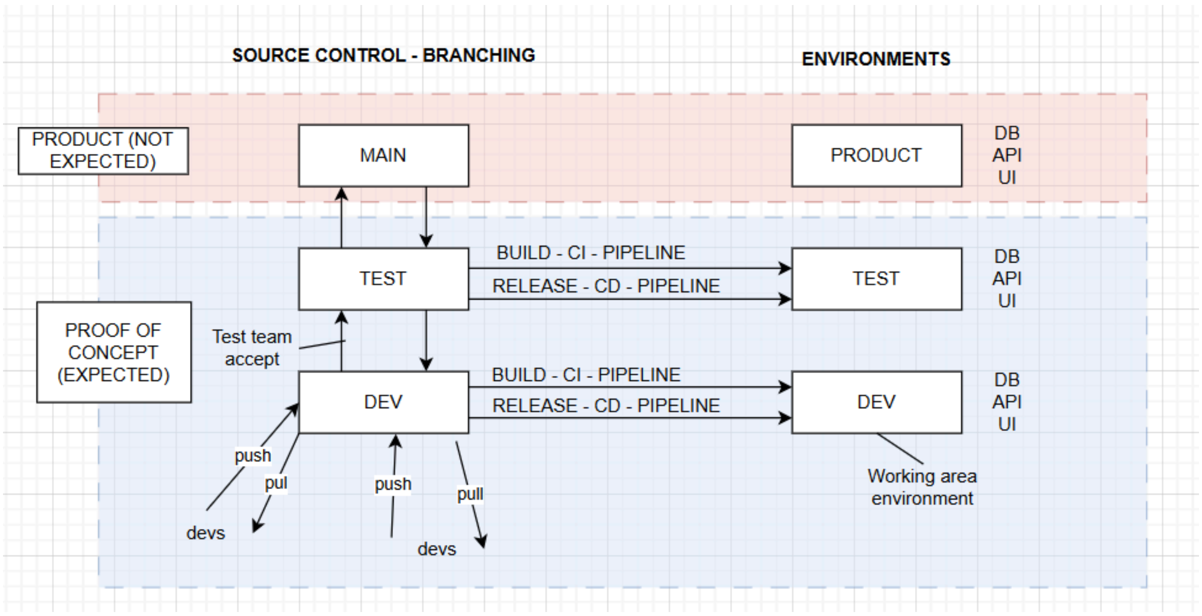


🟦	Real world: Kept in the Location where the order and the experiment were done last time. Database: Stock table
🟩	Real world: Moved to the experiment location. Database: Orders Table

From Workshop

Week workshop 2
** in the final report
- this is what we did, this is what we've got to, this is what we deliver,
- + and this is what we would do or looking forward in the future

** BA design -> developer and tester work against the user stories
** make decisions and justify why we use these tools and why not using these tools, what are the reasons.
** Risk assessment form is expected -> check from the link
** what do you need in order to test the product (may be 10 chemicals) (test data)
** researchers and supervisors not responsible for ordering, storing and disposal the chemical, the stock control person will do it, and take care all of the safety, risk assessment
** supervisor has to approve for ordering and using



Week 3 Workshop:
Focus on Proof of concept: make it simple as possible but satisfy all the requirements.
All the tasks that need to do more work or more complex, put it in the production plan considerations or something like that.

Week6 Workshop: About Presentation 1

be honest -> this is what happened, this is where we are at, this is what we will do

BA
Project brief (refer to initiation doc)
What the business needs are.
Business need - proof of concept
Explain Scope - what in the scope, what out of scope
Introduce Requirements and Assumptions

Architect
Design architecture = solution of the proof of concept. frameworks,
    architect: will do this, what tech did we use and why.

Timeline & work management = **PM + Scrum master** => roadmap, sprint 0
    -> details level in scrum master How we dealt with the schedule.
    Where we are , at the end of Sprint 0
    According to the road map, what are we going to achieve?

Tester
QA Approach - from the roadmap, How to assure quality of product.
Introduce test strategies, what happened with the testing.

PM
Risks issue -> log for each, expected Major issues to be presented and talked about
    . Don't need to show all logs(20 is too much)

Demonstration = if finish sprint 1, expect demonstration (very quick) [Developed, tested, accepted]
    only show what is working, Don't show half-done parts, the audience is the use of the app.

Devops & Exe = **Scrum Mas** + **Devops Engineer** => details on pipeline, no need to show code
    No code needed, What the coding practices, how we peer reviewing, maintaining the code quality

layout = **Dev team + Architect** = what stack used in each environment. how we make the decision to get to those. He does not care what we use, he cares why we use it. e.g. the team familiar with the tools.

presentation technique = do not read while presenting. make eye contact. timeline=> 30min for each group incl questions [20 mins to present **loss mark if over 20 mins** + 5 min questions and 5 min for swap over]

****lesson learnt = expected to see all the changes (Road map changes)
    (last presentation should have a lot of detail.)
    looking back and seeing how we improve things.
    burndown chart
    identify what went wrong
    present from the role, not individual ppl
    present the problem that we have

* no need to introduce the whole team, introduce as a team
    e.g. this is from ba team… , this is from dev team ..

FINAL REPORT

a slide should be a summary of the what person will be talking
consistency (make it like one person do it all)
smooth transition between members
Styling, font, branding
Business presentation like (Audience is Flinders Uni)
Use the Flinders template
Documentation standard (use client's template, standard) make it like it is belong to the client

EXPECTATION FOR THE PRESENTATION

- 4-5 ppl doing the presentation
- everyone will help create the slides
- questions will be asked to ppl that is not presenting
- do not do dead by powerpoint [too much information on the slide, image is clear and seeable]
-
- timeline on each sprint and what we do in each sprint
- if we have a lot of detail, put it in the end, do not present them
  appendix: risks and issue logs, coding standard for dev team
-
- PPT: bullet point, not paragraph 10~15 slides + appendix slides, animation(IF necessary), Flinders Visual Style
-
-
-

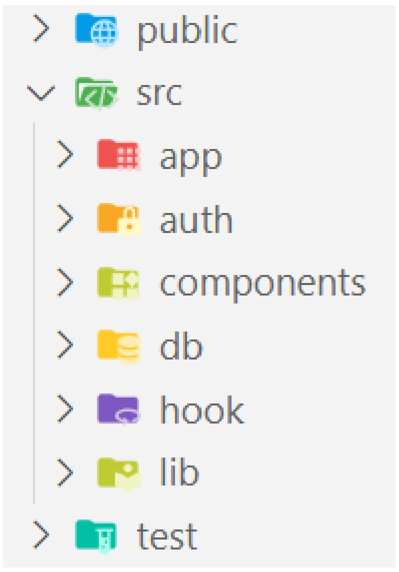
Testing strategy skelton

15th / Aug
Testing Strategy
<u>Database access(API) test</u>
All the **database access** should be done in the files in **"db"** folder.
Database should be accessed via **drizzle ORM**.
Testing files should be stored in **"test" folder**
Testing should be done by running **Javascript files** containing a test function.
One test function test one database access function in each file.
Test functions have proper **arguments and console** out the return value of the database access.

Testers will check the output in the **console manually**.
Tester will test again by adding **another test file** for additional test, whenever there is **a new pull request** with commits to the “db” folder.

Black box testing
Manually by UI
User stories check
Edge case inputs (in later sprints)

Coding Rules



Local components, functions, and utilities
in each folder
Local types
declared in the file it is used

Universal stuff is in below:
Public: All pictures, Risk assessment PDF
Components: Universal reusable parts
db: all the database query (use Drizzle, cache)
hook: universal hooks
lib: utility
types.d.js at src root: universal types
data?: constant data

Naming Convention
Variable, function = Lower camel case

folder(URL) = Snake case new_document
Component, Type = Capital camel case
Constant = Screaming case, (should be outside the function.)
hook = useContext, useOptimistic, useLocalStorage
Singular, plural d.g. user(one user), users(all users), admins(all admin)

We use

Function component (not the class component)

Both OK:

function(){},
const x = ()=>{}

Form:

Both OK:

1, Form, input submit, server action, “use server”
2, useState(), button, onClick(handleSubmit), “use client”

Lean Next js

https://www.youtube.com/watch?v=vwSlYG7hFk0&t=2105s

Learn SOLID principle

especially “universal Components” person.

CSS:

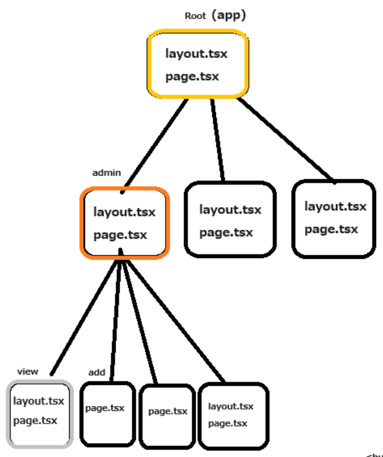
If possible
  Prettier default + "plugins": ["prettier-plugin-tailwindcss"]
With tailwind use:
  cn()
Responsive?
  In that case, “Mobile first” do we use “md: or lg:” ?
Library (shadcn??)
  We might need one library to show charts and others. Let’s discuss.

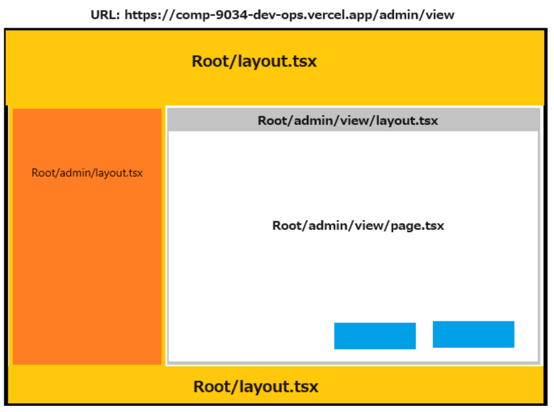
Git

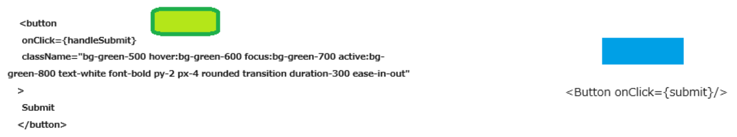
Only merge to Dev branch.
Only commit what you pulled from Dev branch and what you created
One feature one commit.
One task one pull request.
Use bin folder

GitCommit message

Add:	Added a file
Update:	Added a new functionality, or improved
Fix:	Bug fix, refactor
Delete:	Deleted a file or functionality




URL: https://comp-9034-dev-ops.vercel.app/admin/view
Root/layout.tsx
Root/admin/view/layout.tsx
Root/admin/layout.tsx
Root/admin/view/page.tsx
Root/layout.tsx



```
 <button
  onClick={handleSubmit}
  className="bg-green-500 hover:bg-green-600 focus:bg-green-700 active:bg-
green-800 text-white font-bold py-2 px-4 rounded transition duration-300 ease-in-out"
 >
  Submit
 </button>
```

```
<Button onClick={submit}/>
```

API (Database Query Functions via Drizzle)

Returning Format:

{

success: true | false, (true: successful query, false: there is an error)

data: array | JSON | null, (array: when you request more than 1 row, JSON: when you request ONE row e.g getUserByEmail, null: there is an error)

message: string | empty (string: “error message”, empty: successful query)

}

Epic: 2

2.1 addUser() : used to add new user role. [Privilege: Admin]

2.2 updateUserDetails() : used to update user details (first name, last name, email, phone, role, place tag, place tag id). [Privilege: Admin]

~~getUser() 2.3 Done auth() role and user info is fetched in logging in.~~

2.3 getUsers() : used to get the list of all users in descending creation date. [Privilege: Admin]

2.4 updateUserStatus() : used to change user status only e.g change from active -> locked or locked -> active or active -> deactivated [Privilege: Admin]

2.5 updateUserProfile() : used to update user profile details (picture, first name, lastname, phone, email?). [Privilege: All users]

2.6 getUserByEmail() : used to get user profile details. [Privilege: All users]

~~getUsersBy?? or up to Dev?~~

~~Users: What should admin and staff can edit?~~

Epic:5 [Privilege: Admin]

5.1 addInstitute() : used to add new institute.

5.2 updateInstitute() : used to update an existing institute details (name or address).

5.3 removeInstitute() : used to hide an existing institute so that it won't be available for selection in the system.

5.4 getInstitutes() : used to get the list of all institutes in descending institute ID.

5.5 getInstituteById() : used to get institute by id.

5.6 addResearchCentre() : used to add a new research centre.

5.7 updateResearchCentre() : used to update an existing research centre details (name, address, its allocated institute).

5.8 removeResearchCentre() : used to hide an existing research centre so that it won't be available for selection in the system.

5.9 getResearchCentres() : used to get the list of all active research centres in descending centre id.

5.10 getResearchCentreById() : used to get centre by id.

5.11 addLab() : used to add new laboratory.

5.12 updateLab() : used to update an existing laboratory details (name, address, allocated research centre).

5.13 removeLab() : used to hide an existing laboratory so that it won't be available for selection in the system.

5.14 getLabs() : used to get the list of all active laboratories in descending lab ID.

5.15 getLabById() : used to get lab by id.

5.16 addStorage() : used to add a new storage location.

5.17 updateStorage() : used to update an existing storage location details (name, capacity, equipment).

5.18 removeStorage() : used to hide an existing storage location so that it won't be it won't be available for selection in the system.

5.19 getStorages() : used to get the list of all active storage locations in descending id.

5.20 getStorageById() : used to get storage by id.

5.21 addChemicalData() : used to add new chemical data.

5.22 updateChemicalData() : used to update an existing chemical data (common name, systematic name, risk level, expiry date).

5.23 removeChemicalData() : used to hide an existing chemical data so that it won't be available for selection in the system.

5.24 getChemicalData() : used to get the list of all active chemical data.

5.25 getChemicalDataById() : used to get chemical data by id.

~~addInstitution, addCentre, addLob?~~
~~createPlace()~~
~~updatePlace() include remove~~

~~getInstitutions()~~
~~getCentres()~~
~~getLabs()~~

~~createStorage()~~
~~updateStorage()~~
~~getStorage()~~

~~createChemicalData()~~
~~updateChemicalData()~~

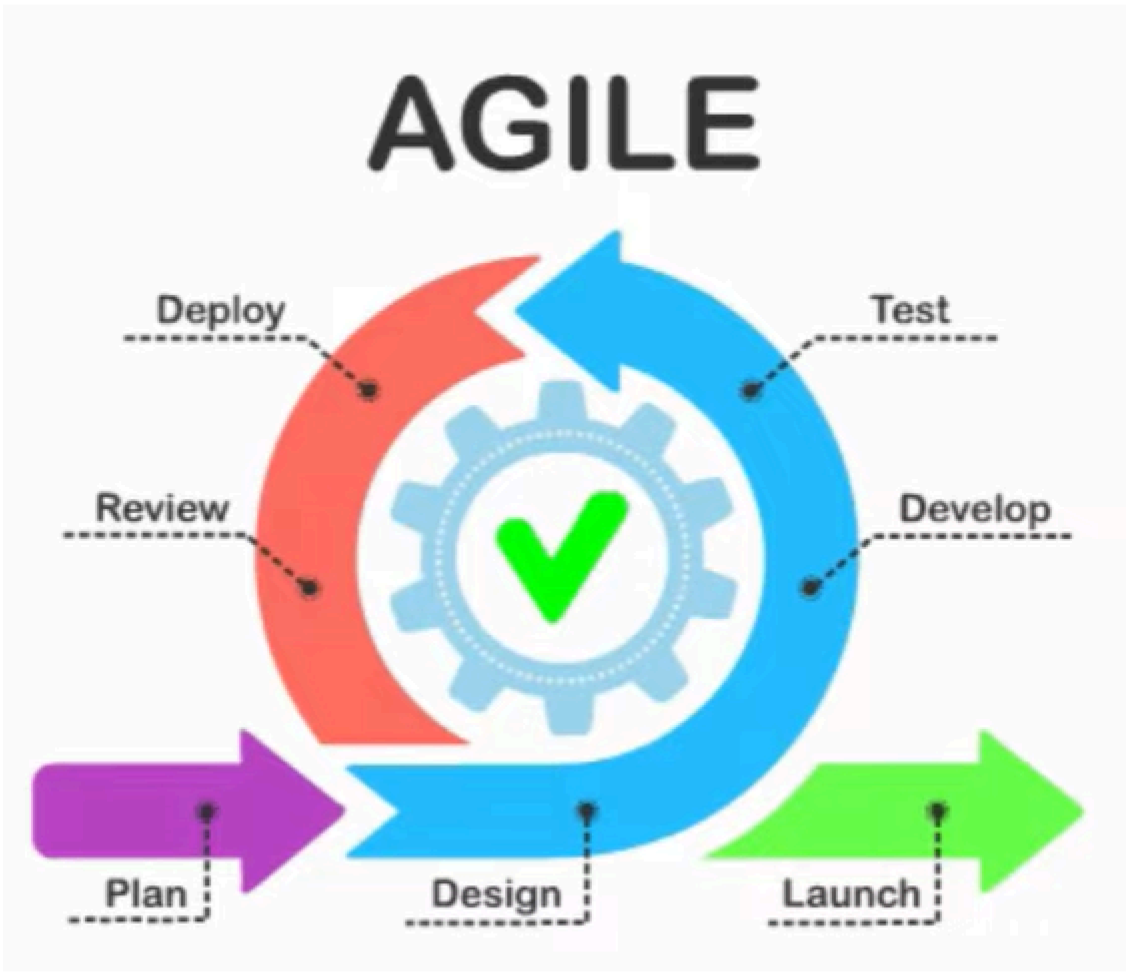
Future:
updateChemicalStock()

vDev Roles:

	Design	
1.	UI design (Design, requirements)	[Alyssa, Min]
	Front end	
2.	Global Layout (design + SEO)	[Min]
3.	Local layout (design + CSS)	[Min]
4.	Universal Components (React) (Optional)	
	Table(page, search)	[Luck]
	modal, button, title(back), input, dropdown	[Tang]
5.	Pages, Feature (Logic and requirements) (Higher)	[Jiaqi, (Luck)]
	Back end	
6.	API (database, JavaScript, Nextjs) (Lower)	[Kong]
	Testing	
7.	Test API (database, testing)	[Jiaqi]
8.	Test UI (requirement, User eXperience)	[Max, Taka]
	Management	
9.	Merging pull request	[Taka]
10.	Task Allocation	[Taka]

Design (Epic2, 5 finished by 25.Aug)

Agile Development

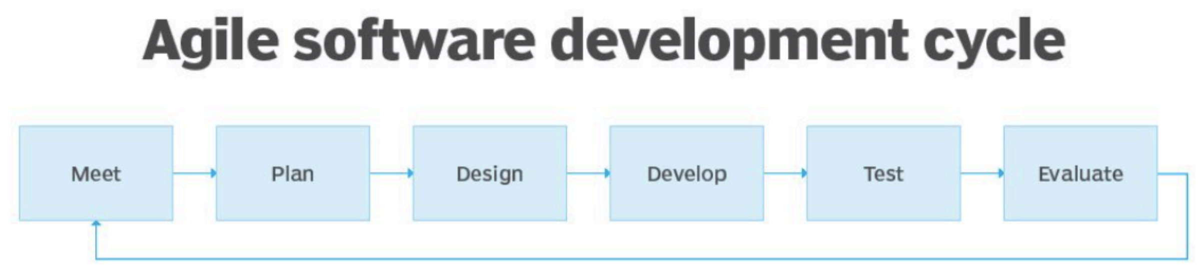


Individual interactions are more important than processes and tools.

A focus on working software rather than thorough documentation. (But!! We still need to record and write down the process as we are producing the final report)

Collaboration instead of contract negotiations.

A focus on responding to change.



Presentation: (20 min is given)

- Scope should be put at first section - PM
- Presentation orders are not required to be the same as the rubrics.
- Demonstration should be quick and only the finished work of Sprint 1 can be demonstrated. otherwise, just let the tutor know what and where you are working.
- No code is expected to be presented.
- Layout&Technology-Dev&Architecture showing what technologies and tools you use, and why you decided to choose these.
- The tutor doesn't expect to see the presenter reading notes or paper without eye contact.
- You will lose marks if you go over 20 minutes.
- The tutor expects 4-5 people to present the whole content. Others can help to answer when there are any questions from the audience.
- Make your roadmap clear and visual presentation so that the audience can clearly see it.
- If you need a lot of information as a reference to help you answer your questions, put them at the end of your slides but don't present them within 20 minutes.
- You can use animations in your slides if they can help you explain your content better and better deliver to the audience.
- **You must NOT USE individual words like i, me. You should use We, our instead.**