



Institute of Technology of Cambodia



Department of Information and Communication Engineering

Single Sign-On (SSO) System and DevOps

Student : Mr. ROTHA DAPRAVITH

Academic Supervisor : Mr. NOP PHEARUM

Company Advisor : Mr. VON SEYHA

Duration : 19 February - 31 May 2024

Academic Year 2023-2024

TABLE OF CONTENTS



I

GENERAL PRESENTATION

II

ANALYSIS AND DESIGN

III

IMPLEMENTATION

IV

CONCLUSION

V

DEMONSTRATION

GENERAL
PRESENTATION

ANALYSIS AND DESIGN

IMPLEMENTATION

CONCLUSION

DEMONSTRATION

I. GENERAL PRESENTATION

1.1. General Presentation of Organization

- The internship took place at the **Ministry of Economy and Finance**
- The internship started from 19 February - 31 May 2024



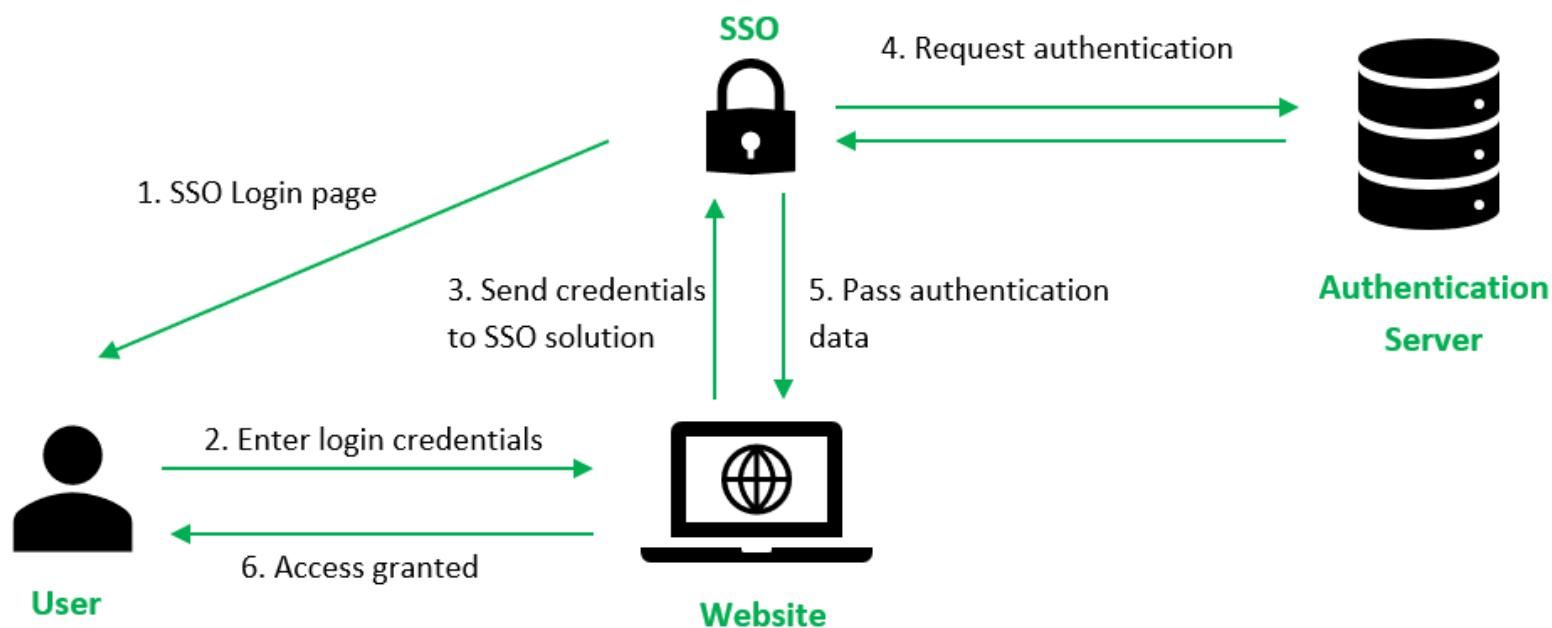
Address	: St. 92, Sangkat Wat Phnom, Khan Daun Penh, Phnom Penh
Tel	: (+855) 97 252 8358
Email	: legalconcilmef@gmail.com
Website	: https://www.legalconcilmef.com/en

The **Legal Council of the Ministry of Economy and Finance** was established in 2013 by Prâkas No. 290 SHV.BRK. It has been revised and modified by Sub-decrees No. 75 ANKR.BK, No. 488 ANKR.BK, and No. 76 ANKR.BK. Its purpose is to ensure legality, consistency, and harmonization in the drafting of legal texts, manage implementation effectively, and participate in the execution of the Public Financial Management Reform Program (PFMRP).

1.2. Project Overview

What is Single Sign-On (SSO) System?

The **Single Sign-On (SSO) System** is an authentication method that allows users to access multiple applications with a single login. By logging in once, users can access all connected systems without needing to re-enter their credentials for each one. This simplifies the user experience, enhances security, and reduces password fatigue.



1.3. Problems Overview

- Frequent logins.
- Password fatigue.
- Scattered user data.
- Data analysis difficulty.
- IT management complexity.



1.4. Objective Overview

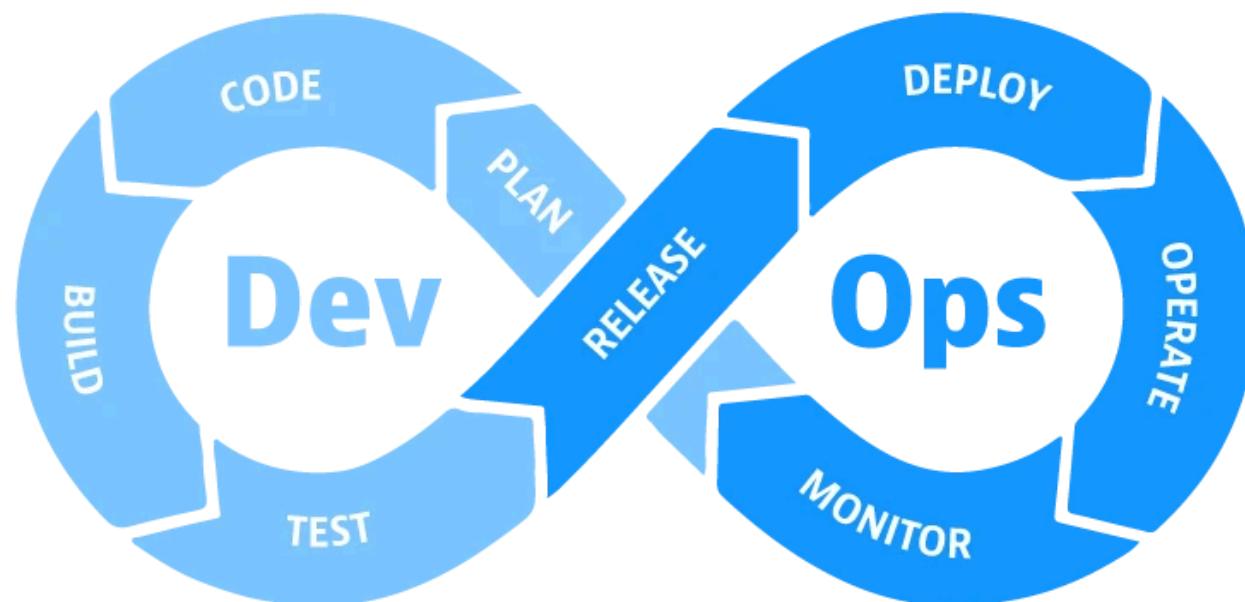
- Centralized user management.
- Single credentials.
- Enhanced security.
- Unified access.
- Improved user experience.



1.5. DevOps Project Overview

What is DevOps ?

DevOps is a methodology that combines software development and IT operations to streamline workflows and enhance collaboration. It aims to shorten development cycles, increase deployment frequency, and deliver reliable software quickly and consistently, closely aligned with business goals. This approach emphasizes automation, continuous measurement, and rapid release cycles.



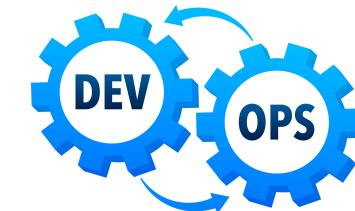
1.6. Problems Overview

- Slow development.
- Siloed between development and operation team.
- Deployment failures.
- Unreliable environment.



1.7. Objective Overview

- Faster development cycle.
- Enhanced production quality.
- Improved collaboration.
- Reliable deployment.



II. ANALYSIS AND DESIGN

2.1. Project planning table

Tasks	Weeks											
	1	2	3	4	5	6	7	8	9	10	11	12
Learn new technologies												
Analysis project requirement & Database design												
Implementation SSO keycloak												
Learn and Implement DevOps												
Maintainance and Testing												
Deployment												
Report												

2.2. Functionality for each roles



Administrator

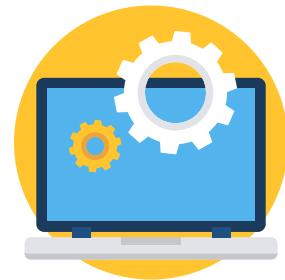
- Manage logins, sessions, and system access.
- Customize user interfaces and security settings.
- Manage realms configurations and role assignments.
- Handle user account registration and credentials.
- Configure authentication flows and manage tokens.
- Administer client and identity provider settings.



Users

- Enable user self-registration and account management.
- Handle logins, logouts, and email verifications.
- Update personal information and passwords.
- Monitor device activity and secure logins with OTP.
- Configure identity and service provider settings.
- Track user activity and generate system reports.

2.3. Functionality requirement for DevOps



Functionality requirement

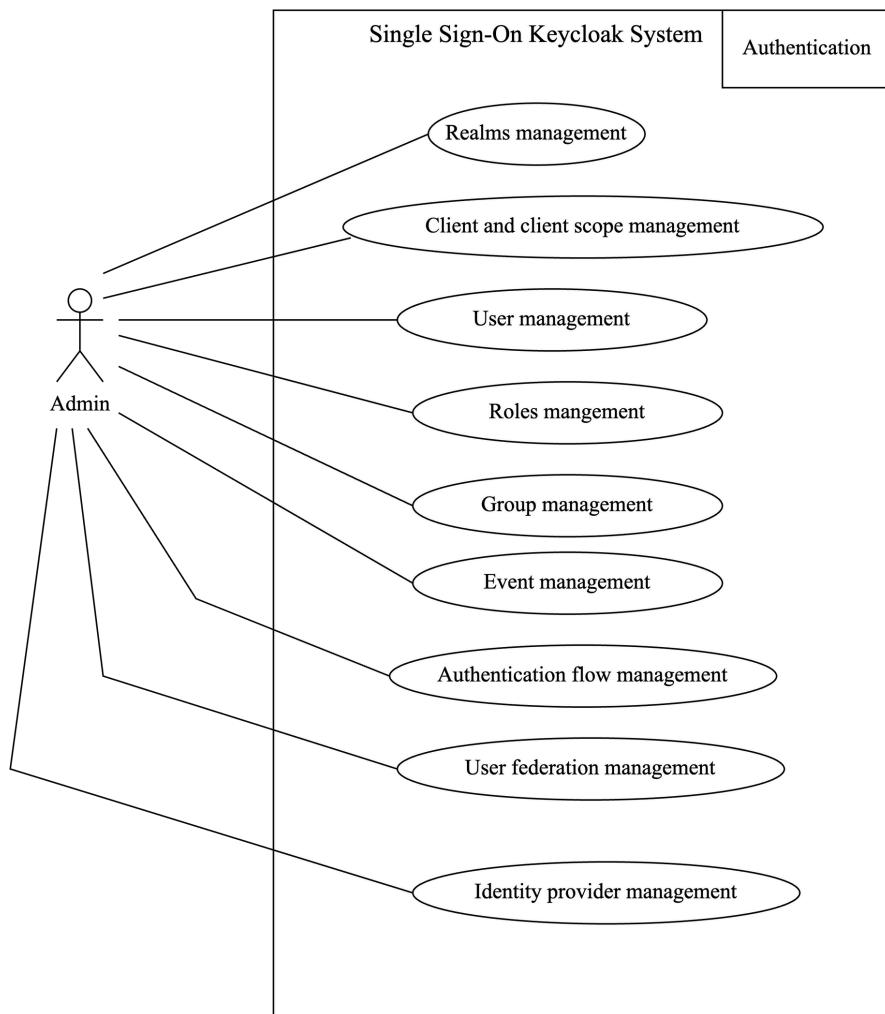
- Version Control.
- Containerization.
- Web-server configuration.
- Continuous Integration (CI).
- Continuous Deployment (CD).
- Infrastructure as Code (IaC).
- Monitoring services.



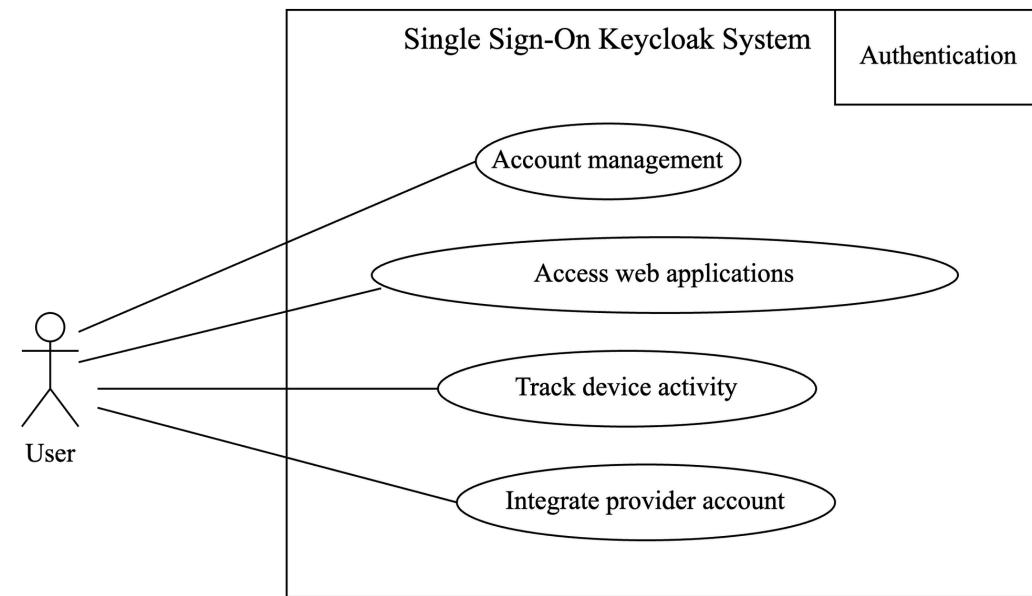
Non-functionality requirement

- Performance.
- Scalability.
- Reliability.
- Security network.
- Telegram for Alerting messages.

2.4. Use Case Diagram

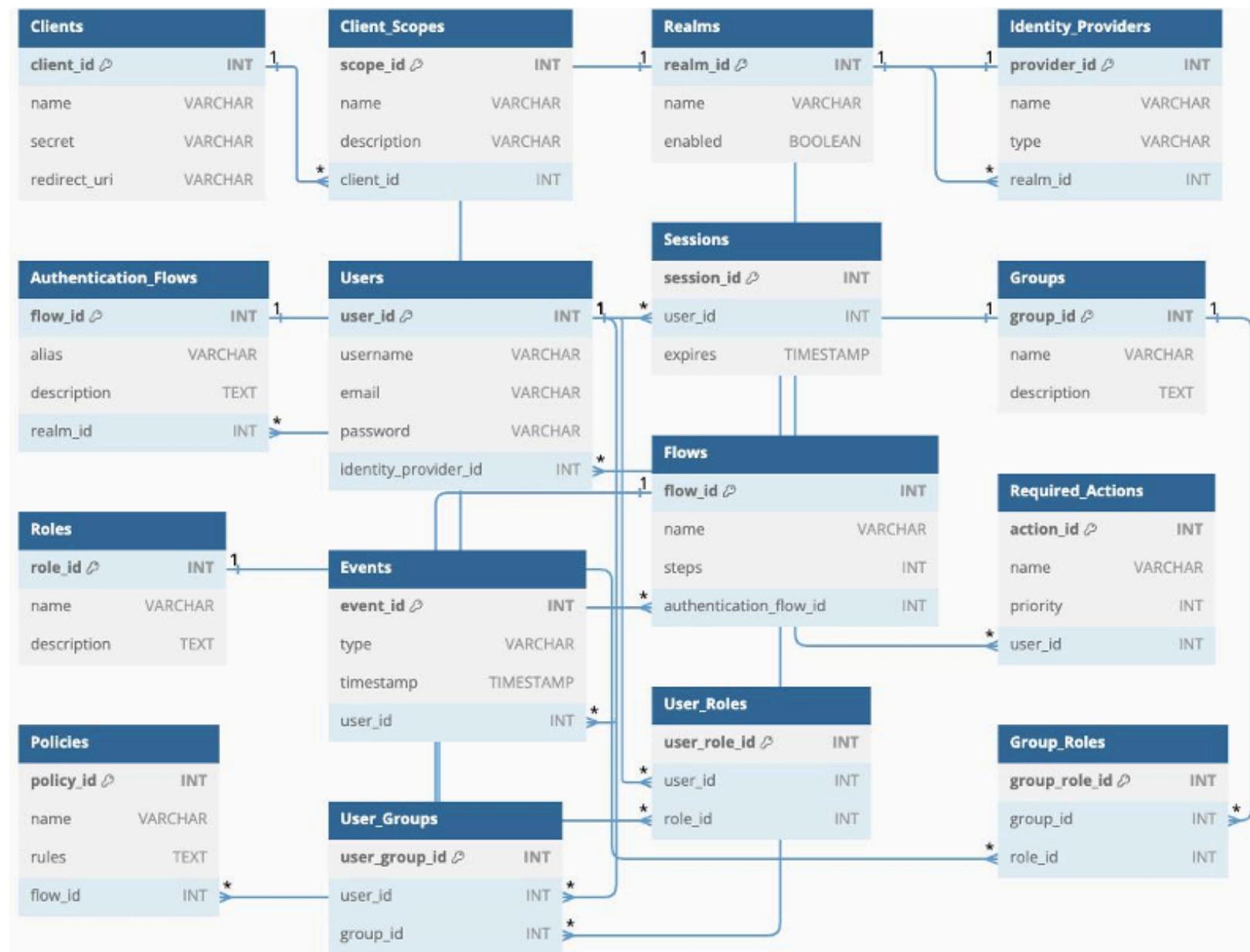


Admin role use case diagram

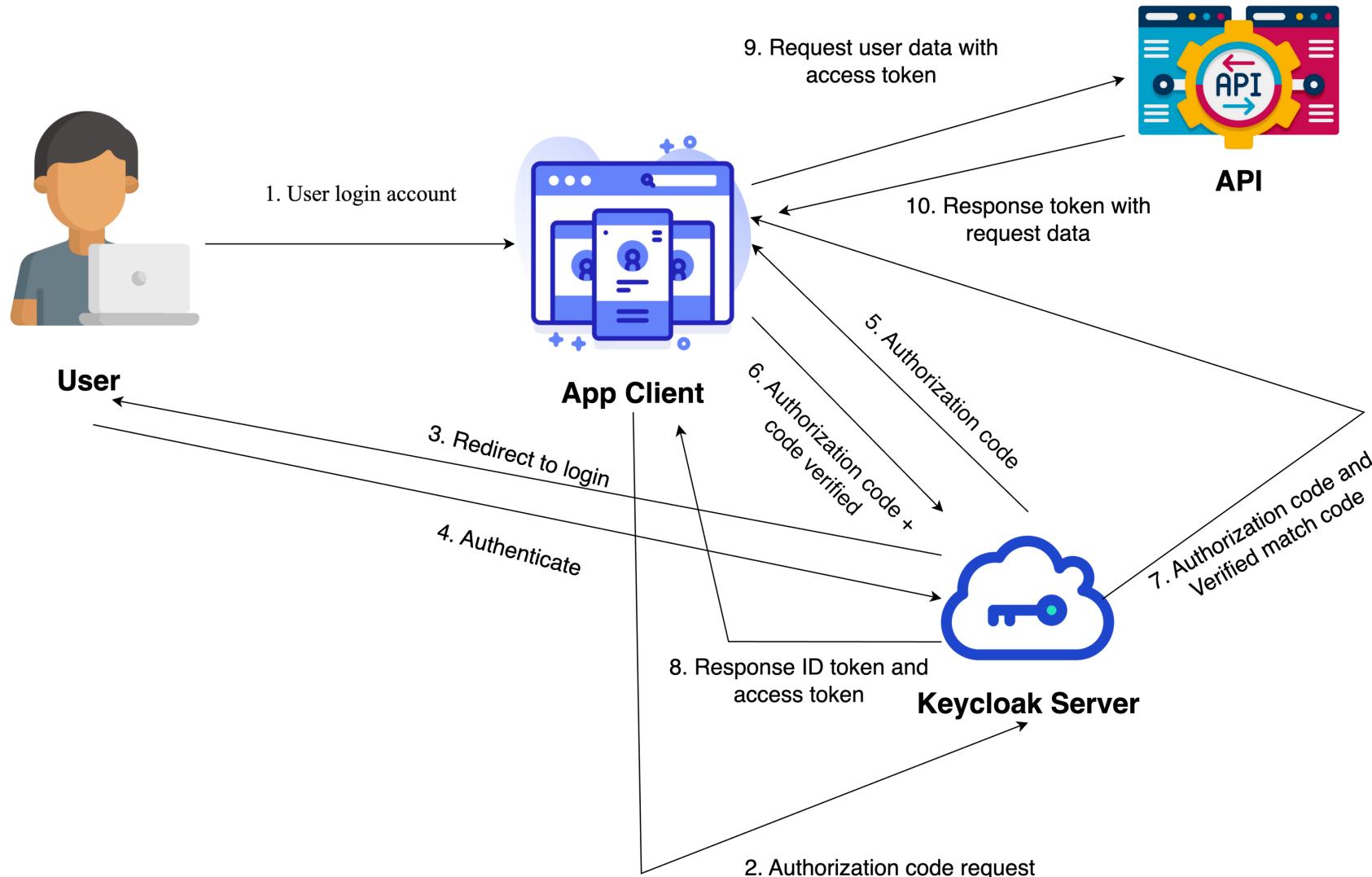


User role use case diagram

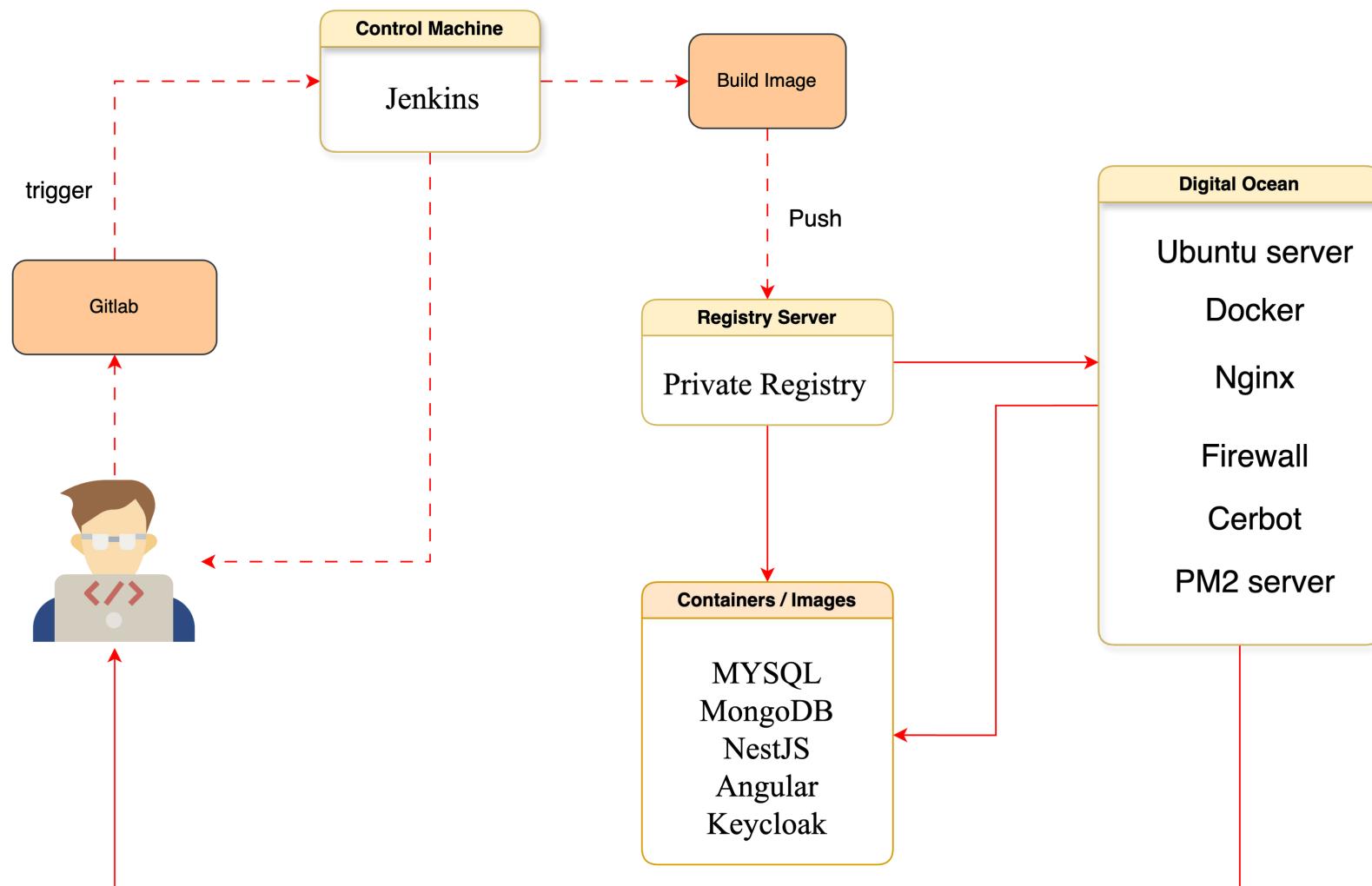
2.5. Database Schema



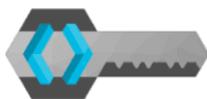
2.6. Architecture diagram of SSO system



2.7. Architecture diagram of DevOps



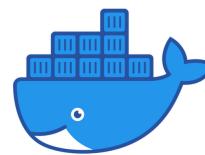
2.8. Choice of Technologies and Tools



Keycloak



Angular



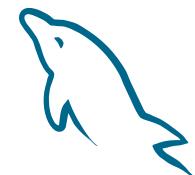
Docker



Nginx



Jenkins



MySQL



MongoDB



Visual studio code



SSH



Termius



SourceTree



GitLab



Pm2 Server



Digital Ocean



Postman



Name.com



Draw.io



Telegram

GENERAL
PRESENTATION

ANALYSIS AND DESIGN

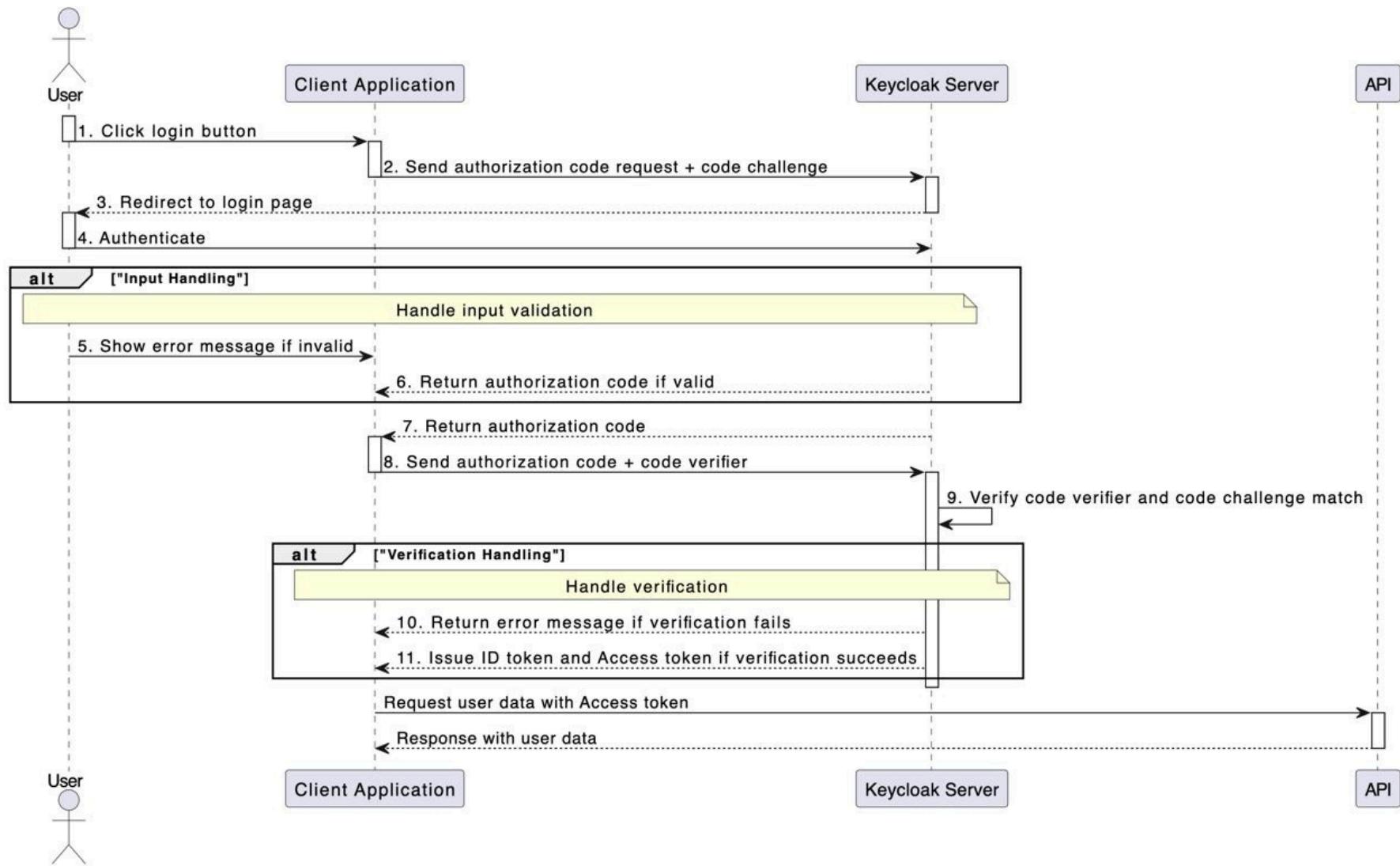
IMPLEMENTATION

CONCLUSION

DEMOSRATION

III. IMPLEMENTATION

3.1. Sequential Diagram for Authentication



3.2. DevOps Project Implementation

- Build Docker file

Node.js environment version

Dockerfile > ...
ROTHA DAPRAVITH, 2 months ago | 1 author (ROTHA DAPRAVITH)

```
1 FROM node:18-alpine
2
3 RUN mkdir -p /home/node/app/node_modules && chown -R node:node /home/node/app
4
5 WORKDIR /home/node/app
6
7 COPY package*.json .
8
9 # Copy your Swagger file to the appropriate location inside the container
10 COPY ./swagger.yml /usr/share/nginx/html/api-docs/swagger.yml
11
12 USER node
13
14 RUN yarn install
15
16 COPY --chown=node:node . .
17
18 EXPOSE 4400
19 CMD [ "yarn", "start:dev" ]
```

install a nest.js dependencies, and set the container port to serve port 4400 to running server and run the nest application using command 'yarn start:dev' or 'nest start --watch'

3.2. DevOps Project Implementation

- Connect GitLab to Jenkins credentials by access token

Project Access Tokens

Generate project access tokens scoped to this project for your applications that need access to the GitLab API. You can also use project access tokens with Git to authenticate over HTTP(S). [Learn more](#).

Active project access tokens (2)							Add new token
Token name	Scopes		Created	Last Used ?	Expires	Role	Action
lc-sso-system	read_repository, write_repository		Apr 29, 2024	4 weeks ago	in 9 months	Maintainer	trash
gitlab-sso	read_repository, write_repository, read_registry, write_registry, ai_features		May 30, 2024	5 minutes ago	in 10 months	Maintainer	trash

- Jenkins's Credentials

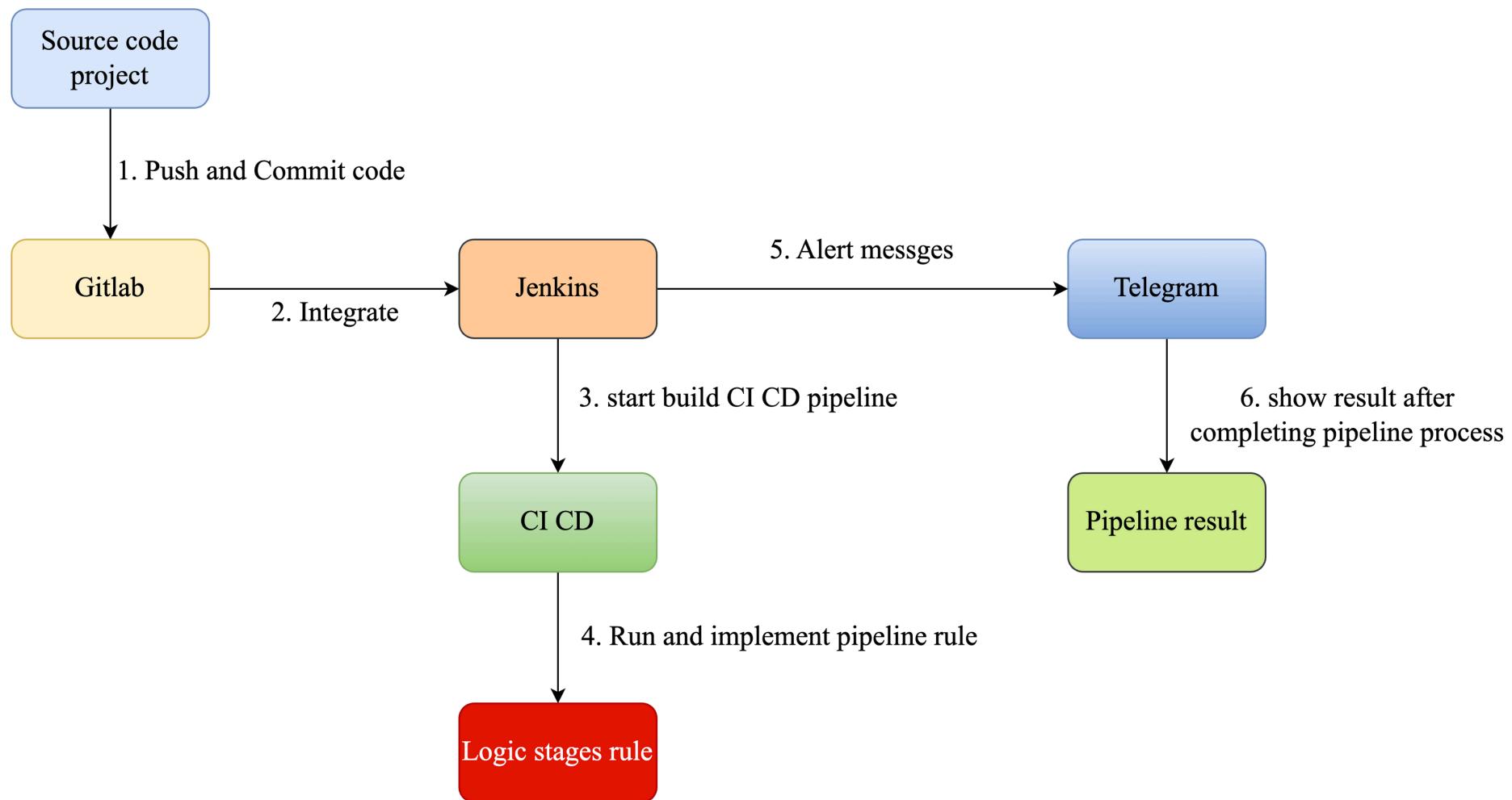
Dashboard > Manage Jenkins > Credentials

Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	003	Dapravith
		System	(global)	004	004
		System	(global)	001	Dapravith/*********
		System	(global)	0004	@Dapravith
		System	(global)	Telegram_BotToken	Telegram_BotToken

3.2. DevOps Project Implementation

- Process of Jenkins Pipeline CI CD



3.2. DevOps Project Implementation

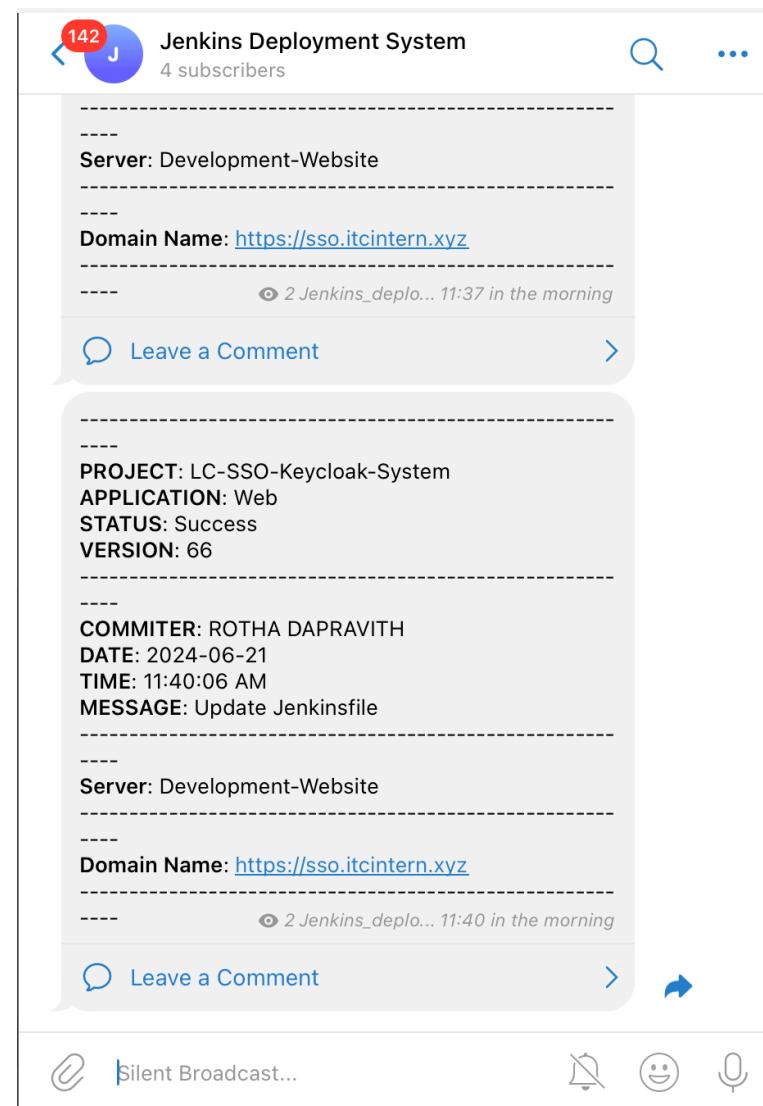
- Jenkins blue ocean stage view

The screenshot shows the Jenkins Blue Ocean interface for a pipeline named 'lc-sso-system < 67'. The top navigation bar includes links for Pipeline, Changes, Tests, Artifacts, and various icons for refresh, settings, and logout. Below the header, project details are displayed: Branch: -, Commit: -, Last build: 20s ago (a few seconds ago), and Started by user Ratha Dapravith. The pipeline stages are shown as a horizontal timeline: Start, Start, Clone Project, Build Docker Image, Deploy Docker Compose, Push Docker Image, End, and End. Each stage is represented by a green circle with a checkmark, indicating successful completion. The 'End' stage is highlighted with a blue border. At the bottom, a log table lists the final steps: 'End of stages, thank you Jenkins!' (Print Message, <1s), 'chmod +x ./1-success-deploy.sh' (Shell Script, <1s), './1-success-deploy.sh' (Shell Script, 2s), and 'Deployment succeeded.' (Print Message, <1s). A 'Restart End' button is also visible.

Stage	Action	Time
End - <1s	> End of stages, thank you Jenkins! — Print Message	<1s
	> chmod +x ./1-success-deploy.sh — Shell Script	<1s
	> ./1-success-deploy.sh — Shell Script	2s
	> Deployment succeeded. — Print Message	<1s

3.2. DevOps Project Implementation

- Telegram chat bot alert messages



GENERAL
PRESENTATION

ANALYSIS AND DESIGN

IMPLEMENTATION

CONCLUSION

DEMONSTRATION

IV. CONCLUSION

4.1. Completed and Uncompleted Tasks for SSO System

Tasks	Status	Responsible by
Install and Configure Keycloak Server	Completed	ROTHA Dapravith
User Authentication	Completed	ROTHA Dapravith
Role-Based Access Control (RBAC)	Completed	ROTHA Dapravith
User Registration	Completed	ROTHA Dapravith
User Profile Management	Completed	ROTHA Dapravith
Administrator Authentication	Completed	ROTHA Dapravith
Session Management	Completed	ROTHA Dapravith
Deployment	Completed	ROTHA Dapravith
Integrate SSO system with another Web Client	In progress	ROTHA Dapravith

4.2. Completed and Uncompleted Tasks for DevOps

Tasks	Status	Responsible by
Manage Version Control	Completed	ROTHA Dapravith
Integration CI/CD with Gitlab and Jenkins	Completed	ROTHA Dapravith
Implementation with Docker	Completed	ROTHA Dapravith
Config automation deployment	Completed	ROTHA Dapravith
Configure DNS with Nginx	Completed	ROTHA Dapravith
Implementation with PM2 Server	Completed	ROTHA Dapravith
Notified Message to Telegram	Completed	ROTHA Dapravith

4.3. Difficulties

- Challenges with new technologies and DevOps.
- Have many requirements tasks in project.
- Time management.

4.4. Experiences

- Real working environments.
- Learn new technologies and DevOps.
- Know how to deployment on production server.
- Self-learning, teamworks and project management.

4.5. Perspectives

- Completed with remain tasks.
- Secure Microservices with single sign-on system.
- Improve performance and security.



V. DEMONOSTATION



THANK YOU FOR YOUR ATTENTION !