

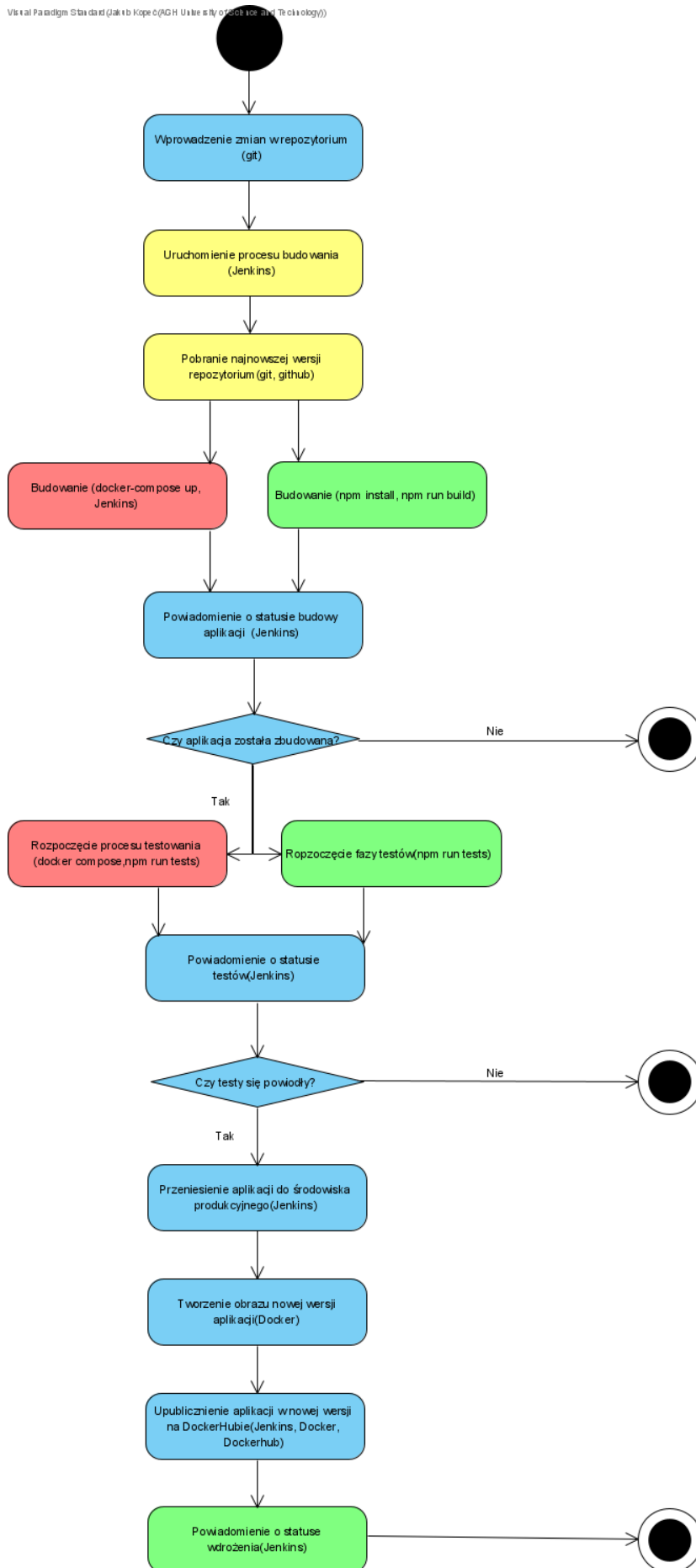
Imię i nazwisko: Jakub Kopeć,  
Grupa laboratoryjna: GL03,  
Data wystania: 27.05.2021

## Podsumowanie planu wdrożeniowego aplikacji deltachat

Wykorzystane technologie:

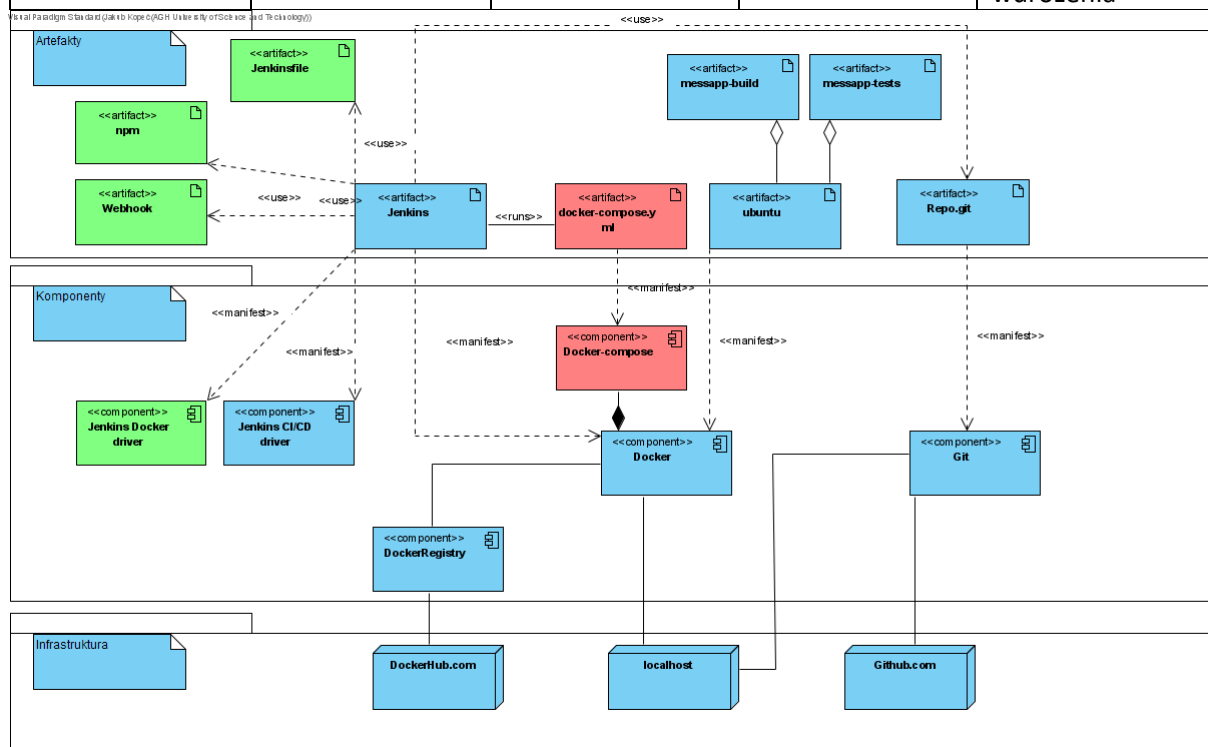
- git, github webhook,
- docker,
- node package manager(npm),
- jenkins z odpowiednimi pluginami,
- ubuntu

W poniższych diagramach oraz tabelach zielony kolor blozków oraz tekstu oznacza dodanie nowych elementów do planu wdrożeniowego, czerwony – usunięcie elementów, które nie są już wykorzystywane, z kolei żółty jest znacznikiem częściowej modyfikacji składowych cyklu wdrożeniowego.



Krok	Technologia	Link	Nr linii	Komentarz
Wprowadzenie zmian w repozytorium	Git, Github	-	-	Zgodnie z planem
Uruchomienie procesu budowania	Jenkins	<a href="#">Jenkinsfile</a>	7	Zamiana kolejności z kolejnym krokiem
Pobranie najnowszej wersji repozytorium	Git	<a href="#">Jenkinsfile</a>	10	Zamiana kolejność z poprzednim krokiem
Budowanie	Docker-compose, Jenkins, npm install, npm run build	<a href="#">Jenkinsfile</a>	12-13	Porzucenie pomysłu budowania obrazu build ponieważ jest to strata czasu i zasobów
Powiadomienie o statusie budowy aplikacji	Jenkins	<a href="#">Jenkinsfile</a>	17	Zgodnie z planem, użytkownik jest powiadamiany niezależnie od wyniku budowy
Rozpoczęcie procesu testowania	Docker compose, npm run tests	<a href="#">Jenkinsfile</a>	27	Porzucenie pomysłu budowania obrazu tests ponieważ jest to strata czasu i zasobów
Powiadomienie o statusie testów	Jenkins	<a href="#">Jenkinsfile</a>	33	Zgodnie z planem, użytkownik jest powiadamiany niezależnie od wyniku testów
Przeniesienie aplikacji do środowiska produkcyjnego	Jenkins	<a href="#">Jenkinsfile</a>	42	Zgodnie z planem
Tworzenie obrazu w nowej wersji	Docker	<a href="#">Jenkinsfile</a> <a href="#">Dockerfile</a> <a href="#">Deploy</a>	46	Zgodnie z planem
Upublicznienie aplikacji w nowej wersji na DockerHubie	Docker, DockerHub, Jenkins	<a href="#">Jenkinsfile</a> <a href="#">Dockerfile</a> <a href="#">Deploy</a>	50-51	Zgodnie z planem

Powiadomienie o statusie wdrożenia	Jenkins	<a href="#">Jenkinsfile</a>	54	Dodanie powiadomienia o statusie wdrożenia (była potrzeba stworzenia powiadomienia, bo docker build/push może też zwrócić błędy). Użytkownik jest powiadamiany niezależnie od wyniku wdrożenia
------------------------------------	---------	-----------------------------	----	--



Grupa	Nazwa kroku	Technologia	Link	Komentarz
Infrastruktura	Gitub.com	Git	-	Zgodnie z planem
	Localhost	Git, Docker, Jenkins	-	Zgodnie z planem
	DockerHub.com	Docker	-	Zgodnie z planem
Komponenty	Git	Git	-	Zgodnie z planem
	Docker	Docker	-	Zgodnie z planem
	Docker-compose	Docker	-	Usunięto z powodu braku potrzeby

				budowania obrazów innych niż ten z sekcji „deploy”
	Jenkins	Jenkins	-	Zgodnie z planem
	Jenkins Docker driver	Jenkins, Docker	-	Plugin do Jenkinsa
	Docker Registry	Docker	-	Zgodnie z planem
Artefakty	Jenkins	Docker, Jenkins	<a href="#">jenkins_dockerfile</a>	Zgodnie z planem
	Jenkinsfile	Jenkins	<a href="#">Jenkinsfile</a>	Artefakt ten został pominięty w poprzednim planie
	Webhook	Github, Jenkins	-	Dodano webhookea z powodu potrzeby automatyzacji rozpoczęcia całego cyklu
	Npm	Npm	-	Artefakt ten został pominięty w poprzednim planie
	Docker-compose.yml	Docker, npm	<a href="#">docker_compose</a>	Usunięto z powodu braku potrzeby budowania obrazów innych niż ten z sekcji „deploy”
	Messapp-build	Docker, npm	<a href="#">build_dockerfile</a>	Usunięto z powodu braku potrzeby budowania obrazów innych niż ten z sekcji „deploy”
	Messapp-tests	Docker, npm	<a href="#">test_dockerfile</a>	Usunięto z powodu braku potrzeby budowania obrazów innych niż ten z sekcji „deploy”
	Ubuntu	Docker, ubuntu	<a href="#">ubuntu_dockerfile</a>	Zmodyfikowano dockerfilea tak, aby mógł zostać

				poprawnie zbudowany oraz aby mógł w sobie uruchomić aplikację
	Repozytorium.git	Git, GitHub	<a href="#">repozytorium</a>	Zgodnie z planem