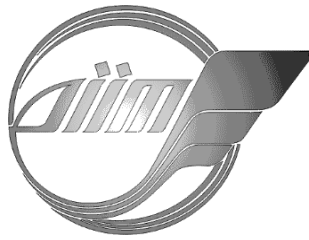


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

**Лабораторна робота №6**

**з дисципліни «Архітектура та проектування програмних засобів»**

**на тему: «Основи моделювання архітектури. Діаграми артефактів та розміщення.»**

Виконав:  
студент гр.ПЗ1911  
Сафонов Д. Є.  
Прийняла:  
Куропятник О. С.

Дніпро, 2020

Тема. Основи моделювання архітектури. Діаграми артефактів та розміщення.

Мета. Ознайомитися з принципами побудови діаграми артефактів та розміщення. Отримати практичні навички з моделювання архітектури системи на основі діаграм артефактів і розміщення.

### **Постановка задачі згідно завдання.**

Для проекту, який розроблявся протягом лабораторних робіт № 1 – 5 побудувати діаграми артефактів та розгортання, виконати їх опис. Для діаграми розгортання вказати, належність артефактів вузлам. Визначити види та стереотипи артефактів.

### **Опис основних артефактів системи.**

В скобках вказані стереотипи артефактів.

Auto(Generic Package) — представляє стан авто — його позицію, швидкість, напрям, тощо.

Driver(Generic Package) — представляє користувача, його режим.

Cam(Generic Package) — представляє камери та інші сенсори, данні з них.

Pedals(Generic Package) — представляє педалі прискорення та сповільнення, їх стан(лінійна сила натиску)

Steering\_wheel(Generic Package) — представляє руль, його стан(кут повороту від нейтрального стану)

Settings\_main(Generic Subprogram) — представляє налаштування користувача(інтерфейс користувача для їх зміни)

Trip\_chooser(Generic Subprogram) — інтерфейс користувача, а саме пасажира або розробника для вибору маршруту.

Engine(Generic Package) — представляє двигун авто, потужність, яка передається на колесну базу.

Controller(Main Program) — Контрольний вузел, поєднує усі частки системи так контролює їх.

Database(Database) — зберігає налаштування користувача.

## Діаграми артефактів та розміщення, їх опис згідно завдання.

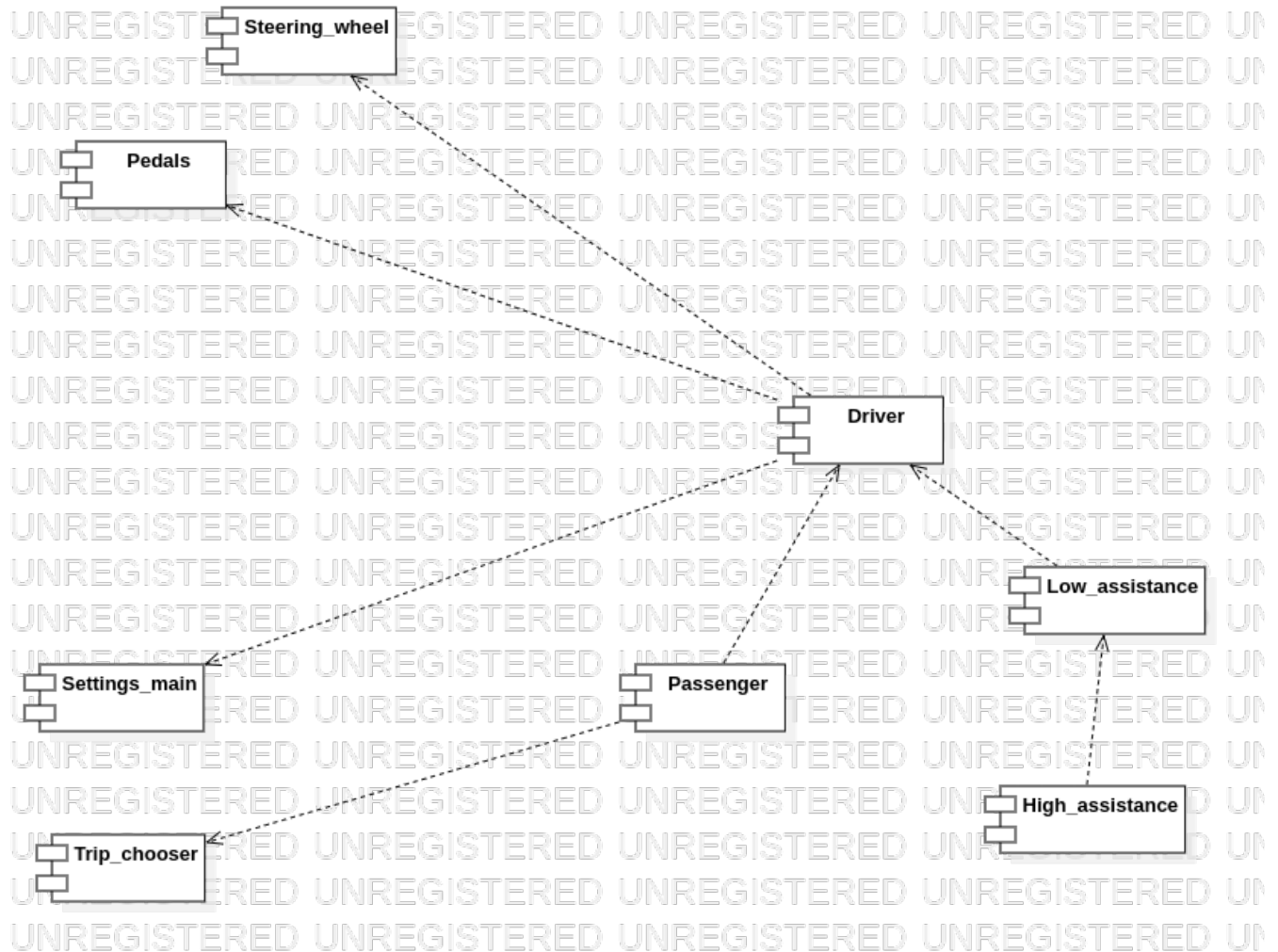


Рисунок 1 (Діаграма артефактів (компонентів) для рівня представлення)

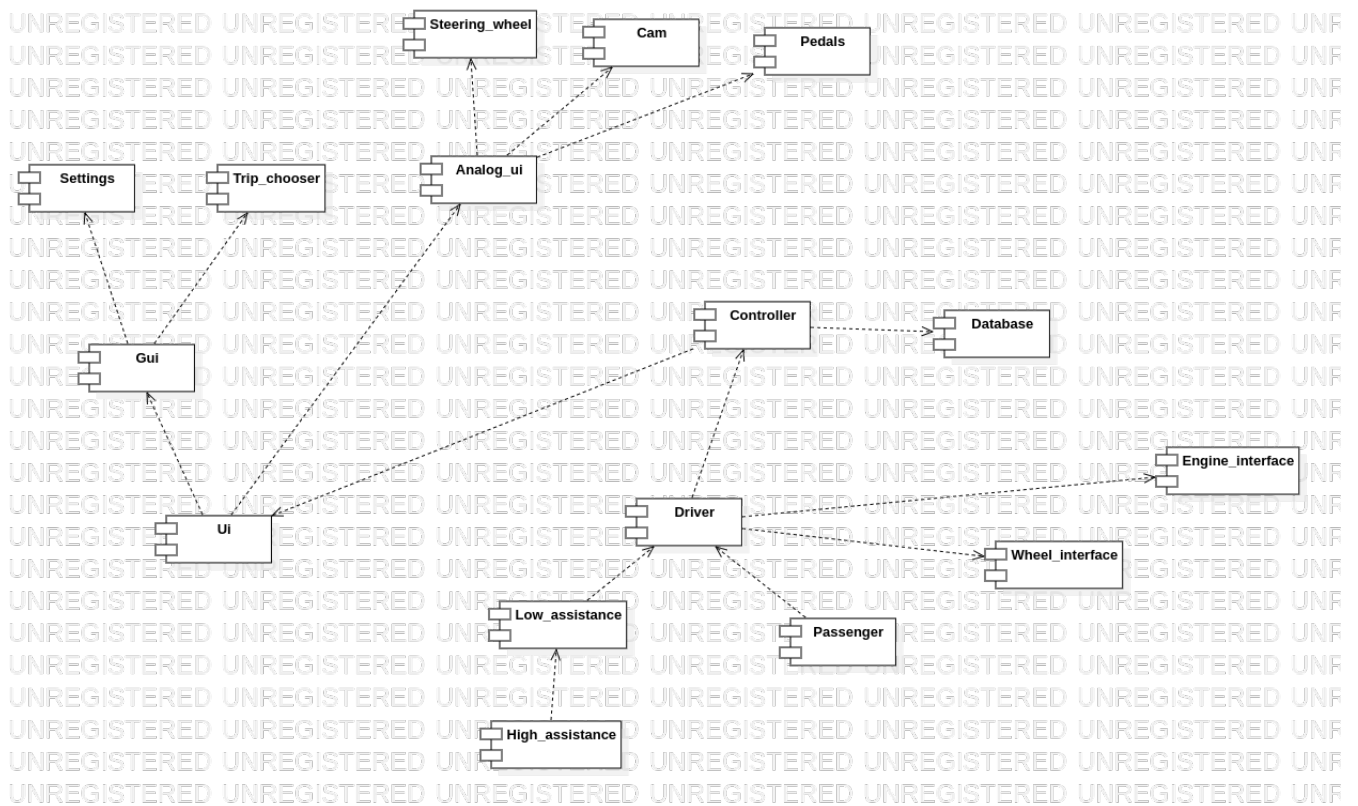


Рисунок 2(Діаграма артефактів (компонентів) для рівня логіки)



Рисунок 3(Діаграма артефактів (компонентів) для рівня даних)

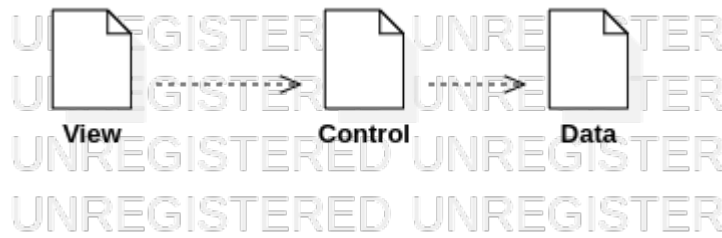


Рисунок 4(Загальна діаграма компонентів проекту)

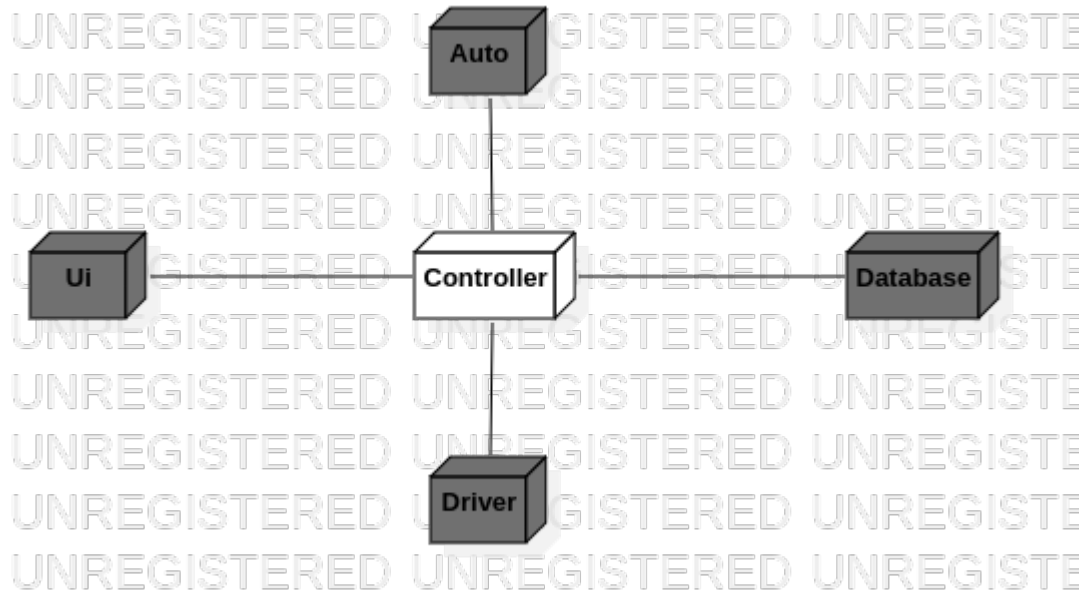


Рисунок 5(Діаграма розгорткування )

Вузел Ui — включає у себе усі компоненти, які пов'язані із налаштуванням автомобіля, а саме: Settings\_main, Trip\_chooser.

Controller — самостійний вузел, який контролює усі інші та є єдиним зв'язком між ними.

Driver — включає у себе усі компоненти, які пов'язані із контролем автомобіля, а саме: Pedals, Steering\_wheel.

Auto — включає у себе компоненти, які є інтерфейсами до внутрішнього контролю автомобіля, а саме: Engine\_interface, wheel\_interface.

Database — самостійний вузел, який зберігає налаштування користувача.

## Аналіз результатів та висновки щодо важливості моделювання архітектури.

На мою думку цей тип діаграм дуже схожий на діаграми класів з ЛР2, можна навіть сказати що ці діаграми у сумі є великою часткою діаграми класів. Виходячи з цього можна сказати, що ці діаграми є ітеративним нащадком діаграми класів(на мою думку), та є обов'язковими для складних систем, але в дуже простих програмах можна обійтись лише діаграмою класів. Окрім

цього діаграма розгорткування необхідна в усіх фізичних системах, бо вона демонструє фізичні зв'язки між елементами системи. Також діаграми артефактів потрібні для більшої систематизації складних систем, наприклад для того, щоб об'єднувати споріднені артефакти.