

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ ІМ.ІГОРЯ
СІКОРСЬКОГО»

Кафедра
автоматизованих систем обробки інформації та управління

КУРСОВА РОБОТА

з дисципліни
“Технології розробки WEB-застосунків на платформі Java”
на тему
"Система керування турнікетом
на станції метро"

Керівник : Іванова Л.М

Виконав: Швець Е.Я.

Допущений до захисту

"__" _____ 2019 _____
підпис

студент гр. ІП-61, ФІОТ

3 курс

№ ІП-6128

Підпис:

Захистив з оцінкою

_____ оцінка _____ підпис
"__" _____ 2019

Київ 2019

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

(назва вищого навчального закладу)

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Технології розробки WEB-застосунків на платформі Java

Напрямок "Програмна інженерія"

Курс 3 Група ІІІ-61

Семестр 6

ЗАВДАННЯ

на курсову роботу студента

Швеця Едуарда Ярославовича

(прізвище, ім'я, по батькові)

1. Тема роботи Система керування турнікетом на станції метро

2. Строк здачі студентом закінченої роботи _____

3. Вихідні дані до роботи вихідні дані відсутні.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів курсової роботи	Термін виконання етапів роботи	Підписи керівника, студента
1.	Отримання теми курсової роботи		
2.	Підготовка ТЗ		
3.	Пошук та вивчення літератури з питань курсової роботи		
4.	Розробка алгоритму вирішення задачі		
6.	Узгодження алгоритму з керівником		
5.	Розробка сценарію роботи програми		
6.	Узгодження сценарію роботи програми з керівником		
7.	Узгодження з керівником інтерфейсу користувача		
8.	Розробка програмного забезпечення		
9.	Налагодження розрахункової частини програми		
10.	Розробка та налагодження інтерфейсної частини програми		
11.	Узгодження з керівником набору тестів для контрольного прикладу		
12.	Тестування програми		
13.	Підготовка пояснювальної записки		
14.	Здача курсової роботи на перевірку		
15.	Захист курсової роботи		

Студент

(підпис)

Швець Е.Я.

(прізвище, ім'я, по батькові)

Керівник

(підпис)

Іванова Л.М.

(прізвище, ім'я, по батькові)

"__" _____ 20__ р.

АНОТАЦІЯ

Пояснювальна записка до курсової роботи: 51 сторінка, 17 рисунків, 4 посилання.

Об'єкт дослідження: концепція створення WEB-застосунків на платформі Java

Мета роботи: навчитися створювати WEB-застосунки на мові програмування Java, використовуючи Spring Framework. Для полегшення розробки представити діючі об'єкти та їх компоненти у вигляді зв'язної ієрархії класів. Організувати відношення між класами та взаємодію класів між собою. Створити відповідну базу даних та організувати доступ до бази даних з програми.

Були вивчені основні методи та принципи написання WEB-додатку на Java з використанням Spring Framework. Приведені змістовні постановки задач, описано детальний процес розв'язання кожної з них.

У розділі «Вступ» розкрито призначення курсової роботи та основна сутність вирішуваної задачі.

У розділі «Постановка задачі» висунуто основні вимоги до програмного забезпечення, перелічено методи, які воно має виконувати.

У розділі «Теоретичні викладки» розкрито теоретичні основи WEB-програмування на Java за допомогою Spring Framework.

У розділі «Опис алгоритму» детально описано загальний алгоритм програми, а також алгоритми основних підзадач.

У розділі «Опис програмного забезпечення» наведено UML-діаграму класів, діаграму архітектури, use-case діаграму, а також детальний опис усіх класів програми.

У розділі «Тестування програмного забезпечення» надано план тестування, а також його результати.

У розділі «Інструкція користувача» надано детальну інструкцію по використанню програмного забезпечення.

У розділі «Висновки» описано результати, отримані після кожного етапу розробки програмного забезпечення

У розділі «Перелік посилань» перелічено літературні джерела, які було використано в ході написання даної роботи.

У розділі «Додаток А» наведено посилання на GitHub, на якому знаходиться остання версія застосунку.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМ. ІГОРЯ СІКОРСЬКОГО”
Кафедра автоматизованих систем обробки інформації та управління

РОЗРОБИВ

Студент

Швець Е.Я.

(підпис) (ініціали, прізвище)

“ ____ ” _____ 2019 р.

ЗАТВЕРДИВ

Керівник

_____ Іванова Л.М.

(підпис) (ініціали, прізвище)

“ ____ ” _____ 2019 р.

Система керування турнікетом на станції метро

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання курсової роботи

з дисципліни:

«Технології розробки WEB-застосувань на платформі Java»

Шифр КПІ ІП-6128.ТЗ

на 15 сторінках

Київ 2019

ЗМІСТ

1	Найменування та область застосування.....	4
2	Підстави для розробки.....	5
2.1	Перелік документів, на підставі яких ведеться розробка	5
2.2	Найменування і умовне позначення теми розробки.....	5
3	Призначення розробки.....	6
3.1	Функціональне призначення розробки	6
3.2	Експлуатаційне призначення	6
4	Вимоги до програмного забезпечення	7
4.1	Вимоги до функціональних характеристик.....	7
4.1.1	Вимоги до складу виконуваних функцій.....	7
4.1.2	Вимоги до організації вхідних даних.....	8
4.1.3	Вимоги до організації вихідних даних.....	8
4.2	Вимоги до надійності.....	8
4.2.1	Вимоги до надійного функціонування програми	8
4.2.2	Час відновлення після відмови	8
4.2.3	Відмови через некоректні дії користувача	9
4.3	Умови експлуатації	9
4.3.1	Кліматичні умови експлуатації.....	9
4.3.2	Вимоги до видів обслуговування	9
4.3.3	Вимоги до кількості та кваліфікації персоналу	9
4.4	Вимоги до складу і параметрів технічних засобів.....	9
4.5	Вимоги до інформаційної і програмної сумісності	10
4.6	Вимоги до маркування та упаковки	10
4.7	Вимоги до транспортування та зберігання.....	10
5	Вимоги до програмної документації.....	11
5.1	Попередній склад програмної документації	11
6	Техніко-економічні показники.....	12

7	Стадії та етапи розробки.....	13
7.1	Стадії розробки.....	13
7.2	Етапи розробки.....	13
7.3	Зміст роботи по етапам.....	13
8	Порядок контролю і прийому	15
8.1	Види досліджень	15
8.2	Загальні вимоги до прийняття роботи	15

1 НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Найменування – «Система керування турнікетом на станції метро».

Програма призначена для обмеження і реєстрації входу-виходу об'єктів (людей) на заданій території (станція метрополітену) через «точки проходу» (турнікети).

2 ПІДСТАВИ ДЛЯ РОЗРОБКИ

2.1 Перелік документів, на підставі яких ведеться розробка

Підставою для проведення розробки являється наказ № 4 від 5 травня 2017 року. Наказ затверджено на засіданні кафедри АСОІУ, далі Замовник і прийнято до виконання студентом Швецем Едуардом Ярославовичем, далі Виконавець, дата затвердження ТЗ.

2.2 Найменування і умовне позначення теми розробки

Найменування теми розробки – «Система керування турнікетом на станції метро».

Умовне позначення теми розробки (шифр теми) – «СКТНСМ 0.1».

3 ПРИЗНАЧЕННЯ РОЗРОБКИ

3.1 Функціональне призначення розробки

Функціональним призначенням розробки програми є створення системи керування турнікетом на станції метро для обмеження і реєстрації входу-виходу об'єктів (людей) на заданій території (станція метрополітену) через «точки проходу» (турнікети).

3.2 Експлуатаційне призначення

Програма може експлуатуватися як система контролю доступу на об'єкт довільної величини і рівня складності.

Також користувачами програми можуть бути співробітники та студенти кафедри АСОІУ, абітурієнти та випускники кафедри для демонстрації роботи системи керування турнікетом на станції метро.

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Вимоги до функціональних характеристик

4.1.1 Вимоги до складу виконуваних функцій

Програма повинна забезпечувати можливість виконання нижче наведених функцій:

а) емуляція проходу через турнікет метро

1) прохід здійснюється за допомогою натискання лівою кнопкою миші на кнопки інтерфейсу, які відповідають за подачу карток до приймача карток, та подальшим натисканням на один з 3-х сенсорів руху (3 жовтих кола на основному екрані програми)

2) логіка руху людини через турнікет побудована згідно з реальною схемою роботи пристроя «точки проходу»

б) навігація по інтерфейсу

1) в залежності від натиснутої кнопки в основному інтерфейсі програми здійснюється потрібна дія

2) користувач може вибрати один з трьох варіантів подачі картки метрополітену до приймача карток (подача дійсної картки з достатньою кількістю поїздок, подача недійсної картки та подача дійсної картки з недостатньою кількістю поїздок)

3) якщо індикатор проходу буде в стані «pass», за допомогою натискання на один з сенсорів руху буде здійснений 5-секундний прохід через турнікет, якщо ж індикатор проходу буде в стані «stop», то прохід через турнікет блокується, та подається спеціальний сигнал

4) інформація згідно проходів з кожного турнікету надсилається за допомогою кнопки інтерфейсу «toDB»

5) адміністратор може ввести свій логін та пароль у спеціальні поля та натиснути кнопку інтерфесу «login». Буде виконаний вхід до автоматизованої системи керування метрополітену для отримання інформації про проходи через відповідні турнікети

б) кнопка «logout» дає змогу адміністратору вийти зі свого особистого кабінету в автоматизованій системі керування метрополітену

4.1.2 Вимоги до організації вхідних даних

Кожного разу, коли програма запускається, вона створює комплекс класів для обробки даних бази даних:

а) логіни та паролі адміністраторів автоматизованої системи керування метрополітеном

б) інформацію про кожен турнікет

в) інформацію про проходження через турнікети

г) інформацію про обслуговуючий персонал

д) інформацію про станції метро

4.1.3 Вимоги до організації вихідних даних

Вихідні данні відсутні.

4.2 Вимоги до надійності

4.2.1 Вимоги до надійного функціонування програми

Надійне функціонування програми повинно бути забезпечене виконанням замовником сукупності організаційно-технічних заходів, перелік яких наведено нижче:

а) організацією безперебійного струмопостачання технічних засобів;

б) використанням ліцензійного програмного забезпечення;

в) використанням своєчасно оновленого програмного забезпечення;

4.2.2 Час відновлення після відмови

Час відновлення після відмови, що спричинена збоєм електропостачання технічних засобів, не крахом операційної системи, не повинно перевищувати 1 години за умови виконання умов експлуатації технічних і програмних засобів.

Час відновлення після відмови, що спричинена неполадкою технічних засобів, крахом операційної системи, не повинно перевищувати часу, який потрібен на ліквідацію неполадок технічних засобів та переустановленням програмних засобів.

4.2.3 Відмови через некоректні дії користувача

Відмови програми можливі внаслідок некоректних дій користувача системи. Задля запобігання відмов програми через вище вказані причини слід обмежити коло користувачів без прав доступу «Адміністратор».

4.3 Умови експлуатації

4.3.1 Кліматичні умови експлуатації

Кліматичні умови експлуатації, при яких повинні забезпечуватися задані характеристики, повинні задовольняти вимогам, що заявлені до технічних засобів зокрема до умов їх експлуатації.

4.3.2 Вимоги до видів обслуговування

Див. Вимоги до забезпечення надійного функціонування програми.

Програма не потребує будь-яких видів обслуговування.

4.3.3 Вимоги до кількості та кваліфікації персоналу

Мінімальна кількість персоналу, що необхідний для роботи програми, повинна складати не менше 1 штатної одиниці – користувач системи з правами «Адміністратор».

Користувач програми з правами «Адміністратор» повинен мати навички роботи з командним рядком операційної системи, досвід програмування на Java та повинен бути знайомим зі Spring Framework.

Персонал повинен бути атестованим на II кваліфікаційну групу по електробезпеці (для роботи з кафедральним обладнанням).

4.4 Вимоги до складу і параметрів технічних засобів

До комп'ютера, на якому має бути розгорнуто програмне забезпечення, висуваються наступні вимоги:

- а) 64-розрядний процесор з тактовою частотою не нижче 1.4 ГГц;
- б) достатній об'єм оперативної пам'яті (не менше 2 ГБ);
- в) достатній об'єм жорсткого диску (не менше 32 ГБ).

4.5 Вимоги до інформаційної і програмної сумісності

Вимоги до додаткового ПЗ не висуваються

4.6 Вимоги до маркування та упаковки

Вимоги до маркування та упаковки не висуваються

4.7 Вимоги до транспортування та зберігання

Вимоги до транспортування та зберігання не висуваються

5 ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

5.1 Попередній склад програмної документації

Склад програмної документації повинен включати в себе:

- а) технічне завдання;
- б) керівництво користувача.

6 ТЕХНІКО-ЕКОНОМІЧНІ ПОКАЗНИКИ

Орієнтовна економічна ефективність не обчислюється.

7 СТАДІЇ ТА ЕТАПИ РОЗРОБКИ

7.1 Стадії розробки

Розробка повинна бути проведена в три стадії:

- а) розробка технічного завдання;
- б) робоче проектування;
- в) впровадження.

7.2 Етапи розробки

На стадії розробки технічного завдання повинен бути виконаний етап розробки, узгодження і затвердження технічного завдання.

На стадії робочого проектування повинен бути виконаний наведений нижче перелік робіт:

- а) розробка програми;
- б) тестування програми;
- в) розробка програмної документації.

На стадії впровадження виконується передача програми кінцевому замовнику.

7.3 Зміст роботи по етапам

На етапі розробки технічного завдання повинні бути виконані наступні роботи:

- а) постановка задачі;
- б) визначення і уточнення вимог до технічних засобів;
- в) визначення вимог до програми;
- г) визначення стадій, етапів і строків розробки програми та документації на неї;
- д) вибір мов програмування;
- е) узгодження та затвердження технічного завдання.

На етапі розробки програми повинні бути виконані роботи по програмуванню і налагодженню програми.

На етапі тестування повинні бути виконані роботи по тестуванню усіх основних функцій програми.

На етапі досліджень програми повинні бути виконані наведені нижче види робіт:

- а) розробка, узгодження і затвердження програми і методик дослідження;
- б) проведення досліджень;
- в) коригування програми і програмної документації по результатам досліджень.

На етапі розробки програмної документації повинна бути виконана розробка програмних документів.

На етапі підготовки та передачі програми повинна бути виконана робота по підготовці та передачі програми і програмної документації в експлуатацію на об'єктах кафедри АСОІУ.

8 ПОРЯДОК КОНТРОЛЮ І ПРИЙОМУ

8.1 Види досліджень

Приймально-здавальні випробування програмного продукту мають проводитися згідно з розробленою виконавцем і узгодженою із замовником “Програмою та методикою випробувань” на об’єкті замовника.

8.2 Загальні вимоги до прийняття роботи

На основі проведених досліджень Виконавець спільно з Замовником підписують Акт прийому-здачі програми в експлуатацію.

Пояснювальна записка до курсової роботи

на тему: Система керування турнікетом на станції метро

Київ 2019

ЗМІСТ

Вступ.....	4
1 Постановка задачі.....	5
2 Теоретичні викладки.....	6
3 Опис алгоритмів	8
3.1 Алгоритм проходження через турнікет	8
3.2 Алгоритм авторизації адміністратора автоматизованої системи керування метрополітеном	9
4 Опис програмного забезпечення	10
4.1 Опис діаграми класів програмного забезпечення.....	10
4.2 Опис класів.....	18
4.2.1 Класи пакету com.demo	18
DemoApplication – головний клас програми, що містить метод main.	18
ControllerSpring – контролер програми.....	18
4.2.2 Класи пакету commands.....	18
4.2.3 Класи пакету converter	18
4.2.4 Класи пакету factories	18
4.2.5 Класи пакету card	19
4.2.6 Класи пакету DAO	19
4.2.7 Класи пакету db	19
4.2.8 Класи пакету dbEntity	19
4.2.9 Класи пакету state.....	19
4.2.10 Класи пакету turnstileSensors	20
4.2.11 Класи пакету model	20
4.3 Діаграма архітектури застосунку	21
4.4 Use-case діаграма.....	22
4.5 Опис бази даних	23
4.5.1 Таблиця PASSES_STATISTIC	23

4.5.2	Таблиця TURNSTILES	23
4.5.3	Таблиця METRO_STATIONS	23
4.5.4	Таблиця SERVICE_STAFF	23
4.5.5	Таблиця POSITION	24
4.5.6	Таблиця ADMINS	24
5	Тестування програмного забезпечення	25
5.1	План тестування	25
5.2	Приклади тестування	25
6	Інструкція користувача	27
6.1	Призначення програми	27
6.2	Вимоги до системи	27
6.3	Інструкція по роботі з програмою	27
	Висновки	29
	Перелік посилань	30
	Додаток А	31

ВСТУП

Основним завданням даної курсової роботи було створити WEB-застосунок з використанням Spring Framework.

Також потрібно було застосовувати відповідні шаблони проектування та створити базу даних автоматизованої системи керування метрополітену та DAO (data access object) до неї.

1 ПОСТАНОВКА ЗАДАЧІ

Розробити систему керування турнікету на станції метро.

За допомогою турнікету контролюється прохід пасажирів у метро та збирається вхідна платня. Турнікет має приймач карток, пристрій для перекривання доступу, таймер, три оптичні датчики для визначення проходження пасажира, пристрій для подачі звукових сигналів, індикатори «Прохід» та «Стоп».

У початковому стані турнікета висвічується індикатор «Стоп», індикатор «Прохід» не горить. Якщо один з датчиків надсилає сигнал, прохід через турнікет одразу ж перекривається та надсилається попереджувальний звуковий сигнал. Для того, щоб пройти, пасажир повинен помістити картку в приймач карток. Турнікет зчитує з неї дані: термін придатності карти та кількість «одиниць» на ній. Якщо дані не зчитуються, картка прострочена або заблокована, то вона повертається пасажирові, і турнікет залишається в початковому стані. В іншому випадку з картки списується одна одиниця, картка повертається з приймача, індикатор «Стоп» гасне, засвічується індикатор «Прохід», і пасажир може пройти через турнікет. Отримавши від одного з датчиків сигнал, турнікет очікує час, визначений на проходження пасажира (5 секунд), після чого він повертається в початковий стан.

Наявність трьох датчиків в турнікеті гарантує, що в разі проходження пасажира, хоч би один з них подасть сигнал. Під час проходження пасажира можлива ситуація, коли всі три датчики посилають сигнали. У цьому випадку приймається тільки перший сигнал і від моменту його прийому відраховується призначений час. Решта сигналів ігнорується.

Турнікет заносить у свою пам'ять кількість усіх сплачених проходжень. В кінці робочого дня він передає усі інформацію, накопичену за день, в автоматизовану систему керування метрополітену.

2 ТЕОРЕТИЧНІ ВИКЛАДКИ

WEB-застосунок був написаний за допомогою Spring Framework. Spring забезпечує вирішення багатьох завдань, з якими стикаються Java-розробники і організації, які хочуть створити інформаційну систему, засновану на платформі Java. Через широку функціональність важко визначити найбільш значущі структурні елементи, з яких він складається. Spring не повністю пов'язаний з платформою Java Enterprise, незважаючи на його масштабну інтеграцію з нею, що є важливою причиною його популярності.

Spring, ймовірно, найбільш відомий як джерело розширень (features), потрібних для ефективної розробки складних бізнес-додатків поза великовагових програмних моделей, які історично були домінуючими в промисловості. Ще один плюс в тому, що він ввів раніше невикористовувані функціональні можливості в сьогоденні панівні методи розробки, навіть поза платформи Java.

Цей фреймворк пропонує послідовну модель і робить її придатною до більшості типів додатків, які вже створені на основі платформи Java. Вважається, що Spring реалізує модель розробки, засновану на кращих стандартах індустрії, і робить її доступною в багатьох областях Java.

Spring має власну MVC-платформу веб-додатків. Клас DispatcherServlet є основним контролером фреймворка і відповідає за делегування управління різних інтерфейсах, на всіх етапах виконання HTTP-запиту.

Spring MVC є фреймворком, орієнтованим на запити. У ньому визначені стратегічні інтерфейси для всіх функцій сучасної запитання-орієнтованої системи. Мета кожного інтерфейсу - бути простим і зрозумілим, щоб користувачам було легко його заново імплементувати, якщо вони того побажають. MVC прокладає шлях до більш чистого front-end-коду. Всі інтерфейси тісно пов'язані з Servlet API. Цей зв'язок залишає особливості Servlet API доступними для розробників, полегшуючи все ж роботу з ним.

Також, були використані шаблони проектування. Шаблон проектування - це повторювана архітектурна конструкція, що представляє собою рішення проблеми

проектування в рамках деякого часто виникаючого контексту. Об'єктно-орієнтовані шаблони показують відносини і взаємодії між класами або об'єктами, без визначення того, які кінцеві класи або об'єкти додатку будуть використовуватися.

Була розроблена база даних та DAO до неї. DAO (data access object) - це об'єкт, який надає абстрактний інтерфейс до якого-небудь типу бази даних або механізму зберігання. Певні можливості надаються незалежно від того, який механізм зберігання використовується і без необхідності спеціальним чином відповідати цьому механізму зберігання. Цей шаблон проектування застосовується до безлічі мов програмування, більшості програмного забезпечення, що потребує зберігання інформації більшої частини баз даних, але традиційно цей шаблон пов'язують з додатками на платформі Java Enterprise Edition, взаємодіючими з реляційними базами даних через інтерфейс JDBC, тому що він з'явився в рекомендаціях від фірми Sun Microsystems.

Врахувавши всі правила та принципи побудови застосунку на Java за допомогою Spring Framework та шаблонів проектування, мною був створений якісний прототип системи керування турнікета метро.

3 ОПИС АЛГОРИТМІВ

3.1 Алгоритм проходу через турнікет

1. ПОЧАТОК

2. Завантажити web-додаток.

3. Відобразити основний інтерфейс програми

3.1. ЯКЩО натиснута кнопка “Attach a valid card”, ТО перейти до пункту 4.

3.2. ЯКЩО натиснута кнопка “Attach an invalid card (expiration date)”, ТО перейти до пункту 4.

3.3. ЯКЩО натиснута кнопка “Attach an invalid card (number of trips)”, ТО перейти до пункту 4.

3.4. ЯКЩО натиснута кнопка “Sensor1”, ТО перейти до пункту 5.

3.5. ЯКЩО натиснута кнопка “ Sensor2”, ТО перейти до пункту 5.

3.6. ЯКЩО натиснута кнопка “ Sensor3”, ТО перейти до пункту 5.

3.7. ЯКЩО натиснута кнопка “ DB”, ТО:

3.7.1. ЯКЩО були зроблені проходження через турнікети, ТО завантажити інформацію про проходження через турнікети до бази даних та перейти до пункту 3.

3.7.2. ЯКЩО не були зроблені проходження через турнікети, ТО перейти до пункту 3

4. Почати роботу приймача карток.

4.1. ЯКЩО картка є недійсною, або кількість поїздок на карті недостатня, ТО не дозволити прохід, вивести повідомлення про помилку та перейти до пункту 3.

4.2. ЯКЩО картка є дійсною та на картці є достатня кількість поїздок, ТО змінити індикатор зі «STOP» на «PASS», заблокувати усі кнопки крім сенсорів та перейти до пункту 5.

5. Почати роботу сенсорів.

5.1. ЯКЩО стан турнікету дорівнює «STOP», ТО заблокувати прохід, вивести повідомлення про блокування проходу та перейти до пункту 3.

5.2. ЯКЩО стан турнікету дорівнює «PASS», ТО запустити таймер на 5 секунд, заблокувати інші сансори на реагування, та після 5 секунд занести інформацію про проходження в турнікет та повернутися до пункту 3.

3.2 Алгоритм авторизація адміністратора автоматизованої системи керування метрополітену

1. ПОЧАТОК

2. Завантажити web-додаток.

3. Відобразити основний інтерфейс програми

3.1. ЯКЩО натиснута кнопка “Submit”, ТО перейти до пункту 4.

4. Почати завантаження даних з полів «Login» та «Password».

4.1. ЯКЩО дані в поля не були введені, ТО вивести повідомлення про помилку та необхідність вводу даних до форми «Login» та «Password» та повернутися до пункту 3.

4.2. ЯКЩО дані в поля були введені, ТО перейти до пункту 5.

5. Почати перевірку правильності логіну та паролю.

5.1. ЯКЩО пароль або логін не збіглися з існуючими даними для авторизації, ТО вивести повідомлення про помилку та перейти до пункту 3.

5.2. ЯКЩО пароль та логін збіглися з існуючими даними для авторизації, ТО виконати вхід до автоматизованої системи керування метрополітену.

5.2.1. ЯКЩО була натиснута кнопка «Logout», то вийти з особистого кабінету адміністратора та перейти до пункта 3.

4 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕНИЯ

4.1 Опис діаграми класів програмного забезпечення

Опис класів та їх зв'язків подано на рисунках 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11 та 4.12 за допомогою UML діаграми класів.

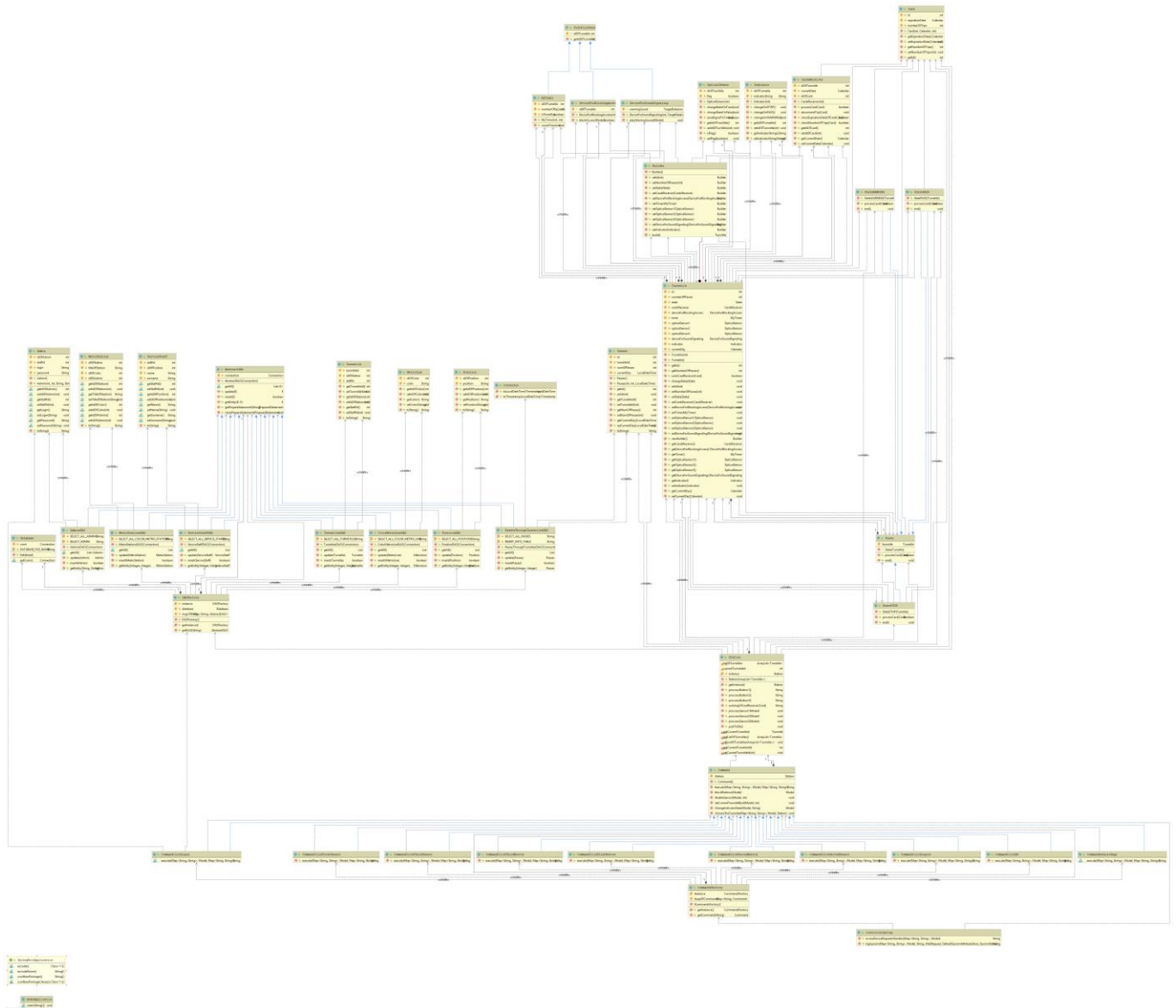


Рисунок 4.1 – загальна UML діаграма класів

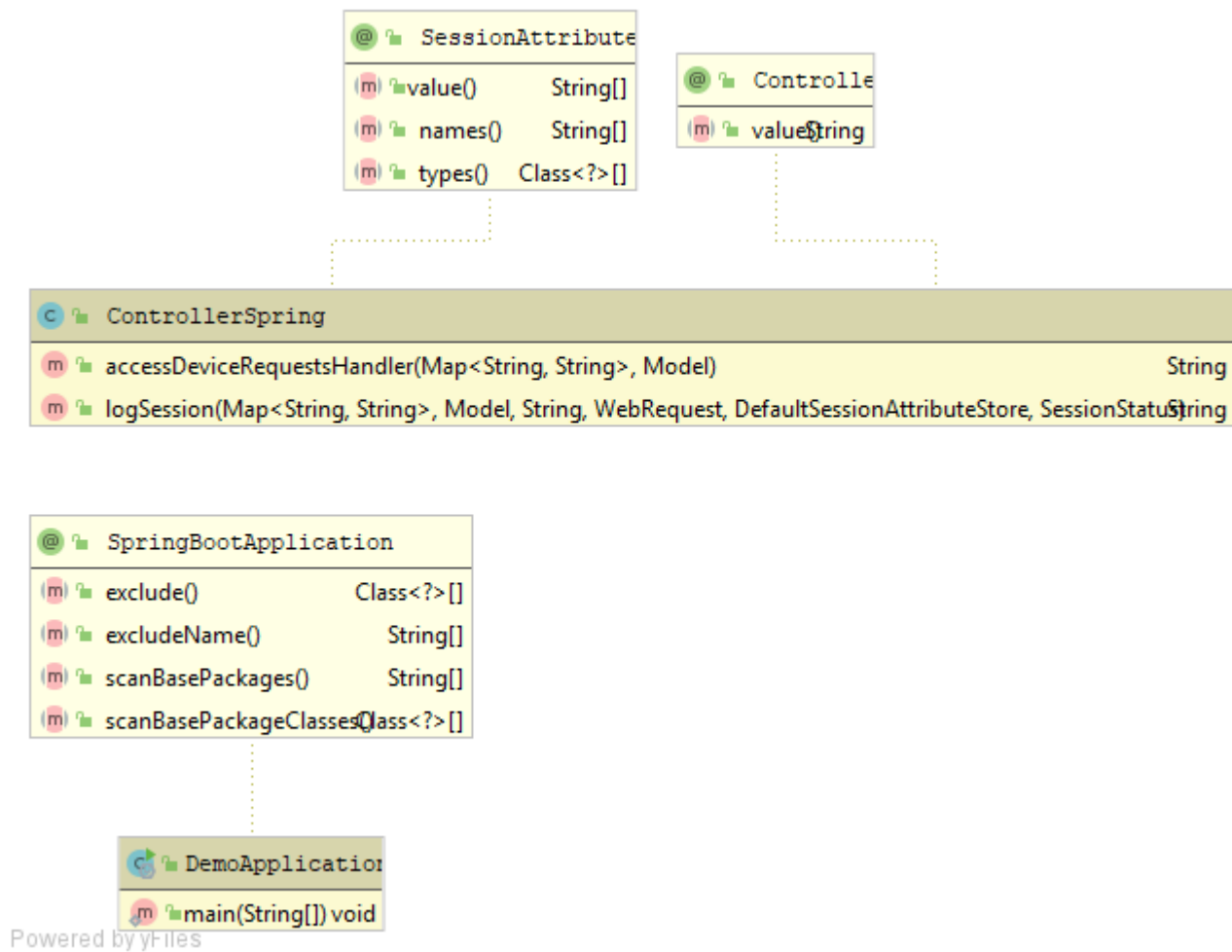


Рисунок 4.2 - UML діаграма класів пакету com.demo

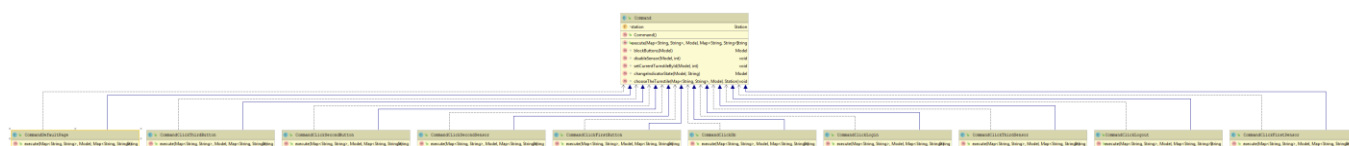


Рисунок 4.3 - UML діаграма класів пакету commands

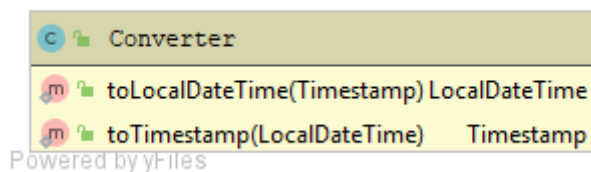


Рисунок 4.4 - UML діаграма класів пакету converter

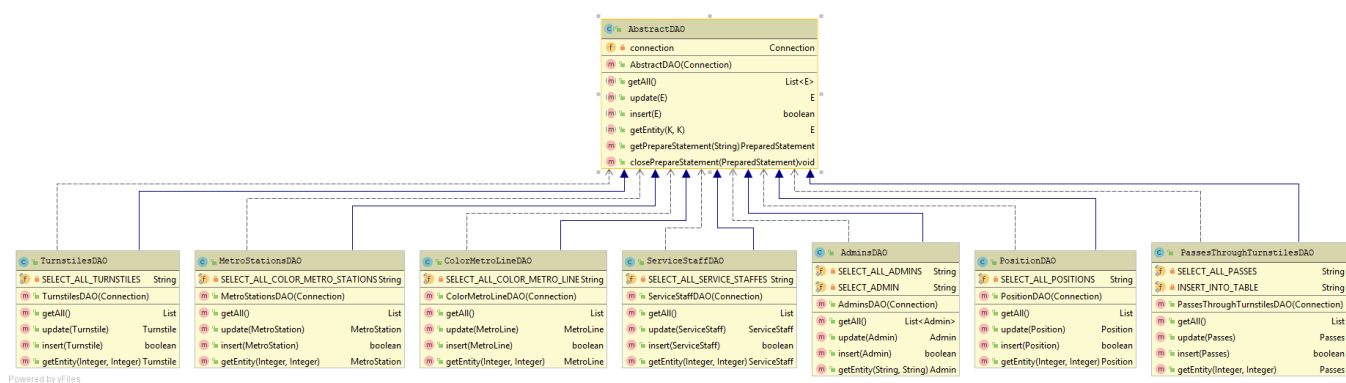


Рисунок 4.5 - UML діаграма класів пакету DAO

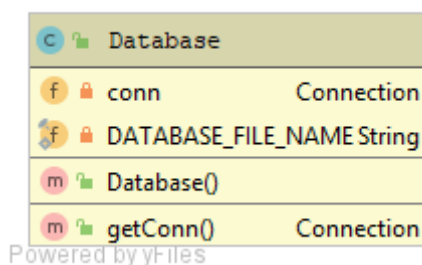
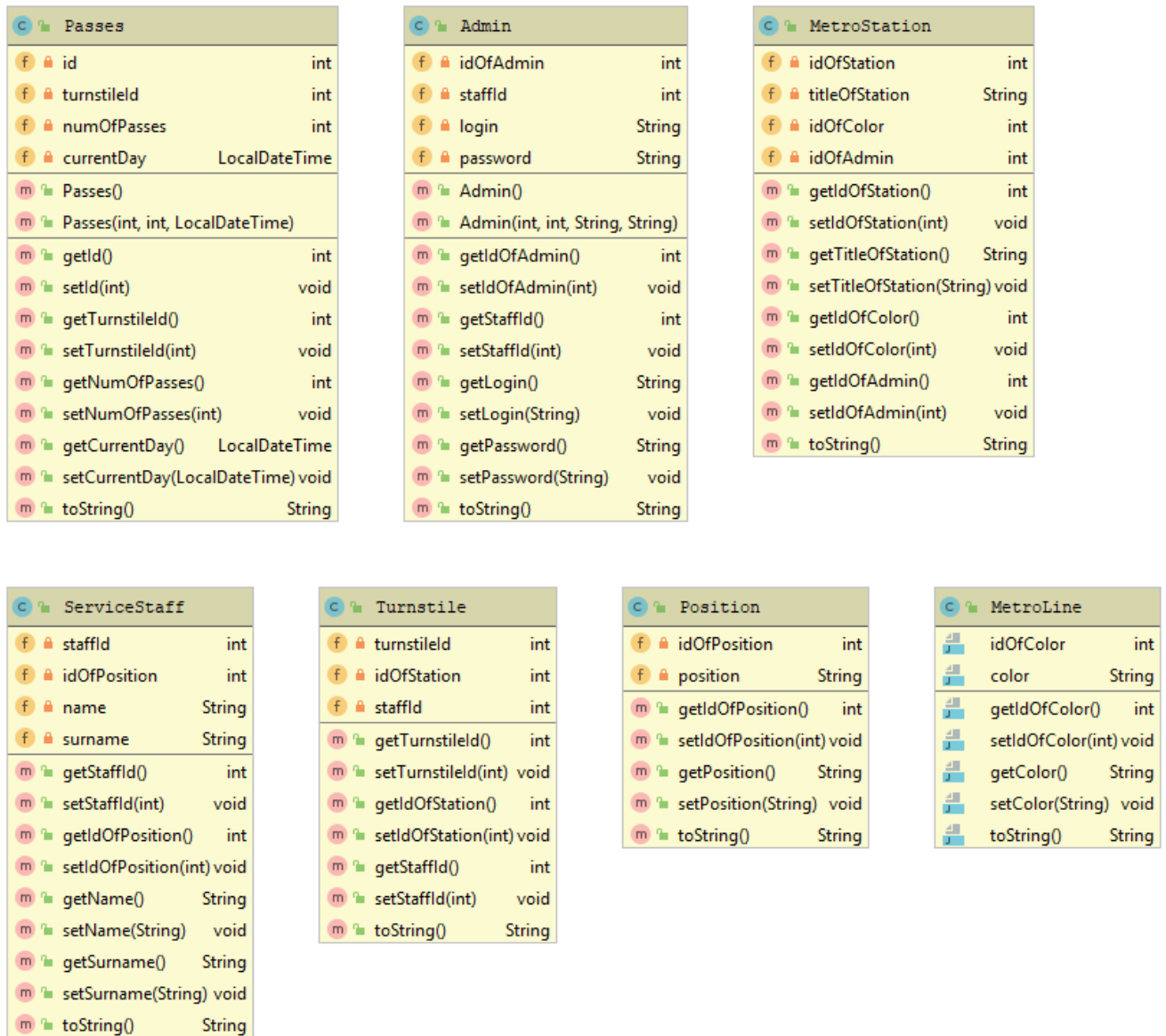
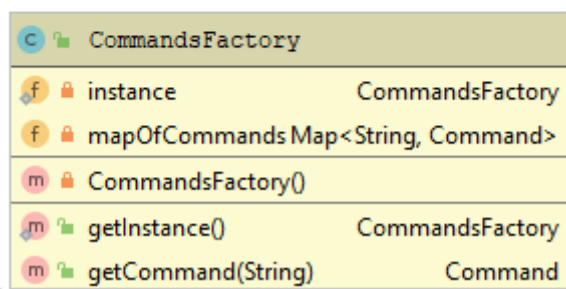
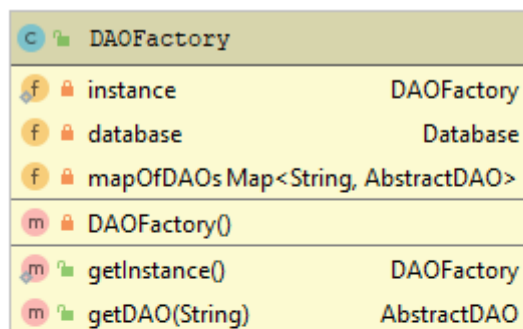


Рисунок 4.6 - UML діаграма класів пакету db



Powered by yfiles

Рисунок 4.7 - UML діаграма класів пакеты dbEntity



Powered by yFiles

Рисунок 4.8 - UML діаграма класів пакету factories

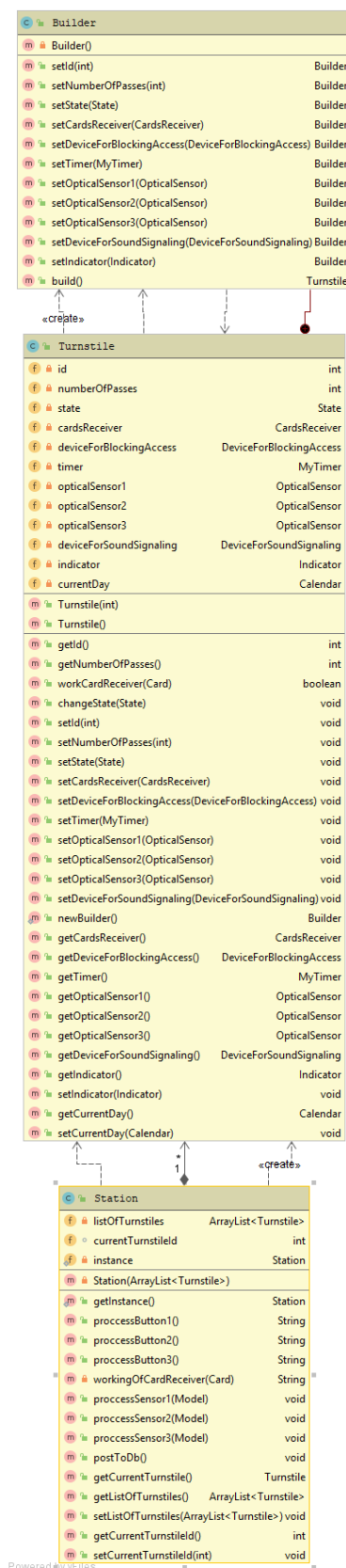


Рисунок 4.9 - UML діаграма класів пакету model

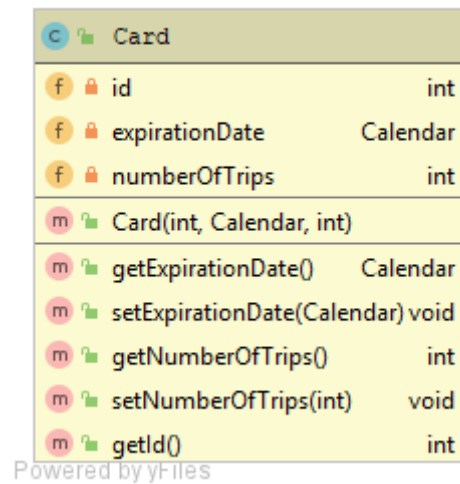


Рисунок 4.10 - UML діаграма класів пакету card

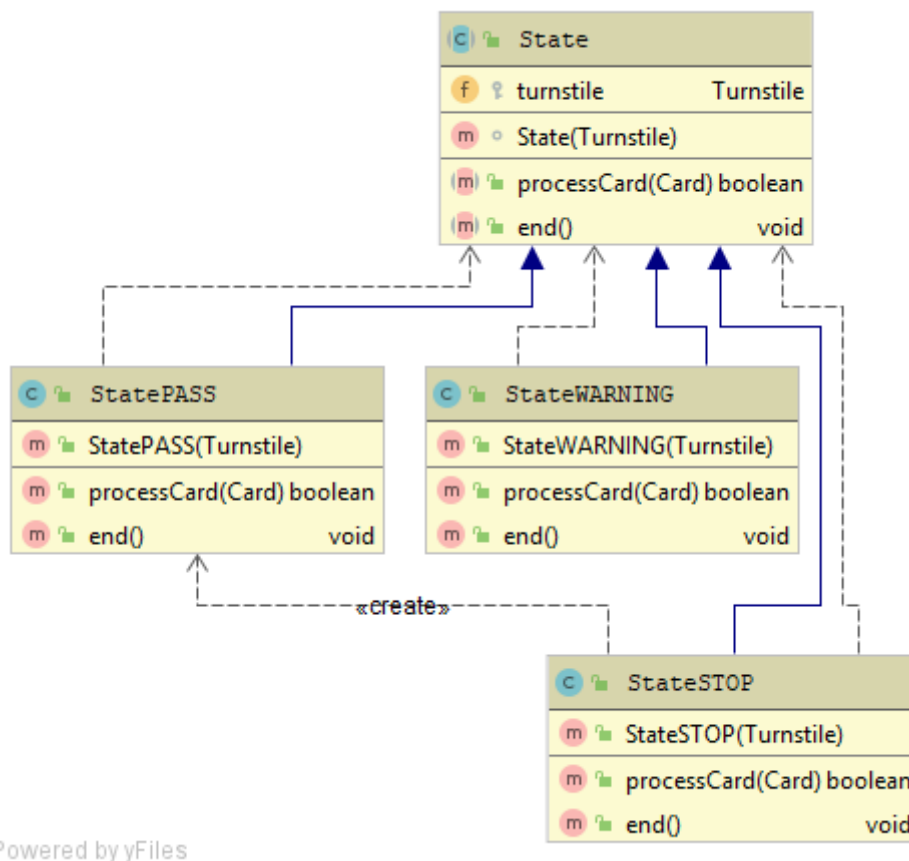
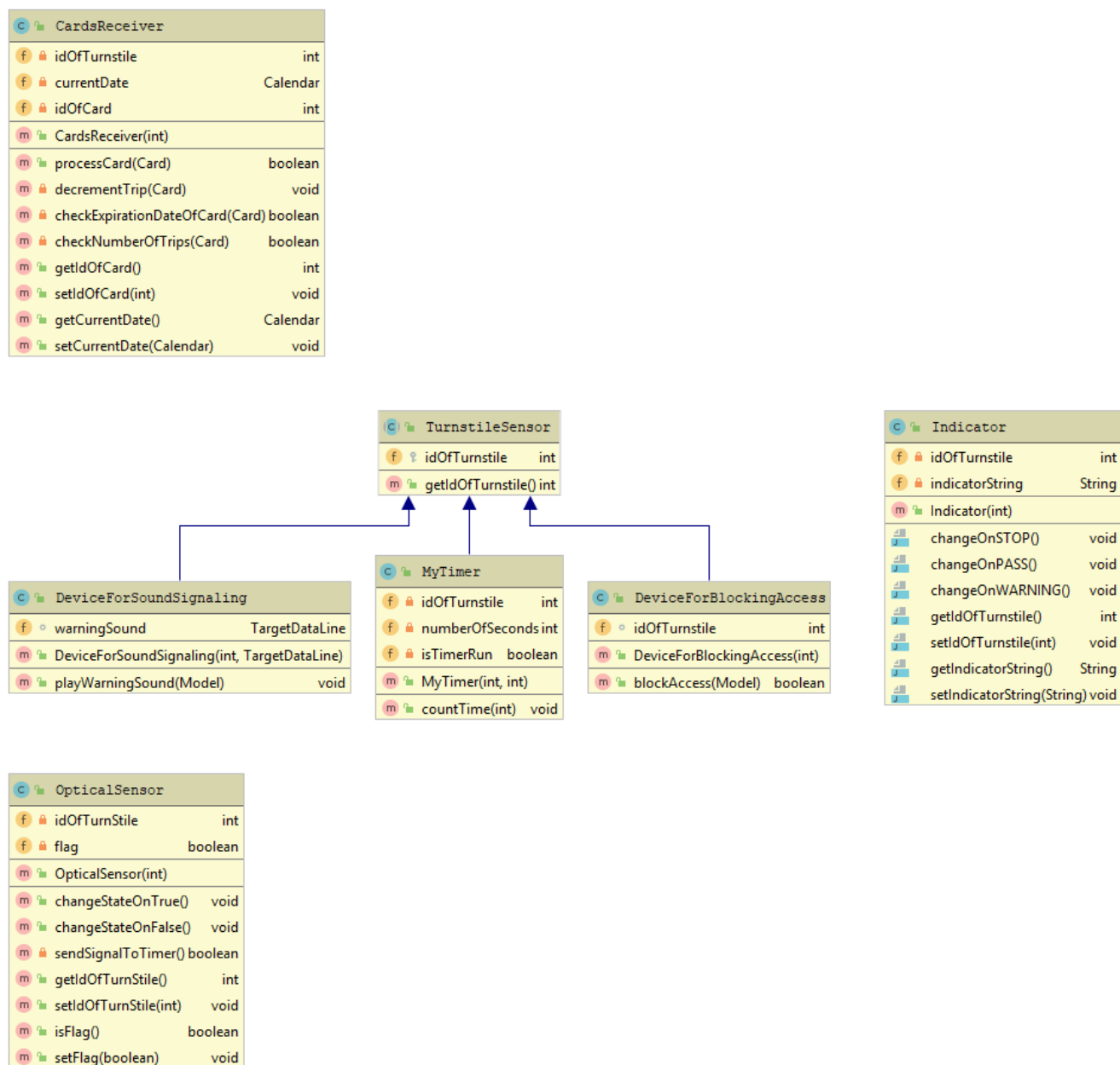


Рисунок 4.11 - UML діаграма класів пакету state



Powered by yfiles

Рисунок 4.12 - UML діаграма класів пакету turnstileSensors

4.2 Опис класів

4.2.1 Класи пакету com.demo

DemoApplication – головний клас програми, що містить метод main.

ControllerSpring – контроллер програми.

4.2.2 Класи пакету commands

Command – абстрактний клас, що описує команду.

CommandClickDb – клас, що описує команду натискання кнопки «DB».

CommandClickFirstButton – клас, що описує команду натискання кнопки «Attach a valid card»

CommandClickFirstSensor – клас, що описує команду натискання кнопки «Sensor1»

CommandClickLogin – клас, що описує команду натискання кнопки «Login»

CommandClickLogout – клас, що описує команду натискання кнопки «Logout»

CommandClickSecondButton – клас, що описує команду натискання кнопки «Attach an invalid card (expiration time)»

CommandClickSecondSensor – клас, що описує команду натискання кнопки «Sensor2»

CommandClickThirdButton – клас, що описує команду натискання кнопки «Attach an invalid card (number of passes)»

CommandClickThirdSensor – клас, що описує команду натискання кнопки «Sensor3»

CommandDefaultPage – клас, що описує команду сторінки «за замовчуванням»

4.2.3 Класи пакету converter

Converter – клас, що описує конвертер зі Timestamp до LocalDateTime та навпаки

4.2.4 Класи пакету factories

CommandsFactory – клас, що описує фабрику команд

DAOFactory – клас, що описує фабрику DAO

4.2.5 Класи пакету card

Card – клас, що описує картку метро

4.2.6 Класи пакету DAO

AbstractDAO – абстрактний клас, що описує DAO

AdminsDAO – клас, що описує DAO до таблиці «ADMINS»

ColorMetroLineDAO – клас, що описує DAO до таблиці «COLOR_METRO_LINE»

MetroStationsDAO – клас, що описує DAO до таблиці «METRO_STATIONS»

PassesThroughTurnstilesDAO – клас, що описує DAO до таблиці «PASSES»

PositionDAO – клас, що описує DAO до таблиці «POSITION»

ServiceStaffDAO – клас, що описує DAO до таблиці «SERVICE STAFF»

TurnstilesDAO – клас, що описує DAO до таблиці «TURNSTILES»

4.2.7 Класи пакету db

Database – клас, що описує з'єднання з базою даних

4.2.8 Класи пакету dbEntity

Admin – клас, що описує сутність «адміністратор»

MetroLine – клас, що описує сутність «гілка метро»

MetroStation – клас, що описує сутність «станція метро»

Passes – клас, що описує сутність «проходи через турнікет»

Position – клас, що описує сутність «посада працівника»

ServiceStaff – клас, що описує сутність «обслуговуючий персонал»

Turnstile – клас, що описує сутність «турнікет»

4.2.9 Класи пакету state

State – абстрактний клас, що описує стан турнікету

StatePASS – клас, що описує стан турнікету «PASS»

StateSTOP – клас, що описує стан турнікету «STOP»

StateWARNING – клас, що описує стан турнікету «WARNING»

4.2.10 Класи пакету **turnstileSensors**

CardsReceiver – клас, що описує сенсор турнікету (приймач карток)

DeviceForBlockingAccess – клас, що описує сенсор турнікету (пристрій для блокування доступу)

DeviceForSoundSignaling – клас, що описує сенсор турнікету (пристрій для звукового сигналу)

Indicator – клас, що описує сенсор турнікету (індикатор)

MyTimer – клас, що описує сенсор турнікету (таймер)

OpticalSensor – клас, що описує сенсор турнікету (оптичний сенсор)

TurnstileSensor – клас, що описує сенсор турнікету (турнікет)

4.2.11 Класи пакету **model**

Station – клас, що описує станцію метрополітену

Turnstile – клас, що описує турнікет

4.3 Діаграма архітектури застосунку

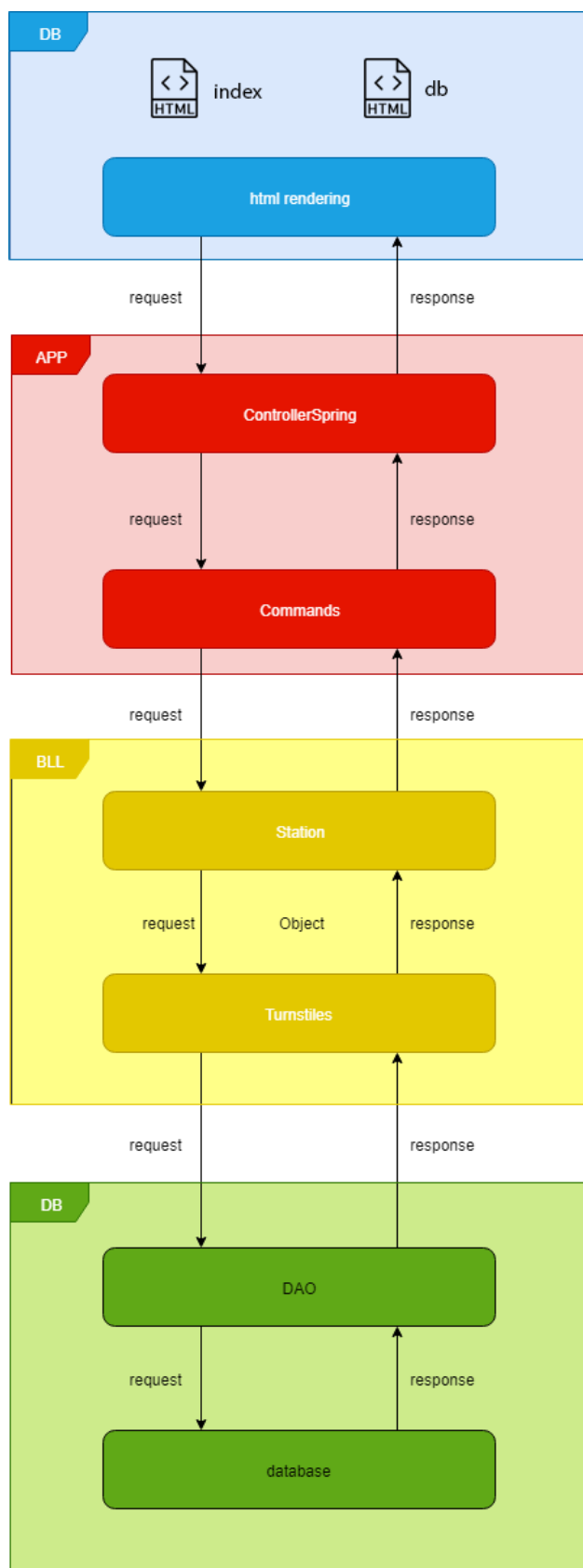


Рисунок 4.13 – діаграма архітектури застосунку

4.4 Use-case діаграма

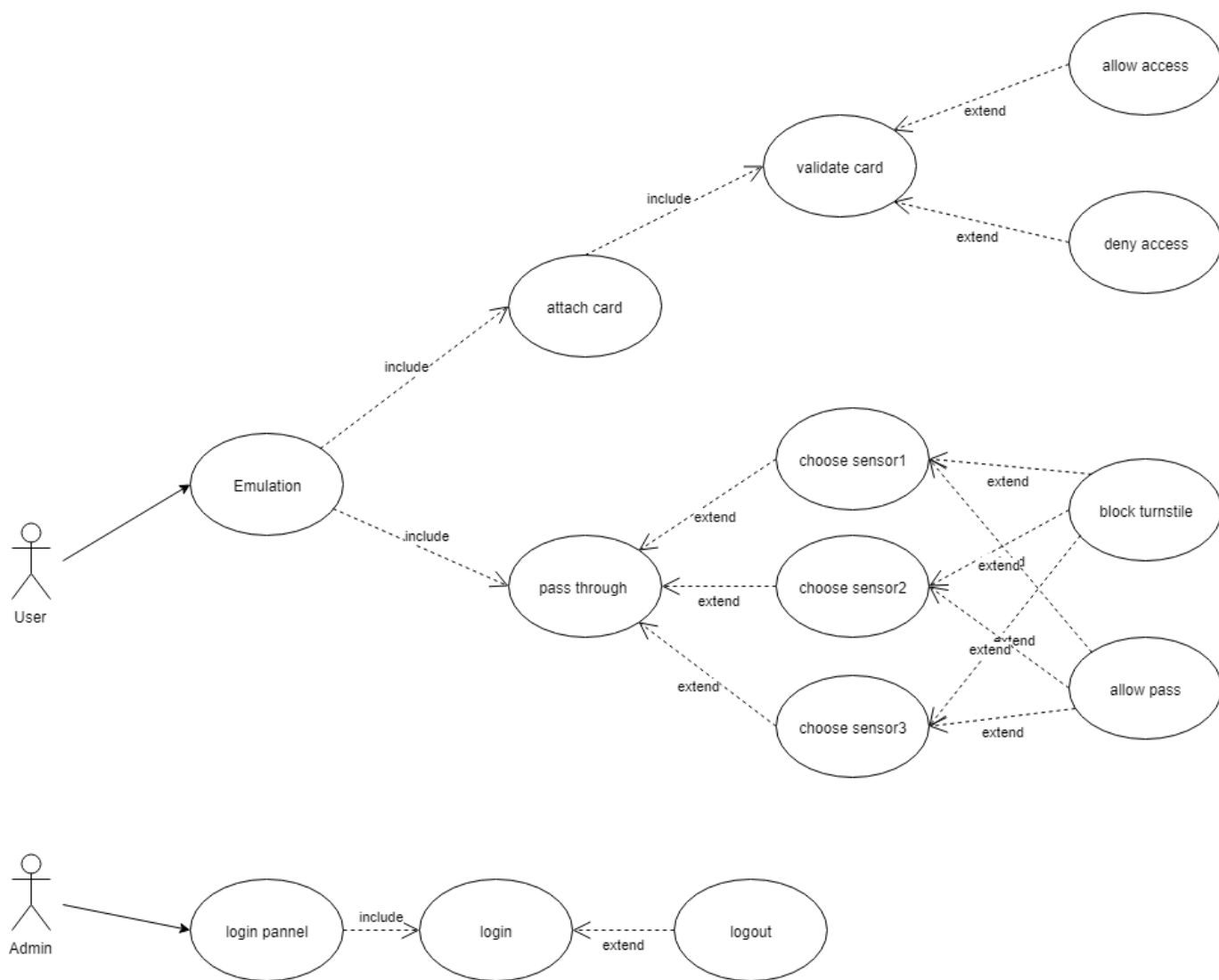


Рисунок 4.14 – Use-case діаграма

4.5 Опис бази даних

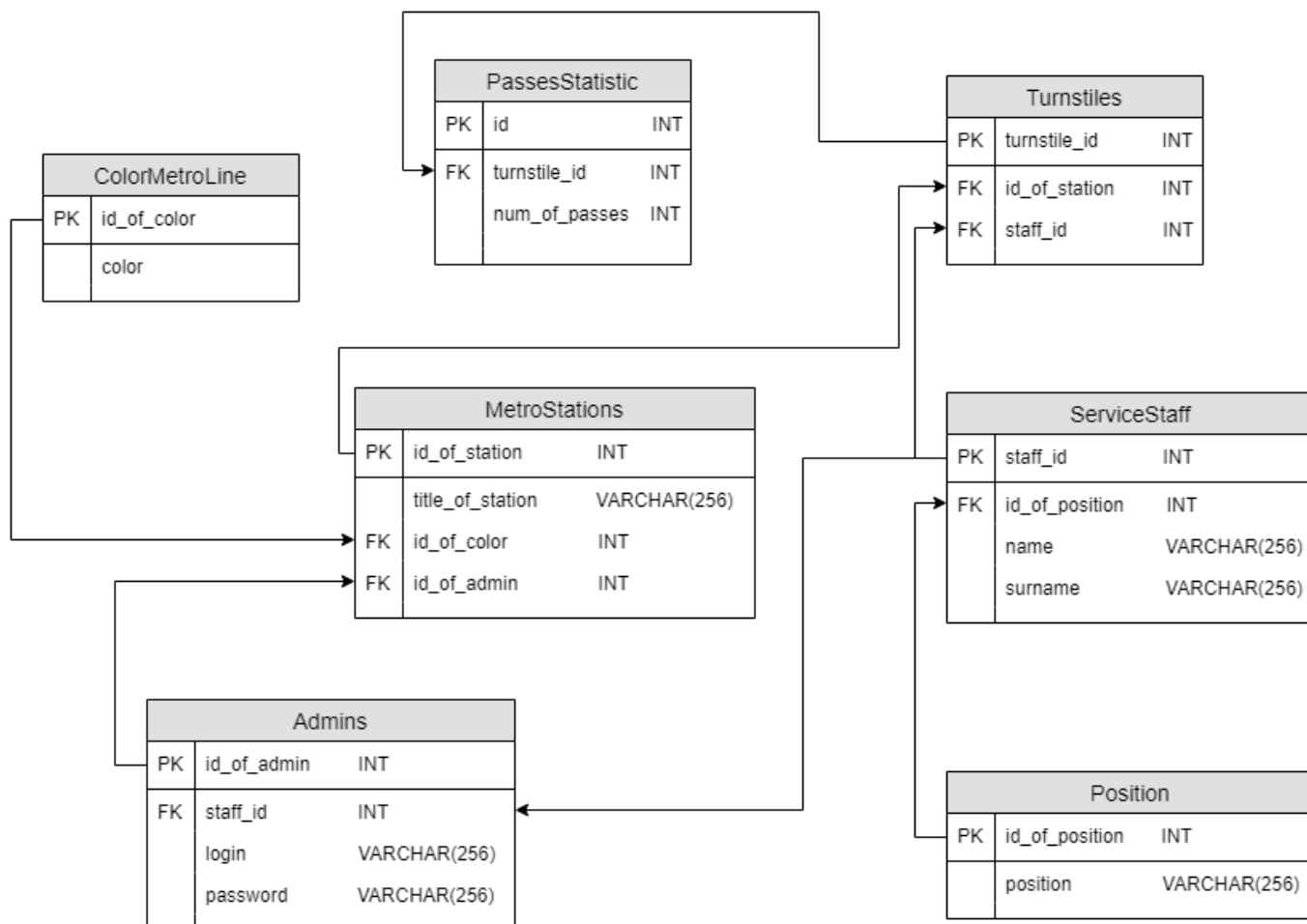


Рисунок 4.15 – схема поточної бази даних

4.5.1 Таблиця PASSES_STATISTIC

Містить інформацію про проходи через кожний турнікет

4.5.2 Таблиця TURNSTILES

Містить інформацію про кожен турнікет

4.5.3 Таблиця METRO_STATIONS

Містить інформацію про кожну станцію метро

4.5.4 Таблиця SERVICE_STAFF

Містить інформацію про обслуговуючий персонал станції метрополітену

4.5.5 Таблиця POSITION

Містить інформацію про посади персоналу

4.5.6 Таблиця ADMINS

Містить таблицю адміністраторів автоматизованої системи управління метрополітену

5 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 План тестування

Тестування програми відбувається після її розробки для виявлення помилок в логіці програми та її реалізації. Наведений нижче план тестування перевіряє усі “слабкі місця” програми.

а) тестування емуляції проходу через турнікет

- 1) прийняття дійсної карти приймачем карток
- 2) прийняття недійсної карти приймачем карток (прострочена картка)
- 3) прийняття недійсної карти приймачем карток (кількість поїздок)
- 4) робота трьох сенсорів руху
- 5) робота пристрою для перекривання доступу
- 6) робота пристрою для подачі звукового сигналу
- 7) робота логіна в автоматизовану систему керування метрополітену
- 8) робота логаута з автоматизованої системи керування метрополітену

б) Тестування збереження результатів до бази даних

- 1) вставка в таблицю нової інформації про проходи через турнікет
- 2) збереження інформації про проходи через турнікет

5.2 Приклади тестування

Тестування програмного забезпечення було проведено згідно з планом, наведеним у підрозділі 5.1. Були отримані наступні результати.

а) тестування емуляції проходу через турнікет

- 1) Приймач карток приймає дійсну картку, на якій є необхідна кількість поїздок
- 2) Приймач не приймає недійсну картку, виводиться повідомлення про помилку читання з карти
- 3) Приймач не приймає дійсну картку без необхідної кількості поїздок, виводиться повідомлення про помилку читання з карти

4) Якщо горить індикатор «STOP» сенсори засікають людину та передають сигнал до датчику блокування доступу, щоб він заблокував прохід, якщо ж горить індикатор «PASS», сенсор передає сигнал таймеру для відліку 5 секунд на проходження людини

5) Якщо сенсор передає сигнал про те, що треба заблокувати прохід, то пристрій для блокування проходу виводить повідомлення про помилку та передає сигнал пристрою для звукового сигналу

6) Якщо пристрій для звукового сигналу отримує сигнал, то він відтворює звук про помилку

7) Якщо пароль та логін не є правильними, то програма видає повідомлення про помилку, якщо логін та пароль є коректними, то програма виконує вход до автоматизованої системи керування метрополітеном

8) Якщо адміністратор натискає на кнопку «Logout», то він коректно покидає особистий кабінет

б) Тестування збереження результатів до бази даних

1) Якщо буде натиснута кнопка «DB», то інформацію про прохід буде коректно додано до записів бази даних

2) Якщо виконати логін в автоматизовану систему керування метрополітеном, то ми побачимо запис, який тільки що зробили

6 ІНСТРУКЦІЯ КОРИСТУВАЧА

6.1 Призначення програми

Дане програмне забезпечення є емулятором турнікету метрополітену, його основним призначенням є обмеження і реєстрація входу-виходу об'єктів (людей) на заданій території (станція метрополітену) через «точки проходу» (турнікети).

Призначенням с точки зору програміста, котрий написав це ПЗ, є здобуття навичок у використанні Spring Framework з мовою програмування Java та застосуванням різних шаблонів проектування.

6.2 Вимоги до системи

- Intel® Core 2 або AMD Athlon® 64 процесор; частота процесора 1.4 GHz або швидше;
- Microsoft Windows XP чи більш пізня версія;
- 1 GB або більше RAM (2 GB рекомендовано);
- 32 GB або більше вільного місця на диску;
- 640 x 480 дисплей (1280x800 рекомендовано);

6.3 Інструкція по роботі з програмою

Інтерфейс програми наведений на рисунках 6.1 та 6.2

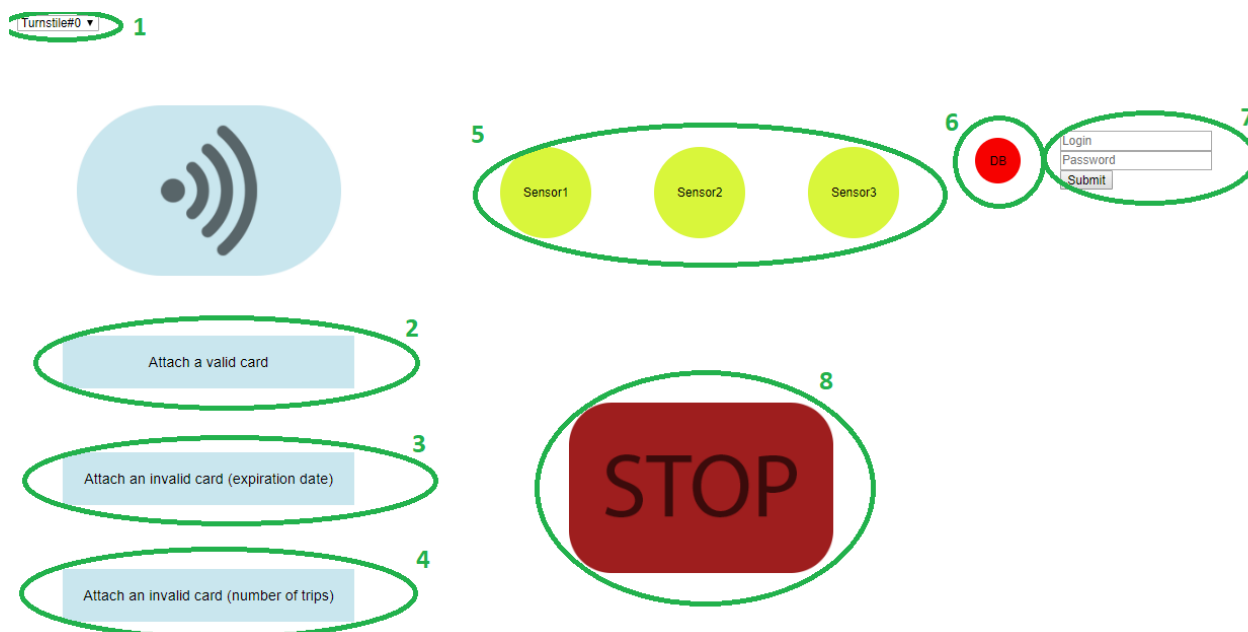


Рисунок 6.1 – Інтерфейс емулятору турнікету

1 – поле для вибору номеру турнікету

2 – кнопка для надання дійсної картки у приймач карток

3 – кнопка для надання недійсної картки у приймач карток

4 – кнопка для надання дійсної картки у приймач карток, але з недостатньої кількістю поїздки на ній

5 – оптичні датчики руху турнікету

6 – індикатор турнікету

7 – поле для введення логіну та паролю та кнопка «Submit»

8 – кнопка для додавання запису про проходи через турнікети до бази даних

Pass statistics

id	turnstile	passes	date
0	0	12	2019-05-30T17:21:00.976
1	1	8	2019-05-30T17:21:00.976
2	2	11	2019-05-30T17:21:00.976
3	3	21	2019-05-30T17:21:00.976
4	4	45	2019-05-30T17:21:00.976
5	0	1	2019-06-02T20:47:44.881
6	2	1	2019-06-02T21:02:27.802
7	0	1	2019-06-02T21:05:31.206
8	4	1	2019-06-02T21:08:50.634

Logout

Рисунок 6.2 – Інтерфейс особистого кабінету в автоматизованій системі керування метрополітену

1 – таблиця бази даних, що дає змогу отримати інформацію про проходи через турнікети

2 – кнопка для виходу с автоматизованої системи керування метрополітену

ВИСНОВКИ

На етапі підготовки до написання курсової роботи були детально розглянуті та вивчені основні концепції роботи Spring Framework.

При проектуванні програмного забезпечення, було вирішено, які компоненти турнікету кожен клас буде реалізовувати та якими способами буде проходити взаємодії між класами. Була створена ієрархія з 46 класів, яка дуже полегшує внутрішню логіку роботи програми.

Наступним етапом, після програмної реалізації системи керування турнікету метрополітену, була перевірка працездатності програми при різних ситуаціях, тестування програмного забезпечення на наявність у ній помилок, що можуть призвести до аварійного завершення роботи програми, чи до помилок у логіці роботи програмного забезпечення. Після проведення тестування помилки та недоліки були усунені.

На даному етапі розвитку функціонал програми (система керування турнікету метрополітену) логічно завершений. Вихідна програма детально прокоментована, тому дозволяє змінити або доповнити частину його функціоналу людині, котра володіє знаннями роботи зі Spring Framework, та має середні навички роботи з мовою програмування Java.

ПЕРЕЛІК ПОСИЛАНЬ

1. Блинов І.Н., Романчик В.С. Java Методи програмування, — М.: «Чотири чверті», 2013.
2. Марк Гранд Шаблони проектування в Java — М.: «Нове знання», 2004.
3. Bryan Bashman, Kathy Sierra Servlets and JSP — L.: «Kelmscott Press», 2008.
4. Clarence Ho, Chris Schaefer, Rob Harrop, Iuliana Cosmina Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools, 5th Edition — USA.: «Apress», 2017.

ДОДАТОК А

*Тексти програмного коду програмного забезпечення
знаходяться за наступним посиланням:*

<https://github.com/EddyShvets/courseWorkJava>

(Найменування програми (документа))

(Вид носія даних)

51 арк

(Обсяг програми (документа), арк., Кб)

студента групи ІІІ-61 ІІІ курсу

Швеця Е.Я.