

Міністерство освіти та науки України
Харківський національний університет радіоелектроніки
Кафедра програмної інженерії

Лабораторна робота № 4
З дисципліни: «Архітектура програмного забезпечення»
на тему: «Програмна система для автоматизації ведення флористичного
бізнесу»

Виконала
ст. гр. ПЗПІ-18-5
Борщова Олександра Вадимівна

Перевірив
ст. викл. каф. ПІ
Сокорчук Ігор Петрович

Харків 2021

Мета: розробити мобільну частину для програмної системи для автоматизації ведення флористичного бізнесу.

Хід роботи:

В якості мобільної платформи обрано ОС Android. Для розробки мобільної частини застосунку використано мову програмування Java 8. Для взаємодії з серверною частиною системи використовується HTTPS протокол та JSON формат транспортування даних. Для реалізації цієї взаємодії використовується бібліотека retrofit 2.

Перед тим, як програмно реалізувати мобільну частину, було проаналізовано предметну область та встановлено всі основні способи використання клієнтського застосунку. Створено UseCase діаграму, що описує сценарій поведінки застосунку у процесі взаємодії з його користувачами. Діаграма варіантів використання наведена на рисунку 1.

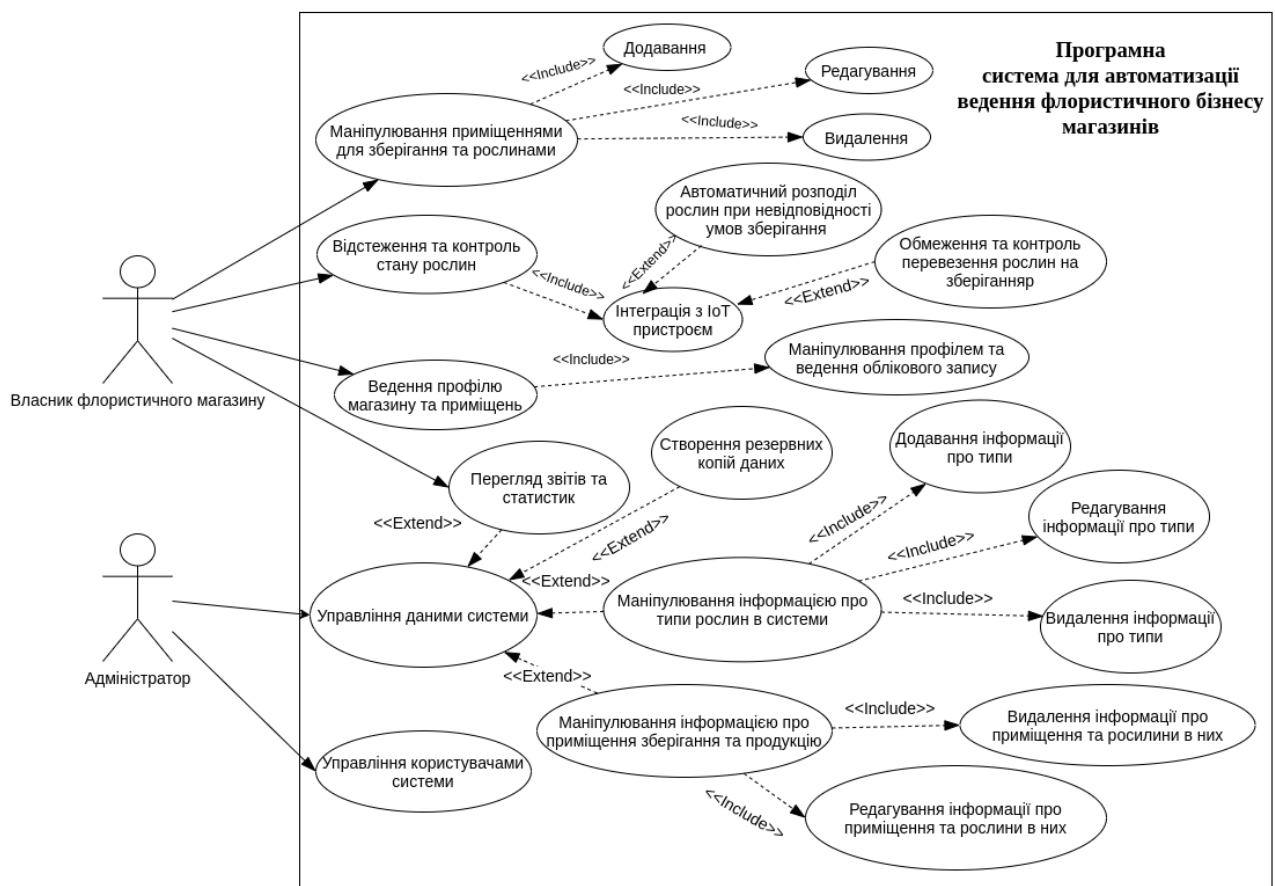


Рисунок 1 – Діаграма варіантів використання для програмної системи «FloristikUp»

Mobile частина програмної системи має одного актора: людину, що є власником або представником флористичного магазину.

Власник флористичного магазину має змогу редагувати свій профіль, маніпулювати приміщеннями, де зберігаються квіти та їх смарт-пристроями, типами квітів, створювати нові зберігання квітів, редагувати їх, переглядати їх, виконувати запит на автоматичний перерозподіл квітів, для яких мікроклімат є незадовільним, до інших приміщень зберігання, відстежувати стан квітів та мікроклімат приміщень зберігання.

Для відображення робочих компонентів мобільної частини системи та відображення логіки їх взаємодії та інженерних рішень під час проектування було створено діаграму компонентів мобільного застосунку програмної системи, яка наведена у додатку А.

З іншого боку було побудовано діаграму, що визначає зміну станів об'єкту у часі – діаграму станів, зображену на рисунку 2. Діаграму побудовано з точки зору системи під час використання її власником приміщень.

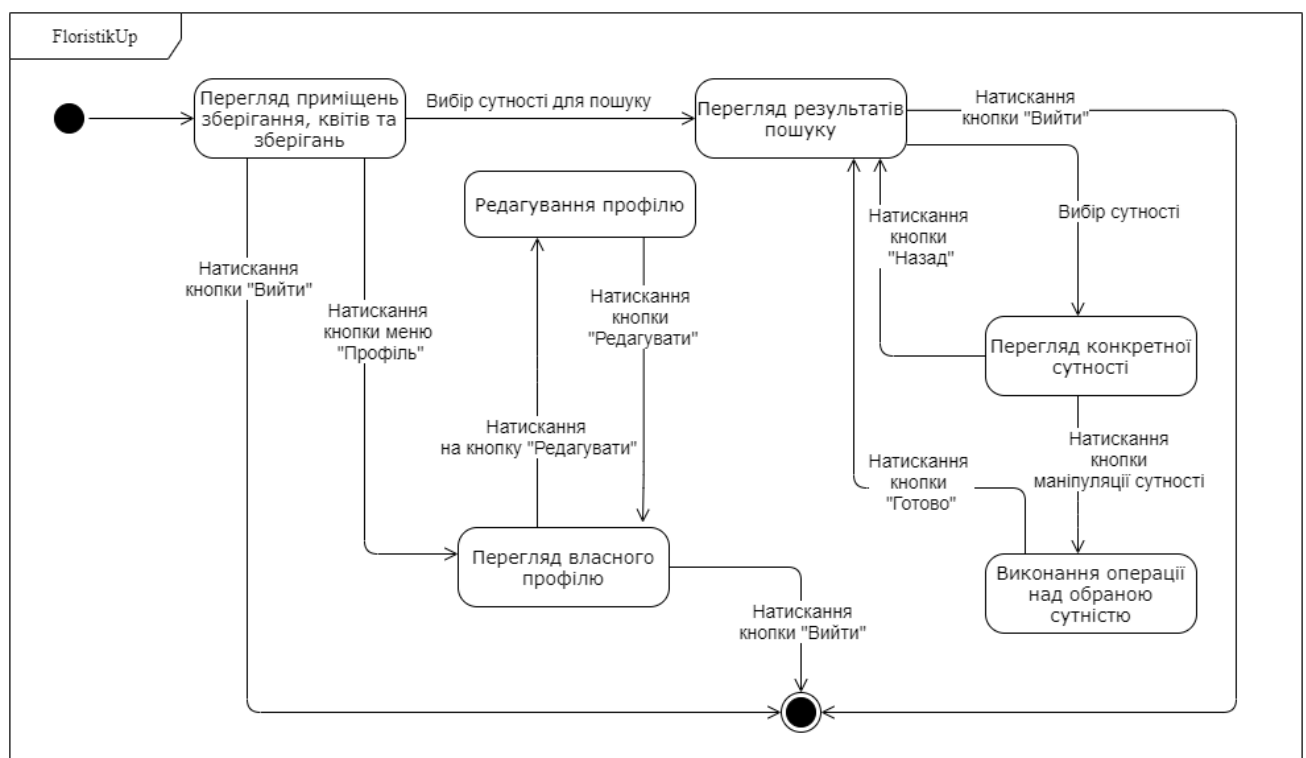


Рисунок 2 – Діаграма станів та переходів для мобільного застосунку програмної системи «FloristikUp»

Для більш детального опису умов переходів системи з одного стану в інший також побудовано діаграму діяльності, яка наведена у додатку Б.

У додатку В наведено частину програмної реалізації обробки результату автоматичного перерозподілу квітів при незадовільних мікрокліматичних умовах.

У додатку Г наведено програмну реалізації ініціалізації адаптеру відображення інформації про зберігання рослин у приміщеннях.

Посилання на архів з програмним кодом та файлом контрольної суми:

<https://drive.google.com/drive/folders/1Gp62pea3PAaDbTnxkLmB7Fa0CDZHg-U0?usp=sharing>

Контрольна сума до архіву: 0f73be475033da0836f1238f221ce0ad

Висновок: під час виконання лабораторної роботи було розроблено mobile частину для програмної системи автоматизації ведення флористичного бізнесу магазинів.

ДОДАТОК А

Діаграма компонентів мобільного застосунку

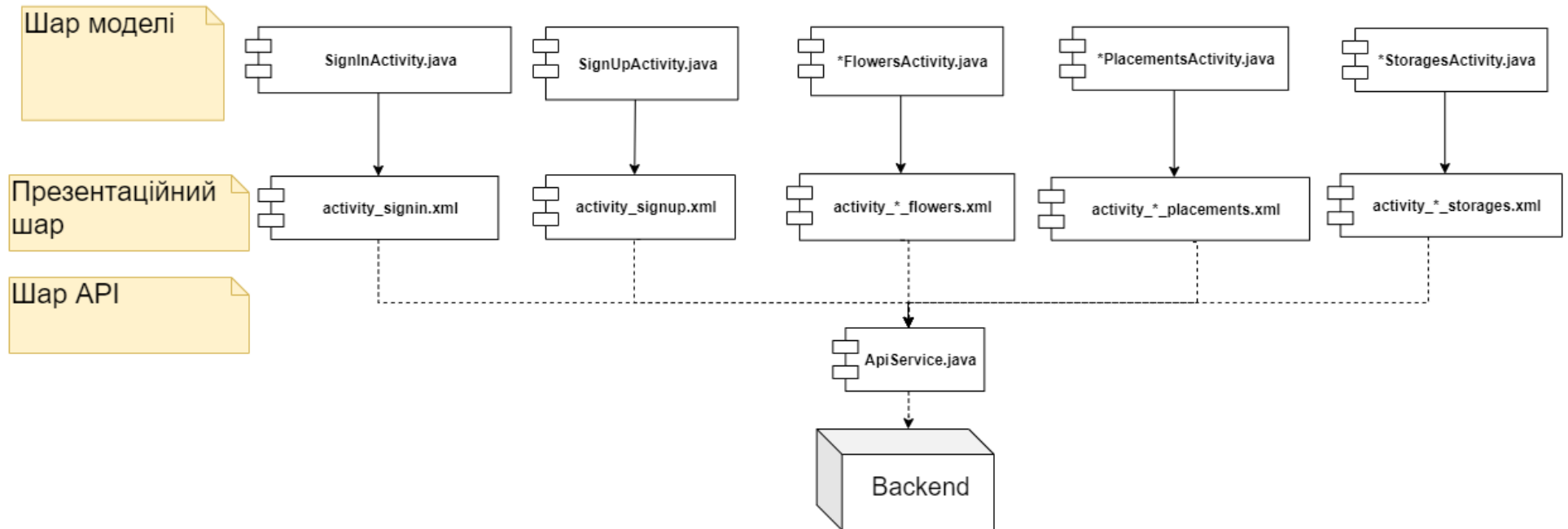


Рисунок Б.1 – Діаграма компонентів мобільного програмного застосунку системи «FloristikUp»

ДОДАТОК Б

Діаграма діяльності для мобільного застосунку

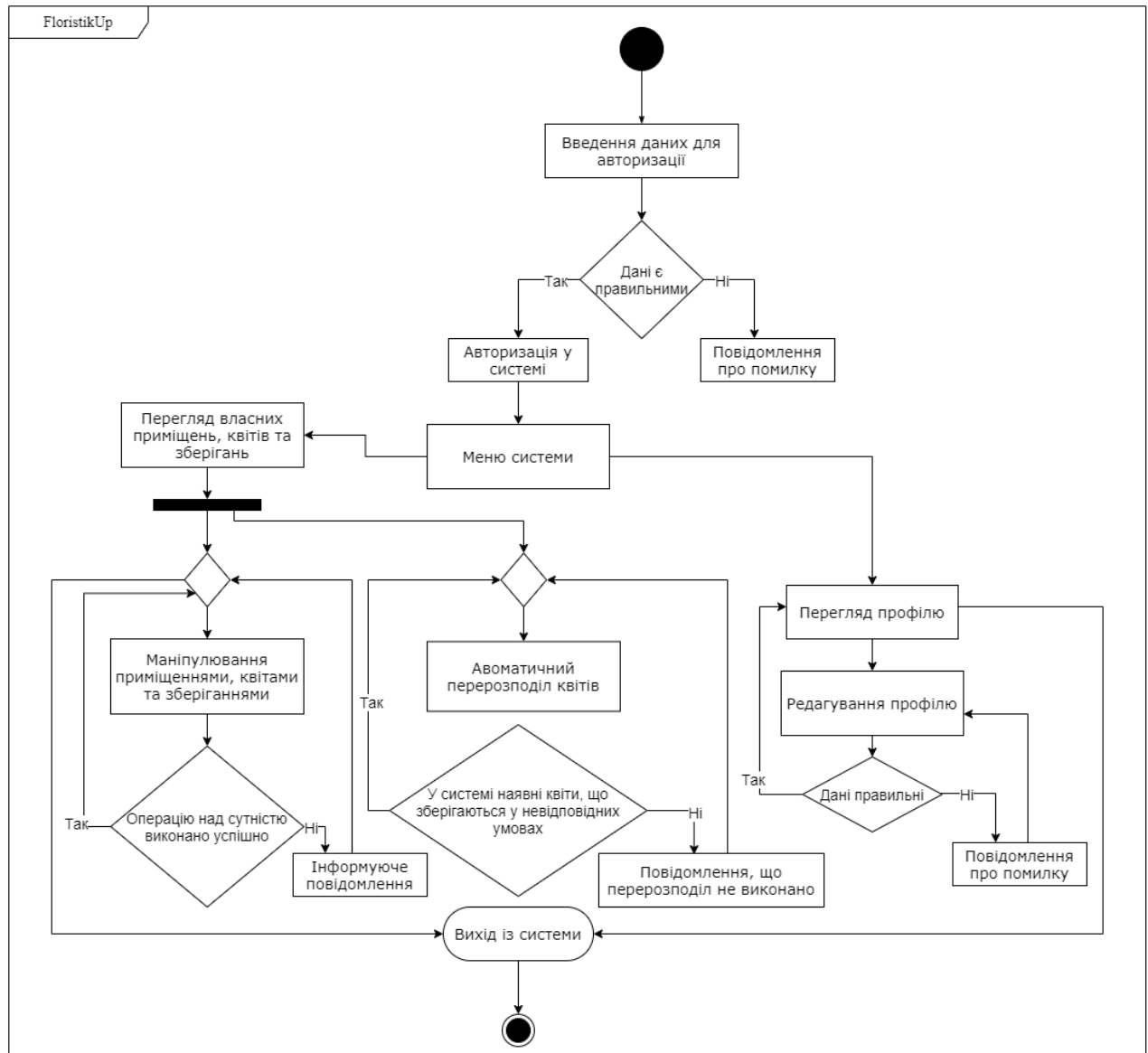


Рисунок В.1 – Діаграма діяльності для мобільного програмного застосунку системи «FloristikUp»

ДОДАТОК В

Частина програмної реалізації обробки результату перерозподілу квітів

```
1  Callback<ArrayList<RedistributionResponseDto>> redistributionCallback =
2      new Callback<ArrayList<RedistributionResponseDto>>() {
3          @SuppressWarnings("DefaultLocale")
4          @Override
5          public void onResponse(
6              Call<ArrayList<RedistributionResponseDto>> call,
7              Response<ArrayList<RedistributionResponseDto>> response
8          ) {
9              ArrayList<RedistributionResponseDto> jsonArray = response.body();
10
11              StringBuilder resultMessage = new StringBuilder();
12              Flower flower;
13              Placement placement;
14              String redistributionInfo =
15                  "Квітку %s (%s) у кількості %d"
16                  + " перерозподілено до приміщення %d (%s, %s %s).\n";
17
18              for (RedistributionResponseDto storage : jsonArray) {
19                  flower = storage.getFlower();
20                  placement = storage.getStorageRoom();
21                  resultMessage.append(String.format(
22                      redistributionInfo,
23                      flower.getName(), flower.getColor(), storage.getAmount(),
24                      placement.getId(), placement.getCity(),
25                      placement.getStreet(), placement.getHouse())
26                  );
27              }
28
29              String result = resultMessage.toString();
30
31              loadingDialog.dismiss();
32              new AlertDialog.Builder(StoragesActivity.this)
33                  .setTitle(getString(R.string.redistribute))
34                  .setMessage(resultMessage)
35                  .setIcon(android.R.drawable.ic_dialog_alert)
36                  .setPositiveButton(android.R.string.yes,
37                      (dialog, whichButton) -> { startActivity(
38                          new Intent(
39                              StoragesActivity.this,
40                              PlacementsActivity.class)
41                      );
42                  }).show();
43          }
44
45          @Override
46          public void onFailure(Call<ArrayList<RedistributionResponseDto>> call,
47              Throwable t) {
48              System.out.println(t);
49              Log.i(TAG, t.getMessage());
50              loadingDialog.dismiss();
51          }
52      };
```

ДОДАТОК Г

Програмна реалізація ініціалізації адаптеру відображення зберігань квітів

```
1 private void initializeAdapter() {
2     StoragesRVA adapter = new StoragesRVA(this, mFlowerStorages);
3     mRecyclerView.setAdapter(adapter);
4     loadingDialog.dismiss();
5
6     if (!mFlowerStorages.isEmpty()) {
7         TextView header = findViewById(R.id.name_tv);
8
9         FlowerStorage storage = mFlowerStorages.get(0);
10
11         Long storageRoomId = storage.getStorageRoomId();
12         String city = storage.getCity();
13         String street = storage.getStreet();
14         String house = storage.getHouse();
15         Integer actualCapacity = storage.getActualCapacity();
16         Long maxCapacity = storage.getMaxCapacity();
17         String temperature = Double.toString(storage.getTemperature());
18         String humidity = Double.toString(storage.getHumidity());
19
20         header.setTextSize(TypedValue.COMPLEX_UNIT_SP, 17);
21         header.setText(String.format(
22             "| %s %d \n| %s, %s %s \n| %s %d/%d \n| %s %s °C / %s %s %%",
23             getString(R.string.placement_number), storageRoomId,
24             city, street, house,
25             getString(R.string.capacity), actualCapacity, maxCapacity,
26             getString(R.string.temperature), temperature,
27             getString(R.string.humidity), humidity)
28         );
29     }
30 }
```