Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний  інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації

і управління

**Звіт**

з лабораторної роботи № 3 з дисципліни

«Компоненти інженерії програмного забезпечення»

«Дослідження процесу розробки програмного забезпечення. Архітектурнета детальне проектування.**»**

Виконали студенти групи ІП-01 Берлінський Ярослав, Князєв Ілля, Гагарін Артем

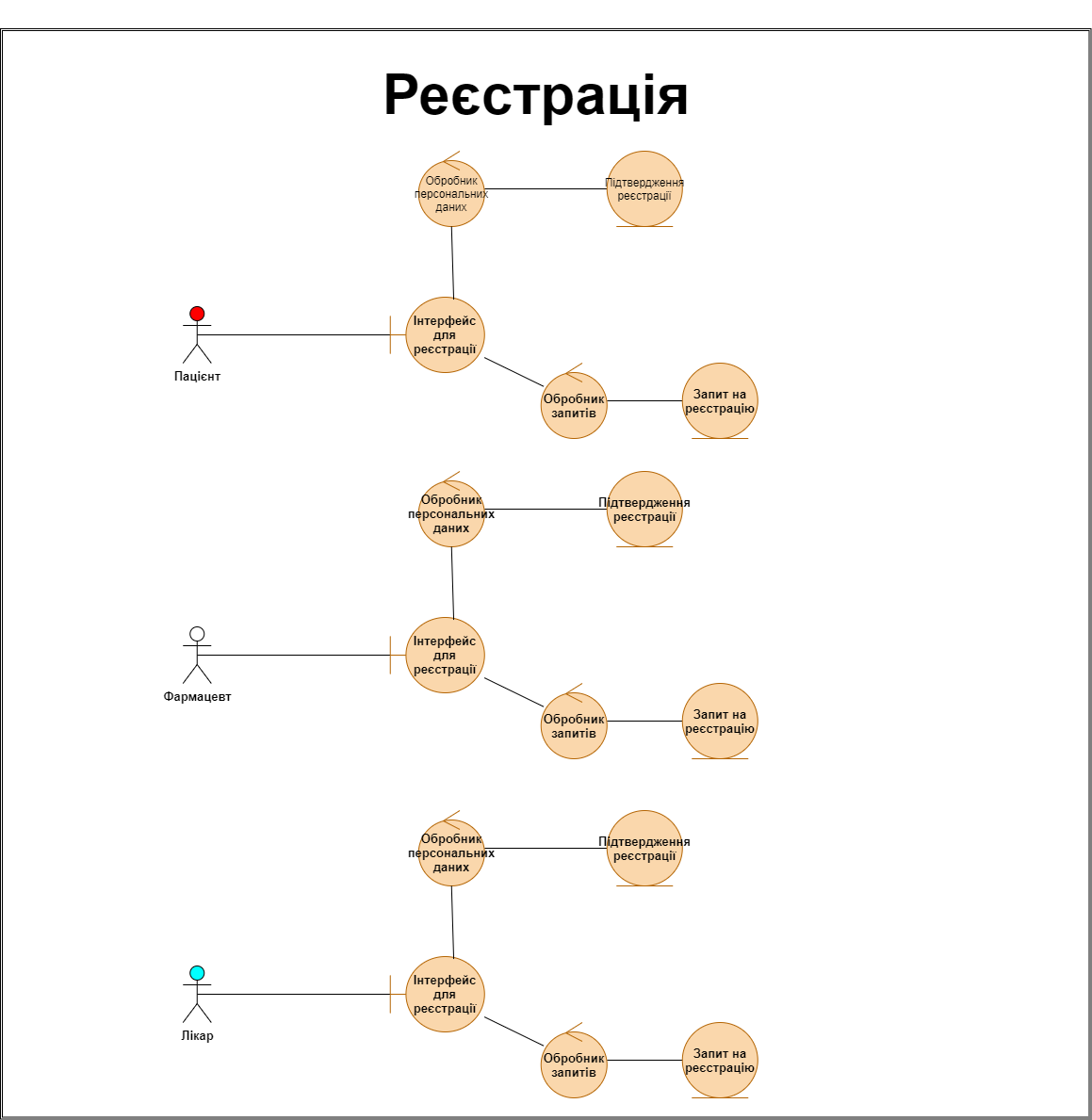
Перевірила Вітковська Ірина Іванівна

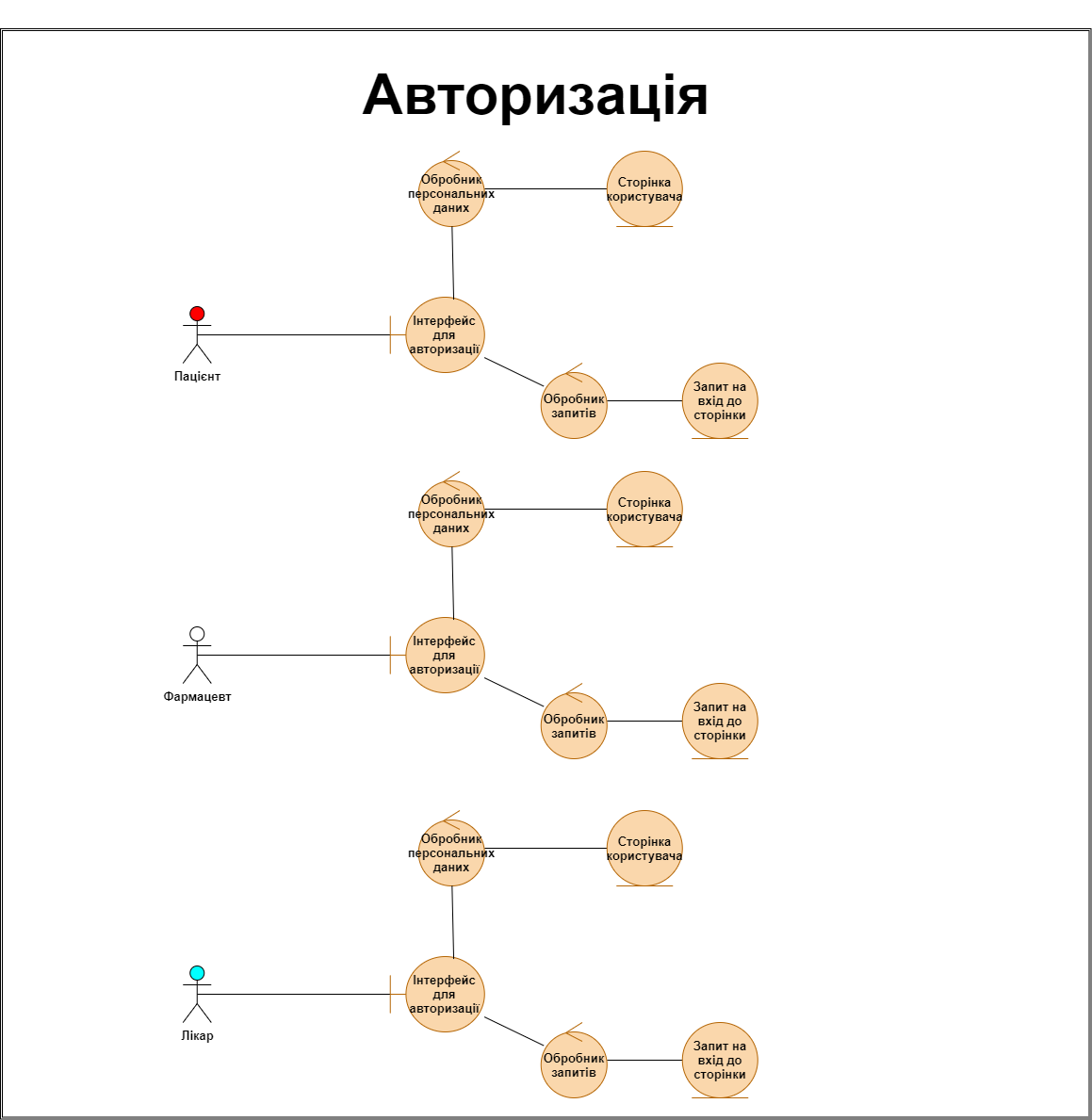
Київ 2021

**Лабораторна робота 3  
Дослідження процесу розробки програмного забезпечення. Архітектурне  
та детальне проектування.**

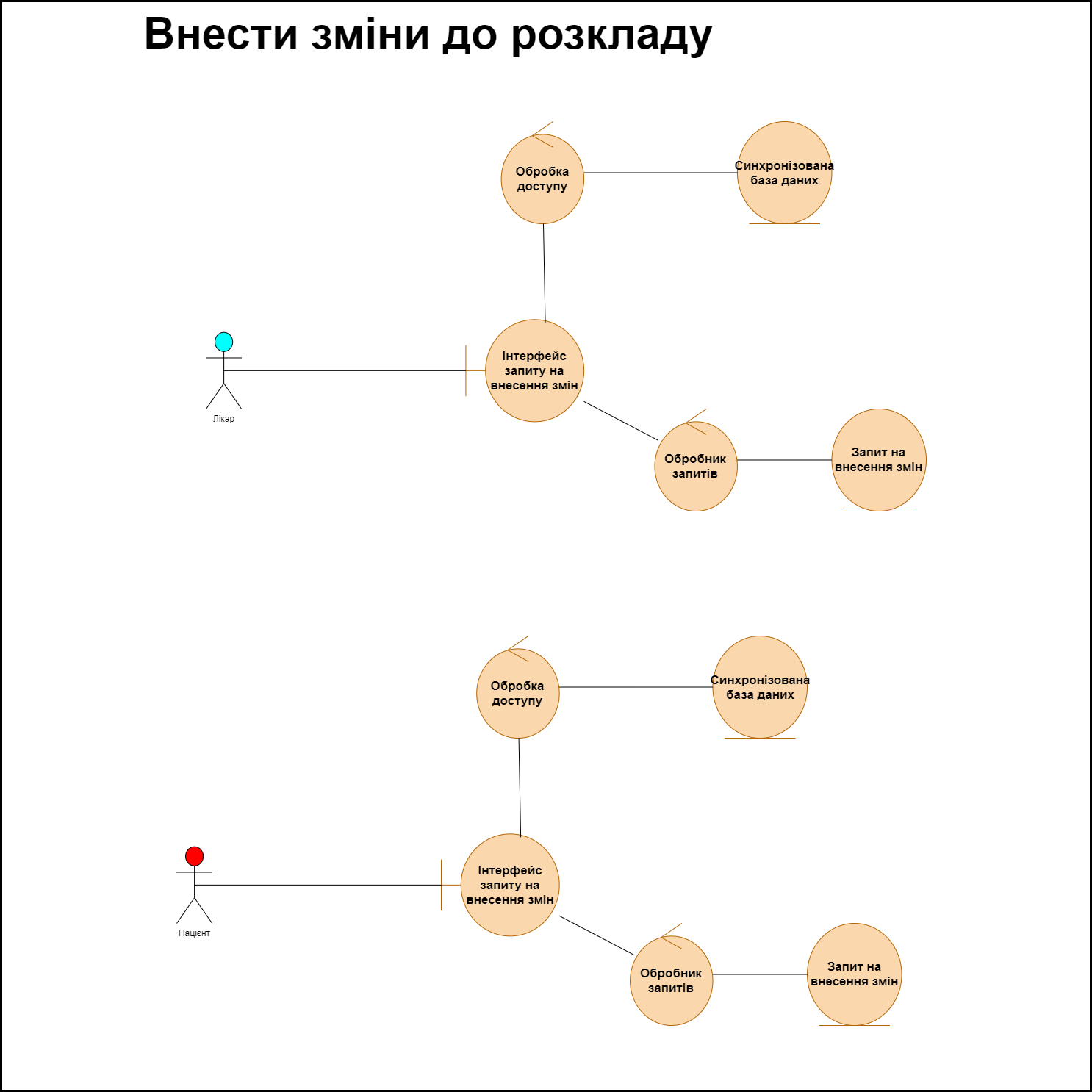
Мета – отримати навички та єдині тактичні прийоми, якими повинні користуватися  
різні елементи системи.

**Діаграми класів аналізу варіанту використання**

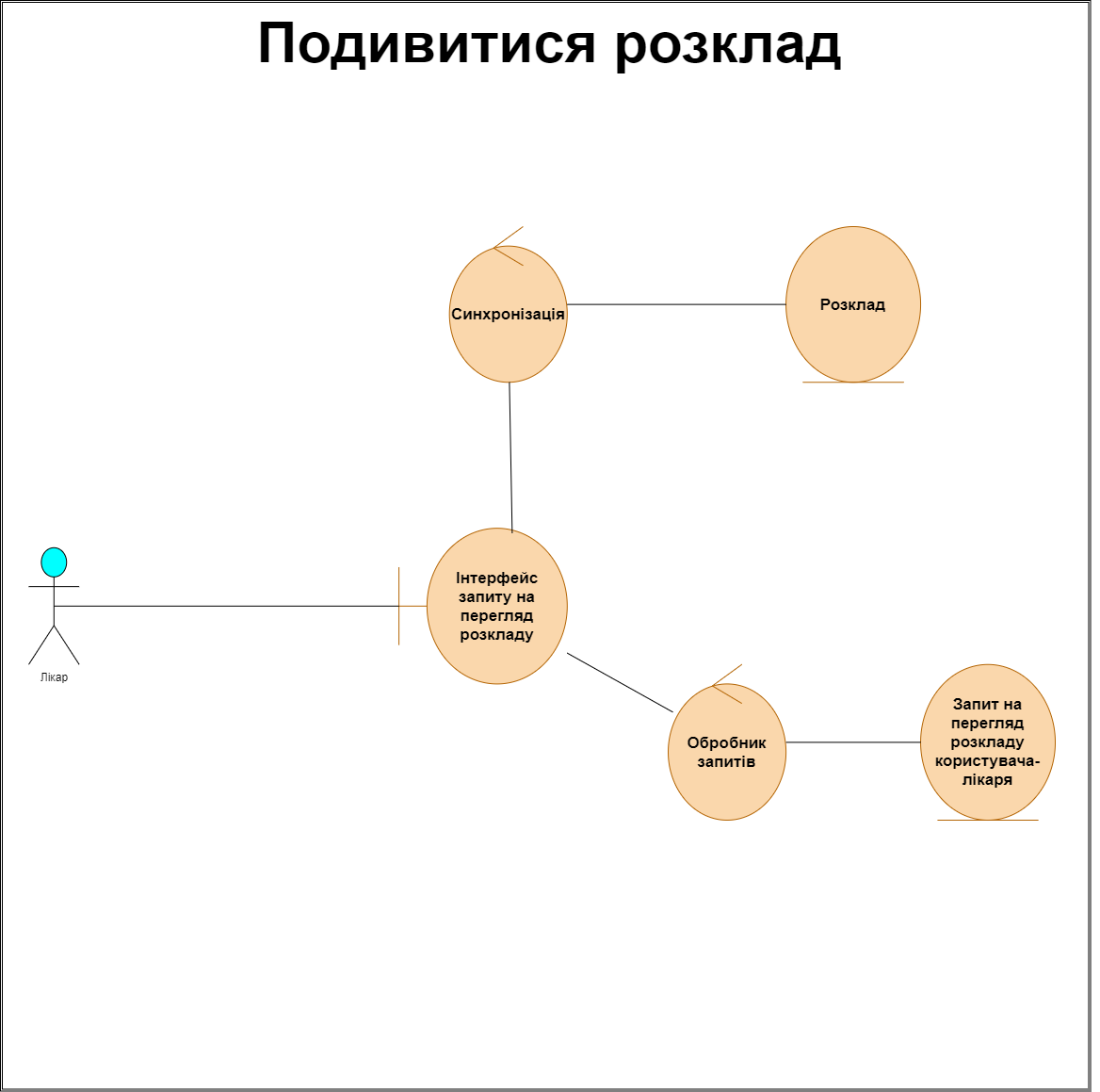


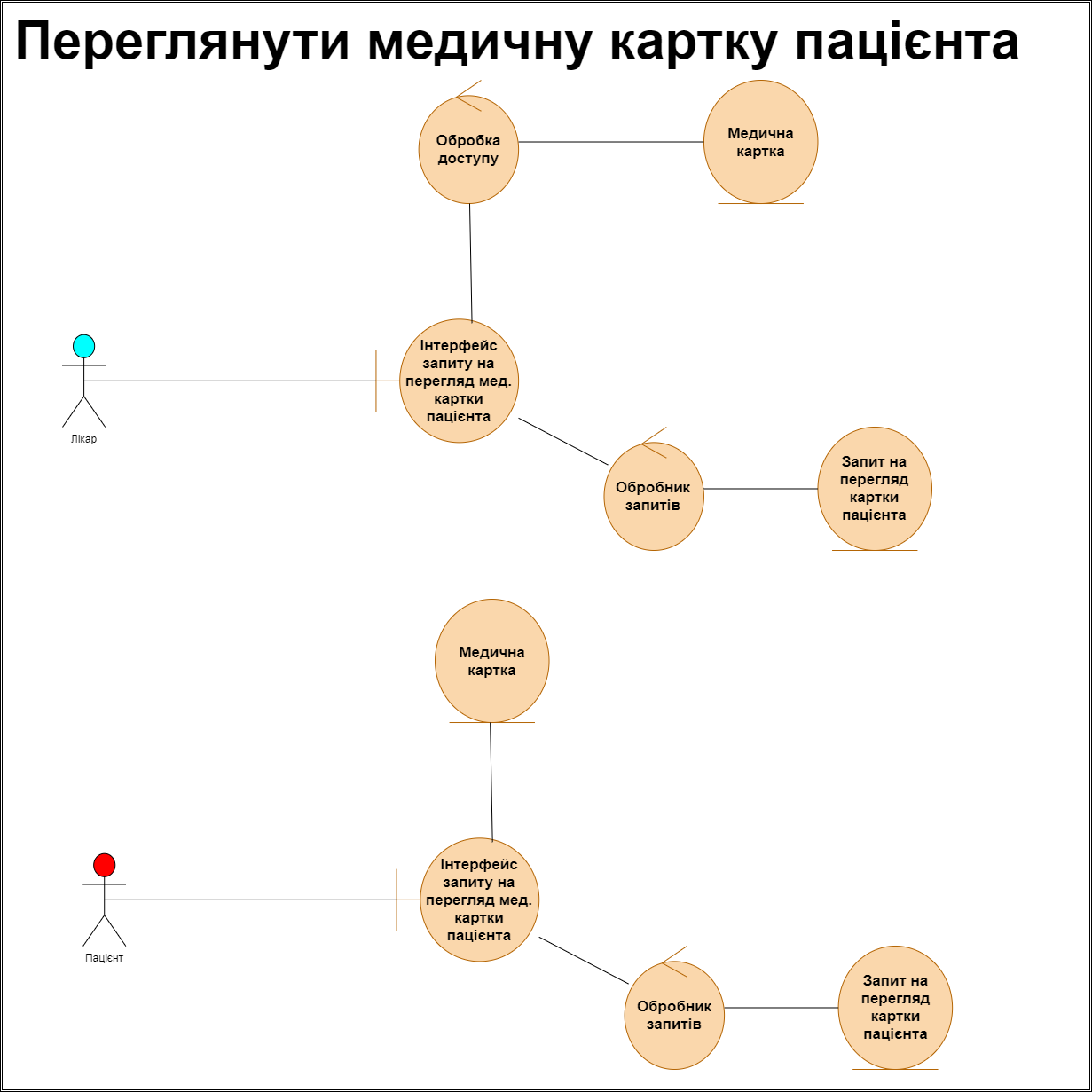




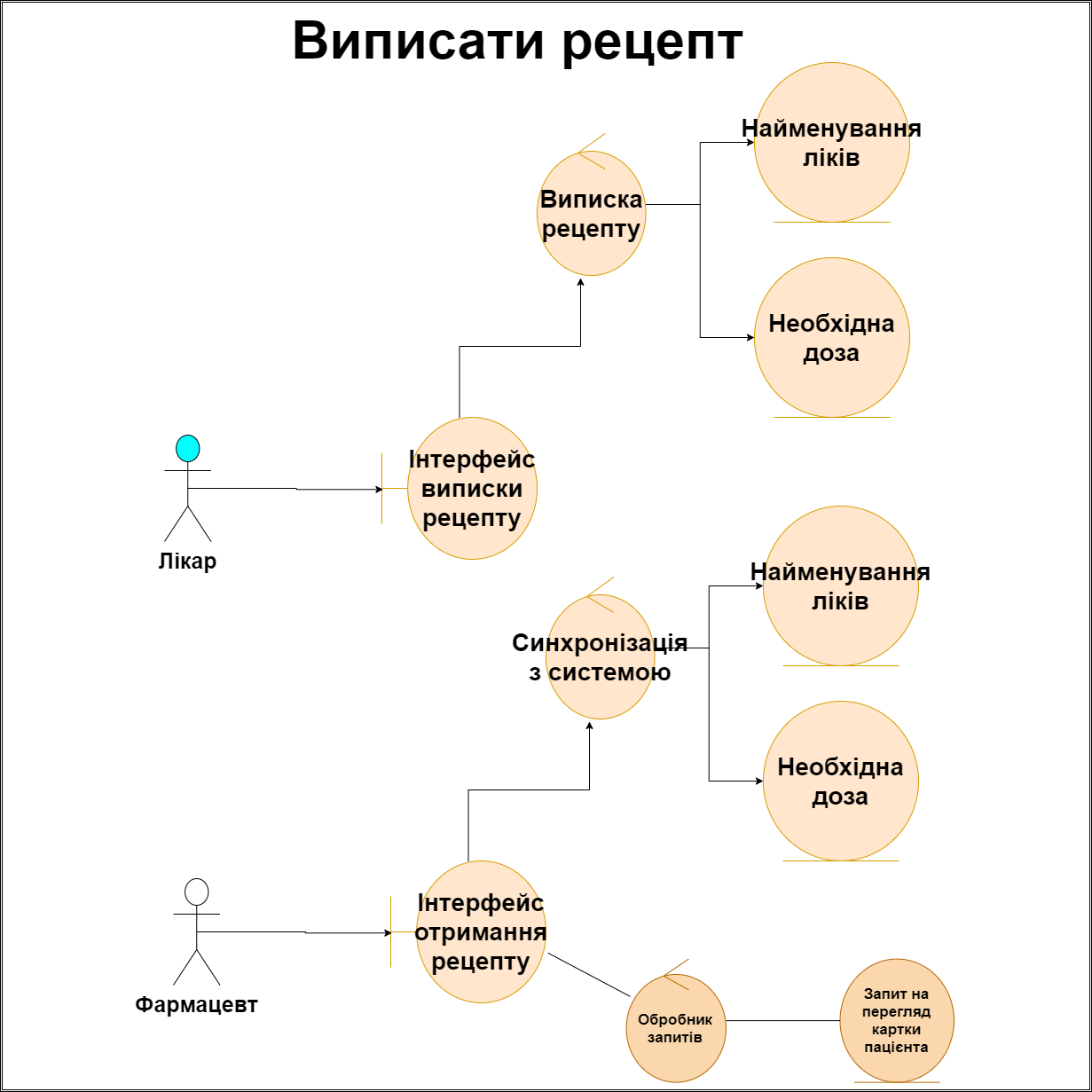




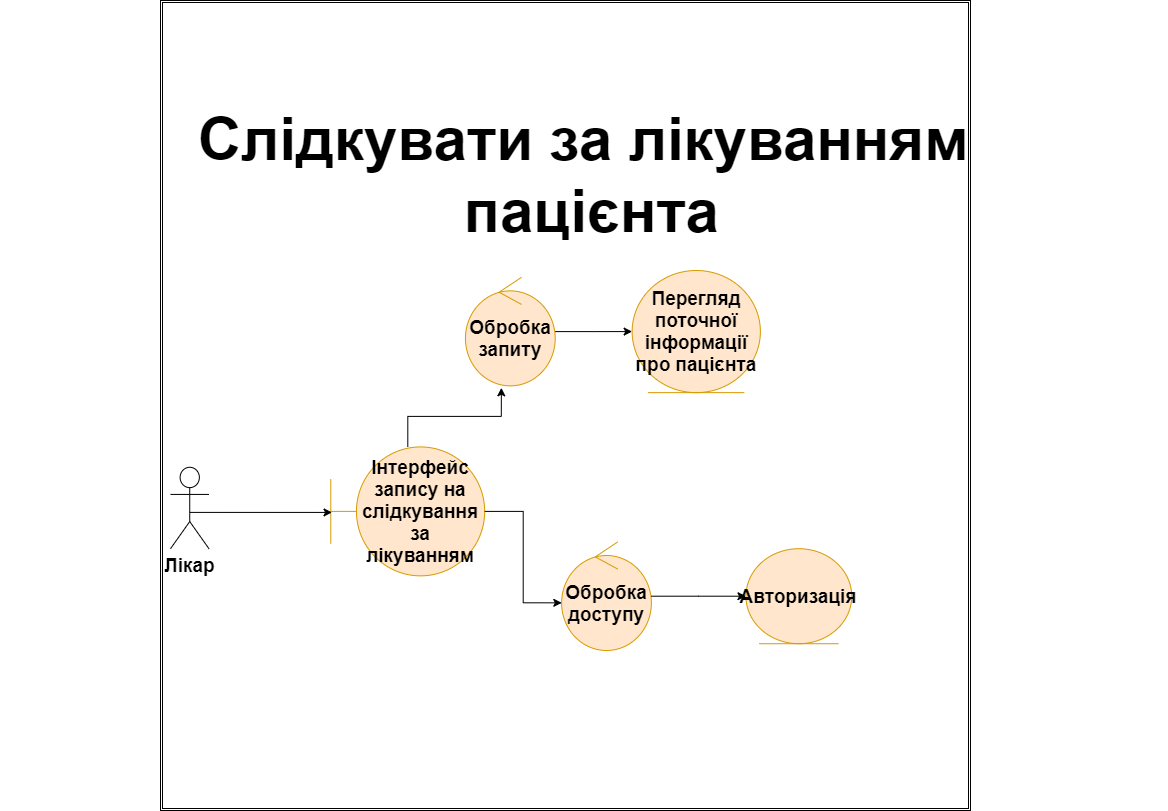


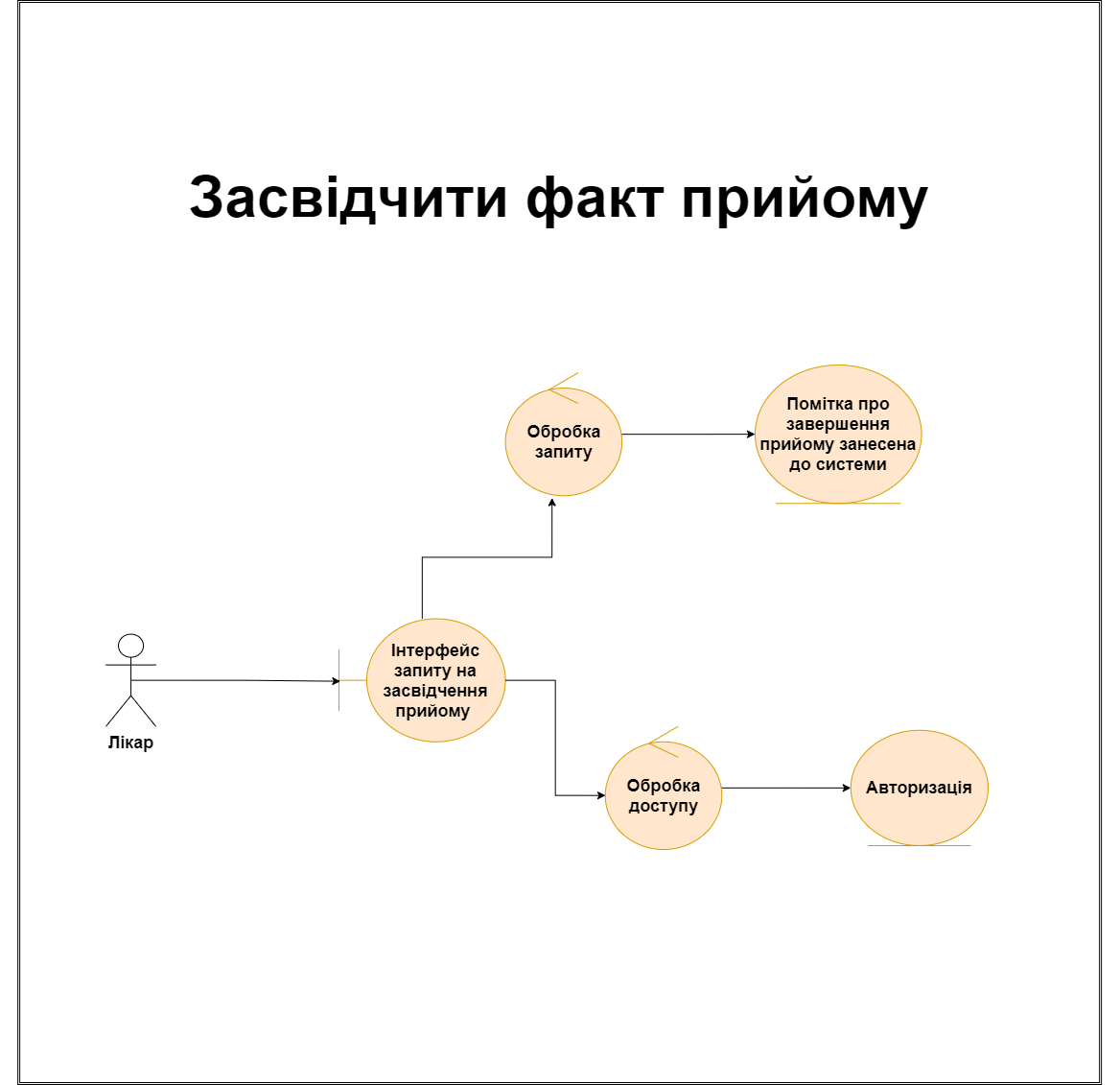


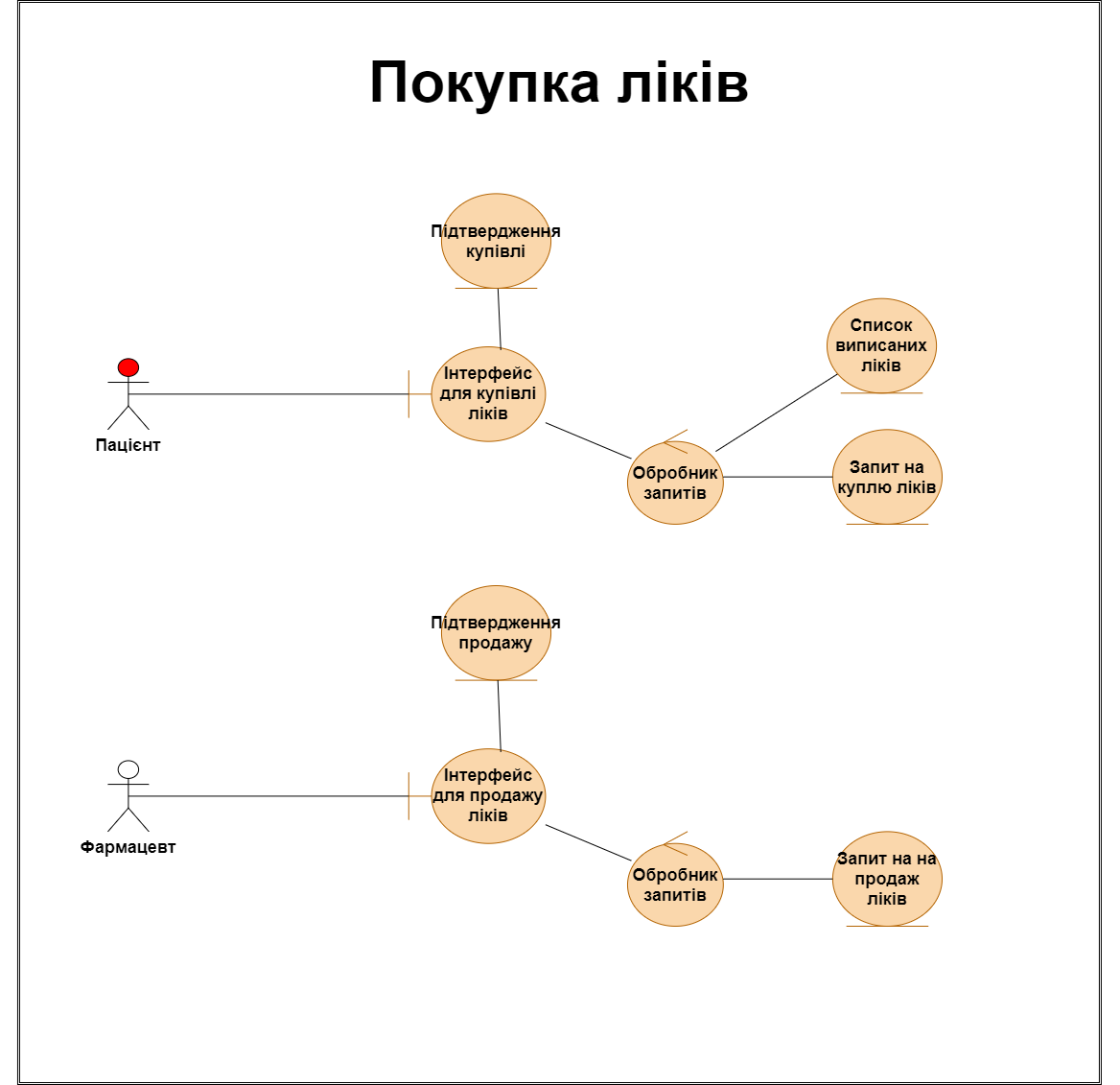


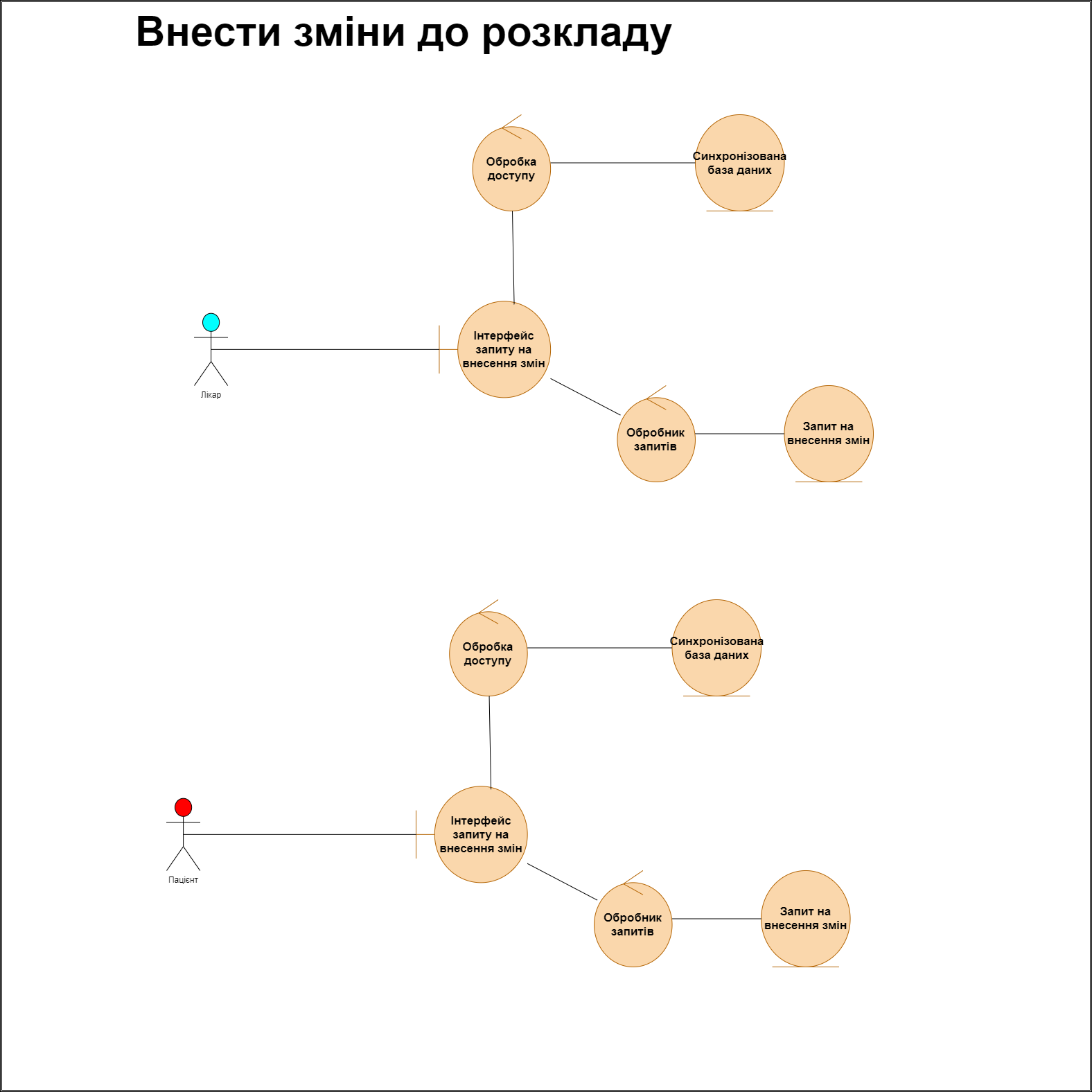


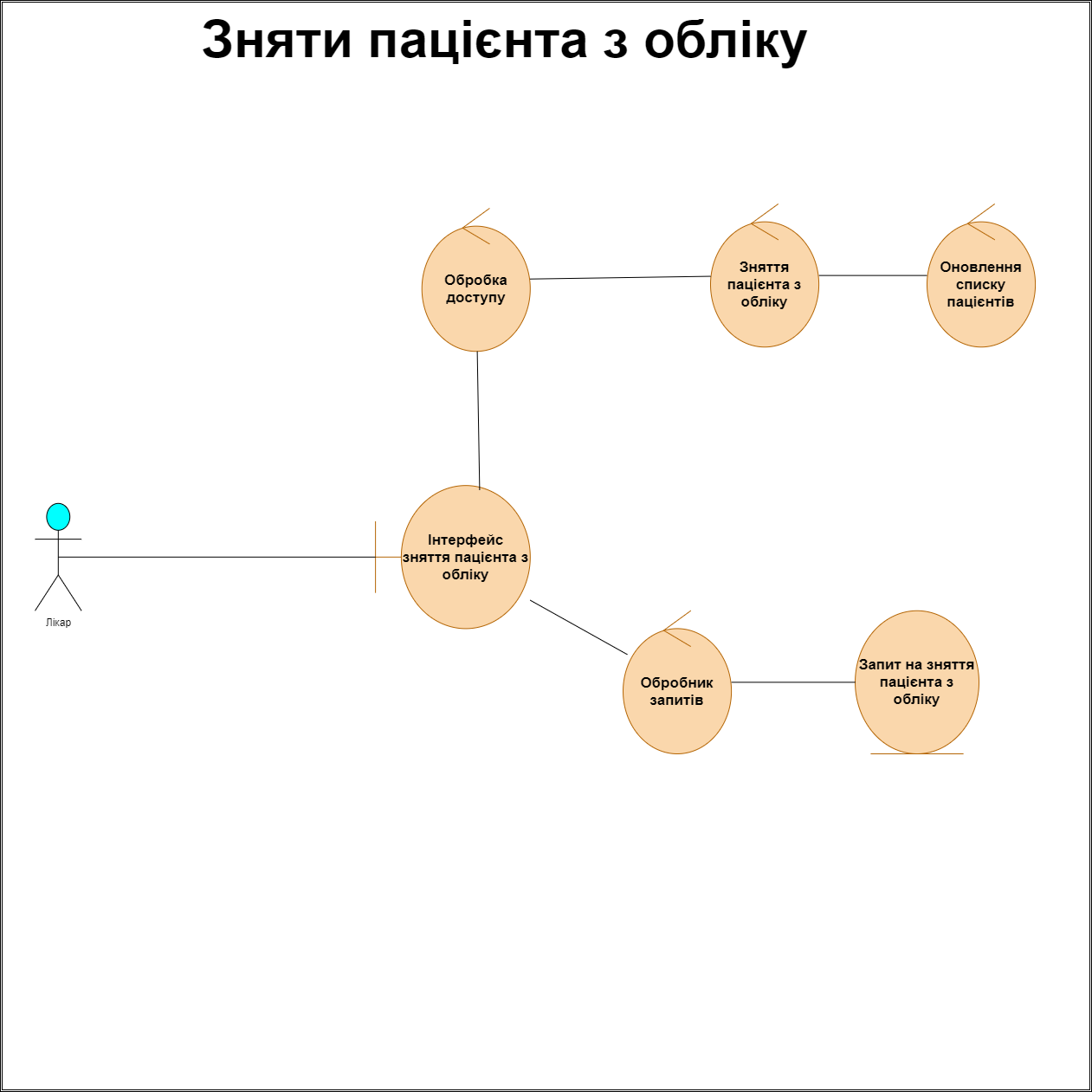










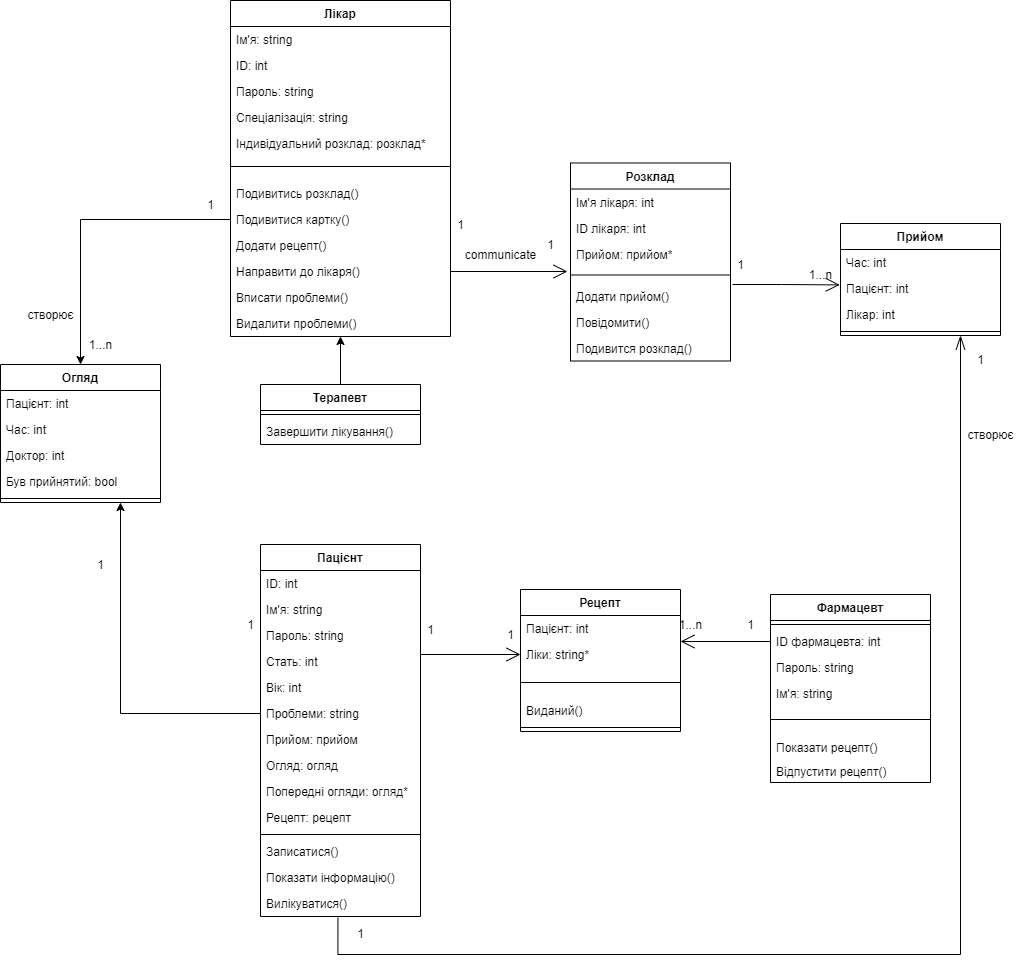


**Семантика класу**

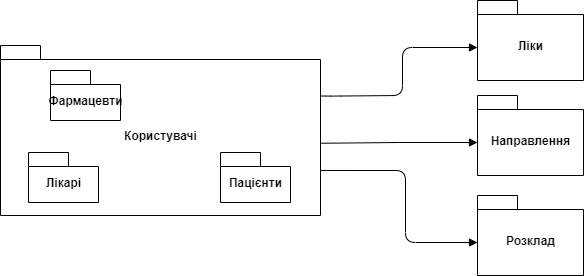
|  |  |
| --- | --- |
| Властивість класу проектування | Опис |
| Назва | Patient |
| Відношення | Відношення агрегації з класом Examination, відношення композиції з класом Recipe. Композиція з Appointment |
| Операції | is\_being\_treated()  make\_appointment(doctor\_id: int, time: int)  show\_info() |
| Атрибути | id: int  password: string  name: string  sex: int  age:int  health\_problems: vector<string>  previous\_examinations: vector<Examinations>  current\_examination: Examination  current\_appointment: Appointment  current\_recipe: Recipe |
| Назва | Doctor |
| Відношення | Відношення асоціації з класами Examination, відношення наслідування з класом Therapist, відношення композиції з класом Timetable, відношення асоціації з классом Recipe |
| Операції | show\_shedule()  show\_password()  add\_recipe(patient\_id: int, medicines: vector<string> medcines)  add\_referral(patient\_id: int, doctor\_id: int, purpose: string)  add\_new\_examination(doctor\_id: int)  add\_health\_problems(problem: vector<string> problems)  remove\_health\_problems(problem: vector<string> problems) |
| Атрибути | doctor\_id: int  password: string  name: string  specialization: string  personal\_shedule: Timetable |
| Назва | Therapist |
| Відношення | Відношення наслідування з классом Doctor |
| Операції | finish\_treatment(patient\_id: int) |
| Назва | Examination |
| Відношення | Відношення асоціації з класом Doсtor, відношення агрегації з класом Patient |
| Атрибути | patient\_id: int  time: int  doctor\_id: int  was\_held: bool = false |
| Назва | Timetable |
| Відношення | Відношення композиції з классом Doctor |
| Операції | add\_appointment(time: int, patient\_id: int, doctor\_id :int)  notificate(doctor\_id: int)  show\_table() |
| Атрибути | doctor\_id: int  doctor\_name: string  appointments: appointment\* |

|  |  |
| --- | --- |
| Назва | Appointment |
| Відношення | Відношення агрегації з класом Timetable. Композиція з Patient |
| Атрибути | time: int  patient\_id: int  doctor\_id: int |
| Назва | Pharmacist |
| Відношення | Відношення асоціації з TimeTable |
| Операції | show\_recipe()  mark\_as\_given |
| Атрибути | id: int  password: string  name: string |
| Назва | Recipe |
| Відношення | Відношення асоціації з Doctor, відношення композиції з Patient |
| Операції | set\_as\_given() |
| Атрибути | patient\_id: int  medicines: string\* |

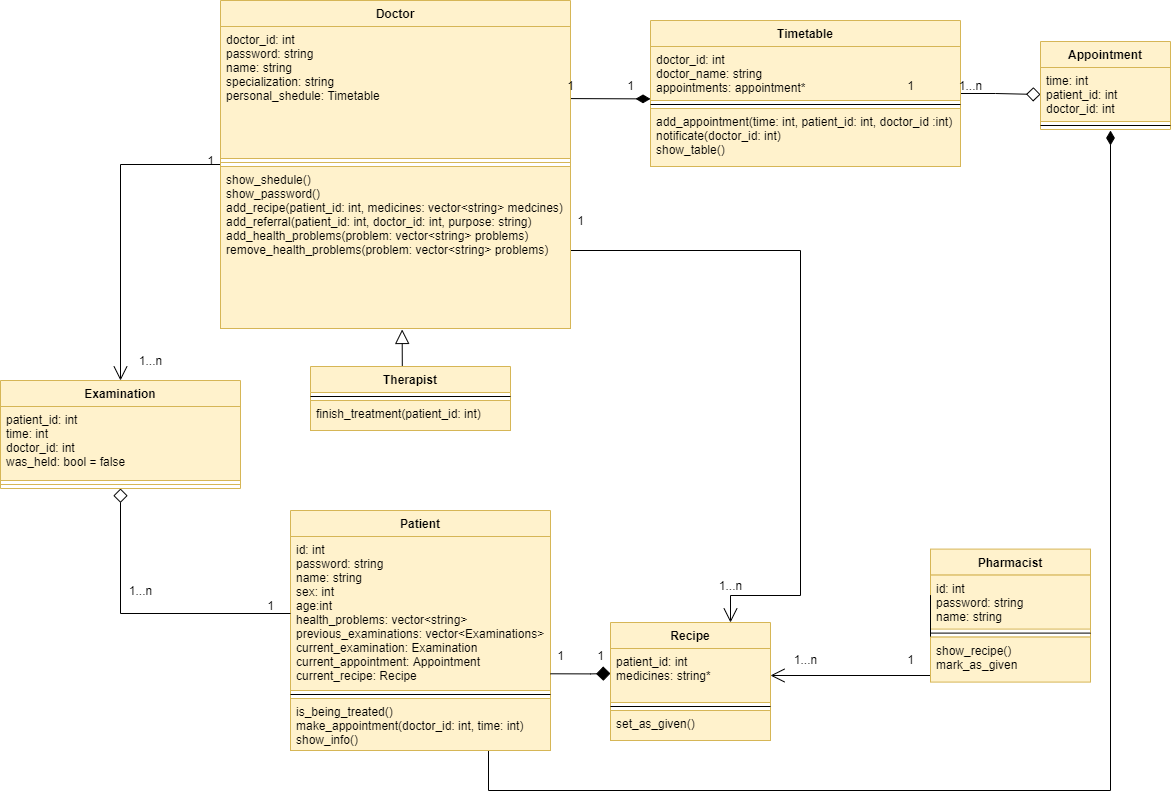
**Модель проектування**

****

**Діаграма пакетів**



**Діаграма класів**

****

**Висновок**

**Отже**, впродовж роботи над етапом проектування програмного забезпечення були проаналізовані та декларативно сформульовані основні аспекти функціонування майбутньої електронної системи поліклініки.

А саме:

* попередній етап визначення вимог та формування прецедентів був уточнений через розробку діаграм аналізу використання;
* були розроблені класи усіх об’єктів та структур, що надалі використовуватимуться у ПЗ;
* була висвітлена модель проектування, базуючись на попередньо створених діаграмах аналізу використання;
* була зібрана діаграма пакетів, що унаочнює структуру взаємодії окремо взятих глобальних компонентів програми;
* була створена діаграма класів, декларуючи основні життєвоважливі компоненти майбутнього програмного забезпечення.