

Міністерство освіти і науки України
Одеський національний політехнічний університет
Інститут комп'ютерних систем
Кафедра інформаційних систем

КУРСОВА РОБОТА

з дисципліни «Технології створення програмних продуктів»

за темою

«PutIO»

Виконав:

студент 3-го курсу

групи AI-181

Євтух А. О.

Перевірив:

Блажко О. А.

Одеса-2020

					ІС КР 122 AI-181 ПЗ	

Анотація

В курсовій роботі розглядається процес створення програмного продукту «PutIO». Робота виконувалась в команді з декількох учасників: Євтух Артем Олександрович, Яковенчук Ілля Валерійович, Карпухін Данило Олександрович. Тому в пояснювальній записці у розділах «Проектування» та «Конструювання» детальніше описано лише одну частину з урахуванням планів проведених робіт з розділу «Планування».

Результати роботи розміщено на *github*-репозиторії за адресою: <https://github.com/Artem3609/PutIO>.

					ІС КР 122 АІ-181 ПЗ	

Перелік скорочень

ОС – операційна система

ІС – інформаційна система

БД – база даних

СКБД – система керування базами даних

ПЗ – програмне забезпечення

ПП– програмний продукт

UML – уніфікована мова моделювання

					ІС КР 122 АІ-181 ПЗ	

Зміст

1 Вимоги до програмного продукту	7
1.1 Визначення потреб споживача	7
1.1.1 Ієрархія потреб споживача	7
1.1.2 Деталізація матеріальної потреби	8
1.2 Бізнес-вимоги до програмного продукту	8
1.2.1 Опис проблеми споживача	8
1.2.1.1 Концептуальний опис проблеми споживача	8
1.2.1.2 Метричний опис проблеми споживача	9
1.2.2 Мета створення програмного продукту	10
1.2.2.1 Проблемний аналіз існуючих програмних продуктів	10
1.2.2.2 Мета створення програмного продукту	12
1.2.3 Назва програмного продукту	12
1.2.3.1 Гасло програмного продукту	12
1.2.3.2 Логотип програмного продукту	12
1.3 Вимоги користувача до програмного продукту	13
1.3.1 Історія користувача програмного продукту	13
1.3.2 Діаграма прецедентів програмного продукту	14
1.3.3 Сценарії використання прецедентів програмного продукту	15
1.4 Функціональні вимоги до програмного продукту	20
1.4.1. Багаторівнева класифікація функціональних вимог	20
1.4.2 Функціональний аналіз існуючих програмних продуктів	23
1.5 Нефункціональні вимоги до програмного продукту	23
1.5.1 Опис зовнішніх інтерфейсів	23
1.5.1.1 Опис інтерфейса користувача	23
1.5.1.1.1 Опис INPUT-інтерфейса користувача	23
1.5.1.1.2 Опис OUTPUT-інтерфейса користувача	24
1.5.1.2 Опис інтерфейсу із зовнішніми пристроями	31

1.5.1.3	Опис програмних інтерфейсів	32
1.5.1.4	Опис інтерфейсів передачі інформації	32
1.5.1.5	Опис атрибутів продуктивності	32
2	Планування процесу розробки програмного продукту	33
2.1	Планування ітерацій розробки програмного продукту	33
2.2	Концептуальний опис архітектури програмного продукту	33
2.3	План розробки програмного продукту	34
2.3.1	Оцінка трудомісткості розробки програмного продукту	36
2.3.2	Визначення дерева робіт з розробки програмного продукту	40
2.3.3	Графік робіт з розробки програмного продукту	45
2.3.3.1	Таблиця з графіком робіт	45
2.3.3.2	Діаграма Ганта	46
3	Проектування програмного продукту	47
3.1	Концептуальне та логічне проектування структур даних програмного продукту	47
3.1.1	Концептуальне проектування на основі UML-діаграми концептуальних класів	47
3.1.2	Логічне проектування структур даних	48
3.2	Проектування програмних класів	49
3.3	Проектування алгоритмів роботи методів програмних класів	50
3.4	Проектування тестових наборів методів програмних класів	53
4	Конструювання програмного продукту	62
4.1	Особливості конструювання структур даних	62
4.1.1	Особливості інсталяції та роботи з СУБД	62
4.1.2	Особливості створення структур даних	62
4.2	Особливості конструювання програмних модулів	64
4.2.1	Особливості роботи з інтегрованим середовищем розробки	64
4.2.2	Особливості створення програмної структури з урахуванням спеціалізованого Фреймворку	65

4.2.3 Особливості створення програмних класів	66
4.2.4 Особливості розробки алгоритмів методів програмних класів або процедур/функцій	68
4.3 Модульне тестування програмних класів	72
5 Розгортання та валідація програмного продукту	84
5.1 Інструкція з встановлення програмного продукту	84
5.2 Інструкція з використання програмного продукту	84
5.3 Результати валідації програмного продукту	90
Висновки до курсової роботи	91

					ІС КР 122 АІ-181 ПЗ	

1 Вимоги до програмного продукту

1.1 Визначення потреб споживача

1.1.1 Ієрархія потреб споживача

Відомо, що в теорії маркетингу потреби людини можуть бути представлені у вигляді ієрархії потреб ідей американського психолога Абрахама Маслоу включають рівні:

- фізіологія (вода, їжа, житло, сон);
- безпека (особиста, здоров'я, стабільність),
- приналежність (спілкування, дружба, любов),
- визнання (повага оточуючих, самооцінка),
- самовираження (вдосконалення, персональний розвиток).

На рисунку 1.1 представлено рівень потреби споживача, який хотілося б задовольнити, використовуючи майбутній програмний продукт.

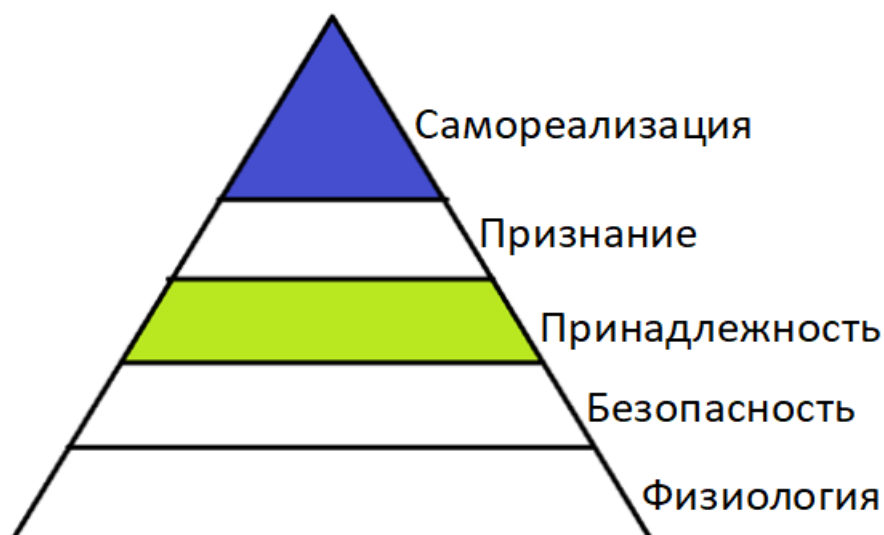


Рисунок. 1.1.1 – Ієрархія потреби споживача. Рівень потреби – приналежність

Був обраний рівень «Приналежність», тому що, використовуючи програмний продукт «RuTio», споживач задовольняє такі потреби, як потреба у дружбі та спілкуванні з іншими користувачами для взаємної гарної подорожі, а також рівень «Самореалізації», тому що споживач пізнає та удосконалює себе, разом з сумісними подорожами.

					ІС КР 122 АІ-181 ПЗ	

1.1.2 Деталізація матеріальної потреби

Для деталізації матеріальної потреби можна скористатися ментальними картами (MindMap). При створенні ментальних карт матеріальна потреба розташовується в центрі карти. Асоціативні гілки можна швидко створити, припускаючи, що в загальному вигляді з об'єктом пов'язані три потоки даних / інформації: вхідний, внутрішній, вихідний. Кожен потік - це асоціативна група, що включає можливі п'ять гілок, що відповідають на п'ять питань: Хто? Що? Де? Коли? Як?

Відповідно до рекомендацій по створенню ментальних карт кожна гілка-асоціація може бути розділена на додаткові асоціативні гілки, які деталізують відповіді на поставлені питання.

Потреба, яка була визначена при аналізі матеріальних проблем споживача, основні та додаткові асоціативні гілки зображені на рис 1.1.2

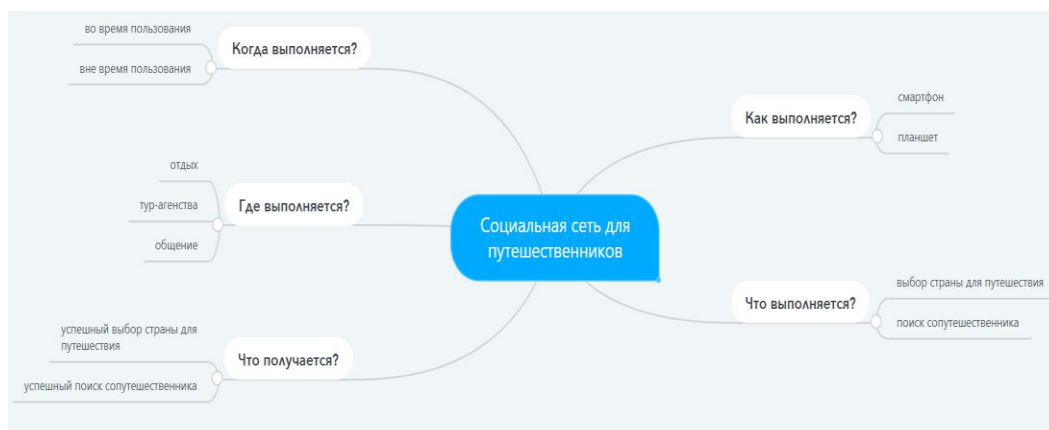


Рисунок - 1.1.2 – Деталізація матеріальної потреби

1.2 Бізнес-вимоги до програмного продукту

1.2.1 Опис проблеми споживача

1.2.1.2 Метричний опис проблеми споживача

Метричний опис проблем споживача відображені у табл.1.2.1

Таблица – 1.2.1

					ІС КР 122 АІ-181 ПЗ	

№	Загальний опис проблеми	Метричні показники незадоволеності споживача
1	Важко обрати місце для подорожі та знайти супутешественника	Низький рівень доступності пошуку місця для подорожі та пошуку

Рівень доступності $AL = NA / N$, де
 NA - кількість власників, які зможуть обійти всі притулки або перейти
на всі існуючі веб-сайти про зниклих тварин.
 N - загальна кількість власників тварин, які загубилися.
 $AL \rightarrow 0$

1.2.2 Мета створення програмного продукту

1.2.2.1 Проблемний аналіз існуючих програмних продуктів

Проблемний аналіз існуючих програмних продуктів відображена у
табл. 2.1

№	Назва продукту	Вартість	Ступінь готовності	Примітка
1	Nomadlist	безкоштовно	1	Невозможность поиска супутешественника и отсутствие отзывов о месте для путешествия
2	Tripadvisor	безкоштовно	1	Невозможность поиска супутешественника
3	Instagram	безкоштовно	1	Трудность выбора места для путешествия и поиска супутешественника
Facebook		безкоштовно	1	Трудность выбора места для путешествия и поиска

					ІС КР 122 АІ-181 ПЗ	

1.2.2.2 Мета створення програмного продукту

Підвищення рівня доступності вибору місця для подорожі і пошуку супутешественника на основі створення апаратно-програмного інтегратора функцій соціальної мережі для мандрівників

1.2.3 Назва програмного продукту

Назва «Putio» відображає мету та гасло програмного продукту. Головна задача створення веб-сервісу – це допомога споживачам мандрувати з задоволенням, а також з своїм супутешественником.

1.2.3.1 Гасло програмного продукту

Гасло — лаконічна фраза, що впадає в око, добре запам'ятовується та висловлює суть продукту.

На початковому етапі розробки було вигадано гасло, яке найкращим чином відображає мету та суть роботи веб-сервісу та пов'язано з логотипом та назвою.

Гасло - «Мандруй з задоволенням!»

1.2.3.2 Логотип програмного продукту

Відмінним способом представлення назви програмного продукту є його логотип, що поєднує зорові образи.

На рис 1.2.3.2 можна побачити розроблений командою логотип, який відображає назву веб-сервісу.



1.3 Вимоги користувача

1.3.1 Історія користувача програмного продукту

Гость может:

- Зарегистрироваться

					ІС КР 122 АІ-181 ПЗ	

- Авторизоваться

Пользователь может:

- Заполнить профиль
- Выставить публикацию
- Оставить комментарий под постом/ лайкнуть пост/ сохранить пост
- Написать другому пользователю/ подписаться на другого пользователя
- Найти путешественника
- Найти информацию по месту для путешествия

Модератор может:

- Заполнить профиль
- Выставить публикацию
- Оставить комментарий под постом/ лайкнуть пост/ сохранить пост/ редактировать пост/ удалить пост
- Написать другому пользователю/ подписаться на другого пользователя/ редактировать профиль другого пользователя/ забанить другого пользователя
- Найти путешественника
- Найти информацию по месту для путешествия
- Просмотреть статистику пользования программным продуктом

1.3.2 Діаграма прецедентів програмного продукту

Діаграма прецедентів (Use Case UML-діаграма) включає:

- актори (зацікавлені особи і зовнішні системи зі своїм API);
- прецеденти як основні функції ПП;
- зв'язки між прецедентами і акторами як множиною зацікавлених осіб;
- можливі зв'язки-узагальнення між акторами.

Діаграма прецедентів програмного продукту зображена на рис. 1.3.1

					ІС КР 122 АІ-181 ПЗ	



Рисунок - 1.3.2. Діаграма прецедентів програмного продукту

1.3.3 Сценарії використання прецедентів програмного продукту

Для кожного прецедента описан сценарій використання з урахуванням пунктів:

- назва прецеденту;
- передумови початку виконання прецеденту;
- актори як зацікавлені особи у виконанні прецеденту;
- актор-основна зацікавлена особа як ініціатор початку прецеденту;
- гарантії успіху (що отримають актори у разі успішного завершення прецеденту);
- основний успішний сценарій;
- альтернативні сценарії, прив'язані до кроків основного успішного сценарію.

Название прецедента: Регистрация

Актеры: гость

Предусловие: открытое окно регистрации

Постусловие: успешная регистрация

Сценарий:

1. Гость открывает страницу регистрации
2. Система показывает форму регистрации
3. Гость вводит данные (почта и пароль)
4. Гость зарегистрирован

Название прецедента: Авторизация

Актеры: гость

Предусловие: открытое окно авторизации

Постусловие: успешная авторизация

Сценарий:

1. Гость открывает окно авторизации
2. Система показывает форму авторизации
3. Гость вводит почту и пароль
4. Гостя авторизировано

Название прецедента: Заполнение профиля

Актеры: пользователь, модератор

Предусловие: открытое окно редактирования профиля

Постусловие: успешное заполнение профиля

Сценарий:

1. Пользователь/ модератор открывает окно редактирования профиля

					IC KP 122 AI-181 ПЗ	

2. Пользователь/ модератор вводит данные о личности, добавляет фотографии
3. Система добавляет данные о пользователе в БД

Название прецедента: Выставление публикаций

Актеры: пользователь, модератор

Предусловие: открытое окно добавления публикации

Постусловие: успешное добавление публикации

Сценарий:

1. Пользователь/ модератор открывает окно для добавления публикации
2. Пользователь/ модератор заполняет публикацию
3. Система выставляет публикацию

Название прецедента: Поиск информации по месту для путешествия

Актеры: пользователь, модератор

Предусловие: открытая строка поиска информации по месту для путешествия

Постусловие: результат поиска информации по месту для путешествия

Сценарий:

1. Пользователь/ модератор открывает строку поиска информации по месту для путешествия
2. Пользователь/ модератор вводит желаемые страну/ город для путешествия
3. Система показывает посты про искомые страну/ город

Название прецедента: Поиск сопутешественника

					IS KP 122 AI-181 ПЗ	

Актеры: пользователь, модератор

Предусловие: открытая строка поиска путешественника

Постусловие: успешный поиск потенциальных путешественников

Сценарий:

1. Пользователь/ модератор открывает строку поиска путешественника
2. Пользователь/ модератор вводит желаемые параметры поиска путешественника
3. Система подбирает пользователей со схожими параметрами

Название прецедента: Взаимодействие с публикациями других пользователей

Актеры: пользователь

Предусловие: открытое окно просмотра публикаций

Постусловие: успешное взаимодействие с публикацией другого пользователя

Сценарий:

1. Пользователь открывает окно просмотра публикаций
2. Пользователь лайкает публикацию/ сохраняет публикацию/ оставляет комментарий под публикацией

Название прецедента: Взаимодействие с другими пользователями

Актеры: пользователь

Предусловие: открытый профиль другого пользователя

Постусловие: успешное взаимодействие с другим пользователем

Сценарий:

1. Пользователь открывает профиль другого пользователя

					IS KP 122 AI-181 ПЗ	

2. Пользователь подписывается на другого пользователя/ пишет сообщение другому пользователю

Название прецедента: Особое взаимодействие с публикациями других пользователей

Актеры: модератор

Предусловие: открытое окно просмотра публикаций

Постусловие: успешное взаимодействие с публикацией другого пользователя

Сценарий:

1. Модератор открывает окно просмотра публикаций
2. Модератор лайкает публикацию/ сохраняет публикацию/ оставляет комментарий под публикацией/ редактирует публикацию/ удаляет публикацию

Название прецедента: Особое взаимодействие с другими пользователями

Актеры: модератор

Предусловие: открытый профиль другого пользователя

Постусловие: успешное взаимодействие с другим пользователем

Сценарий:

1. Модератор открывает профиль другого пользователя
2. Модератор подписывается на другого пользователя/ пишет сообщение другому пользователю/ редактирует профиль другого пользователя/ банит другого пользователя

Название прецедента: Просмотр статистики

Актеры: модератор

Предусловие: открытое окно просмотра статистики

Постусловие: успешный просмотр статистики

					IS KP 122 AI-181 ПЗ	

Сценарий:

1. Модератор открывает окно для просмотра статистики
2. Система открывает форму просмотра статистики
3. Модератор выбирает нужные для статистики фильтры (по публикациям, тегам и т.д.)
4. Модератор просматривает нужную информацию

1.4 Функціональні вимоги до програмного продукту

1.4.1. Багаторівнева класифікація функціональних вимог

Таблица – 1.4.1. Багаторівнева класифікація функціональних вимог

FR 1	Создание учетной записи
FR 1.1	Запрос на ввод конфиденциальных данных (логин, пароль)
FR 1.2	Запрос на ввод <u>персональных</u> данных для идентификации системой интересов пользователя
FR 1.3	Создания сессии для пользователя (если вход первый- то сразу запуск системы рекомендаций)
FR 2	Авторизация
FR 2.1	Запрос на ввод конфиденциальных данных (логин, пароль)
FR 2.2	Проверка введенных данных/ поиск в базе данных пользователя
FR 2.3	Переход на главное меню

FR 3	Создание публикации
FR 3.1	Загрузка основного контента
FR 3.2	Добавление описание с использованием идентификаторов для попадания в систему
FR 3.3	<u>Модерация</u>
FR 4	Умный поиск
FR 4.1	Ввод данных
FR 4.2	Трансформация введенных данных в идентификаторы
FR 4.3	Переход на интересующий контент

FR 5	Запрос на поиск <u>сопутешественника</u>
FR 5.1	Запрос данных для характерного поиска
FR 5.2	Подбор списка желаемых профилей
FR 5.3	Отправление запроса
FR 5.4	<u>Модерация</u> запроса

FR 6	Просмотр статистики
FR 6.1	Анализ статистики соц. сети в целом
FR 6.2	Анализ соц. сети исключительно пользователя
FR 7	Взаимодействие с постами
FR 8	Взаимодействие с другими пользователя

FR 9	Особое Взаимодействие с постами
FR 10	Особое Взаимодействие с другими пользователями
FR 11	Заполнения профиля
FR 11.1	Выбор интересов
FR 11.2	Публикация первых постов

1.4.2 Функціональний аналіз існуючих програмних продуктів

Таблица – 4.2. Функціональний аналіз існуючих програмних продуктів

					IC KP 122 AI-181 ПЗ	

Ідентифікатор функції	Instagram	Facebook
FR1	+	+
FR2	+	+
FR3	+	+
FR4	-	-
FR5	+-	+-
FR6	+	+
FR7	+	+
FR8	+	+
FR9	-	-
FR10	-	-
FR11	-	-

1.5 Нефункціональні вимоги до програмного продукту

1.5.1 Опис зовнішніх інтерфейсів

1.5.1.1 Опис інтерфейса користувача

1.5.1.1.1 Опис INPUT-інтерфейса користувача

Таблиця – 1.5.1.1.1. Опис INPUT-інтерфейса користувача

Ідентифікатор функції	Засіб INPUT-потoku	Особливості використання
FR1	Маніпулятор типу миша, клавіатура, touch screen	-
FR2	Маніпулятор типу миша, клавіатура, touch screen	-
FR3	Маніпулятор типу миша, клавіатура, touch screen	-
FR4	Маніпулятор типу миша, клавіатура, touch screen	-
FR5	Маніпулятор типу миша, Клавіатура, touch screen	-
FR6	Маніпулятор типу миша, клавіатура, touch screen	-

FR7	Маніпулятор типу миша, клавіатура, touch screen	-
FR8	Маніпулятор типу миша, клавіатура, touch screen	-
FR9	Маніпулятор типу миша, клавіатура, touch screen	-
FR10	Маніпулятор типу миша, клавіатура, touch screen	-
FR11	Маніпулятор типу миша, клавіатура, touch screen	-

1.5.1.1.2 Опис OUTPUT-інтерфейса користувача

Ідентифікатор функції	Засіб OUTPUT-потoku	Особливості використання
FR1	Графічний інтерфейс	-
FR2	Графічний інтерфейс	-
FR3	Графічний інтерфейс	-
FR4	Графічний інтерфейс	-
FR5	Графічний інтерфейс	-
FR6	Графічний інтерфейс	-
FR7	Графічний інтерфейс	-
FR8	Графічний інтерфейс	-
FR9	Графічний інтерфейс	-
FR10	Графічний інтерфейс	-
FR11	Графічний інтерфейс	-

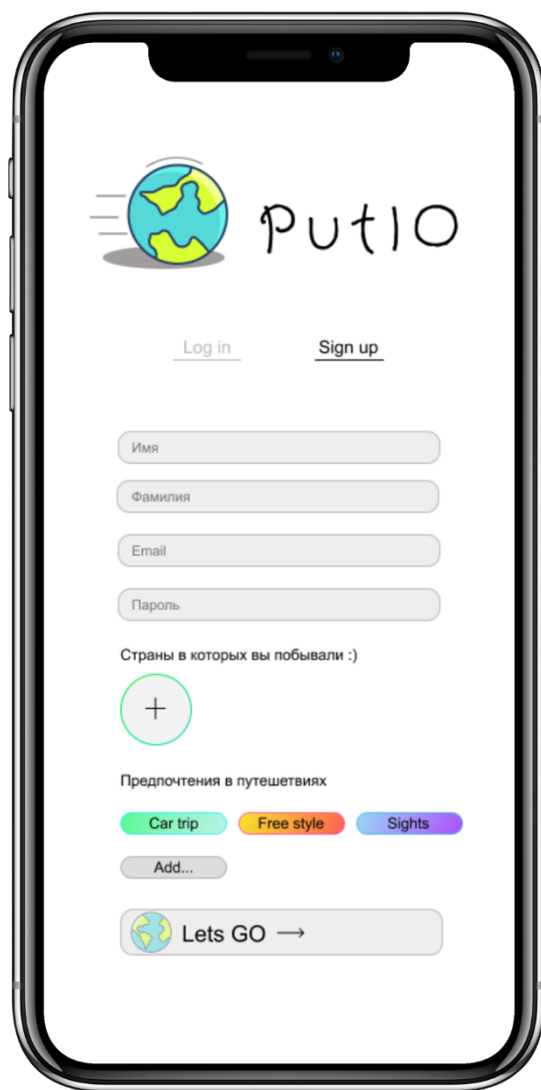


Рис. 1.5.1.1.2.1 – сторінка реєстрації

					IC KP 122 AI-181 ПЗ	

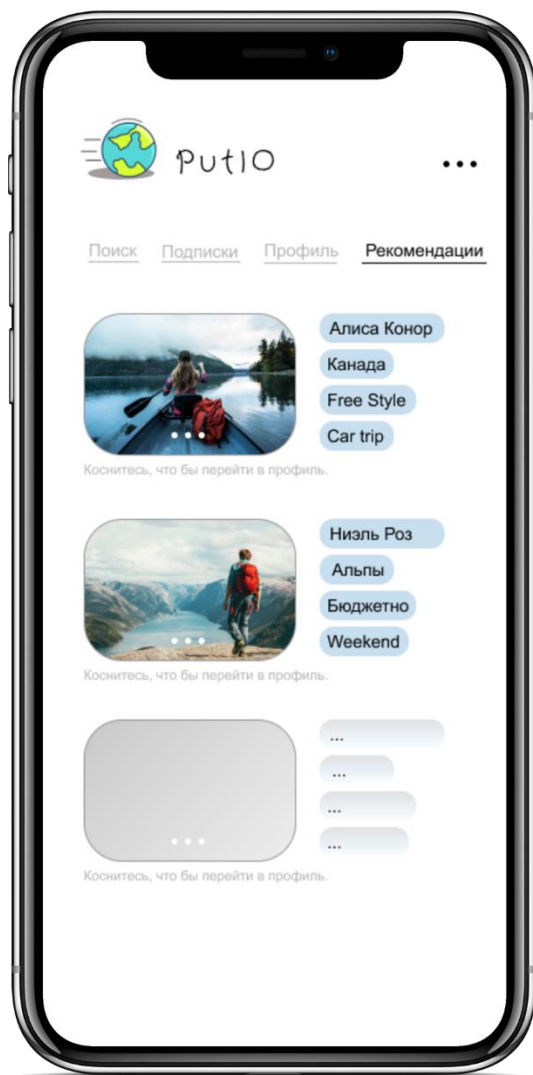


Рис. 1.5.1.1.2.2 – сторінка рекомендацій

					IC KP 122 AI-181 ПЗ	



Рис. 1.5.1.1.2.3 – сторінка пошуку спутешественнік

1.5.1.2 Опис інтерфейсу із зовнішніми пристроями

FR1.1 – Desktop, смартфон

FR4.3 – Desktop, смартфон

FR5.1 – Desktop, смартфон

1.5.1.3 Опис програмних інтерфейсів

					IC KP 122 AI-181 ПЗ	

Программные интерфейсы:

- Android,
- Веб-браузер ОС Windows.

1.5.1.4 Опис інтерфейсів передачі інформації

Интерфейсы передачи информации:

- Проводные (Ethernet),
- Беспроводные (Wi-Fi).

1.5.1.5 Опис атрибутів продуктивності

Ідентифікатор функції	Максимальний час реакції ПП на дії користувача, секунди
FR1.1	4
FR4.3	4
FR5.1	5

2 Планування процесу розробки програмного продукту

2.1 Планування ітерацій розробки програмного продукту

З метою забезпечення для вимог таких рекомендацій IEEE-стандарту, як

необхідність, корисність при експлуатації, здійсненність функціональних вимог до ПП, було визначено функціональні пріоритети, які будуть використані при плануванні ітерацій розробки ПП.

При створенні пріоритетів необхідно врахувати:

— сценарні залежності між прецедентами, до яких належать функції, на основі аналізу пунктів передумов початку роботи прецедентів, вказаних в описі сценаріїв роботи прецедентів;

					ІС КР 122 АІ-181 ПЗ	

– вплив роботи прецеденту, до якого належить функція, на досягнення мети ПП, наприклад у відсотках, на основі аналізу пунктів гарантій успіху, вказаних в описі сценаріїв роботи прецедентів.

Сценарні залежності будуть перетворені у відповідні функціональні залежності. Вплив роботи прецеденту буде поширено на всі підлеглі функції ієрархії. При визначенні пріоритетів рекомендується використовувати наступні позначки:

- M (Must) – функція повинна бути реалізованою у перших ітераціях за будь-яких обставин;
- S (Should) – функція повинна бути реалізованою у перших ітераціях, якщо це взагалі можливо;
- C (Could) – функція може бути реалізованою, якщо це не вплине негативно на строки розробки;
- W (Want) – функція може бути реалізованою у наступних ітераціях.

Опису представлено в таблиці 2.1

Таблиця 2.1 – Опис функціональних пріоритетів

Функции	Функциональные зависимости	Влияние на достижение цели, %	Приоритет функции
FR 1	-	5	M
FR 2	FR1	-	M
FR 3	-	-	C
FR 4	-	25	M
FR 5	-	45	M
FR 6	-	-	C
FR 7	-	5	S
FR 8	-	10	S
FR 9	-	-	S
FR 10	-	-	S
FR 11	-	10	M

2.2 Концептуальний опис архітектури програмного продукту

Концептуальний опис архітектури програмного продукту зображено на рис. 2.2

					IC KP 122 AI-181 ПЗ	

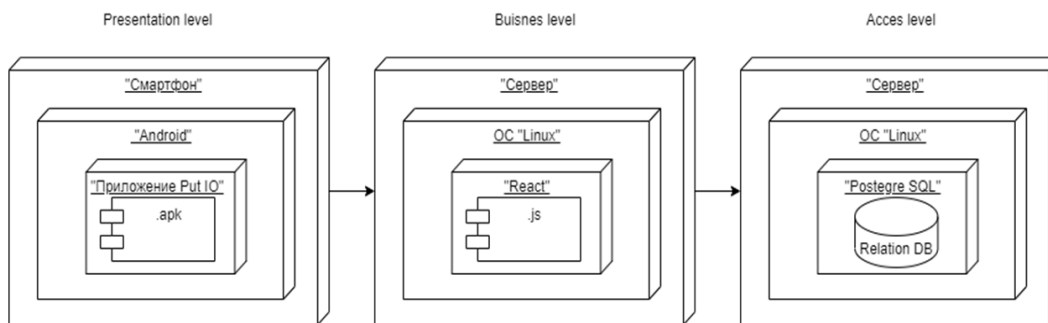


Рисунок – 2.2. Концептуальний опис архітектури програмного продукту

2.3.1 Оцінка трудомісткості розробки програмного продукту

Визначення нескорегованого показника UUCP (Unadjusted Use Case Points)

1. Визначення вагових показників акторів А

Всі актори діляться на три типи: прості, середні і складні.

Простий актор представляє зовнішню систему з чітко визначеним Програмним інтерфейсом.

Середній актор представляє або зовнішню систему, що взаємодіє з ПП за

допомогою мережевих протоколів, або особистість, що користується текстовим інтерфейсом (наприклад, алфавітно-цифровим терміналом).

Складний актор представляє особистість, що користується графічним інтерфейсом. Загальна кількість акторів кожного типу помножується на відповідний ваговий коефіцієнт, потім обчислюється загальний ваговий показник(таблиця 2.3.1.1).

Таблиця – 2.3.1.1. Показники складності акторів і прецедентів

					ІС КР 122 АІ-181 ПЗ	

Тип актёра	Весовой коэффициент	Количество
Простой	1	1
Средний	2	1
Сложный	3	1

Тип прецедента	Количество шагов сценария	Весовой коэффициент	Количество
Простой	2-3	5	8
Средний	4-7	10	3
Сложный	>7	15	0

$$A = 6 \quad UC = 70 \quad UUCP = A + UC = 76$$

Активация Windows
Чтобы активировать Windows, перейдите в

Технічна складність проекту (TCF - Technical Complexity Factor)
обчислюється з урахуванням показників технічної складності.

Кожному показнику присвоюється значення ST_i в діапазоні від 0 до 5:

- 1) 0 означає відсутність значимості показника для даного проекту;
- 2) 5 - високу значимість.

Показники технічної складності проекту TCF представлено в таблиці 2.3.1.2.

Таблиця – 2.3.1.2. Показники технічної складності проекту

Показатель	Описание показателя	Вес	ST
T1	Распределенная система	2	4
T2	Высокая производительность (пропускная способность)	1	3
T3	Работа конечных пользователей в режиме онлайн	1	5
T4	Сложная обработка данных	-1	3
T5	Повторное использование кода	1	2
T6	Простота установки	0.5	2
T7	Простота использования	0.5	2
T8	Переносимость	2	2
T9	Простота внесения изменений	1	3
T10	Параллелизм	1	3
T11	Специальные требования к безопасности	1	4
T12	Непосредственный доступ к системе со стороны внешних пользователей	1	2
T13	Специальные требования к обучению пользователей	1	1

$$TCF = 0,6 + (0,01 * (ST_i * Вес_i)) = 0.94$$

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рівень кваліфікації розробників (EF - Environmental Factor) обчислюється

з урахуванням наступних показників (табл. 2.3.1.3).

Таблиця – 2.3.1.3. Показники рівня кваліфікації розробників

Показатель	Описание показателя	Вес	SF
F1	Знакомство с технологией	1.5	2
F2	Опыт разработки приложений	0.5	2
F3	Опыт использования объектно-ориентированного подхода	1	3
F4	Наличие ведущего аналитика	0.5	1
F5	Мотивация	1	5
F6	Стабильность требований	2	2
F7	Частичная занятость	-1	3
F8	Сложные языки программирования	-1	1

$$EF = 1,4 + (-0,03 * (SF_i * Beci)) = 0.965$$

Активация Windows
Чтобы активировать Windows, перейдите к...

2.3.2 Визначення дерева робіт з розробки програмного продукту

При створенні дерева робіт (Work BreakDown Structure- WBS)

використовується дерево функцій, яке було створено раніше.

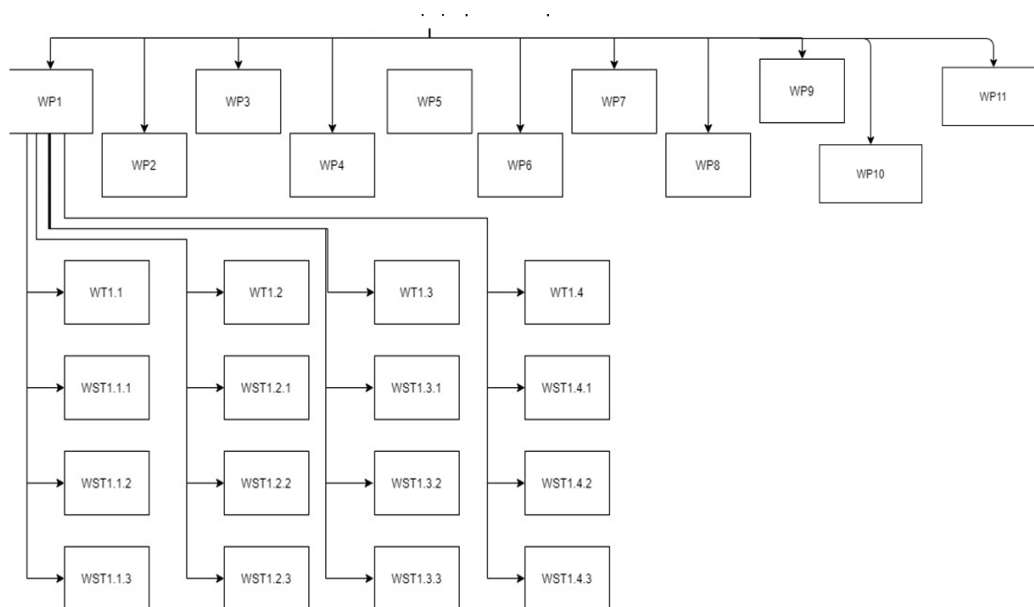
Кожна функція 1-го рівня ієрархії перетворюється в Work Package (WP)

Кожна функція 2-го рівня ієрархії перетворюється в Work Task (WT).

Для кожної задачі визначаються підзадачі - Work SubTask (WST) з урахуванням базових процесів розробки програмних модулів: проектування, конструювання, модульне тестування, збірка та системне тестування.

WBS представлено на рисунку 2.3.2 .1

					IC KP 122 AI-181 ПЗ	



Автоматизация Windows

Рис. 2.3.2.1 - Дерево робіт з розробки програмного продукту

2.3.3 Графік робіт з розробки програмного продукту

2.3.3.1 Таблиця з графіком робіт

Для кожної підзадачі визначається виконавець, що фіксується у вигляді

таблиці, яка представлена в таблиці 2.3.3.1

WST	Дата начала	Дни	Дата окончания	Исполнитель
1.1	01.11.2020	3	03.11.2020	Яковенчук И.В.
1.2	01.11.2020	3	03.11.2020	Евтух А.А.
1.3	01.11.2020	3	03.11.2020	Карпухин Д.А.
2.1	04.11.2020	2	05.11.2020	Карпухин Д.А.
2.2	04.11.2020	2	05.11.2020	Яковенчук И.В.
2.3	04.11.2020	2	05.11.2020	Евтух А.А.
3.1	06.11.2020	3	08.11.2020	Евтух А.А.
3.2	06.11.2020	3	08.11.2020	Карпухин Д.А.
3.3	06.11.2020	3	08.11.2020	Яковенчук И.В.
4.1	09.11.2020	5	14.11.2020	Евтух А.А.
4.2	09.11.2020	5	14.11.2020	Яковенчук И.В.
4.3	09.11.2020	5	14.11.2020	Карпухин Д.А.
5.1	15.11.2020	3	19.11.2020	Яковенчук И.В.
5.2	15.11.2020	5	19.11.2020	Карпухин Д.А.
5.3	15.11.2020	5	19.11.2020	Евтух А.А.
5.4	19.11.2020	2	21.11.2020	Яковенчук И.В.
6.1	21.11.2020	1	21.11.2020	Карпухин Д.А.
6.2	21.11.2020	1	21.11.2020	Евтух А.А.
7	22.11.2020	2	23.11.2020	Яковенчук И.В.
8	22.11.2020	2	23.11.2020	Евтух А.А.
9	22.11.2020	3	24.11.2020	Карпухин Д.А.
10	24.11.2020	3	27.11.2020	Яковенчук И.В.
11.1	24.11.2020	3	27.11.2020	Евтух А.А.
11.2	25.11.2020	2	27.11.2020	Карпухин Д.А.

Рис. 2.3.3.1

2.3.3.2 Діаграма Ганта

Приклад діаграми Ганта рис 2.3.3.2



Рис 2.3.3.2

3. Проектування програмного продукту

3.1 Концептуальне та логічне проектування структур даних програмного продукту

3.1.1 Концептуальне проектування на основі UML-діаграми концептуальних класів

Використовуючи кроки основного успішного та альтернативного сценаріїв роботи прецедентів ПП, було спроектовано UML-діаграми концептуальних класів.

Моделювання проведено з урахуванням наступної послідовності кроків.

1. Визначити імена класів.
2. Визначити імена атрибутів класів.
3. Визначити зв'язку між класами (узагальнення, іменовані асоціації, агреговані асоціації).

4. Для зв'язків-асоціацій визначити назви, кратність і можливість агрегації.

5. Створити UML-діаграму класів в будь-якому графічному редакторі.

					ІС КР 122 АІ-181 ПЗ	

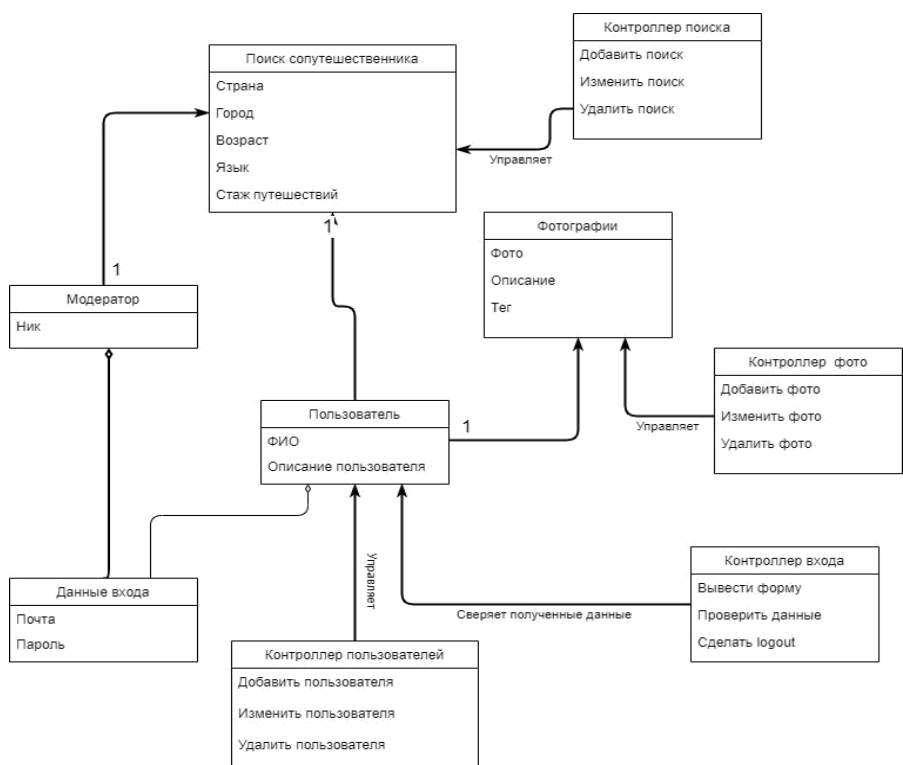


Рисунок - 3.1.1. UML-діаграма

класів

3.1.2 Логічне проектування структур даних

UML-діаграма концептуальних класів була перетворена в опис структур даних з використанням моделі, яка була обрана в концептуальному описі архітектури ПП.

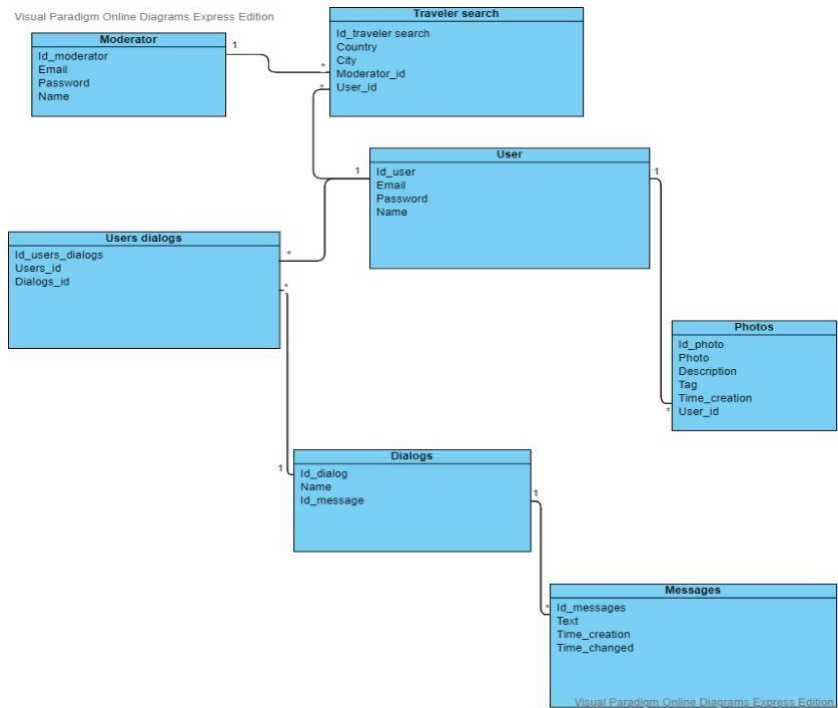


Рисунок – 3.1.2. Схема БД

					IC КР 122 АІ-181 ПЗ	

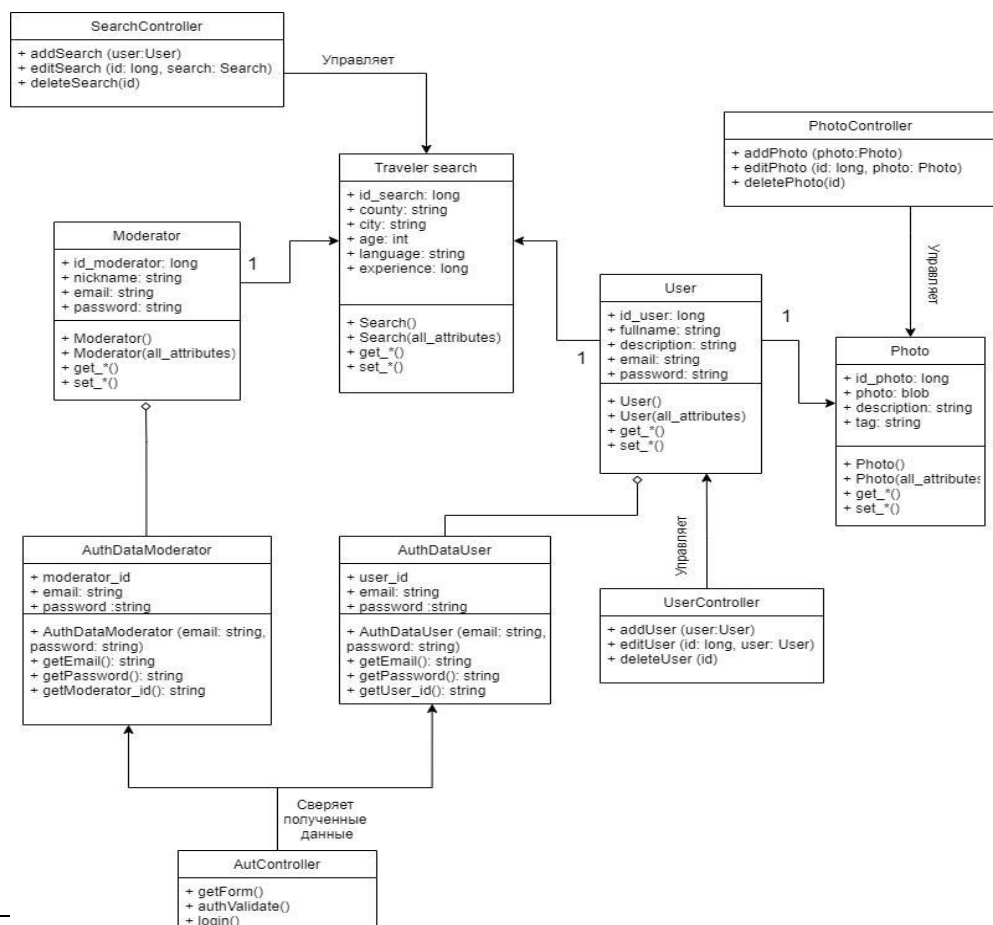
3.2 Проектування програмних класів

1. На основі UML-діаграми концептуальних класів

були спроектовані програмні класи:

- англійські або транслітерацію україномовних назви класів та їх атрибутів;
- абстрактні класи, їх класи-нащадки та інші класи;
- зв'язки між класами (наслідування, іменована асоціація, агрегатна асоціація, або агрегація, композитна асоціація або композиція) та їх кратності;
- атрибути класів с типами даних (цілий, дійсний, логічний, перелічуваний, символьний з урахуванням розміру), та типом видимості (публічний, захищений, приватний);
- методи-конструктори ініціалізації екземплярів об'єктів класу, set-методи та get-методи для доступу до атрибутів класу

2. На основі ієрархії функцій додайте до класів методи доступу, які будуть реалізовувати алгоритм роботи функцій.



3.3 Проектування алгоритмів роботи методів програмних класів

Процес розробки алгоритмів:

1. З усіх методів виділили ті, що мають операції доступу до БД або містять керуючі умови (if-then-else, while).
2. Опишіть алгоритм роботи у вигляді UML-діаграми активності: – кожен опис алгоритму представили в текстовому файлі на мові PlantUML, використовуючи веб-редактор.

Алгоритм роботи методу «User» мовою PlantUML:

```
@startuml
title addUser()
start
repeat
:Вывод экранной формы для регистрации пользователя;
:Ввод пользователем username, password, email и других персональных
данных;
:Проверка корректности введенных данных;
backward:Информирование пользователя об
ошибке; repeat while(Ошибка данных)
:Передача от пользователя параметров аутентификации в БД;
:Сохранение параметров аутентификации в БД;
:Информирование пользователя об успешной регистрации;
stop
@enduml
```

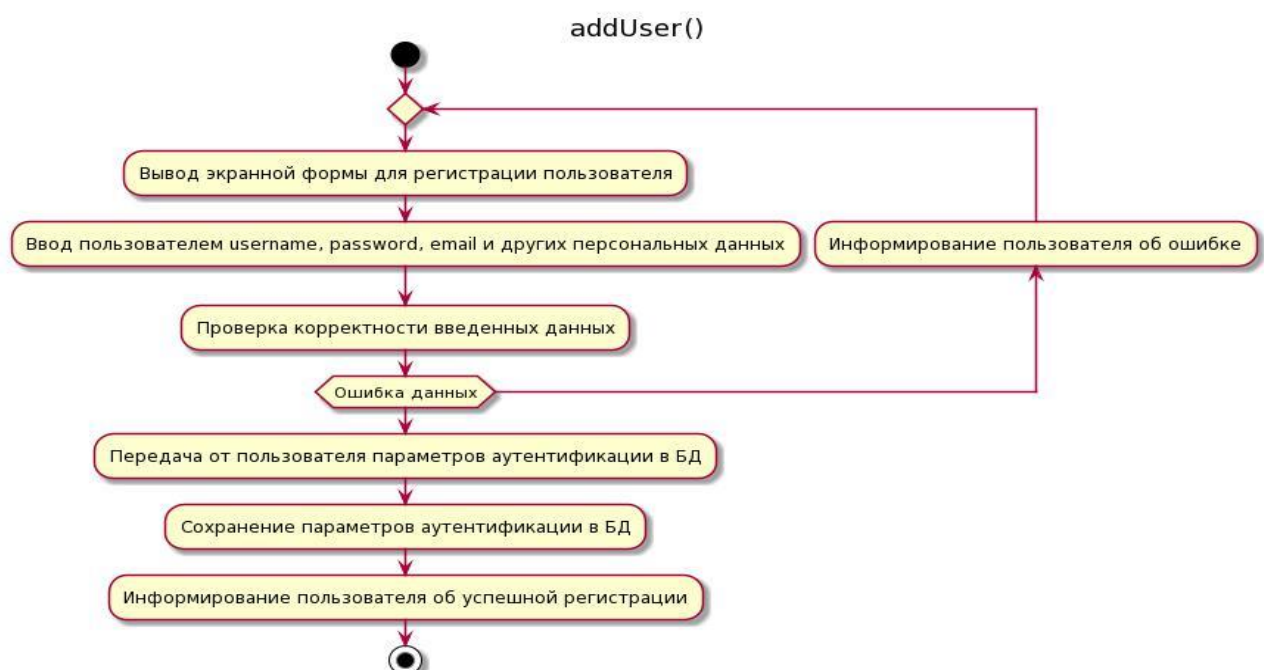


Рисунок – 3.3.1. Алгоритм методу

“addUser” Алгоритм роботи методу “authUser”:

@startuml

title authUser()

start

repeat

:Вывод экранной формы для авторизации

пользователя; :Ввод пользователем username, password;

:Передача от пользователя параметров аутентификации;

:Сравнение параметров аутентификации пользователя с имеющимися в БД; backward:Информирование пользователя об ошибке;

repeat while(Отсутствие совпадений параметров в БД)

:Получение параметров аутентификации

пользователя; :Вывод профиля авторизованного

пользователя; stop

@enduml

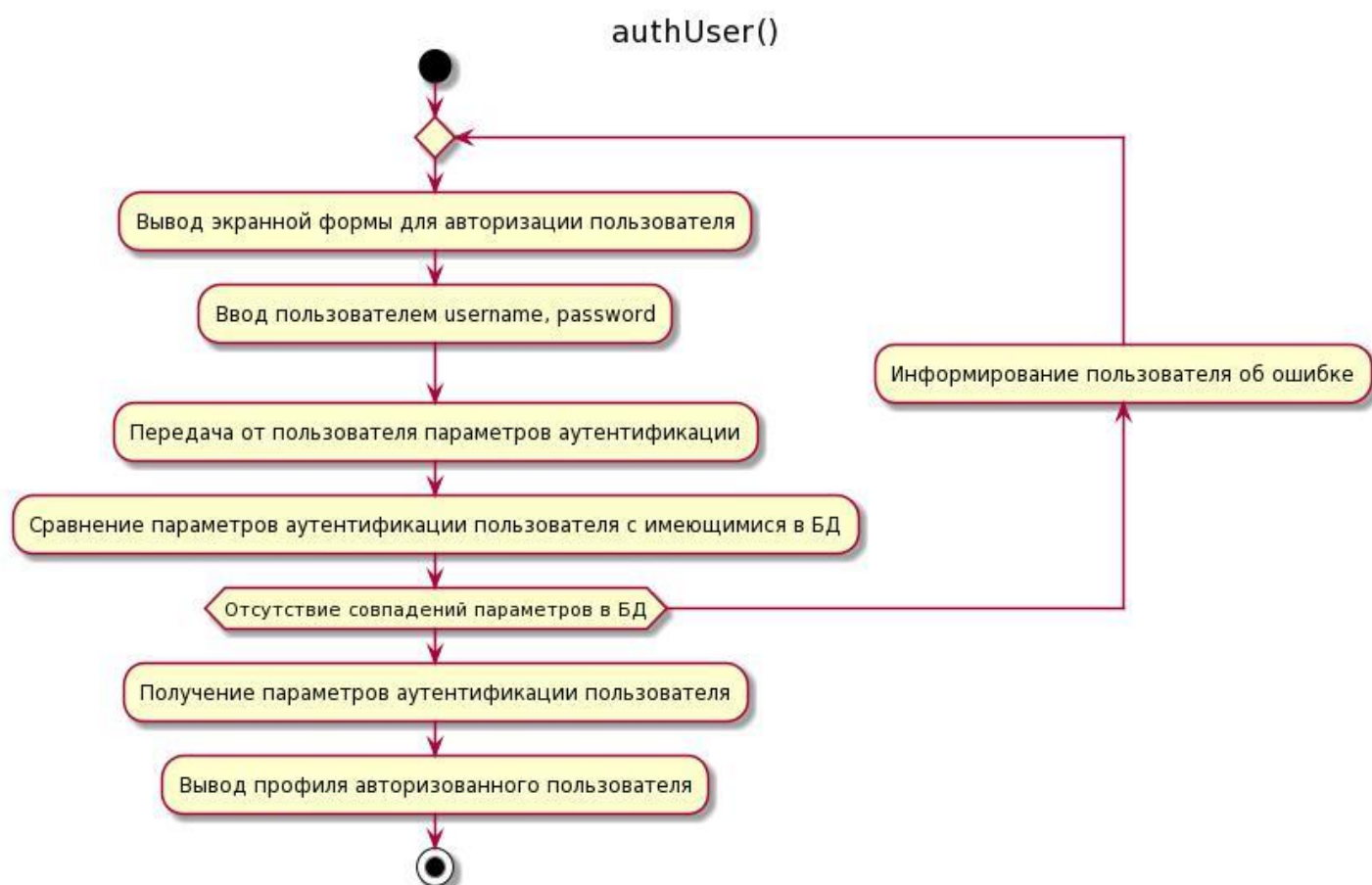


Рисунок – 3.3.2. Алгоритм методу “authUser”

					IC KP 122 AI-181 ПЗ	

Добавление публикации

@startuml

title addPost()

start

repeat

:Вывод экранной формы для добавления

публикации; :Загрузка пользователем контента;

:Добавление описания контента; :Редактирование

пользователем контента; :Проверка на соответствие

требованиям контента; backward:Информирование

пользователя об ошибке; repeat while(Несоответствие

требованиям контента) :Передача данных о контенте

в БД; :Сохранение контента в БД;

:Информирование пользователя об успешном добавлении публикации;

stop

@enduml

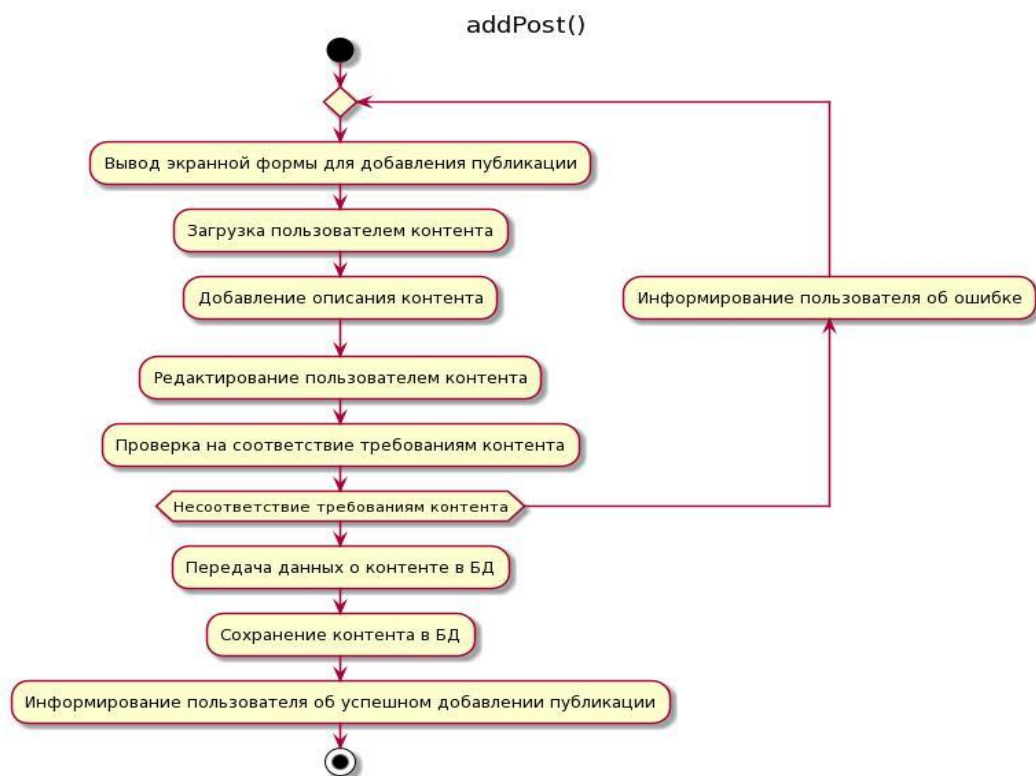


Рисунок – 3.3.3. Алгоритм методу “addPost”

Поиск спутешественника

@startuml

title findCompanion()

start

repeat

:Вывод экранной формы для поиска спутешественника; :Ввод

параметров поиска пользователя; :Передача от пользователя

параметров поиска;

					ИС КР 122 АИ-181 ПЗ	

```

:Сравнение параметров, введенных пользователем, с параметрами
всех пользователей в БД;
backward:Информирование пользователя об ошибке;
repeat while(Отсутствие совпадений с параметрами всех пользователей в БД)
:Вывод списка пользователей соответствующих введенным параметрам;
stop
@enduml

```

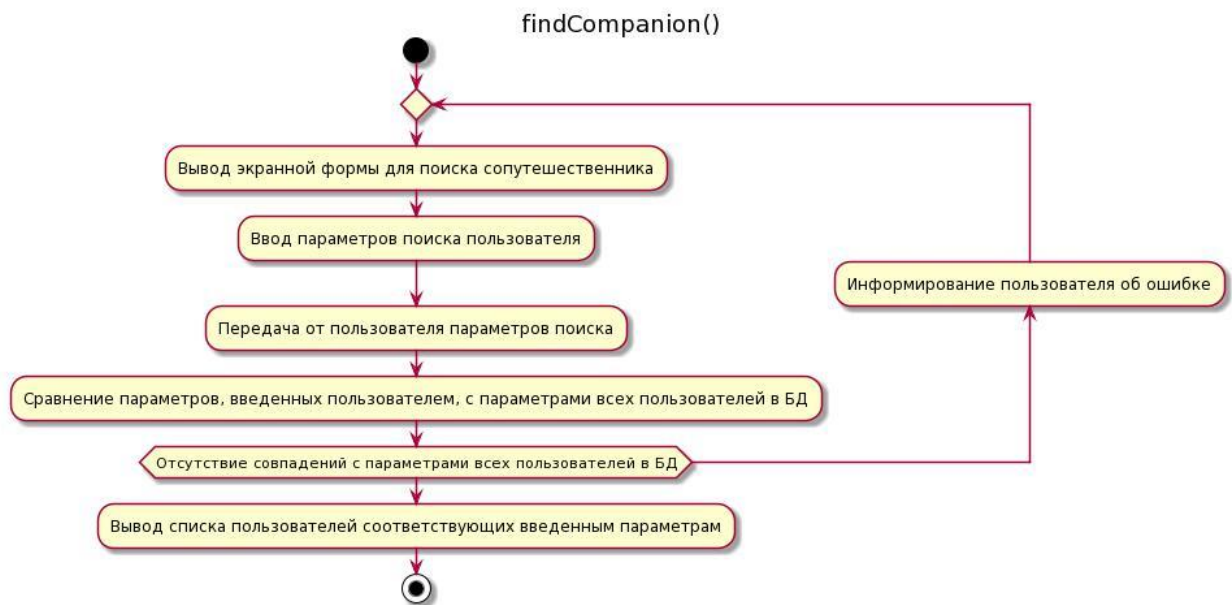


Рисунок – 3.3.4. Алгоритм методу “findCompanion”

Заполнение профиля

```

@startuml
title fillProfile()
start
repeat
:Вывод экранной формы для заполнение профиля;
:Ввод пользователем данных о себе; :Проверка
корректности введенных данных;
backward:Информирование пользователя об
ошибке; repeat while(Ошибка данных)
:Передача и сохранение данных в БД;
:Информирование пользователя об успешном заполнении профиля;
stop
@enduml

```

					ИС КР 122 АИ-181 ПЗ	

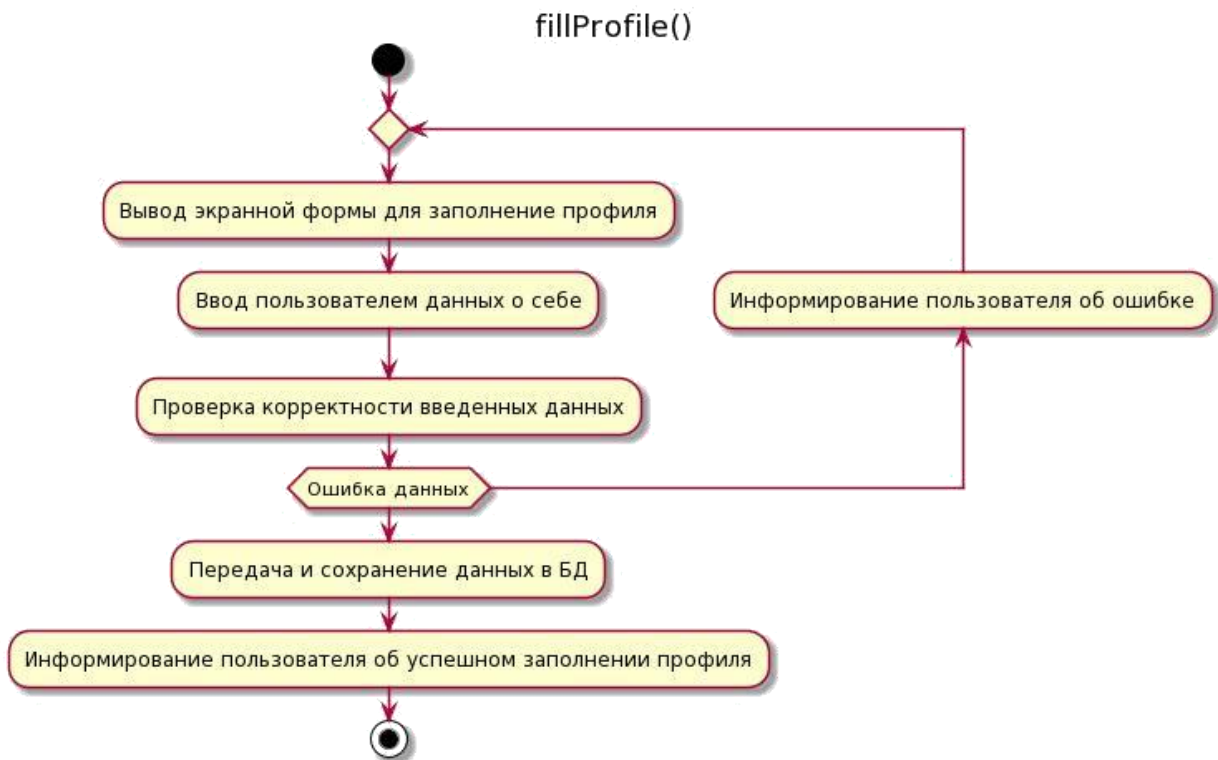


Рисунок – 3.3.5. Алгоритм методу “fillProfile”

Взаимодействие с постами других пользователей

@startuml

title interPost()

start

:Вывод экранной формы для просмотра публикации; split

:Поставить отметку "нравится";

split again

:Оставить комментарий о публикации; end split

:Передача и сохранение данных о взаимодействии БД;

:Информирование пользователя о успешном выполнении;

stop

@enduml

					ИС КР 122 АИ-181 ПЗ	



Рисунок – 3.3.6. Алгоритм методу “interPost”

Взаимодействие с другими пользователями

@startuml

title interUser()

start

:Вывод экранной формы для просмотра профиля
пользователя; split

:Написать сообщение;

split again

:Подписаться на пользователя;

end split

:Передача и сохранение данных о БД;

:Информирование об успешном взаимодействии;

stop

@enduml

					ИС КР 122 АИ-181 ПЗ	

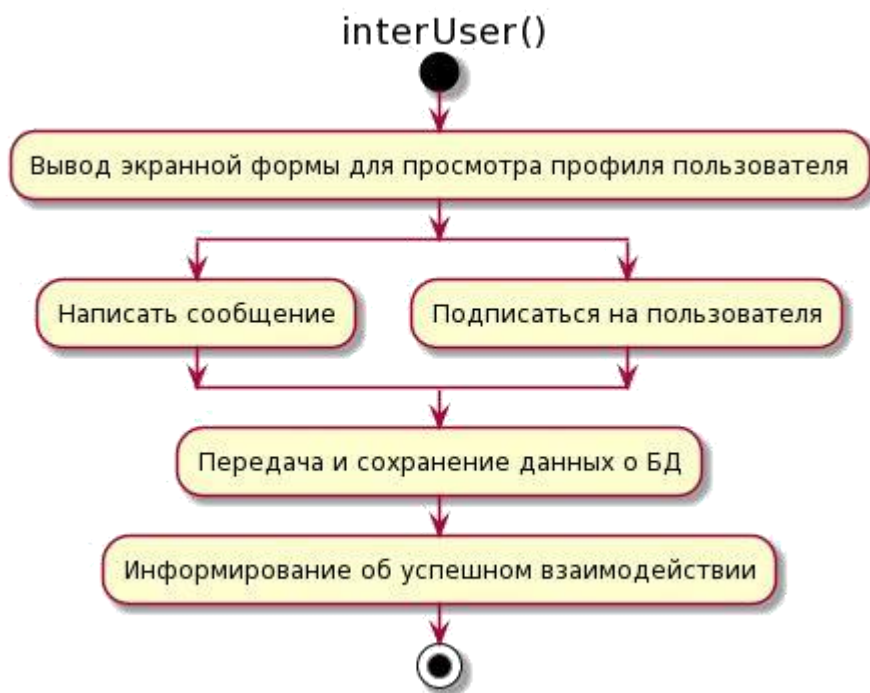


Рисунок – 3.3.7. Алгоритм методу “interUser”

Особое взаимодействие с постами других пользователей

@startuml

title specialInterPost()

start

:Вывод экранной формы для просмотра публикации; split

:Отредактировать публикацию;

split again

:Удалить публикацию;

end split

:Передача данных об изменении/удалении в БД;

:Информирование модератора об успешном выполнении;

stop

@enduml

					ИС КР 122 АИ-181 ПЗ	

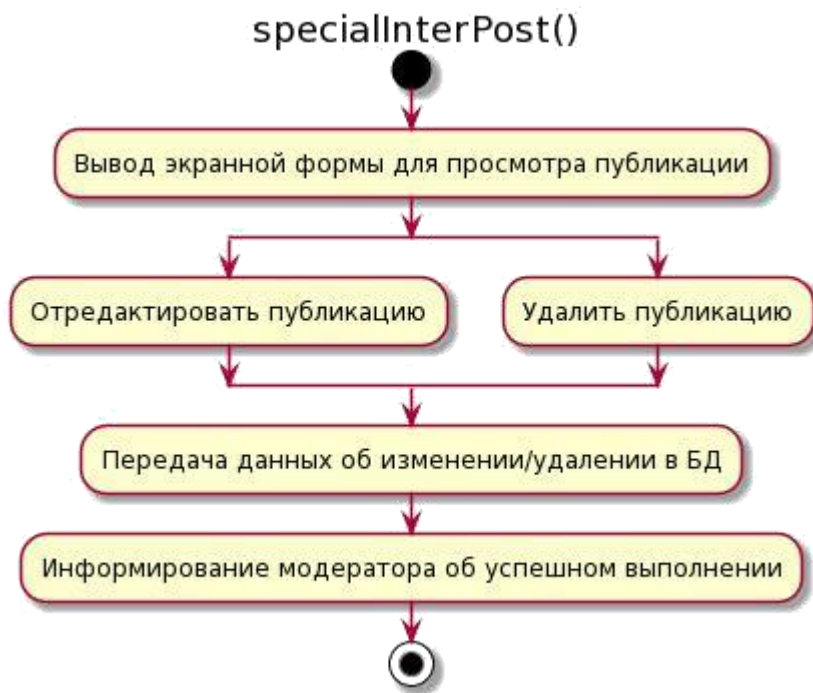


Рисунок – 3.3.8. Алгоритм метода “specialInter”

Особое взаимодействие с другими пользователями @startuml
 title specialInterUser()
 start
 :Вывод экранной формы для просмотра профиля пользователя; split
 :Отредактировать профиль;
 split again
 :Удалить профиль;
 end split
 :Передача данных об изменении/удалении в БД;
 :Информирование модератора об успешном выполнении;
 stop
 @enduml

					ИС КР 122 АИ-181 ПЗ	

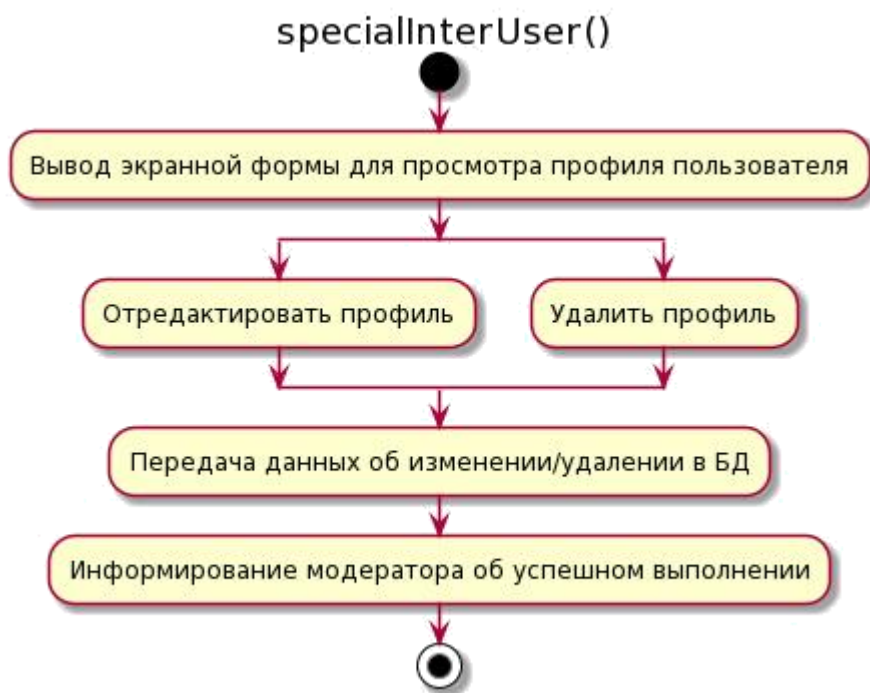


Рисунок – 3.3.9. Алгоритм методу “specialInterUser”

Просмотр статистики

@startuml

title viewStatistics()

start

:Вывод экранной формы для просмотра
статистики; :Выбор нужных данных для
просмотра; :Передача запроса к БД;

if (Данные отсутствуют) then (да)

:Информирование модератора об отсутствии данных;

else (нет)

:Вывод данных;

endif

stop

@enduml

					ИС КР 122 АИ-181 ПЗ	

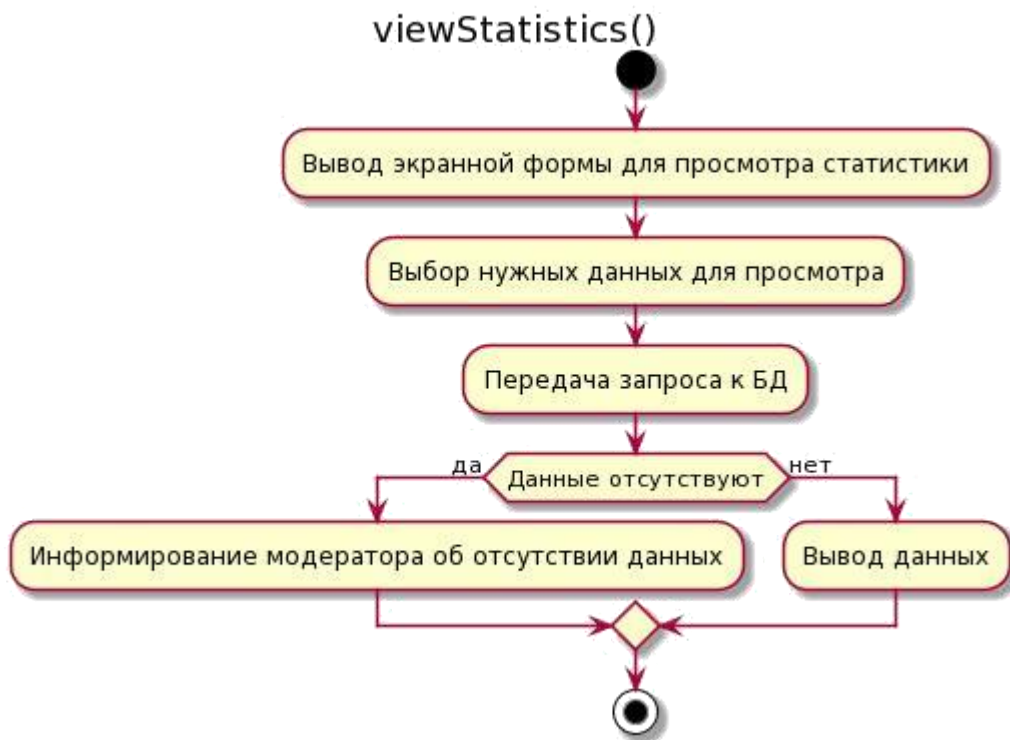


Рисунок – 3.3.10. Алгоритм методу “viewStatistics”

3.4 Проектування тестових наборів методів програмних клас

Регистрация/ авторизация

# Текста	Входные данные	Ожидаемый результат
1	user = user1 Email=Milo@milo.com password = 12345678	1) Строка з повідомленням "User added with user_id = 1" 2) не пуста відповідь на запит select * from users where user_id = 1;
2	user = user1 Email=Milo-milo.com password = 12345678	Неправильное заполнение поля Email Вывод ошибки- предложения перезаполнить
3	user = usefnfgnlsfnglsfkgnlsgnslkngslknglsdnglkdnglkdnglskdnglksr1 Email=Milo@milo.com password = 12345678	Некорректное кол-во символов в поле User, должно быть не больше 20
4	user = user1 Email=Milo@milo.com password = 12%;?(%45678	Пароль имеет недопустимые символы «№;%:?»

Добавление публикации

[illegible]

Поиск спутешественника

# Текста	Входные данные	Ожидаемый результат
1	City=Lviv Country=Ukraine Language=English Age=30 Experience=4	Запрос отправляется на модерацию, данные корректны
2	City=#hfh157 Country=Ukraine Language=English Age=25 Experience= 8	Неправильное заполнение поля City Вывод ошибки-предложения перезаполнить
3	City=Lviv Country=Ukrtttd;g;slmg;l;smdgl;m;m;lm;lm;m;;lm;lm;lm;lm;lml;mnkjjvjhvjvj vjhvkjbjhaine Language=English Age=40 Experience=2	Некорректное кол-во символов в поле Country, должно быть не больше 15
4	City=Odessa Country=France Language=English Age=27 Experience=1	Не совпадает название города с названием страны

					IC KP 122 AI-181 ПЗ	

Заполнение профиля

# Текста	Входные данные	Ожидаемый результат
1	Username=Travelinna Description= Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне.	Запрос отправляется на модерацию, данные корректны
2	Username=Travel\$^%*(&&inna Description= Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне.	Неправильное заполнение поля Username, введены некорректные символы
3	Username=Travelinna Description= Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне. Lorem Ipsum является стандартной "рыбой" для текстов на латинице с начала XVI века. В то время некий безымянный печатник создал большую коллекцию размеров и форм шрифтов, используя Lorem Ipsum для распечатки образцов.	Некорректное кол-во символов в поле Description, должно быть не больше 75 символов

Взаимодействие с другими пользователями

# Текста	Входные данные	Ожидаемый результат
1	Followed=Yes Message=Привет. Как поживаешь?	Запрос отправляется на модерацию, данные корректны
2	Followed=No Message=Привет. Как поживаешь?	Отсутствие подписки на пользователя, сообщение не отправляется
3	Followed=Yes Message=	Отсутствие кнопки отправки сообщения, невозможно отправить пустое сообщение

Взаимодействие с постами других пользователей

# Текста	Входные данные	Ожидаемый результат
1	Liked=Yes\No Comment= Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне. Lorem Ipsum является стандартной "рыбой" для текстов на латинице с начала XVI века. В то время некий безымянный печатник создал большую коллекцию размеров и форм шрифтов, используя Lorem Ipsum для распечатки образцов.	Запрос отправляется на модерацию, данные корректны
2	Liked=Yes\No Comment= Lorem Ipsum - это текст-"рыба", часто используемый в печати и вэб-дизайне. Lorem Ipsum является *****текстов на латинице с начала XVI века. В то время некий безымянный печатник создал большую коллекцию размеров и форм шрифтов, используя Lorem Ipsum для распечатки образцов.	Нецензурная лексика

					ИС КР 122 АИ-181 ПЗ	

4 Конструювання програмного продукту

4.1 Особливості конструювання структур даних

Архітектура програмного забезпечення – це представлення системи програмного забезпечення, яке дає інформацію про складові компоненти системи, про взаємозв'язки між цими компонентами і правила, що регламентують ці взаємозв'язки. Це процес, що передбачає послідовність дій для створення або зміни архітектури системи, і проекту системи по цій архітектурі, з врахуванням безлічі обмежень.

4.1.1 Особливості інсталяції та роботи з СУБД

Розглянуто створення бази даних для роботи рекламного агентства з використанням СУБД PostgreSQL.

PostgreSQL називають найбільш досконалою з наявних сьогодні СУБД з відкритим вихідним кодом. Таку репутацію вона завоювала завдяки зусиллям розробників протягом десятиліть. Будучи повнофункціональної реляційної СУБД з відкритим вихідним кодом, PostgreSQL володіє багатьма якостями, необхідними для підтримки найважливіших додатків з великим числом транзакцій.

Діаграма баз даних є структурою частиною системи баз даних описаною формальною мовою, яка підтримується системою керування баз даних і відноситься до організації даних з розподілом на таблиці.

4.1.2 Особливості створення структур даних

Рассмотрим SQL-запросы для создания таблицы пользователя:

```
CREATE TABLE User (  
id_User BIGINT PRIMARY KEY;  
fullname VARCHAR;  
description STRING;  
Email STRING;  
Password STRING)
```

					IC KP 122 AI-181 ПЗ	

Для хранения первичного ключа события используется тип данных BIGINT, потому что операции сравнения и поиска скорее всего работают с цифровыми типами.

База данных состоит из следующих таблиц: Посты, Пользователи, Сопутешественник, Модератор.

Таблица «Посты» содержит информацию о постах - фотографии, количество лайков. В ней присутствуют такие поля, как уникальный идентификатор постов, контент поста, его описание, и другая информация.

Таблица «Пользователи» содержит информацию о всех пользователях системы. В ней присутствуют такие поля, как уникальный идентификатор анкеты (первичный ключ), имя, фамилия, пароль и почта (имя пользователя) для входа в систему.

4.2 Особливості конструювання структур даних

Для реалізації структур, описаних у розділі вище, було обрано створення веб-застосунку, де і буде реалізовано структури даних за допомогою програмних класів.

4.2.1 Особливості роботи з інтегрованим середовищем розробки

Інформаційна система «Лапка допомоги» для допомоги притулкам представляє собою клієнт-серверне застосування.

Програмне забезпечення розроблялося у середовищі IntelliJ IDEA. Там було створено проект.

Для реалізації серверної частини спроектованої системи була обрана мова об'єктно-орієнтованого програмування Java із використанням платформи Android Studio.

Фронт-енд системи реалізований з використанням фреймворку Bootstrap, який включає в себе HTML- і CSS- шаблони оформлення різних компонентів веб-інтерфейсу.

					ІС КР 122 АІ-181 ПЗ	

Всі дані в створеній системі зберігаються в базі даних PostgreSQL, яка розгорнута на хмарної PaaS-платформі Heroku

4.2.2 Особливості створення програмної структури з урахуванням спеціалізованого Фреймворку

Приведем как пример класс для создания публикации.

```
package com.freecoders.photobook.classes;

{

import android.content.Context;
import android.content.res.TypedArray;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.TextView;

import com.freecoders.photobook.R;

import java.util.ArrayList;

/**
 * Created by aleksey.boev on 2015-03-
 * 25. */
public class BookmarkAdapter {
    private Context context;
    private String[] items;
    private TypedArray icons;
    private ViewGroup parentView;
    private View colorSelector;
    private OnItemSelectedListener listener;
    private int[] colors;
    public int selectedPosition = 0;

    public BookmarkAdapter(Context context, ViewGroup parentView, View colorSelector,
        String[] items, int iconsResID) {
        this.context = context;
        this.items = items;
        this.parentView = parentView;
        this.colorSelector = colorSelector;
        this.colors = context.getResources().getIntArray(R.array.bookmark_colors);
        this.icons = context.getResources().obtainTypedArray(iconsResID);
        redraw();
    }

    public void setParentView(ViewGroup parentView) {
        this.parentView = parentView;
        redraw();
    }

    private View getView(final int position) {
        String inflater = Context.LAYOUT_INFLATER_SERVICE;
        LayoutInflater vi = (LayoutInflater)context.getSystemService(inflater);
        View view;
        if (position == selectedPosition)
            view = vi.inflate(R.layout.item_tab_selected, parentView, false);
        else
            view = vi.inflate(R.layout.item_tab, parentView, false);
        TextView tv = (TextView)view.findViewById(R.id.txtViewBookmark);
        ImageView iv = (ImageView)view.findViewById(R.id.imgViewIcon);
        tv.setText(items[position]);
    }
}
```

					IC KP 122 AI-181 ПЗ	

```

        iv.setImageResource(icons.getResourceId(position % icons.length(), -1));

        view.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                setSelectedPosition(position);
            }
        });
        return view;
    }

    private void redraw(){
        if (parentView.getChildCount() > 0)
            parentView.removeAllViews();
        for (int i = 0; i < items.length; i++) {
            View view = getView(i);
            parentView.addView(view);
        }
    }

    public void setSelectedListener(onItemSelectedListener listener){
        this.listener = listener;
    }

    public interface onItemSelectedListener {
        public void onItemSelected(int position);
    }

    public String[] getItems() {
        return items;
    }

    public void setSelectedPosition(int position) {
        selectedPosition = position;
        redraw();
        if (listener != null)
            listener.onItemSelected(position);
    }
}

```

4.3 Тестування програмних модулів

Функції для тестування обиралися виходячи з найбільшої вірогідності виникнення в них помилки.

4.3.1 Тестування методу `getProfile()`

Специфікація функції `getProfile` класу `SignUp` має наступний опис.

Вхідні параметри відсутні.

Вихідні дані: текстовий рядок, що включає собою дані профілю

№ тесту	Опис значень початкових даних	Опис очікуваних значень результату
1	Поточні дані - <i>Profile = {[id_user=1, id_post=1, id_ptofile=1 [id_user=1=1 id_post =1, id_ptofile =1, quantity=1 [</i>	Видача профіля користувача

Опис тестових наборів для функції `getProfile`

					ІС КР 122 АІ-181 ПЗ	

Test Results	32 ms	"C:\Program Files\Java\jdk-15\bin\java.exe" ...
OrderTest	32 ms	Expected: dish1 (2 wr.), dish2 (1 wr.), dish3 (2 wr.)
getDishNames()	32 ms	Actual : dish1 (2 wr.), dish2 (1 wr.), dish3 (2 wr.)

№ тесту	Опис значень початкових даних	Опис очікуваних значень результату
2	Поточні дані – Country= Ukraine City= Kiev	User2, User5, User9

```
@Test
void addDetail() {
    Dish dish1 = new Dish( name: "dish1", price: 50);
    Dish dish2 = new Dish( name: "dish2", price: 100);
    Dish dish3 = new Dish( name: "dish3", price: 10);
    Details details1 = new Details(dish1, quantity: 2);
    Details details2 = new Details(dish2, quantity: 1);
    Details details3 = new Details(dish3, quantity: 2);
    Dish dish4 = new Dish( name: "dish4", price: 20);
    Details details4 = new Details(dish4, quantity: 3);
    Order order = new Order(new ArrayList<>(Arrays.asList(details1, details2, details3)));
    order.techAddDetail(details4);
    String actual = order.getDishNames();
    String expected = "dish1 (2 шт.), " + "dish2 (1 шт.), " + "dish3 (2 шт.), " + "dish4 (3 шт.)";
    assertEquals(expected, actual);
}
```

Test Results	137 ms	C:\Program Files\Java\jdk-13.0.1\bin\java.exe ...
OrderTest	137 ms	
addDetail()	137 ms	Actual value: dish1 (2 ут.), dish2 (1 ут.), dish3 (2 ут.), dish4 (3 ут.) Expected value: dish1 (2 ут.), dish2 (1 ут.), dish3 (2 ут.), dish4 (3 ут.)

					IC KP 122 AI-181 ПЗ	

Специфікація функції Delete класу User має наступний опис.

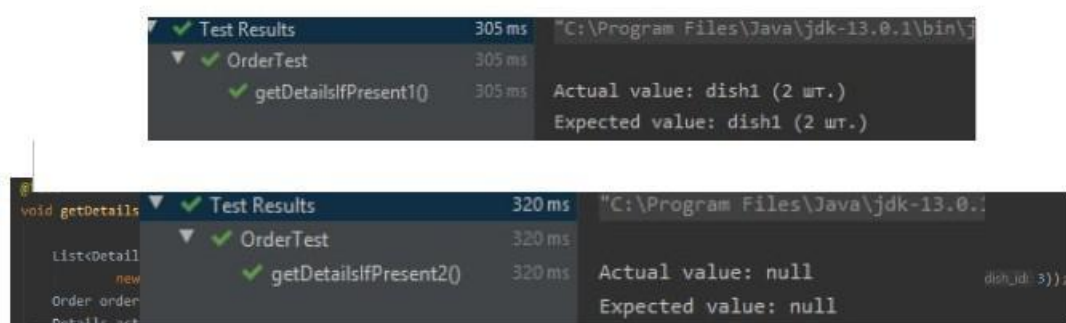
Вхідні параметри:

1) id – ідентифікатор Користувача, що запитується (ціле число). Значення, що повертаються функцією:

detail – об'єкт однойменного класу, що містить деталі

Профілю; null – якщо об'єкт з даним id не знайдено

№ тесту	Опис значень початкових даних	Опис очікуваних значень результату
4	Поточні дані - <code>details = {id_user=1, UserName=Shishkin, Email=LiliDog@ron.com, password:***** }</code> Вхідні дані: id = 1	Drop user1
5	Поточні дані - <code>details = {id_user=2, UserName=Shishkin, Email=LiliDog@ron.com, password:***** }</code> Вхідні дані: id = 1	null



5 Розгортання та валідація програмного продукту пояснювальної записки курсової роботи

5.1 Інструкція з встановлення програмного продукту

Розроблене програмне забезпечення підтримується усіма веб-браузерами, усіма версіями як на ОС Windows, Mac OS , так и на Linux ОС.

Здійснювати дії на веб-сервісі та користуватися ним користувач може за допомогою маніпулятора «миша» та клавіатури. За допомогою маніпулятора «миша» користувач може натиснути на кнопку/текст, а за допомогою клавіатури – вводити дані у різні поля та форми.

Система також має підтримуватися у всіх веб-браузерах мобільних пристроїв та усі дії будуть реалізовані користувачем за допомогою сенсора. Але, на жаль, адаптування мобільної версії ще не розроблено та вона виглядає так же

					ІС КР 122 АІ-181 ПЗ	

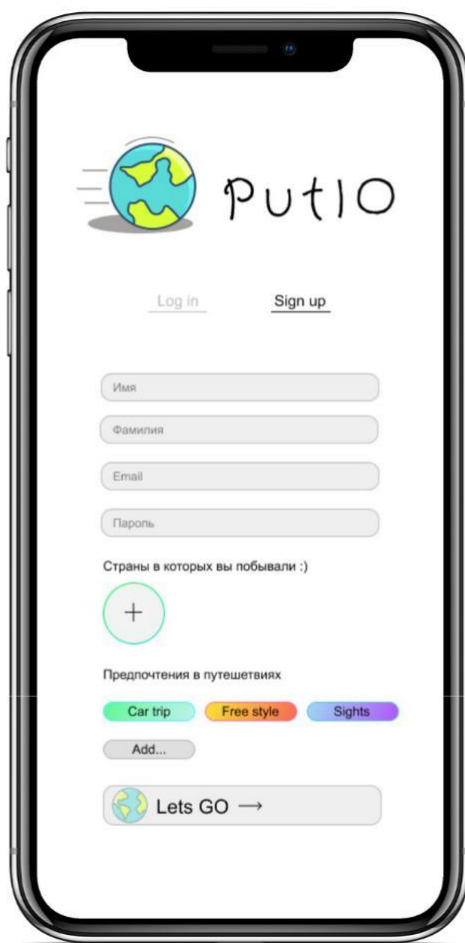
само, як і на десктопній версії, що є незручним до користувача. Але у подальшому адаптація для мобільних пристроїв теж буде розроблена.

5.2 Інструкція з використання програмного продукту

5.2.1 Реєстрація користувача

На рис. 5.2.1 зображена стартова сторінка соціальної мережі, з якої можна на вашу анкету.

Рис. 5.2.1



На рис. 5.2.2 показана главная страница, где можно просмотреть рекомендации, а также перейти к разделам – Профиль, Подписки, Поиск.

					ІС КР 122 АІ-181 ПЗ	

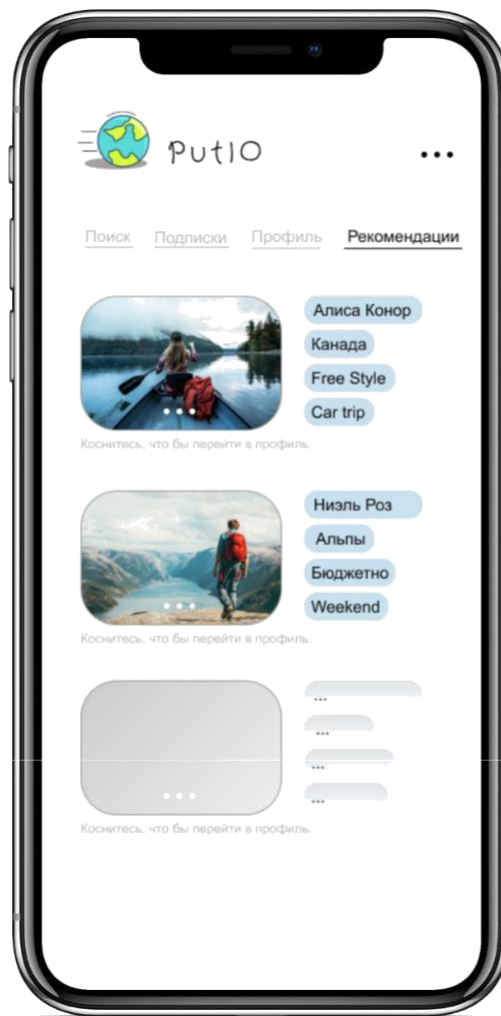


Рис. 5.2.2

На рис. 5.2.3 изображен раздел – Поиск, где можно выполнить поиск путешественника по параметрам.

					ИС КР 122 АИ-181 ПЗ	

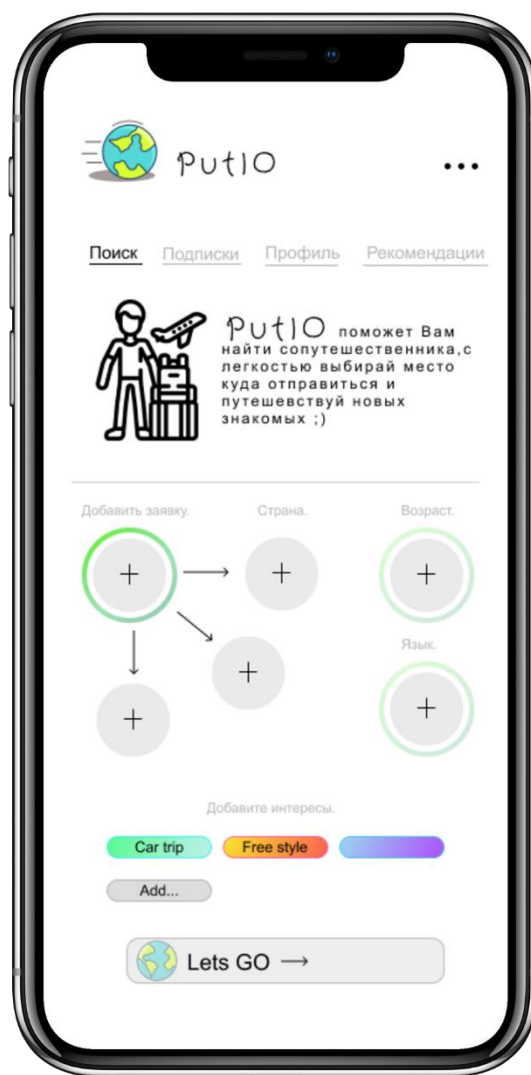


Рис. 5.2.3

					IC KP 122 AI-181 ПЗ	

ВИСНОВКИ

В результаті створення програмного продукту була досягнута наступна мета його споживача: «Підвищення рівня доступності вибору місця для подорожі і пошуку супутешественника на основі створення апаратно-програмного інтегратора функцій соціальної мережі для мандрівників».

Доказом цього є наступні факти. Програмний продукт «PutIO» об'єднує людей у мережі зі схожими інтересами задля перспективи сумісної подорожі.

«PutIO» задовольняє такі потреби споживача:

- 1) пошук супутешественника;
- 2) можливість комунікації з людьми з різних частин світу;
- 3) вибір місця для майбутньої подорожі;
- 4) можливість слідкувати за активністю інших користувачів.

В процесі створення програмного продукту виникли такі труднощі:

- 1) організаційні труднощі роботи у команді;
- 2) брак часу;
- 3) відсутність досвіду у front-end розробці.

Через вищеописані непередбачені труднощі, а також через обмежений час на створення програмного продукту, залишилися нереалізованими такі прецеденти або їх окремі кроки роботи:

- 1) розумний пошук по тегах, опорним словам;
- 2) можливість проведення прямого ефіру.

Зазначені недоробки планується реалізувати в майбутніх курсових роботах з урахуванням тем дисциплін наступних семестрів.

					ІС КР 122 АІ-181 ПЗ	

					IC KP 122 AI-183 ПЗ	57

					IC KP 122 AI-183 ПЗ	58