

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики



Розробка веб-застосунку на платформі ServiceNow
Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення»- 121

Керівник курсової роботи

Кандидат фізико-математичних наук,

доцент

Ющенко Ю. О.

(підпис)

“ ____ ” _____ 2021 р.

Виконала студентка ІПЗ-4:

Олійник Д.В.

“ ____ ” _____ 2021 р.

Київ 2021

Календарний план виконання роботи

Тема: Розробка веб-застосунку на платформі ServiceNow

№	Назва етапу	Термін виконання	Примітка
1.	Визначення теми курсової	26.10.2020	
2.	Пошук тематичної наукової літератури	10.11.2020	
3.	Ознайомлення з науковою літературою	25.12.2020	
4.	Ознайомлення з платформою ServiceNow	10.01.2021	
5.	Визначення теми веб-застосунку для реалізації	15.01.2021	
6.	Дослідження предметної області	25.01.2021	
7.	Визначення структури та необхідних даних для застосунку	03.02.2021	
8.	Реалізація практичної частини	10.03.2021	
9.	Написання текстової частини курсової роботи	01.04.2021	
10.	Написання висновків курсової роботи	09.04.2021	
13.	Створення презентації	12.04.2021	
14.	Захист роботи	19.04.2021	

Студент Олійник Д. В

Керівник Ющенко Ю. О.

“ ”

Зміст

Календарний план виконання роботи	1
Анотація	4
Вступ.....	5
Постановка задачі	6
Розділ 1: Опис хмарних технологій.....	7
1.1 Загальні відомості про хмарні платформ	7
1.2 Переваги та недоліки хмарних технологій	8
1.3 Види хмарних технологій	10
1.4 Рівні хмарних технологій.....	11
Розділ 2: Характеристика платформи ServiceNow	12
2.1 Опис компанії ServiceNow.....	12
2.2 Можливості, що реалізує платформа	12
2.2.1 Information Technology Service Management	13
2.2.2 Information Technology Business Management.....	14
2.2.3 Customer Service Management.....	15
2.2.4 Information Technology Asset Management	15
2.2.5 Human Resource Service Delivery.....	15
2.3 Початок роботи з платформою	16
2.4 Основний функціонал платформи.....	17
Розділ 3: Приклад створення веб-застосунку на платформі ServiceNow	19
3.1 Технічне завдання застосунку.....	19
3.1.1 Короткий опис	19
3.1.2 Актуальність теми.....	19
3.1.3 Опис предметної області.....	19
3.1.4 Основний функціонал застосунку.....	20
3.2 Опис розробки застосунку	21
3.2.1 Створення середовища розробки в ServiceNow	21
3.2.2 Налаштування відображення даних.....	23
3.2.3 Додання нових компонент для відображення даних	27
3.3 Опис результатів розробки.....	31

Висновок	37
Список літератури	38

Анотація

Метою роботи є дослідження та ознайомлення з хмарними технологіями, а саме моделі PaaS та підходу ITSM на прикладі платформи ServiceNow.

Було досліджено особливості та можливості, які надає платформа. Також було створено застосунок ресурсами ServiceNow.

Ключові слова: хмарні технології, PaaS, ITSM, платформа ServiceNow, автоматизація бізнес-процесів.

Вступ

Кожен день у світі з'являється безліч нової інформації, створюються нові компанії та виготовляються нові продукти. Тож перед людством постає дуже важлива задача організації даних для подальшого ефективного їх використання. Зокрема, кожна сучасна компанія прагне зробити бізнес-процеси, що відбуваються всередині, якомога простим та зрозумілим для співробітників. На сьогоднішній день існує багато варіантів вирішення таких завдань. Одним з них є використання хмарних технологій.

Останнім часом використання хмарних технологій стає все більш популярним. Це відбувається через те, що в такий спосіб користувач отримує необхідні йому технології та ресурси, але, при цьому, йому не доводиться турбуватися про те, як саме вони влаштовані та яку мають структуру. Прикладами компаній, що надають такі ресурси, є Google Cloud, AWS, Azure, ServiceNow.

Різниця серед даних компаній полягає в тому, які саме послуги вона надає. Наприклад платформа ServiceNow є одним із лідерів серед систем, що надають послуги для керування ІТ інфраструктурою для керування бізнесом (ITSM). Такий підхід дозволяє постійно підтримувати та оновлювати систему, що робить її актуальною.

Тож, враховуючи актуальність та стрімкий розвиток платформи ServiceNow, метою роботи є вивчення особливостей роботи з платформою та дослідження її можливостей.

Робота складається з трьох розділів.

Перший розділ присвячено опису хмарних технологій в цілому. Розглянуто різноманітні види хмарних платформ. Описані й рівні даних технологій. Крім того наведено перелік переваг та недоліків їх використання.

Другий розділ детально описує платформу ServiceNow, її основні можливості та переваги, завдяки яким дане середовище здобуло значну популярність та продовжує нарощувати кількість клієнтів. До того ж перелічено основний функціонал платформи.

Третій розділ містить в собі опис розробки веб-застосунку для ведення електронного шкільного щоденника з використанням можливостей платформи ServiceNow. Зазначено основні вимоги до додатку. Описано використані технології та наведено приклади їх використання.

Постановка задачі

Вивчення особливостей розробки за допомогою хмарних технологій на прикладі платформи ServiceNow. Дослідження можливостей вказаної платформи та реалізація веб-застосунку ресурсами ServiceNow.

Вимоги:

1. Надати опис хмарних технологій.
2. Надати визначення та описати різні види хмарних технологій.
3. Описати платформу ServiceNow.
4. Проаналізувати можливості платформи та описати інструменти, які вона надає.
5. Продемонструвати веб-застосунок реалізований на платформі ServiceNow.

Розділ 1: Опис хмарних технологій

1.1 Загальні відомості про хмарні платформ

Хмарні платформи – це технології, що надають інтернет-користувачам доступ до комп’ютерних ресурсів, які розміщуються в хмарі, тобто на віддаленому сервері.

В такий спосіб розробники мають можливість користуватися всіма необхідними послугами та зберігати дані віддалено, не турбуючись про створення базового функціоналу. Тобто при розробці нового додатку, клієнти відразу отримують більшу частину застосунку вже в готовому стані. Все це надає хмарним технологіям великий потенціал. [1]

На сьогоднішній день хмарні технології дуже стрімко розвиваються. З кожним роком все більше компаній починає використовувати подібні можливості. Так, наприклад, одна з провідних світових компаній сфері світових технологій – Gartner, в листопаді 2020 року опублікувала статтю, в якій описує, як збільшилося використання хмарних платформ та прогнозує подальший їх стрімкий розвиток.

	2019	2020	2021	2022
Cloud Business Process Services (BPaaS)	45,212	44,741	47,521	50,336
Cloud Application Infrastructure Services (PaaS)	37,512	43,823	55,486	68,964
Cloud Application Services (SaaS)	102,064	101,480	117,773	138,261
Cloud Management and Security Services	12,836	14,880	17,001	19,934
Cloud System Infrastructure Services (IaaS)	44,457	51,421	65,264	82,225
Desktop as a Service (DaaS)	616	1,204	1,945	2,542
Total Market	242,696	257,549	304,990	362,263

Рисунок 1.1 Прогноз витрат кінцевих користувачів на загальнодоступні хмарні послуги (мільйони доларів США) [2]

1.2 Переваги та недоліки хмарних технологій

Основні переваги:

1. Доступ до даних з будь-якої точки планети

Оскільки дані зберігаються віддалено, користувачі мають можливість взаємодіяти з ними з будь-якої точки світу. Це дозволяє успішно проводити розробку продукту, навіть, якщо учасники команди знаходяться в різних країнах.

2. Економія коштів

Так як хмарна платформа надає ряд ресурсів, які він може використовувати та засоби для зберігання даних, клієнту не доводиться платити за всі ці можливості окремо. До того ж не потрібно наймати окремих співробітників, які б займалися постійною підтримкою цих програм.

3. Швидке відновлення даних

Хмарні платформи надають можливість резервного зберігання даних. Тож користувачам не доведеться турбуватися про втрату даних.

4. Безпека

Сучасні хмарні платформи надають можливість обмінюватися даними за найвищими стандартами безпеки. Тому цю важливу складову свого застосунку клієнти також можуть довірити хмарним технологіям.

5. Легка масштабованість системи

Якщо користувач планує розширювати або, навпаки, зменшувати масштаби свого застосунку, то все, що йому потрібно для цього знати – вказати необхідну кількість пам'яті.

Недоліки:

1. Складність при переміщенні даних

Якщо з'явилася необхідність у використанні іншої хмарної платформи, то перемістити всі свої дані буде досить проблематично, особливо, якщо застосунок достатньо великий.

2. Необхідний постійний доступ до інтернету

Так як всі дані та ресурси розміщені в хмарі, користувачам потрібно завжди мати можливість підключитися до інтернету. До того ж якщо якість з'єднання невисока, то й доступ до даних буде проходити повільніше.

3. Відсутність повного контролю над застосунком

Значна частина ресурсів надається саме платформою, тому клієнт не має уявлення як саме відбуваються всі процеси всередині. Хмарні технології надають можливість лише налаштовувати процеси на свій смак.

Зважаючи на всі перераховані переваги та недоліки, можна зробити висновок, що хмарні технології задовільняють не всі види компаній. Однак для більшості можливих варіантів, вони стануть найкращим рішенням та допоможуть заощадити час та кошти.

1.3 Види хмарних технологій

Публічна хмара

Публічна хмара надає доступ до ресурсів багатьом підприємствам використовуючи модель pay-per-use (оплата за використання). В такий спосіб користувачі отримують необхідні інфраструктури та послуги та заощаджує на їх розробці та підтримці.

Приватна хмара

Приватна хмара – інфраструктура, яка належить одному бізнесу. В такий спосіб компанії надається більше контролю над ресурсами, що, в свою чергу більш налаштовані саме на потреби конкретного бізнесу. Зазвичай такий вид хмари більш дорожчий, однак він пропонує вищий рівень безпеки та більше можливостей для налаштування системи.

Суспільна хмара

Така інфраструктура призначена для використання декількома компаніями, що мають спільні або схожі задачі для вирішення. Відповідно доступ для конфігурацій та управління також може надаватися кільком організаціям.

Гібридна хмара

Гібридна хмара являє собою суміщення двох або більше інших видів. Тобто надає більше можливостей для конфігурації, однак все ще багато технологій залишаються стандартизованими.

1.4 Рівні хмарних технологій

IaaS (Infrastructure-as-a-Service) – вважається найнижчим рівнем хмарних обчислень. Ця модель обслуговування хмарних технологій надає базові ресурси для реалізації власних потреб. Даний рівень найбільше підходить невеликим компаніям, що шукають ефективне рішення для свого бізнесу, яке буде до того ще й економічним.

PaaS (Platform-as-a-Service) – наступний рівень, який, крім фундаментальної інфраструктури, надає ще й можливість завантажувати власні програми. Також користувачам надається доступ для налаштувань, що робить систему гнучкою та надійною.

SaaS (System-as-a-Service) – найвищий рівень хмарних обчислень, що через інтернет надає готове рішення для різних організацій. Користувачу не потрібно турбуватися про підтримку такого застосунку, однак надається можливість налаштовувати лише обмежену кількість конфігурацій.

Розділ 2: Характеристика платформи ServiceNow

2.1 Опис компанії ServiceNow

ServiceNow – хмарна платформа, що надає можливість зробити роботу організацій більш ефективною шляхом структурування, впорядкування та автоматизації різноманітних робочих процесів.

Компанія ServiceNow була заснована в 2004 році Фредом Ладді (Fred Luddy). На даний момент із ServiceNow співпрацює близько 6900 клієнтів, серед них й такі компанії як 7-Eleven, Ricoh, IBM. Акції платформи постійно зростають, а в 2018 році видання Forbes назвали ServiceNow найбільш інноваційною компанією світу [7].

Спочатку ServiceNow вважалася SaaS платформою, згодом почала надавати послуги як PaaS платформа, що надає можливості для обробки даних, їх зберігання, та інтеграції власних додатків. Платформа слідує ITIL (Information Technology Infrastructure Library) та надає ITSM (Information Technology Service Management), що дозволяє ServiceNow підтримувати всі необхідні бізнес процеси від обслуговування користувачів до управління персоналом.

Платформа активно розвивається, та постійно додає новий функціонал. Кожні півроку (зазвичай в березні та вересні) з'являється оновлення, на даний момент Quebec являється останньою версією.

2.2 Можливості, що реалізує платформа

Як вже було зазначено раніше ServiceNow слідує бібліотеці ITIL. IT Infrastructure Library – це набір з різноманітних практик та процесів, що дозволяють найкращим способом організувати роботу компаній, що задіяні в IT індустрії. Бібліотека описує процеси для забезпечення проектування, експлуатації застосунку та переходу між двома раніше вказаними станами. Це

відбувається за допомогою управління такими службами, як: обслуговування клієнтів, управління персоналом, доступність даних, рівень обслуговування, активами, конфігураціями, інцидентами.

Платформа, в свою чергу, надає інструменти для реалізації всіх необхідних процесів базуючись на підході ITSM.

2.2.1 Information Technology Service Management

ITSM – підхід до управління, що спрямований на надання ІТ-послуг для реалізації потреб компанії. ITSM включає в себе проектування, створення, надання та підтримку ІТ-послуг. Цей підхід дозволяє керувати процесами, технологіями та людьми для досягнення найбільшої ефективності. ITSM передбачає заощадження коштів шляхом реалізації передбачуваної ІТ організації та впровадження рішень, що допомагають у прийнятті рішень.

Найбільшої ефективності система ITSM досягає завдяки наступним принципам та процесам [9]:

1. Управління інцидентами -- система відстежує та оброблює інциденти (відхилення у нормальному режимі роботи) і запити на нові послуги, обладнання або додатки, щоб якнайшвидше забезпечити клієнтів всім необхідним. Також є можливість розставити пріоритети для того, щоб персонал міг зосередити свої сили на найбільш важливих задачах.
2. Управління проблемами -- відстеження та виявлення причин виникнення інцидентів, задля подальшого їх усунення, з метою уникнення появи повторюваних та схожих відхилень у роботі системи.
3. Контроль змін та оновлень -- відстеження заплановані зміни для мінімізації ризиків та помилок та з метою впровадження нового функціоналу послідовно та якомога швидше.
4. Керування запитами -- управління запитами на обслуговування для можливості відстеження їх виконання.

5. Контроль налаштувань – забезпечення підтримки та контролю над всіма необхідними налаштуваннями обладнання, програмного забезпечення та важливими даними про персонал та документацію. Це надає команді чітку інформацію про всі важливі взаємозв'язки в системі.
6. Постійне вдосконалення – поступове впровадження покращень у систему для відстеження їх ефективності, що дозволяє постійно підвищувати якість обслуговування.
7. Управління талантами – призначення людей на посади, що найкраще відповідають їх навичкам.

2.2.2 Information Technology Business Management

Управління IT-бізнесом (ITBM) – набір інструментів та процесів, для розвитку, надання та управління бізнес-рішеннями орієнтованими на IT. ITBM дозволяє передбачити потреби бізнесу та оптимально розподілити ресурси.

Функції ITBM [10]:

1. Управління додатками – проведення комплексного аналізу всіх додатків, що використовуються, для ефективного управління витратами та легкої адаптації до нових змін.
2. Керування «портфоліо» проекту – комплексне рішення, що надає можливість керувати повним життєвим циклом проекту: від зародження ідеї до її реалізації.
3. Управління інноваціями – дозволяє відстежувати ідеї та пропозиції команди і перетворювати їх на критерії та вимоги до проекту.
4. Спрощення процесу фінансування
5. Контроль процесу розробки програмного забезпечення – управління розробки та тестування проекту за допомогою таких методологій як scrum, hybrid, waterfall.

2.2.3 Customer Service Management

Управління обслуговуванням клієнтів (CSM) – передбачає вирішення проблем клієнтів та миттєву обробку питань користувачів.

Зокрема ServiceNow надає можливості для налаштування різноманітних інструментів для ефективної взаємодії з клієнтом, що дозволяють у режимі самообслуговування вирішувати різноманітні питання, наприклад:

- ініціювати автоматизоване рішення загальних запитів, таких як відновлення паролю та зміна адреси;
- створювати запити та отримувати відповіді використовуючи чат-бота;
- знаходити рішення за допомогою експертів;
- шукати відповіді використовуючи інформацію, що надає система;
- переглядати актуальний стан придбаних продуктів та рішень.

2.2.4 Information Technology Asset Management

Управління активами (ITAM) – процеси, що зосереджені на відстеженні фінансового стану компанії. Метою ITAM є [11]:

- контроль інструментів, що використовуються та закуповуються;
- зниження витрат на закупівлю та управління активів;
- вибір ефективних інструментів для управління активами;
- контроль життєвого циклу кожного активу;
- покращення послуг для кінцевого користувача;
- створення стандартів для управління активами.

2.2.5 Human Resource Service Delivery

Управління людськими ресурсами (HRSM) – набір інструментів метою якого є покращення продуктивності роботи працівників шляхом застосування

цифрових процесів. Шляхом надання працівникам місця, де вони можуть вказати свої потреби та труднощі, з якими вони стикаються, роботодавці мають можливість забезпечити їх всім необхідним.

Основні можливості HRSM [12]:

- організація, автоматизація та відновлення робочих місць у разі виникнення надзвичайної ситуації;
- спрощення доступу до HR послуг для працівників;
- ефективна організація співпраці робітників;
- заміна ручної праці на цифрові процеси;
- створення комфортних умов для працівників за допомогою використання новітніх технологій.

2.3 Початок роботи з платформою

Почати роботу з платформою досить просто: достатньо лише відкрити головну сторінку (<https://www.servicenow.com>) та зареєструватися. Далі в пункті меню «Розробка застосунку» необхідно створити персональний екземпляр розробника (PDI). Personal Developer Instance надає безкоштовний доступ до всіх можливостей платформи на рівні адміністратора. Отримавши своє особисте посилання (тобто PDI) та перейшовши за ним, користувач потрапляє до середовища, де йому надається можливість для створення власного застосунку та вивчення можливостей платформи. Крім того платформа надає свій набір даних та різноманітних плагінів, які можна використовувати.

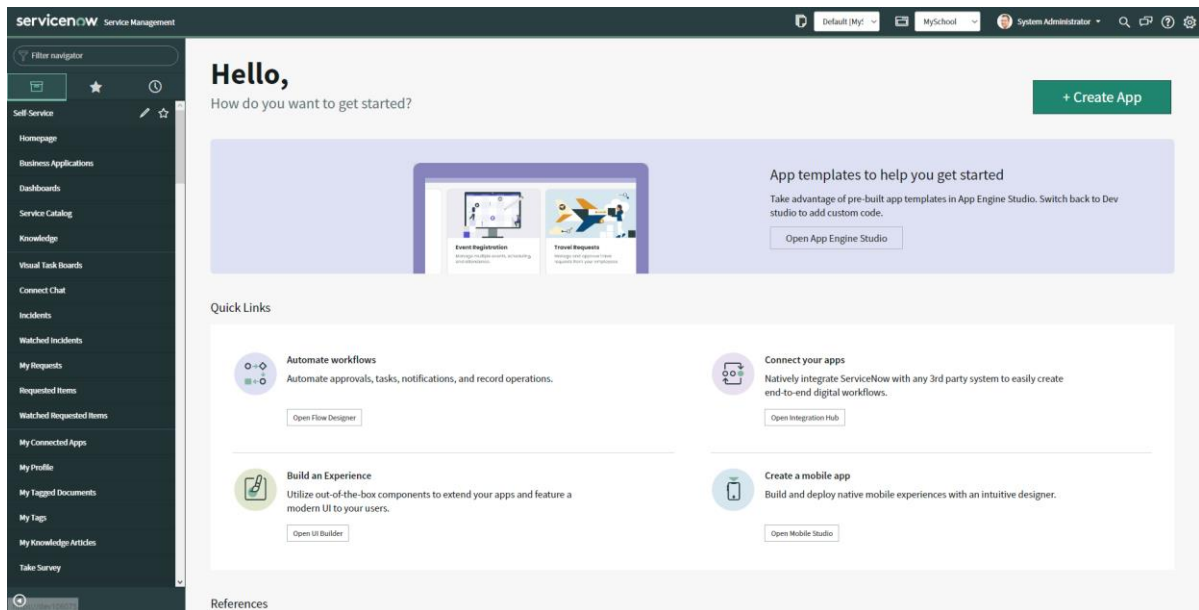


Рисунок 2.1 Вигляд PDI

На перший погляд набір всіх інструментів в меню може здаватися занадто складним, однак користувачам платформи надаються різні курси та навчальні плани для вивчення платформи та її можливостей. Проте, варто зазначити, що для зареєстрованих користувачів доступна функція відслідковування прогресу та контролю пройденого матеріалу.

Також платформа надає доступ до розгорнутої документації всіх інструментів ServiceNow, де можна знайти й рекомендації щодо їх використання. Якщо ж у користувача, залишаться питання, які він не знайшов в документації, то він може звернутися до Now Community – середовище для ServiceNow розробників для обговорення особливостей роботи з платформою.

Крім того ServiceNow регулярно організовує різноманітні події, такі як зустрічі, воркшопи та хакатони, де розробники можуть спробувати свої навички, перевірити знання та дізнатися щось нове про платформу.

2.4 Основний функціонал платформи

ServiceNow включає в себе набір різноманітних інструментів, що дозволяють забезпечити застосунок всіма необхідними технологіями. Так, наприклад, платформа надає доступ до наступних можливостей:

1. Now Experience UI Framework – фреймворк, розроблений ServiceNow для створення власних веб-компонент.
2. Flow Designer – надає можливість для автоматизації бізнес-процесів.
3. Mobile Studio – забезпечує зручну розробку та створення мобільних застосунків.
4. Guided App Creator – простий інструмент для створення даних, що будуть використовуватися у застосунку, та визначення взаємозв'язків між ними.
5. Performance Analytics and Reporting – дозволяє легко створювати всі необхідні графіки та звіти відповідно до вказаних даних.
6. Now Experience Components – веб-компоненти, розроблені командою ServiceNow, які можна використовувати під час дизайну та розробки власного продукту.
7. Notifications – надає можливість для комунікації між користувачем та застосунком.
8. Studio IDE – середовище розробки для створення та контролю над власними продуктами.
9. UI Builder – інструмент, що дозволяє зручно створювати дизайн будь-якої зі сторінок власного застосунку. Містить багато різноманітних елементів для відображення даних

Вище було зазначено основні інструменти, які надає платформа, однак існує ще й багато інших корисних функцій. Детальніше про них можна дізнатися вивчаючи документацію та завдяки різноманітним курсам, які надає ServiceNow.

Розділ 3: Приклад створення веб-застосунку на платформі ServiceNow

3.1 Технічне завдання застосунку

3.1.1 Короткий опис

Для демонстрації створення застосунку на платформі ServiceNow було обрано тему «Електронний шкільний щоденник». Що передбачає реалізацію додатку, який надає змогу учням переглядати розклад, власні оцінки, завдання на урок, та отримувати відгуки на свою роботу від вчителів.

3.1.2 Актуальність теми

Щоденник – це невід’ємна частина життя кожного учня. Він є ресурсом інформації, що являється запорукою успішності дитини у навчанні. Однак, зважаючи на різноманітні причини учням не завжди вдається заносити в щоденник всі важливі дані. Саме тому такий застосунок, як «електронний шкільний щоденник» зможе стати гарним інструментом для зручності контролю навчального процесу та мінімізації проблем, що виникають через відсутність або невідповідність даних у щоденнику.

3.1.3 Опис предметної області

Під час навчання у школі, кожен учень веде свій щоденник. В ньому дитина записує розклад занять на тиждень та домашні завдання до кожного предмету. Також щоденник слугує ресурсом для відслідковування успішності, оскільки в нього записують оцінки учня.

Розклад складається директором та являє собою список предметів, що мають відбутися в певний день та певний час. Час для проведення занять завжди однаковий – 45 хвилин. Початок уроків та перерви між ними також завжди відбуваються в однаковий час.

Предмети можуть різні для різних класів (в залежності від профілю та року навчання). Кожен предмет складається з різних тем, що також можуть відрізнятися в залежності від вчителя, або профілю класу. З кожної теми має бути проведена певна кількість уроків. В кінці проходження теми кожен учень, що її прослухав отримує тематичну оцінку, що обраховується з оцінок, отриманих під час роботи на уроці або за різноманітні завдання. Оцінки можуть мати різну вагу, тобто по-різному впливати на фінальний результат.

Як було зазначено, раніше учень має можливість отримати оцінки за певні завдання або за роботу на уроці. Кожне таке завдання ставиться вчителем на уроці та має певний дедлайн, наприклад: наступне заняття. Крім того під час заняття учень може отримати відгук від вчителя, що також заноситься до щоденника.

3.1.4 Основний функціонал застосунку

Передбачається, що застосунок матиме три ролі користувачів:

- 1) Директор. Має доступ до всіх конфігурацій проекту. Лише користувач, із зазначеною роллю може створювати нові предмети, додавати нових учнів та вчителів, формувати класи та змінювати розклад.
- 2) Викладач. Має можливість створювати нові уроки, теми до них, а також додавати нові завдання та відгуки.
- 3) Учні. Можуть переглядати інформацію про розклад на тиждень, свої завдання та відгуки на свою роботу. Змінення будь-яких даних користувачами, що мають роль «Учень» не передбачено.

3.2 Опис розробки застосунку

3.2.1 Створення середовища розробки в ServiceNow

Після отримання власного посилання для розробки, тобто інстансу – <https://dev106075.service-now.com>. Платформа одразу пропонує розпочати створення власного застосунку за допомогою App Engine Studio. Цей інструмент дозволяє створити всі необхідні таблиці, а також вказати ролі користувачів в застосунку.

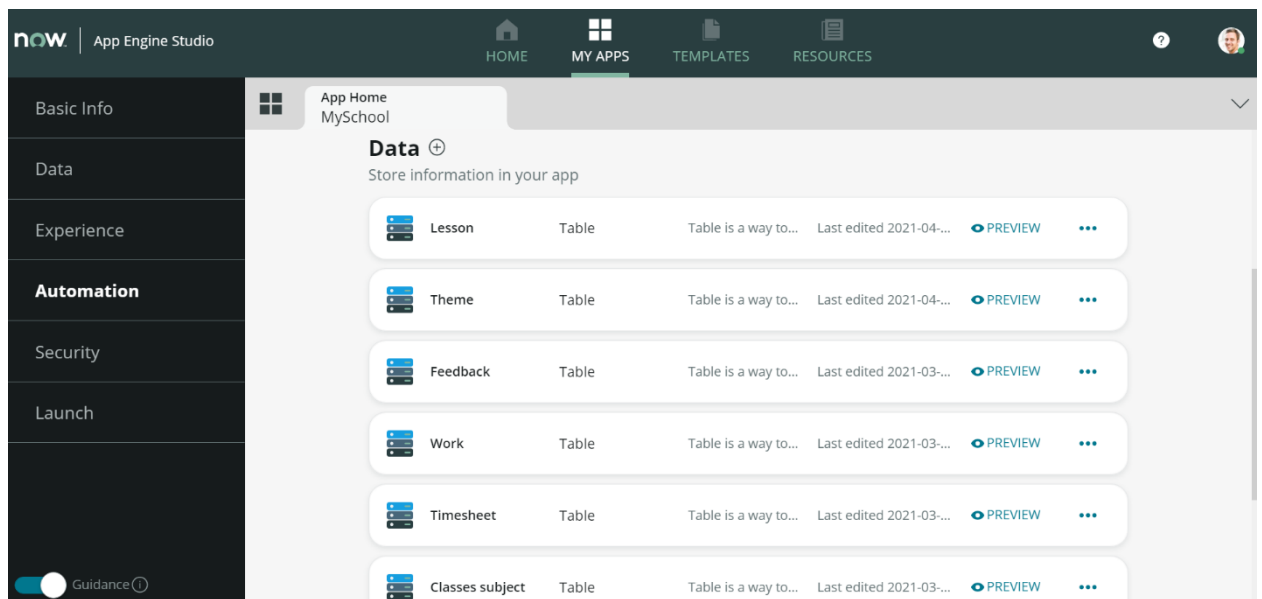


Рисунок 3.1 Створення таблиць в App Engine Studio

Таблиці було створення відповідно до попередньо розробленої ER-моделі даних. Однак, враховуючи особливості розробки на платформі, до кожної з створених таблиць було додано поля, що описують дату створення, коли було модифіковано в останнє, ким було змінено та ким створено. До того ж, варто зазначити, що в ServiceNow кожен екземпляр з таблиці має свій унікальний номер – `sys_id`, тому в якості ключа також було використано дане поле.

Платформа надає можливість користуватися стандартними таблицями, наприклад – `sys_user`, що містить всі необхідні поля для користувача. Тому замість створення нових сутностей для таких таблиць, як учні та вчителі було

використано стандартну таблицю користувачів. Натомість, було створено спеціальні групи користувачів такі як Teachers та Schoolchlds для заміни таблиць вчителі та учні з ER-моделі відповідно.

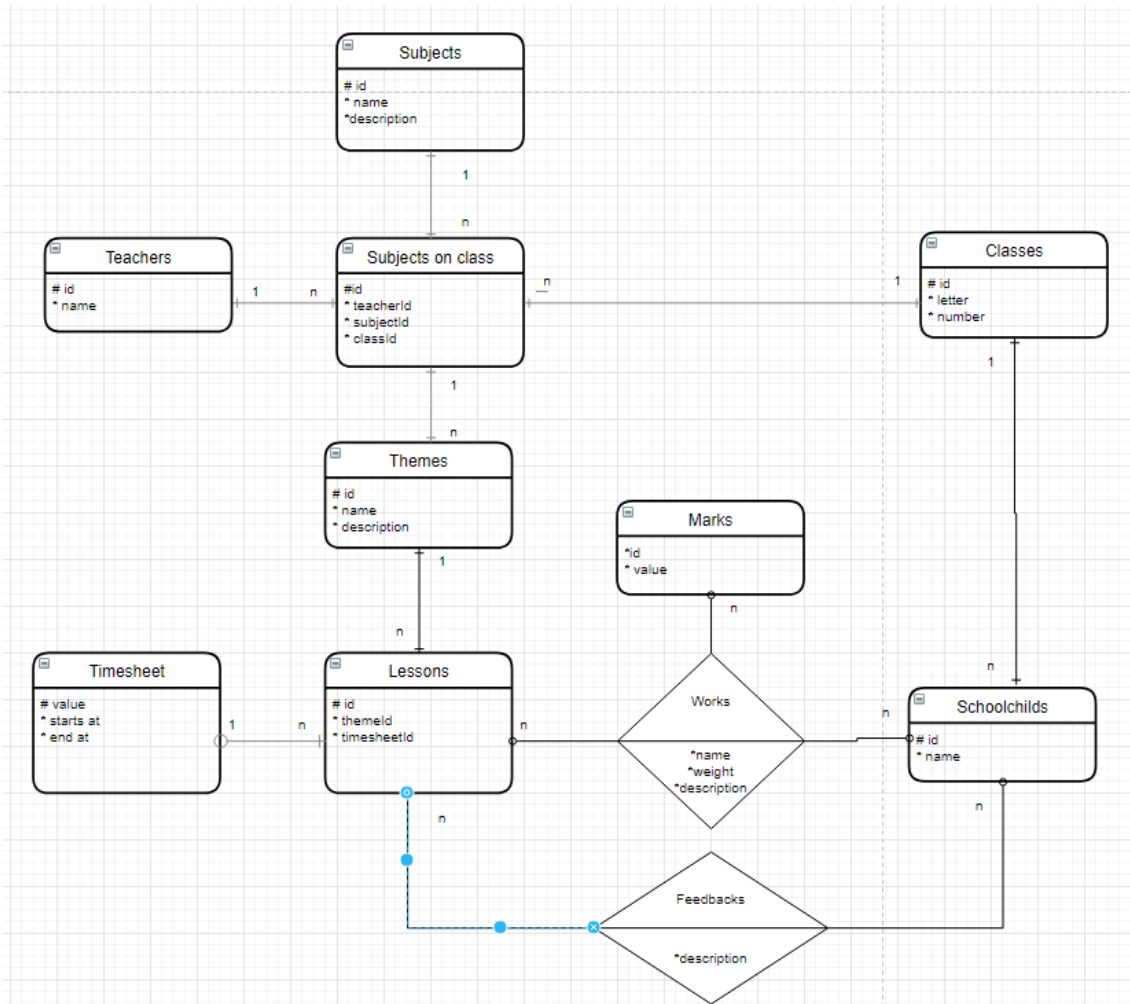


Рисунок 3.2 ER-модель даних

Крім того використовуючи App Engine Studio було також додано три ролі користувачів: директор, вчитель та учень. Також даний інструмент дозволяє з легкістю встановити рівень прав для кожного користувача відповідно до таблиць, тобто дозволити створювати, переглядати, змінювати та видаляти дані або ні. Тому враховуючи можливості кожної ролі було надано відповідні права доступу.

3.2.2 Налаштування відображення даних

Після створення всіх необхідних даних, на своєму instance відразу можна знайти всі створені таблиці та заповнити їх даними. Однак, для зручності використання застосунку аудиторією, необхідно додати ряд конфігурацій, що допоможуть користувачам використовувати додаток більш ефективно.

Для початку слід налаштувати відображення меню для користувачів з різними ролями. Наприклад, для вчителів має бути доступна для перегляду таблиця, що містить всіх вчителів, а для учнів – ні. Такі конфігурації можна здійснити в Application Menu. Відповідно створивши необхідні модулі, тобто пункти меню, та налаштувавши доступ користувачів до них.

	Title	Table	Active	Filter	Order	Link type	Device type	R
<input type="checkbox"/>	Timesheet	Timesheet [x_547266_myschoo_0_timesheet]	true			List of Records		adm x_54 x_54
<input type="checkbox"/>	Subjects	Subject [x_547266_myschoo_0_subject]	true			List of Records		adm x_54 x_54
<input type="checkbox"/>	Themes	Theme [x_547266_myschoo_0_theme]	true			List of Records		adm x_54 x_54
<input type="checkbox"/>	Teachers	User [sys_user]	true	roles=x_547266_myschoo_0.teacher^EQ		List of Records		
<input type="checkbox"/>	Classes Subject	Classes subject [x_547266_myschoo_0_classes_subject]	true			List of Records		adm x_54 x_54

Рисунок 3.3 Налаштування меню застосунку

В результаті, наприклад, користувач, що має роль вчителя, буде бачити наступні пункти в меню: Предмети, Теми, Уроки, Учні, Відгуки, Завдання.

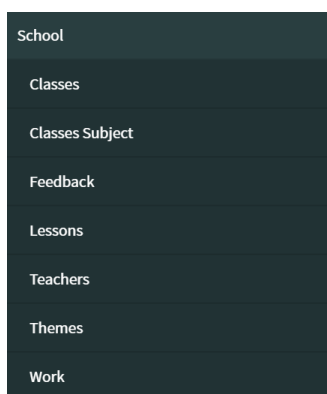


Рисунок 3.4 Вигляд меню для користувача, що має роль "Вчитель"

Зазвичай пункти меню просто пов'язані з певною таблицею і не потребують додаткових налаштувань. Проте, якщо потрібно відобразити лише деякі дані з таблиці, як, наприклад, для пункту меню «Вчителі», то можна додати відповідні фільтри.



Рисунок 3.5 Налаштування модулю "Teachers"

Крім того ServiceNow надає безліч інструментів для налаштувань відображення даних таблиць, тобто відображення форми. Вони передбачають додавання всіх необхідних умов відповідно до потреб користувачів.

Інструменти для конфігурації даних на формі:

- *Client scripts* – надає можливість запускати код, написаний на JavaScript, що відбувається на клієнтській стороні. Дозволяє додавати події, які відбуваються під час завантаження форми, після відображення форми, або, коли значення поля змінюється. Наприклад: зробити поле видимим або обов'язковим.
- *Business rules* – скрипт на серверній стороні, що відбувається під час відображення, додавання, оновлення чи видалення запису в таблиці.
- *UI policies* – динамічно змінює поведінку інформації на формі.

- *Data policies* – дозволяє контролювати доступність даних за допомогою встановлення обов’язкових полів та полів тільки для читання.
- *UI actions* – дозволяє додавати на форму або список кнопки, посилання. Це дозволяє зробити дизайн більш гнучким та орієнтованим на клієнта.

До того ж, окрім налаштувань всієї форми, можна налаштовувати окремі поля. Наприклад, для того, щоб під час створення нового відгуку або роботи до поля «Школярі» пропонувалися лише користувачі, що мають роль учня, можна налаштувати посилання під час пошуку.

Reference Specification | Choice List Specification | Dependent Field | Calculated Value | Default Value

The **Reference** field specifies what table this field displays values from.

* Reference: User

Use reference qualifier: Advanced

Reference qual: javascript:new x_547266_myschoo_0.Utils().getSchoolchilds(current.getValue('lesson'))

Рисунок 3.6 Конфігурація відображення можливих варіантів поля

Тут, як можна помітити, виконується javascript код, саме завдяки йому дані під час пошуку будуть фільтруватися. Однак для, щоб мати можливість використовувати функцію `getSchoolchilds()`, спочатку необхідно її створити. Для цього використовують *Script includes* – включення, написані на мові JavaScript, що дозволяють запускати описані частини коду на серверній

стороні.

Script Include
Utils

Name: Utils

API Name: x_547266_myschoo_0.Utils

Client callable: ☒

Application: MySchool

Caller Access: -- None --

Accessible from: All application scopes

Active: ☒

Description:

Script

```

1 var Utils = Class.create();
2 Utils.prototype = Object.extend(Object.prototype, {
3
4   getUserByRole: function(userRole){
5     var users = [];
6     var userRoleGr = new GlideRecord('sys_user_has_role');
7     userRoleGr.addEncodedQuery('role.nameSTARTSWITHx_547266_myschoo_0.'+userRole);
8     userRoleGr.query();
9     while(userRoleGr.next()){
10      users.push(userRoleGr.user.sys_id);
11    }
12    return users;
13  },

```

Рисунок 3.7 Приклад створення Script include

Під час написання коду для Script include можна використовувати стандартні класи такі як GlideDate, GlideRecord, GlideSystem та інші. Вони використовуються для доступу до різноманітних елементів системи. Так, наприклад, GlideRecord дозволяє легко отримати дані з будь-якої таблиці, а також він дозволяє змінювати та додавати нову інформацію. Приклад його використання можна побачити в функції, яка відповідає за пошук школяра, що має бути присутнім на певному уроці.

```

getSchoolchilids: function(lesson){
  var schoolchilids = [];
  var lessonGr = new GlideRecord('x_547266_myschoo_0_Lesson');
  lessonGr.get(lesson);
  var groupID = lessonGr.subject_on_class.classes_subject.class_id;
  var userGroupGr = new GlideRecord('sys_user_grmember');
  userGroupGr.addQuery('group.sys_id', groupID);
  userGroupGr.query();
  while(userGroupGr.next()){
    schoolchilids.push(userGroupGr.user.sys_id);
  }
  return 'sys_idIN'+schoolchilids.join(',');
},

```

Рисунок 3.8 Приклад використання GlideRecord

3.2.3 Додання нових компонент для відображення даних

Крім стандартного вигляду, що надає платформа, також можна створювати й власні. Для цього існує такий інструмент, як UI Builder. Він надає можливість створювати нові сторінки та стилізувати їх на власний смак. Інструмент містить багато різноманітних компонент, які можна використовувати. Якщо ж серед них не знайдеться тих, що задовільняють потреби користувача, він може створити власну за допомогою ServiceNow CLI.

ServiceNow CLI – інтерфейс командного рядка, що дозволяє створювати компоненти для відображення даних використовуючи UI Framework, а також команди для налаштування застосунку з командної стрічки.

UI Framework – фреймворк, розроблений ServiceNow, що надає можливість створювати власні елементи відображення даних, використовуючи різноманітні можливості платформи, наприклад: використання вже існуючих шаблонів для деяких компонентів.

Саме потужностями UI Framework та ServiceNow CLI було створено сторінку для перегляду учнем власного розкладу, оцінок, відгуків та завдань.

Для початку використання даних інструментів слід встановити зазначені в документації версії Node та npm, адже вони являються необхідними ресурсами. Після інсталяції всіх ресурсів користувачу необхідно лише увійти до системи зі своїми даними за допомогою наступної команди: `“now-cli login --host http://myinstance.service-now.com --method basic --username username --password password”`. Виконавши всі перераховані вище кроки, користувач може створювати власний проект та починати розробку.

Загальні принципи UI Framework дещо схожі на можливості, що надає фреймворк React. Так, компоненти створені за допомогою фреймворку ServiceNow також можуть використовувати `property` та `state`.

Для того, щоб створити власну компоненту необхідно викликати функцію `createCustomElement()`, де буде описано назву елемента та всі необхідні конфігурації, такі як: вигляд, обробка подій, стилі, стани та властивості. Властивості (`properties`) елемент отримує, як параметр під час створення, однак можна також вказати значення за замовчанням.

```
createCustomElement( {elementName: 'child-view', config: {
  renderer: {type: snabbdom},
  initialState: {
    classSysId: undefined,
    subjects: [],
    loading: true,
    selectedItem: "calendar"
  },
  properties: {
    childSysId: {
      default: "fe82abf03710200044e0bfc8bcbce5d01"
    }
  },
  view,
  ...actionsList,
  styles,
});
```

Рисунок 3.9 Приклад створення власної компоненти

Компонента може оброблювати стандарті події, такі як: `COMPONENT_BOOTSTRAPPED` (відбувається після відображення компоненти у вікні), або `COMPONENT_ERROR_THROWN` (генерується під час виникнення помилок). Також елемент може опрацьовувати подій викликані в дочірніх компонентах або власні події.

Для генерації подій використовується функція `dispatch()`. Подія, викликана у такий спосіб проходить шлях вгору від дочірнього елемента до батьківського.

Наприклад, генерація події, що викликає запит для отримання об'єкту класу, в якому навчається учень, виглядає наступним чином:

```
dispatch(SCHOOL.GET_CLASS.REQUEST,
  {
    sysparm_query: query,
    sysparm_fields: `group`
  });
```

Рисунок 3.10 Приклад виклику події

Можна помітити, що дана функція приймає два параметри: перший відповідає за назву події, що згенерується, а другий – це об’єкт, який отримає приймаюча функція.

Також UI Framework надає власну функцію для здійснення запитів на сервер: `createHttpEffect()`. Функція приймає `url` запиту та об’єкт, що його описує.

```
export const fetchLessons = createHttpEffect( url: `api/now/table/${SCHOOL.TABLE_NAMES.LESSONS}`, options: {
  method: 'GET',
  queryParams: ['sysparm_query', 'sysparm_fields'],
  headers: {"Content-Type" : "application/json",
    "Accept": "application/json"},
  successActionType: SCHOOL_CALENDAR.FETCH_LESSONS.SUCCEEDED,
  errorActionType: SCHOOL_CALENDAR.FETCH_LESSONS.FAILED
});
```

Рисунок 3.11 Приклад створення запиту до сервера

Крім стандартних параметрів, що описують запит, таких як метод чи заголовки, присутні й додаткові, наприклад: `queryParams`.

Властивість `queryParams` дозволяє зручно фільтрувати дані, що мають надійти, та одразу вказати поля, які необхідно отримати з серверу. Однак, для того, щоб отримати бажаний результат, необхідно спочатку правильно сформулювати запит. В цьому випадку в нагоді стає REST API Explorer.

Всі запити, що створюються таким способом та звертаються до ServiceNow можна зручно протестувати на платформі за допомогою REST API. Цей інструмент надає можливість зручно переглянути, які параметри є необхідними для формування певного запиту, а які – опціональні.

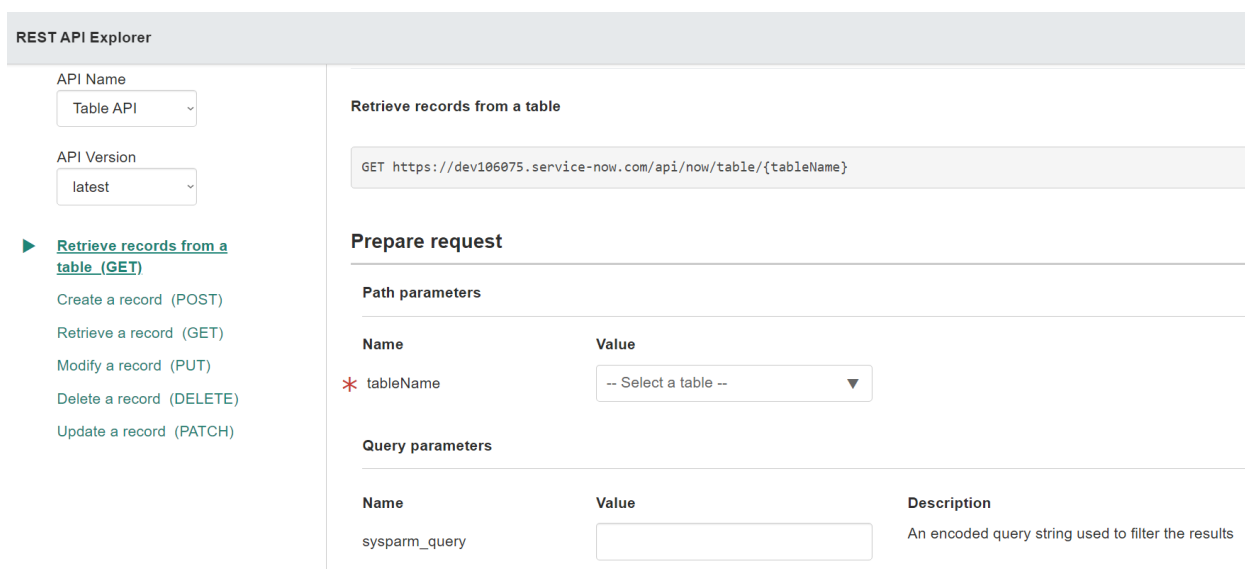


Рисунок 3.12 Rest API Explorer

Як було згадано раніше, крім створення елементів, існує також можливість додавати до власних компонентів – стандартні. Все, що для цього необхідно зробити, це знайти потрібну компоненту в бібліотеці, встановити її, та налаштувати відповідно до власних потреб.

```
return (
  <div>
    <now-tabs items={SCHOOL.MENU_ITEMS}
              selectedItem={selectedItem}
    />
    {!loading ? renderView() : null}
  </div>
);
```

Рисунок 3.13 Приклад використання стандартної компоненти «now-tabs»

Після здійснення всіх необхідних налаштування власної компоненти її необхідно задеплоїти до платформи, це відбувається за допомогою команди: “now-cli deploy”. Далі такий елемент можна використовувати під час заповнення будь-якої сторінки застосунку в UI Builder.

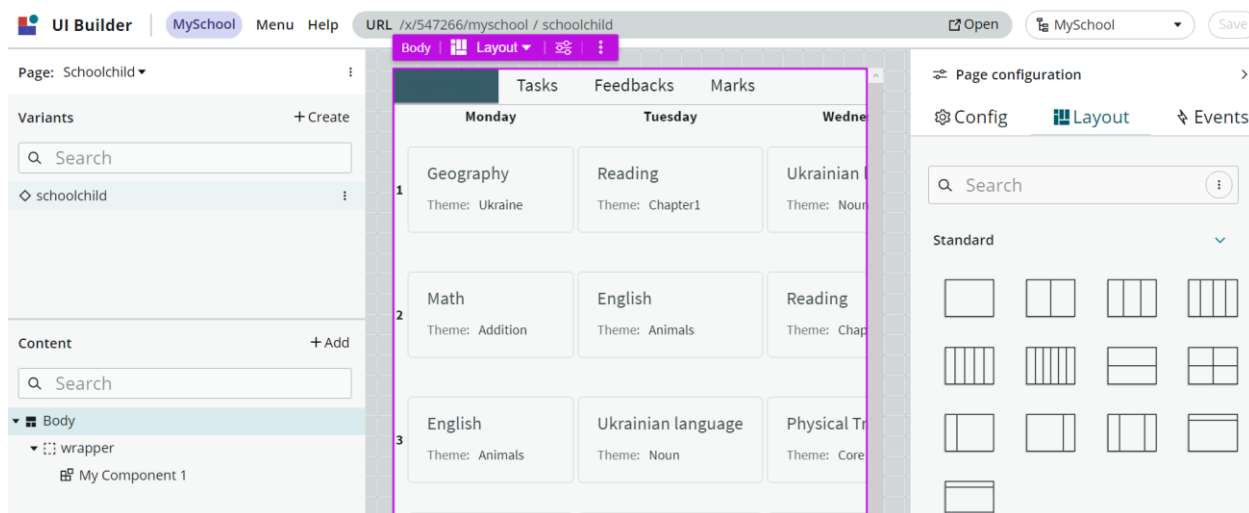


Рисунок 3.14 Додання власної компоненти в UI Builder

3.3 Опис результатів розробки

Застосунок має три ролі користувачів. Кожна роль надає доступ до різного рівня можливостей взаємодії з застосунком.

Директор має можливість налаштовувати та працювати з усіма даними у системі, тобто в цілому за потреби такий користувач може сам конфігурувати систему.

Вчителі мають обмежений доступ до застосунку і можуть виконувати лише дозволені дії, наприклад:

- 1) Перегляд всіх існуючих в системі класів;

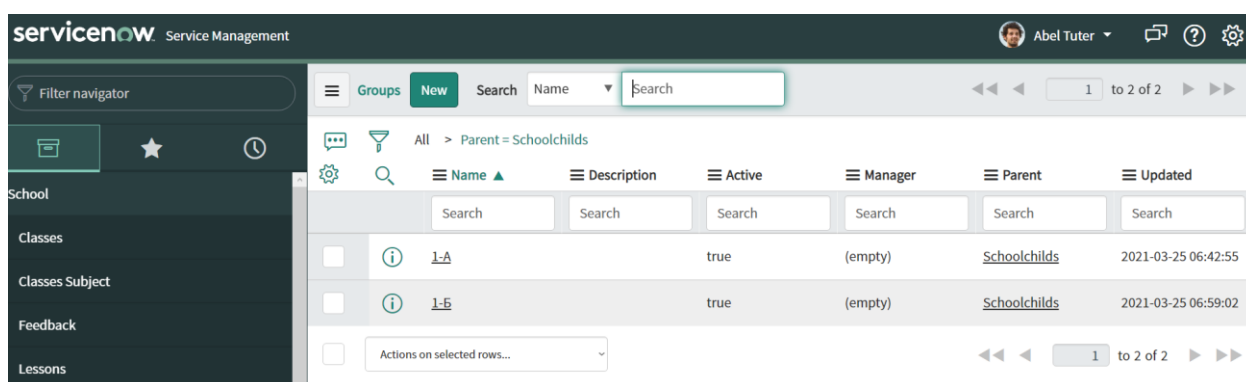


Рисунок 3.15 Перегляд класів вчителем

- 2) Перегляд всіх вчителів;

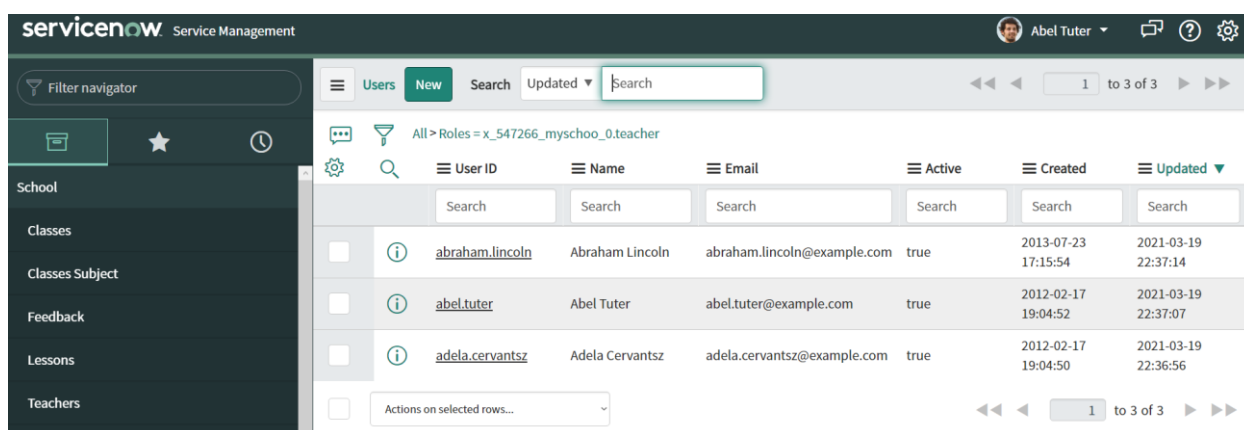


Рисунок 3.16 Перегляд списку вчителів вчителем

3) Перегляд та робота з таблицею, що зберігає уроки;

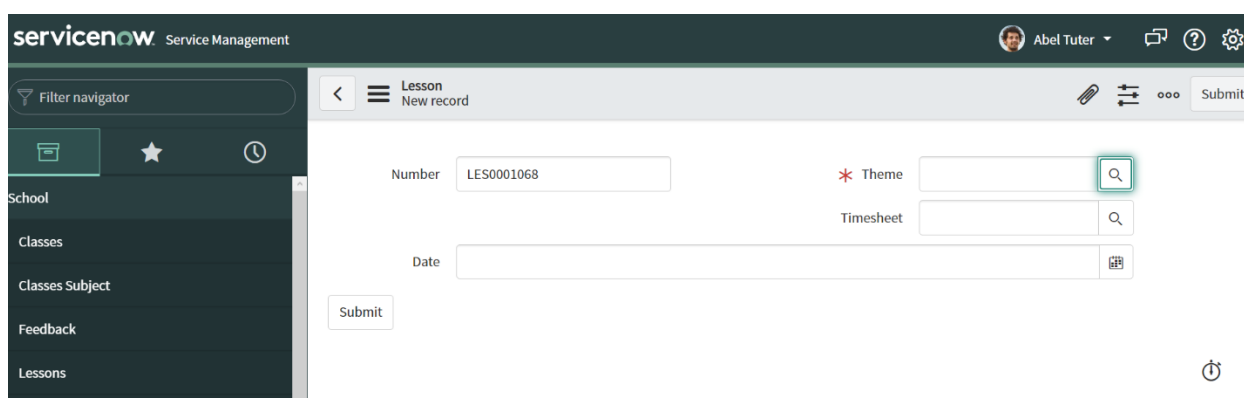


Рисунок 3.17 Створення нового уроку вчителем

4) Переглядати, створювати та видаляти відгуки на роботу учнів під час уроку;

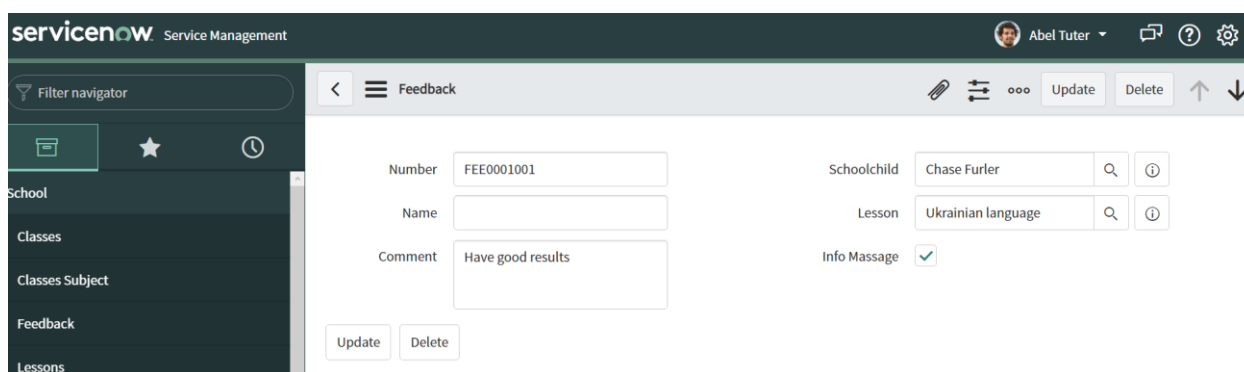


Рисунок 3.18 Перегляд відгуку вчителем

5) Переглядати, створювати та налаштовувати завдання.

	Number	Name	Description	Due date	Mark	Lesson	Weight
<input type="checkbox"/>	WOR0001003	HW	HW	2021-04-02 14:39:02	10	Logic	5
<input type="checkbox"/>	WOR0001004	HW	HW	2021-03-30 14:39:59	6	English	5
<input type="checkbox"/>	WOR0001005	HW		2021-04-05 14:44:43	1	Geography	1
<input type="checkbox"/>	WOR0001007	HW	Logic HW	2021-04-13 14:43:33	(empty)	Logic	5

Рисунок 3.19 Перегляд робіт, пов'язаних з предметами, що викладає вчитель

Користувачі, що мають роль учня мають найменше прав на роботу з даними. Однак, так як метою даного застосунку було створення електронного щоденника. Для таких користувачів крім стандартного середовища, існує окрема сторінка, де вони можуть зручно переглянути свій розклад, оцінки, завдання та відгуки на власну роботу.

	Number	Name	Description	Due date	Mark	Lesson	Weight
<input type="checkbox"/>	WOR0001005	HW		2021-04-05 14:44:43	1	Geography	1
<input type="checkbox"/>	WOR0001006	HW	Description of HW	2021-04-11 14:42:16	(empty)	Math	5
<input type="checkbox"/>	WOR0001007	HW	Logic HW	2021-04-13 14:43:33	(empty)	Logic	5

Рисунок 3.20 Стандартний вигляд застосунку для користувача з роллю «Учень»

При відкритті меню з оцінками, користувач може побачити предмети, які він зараз прослуховує та середню оцінку по кожному предмету, вона обраховується з тематичних оцінок за цей предмет.

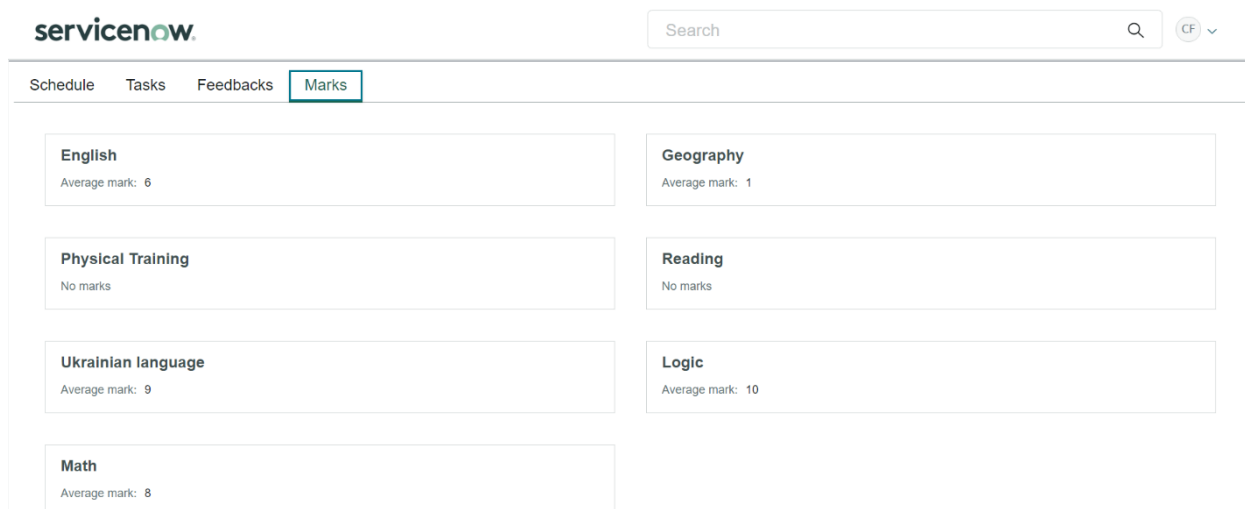


Рисунок 3.21 Перегляд оцінок на всі предмети учня

При натисненні на будь-яку з карток користувач потрапить у вікно, де можна переглянути всі оцінки за обраний предмет, відсортовані по темах.

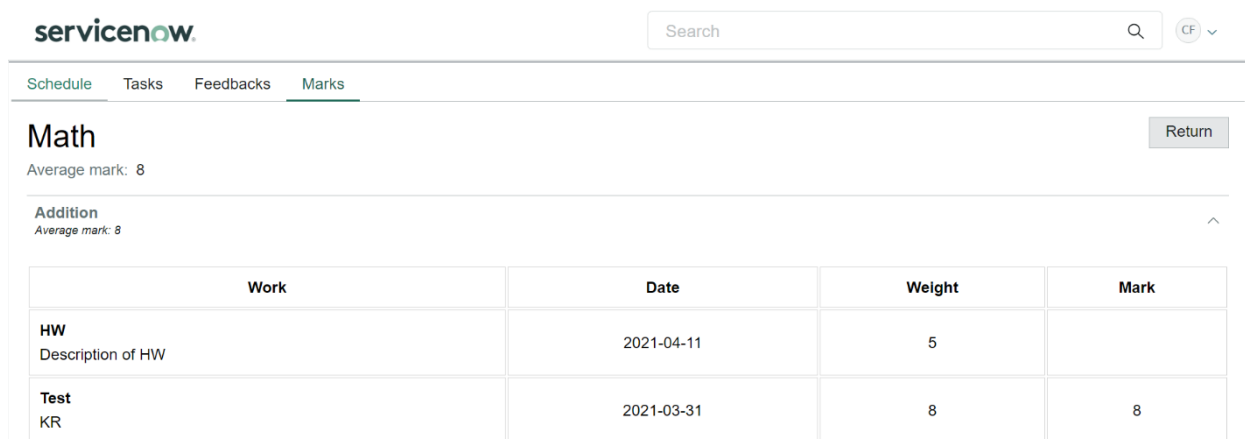


Рисунок 3.22 Перегляд оцінок учня на предмет «Математика»

У пункті меню «Завдання» учень має можливість переглянути всі завдання, які він має зробити та вже зроблені, вони відображаються відповідно у різних колонках.

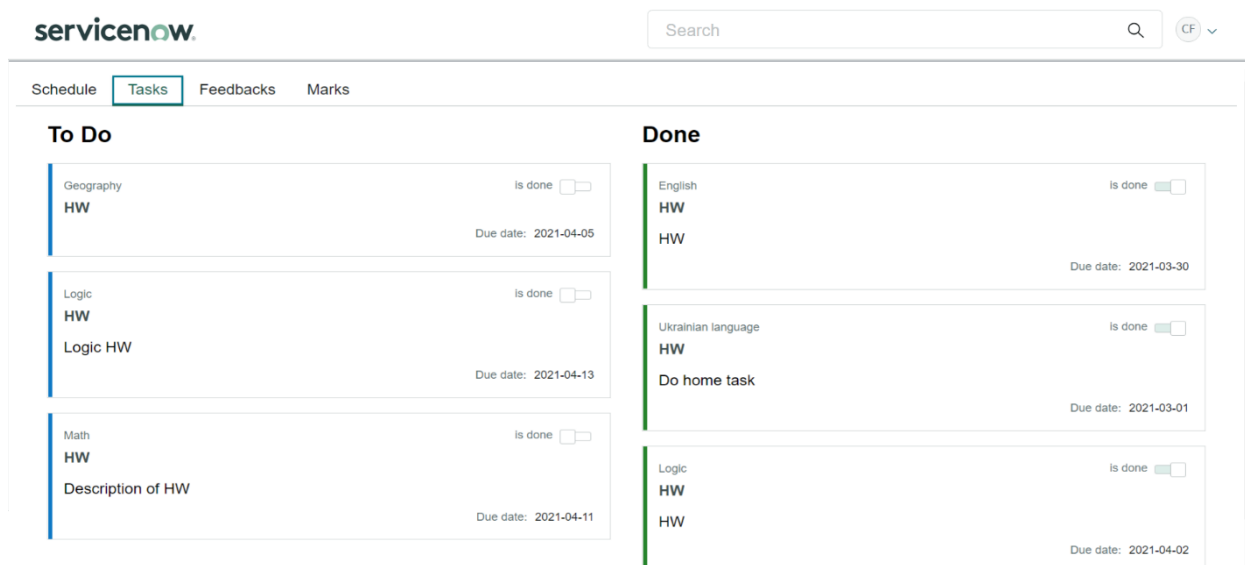


Рисунок 3.23 Перегляд завдань учня

Обравши пункт меню «Відгуки» школяр побачить всі зауваження та відгуки про свою роботу під час занять.

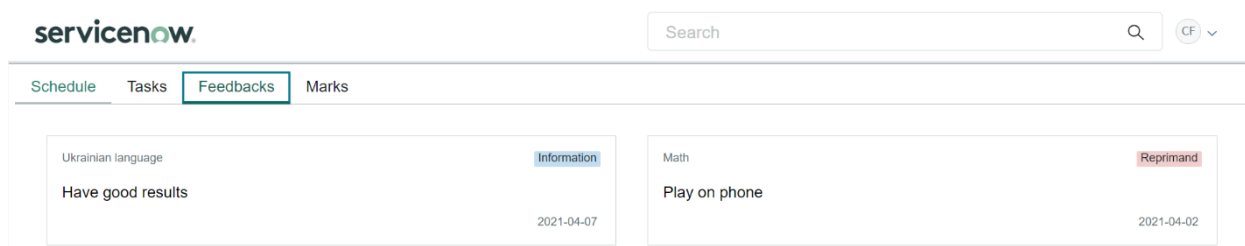


Рисунок 3.24 Перегляд відгуків на роботу учня

Свій розклад на тиждень користувач може подивитися у відповідному пункті меню «Розклад». Поряд з кожним предметом у розкладі також відображається тема уроку.

servicenow

Search

CF

	Schedule	Tasks	Feedbacks	Marks	
	Monday	Tuesday	Wednesday	Thursday	Friday
1	Geography Theme: Ukraine	Reading Theme: Chapter1	Ukrainian language Theme: Noun	Math Theme: Addition	English Theme: Animals
2	Math Theme: Addition	English Theme: Animals	Reading Theme: Chapter1	Physical Training Theme: Core	Reading Theme: Chapter1
3	English Theme: Animals	Ukrainian language Theme: Noun	Physical Training Theme: Core	Geography Theme: Ukraine	Logic Theme: Theme 1
4	Physical Training Theme: Core	Logic Theme: Theme 1	Math Theme: Addition	Ukrainian language Theme: Noun	Math Theme: Addition
5	Logic Theme: Theme 1				

Рисунок 3.25 Перегляд розкладу на тиждень учня

Висновок

У рамках написання курсової роботи було досліджено хмарні технології, а саме платформу ServiceNow. Платформа зосереджує в собі ряд інструментів та сервісів, що допомагають компаніям зручно створювати власні додатки, використовуючи при цьому мінімальну кількість коду. Однак, у випадках, коли організації все ж потрібно додати свої власні рішення, ServiceNow надає й таку можливість.

Крім того під час написання курсової роботи було розроблено власний застосунок, що має на меті реалізацію електронного щоденника учня. Відповідно до предметної області було налаштовано середовище ServiceNow. Додаток має три види користувачів, що різний доступ до даних.

Також було за допомогою UI Framework було створено окрему сторінку для учнів, де вони можуть переглядати всі актуальні дані, що стосуються їх навчання.

В майбутньому застосунок має багато перспектив для розвитку, таких як: розширення функціону та автоматизація шкільного журналу, що дозволить зручно вести облік успішності всієї школи та збирати дані з метою покращення всього учбового процесу. ServiceNow, в свою чергу, може стати зручним рішенням для подальшого розвитку проекту.

Список літератури

1. A short introduction to cloud platforms. Автор: Девід Чаппел. Рік видання: 2008.

<http://www.davidchappell.com/CloudPlatforms--Chappell.pdf>

2. What is cloud computing? Everything you need to know now. Автор: Ерік Хопп.

<https://www.infoworld.com/article/2683784/what-is-cloud-computing.html>

3. Gartner Forecasts Worldwide Public Cloud End-User Spending to Grow 18% in 2021

<https://www.gartner.com/en/newsroom/press-releases/2020-11-17-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-grow-18-percent-in-2021>

4. Pros and Cons of Cloud Storage.

<https://www.securestorageservices.co.uk/article/11/pros-and-cons-of-cloud-storage>

5. What are the Types of Cloud Computing?

[https://www.leadingedgetech.co.uk/it-services/it-consultancy-services/cloud-computing/what-are-the-types-of-cloud-computing/#:~:text=and%20public%20clouds,-.Cloud%20Services,as%20a%20Service%20\(SaaS\).](https://www.leadingedgetech.co.uk/it-services/it-consultancy-services/cloud-computing/what-are-the-types-of-cloud-computing/#:~:text=and%20public%20clouds,-.Cloud%20Services,as%20a%20Service%20(SaaS).)

6. The NIST Definition of Cloud Computing. Автори: Петер Мелл, Тімоті Гранс.

<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>

7. The world`s most innovative companies.

<https://www.forbes.com/innovative-companies/#79d1bf7a1d65>

8. What is IT Service Management (ITSM)?

<https://www.atlassian.com/itsm>

9. What is ITSM?

<https://www.servicenow.com/products/itsm/what-is-itsm.html>

10. IT Business Management.

https://docs.servicenow.com/bundle/quebec-it-business-management/page/product/it-business-management/reference/r_ITBusinessManagement.html

11. Asset Management.

https://docs.servicenow.com/bundle/quebec-it-service-management/page/product/asset-management/concept/c_AssetManagement.html

12. Software Asset Management.

<https://docs.servicenow.com/bundle/orlando-software-asset-management/page/product/software-asset-management2/concept/software-asset-management.html>