

ВИПУСКНИЙ КВАЛІФІКАЦІЙНИЙ ПРОЕКТ

на тему:

Розробка мобільного веб-додатку для забезпечення доступу до розкладу
КНТЕУ

Студента 2 курсу, 5м групи,
Спеціальності 121 «Інженерія
програмного забезпечення»
Спеціалізації
«Інженерія програмного
забезпечення»

підпис студента

Білецький Вадим
Анатолійович

Науковий керівник
кандидат педагогічних наук,
старший викладач

підпис керівника

Цензура Микола
Олександрович

Гарант освітньої програми
кандидат технічних наук,
доцент

підпис керівника

Криворучко Олена
Володимирівна

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1	4
АНАЛІЗ ОСВІТНЬОГО ПРОЦЕСУ СТУДЕНТА КНТЕУ	4
1.1. Основні поняття освітнього процесу	4
1.2. Особливості освітнього процесу студента КНТЕУ	8
1.3. Висновок до розділу 1	12
РОЗДІЛ 2	13
ДОСЛІДЖЕННЯ ПІДХОДІВ ТА ЗАСОБІВ РОЗРОБКИ МОБІЛЬНИХ ТА ВЕБ ДОДАТКІВ	13
2.1. Дослідження архітектурних рішень програмного забезпечення	13
2.3. Технології розробки мобільних додатків	19
2.4. Технології розробки веб додатків	21
2.5. Огляд функціональних можливостей системи Firebase	23
2.6. Висновок до розділу 2	25
РОЗДІЛ 3	27
РОЗРОБКА АРХІТЕКТУРИ ДОДАТКУ ДЛЯ ОТРИМАННЯ РОЗКЛАДУ СТУДЕНТА КНТЕУ	27
3.1. Проектування нереляційної бази даних	27
3.2. Розробка архітектури мобільного Android додатку	35
3.3. Розробка архітектури веб додатку на базі фреймворку Angular2	41

					КНТЕУ 121-05-04. МР			
					Розробка програмного забезпечення для отримання розкладу студента КНТЕУ	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Р4	1	68
Зав. каф.		Криворучко О.В.						
Керівник		Цензура М.О.						
Гарант		Криворучко О.В.						
Розробив		Білецький В.А.						
					ЗМІСТ	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5м група		

3.4. Висновок до розділу 3	43
РОЗДІЛ 4	44
РОЗРОБКА РОЗПОДІЛЕНОГО ДОДАТКУ ДЛЯ ОТРИМАННЯ ДОСТУПУ ДО РОЗКЛАДУ СТУДЕНТА КНТЕУ МОВАМИ ПРОГРАМУВАННЯ JAVA ТА TYPESCRIPT	44
4.1. Інтеграція робочих проектів з системою Firebase	44
4.2. Середовище та інструменти розробки мобільного додатку	47
4.3. Програмування мобільного додатку мовою Java	52
4.4. Середовище та інструменти розробки веб додатку	59
4.5. Програмування Angular2 веб додатку мовою TypeScript	60
4.6. Висновок до розділу 4	64
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
ДОДАТКИ	69

					<i>КНТЕУ 121-05-04.MP</i>	Аркуш
						2
<i>Ізм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

ВСТУП

Однією з основних ознак двадцять першого сторіччя є повсюдне проникнення інтернету в повсякденне життя людей. З точки користувача найважливішою функцією будь-якого комп'ютера чи смартфона є можливість виходу до глобальної мережі. Згідно з дослідженням німецького статистичного порталу Statista проведеного в 2019 році: 67% населення Землі мають мобільні телефони що становить близько 4.68 мільярда осіб. В середньому людина витрачає більше 3,5 годин на користування смартфоном щодня, з яких всього 30 хвилин це власне телефонні розмови. Нині гаджети стали основним засобом виходу в інтернет: в 2019 році 63,4% людей користувалися інтернетом за допомогою мобільних пристроїв, для порівняння в минулому році це число сягало 61,2%. Безперечно, такий стрімкий розвиток не обійшов стороною і студентство, згідно з дослідження американської аналітичної компанії Comscore – найбільше користуються інтернетом люди у віці від 18-24 років.

Актуальність: Оскільки все більше людей мають декілька гаджетів, наприклад: ноутбук, смартфон, планшет та стаціонарний комп'ютер зростає потреба в синхронізації даних між ними, незалежно від операційної системи.

Мета дипломної роботи: дослідження особливостей розробки мобільних та веб додатків в процесі розробки додатку для отримання розкладу студента КНТЕУ.

Завдання дипломної роботи: дослідження особливостей мобільних платформ Android та веб фреймворку Angular2, дослідити та реалізувати ключові архітектурні принципи.

Об'єкт дослідження: розробка мобільних та веб додатків і їхня взаємодія.

Предмет дослідження: Додаток «Розклад студента КНТЕУ».

					КНТЕУ 121-05-04.МР			
					Розробка програмного забезпечення для отримання розкладу студента КНТЕУ	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Вступ	3	68
Зав. каф.		Криворучко О.В.						
Керівник		Цензура М.О.						
Гарант		Цензура М.О.						
Розробив		Білецький В.А.						
					Вступ	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5 група		

РОЗДІЛ 1

АНАЛІЗ ОСВІТНЬОГО ПРОЦЕСУ СТУДЕНТА КНТЕУ

1.1. Основні поняття освітнього процесу

Перед початком проектування програмного забезпечення важливо визначити процеси які воно автоматизує, в тому числі виділити його основні поняття. Освітній процес у вищих закладах освіти (надалі — навчальний процес) — це система організаційних і дидактичних заходів, спрямованих на реалізацію змісту освіти на певному освітньому або кваліфікаційному рівні відповідно до державних стандартів освіти.

Освітній процес організується з урахуванням можливостей сучасних інформаційних технологій навчання та орієнтується на формування освіченої, гармонійно розвиненої особистості, здатної до постійного оновлення наукових знань, професійної мобільності та швидкої адаптації до змін і розвитку в соціально-культурній сфері, в галузях техніки, технологій, системах управління та організації праці в умовах ринкової економіки. Він базується на ступеневій системі вищої освіти та принципах науковості, гуманізму, демократизму, наступності та безперервності, незалежності від втручання будь-яких політичних партій, інших громадських та релігійних організацій.

Начальний процес може здійснюватися у різних формах

Форма навчання – зовнішня організація навчального процесу, для студентів ЗВО доступні такі:

- Денна
- Заочна

					КНТЕУ 121-05-04.MP			
					Розробка програмного забезпечення для отримання розкладу студента КНТЕУ	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Р1	4	68
Зав. каф.	Криворучко О.В.							
Керівник	Цензура М.О.							
Гарант	Цензура М.О.				Аналіз освітнього процесу студента КНТЕУ	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5 група		
Розробив	Білецький В.А.							

- **Екстенатна**

Форми навчання можуть бути поєднані. Терміни навчання за відповідними формами визначаються можливостями виконання освітньо-професійної програми підготовки (ОПП) фахівців певного освітньо-кваліфікаційного рівня

Навчання за денною та заочною формами навчання здійснюється за кошти державного бюджету, а також на підставі договорів між університетом та підприємствами, установами, організаціями, фізичними особами.

Екстернат є особливою формою навчання осіб (екстернів) для здобуття певного рівня вищої освіти шляхом самостійного вивчення навчальних дисциплін і складання в університеті заліків, екзаменів та проходження інших форм підсумкового контролю, передбачених навчальним планом.

Навчальний процес в університеті реалізується в таких формах: навчальні заняття, виконання індивідуальних завдань, практики, контрольні заходи, самостійна робота.

Основними видами навчальних занять є: лекції, лабораторні, практичні, семінарські заняття, комп'ютерний практикум та консультації.

Лекція – основний вид навчальних занять, призначених для викладення теоретичного матеріалу. Тематика лекцій визначається робочою програмою навчальної дисципліни.

Лабораторне заняття – вид навчального заняття, на якому студент під керівництвом викладача проводить натурні або імітаційні експерименти чи дослідження з метою практичного підтвердження окремих теоретичних положень, набуває практичних навичок роботи з лабораторним обладнанням, оснащенням, обчислювальною технікою, вимірювальною апаратурою, оволодіває методикою експериментальних досліджень у конкретній предметній галузі та обробки отриманих результатів.

Перелік тем лабораторних занять визначається робочою програмою навчальної дисципліни.

					<i>КНТЕУ 121-05-04.MP</i>	Аркуш
						5
<i>Ізм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

Лабораторне заняття включає проведення контролю щодо підготовки студентів до виконання конкретної лабораторної роботи, виконання власне лабораторних досліджень, оформлення індивідуального звіту про виконану роботу та його захист.

Практичне заняття – вид навчального заняття, на якому студенти під керівництвом викладача шляхом виконання певних відповідно сформульованих завдань закріплюють теоретичні положення навчальної дисципліни і набувають умінь та навичок їх практичного застосування.

Оцінки, одержані студентом на практичних заняттях враховуються при визначенні підсумкової оцінки з даної навчальної дисципліни

Семінарське заняття – вид навчального заняття, на якому викладач організує дискусію з попередньо визначених проблем. На підставі індивідуальних завдань (рефератів) студенти готують тези виступів з цих проблем.

Індивідуальне навчальне заняття – проводиться з окремими студентами, які виявили особливі здібності в навчанні та схильність до науково-дослідної роботи і творчої діяльності з метою підвищення рівня їх підготовки та розкриття індивідуального творчого обдарування, а також зі студентами, які мають певні проблеми (відставання у навчанні). Індивідуальні навчальні заняття організовуються у позанавчальний час за окремим графіком, складеним кафедрою.

Консультація – вид навчального заняття, на якому студент отримує від викладача відповіді на конкретні питання або пояснення окремих теоретичних положень чи їх практичного використання. Під час підготовки до екзаменів (семестрових, державних) проводяться групові консультації.

Протягом семестру консультації з навчальних дисциплін проводяться за встановленим кафедрою розкладом.

Самостійна робота студента (СРС) є основним засобом засвоєння навчального матеріалу у вільний від аудиторних занять час.

Самостійна робота студента включає: опрацювання навчального матеріалу, виконання індивідуальних завдань, науково-дослідну роботу.

					КНТЕУ 121-05-04.МР	Аркуш
						6
Изм.	Аркуш	№ докум	Підпис	Дата		

Навчальний час, відведений на самостійну роботу студента денної форми навчання, регламентується робочим навчальним планом і може складати від 1/3 до 2/3 від загального обсягу навчального часу, відведеного на вивчення конкретної дисципліни.

Зміст самостійної роботи студента над конкретною дисципліною визначається робочою програмою, методичними матеріалами, завданнями та вказівками викладача.

Самостійна робота студента забезпечується системою навчально-методичних засобів, передбачених робочою програмою навчальної дисципліни: підручниками, навчальними та методичними посібниками, конспектами лекцій, збірниками завдань, комплектами індивідуальних семестрових завдань, практикумами, комп'ютерними навчальними комплексами, методичними рекомендаціями з організації СРС.

Курсова робота (КР) – це індивідуальне завдання, яке передбачає розробку сукупності документів (розрахунково-пояснювальної або пояснювальної записки, при необхідності – графічного, ілюстративного матеріалу) та є творчим або репродуктивним рішенням конкретної задачі щодо об'єктів діяльності фахівця (програмних засобів, або їх окремих частин; економічних, соціальних, лінгвістичних проблем тощо), виконаним студентом самостійно під керівництвом викладача згідно із завданням, на основі набутих з даної та суміжних дисциплін знань та умінь.

					КНТЕУ 121-05-04.МР	Аркуш
						7
Ізм.	Аркуш	№ докум	Підпис	Дата		

1.2. Особливості освітнього процесу студента КНТЕУ

Рис. 1.1. Організаційна структура КНТЕУ

КНТЕУ – один із найавторитетніших закладів вищої освіти України, атестований за найвищим, IV рівнем акредитації. Його історія бере початок від 1946 р. Указом Президента України у 2000 р. університету надано статус національного. У 2006 р. КНТЕУ приєднався до Великої Хартії університетів.

Управління діяльністю університету здійснюється на принципах автономії і самоврядування, демократизації прийняття рішень. Система управління якістю КНТЕУ сертифікована за міжнародними стандартами.

Університет — лідер у реформуванні вищої освіти України, першим у країні здійснив суттєве оновлення змісту освіти, впровадження новітніх навчальних технологій на базі програмних продуктів та глобальних інформаційних мереж. У вищому навчальному закладі успішно розвивається інтегрована ступенева освіта, приділено увагу міжнародній мобільності студентів і викладачів.

Створено сучасний бібліотечний комплекс з електронним обслуговуванням читачів, доступом до повнотекстових електронних ресурсів міжнародних баз даних.

В університеті навчається більше 40 тис. студентів.

Функціонують 6 факультетів:

- міжнародної торгівлі та права, ФМТП;
- економіки, менеджменту та психології, ФЕМП;
- фінансів та банківської справи, ФФБС;

- обліку, аудиту та інформаційних систем, ФОАІС;
- ресторанно-готельного та туристичного бізнесу, ФРГТБ;
- торгівлі та маркетингу, ФТМ;

Здійснюється професійна підготовка фахівців за 42 бакалаврськими і 47 магістерськими програмами для сфер внутрішньої та зовнішньої торгівлі, фінансової системи і права, митної служби, антимонопольної діяльності, економіки підприємництва, банківської і страхової справи, бухгалтерського обліку, фінансового контролю, товарознавства, захисту прав споживачів, ресторанного господарства, туризму, готельного господарства, маркетингово-комунікаційної галузі тощо. Студенти проходять практику на вітчизняних та іноземних підприємствах як в Україні так і за кордоном.

Центр КНТЕУ знаходиться на Лівому березі Києва, поблизу станції метро Лісова, між вулицями Мілютенка та Кіото, де розміщується студентський кампус з навчальними корпусами і гуртожитками. Загалом київський підрозділ має 10 корпусів та 7 гуртожитків більшість з яких знаходиться на «Лісовій». А також корпуси «Н», «Р», «М» що знаходяться на правому березі поблизу станцій «Печерська» та «Дружби народів»

					КНТЕУ 121-05-04.МР	Аркуш
						9
Ізм.	Аркуш	№ докум	Підпис	Дата		



Рис. 1.2. Розташування київських корпусів КНТЕУ

Київські корпуси КНТЕУ:

- А - вул. Кіото 19 (головний корпус)
- Б - вул. Кіото 19 (бібліотечний корпус)
- В - вул. Кіото 19 (Конгрес-центр)
- Г - вул. Мілютенка 8
- Д - вул. Кіото 21
- Е - вул. Мілютенка 4
- Л - вул. Кіото 23
- М - вул. Чигоріна 57
- Н - вул. Чигоріна 57а
- Р - вул. Раєвського 36

Перший тиждень	Другий тиждень	Пара	День тижня
Л ІТ-право В Л-410 доц. к.ю.н. Тімашов В.О.	Л ІТ-право В Л-310 доц. к.ю.н. Тімашов В.О.	1	Понеділок
П ІТ-право В Л-410 доц. к.ю.н. Тімашов В.О.	П ІТ-право В Л-310 доц. к.ю.н. Тімашов В.О.	2	
		3	
		4	
Л Проєктування мультимед сист В Б-514 ст.в. Жирова Т.О.	Л Технології аналізу даних Б-520 проф. Роскладка А.А.	1	Вівторок
Л Проєктування мультимед сист В Б-514 ст.в. Жирова Т.О.	Л Технології аналізу даних Б-520 проф. Роскладка А.А.	2	
		3	
		4	
Л61 Проєктування мультимед сист В Б-510 ст.в. Жирова Т.О.	Л61 Проєктування мультимед сист В Б-510 ст.в. Жирова Т.О.	1	Середа
Л62 Технології аналізу даних Б-515 проф. Роскладка А.А.	Л62 Технології аналізу даних Б-515 проф. Роскладка А.А.	2	
Л61 Проєктування мультимед сист В Б-510 ст.в. Жирова Т.О.	Л61 Проєктування мультимед сист В Б-510 ст.в. Жирова Т.О.		
Л62 Технології аналізу даних Б-515 проф. Роскладка А.А.	Л62 Технології аналізу даних Б-515 проф. Роскладка А.А.	3	
Л62 Проєктування мультимед сист В Б-510 ст.в. к.п.н. Котенко Н.О.	Л62 Проєктування мультимед сист В Б-510 ст.в. к.п.н. Котенко Н.О.		
Л61 Технології аналізу даних Б-523 проф. Роскладка А.А.	Л61 Технології аналізу даних Б-515 проф. Роскладка А.А.	4	
Л62 Проєктування мультимед сист В Б-510 ст.в. к.п.н. Котенко Н.О.	Л62 Проєктування мультимед сист В Б-510 ст.в. к.п.н. Котенко Н.О.		
Л61 Технології аналізу даних Б-523 проф. Роскладка А.А.	Л61 Технології аналізу даних Б-515 проф. Роскладка А.А.		
		1	Четвер
	Самостійна робота	2	
Л Технології аналізу даних Б-520 проф. Роскладка А.А.		3	
Л Технології аналізу даних Б-520 проф. Роскладка А.А.		4	
Самостійна робота	Самостійна робота		П'ятниця

Рис. 1.3. Приклад розкладу студента КНТЕУ

Місце проведення занять залежить від багатьох факторів: кафедри яка викладає предмет, курсу та форми навчання що відображається в розкладі студента. Студенти старших курсів мають більше занять вузького профілю в той час як студенти перших курсів мають багато занять з загальноосвітніх предметів що проводяться за місцем кафедри. Деканат Факультету обліку, аудиту та інформаційних систем розташований в корпусі «Д» а кафедра програмної інженерії та кібербезпеки ФОАІС розташована в корпусі Б, відповідно студенти старших курсів більшість часу проводять в корпусів «Б» та «Д». У зв'язку з великою кількістю студентів розклад студента розділяється на тижні: перший або другий що часто створює незручності. Типовим розкладом студента є таблиця що складається з п'яти рядків (Понеділок, Вівторок, Середа, Четвер, П'ятниця) що поділяються на два стовпця (Перший тиждень, Другий тиждень). Кожному дню відповідає пронумерований список занять. В рядку про заняття зазначається інформація про вид заняття (Лекція, Практикум або Лабораторна) та назва

предмету, нижче зазначається корпус та аудиторія, наукове звання викладача (Н-д: старший викладач, доцент, професор) та ПІБ викладача.

1.3.Висновок до розділу 1

Освітній процес є складним процесом що охоплює багато понять.

Може відбуватися в різних формах що визначаються визначеними умовами.

Київський Національний Торгівельно-Економічний Університет має складну багатоланкову організаційну структуру управління. Має 17 регіональних підрозділів в 9 містах України та 6 факультетів де навчається більше 40 тисяч студентів.

Кожен регіональний розділ має корпуси, наймасштабнім є Київський – що має 9 навчальних корпусів в декількох точках міста.

Розклад навчання студента визначають багато факторів, серед основних безумовно спеціальність та факультет, курс навчання, кафедра та інші.

В розкладі зазначається інформація про предмет, викладача з позначенням його вченого звання, порядковий номер заняття та тип заняття, що може бути:

- Лабораторним (ЛБ)
- Практичним (П)
- Лекційним (Л)

Розклад студента також залежить від номеру поточного тижня.

					КНТЕУ 121-05-04.МР	Аркуш
						12
Изм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ПІДХОДІВ ТА ЗАСОБІВ РОЗРОБКИ МОБІЛЬНИХ ТА ВЕБ ДОДАТКІВ

2.1. Дослідження архітектурних рішень програмного забезпечення

Перед початком розробки програмного забезпечення важливо обрати архітектурний шаблон який міг би забезпечити найкращий спосіб вирішення функціональних завдань програмного забезпечення що розробляється.

Архітектурний шаблон – загальне вирішення проблеми що часто виникає в архітектурі ПЗ в певному контексті.

Одними з популярних архітектурних шаблонів є:

- 1- Багаторівневий шаблон
- 2- Клієнт-серверний шаблон
- 3- Шаблон Event bus
- 4- Модель – Вигляд – Контролер

1. Багаторівневий шаблон – цей шаблон може використовуватися для впорядкування ПЗ що можуть бути розділені на групи та підзадачі, кожна з яких перебуває на певному рівні абстракції. Кожен рівень забезпечує сервіси для наступного вищого рівня.

Основними рівнями є :

- Рівень представлення (Рівень UI інтерфейсу)

					КНТЕУ 121-05-04.МР			
					Розробка програмного забезпечення для отримання розкладу студента КНТЕУ	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Р2	13	68
Зав. каф.	Криворучко О.В.							
Керівник	Цензура М.О.							
Гарант	Цензура М.О.							
Розробив	Білецький В.А.				Дослідження підходів та засоби розробки мобільних та веб додатків	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5 група		

- Програмний рівень (також відомий як Рівень сервісів)
- Рівень бізнес логіки
- Рівень доступу до даних

Використання:

- Прикладні застосунки (Desktop Applications)
- Веб додатки

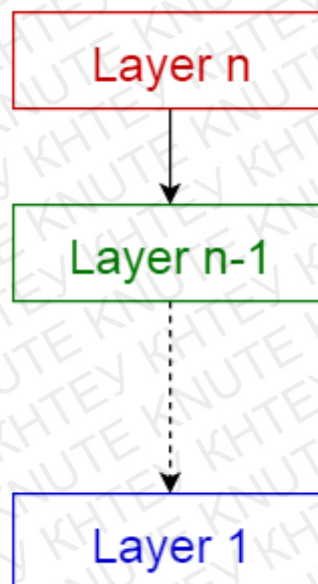


Рис. 2.1. Багаторівневий архітектурний шаблон

1. Клієнт-серверний шаблон

Цей шаблон складається з двох частин; серверу та низки клієнтів.

Сервер надає сервіси для клієнтів. Клієнти надсилаються запити на сервер який у відповідь надає відповідні сервіси. Крім того, сервер продовжує очікувати запити від клієнтів.

Застосування:

- Онлайн додатки, такі як email, поширення документів та банкінг.

					КНТЕУ 121-05-04.MP	Аркуш
						14
Изм.	Аркуш	№ докум	Подпись	Дата		

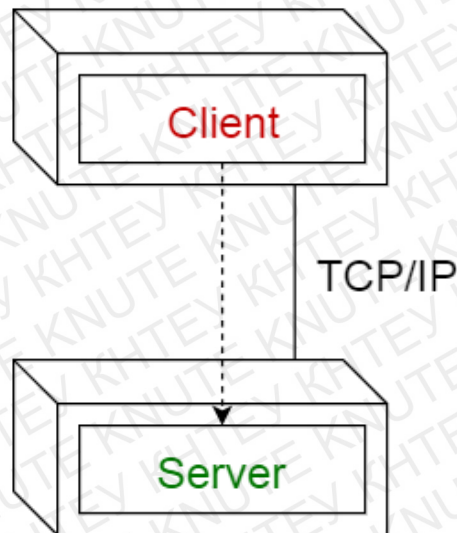


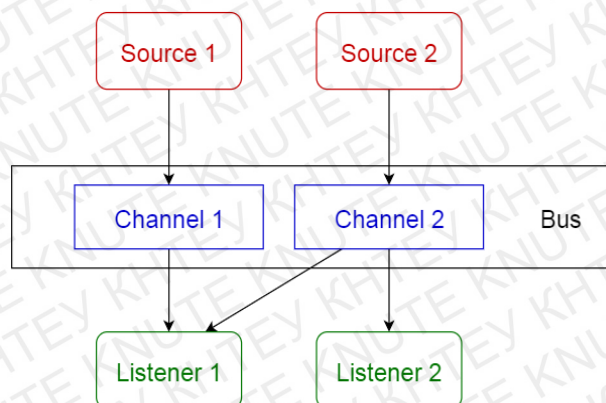
Рис. 2.2. клієнт-серверний шаблон

3. Шаблон Event-Bus

Цей шаблон має 4 основний компоненти : джерело події (event source), слухач події (event listener), канал та event bus. Джерела публікують повідомлення на визначених каналах проміжної ланки яка називається event bus. В свою чергу слухачі підписуються до певних каналів відтак отримують сповіщення про повідомлення що були передані на ті канали до яких підписані .

Застосування:

- Розробка для Android
- Сервіси сповіщень
-



					КНТЕУ 121-05-04.МР	Аркуш
						15
Изм.	Аркуш	№ докум	Подпись	Дата		

Рис. 2.3. шаблон Event Bus

Шаблон Модель-Вигляд-Контролер (Model-View-Controller)

Цей шаблон, що також відомий як MVC, розділяє інтерактивний додаток на три частини:

1. Модель – містить базову функціональність та дані
2. Вигляд – відображає інформацію користувачеві
3. Контролер – опрацьовує вхідні дані від користувача

Шаблон розроблений для відокремлення внутрішньої реалізації бізнес процесів від інформації що відображається користувачеві та надсилається користувачем на сервер.

Використання:

- Архітектура більшості Web додатків
- Веб фреймворки такі як Django та Rails

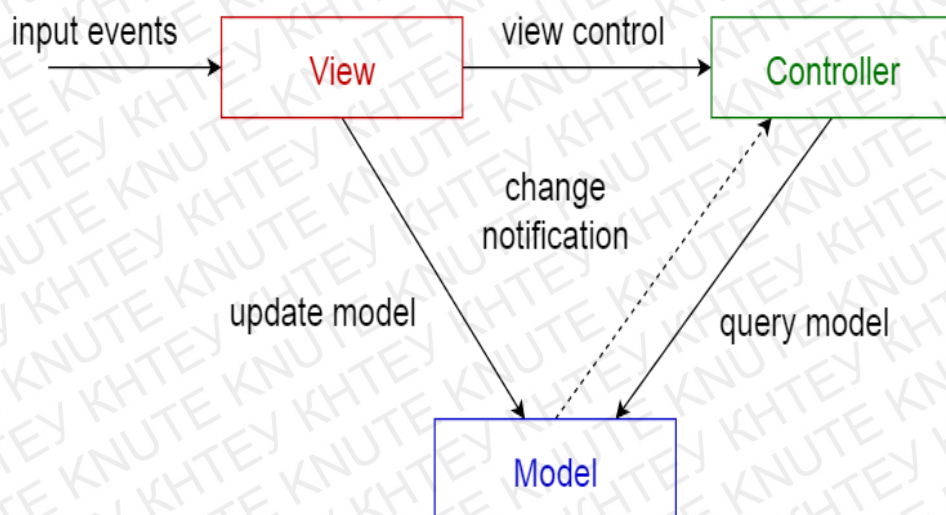


Рис. 2.4. шаблон Модель-Вигляд-Контролер

2.2. Огляд мобільних операційних систем

Мобільний застосунок або додаток — програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях.

					КНТЕУ 121-05-04.МР	Аркуш
						16
Ізм.	Аркуш	№ докум	Підпись	Дата		

Багато мобільних застосунків встановлені на самому пристрої або можуть бути завантажені на нього з онлайн магазинів мобільних застосунків, таких як App Store, Google Play, Windows Phone Store та інших, безкоштовно або за плату.

Мобільні застосунки розробляються спеціально для мобільних операційних систем, сьогодні на глобальному ринку існує дві мобільні операційні системи, ними є Android та iOS, раніше існували системи Symbian, BlackBerry та інші, але вони не витримали конкуренції і вийшли з ринку.



Рис. 2.5. логотип операційної системи Android

Поточна версія Android 9.0 «Pie»

Android — операційна система і платформа для мобільних телефонів та планшетних комп'ютерів, створена компанією Google на базі ядра Linux.

Хоча Android базується на ядрі Linux, він стоїть дещо осторонь Linux-спільноти та Linux-інфраструктури.

Базовим елементом цієї операційної системи є реалізація Dalvik віртуальної машини Java, і все програмне забезпечення і застосування спираються на цю реалізацію Java.

У 84 % смартфонів, проданих у 3-му кварталі 2014 року, була встановлена операційна система Android.

У березні 2017 року ОС Android стала найпопулярнішою ОС, з якої виходили в інтернет.

					КНТЕУ 121-05-04.МР	Аркуш
						17
Изм.	Аркуш	№ докум	Подпись	Дата		

Так, 37,93 % користувачів заходили в інтернет із Android'a, а з Windows — 37,91 % користувачів.

В Азії показники ще вищі — 52,2 % і 29,2 % відповідно.



Рис. 2.6. Логотип операційної системи iOS

Поточна версія системи 13.1

iOS — це власницька мобільна операційна система від Apple.

Розроблена спочатку для iPhone, згодом також вдосконалена для використання на iPad.

Apple не дозволяє роботу ОС на мобільних телефонах інших фірм.

Користувачський інтерфейс iOS заснований на концепції прямої маніпуляції з використанням жестів Multi-Touch.

Елементи інтерфейсу управління складаються з повзунків, перемикачів і кнопок.

Він призначений для безпосереднього контакту користувача з екраном пристрою.

Внутрішній акселерометр використовуються деякими програмами для реагування на струшування пристрою,

яке є також загальною командою скасування, або обертання пристрою у трьох вимірах, що є загальною командою перемикання між книжковим та альбомним режимами.

					КНТЕУ 121-05-04.МР	Аркуш
						18
Изм.	Аркуш	№ докум	Подпись	Дата		

2.3. Технології розробки мобільних додатків

Кожна мобільна операційна система розробляє та підтримує набір інструментів для розробки мобільних додатків. Для системи Android це Android NDK та мови програмування Java і Kotlin, а для iOS існує iOS SDK і мова програмування Swift.

Android NDK (Android Native Development Kit) — необхідний набір інструментарію для розробки компонентів програмного забезпечення для платформи Android, який базується на C/C++ та інших мовах програмування. Містить у собі лімітований набір загальноживаних низькорівневих (нативних) бібліотек та API, написаних на C/C++ та інших мовах програмування, документацію і мінімальний набір прикладів для демонстрації базового функціоналу.

За допомогою NDK розробник застосунку для операційної системи Android може імплементувати окремі його частини, використовуючи такі мови, як C/C++, а не тільки Java. Це надає можливість використати деякі переваги, оскільки в окремих випадках код, написаний на C/C++, виконуватиметься швидше в порівнянні з кодом на Java. Android NDK може бути використаний для платформи Android 1.5 (API Level 3) і новіших версій.

iOS SDK – комплект засобів розробки для iOS, до якого входять:

Cocoa Touch

- мультитач управління
- підтримка акселерометра
- ієрархія видів
- локалізація
- підтримка камери

Мультимедіа

					КНТЕУ 121-05-04.МР	Аркуш
						19
Изм.	Аркуш	№ докум	Подпись	Дата		

- OpenAL
- Мікшування і запис аудіо
- відтворення відео
- формати зображень
- Quartz
- анімаційне ядро
- OpenGL ES

Сервісне ядро

- Мережа
- Вбудована база даних SQLite
- Локаційне ядро
- Мережа
- CoreMotion

Ядро OS X

- TCP / IP
- сокети
- управління живленням
- файлова система
- Безпека

Swift — багатопарадигмова компільована мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective C і бути стійкішою до помилкового коду.

Swift була представлена на конференції розробників WWDC 2014.

Безкоштовний посібник мови програмування Swift доступний для завантаження у магазині iBooks

Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java.

З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems».

					КНТЕУ 121-05-04.MP	Аркуш
						20
Изм.	Аркуш	№ докум	Подпись	Дата		

В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано.

Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розробки об'єктно-орієнтованих програм.

Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині.

Передусім Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням,

що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми,

написані іншими мовами програмування.

2.4. Технології розробки веб додатків

Веб-додаток — додаток, в якому клієнтом є оглядач Інтернету(браузер), а сервером — веб-сервер. Оглядач Інтернету може бути реалізацією так званих тонких клієнтів. Він відображає веб-сторінки і входить до складу операційної системи, а його оновлення та супровід виконує постачальник операційної системи. Логіка додатку зосереджена на сервері, а браузер найчастіше лише відображає інформацію, завантажену з сервера, та передає на сервер дані користувача.

Однією з переваг такого підходу є той факт, що клієнти не залежать від конкретної операційної системи користувача, і веб-додатки, таким чином, є міжплатформеними сервісами.

Сучасний IT-ринок пропонує великий спектр технологій для розробки веб додатків, основні з яких:

Node.js — платформа з відкритим кодом для виконання високопродуктивних мережових застосунків, написаних мовою JavaScript.

					КНТЕУ 121-05-04.MP	Аркуш
						21
Изм.	Аркуш	№ докум	Подпись	Дата		

Засновником платформи є Раян Дал (Ryan Dahl). Якщо раніше Javascript застосовувався для обробки даних в браузері користувача, то node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їх виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

Microsoft .NET — програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Багато в чому є продовженням ідей та принципів, покладених в технологію Java. Однією з ідей .NET є сумісність служб, написаних різними мовами. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість.

Кожна бібліотека (збірка) в .NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок.

.NET — крос-платформова технологія, в цей час існує реалізація для платформи Microsoft Windows, FreeBSD (від Microsoft) і варіант технології для ОС Linux в проекті Mono (в рамках угоди між Microsoft з Novell), DotGNU .

.NET поділяється на дві основні частини — середовище виконання (по суті віртуальна машина) та інструментарій розробки.

ASP.NET Core — вільне та відкрите програмне забезпечення каркасу веб застосунків, з продуктивністю вищою ніж у ASP.NET, розроблена корпорацією Microsoft і співтовариством . Це модульна структура, яка працює як на повній платформі .NET Framework, так і на платформі .NET Core.

Не зважаючи на те, що це є новим фреймворком, побудованим на новому веб-стеку, ASP.NET Core має високу ступінь сумісності концепцій з ASP.NET MVC, який об'єднує функціональність MVC, Web API та Web Pages. В попередніх версіях платформи дані технології реалізовані окремо і тому містять багато дублювальної функціональності. Тепер це об'єднано в одну програмну модель

					КНТЕУ 121-05-04.МР	Аркуш
						22
Изм.	Аркуш	№ докум	Подпись	Дата		

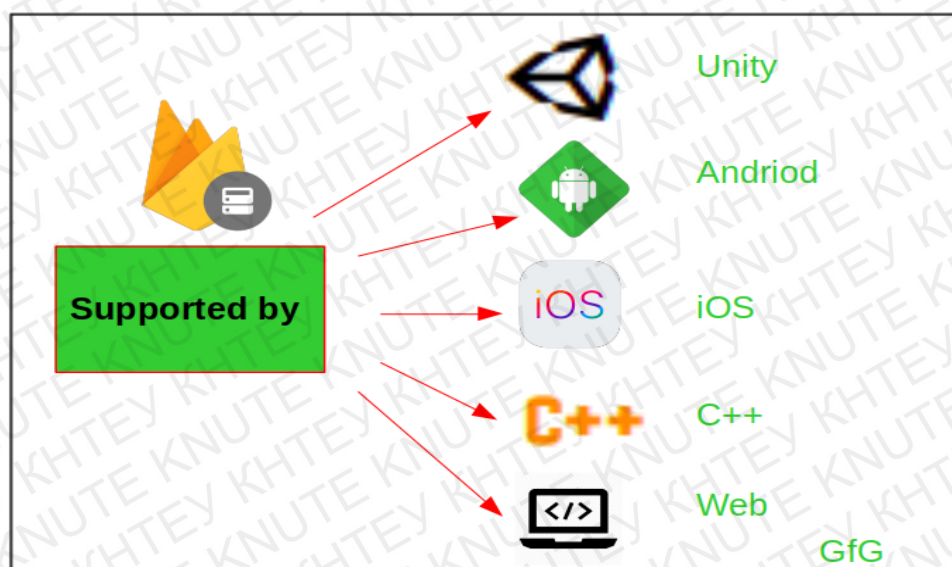
ASP.NET Core MVC. Веб-форми повністю вийшли в минуле. Програми ASP.NET Core підтримують програмні версії, в якій різні програми, що працюють на одному комп'ютері, можуть орієнтуватися на різні версії ASP.NET Core. Це не можливо з попередніми версіями ASP.NET Core.

AngularJS — JavaScript-фреймворк з відкритим програмним кодом, який розробляє Google. Призначений для розробки односторінкових додатків, що складаються з одної HTML сторінки з CSS і JavaScript. Його мета — розширення браузерних застосунків на основі шаблону Модель-вид-контролер (MVC), а також спрощення їх тестування та розробки.

Angular 2+ - це відкрита і вільна платформа для розробки веб-додатків, написана на мові TypeScript, що розробляється командою з компанії Google, а також спільнотою розробників з різних компаній. Angular - це повністю переписаний фреймворк від тієї ж команди, яка написала AngularJS.

Фреймворк працює зі сторінкою HTML, що містить додаткові атрибути і пов'язує області вводу або виводу сторінки з моделлю, яка є звичайною змінною TypeScript. Значення цих змінних задаються вручну або отримуються зі статичних або динамічних JSON-даних.

2.5. Огляд функціональних можливостей системи Firebase



					КНТЕУ 121-05-04.МР	Аркуш
						23
Изм.	Аркуш	№ докум	Подпись	Дата		

Рис. 2.7. Операційні системи що підтримуються Firebase

Firebase - це платформи розробки мобільних та веб- застосунків що має спектр служб та рішень.

Firebase Analytics

Firebase Analytics — безкоштовне рішення для оцінки додатків, яке дає змогу ознайомитись із використанням застосунків та залученням користувачів.

Firebase Cloud Messaging

Раніше відомий як Google Cloud Messaging (GCM), Firebase Cloud Messaging (FCM) — це крос-платформове рішення для повідомлень і нотифікацій для Android, iOS та веб-застосунків, які наразі можна використовувати безкоштовно.

Firebase Auth

Firebase Auth — це служба, яка може аутентифікувати користувачів, використовуючи лише код на стороні клієнта. Він підтримує соціальні логін-провайдери Facebook, GitHub, Twitter і Google (і Google Play Games). Крім того, вона включає в себе систему управління користувачами, за допомогою якої розробники можуть увімкнути автентифікацію користувача за допомогою входу з електронної пошти та пароля, що зберігаються в Firebase.

Realtime Database

Firebase надає в режимі реального часу базу даних та бекенд як службу. Ця служба надає розробникам застосунків API, який дозволяє синхронізувати дані застосунків між клієнтами та зберігати їх у хмарі Firebase. Компанія також надає клієнтські бібліотеки, які дозволяють інтеграцію із застосунками Android, iOS, JavaScript / Node.js, Java, Objective-C, Swift. База даних також доступна через REST API та прив'язки до декількох сценаріїв JavaScript, таких як AngularJS, React, Ember.js та Backbone.js. REST API використовує протокол подій із сервером, який є інтерфейсом для створення HTTP-з'єднань для отримання push-повідомлень від сервера. Розробники, які використовують Realtime Database, можуть захищати свої дані за допомогою правил безпеки, що застосовуються на сервері.

					КНТЕУ 121-05-04.MP	Аркуш
						24
Изм.	Аркуш	№ докум	Подпись	Дата		

Cloud Firestore, яка є наступною генерацією Firebase Realtime Database, була випущена у бета-версії.

Firebase Storage

Firebase Storage забезпечує надійне завантаження та вивантаження файлів для застосунків Firebase незалежно від якості мережі. Розробник може використовувати його для зберігання зображень, аудіо-, відео- чи іншого вмісту, створеного користувачами. Зберігання Firebase підтримується Google Cloud Storage.

Firebase Hosting та Functions

Firebase Hosting — це статичний та динамічний веб-хостинг, який було запущено 13 травня 2014 року. Він підтримує хостинг статичних файлів, таких як CSS, HTML, JavaScript та інші файли, а також динамічну підтримку Node.js через Cloud Functions. Служба передає файли через мережу доставки контенту (CDN) за допомогою протоколу HTTPS та шифрування SSL. Firebase підтримує Fastly, CDN, щоб забезпечити підтримку CDN Firebase Hosting. Компанія стверджує, що хостинг Firebase виріс із запитів клієнтів; розробники використовували Firebase для своєї бази даних в режимі реального часу, але вони потребували місця для розміщення їхнього вмісту.

З вищеперелічених сервісів для розробки індивідуального розкладу студента КНТЕУ використовуються Firebase Auth - для автентифікації та Realtime Database для зберігання та обмін даними.

2.6. Висновок до розділу 2

Зважаючи на поширеність та відкритість операційної системи Android для реалізації завдання зі створення додатку для отримання розкладу студента доцільним є розробка для цієї системи. В якості архітектурного рішення обраний Клієнт-Серверний шаблон що буде представлений серверною частиною яка приймає запити від клієнтів, зберігає та обробляє дані. Клієнти представлені мобільним Android додатком та Angular2+ веб додатком. Безумовною перевагою

					КНТЕУ 121-05-04.MP	Аркуш
						25
Ізм.	Аркуш	№ докум	Підпись	Дата		

Angular2+ над іншими Front-End фреймворками є використання мови TypeScript що забезпечує такі можливості:

- можливість явного визначення типів (статична типізація),
- підтримка використання повноцінних класів (як в традиційних об'єктно-орієнтованих мовах),
- підтримка підключення модулів.

Система розробки та підтримки додатків Firebase має широкі функціональні можливості що розробляти та публікувати кросплатформові додатки без створення власної серверної частини.

					КНТЕУ 121-05-04.МР	Аркуш
						26
Изм.	Аркуш	№ докум	Подпись	Дата		

РОЗДІЛ 3

РОЗРОБКА АРХІТЕКТУРИ ДОДАТКУ ДЛЯ ОТРИМАННЯ РОЗКЛАДУ СТУДЕНТА КНТЕУ

3.1. Проектування нереляційної бази даних

В світі технологій баз даних існує два основних напрямки: SQL і NoSQL, реляційні та нереляційні бази даних. Відмінності між ними полягають в тому, як вони спроектовані, які типи даних підтримують, як зберігають інформацію.

Про вибір SQL-баз даних

Хоча NoSQL-бази стали популярними завдяки швидкодії і хорошій масштабованості, в деяких ситуаціях кращими можуть виявитися структуровані SQL-сховища. Ось дві причини для вибору SQL-баз:

- Необхідність відповідності бази даних вимогам ACID (Atomicity, Consistency, Isolation, Durability - атомарність, несуперечливість, ізолюваність, довговічність). Це дозволяє зменшити ймовірність підозрілої поведінки системи і забезпечити цілісність бази даних, що досягається шляхом жорсткого визначення того, як саме транзакції взаємодіють з базою даних.

- Дані, з якими ви працюєте, структуровані, при цьому структура не схильна до частих модифікацій.

Про вибір NoSQL-баз даних

Якщо є підозри, що база даних може стати вузьким місцем якогось проекту, заснованого на роботі з великими обсягами інформації, варто віддати перевагу NoSQL-базам. Ось можливості, які стали причиною популярності NoSQL баз даних

					КНТЕУ 121-05-04. МР			
					Розробка програмного забезпечення для отримання розкладу студента КНТЕУ	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		РЗ	27	68
Зав. каф.	Криворучко							
Керівник	Цензура М.О.							
Гарант	Цензура М.О.							
Розробив	Білецький В.А.							
					Розділ 3	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5 група		

- Зберігання великих обсягів неструктурованої інформації. База даних NoSQL не накладає обмежень на типи даних, що зберігаються. Більш того, при необхідності в процесі роботи можна додавати нові типи даних.

- Використання хмарних обчислень і сховищ. Хмарні сховища - відмінне рішення, але вони вимагають, щоб дані можна було легко розподілити між кількома серверами для забезпечення масштабування. Використання, для тестування і розробки локального устаткування, а потім перенесення системи в хмару, де вона і працює - це саме те, для чого створені NoSQL бази даних.

- Швидка розробка. Якщо ви розробляєте систему, використовуючи agile-методи, застосування реляційної БД здатне уповільнити роботу. NoSQL бази даних не потребують того ж обсязі підготовчих дій, які зазвичай потрібні для реляційних баз.

При порівнянні SQL та NoSQL слід акцентувати увагу на головних відмінностях:

1. Масштабованість. Одна з головних відмінностей полягає в тому, що NoSQL краще піддаються масштабуванню. Наприклад, в MongoDB є влаштована підтримка реплікації та шардінгу (горизонтального поділу даних) для забезпечення масштабованості. В реляційних базах даних також підтримується масштабування, але це потребує значно більше витрат людських та апаратних ресурсів.

2. Індексція. У системах RDBMS індексція використовується для прискорення операцій отримання даних з баз. Відсутність індексу означає, що таблиця повинна бути переглянута цілком для того, щоб виконати запит на читання.

І в SQL, і в NoSQL-базах індекси застосовані для однієї і тієї ж мети - прискорити і оптимізувати вилучення даних. Але механізм їх роботи відрізняється через відмінності архітектури баз даних і особливостей зберігання інформації в базі.

					КНТЕУ 121-05-04.МР	Аркуш
						28
Ізм.	Аркуш	№ докум	Підпис	Дата		

3. CRM-системи. Всі розробники таких програм використовують і SQL, і NoSQL бази даних. І, хоча велика частина даних транзакцій все ще зберігається в SQL-базах, застосування знаходять загальнодоступні системи класу DBaaS (data-base-as-a-service, база даних як сервіс), на зразок AWS DynamoDB і Azure DocumentDB, в результаті, серйозне навантаження по обробці даних може бути перенесене в хмарні NoSQL-бази.

Головна перевага таких систем - використання неструктурованих даних, схожих на документи. Такі дані можуть подаватися на вхід статистичних моделей, які дають компаніям можливість виконувати різні види аналізу. CRM-програми, крім того, є досить вдалим прикладом, в якому дві системи баз даних виступають не конкурентами, а існують в гармонії, граючи кожна свою роль у великій архітектурі управління даними.

Ось ознаки проектів, для яких ідеально підійдуть SQL-бази:

- Є логічні вимоги до даних, які можуть бути визначені заздалегідь.
- Дуже важлива цілісність даних.

Потрібна заснована на усталених стандартах, добре зарекомендувала себе технологія, за допомогою якої можна розраховувати на великий досвід розробників та технічну підтримку.

А ось властивості проектів, для яких підійде щось зі сфери NoSQL:

- Вимоги до даних нечіткі, невизначені.
- Одні з основних вимог до бази даних - швидкість обробки даних і масштабованість.

Документи можуть бути організовані (згруповані) в колекції. Їх можна вважати віддаленим аналогом таблиць реляційних СУБД, але колекції можуть містити інші колекції. Хоча документи колекції можуть бути довільними, для ефективнішого індексування краще об'єднувати у колекцію документи зі схожою структурою/

Документо-орієнтовані бази даних застосовуються у системах керування вмістом, видавничій справі, документальному пошуку тощо. Приклади СУБД

					КНТЕУ 121-05-04.МР	Аркуш
						29
Ізм.	Аркуш	№ докум	Підпис	Дата		

даного типу - CouchDB, Couchbase, MarkLogic, MongoDB, eXist, Berkeley DB XML.

Документи всередині документо-орієнтованої бази даних у певному розумінні подібні до записів або рядків реляційних баз даних, але вимоги до них не такі жорсткі. Вони не потребують задовольняти стандартній схемі, ані мати однакові секції, частини або ключі. Наприклад, такий документ:

```
{
  Ім'я: "Микола",
  Адреса: "Коцюбинського, 5",
  Захоплення: "рибалка"
}
```

Рис. 3.1. Структура документу

Батьківський документ буде мати такий вигляд:

```
{
  Ім'я: "Петро",
  Адреса: "Панаса Мирного, 15",
  Діти: [
    {Ім'я: "Михайло", Вік: 10},
    {Ім'я: "Євгенія", Вік: 8},
    {Ім'я: "Одарка", Вік: 5},
    {Ім'я: "Олена", Вік: 2}
  ]
}
```

Рис. 3.2. Структура документу з вкладеними документами

При проектуванні NoSQL баз даних варто дотримуватись наступних принципів:

Уникання вкладення даних:

Зважаючи на те що Firebase Realtime Database дозволяє вкладення даних до 32 рівня, розробники можуть бути схильними вважати що таке проектування має місце.

Однак, при вибірці даних, з бази повертаються всі дочірні елементи. Також, при наданні доступу читання та запису до елементу бази даних, таким чином Ви надаєте доступ до всіх вкладених документів цього елемента.

Таким чином, на практиці, краще зберігати уникаючи вкладень.

					КНТЕУ 121-05-04.MP	Аркуш
						30
Ізм.	Аркуш	№ докум	Підпис	Дата		

Для прикладу чому вкладеність не є хорошим способом організації даних, можна розглянути наступну структуру:

{ Дана структура є неправильно спроектованою, оскільки для того аби отримати список заголовків бесід потрібно вибрати все дерево, завантажучи мегабайти даних.

```
"chats": { // бесіди
  "one": {
    "title": "Historical Tech Pioneers",
    "messages": {
      "m1": {"sender": "ghopper", "message": "Relay malfunction found. Cause: moth."},
      "m2": {...},
      // дуже довгий список повідомлень
    }
  },
  "two": {...}
}
```

З такою структурою, перебір даних стає проблематичним. Наприклад, перебір списку заголовків документа потребує вибірки всього дерева чатів, включно зі всіма вкладеними складовими.

Вирівняна структура даних (Денормалізація)

Коли дані, навпаки, розділені на окремі сутності, що також називається денормалізацією вони можуть бути ефективно завантажені окремими запитами, якщо це потрібно.

					КНТЕУ 121-05-04.МР	Аркуш
Изм.	Аркуш	№ докум	Підпис	Дата		31

Розглянемо вирівняну структуру:

```
{
  // Документ бесід зберігає тільки метадані
  "chats": {
    "one": {
      "title": "Historical Tech Pioneers",
      "lastMessage": "ghopper: Relay malfunction found. Cause: moth.",
      "timestamp": 1459361875666
    },
    "two": {... },
    "three": {... }
  },
  // учасники бесіди зберігаються в окремому документі
  "members": {
    "one": {
      "ghopper": true,
      "alovelace": true,
      "eclarke": true
    },
    "two": {... },
    "three": {... }
  },
  // Повідомлення зберігаються в окремому документі з унікальним ключем
  "messages": {
    "one": {
      "m1": {
        "name": "eclarke",
        "message": "The relay seems to be malfunctioning.",
        "timestamp": 1459361875337
      }
    }
  }
}
```

					КНТЕУ 121-05-04.МР	Аркуш
						32
Ізм.	Аркуш	№ докум	Підпис	Дата		

Після виконаної денормалізації стає можливо перебирати список кімнат, завантажуючи лише декілька байтів на бесіду.

Повідомлення можуть бути завантажені окремо і відображені окремо одразу коли вони надходять, дозволяючи користувацькому інтерфейсу бути адаптивним і швидкодієним.

Проектування сутностей бази даних розкладу студента

Для зберігання даних про розклад студента розробляється документо орієнтована база даних що має таку структуру:

Викладач - містить інформацію про викладача:

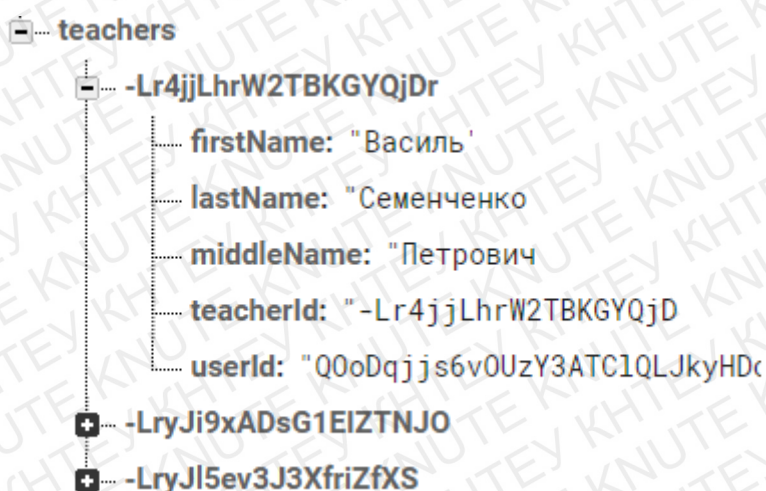


Рис. 3.3. Структура документу «Викладач»

1. firstName - ім'я
2. lastName - прізвище
3. middleName - по-батькові
4. teacherId - унікальний ідентифікатор
5. userId - ідентифікатор користувача

Предмет (subject)

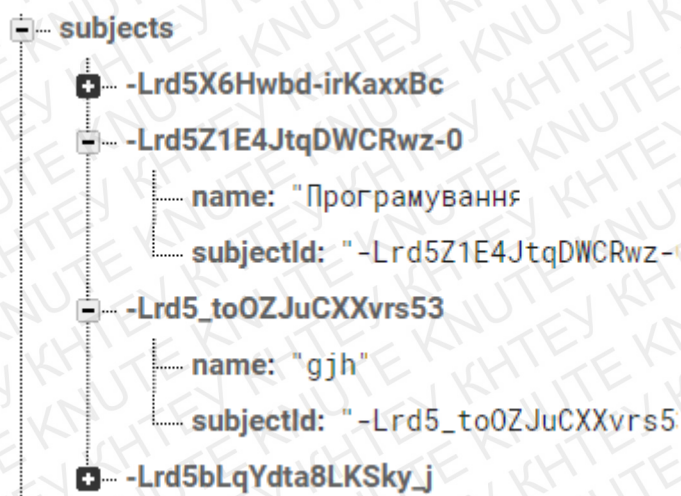


Рис. 3.4. Структура документу «Предмет»

name - назва предмета

subjectId - ідентифікатор предмета

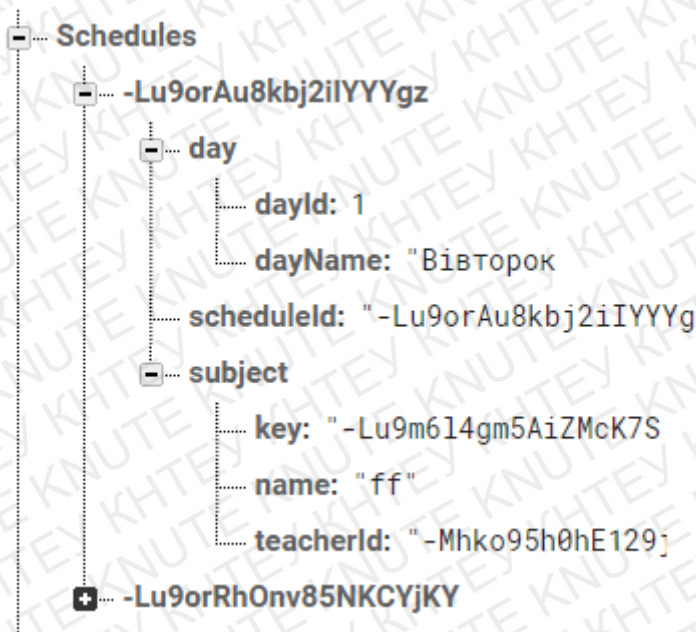


Рис. 3.5. Структура документу «Розклад»

Розклад (Schedule)

1. Ідентифікатор розкладу

2. День тижня:

- Ідентифікатор (dayId)
- Назва дня (dayName)

					КНТЕУ 121-05-04.MP	Аркуш
						34
Ізм.	Аркуш	№ докум	Підпис	Дата		

3. Предмет

- Ідентифікатор предмета (subjectId)
- Назва предмета (name)
- Ім'я викладача (teacherName)
- Ідентифікатор викладача (teacherSurname)

3.2. Розробка архітектури мобільного Android додатку

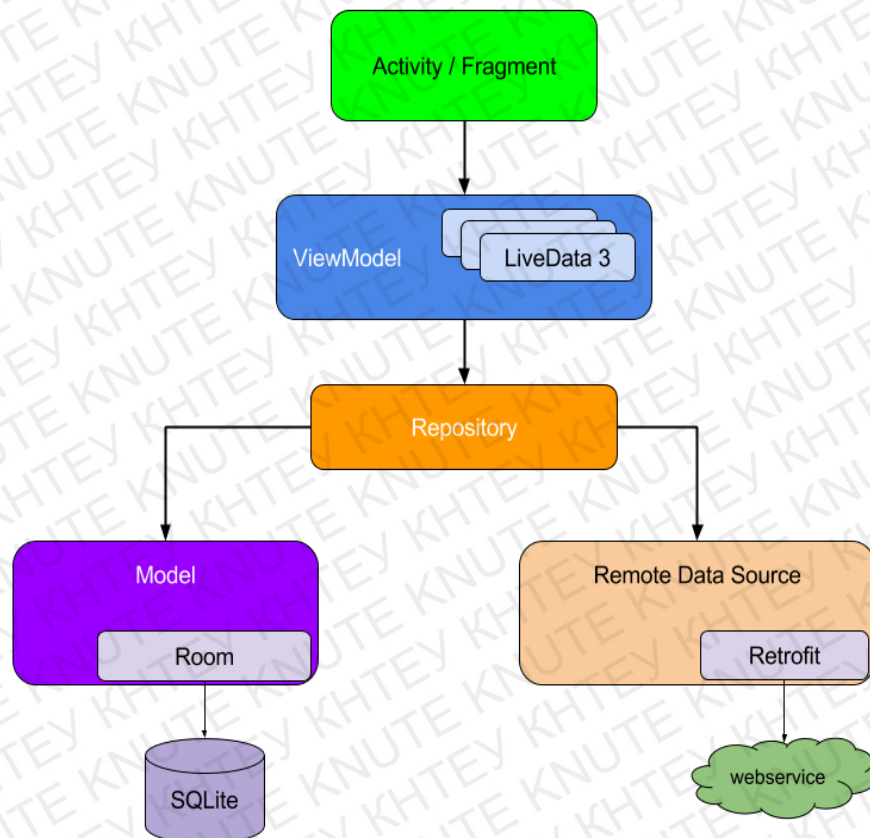


Рисунок 3.5. Архітектура системи Android

Основою всіх Android додатків є набір систем і служб:

1. Система представлень (View System) - це багатий набір уявлень функціональністю що розширюється, який служить для побудови зовнішнього вигляду додатків, що включає такі компоненти, як списки, таблиці, поля введення, кнопки і т.п.
2. Контент-провайдери (Content Providers) - це служби, які дозволяють додаткам отримувати доступ до даних інших додатків, а також надавати доступ до своїх даних.

					КНТЕУ 121-05-04.MP	Аркуш
						35
Ізм.	Аркуш	№ докум	Підпис	Дата		

3. Менеджер ресурсів (Resource Manager) призначений для доступу до строкових, графічних і іншим типам ресурсів.
4. Менеджер повідомлень (Notification Manager) дозволяє будь-якому додатком відображати призначені для користувача повідомлення в рядку статусу.
5. Менеджер дій (Activity Manager) управляє життєвим циклом додатків і надає систему навігації по історії роботи з діями.

Рівень бібліотек (Libraries)

Платформа Android включає набір C / C ++ бібліотек, використовуваних різними компонентами ОС. Для розробників доступ до функцій цих бібліотек реалізований через використання Application Framework. Нижче представлені деякі з них:

1. System C library - BSD-реалізація стандартної системної бібліотеки C (libc) для вбудованих пристроїв, заснованих на Linux.
2. Media Libraries - бібліотеки, засновані на PacketVideo's OpenCORE, призначені для підтримки програвання і записи популярних аудіо- і відео-форматів (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG і т.п.).
3. Surface Manager - менеджер поверхонь управляє доступом до підсистеми відображення 2D- і 3D- графічних шарів.
4. LibWebCore - сучасний засіб web-браузера, який надає всю міць вбудованого Android-браузера.
5. SGL - технологія для роботи з 2D-графікою.
6. 3D libraries - бібліотека для роботи з 3D-графікою, заснована на OpenGL ES 1.0 API.
7. FreeType - бібліотека, призначена для роботи зі шрифтами.
8. SQLite - потужний легкий засіб для роботи з реляційними БД.

Рівень середовища виконання (Android Runtime)

До складу Android входить набір бібліотек ядра, які надають більшу частину функціональності бібліотек ядра мови Java.

					КНТЕУ 121-05-04.MP	Аркуш
Изм.	Аркуш	№ докум	Підпис	Дата		36

Платформа використовує оптимізовану, реєстр-орієнтовану віртуальну машину Dalvik, на відміну від неї стандартна віртуальна машина Java - стек-орієнтована.

Кожна програма запускається в своєму власному процесі, зі своїм власним примірником віртуальної машини. Dalvik використовує формат Dalvik Executable (*.dex), оптимізований для мінімального використання пам'яті додатком.

Це забезпечується такими базовими функціями ядра Linux, як організація потокової обробки і низькорівневе управління пам'яттю. Байт-код Java, на якому написані додатки компілюються в dex-формат за допомогою утиліти dx, що входить до складу SDK.

Активність (Activity) є основою Android-інтерфейсу користувача. Кожен додаток може мати кілька активностей, що представляють для користувача різноманітні екрани.

Сервіс (Service) - забезпечує процеси що відбуваються на тлі та не представляють користувацького інтерфейсу.

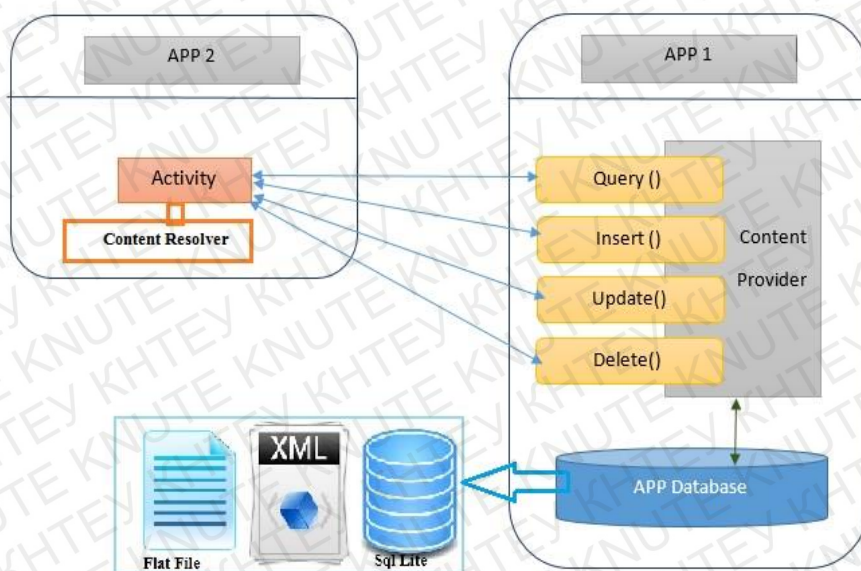


Рисунок 3.6. Принцип роботи постачальника вмісту

Постачальник вмісту (Content Provider) - пропонує можливості зберігання та пошуку даних для інших компонентів. Він може використовуватися для обміну та збереження даних в межах компонента додатку, та до інших додатків.

Трансляційний приймач (Broadcast Receiver) - асинхронно реагує на загальносистемні повідомлення передачі. Трансляційний приймач зазвичай є елементом котрий приймає дані із зовнішніх джерел та передає її до інших компонентів додатку, таких як активності або сервіси для подальшої обробки.

Android також підтримує компоненти нижчого рівня, які називаються фрагментами що надає ще один шар абстракції в рамках Активності.

Фрагмент функціонує у якості модульного розділу Активності, і є відповідальним за обробку подій пов'язаних зі складовими користувацького інтерфейсу. Кожен Фрагмент має власний життєвий цикл, отримує опрацьовує визначені для нього події та може бути доданим або видаленим з Активності під час її роботи. Компоненти ті всі додатки в Android контактують один з одним використовуючи різноманітні механізми, проте спеціалізована література не застосовує для зазначених компонентів визначення "конектори". Середовище виконання Android безліч типів конекторів, основні з яких:

Явні конектори на основі повідомлень (*Explicit Message-Based Connectors*) - дозволяють обмінюватись явними Інтентами (намірами) за допомогою механізму міжпроцесної взаємодії.

					КНТЕУ 121-05-04.МР	Аркуш
						38
Изм.	Аркуш	№ докум	Підпис	Дата		

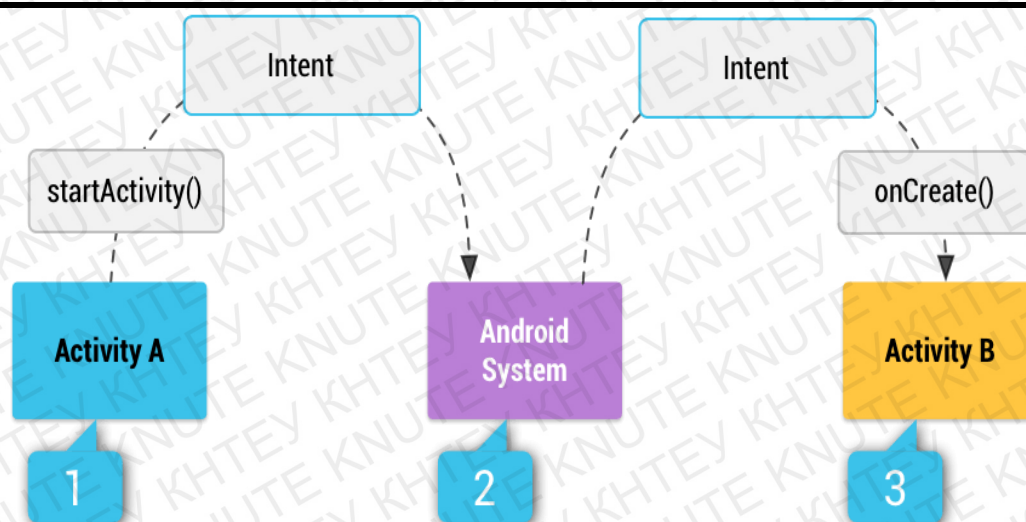


Рисунок 3.7. Ілюстрація того як неявний намір взаємодіє з системою аби запустити іншу Активність: 1. Активність А створює Намір з описом події та передає її як аргумент в метод startActivity() 2. Android система здійснює пошук для того щоб знайти фільтр що відповідає надісланому Наміру. 3. Коли фільтр знайдений, система зв'язується з активністю, викликаючи її метод onCreate() та передає туди Намір.

Повідомлення про намір (Intent) є носієм корисної інформації разом з метаданими що її описує. Мета дані містять події, дані та категорії що вказують на дію яка повинна бути здійснена компонентом отримувачем, тип корисної інформації що передається та вид компонента котрий оброблятиме намір. Явним наміром є той, що наочно зазначає ім'я отримувача повідомлення.

Неявні конектори на основі повідомлень (*Implicit Message-Based Connectors*) - дозволяють обмінюватися неявними намірами ґрунтуючись на Android механізмі ІРС. Неявний конектор є той що не зазначає ім'я отримувача при надсиланні повідомлення, це відбувається тоді коли, його мета дані співпадають з такими що задекларовані у фільтрах наміру компонента-отримувача. Фільтри намірів (Intent-filters) описуєть види запитів на які може відповісти отримувач.

Конектор доступу даних (*Data Access Connector*) по структурі відповідає Постачальнику даних котрий пропонує інтерфейс для додавання, видалення,

і отримання даних що зберігаються.

Віддалені процедурні виклики (*Remote Procedure Calls*) - реалізують взаємодію між процесами IPC за допомогою взаємодії через методи виклику, використовуючи заглушки (stubs) що автоматично генеруються зі специфікації інтерфейсів компонента мовою опису інтерфейсів (Android Interface Definition Language).

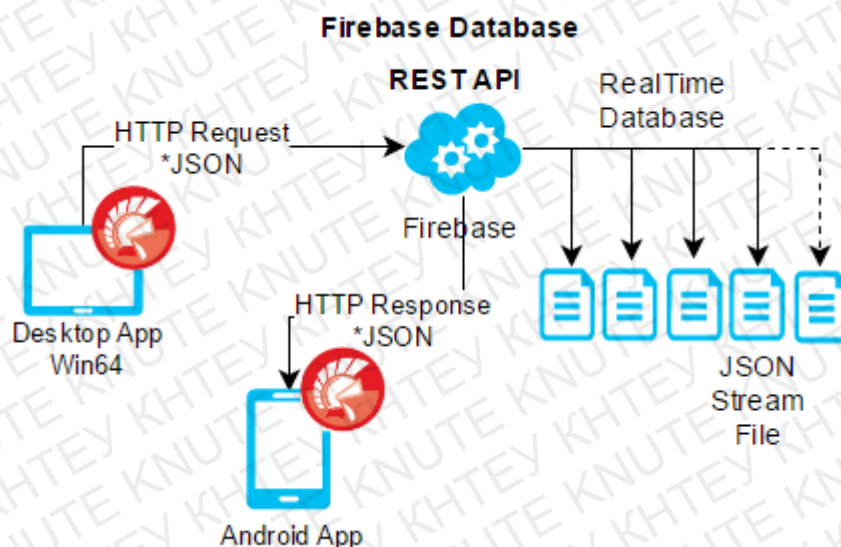


Рис. 3.8. Взаємодія між клієнтами ма базою даних в системі Firebase

Мобільний додаток є одним з клієнтів для прийому та надсилення запитів на сервер що реалізується засобами системи Firebase. Android додаток частково реалізує архітектурний шаблон MVC в якому роль представлення відіграють xml файли з розміткою що відображає інтерфейс сторінок додатку. Представлення реагує на дії користувача та надсилає запит в компонент активності з якою зв'язане представлення де для отримання даних викликаються функції репозиторію даних: отримання, створення, видалення та редагування даних. В тілі кожної з функцій, в свою чергу, здійснюються запити на Firebase сервер.

					КНТЕУ 121-05-04.MP	Аркуш
Изм.	Аркуш	№ докум	Підпис	Дата		40

3.3. Розробка архітектури веб додатку на базі фреймворку Angular2

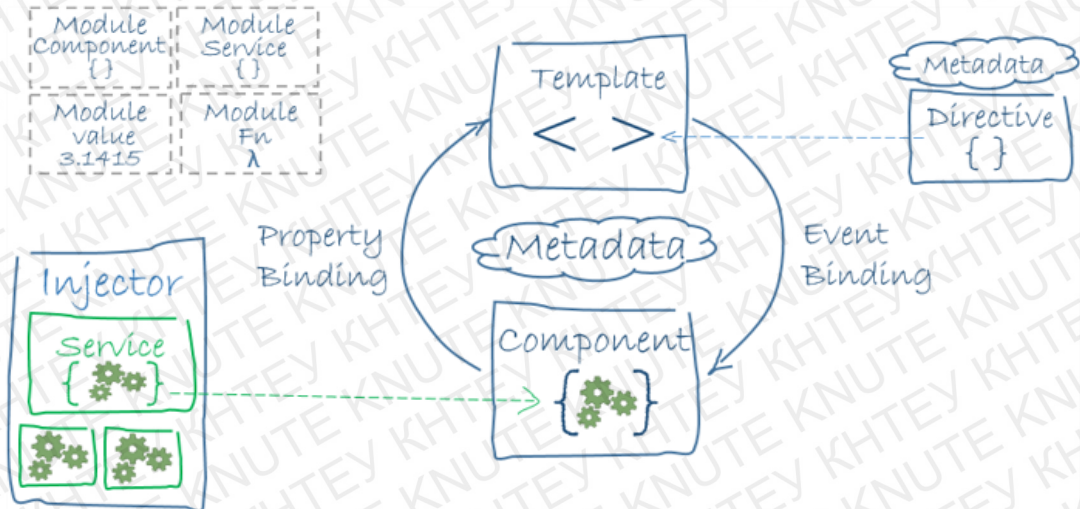


Рис.3.9. Складові проекту Angular2

Angular – це платформа та фреймворк для створення клієнтських додатків з допомогою TypeScript та HTML. Angular написаний на TypeScript. Базовими блоками архітектури Angular додатку є NgModule модулі що забезпечують контекст компіляції для компонентів. Angular додаток представлений набором модулів, кожен додаток має щонайменше один модуль root module що запускає програму та зазвичай має багато інших модулів.

- Компоненти визначають представлення, що містять елементи інтерфейсу і можуть бути змінені в залежності від бізнес логіки.
- Компоненти використовують сервіси що забезпечують спеціальні функції що не пов'язані з представленнями напряду.

Як компоненти так і сервіси є класами з декораторами в яких зазначаються метадані про їхній тип та інформація про застосування

- Метадані класу компонента асоціюють його з шаблоном який представляє відображає інтерфейс користувача. Шаблон поєднує HTML розмітку з Angular директивами що дозволяють модифікувати HTML перед його відображенням.

Модулі

Кожен додаток Angular має кореневий модуль, умовно названий AppModule, який забезпечує механізм завантаження, що запускає додаток. Додаток, як правило, містить багато функціональних модулів.

Як і модулі JavaScript, NgModules можуть імпортувати функціональність з інших NgModules та дозволяти експортувати та використовувати їх власні функціональні можливості іншими NgModules. Наприклад, щоб використовувати послугу маршрутизатора у вашій програмі, ви імпортуєте маршрутизатор NgModule.

Компоненти

Кожна програма Angular має щонайменше один компонент, кореневий компонент, який з'єднує ієрархію компонентів із моделлю об'єкта сторінки документа (DOM). Кожен компонент визначає клас, який містить дані програми та логіку, і асоціюється з HTML-шаблоном, який визначає подання для відображення в цільовому середовищі.

Декоратор @Component () ідентифікує клас безпосередньо під ним як компонент та надає шаблон та пов'язані з ним специфічні метадані.

Сервіси та ін'єкція залежності

Для даних або логіки, які не пов'язані з певним представленням, створюється клас сервісу. Визначенню класу сервісу безпосередньо передую декоратор @Injectable (). Декоратор надає метадані, що дозволяють іншим постачальникам вводитись як залежність у клас.

Ін'єкційна залежність Dependency Injection (DI) дозволяє зберігати класи компонентів худорлявими та ефективними. Вони не отримують дані з сервера, не перевіряють введення користувачів або входять безпосередньо в консоль; вони делегують такі завдання службам.

					КНТЕУ 121-05-04.MP	Аркуш
						42
Ізм.	Аркуш	№ докум	Підпис	Дата		

3.4. Висновок до розділу 3

В якості системи керування даними проекту предметної області була обрана NoSQL база даних.

Основними причинами популярності нереляційних документно-орієнтованих баз даних є:

- Зберігання великих обсягів неструктурованої інформації
- Використання хмарних обчислень і сховищ
- Швидка розробка

Було проведено дослідження основних принципів проектування NoSQL баз даних та сформована структура сутностей проекту: «Розклад», «Предмет», «Викладач», «Календар».

Зважаючи на розвиненість та перспективність веб технологій найоптимальнішим рішенням автоматизації отримання та управління розкладом студента є розподілена структура, що складається з Мобільного додатку та Веб рішення що синхронізуються з базою даних системи Firebase в режимі реального часу.

Проведене дослідження базових архітектурних рішень системи Android, визначені ключові структурні елементи системи та принципи взаємодії з ними.

Разом з тим надана детальна характеристика структурних елементів платформи Angular2.

Проведена робота дозволяє надає усі підстави для початку процесу програмування ПЗ предметної області.

					КНТЕУ 121-05-04.MP	Аркуш
						43
Изм.	Аркуш	№ докум	Підпис	Дата		

РОЗДІЛ 4

РОЗРОБКА РОЗПОДІЛЕНОГО ДОДАТКУ ДЛЯ ОТРИМАННЯ ДОСТУПУ ДО РОЗКЛАДУ СТУДЕНТА КНТЕУ МОВАМИ ПРОГРАМУВАННЯ JAVA ТА TYPESCRIPT

4.1. Інтеграція робочих проектів з системою Firebase

Відповідно до технічного завдання розробляється розподілене програмне забезпечення представлене мобільним Android додатком та веб рішенням на базі Angular2. Роль зберігання та обробки даних покладено на можливості системи Firebase.

Перед створенням проекту в Firebase необхідна наявність робочих проектів, тому перед першим кроком є створення пустих проектів для Android та Angular2.

Далі можна створити робочий проект в Firebase консолі, перейшовши за посиланням <https://console.firebase.google.com/> що містить перелік проектів

					КНТЕУ 121-05-04. МР			
					Розробка програмного забезпечення для отримання розкладу студента КНТЕУ	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Р4	44	68
Зав. каф.	Криворучко О.В.					Факультет обліку, аудиту та інформаційних систем, 2 курс, 5 група		
Керівник	Цензура М.О.							
Гарант	Цензура М.О.							
Розробив	Білецький В.А.							
					Розробка розподіленого додатку для отримання доступу до розкладу студента кнтеу мовами програмування java та typescript			

Необхідно натиснути кнопку *Додати проект* (англ.: *Add Project*)

Потім потрібно ввести назву проекту, після чого ознайомитися з умовами використання системи та вказати місцезнаходження.

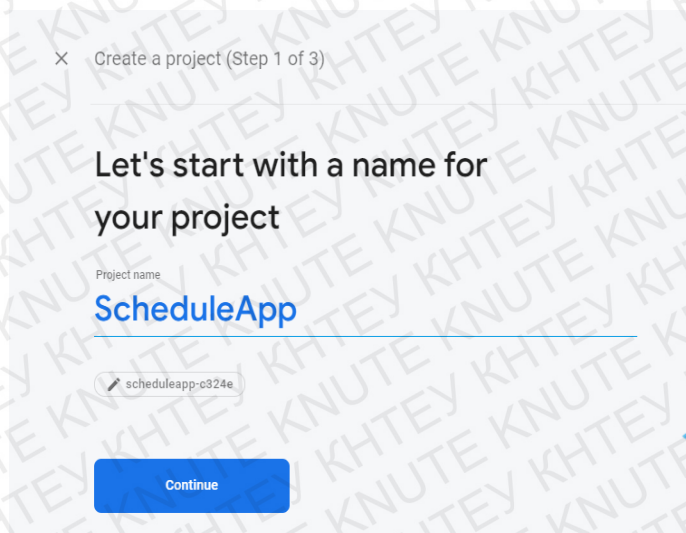


Рис. 4.1. Створення проекту в системі Firebase

Після додавання нового проекту до Firebase консолі можна почати процес налаштування інтеграції системи з проектами що розробляються.

На головній сторінці присутня панель з переліком додатків доступних для інтеграції, серед яких: iOS, Android, Веб додатки, та Unity.

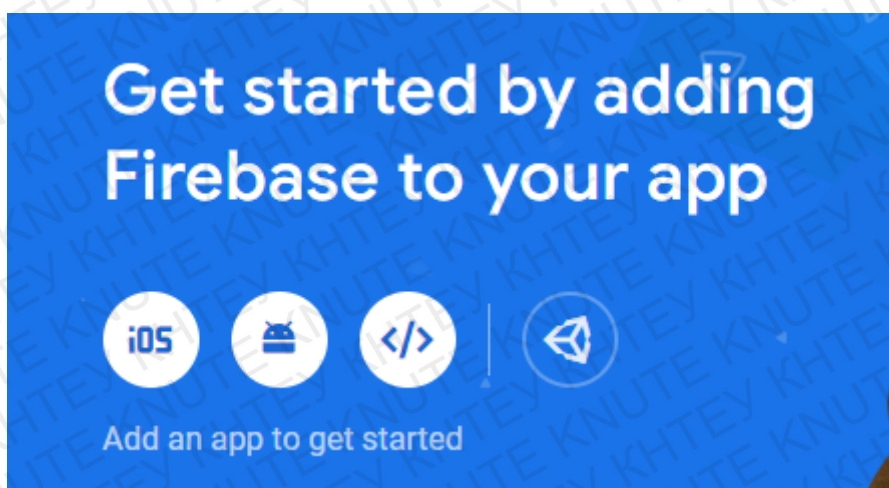


Рис. 4.2. Перелік типів проектів в системі Firebase

Отже спершу обираємо інтеграцію з Android. Після чого переходимо на сторінку реєстрації додатку, де необхідно заповнити форму даними про назву

					КНТЕУ 121-05-04.MP	Аркуш
						45
Изм.	Аркуш	№ докум	Подпись	Дата		

Android пакету, псевдонім додатку, наступним кроком є завантаження конфігураційного файлу google-services.json який необхідно додати в теку app робочого Android проекту. Даний файл містить json конфігурації що представлені трьома об'єктами:

project_info – інформація про проект, має такі властивості:

project_number – номер проекту

firebase_url – url проекту на firebase

project_id – ідентифікатор проекту

storage_bucket – адреса доступу до даних

client – зберігає інформацію про клієнта

services – масив об'єктів що зберігає інформацію про підключені сервіси.

Після додавання файлу конфігурацій необхідно скопіювати проект.

Після завершення інтеграції Android проекту можна почати інтеграцію з Angular2 для цього необхідно повернутися на основну сторінку консолі та обрати створення проекту для веб рішень. Далі заповнивши форму про назву додатку можна отримати налаштування для інтеграції, які додаються блоком скрипту до HTML шаблону компонента Angular2.

					КНТЕУ 121-05-04.MP	Аркуш
						46
Изм.	Аркуш	№ докум	Подпись	Дата		

```

<!-- The core Firebase JS SDK is always required and must be listed first
<script src="https://www.gstatic.com/firebasejs/7.5.0/firebase-app.js">

<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#available-libraries -->
<script src="https://www.gstatic.com/firebasejs/7.5.0/firebase-analytics">

<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyA8G1XIeeydQfJ2Bppdrybrfhiv4ofSD1k",
    authDomain: "scheduleapp-c324e.firebaseio.com",
    databaseURL: "https://scheduleapp-c324e.firebaseio.com",
    projectId: "scheduleapp-c324e",
    storageBucket: "scheduleapp-c324e.appspot.com",
    messagingSenderId: "1091010118962",
    appId: "1:1091010118962:web:6f23af77d8c67df30d432b",
    measurementId: "G-B9KV0RGV6J"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
  firebase.analytics();
</script>

```

Рис. 4.3. Конфігурації Firebase для проекту Angular2

4.2. Середовище та інструменти розробки мобільного додатку

Мобільний додаток предметної області розробляється на основі сім'ї технологій Java, зокрема весь код програми написаний мовою Java.

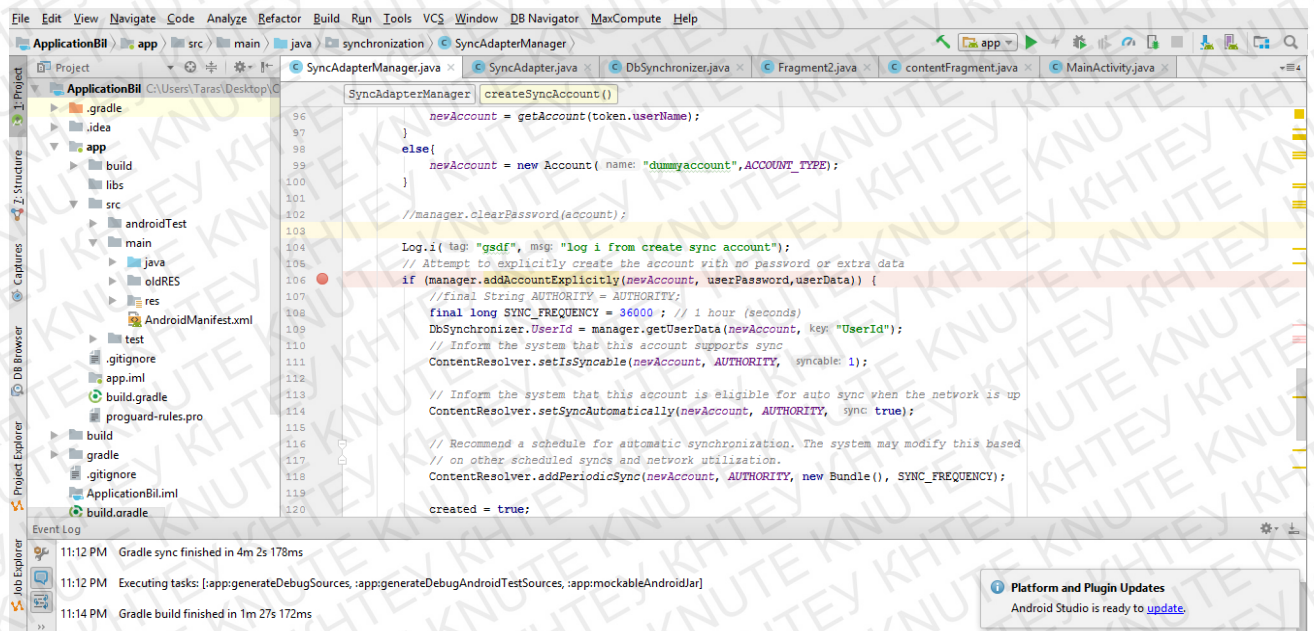


Рисунок 4.4. вікно середовища розробки Android Studio

					<i>КНТЕУ 121-05-04.МР</i>	Аркуш
						47
Изм.	Аркуш	№ докум	Подпись	Дата		

Android Studio — інтегроване середовище розробки (IDE) для платформи Android, представлене 16 травня 2013 року на конференції Google I/O менеджером по продукції корпорації Google — Еллі Паверс (англ. Ellie Powers). 8 грудня 2014 року компанія Google випустила перший стабільний реліз Android Studio 1.0

Для прискорення розробки застосунків представлена колекція типових елементів інтерфейсу і візуальний редактор для їхнього компонування, що надає зручний попередній перегляд різних станів інтерфейсу застосунку (наприклад, можна подивитися як інтерфейс буде виглядати для різних версій Android і для різних розмірів екрану). Для створення нестандартних інтерфейсів присутній майстер створення власних елементів оформлення, що підтримує використання шаблонів. У середовищі вбудовані функції завантаження типових прикладів коду з GitHub.

Після створення проекту структура проекту Android відображається в наступному вигляді:

					КНТЕУ 121-05-04.МР	Аркуш
						48
Изм.	Аркуш	№ докум	Подпись	Дата		

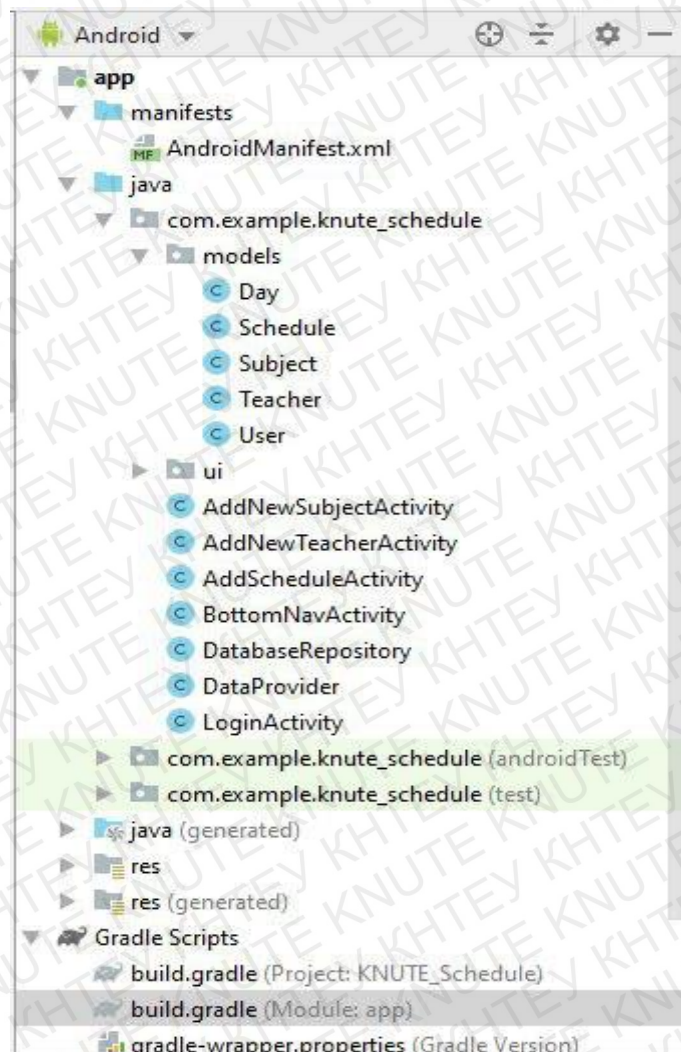


Рис. 4.5. Структура проекту для ОС Андроїд

Проект Android може складатися з різних модулів. За замовчуванням, створюється один модуль - `app`.

Модуль має три папки:

manifests: зберігає файл маніфесту `AndroidManifest.xml`, який визначає конфігурацію

java: зберігає файли коду на мові `java`, які структуровані по окремих пакетах

res: містить використовувані в додатку ресурси

Окремий елемент *Gradle Scripts* містить ряд скриптів `gradle` (як для модуля `app` або інших можливих модулів, так і для всього проекту), які використовуються при побудові програми.

					КНТЕУ 121-05-04.МР	Аркуш
						49
Изм.	Аркуш	№ докум	Подпись	Дата		

Перейдемо до повної структури проекту. Для цього можна подвійним клацанням натиснути на назву проекту.

Всі модулі в проекті описуються файлом `setting.gradle`. За замовчуванням він має наступний вміст: `«include ':app'»`

Файл `build.gradle` містить інформацію, яка використовується при побудові проекту.

Кожен модуль має свій файл `build.gradle`, який визначає конфігурацію побудови проекту, специфічну для даного модуля. Так, якщо ми подивимося на вміст папки `app`, то як раз знайдемо в ній такий файл. На початковому етапі дані файли не настільки важливі, досить лише розуміти, для чого вони потрібні.

У модулі `app` ми можемо побачити кілька папок і файлів, з яких для нас найважливішими є:

-*каталог `libs`* - призначений для зберігання бібліотек, використовуваних додатком

-*каталог `src`* - призначений для зберігання вихідного коду. Він містить ряд підкаталогів. Каталоги `androidTest` і `test` призначені для зберігання файлів тестів додатку. А власне вихідні коди розташовуються в папці `main`.

Папка `main` має складну структуру:

`AndroidManifest.xml` представляє файл маніфесту, який описує фундаментальні характеристики додатку, його конфігурацію і визначає кожен з компонентів цього додатка.

Папка `java` містить вихідні файли програми. За замовчуванням в ньому знаходиться файл класу `MainActivity`, який запускається за замовчуванням при старті програми.

Папка `res` містить каталоги з ресурсами. Зокрема, такі каталоги:

папка *`drawable`* призначена для зберігання зображень, використовуваних в додатку

					КНТЕУ 121-05-04.MP	Аркуш
						50
Изм.	Аркуш	№ докум	Подпись	Дата		

папка *layout* призначена для зберігання файлів, що визначають графічний інтерфейс. За замовчуванням тут є файл *activity_main.xml*, який визначає інтерфейс для єдиною в проєкті *activity* - *MainActivity*

папки *tiptar-xxxx* містять файли зображень, які призначені для створення іконки програми при різних дозволах екрану. І для кожного виду дозволу тут є свій каталог

папка *values* зберігає різні *xml*-файли, що містять колекції ресурсів - різних даних, які застосовуються в додатку.

Для емуляції мобільного пристрою під час розробки, використовується система Genymotion що надає широкий перелік пристроїв.

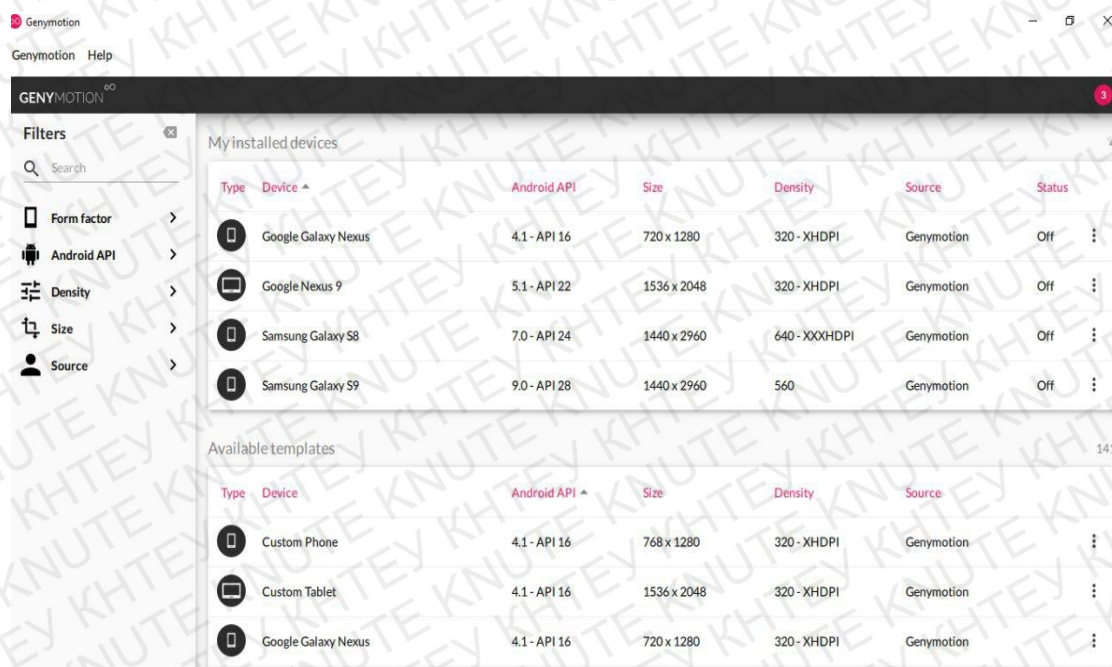


Рис. 4.5. Перелік доступних емуляторів в Genymotion

З наданого переліку можна обрати потрібний пристрій та запустити його. Після чого відкриється вікно програми яку можна використовувати як справжній смартфон, встановлюючи та користуючись додатком що розробляється.

4.3. Програмування мобільного додатку мовою Java

Керування користувачами додатку

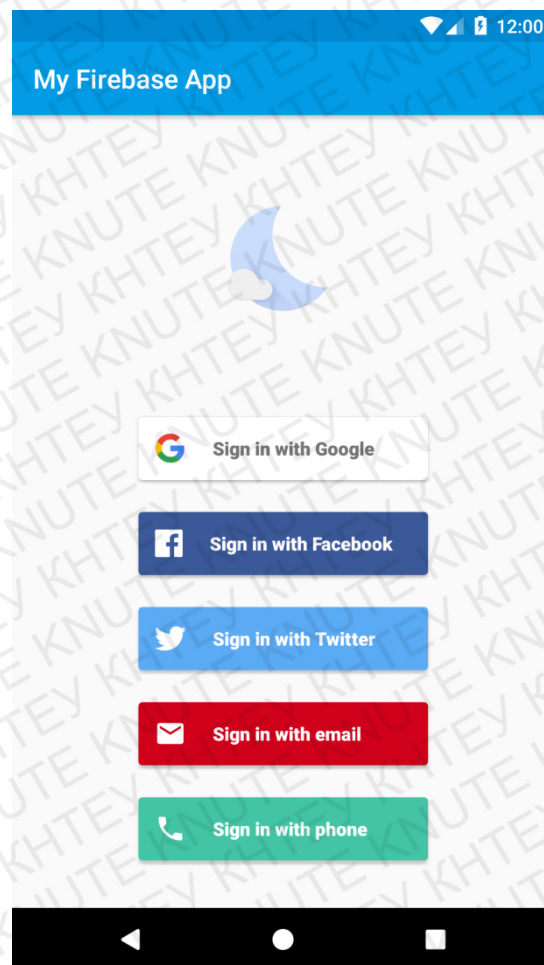


Рис. 4.6. Сторінка авторизації користувачів FirebaseUI

Більшість додатків потребують інформації про користувача.

Ідентифікація користувача дозволяє додатку безпечно зберігати його дані в хмарі та надавати персоналізовані функції в додатку на всіх пристроях.

Firebase автентифікація забезпечує легкі до впровадження серверні сервіси та готовий інтерфейс для автентифікації користувачів.

					КНТЕУ 121-05-04.MP	Аркуш
						52
Изм.	Аркуш	№ докум	Подпись	Дата		

Підтримується автентифікація використовуючи пароль, номер телефону, популярні способи ідентифікації: Google, Facebook, Twitter та інші.

Є можливість авторизувати користувача через спеціальний інтерфейс Firebase або використовуючи власний шаблон узгодивши його з Firebase Authentication SDK.

FirebaseUI Auth - є рекомендованим способом реалізації модуля автентифікації користувача, що пропонує зручне впровадження найбільш безпечних і перевірених прикитик.

Дане рішення здійснює взаємодію інтерфейсу під час авторизації користувача. Окрім реєстрації та авторизації компонент пропонує реалізацію відновлення та заміни паролю.

Всі візуальні представлення компонента можуть бути легко підлаштованими до вимог та задач розробника і користувача.

Щоб авторизувати користувача спершу порібно отримати інформацію від нього.

Такою інформацією можуть бути email користувача та пароль або OAuth токен доступу від одного з постачальників (Facebook, Twitter ітд)

Потім ці дані передаються до на сервер Firebase, де вони верифікуютьс та після чого повертають відповідь клієнту.

Після успішної авторизації відкривається доступ до інформації користувача. Користувачі у Firebase.

Профіль зареєстрованого користувача в проекті представлений об'єктом що має низку властивостей.

Користувачі Firebase мають фіксований набір властивостей - унікальний ідентифікатор, email, ім'я та URL світлини - що зберігається в базі даних проекта і може бути зміненим користувачем.

Розробник не може додати інші властивості до об'єкту напряму, натомість, це вони можуть зберігатися в будь-якому іншому сховищі, такому як Google Cloud Firestore.

					КНТЕУ 121-05-04.MP	Аркуш
						53
Изм.	Аркуш	№ докум	Подпись	Дата		

В момент реєстрації користувача в додатку, профіль користувача наповнюється такою інформацією:

- Ознака того чи користувач зареєструвався використовуючи email та пароль
- Ознака того чи користувач зареєструвався через інший провайдер ідентифікації (Google, Facebook ітд)
- Ознака того чи користувач зареєструвався через власноруч розроблений модуль ідентифікації

Поточний користувач.

Коли користувач зареєструвався або авторизувався, він стає поточним користувачем Auth змінної (або об'єкту автентифікації).

Змінна зберігає стан користувача так що після оновлення сторінки (в браузері) або перезавантаженні додатку ці дані не втрачаються.

Коли користувач виходить з профіля, змінна перестає зберігати посилання на об'єкт користувача відтак втрачає ознаку його стану; поточний користувач зникає.

Життєвий цикл користувача

Рекомендований спосіб відслідковувати поточний стан Auth змінної - використовувати функції-слухачі(listeners).

Слухачі отримують сповіщення щоразу коли щось пов'язане з ними змінюється.

Слухачі спрацьовують в таких ситуаціях:

- Користувач увійшов
- Користувач вийшов
- Токен доступу поточного користувача оновився
- вийшов термін придатності токена
- користувач змінює пароль

					КНТЕУ 121-05-04.MP	Аркуш
						54
Изм.	Аркуш	№ докум	Подпись	Дата		

- користувач повторно авторизується. Деякі дії потребують повторної авторизації, такі як видалення профілю, зміна пароля або інших даних користувача.

На практиці модуль реалізується створенням файлу активності LoginActivity

В якому визначаються основні функції автентифікації користувача:

В методі onCreate що викликається в момент створення активності відбувається перевірка поточного статусу користувача, для цього використовуються методи об'єкту FirebaseAuth що доступний завдяки підключенню залежності com.google.firebase:firebase-auth:19.0.0.

```
mFirebaseAuth = FirebaseAuth.getInstance();

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
if (user != null) {
    Intent intent = new Intent( packageContext: this, BottomNavActivity.class);
    startActivity(intent);
    finish();
    return;
}

List<AuthUI.IdpConfig> providers = Arrays.asList(
    new AuthUI.IdpConfig.EmailBuilder().build()
);

Intent intent = AuthUI.getInstance() AuthUI
    .createSignInIntentBuilder() SignInIntentBuilder
    .setIsSmartLockEnabled(!BuildConfig.DEBUG) SignInIntentBuilder
    .setAvailableProviders(providers) SignInIntentBuilder
    .setLogo(R.mipmap.ic_launcher_round) SignInIntentBuilder
    .build();
startActivityForResult(intent, RC_SIGN_IN);
```

Рис. 4.7. Алгоритм авторизації користувача мовою програмування Java
Спочатку створюється об'єкт сервісу.

Далі створюється об'єкт поточного користувача. Після чого відбувається перевірка об'єкту користувача, якщо значення змінної дорівнює null це свідчить про те що поточний користувач не авторизований в системі Firebase відповідно алгоритм ініціює процес авторизації користувача та перехід до імпортованої активності авторизації FirebaseAuthUI.

					КНТЕУ 121-05-04.MP	Аркуш
						55
Изм.	Аркуш	№ докум	Подпись	Дата		

В ситуації коли змінна має значення відмінне від null виконується перехід на основну сторінку додатку.

База даних подається у вигляді класів Java кожен клас містить перелік властивостей що відповідають властивостям документів NoSQL бази даних на сервері Firebase.

```
6  @IgnoreExtraProperties
7  public class Day {
8
9      public int dayId;
10     public String dayName;
11
12     @
13     public Day () {
14
15     }
16
17     @
18     public Day (int dayId, String dayName) {
19         this.dayId = dayId;
20         this.dayName = dayName;
21     }
22
23     @Override
24     public String toString() { return dayName; }
25
26 }
```

Рис. 4.8. Модель даних «День тижня»

Бізнес логіка щодо управління даними визначається в контейнерах що включають фрагменти, кожен фрагмент містить методи для додавання, видалення, оновлення, видалення інформації БД. Для відслідковування в реальному часі змін іншими клієнтами інформації БД в фрагменті визначаються методи що підписуються що спеціальних подій об'єкту FirebaseDatabase.

Основними подіями є:

addListenerForSingleValueEvent – подія що викликається перший раз при переході на фрагмент. Існує для ініціалізації елементів інтерфейсу даними з БД.

addValueEventListener – подія що викликається щоразу коли якийсь із клієнтів змінює інформацію БД (додає, видаляє, редагує, тощо).

Основними елементами інтерфейсу є випадний список ExpandableListView та звичайний список ListView.

					КНТЕУ 121-05-04.MP	Аркуш
						56
Изм.	Аркуш	№ докум	Подпись	Дата		

Для реалізації випадного списку необхідно створити адаптер в якому дані розподіляються на основний та підпорядкований масиви даних, що дозволяє зручно відображати дані та отримувати здійснювати взаємодію між діями користувача та обробкою даних. Адаптер CustomExpandableListAdapter повинен наслідуватися від базового класу BaseExpandableListAdapter та реалізовувати методи:

getGroup(int listPosition) – отримує індекс списку та повертає об'єкт елементу списку основного масиву

getChild(int listPosition, int expandedListPosition) – отримує позицію елементу основного списку та підпорядкуваного, повертає об'єкт елементу підпорядкуваного списку.

View getView(int listPosition, boolean isExpanded, View, ViewGroup parent) - отримує інформацію про поточну позицію елемента списку даних, ознаку розгорнутості елемента інтерфейсу та посилання на шаблон елемента представлення. В тілі методу ініціалізує об'єкт елемента інтерфейсу та повертає цей об'єкт.

View getChildView(int listPosition, final int expandedListPosition, ViewGroup parent) – отримує позиції елементів основного та підпорядкуваного масивів, посилання на елемент інтерфейсу основного списку.

					КНТЕУ 121-05-04.МР	Аркуш
						57
Изм.	Аркуш	№ докум	Подпись	Дата		

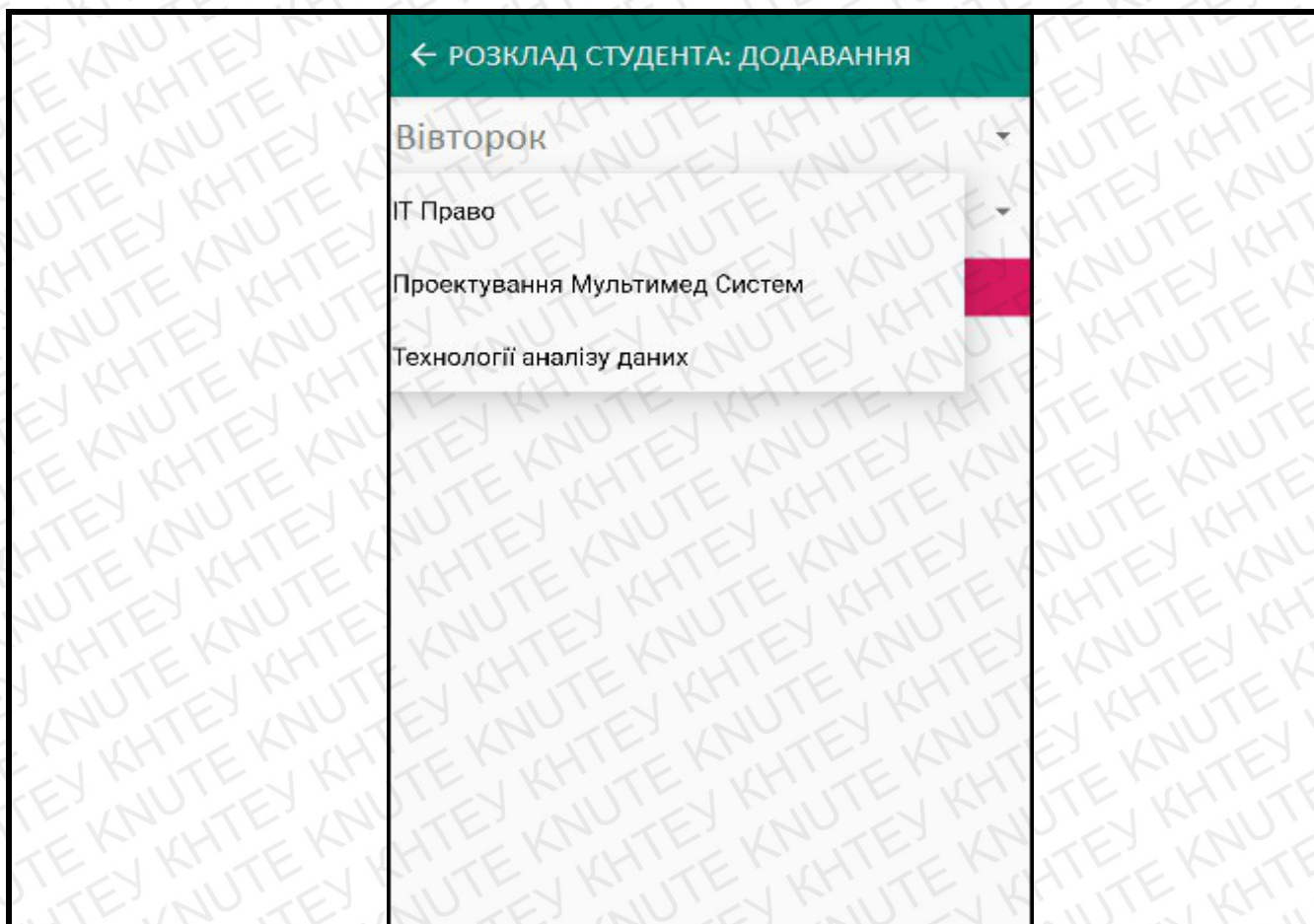


Рис. 4.9. Сторінка додавання розкладу студента на день тижня

Операції додавання або редагування даних відбуваються в спеціальних фрагментах де здійснюється збір інформації від користувача та її передача на сервер Firebase.

					КНТЕУ 121-05-04.MP	Аркуш
						58
Изм.	Аркуш	№ докум	Подпись	Дата		

4.4. Середовище та інструменти розробки веб додатку

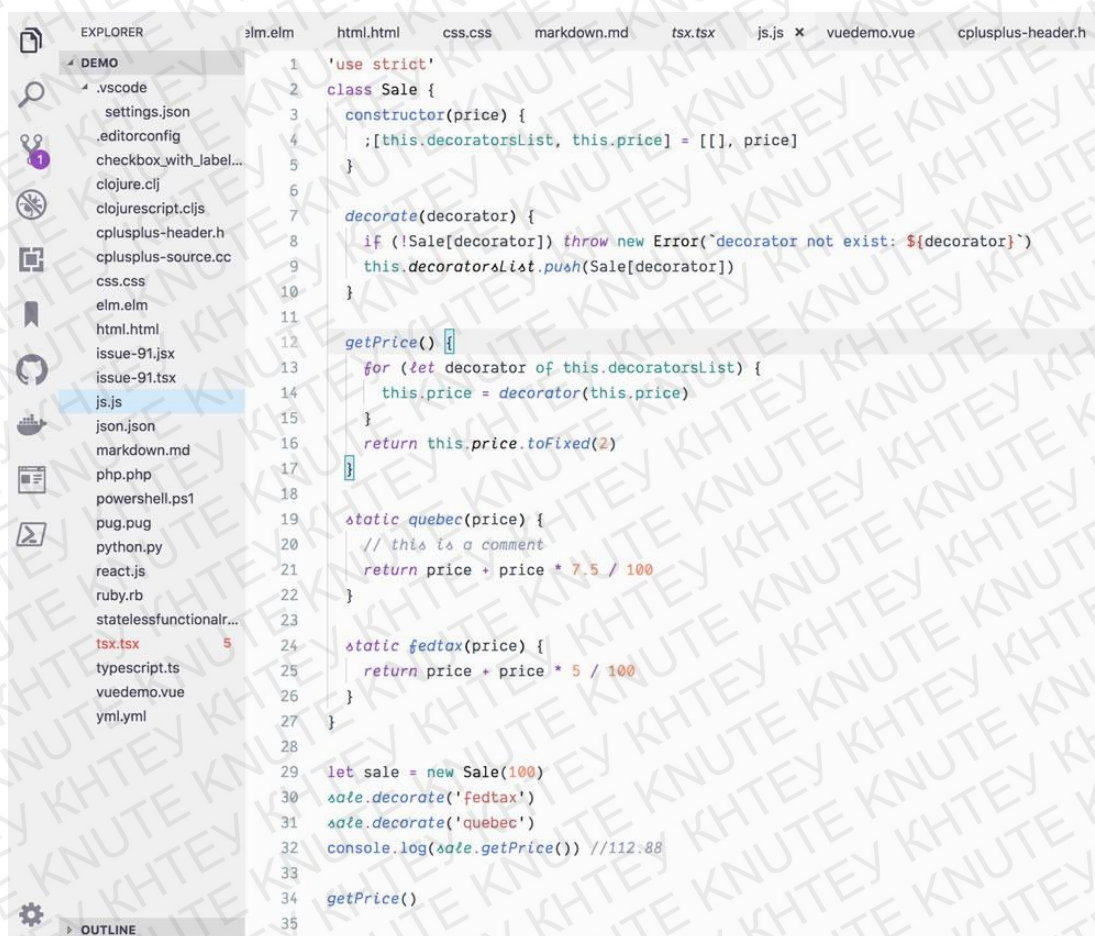


Рис. 4.10. Вікно середовища розробки Visual Studio Code

Visual Studio Code — засіб для створення, редагування та зневадження сучасних веб-застосунків і програм для хмарних систем. Visual Studio Code розповсюджується безкоштовно і доступний у версіях для платформ Windows, Linux і OS X.

Компанія Microsoft представила Visual Studio Code у квітні 2015 на конференції Build 2015. Це середовище розробки стало першим крос-платформовим продуктом у лінійці Visual Studio.

За основу для Visual Studio Code використовуються напрацювання вільного проекту Atom, що розвивається компанією GitHub.

Зокрема, Visual Studio Code є надбудовою над Atom Shell, що використовують браузерний рушій Chromium і Node.js. Примітно, що про

					КНТЕУ 121-05-04.МР	Аркуш
						59
Изм.	Аркуш	№ докум	Подпись	Дата		

використання напрацювань вільного проекту Atom на сайті Visual Studio Code і в прес-релізі і в офіційному блозі не згадується.

Редактор містить вбудований зневаджувач, інструменти для роботи з Git і засоби рефакторингу, навігації по коду, автодоповнення типових конструкцій і контекстної підказки. Продукт підтримує розробку для платформ ASP.NET і Node.js, і позиціонується як легковагове рішення, що дозволяє обійтися без повного інтегрованого середовища розробки. Серед підтримуваних мов і технологій: JavaScript, C++, C#, TypeScript, jade, PHP, Python, XML, Batch, F#, DockerFile, Coffee Script, Java, HandleBars, R, Objective-C, PowerShell, Luna, Visual Basic, Markdown, JSON, HTML, CSS, LESS і SASS, Haxe.

Angular CLI – інструмент представлений консоллю через яку завдяки спеціальним командам можна створювати та керувати Angular проектами, , встановлювати розширення, генерувати структурні елементи, тощо.

4.5. Програмування Angular2 веб додатку мовою TypeScript

Angular2 додаток, як вже зазначалось складається з модулів компонентів та сервісів. Основним модулем є AppModule що зберігається в кореневій папці проекту і де зазначаються усі сервіси та компоненти що використовуються в проекті.

```
@NgModule({  
  declarations: [ // оголошення основних структурних одиниць (компонентів)  
    AppComponent,  
    SubjectComponent,  
    LoginComponent,  
    TeachersComponent,  
    ScheduleComponent,  
    CalendarComponent  
  ],  
  imports: [
```

					КНТЕУ 121-05-04.МР	Аркуш
						60
Изм.	Аркуш	№ докум	Подпись	Дата		

```

AppRoutingModule, // імпортування сервісів та модулів
BrowserModule,
AngularFireModule.initializeApp(config),
AngularFireAuthModule,
AngularFireDatabaseModule,
CalendarModule
],
providers: [],
bootstrap: [AppComponent] // визначення стартового компоненту
})
export class AppModule {
}

```

					КНТЕУ 121-05-04.МР	Аркуш
						61
Изм.	Аркуш	№ докум	Подпись	Дата		

Структура веб додатку представлена чотирма компонентами:

- Розклад
- Предмети
- Викладачі
- Календар

Компонент «Розклад»

Візуальне відображення даного компоненту представлено у вигляді випадного списку з днями тижня та предметами для кожного з них, що розміщується в лівій частині екрану та блоку додавання розкладу – два комбіновані списки: «день тижня» та «предмет». Після вибору предмету для дня тижня, інформація негайно додається до випадного списку без перезавантаження сторінки.

Рис. 4.11. Сторінка перегляду і додавання розкладу студента КНТЕУ

Програмна реалізація компоненту представлена конструктором та трьома публічними методами:

- addSchedule - додати розклад
- deleteSchedule - видалити розклад
- updateSchedule - оновити розклад

					КНТЕУ 121-05-04.MP	Аркуш
						62
Изм.	Аркуш	№ докум	Подпись	Дата		

І одним приватним методом `getScheduleForDay` – отримує список предметів розкладу для зазначеного дня тижня.

Модель даних компоненту представлена класом `ScheduleHolder` що містить зібрану деталізовану інформацію про предмет, день тижня та викладача:

```
class ScheduleHolder {
  scheduleId: any;
  day: any;
  @Input() subjects: Array<any>;
  teacher: any;
  constructor(){
    this.subjects = new Array<any>();
  }
}
```

Компоненти «Предмети» та «Викладачі» представлені подібним чином.



Рис. 4.12. Сторінка «Предмети»

Взаємодія з Firebase здійснюється через об'єкт класу `AngularFireDatabase` що бібліотеки `@angular/fire/database`.

Основними обробниками подій є:

`snapshotChanges` – спрацьовує лише один раз при відкритті компоненту

`valueChanges` – спрацьовує щоразу коли дані бази змінюються іншими клієнтами.

					КНТЕУ 121-05-04.MP	Аркуш
						63
Изм.	Аркуш	№ докум	Подпись	Дата		

Компонент «Календар»

Реалізує функції бібліотеки Angular 6.0+ Calendar .

Візуально представлений інтерактивним календарем в якому можна встановлювати плани. Має режими відображення місяця, тижня та поточного дня. Також містить три кнопки швидкої навігації «Попередій місяць», «Сьогодні», «Наступний»

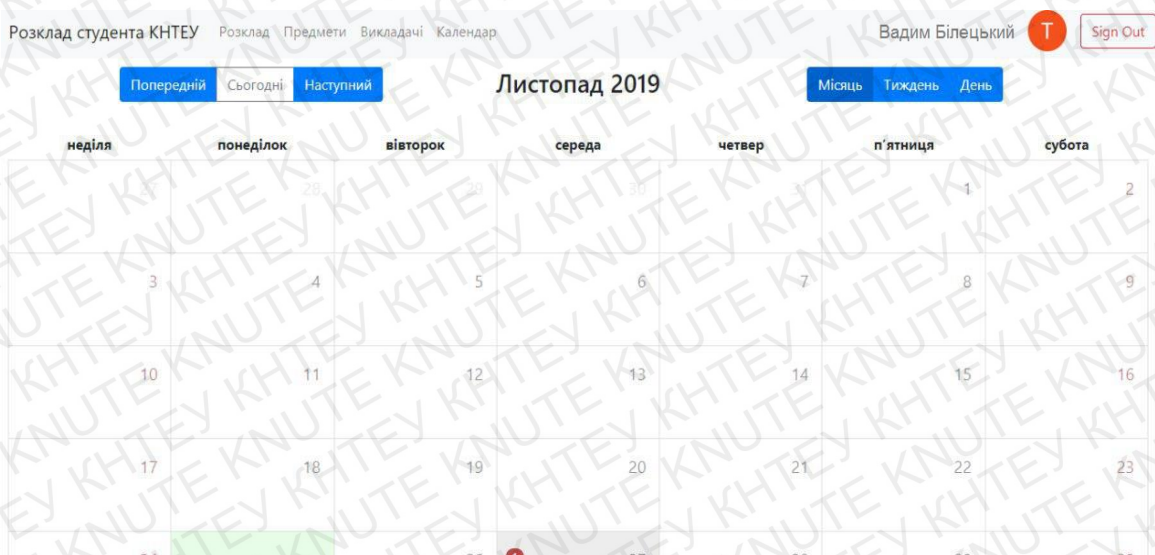


Рис. 4.13. Сторінка перегляду календаря

Користувач має можливість обрати день місяця та встановити перелік подій з вказуванням пріоритетності та часу тривалості.

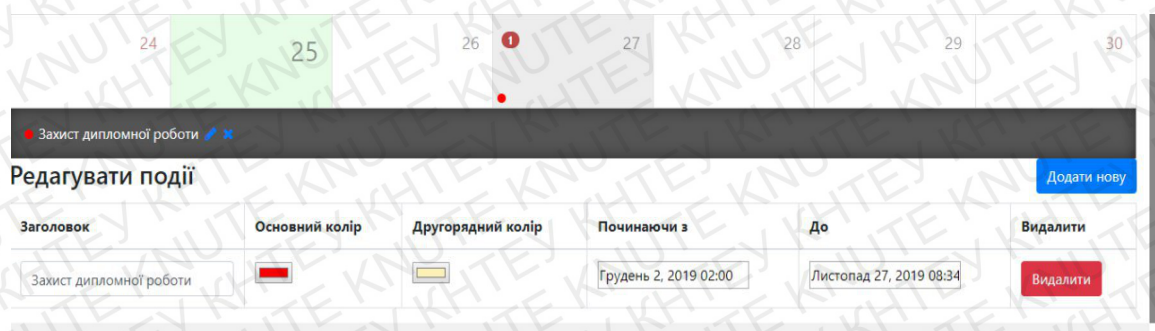


Рис. 4.14. Додавання події до календарного дня

4.6. Висновок до розділу 4

Firebase – є ефективним рішенням при постановці задач щодо зберігання на обробки даних. Задня використання сервісу необхідно здійснити ряд кроків щодо інтеграції проекту з робочими проектами Android та Angular2 програмного забезпечення що розробляється.

					КНТЕУ 121-05-04.MP	Аркуш
						64
Изм.	Аркуш	№ докум	Подпись	Дата		

Мова TypeScript дозволяє писати чітко структурований код для ефективного вирішення складних завдяки типам мови.

					КНТЕУ 121-05-04.МР	Аркуш
						65
Изм.	Аркуш	№ докум	Подпись	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

У проведеному дослідженні були проаналізовані основні архітектурні рішення та підходи для створення програмного забезпечення. Проведений аналіз освітнього процесу студента КНТЕУ, на основі якого визначені основні функціональні вимоги до програмного забезпечення.

Спираючись на технічне завдання кваліфікаційної роботи, був обраний та обґрунтований найбільш оптимальний підхід та обрані архітектурні шаблони вирішення типових функціональних задач програмного забезпечення що розробляється.

Проведений порівняльний аналіз технологій та засобів реалізації поставлених задач відтак обрані найоптимальніші з них.

Базуючись на проведеній аналітиці спроектована структура бази даних з дотриманням існуючих правил проектування.

В результаті було розроблено розподілене програмне забезпечення представлено мобільним додатком для системи Android та веб доаток у фреймворку Angular2.

При розробці мобільного додатку були розглянуті переваги мови програмування Java.

В ході розробки додатку на базі Angular2 продемонстровані основні переваги мови програмування TypeScript.

					КНТЕУ 121-05-04. МР			
					<i>Розробка програмного забезпечення для отримання розкладу студента КНТЕУ</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		В	66	68
Зав. каф.	Криворучко О.В.							
Керівник	Цензура М.О.							
Гарант	Цензура М.О.							
Розробив	Білецький В. А.				<i>Висновки та пропозиції</i>	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5 група		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

Книги колективу авторів:

1. «Android. Програмування для професіоналів» Б. Харді, Б. Філліпс, К. Стюарт, К. Марсикано - 451с.
2. Android для разработчиков. 3-е издание Дейтел П., Дейтел Х. 2017
3. П. Дейтел, Х. Дейтел, А. Уолд. Android для разработчиков. 3-е издание(2016) - 240 с.
4. Хашими С., Коматинени С., Маклин Д.: "Разработка приложений для Android" - 117 с.

Багатотомні видання:

5. Шилдт Г. - Java 8. Полное руководство. 9-е издание (2015) – 533 с.

Видання іноземною мовою

6. K. Manikas, K. M. Hansen, Software ecosystems - a systematic literature review, Journal of Systems and Software 86 (5) (2013) 1294–1306. [8] J. Bosch, From software product lines to software ecosystems, in: International Software Product Line Conference, San Francisco, California, 2009, pp. 111–119.
7. G. P. Picco, C. Julien, A. L. Murphy, M. Musolesi, G.-C. Roman, Software engineering for mobility: Reflecting on the past, peering into the future, in: 2014 Future of Software Engineering (FOSE'14), 2014, pp. 13–28.
8. R. Taylor, N. Medivodivc, E. Dashofy, Software Architecture: Foundations, Theory, and Practice, Wiley, Hoboken, NJ, USA, 2009.
9. N. Medvidovic, M. Mikic-Rakic, N. R. Mehta, S. Malek, Software architectural support for handheld computing, IEEE Computer 36 (9) (2003) 66–73.

					КНТЕУ 121-05-04. МР			
					Розробка програмного забезпечення для отримання розкладу студента КНТЕУ	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		СД	67	68
Зав. каф.	Криворучко О.В.							
Керівник	Цензура М.О.							
Гарант	Цензура М.О.							
Розробив	Білецький В. А.				Список використаних джерел	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5 група		

10.S. Malek, M. Mikic-Rakic, N. Medvidovic, Reconceptualizing a family of heterogeneous embedded systems via explicit architectural support, in: ICSE, ICSE '07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 591–601.

Електронні ресурси:

11.<https://firebase.google.com/docs/guides> - Firebase Guides

12.Wasserman, Software engineering issues for mobile application development, in: Proceedings of the FSE/SDP workshop on Future of software engineering research (FoSER'10), 2010, pp. 397–400. [6] Google play market. URL <http://play.google.com/store/apps/>.

13.U. Eklund, J. Bosch, Architecture for embedded open software ecosystems, Journal of Systems and Software 92 (2014) 128–142. doi:10.1016/j.jss.2014.01.009. URL <http://www.sciencedirect.com/science/article/pii/S0164121214000211> [10] C. Ebert, C. Jones, Architecture for embedded open software ecosystems, Computer 42 (2009) 42–52. URL <http://dx.doi.org/10.1109/MC.2009.118>

					КНТЕУ 121-05-04. МР			
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка програмного забезпечення для отримання розкладу студента КНТЕУ	Стадія	Аркуш	Аркушів
						СД	68	68
Зав. каф.	Криворучко О.В.				Список використаних джерел	Факультет обліку, аудиту та інформаційних систем, 2 курс, 5 група		
Керівник	Цензура М.О.							
Гарант	Цензура М.О.							
Розробив	Білецький В. А.							

ДОДАТКИ

Додаток А

Лістинг програмного коду авторизації користувачів в мобільному додатку

```
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    mFirebaseAuth = FirebaseAuth.getInstance();

    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    if (user != null) {
        Intent intent = new Intent(this, BottomNavActivity.class);
        startActivity(intent);
        finish();
        return;
    }

    List<AuthUI.IdpConfig> providers = Arrays.asList(
        new AuthUI.IdpConfig.EmailBuilder().build()
    );

    Intent intent = AuthUI.getInstance()
        .createSignInIntentBuilder()
        .setIsSmartLockEnabled(!BuildConfig.DEBUG)
        .setAvailableProviders(providers)
        .setLogo(R.mipmap.ic_launcher_round)
        .build();
    startActivityForResult(intent, RC_SIGN_IN);
}
```

@Override

```
public void onClick(View view)
```

```
{
```

```
switch (view.getId()) {
```

```
//     case R.id.registerBtn:
```

```
//         loginWithEmail();
```

```
//         break;
```

```
}
```

```
}
```

```
private boolean loginWithEmail(){
```

```
String emailVal = mEmailET.getText().toString();
```

```
String passwordVal = mPasswordET.getText().toString();
```

```
mFirebaseAuth.createUserWithEmailAndPassword(emailVal,passwordVal);
```

```
return true;
```

```
}
```

@Override

```
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data)
```

```
{
```

```
super.onActivityResult(requestCode, resultCode, data);
```

```
Toast.makeText(this,"Hi bitch, listen to me now!",Toast.LENGTH_LONG);
```

```
if (requestCode == RC_SIGN_IN) {
```

```
handleSignInResponse(resultCode, data);
```

```
}
```

```
}
```

```
private void handleSignInResponse(int resultCode, @Nullable Intent data) {
```

```
IdpResponse response = IdpResponse.fromResultIntent(data);
```

```
// Successfully signed in
if (resultCode == RESULT_OK) {
    Intent intent = new Intent(this, BottomNavActivity.class);
    startActivity(intent);
    finish();
} else {
    // Sign in failed
    if (response == null) {
        // User pressed back button
        //showSnackBar(R.string.sign_in_cancelled);
        return;
    }

    if (response.getError().getErrorCode() == ErrorCodes.NO_NETWORK) {
        return;
    }

    if (response.getError().getErrorCode() ==
        ErrorCodes.ANONYMOUS_UPGRADE_MERGE_CONFLICT) {
    }

    //showSnackBar(R.string.unknown_error);
    Log.e(TAG, "Sign-in error: ", response.getError());
}
}
}
```

Лістинг компоненту «Розклад» додатку Angular2

```
@Input()schedulesList : Array<ScheduleHolder>

constructor(db: AngularFireDatabase, router: Router) {

  this.schedulesList = new Array<ScheduleHolder>();

  this.router = router;

  this.db = db;

  this.scheduleRef = db.list('Schedules');

  this.daysRef = db.list("days");


  this.days = this.daysRef.valueChanges();

  this.days.subscribe(

    days => {

      days.map(day => {

        let schedHolder = new ScheduleHolder();

        schedHolder.day = day;

        this.getSubjectsForDay(day.dayId, schedHolder);

        this.schedulesList.push(schedHolder);

      });

    });

  this.subjectsRef = db.list('subjects');

  this.subjects = this.subjectsRef.snapshotChanges().pipe(

    map(changes =>

      changes.map(c => ({ key: c.payload.key, ...c.payload.val() })))

    )

  );

  this.teachersRef = db.list('teachers');

  this.teachers = this.teachersRef.snapshotChanges().pipe(

    map(changes =>
```

```

        changes.map(c => ({ key: c.payload.key, ...c.payload.val() })))
    )
);

this.schedules = this.scheduleRef.valueChanges();
this.scheduleRef.valueChanges().subscribe(data=>{
    for(var i=0;i<this.schedulesList.length;i++){
        this.schedulesList[i].subjects = [];
    }
});
}
deleteSchedule(scheduleId: any){
    this.scheduleRef.remove(scheduleId);
}

deleteEverything (){
    console.log(this.schedules);
}

private getSubjectsForDay(dayId: any, instance: ScheduleHolder){
    return this.schedules.subscribe(
        schedules => {
            schedules.map(schedule => {
                if(schedule.day.dayId == dayId){
                    schedule.subject.scheduleId = schedule.scheduleId;
                    instance.subjects.push(schedule.subject);
                }
            })
        });
}
}

```

```
addSchedule(){  
  debugger;  
  var key = this.scheduleRef.push({  
    day: this.selectedDay,  
    subject: this.selectedSubject,  
  }).key;  
  this.scheduleRef.update(key, {scheduleId: key });  
}
```