

Міністерство освіти і науки України
Департамент науки і освіти Харківської обласної державної адміністрації
Комунальний заклад «Харківська обласна Мала академія наук
Харківської обласної ради»

Відділення інженерії та матеріалознавства
Секція: інформаційно-телекомунікаційні системи та технології

РОЗРОБКА КОМУНІКАЦІЙНОГО СЕРВІСУ УНІВЕРСАЛЬНОЇ ПЕРСОНАЛЬНОЇ МОБІЛЬНОЇ ЛАБОРАТОРІЇ

Роботу виконав:
Вайнштейн Артем Олегович,
учень 10 класу Комунального
закладу «Харківський ліцей № 141
Харківської міської ради»

Наукові керівники:
Козуб Павло Анатолійович,
доцент кафедри медіасистем та
технологій Харківського
національного університету
радіоелектроніки

Булкіна Ірина Анатоліївна,
вчитель фізики Комунального
закладу «Харківський ліцей № 141
Харківської міської ради»,
спеціаліст вищої категорії

РОЗРОБКА КОМУНІКАЦІЙНОГО СЕРВІСУ УНІВЕРСАЛЬНОЇ ПЕРСОНАЛЬНОЇ МОБІЛЬНОЇ ЛАБОРАТОРІЇ

Вайнштейн Артем Олегович, учень 10 класу Комунального закладу «Харківський ліцей № 141 Харківської міської ради»

Козуб Павло Анатолійович, доцент кафедри медіасистем та технологій Харківського національного університету радіоелектроніки

Булкіна Ірина Анатоліївна, вчитель фізики Комунального закладу «Харківський ліцей № 141 Харківської міської ради», спеціаліст вищої категорії

Проведення лабораторних робіт є однією з важливіших ланок навчання, яку зазвичай сприймають як один із способів засвоєння нового матеріалу учнями. Зв'язок між абстрактними та реальними об'єктами дозволяє нам краще запам'ятовувати теорію, розуміти, використовувати її на практиці.

Створення нової лабораторної бази є одним із частин загальної методології підвищення рівня освіти, але воно не можливо без рішення цілого комплексу завдань – створення лабораторних приладів, створення методик їх використання та користування, створення інформаційних ресурсів для поширення досвіду використання та впровадження.

Причому інформаційне забезпечення розробки нового обладнання є одним із найбільш актуальних та важливих складових цієї великої роботи, і повинно проводитись починаючи з самих перших її етапів.

Тому метою даної роботи є розроблення інформаційної технології, алгоритмів та програмного забезпечення для комунікації універсальної персональної лабораторії для підвищення ефективності якості навчання.

Об'єктом дослідження є дистанційне навчання, а предметом дослідження є моделі методів та моделювання потоків даних в комунікаційній системі.

Завданням роботи стало розробка телеграм боту для проведення занять з учнями, та їх зв'язку з вчителями та науковими керівниками. На цій стадії передбачається відпрацювання основ дизайну системи, основних комунікаційних засобів, структури та необхідної кількості даних, методів взаємодії користувачів один з одним.

В результаті роботи було створено робочу версію телеграм бота та проведено її перевірку на учнях. Для повноцінної роботи боту було створено сайт допомоги та інформаційних ресурсів для саме для боту.

Ключові слова: лабораторні роботи, інформаційне забезпечення, комунікаційний сервіс, телеграм бот

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. РОЗРОБКА ПРОТОТИПУ ПРОЄКТА.....	6
1.1. Роль лабораторного обладнання у навчанні природничим наукам	6
1.2. Перспективи розвитку лабораторного обладнання	7
1.3. Інформаційне забезпечення освітнього процесу	9
1.4. Наявні системи управління навчальним процесом	10
РОЗДІЛ 2. СТВОРЕННЯ СЕРВЕРНОЇ ЧАСТИНИ ПРОЄКТУ	12
2.1. Розробка стратегії створення комунікаційної системи	12
2.2. Аналіз та вибір бази даних	14
2.3. Виділення структури бази даних	15
2.4. Створення API та його функціоналу	18
РОЗДІЛ 3. СТВОРЕННЯ КОМУНІКАЦІЙНОЇ ЧАСТИНИ ПРОЄКТУ	21
3.1. Вибір функціоналу телеграм бота	21
3.2. Створення телеграм бота.....	21
3.3. Створення сайту проєкту.....	27
РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ	30
4.1. Методика проведення дослідження	30
4.2. Результати дослідження	31
ВИСНОВКИ.....	32
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	33

ВСТУП

Проведення лабораторних робіт є однією з важливіших ланок навчання, яку зазвичай сприймають як один із способів засвоєння нового матеріалу учнями. Зв'язок між абстрактними та реальними об'єктами дозволяє нам краще запам'ятовувати теорію, розуміти, використовувати її на практиці.

Створення нової лабораторної бази є одним із частин загальної методології підвищення рівня освіти, але воно не можливо без рішення цілого комплексу завдань – створення лабораторних приладів, створення методик їх використання та користування, створення інформаційних ресурсів для поширення досвіду використання та впровадження.

Причому інформаційне забезпечення розробки нового обладнання є одним із найбільш актуальних та важливих складових цієї великої роботи, і повинно проводитись починаючи з самих перших її етапів. Російське вторгнення в Україну ще більше підвищило важливість цієї задачі.

Зараз багато дітей роз'їхалося по світу, або перебувають в інших містах України при цьому навчаючись онлайн [1]. Навіть, діти які перебувають в тому ж місті, що і навчаються можуть перебувати на онлайн навчанні, а їх процес комунікації дуже поганий [11]. Із за цього з'явилася потреба в створенні комунікаційної частини для мобільної лабораторії, ціль якої є поєднання учнів без потреби бути офлайн в одному місці, надати можливість навчатися в будь якій точці світу при цьому маючи свою міні лабораторію.

Метою даної роботи є розроблення інформаційної технології, алгоритмів та програмного забезпечення для комунікації універсальної персональної лабораторії для підвищення ефективності якості навчання.

В роботі було досліджено досвід таких універсальних та знаних платформ як Google Classroom [2], Microsoft Teams [3], Moodle [4] та Schoolguide [5]. Для реалізації поставленої мети було обрано телеграм бот, як найбільш поширений в учнівському середовищі комунікатор.

Об'єктом дослідження є дистанційне навчання, а предметом – дослідження є моделі методів та моделювання потоків даних в комунікаційній системі.

Завданням роботи стало розробка телеграм боту для проведення занять з учнями, та їх зв'язку з вчителями та науковими керівниками. На цій стадії передбачається відпрацювання основ дизайну системи, основних комунікаційних засобів, структури та необхідної кількості даних, методів взаємодії користувачів один з одним.

В результаті роботи було створено робочу версію телеграм бота та проведено її перевірку учнями. Для повноцінної роботи боту було створено сайт допомоги та інформаційних ресурсів для саме для боту.

РОЗДІЛ 1

РОЗРОБКА ПРОТОТИПУ ПРОЄКТА

1.1. Роль лабораторного обладнання у навчанні природничим наукам

Проведення лабораторних робіт є однією з важливіших ланок навчання, яку зазвичай сприймають як один із способів засвоєння нового матеріалу учнями. Зв'язок між абстрактними та реальними об'єктами дозволяє нам краще запам'ятовувати теорію, розуміти, використовувати її на практиці.

За своєю сутністю відображення певного фізичного, хімічного, біологічного, астрономічного явища у звичних для спостерігача речах, або демонстрацію особливостей ще не знайомого для спостерігача обладнання подібно використанню скануючого тунельного мікроскопу для відображення атомів при виконанні складних досліджень, хоча суттєво і відрізняється за складністю обладнання.

Для досягнення максимального демонстраційного ефекту використовують спеціальне обладнання (для демонстраційних робіт), або проводять лабораторну роботу на спеціальному обладнанні за особливих умов (тренувальні лабораторні роботи), рівень складності якого тісно пов'язаний з рівнем розвитку освіти.

Так оснащеність лабораторними приладами в школах країн Європи значна краща, ніж в школах України, а лабораторне обладнання вищих учбових закладів американських технічних університетів є недосяжним за параметрами навіть для провідних наукових установ України.

Для доведення лабораторного обладнання у вітчизняних навчальних закладів до рівня розвинутих країн необхідні величезні кошти, яких немає, але навіть при їх наявності цей процес буде дуже тривалим. Крім того, це зовсім не гарантує, що нове обладнання не буде знову застарілим на момент початку його використання.

Таким чином, на даний момент при створенні будь-якої лабораторної техніки, у будь-якому навчальному або науково-дослідному закладі виникають проблеми, які принципово не можуть бути вирішені без зміни самої концепції цієї частини освітнього процесу.

Так сучасна концепція лабораторної бази передбачає:

- стаціонарний характер (клас, лабораторія, організація);
- термін використання не менше 5-10 років;
- обмежена кількість видів приладів та виробників;
- переважно гарний вигляд (на момент створення);
- неможливість модифікації в процесі роботи;
- принцип роботи – колективний (група або черговість).

Для вивчення складних природних процесів (космічні, нано-, біо- технології, медицина, геологія та астрономія) така концепція виправдовує себе. Демонстрація явищ, дослідження та навчання спеціалістів у цих сферах освіти неможлива без складного та високоточного обладнання. Але для середньої школи, професійної освіти, вищої школи, такий підхід вже є застарілим і не виконує своїх функцій.

1.2. Перспективи розвитку лабораторного обладнання

Для сучасної освіти лабораторне обладнання повинно постійно оновлюватись у зв'язку з постійним розвитком технологій та рівня викладання. Воно повинно відповідати за простотою використання рівню розвитку учня. Його різноманітність повинна покривати всі рівні розвитку учнів. Воно повинно бути доступним для максимальної кількості учнів. Воно повинно використовуватись для всіх можливих форм навчальної діяльності – демонстрації, тренування навичок, наукових досліджень. Основою нової концепції лабораторного обладнання має стати максимально можлива участь у його створенні представників освіти.

В залежності від рівня підготовки учнів та викладачів доля такої участі може бути різною, але основні принципи для всіх ланок освітнього процесу можна визначити таким чином:

- Індивідуальний підхід (кожен учень має свій інструмент);
- Мінімальна достатність (інструмент вимірює всі необхідні для уроку параметри);
- Мобільність (заняття в полі, лісі, лікарні, на відпочинку);
- Універсальність (використання для всіх дисциплін);
- Наочність і практичність.

Реалізація такого підходу принципово змінює процес оснащення навчальних закладів лабораторними приладами і навіть сам процес навчання. По-перше переоснащення може провести не за десятиріччя, а за декілька років. По-друге, воно буде завжди встигати за розвитком технологій. По-третє, проведення переоснащення буде значно дешевшим. По-четверте, воно не буде залежати від виробників. По-п'яте, воно буде враховувати новітні освітні тенденції, такі як глобалізація освітнього процесу, використання дистанційних методів. По-шосте, функціональність обладнання дуже швидко може стати близькою до функціональності професійного лабораторного обладнання.

Обладнання повинно стати тісно інтегрованим у всі види навчального процесу (демонстрація, навички роботи з обладнанням, використання обладнання для досліджень). Воно має бути настільки зручним та простим у використанні, щоб не відлякувати учнів та вчителів. У його створенні повинні приймати участь учні, вчителі та дослідники. Обладнання повинно передбачати використання його після демонстрації учбового матеріалу для проведення дослідницьких робіт, створення практично важливих пристроїв для школи, та використання потім їх удома. Обладнання повинно бути розраховано на короткий термін використання (1-2 роки) для постійного врахування зміни в учбовому процесі та технологій, та зважаючи на низький рівень кваліфікації користувачів. Воно повинно бути таким простим, щоб його створення можна було довірити учням, викладачам або дослідникам. Обладнання повинно бути максимально дешевим для можливості його постійного оновлення, дублювання і виготовлення нових зразків. Обладнання повинно бути виготовлено з універсальних частин, які є завжди в наявності і можуть бути замінені при необхідності у будь-який момент при виході їх з ладу, що дозволить створити єдину методичну базу використання та створення обладнання на регіональному, національному і, можливо, глобальному (міжнаціональному) рівнях. Обладнання повинно бути мобільним для можливості використання його в умовах дистанційного навчання, проведення індивідуальних дослідницьких робіт за межами лабораторії. Точність вимірювання обладнання повинно враховувати рівень навчання.

Аналіз можливостей сучасних електронних приладів та тенденцій їх розвитку дозволяє стверджувати що всі ці умови для нового обладнання забезпечити в найкоротші терміни при використанні мікроконтролерів Atmel AVR. Вони дешеві, мають велику кількість додаткових частин, датчиків, та виконавчих механізмів, всі їх клони є максимально сумісними, для них є найбільше навчальних посібників, та прикладних програм. Для них постійно створюються нові додаткові частини. Більше того, створені на їх основі прилади можуть бути швидко перероблені у більш професійні версії.

1.3. Інформаційне забезпечення освітнього процесу

Ключовими аспектами інформаційної системи є педагогічні принципи, технології, залучення інтерактивності та відповідність конкретним вимогам цільової аудиторії, що потрібно враховувати при створенні інформаційних освітніх ресурсів і що вимагає великої роботи.

Педагогічні принципи:

Спрямованість на студента (learner-centered): Сучасні ресурси акцентують на потребах та інтересах учнів, надаючи їм можливість визначати свій власний темп та спосіб навчання.

Активне залучення (active engagement): Використання інтерактивних завдань, вправ та сценаріїв, щоб забезпечити активну участь учнів у навчальному процесі.

Оцінка та зворотний зв'язок: Включення системи оцінювання та надання зворотного зв'язку для покращення розуміння та поглиблення знань.

Технології:

Мультимедіа: Використання текстів, зображень, відео, аудіо та інших мультимедійних елементів для збагачення змісту та зрозумілості матеріалу.

Платформи для віддаленого навчання: Використання спеціалізованих платформ, таких як Moodle, Blackboard, або користування системами управління навчанням (Learning Management Systems, LMS).

Відкриті ресурси та масові відкриті онлайн-курси (MOOCs): Забезпечення доступу до навчальних матеріалів великій кількості учнів через Інтернет.

Залучення інтерактивності:

Форуми та обговорення: Створення спільноти для обміну думками та досвідом.

Інтерактивні вправи та лабораторії: Використання вправ, які дозволяють студентам застосовувати свої знання на практиці.

Типи даних:

Текстові дані: Інформаційні матеріали у вигляді тексту, які можуть містити теоретичний матеріал, інструкції, відомості.

Мультимедіа-дані: Зображення, відео, аудіо, які розширюють можливості навчання та поліпшують засвоєння матеріалу.

Інтерактивні дані: Вправи, тести, опитування, які дозволяють студентам активно взаємодіяти з матеріалом та перевіряти свої знання.

Однак важливо розуміти, що успішність навчального процесу залежить від якості комунікації між учнями та вчителем. Слід зазначити що комунікація повинна бути здійснена максимально просто та підтримувати передачу всіх даних з можливістю інтерактивності з нею.

1.4. Найвні системи управління навчальним процесом

На даний момент існує дуже мало аналогів даному проєкту з них можна виділити Google Classroom [2], Microsoft Teams [3], Schoolguide [5] та Moodle [4]. Дослідження конкурентів буде оцінюватися за 5 факторами:

- Простота
- Інтеграція
- Комунікація
- Функціонал
- Доступність

Google Classroom [2] – має нескладний та інтуїтивно зрозумілий інтерфейс [12]. Є можливість інтеграцій у свої системи. Комунікація розвинена на поганому рівні, як і між учасниками, так і між вчителем та учнем. Функціонал дуже великий за рахунок великої екосистеми додатків Google.

Microsoft Teams [3] – має складний та не дуже зрозумілий інтерфейс, є можливість інтеграцій у свої системи. І дуже гарно реалізована комунікація між всіма. Функціонал дуже великий за рахунок системи додатків Microsoft та власних можливостей додатка. Головним мінусом додатку є платний функціонал.[13]

Schoolguide [5] – схожий за реалізацією проєкт, який є системою чат ботів. Має не складний інтерфейс. Інтеграція можлива. Комунікація не реалізована та функціонал не великий. Доступність при можливості самостійного розгортання.

Moodle [4] – має великий інтерфейс і залежить від налаштувань самого додатку. Має чудову інтеграцію та гарно реалізовану комунікацію. Функціонал із за open-source та можливості додавання свого функціоналу дуже великий. Але це все потребує затрат на використання та розгортання [14].

Отже враховуючи недоліки інших додатків можна констатувати, що комунікаційна частина нової система повинна мати не складний інтерфейс, можливість інтеграцій, гарно реалізовану комунікацію та при цьому мати великий функціонал та бути доступний всім. Отже з цих головних принципів потрібно вибрати етапи реалізації та методи реалізації. Щоб забезпечити простоту та доступність в якості основної платформи проєкту став месенджер Telegram і отже в якості платформи буде телеграм бот. Далі для забезпечення інтеграції потрібно створити API, щоб кожен користувач міг робити свої системи та розширювати можливості проєкту. Ну і звісно, щоб популяризувати проєкт потрібно створити тематичну веб сторінку з оптимізацією для пошукових систем, щоб проєкт можна було знайти через пошукові системи. Головною мовою програмування для проєкту буде JavaScript та його бібліотеки.

РОЗДІЛ 2

СТВОРЕННЯ СЕРВЕРНОЇ ЧАСТИНИ ПРОЄКТУ

2.1. Розробка стратегії створення комунікаційної системи

Діаграма моделювання потоків даних (DFD) є ефективним інструментом для візуалізації та аналізу потоків і обробки даних в системах. У випадку комунікаційного сервісу, створення діаграми потоків даних є важливою складовою процесу проектування та розробки системи з урахуванням інформаційних технологій.

Основною метою створення діаграми потоків даних для комунікаційного сервісу є наочне представлення та аналіз потоків даних у системі. Це дозволяє зрозуміти, які дані входять у систему, як вони обробляються та як вони взаємодіють з іншими компонентами системи. Із за цього на діаграмі можна побачити всі труднощі та недоліки проекту і на ранніх етапах сервісу за допомогою даного інструменту інформаційних технологій можна вирішити проблеми, які можуть виникнути.

Отже було розроблено DFD діаграму згідно проведеному аналізу існуючих комунікаційних систем та власних виділених вимог.

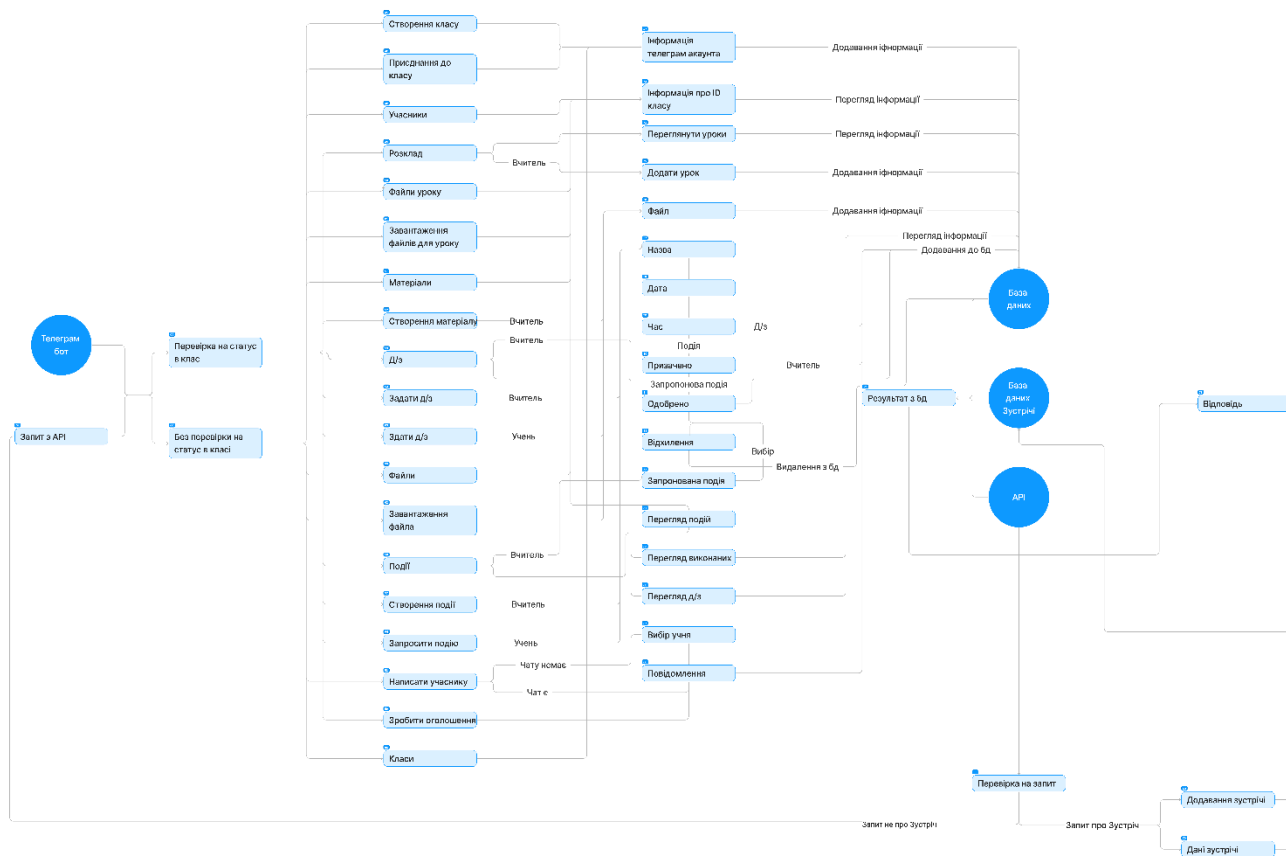


Рис. 1.1. DFD діаграма потоків даних комунікаційної платформи(рис. автора)

Далі для виділення основних етапів реалізації та їх методів потрібно описати діаграму потоків даних, щоб краще зрозуміти кожний її елемент.

Зовнішні сутності:

API - ця сутність представляє зовнішню систему або сервіс, який взаємодіє з системою, надсилаючи або отримуючи дані через API. Це може включати запит інформації, подання оновлень або запуск дій всередині системи.

Telegram-бот - зовнішня сутність, яка взаємодіє з системою для надсилання повідомлень, отримання команд або полегшення комунікації з користувачами.

Процеси:

Перевірка статусу в класі - процес, який включає перевірку того, чи має певний статус певний користувач.

Перевірка стану чата - процес, який перевіряє чи є в даний момент чат у користувача.

Створення д/з або події або уроки - процес який включає в себе створення запису в базі даних з часом, назвою події.

Завантаження файлу - процес який включає створення запису файлу в базі даних.

Сховища даних:

База даних - база даних яка включає в себе всі дані комунікаційної системи.

База даних Зустрічі - база даних, яка зберігає дані додатку Зустріч.

Потоки даних на цій DFD-діаграмі представляють інформацію, яка рухається між зовнішніми сутностями і процесами всередині системи, а також між самими процесами. Кожен потік даних позначений, що вказує на характер інформації, яка передається в системі, що підтримує роботу системи.

Отже створено діаграму потоків даних для створення плану реалізації проєкту. А саме вибір бази даних та її налаштування, створення API, що є серверною частиною комунікаційної системи. Далі створюється клієнтська частина, а саме телеграм бот та сайт до цього боту. Діаграма також доступна за посиланням: <https://www.figma.com/file/jBFQmg4rEBPjaGDy7Ro5Ex/DFD-%D0%B4%D1%96%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%B0?type=whiteboard&node-id=0%3A1&t=mRB8Te19CX9PiM24-1>

2.2. Аналіз та вибір бази даних

При виборі бази даних розглянуто кілька популярних варіантів та проаналізував їх з огляду на функціональність і можливості масштабування.

Реляційні СУБД на кшталт MySQL[6] та PostgreSQL[7] дозволяють створювати складні зв'язки між таблицями та забезпечують цілісність даних. Проте вони вимагають заздалегідь спроектованої жорсткої схеми, складні для змін. Крім

того, вони не оптимізовані для швидкої обробки великих обсягів даних в режимі реального часу.

Нереляційні СУБД, такі як MongoDB[8], навпаки, дозволяють гнучко структурувати дані у вигляді JSON подібних документів. Це спрощує розробку для нашого чат-бота. MongoDB[8] легко масштабується на кілька серверів та сховищ. Вона оптимізована для швидкої запису даних, що дуже важливо при обміні повідомленнями бота. З MongoDB[8] легко реалізувати пошук та складні запити до неструктурованих метаданих чату.

Гнучкість та можливості масштабування MongoDB[8] ідеально пасують для проекту, а саме телеграм бота, який повинен швидко обробляти дані та розширюватися при зростанні кількості користувачів. Крім того MongoDB[8] легко масштабувати на різні технології та проекти, що зробить дуже гарну можливість інтеграції проекту.

Ще доволі важливим пунктом стала доступність бази даних, Atlas дає базу даних MongoDB[8] на 512 мегабайт на все життя, при тому залишаючи можливість покращення бази даних з більшими можливостями за різними тарифами. Що ідеально підходить для починаючого проекту, при цьому залишаючи собі можливість для подальшого розширення при цьому залишаючись потужною та багатofункціональною базою даних.

2.3. Виділення структури бази даних

Для реалізації проекту після вибору бази даних створити логічну та ефективну структуру бази даних. Тому враховуючи потреби додатку потрібно створити 4 колекції:

- classrooms (Класи)
- users (Користувачі)
- lessons (Уроки)
- homework (Домашні завдання)

Відповідно до назв там будуть зберігатися відповідні дані. Та як MongoDB[8] зберігає елементи об'єктами, потрібно зробити чітку структуру елементів колекції.

```

    _id: "EV8AtpCGeilSgGE3DHELg"
    name: "Тест клас"
    idS: "UR-UqojKBs2DLII20-mIq"
  ▾ files: Array (1)
    ▾ 0: Object
      chatID: 5271590214
      msgID: 9027
  ▾ events: Array (1)
    ▾ 0: Object
      id: "1lNdLm6zo8gtSkEYJcU00"
      text: "пропро"
      date: "04.12.2023"
      time: "18:27"
      who: "для всіх"
      status: false
  ▶ marks: Array (empty)
  ▶ statisticks: Array (empty)
  ▾ materials: Array (1)
    ▾ 0: Object
      chatID: 5271590214
      msgID: 9031
      lastModified: 2023-12-28T13:03:33.675+00:00

```

Рис. 2.1. Структура елемента в колекції classrooms(рис. автора)

Об'єкт елемента в classrooms складається з унікального id в колекції(_id), який на додачу ще є id приєднання вчителя в клас. Далі йде name, тобто назва класу і далі йде id для приєднання учня в клас(idS). І далі вже йдуть елементи об'єкту для функціоналу взаємодії. В files зберігаються файли класу. В events зберігаються події та їх статус запланована подія(false) чи не перевірена запланована подія(true). І далі йдуть на майбутнє оцінки та статистика і останнє це матеріали, які викладає вчитель.

```

_id: ObjectId('64fae7a6524a778a834d5b44')
nameC: "Тест гуру ч2"
name: "+Артем+"
username: "Artemis_Vainshtein"
id: 1052973544
role: 1
classId: "x8rUMHXXEm50QVrs6N_fg"

```

Рис. 2.2. Структура елемента в колекції users(рис. автора)

Об'єкт елемента в users складається з унікального id в колекції(_id), далі йде nameС це назва класу. name це ім'я в телеграмі та username це username в телеграм. id це id в телеграмі, role 1 це вчитель та role 0 це учень. І classId це унікального id в колекції(_id) classrooms, тобто id класу.

```
_id: ObjectId('652444b20c37419d9e64edbf')
classId: "x8rUMHXXEm50QVrs6N_fg"
week: 1
day: "09.10.2023"
▼ file: Array (2)
  ▼ 0: Object
    msgId: 7378
    chatId: 1052973544
  ▼ 1: Object
    msgId: 7377
    chatId: 1052973544
text: "тест урок"
time: "08:30-09:1"
meet: "gdfgdf"
lastModified: 2023-10-09T18:26:45.504+00:00
```

Рис. 2.3. Структура елемента в колекції lessons(рис. автора)

Об'єкт елемента в users складається з унікального id в колекції(_id), далі йде classId що є id класу в classrooms, далі йде дата. Наступним є файли уроку, які мають аналогічну структуру що і в classrooms. Текст це про що урок, time це час уроку та meet це посилання на зустріч або її відсутність.

```
_id: ObjectId('6523f6d603f9eed18b1ff2c5')
type: 0
▼ whoMade: Array (1)
  ▼ 0: Object
    ▼ files: Array (1)
      ▶ 0: Object
        who: "Nyanya"
        id: 5271590214
        date: "2023-10-09"
        time: "15:54"
    name: "Тестове д/з"
  ▼ task: Array (2)
    ▼ 0: Object
      chatId: 5271590214
      msgId: 7156
    ▼ 1: Object
      chatId: 5271590214
      msgId: 7155
    date: "10.10.2023"
    time: "20:00"
    teacher: "Nyanya"
    id: "-CVV-xE1_MTrPNtZmdgbP"
    tchId: 5271590214
    lastModified: 2023-10-09T12:54:35.657+00:00
```

Рис. 2.4. Структура елемента в колекції homework(рис. автора)

Отже як і всі елементи об'єкт має унікальний id, далі йде type 0 це домашнє завдання з надсиланням файлів/тексту та type 1 це тест(не реалізовано функціонал). whoMade означає хто зробив домашнє завдання і далі є файли які працюють як і до цього, who хто здав і його ім'я в телегамі, id це id в телеграмі, date це дата здачі і time це час здачі. name це назва дз і task саме завдання, яке працює як файли. date дата до котрої потрібно виконати дз і time час до котрої потрібно виконати. teacher це хто задав домашнє завдання.

2.4. Створення API та його функціоналу

API створює можливість керування навчальним процесом. Технологія на якій реалізовано API є express js, який дозволяє швидко та просто створювати API. API дозволяє створювати навчальні групи, додавати до них учасників з різними ролями - викладачі та учні. Також реалізована можливість поширення матеріалів,

створення подій в розкладі, завдання та прийом домашніх робіт. Додатково передбачена взаємодія між учасниками через особисті повідомлення.

API використовує базу даних для зберігання інформації про групи, учасників, матеріали, розклад та інше. Також є можливість отримання даних через запити до бази. Є операції створення, отримання даних, оновлення та видалення.

Також було додано можливість взаємодії з додатком «Зустріч», а саме планування зустрічей та їх аналіз. Це було реалізовано за допомогою підключення до іншої бази даних Firebase.

```
app.post('/api/zustrich', async (req, res)=>{
  const firebaseConfig = {
    apiKey: process.env.API_KEY_FIREBASE,
    authDomain: "zustrich-be18b.firebaseio.com",
    databaseURL: "https://zustrich-be18b-default-rtdb.firebaseio.com",
    projectId: "zustrich-be18b",
    storageBucket: "zustrich-be18b.appspot.com",
    messagingSenderId: "125274425353",
    appId: "1:125274425353:web:6f38bd6892f88805ee10d8",
    measurementId: "G-4MH57WVHBD"
  };

  if(req.body.type){
    const app = initializeApp(firebaseConfig);
    const database = getDatabase(app);
    const room = nanoid()
    await set(ref(database, "rooms/" + room), {
      time: req.body.time,
      members:[]
    });

    await res.status(200).json({idRoom: room})
  }else{
    const app = initializeApp(firebaseConfig);
    const database = getDatabase(app);
    const starCountRef = ref(database, 'rooms/');

    onValue(starCountRef, (snapshot) => {
      snapshot.forEach(childSnapshot => {
        if (childSnapshot.key === req.body.idRoom) {
          let newUs = childSnapshot.val().members;
          res.status(200).json({members: newUs})
        }
      });
    });
  }
});
```

```
    res.status(200).json({members: []})  
  });  
}  
})
```

Отже API має свій функціонал, який дозволяє взаємодіяти напряму з ядром проєкта за допомогою `https` запитів, замість використання телеграм бота. Що дає можливість розгортання власних систем на основі створеної.

РОЗДІЛ 3

СТВОРЕННЯ КОМУНІКАЦІЙНОЇ ЧАСТИНИ ПРОЄКТУ

3.1. Вибір функціоналу телеграм бота

Щоб створити телеграм бота спочатку треба обрати функціонал, враховуючи що структура бази даних вже створена, тоді спираючись на неї та діаграму потоку даних потрібно розробити функціонал бота. Але головною умовою при створенні функціоналу це реалізувати це достатньо просто та легко, щоб кожен міг зорієнтуватися. Отже при запуску бота у користувача буде питати чи він хоче створити клас чи доєнатися, якщо через простий /start і додавати в клас якщо через посилання. Далі якщо користувач хоче створити клас то запитати назву класу і тоді створити клас і дати id для приєднання та посилання для приєднання. Якщо користувач хоче приєднатися через id, то бот запитає ким він хоче приєднатися та попросить id і приєднає в відповідний клас. І далі вже у користувача буде доступний весь функціонал і далі потрібно його зрозуміло розставити. Зверху повинні бути дії з уроками та інформацією про клас, в середині повинні бути матеріали, файли, дз та події. Та в кінці повинні бути кнопки оголошення(для вчителя), запропонувати подію(для учня), чат між учасниками та перехід між різними класами. Всі ці дії реалізовані через кнопки в телеграмі, що дозволяє користувачу швидко взаємодіяти з ботом. Але крім того потрібно додати функції через /, які повинні бути у користувача як швидкими діями. Туди було вирішено додати /start, /help, /reset, /chat та /gpt. /start для початкового повідомлення, /help для виклику сайту з інформацією про використання бота та деякі поради, /reset для ануляції всіх дій при можливих помилках, /chat чат з тех підтримкою та /gpt III в тестовій формі для допомоги учням.

3.2. Створення телеграм бота

Щоб розробити телеграм бота було вибрано бібліотеку Telebot[9] з JavaScript.

Отже головна структура коду бота виглядає так:

- Виклик всіх бібліотек

- Оголошення змінних з об'єктами, які запам'ятовують користувачів і їх дії для взаємодії з ботом(було використано замість плагіну askUser із за повільності плагіну)

- Функції для перевірок часу, дат тощо
- Функція яка отримує повідомлення, які пише користувач
- Функція inline кнопок

Виклик всіх бібліотек

```
import TeleBot from "telebot"  
import { MongoClient, ObjectId } from 'mongodb';  
import { nanoid } from 'nanoid'  
import moment from 'moment-timezone';
```

Це є всі викликані бібліотеки, з них сам телеграм бот, база даних, бібліотека для створення id та бібліотека для взаємодії з часом.

Оголошення змінних з об'єктами

```
let lastUserMessage = {};  
let userStatus = {};  
let userClass = {};  
let userChat = {};  
let userAction = {};
```

Отже в змінних зберігається остання повідомлення користувача, його роль в класі(вчитель або учень), його id класу, чи є стан переписування з кимось та остання якісь дії. Всі ці дані зберігаються з ключем id користувача в телеграмі.

Функція яка отримує повідомлення, які пише користувач

Це функція з купою логічних операторів, які визначають, яку дію вибрав користувачів. Одним з прикладів такого логічного оператора є:

```
if(text === "Файли" && userAction[msg.from.id] === undefined){  
  // let replyMarkup = bot.inlineKeyboard([  
  //   [
```

```

//     bot.inlineButton('Загрузити файл', {callback: "Загрузити файл"}),
//   ], [
//     bot.inlineButton('Отримати файли', {callback: "Отримати файли"})
//   ]
// ];
const client = await MongoClient.connect(
`mongodb+srv://${process.env.MONGO_USER}:${process.env.MONGO_PASSWORD}@${process.env.MONGO_URI}?retryWrites=true&w=majority`,
  { useNewUrlParser: true, useUnifiedTopology: true }
);
// const coll = client.db('artem-school').collection('users');
// const filter = {id: msg.from.id};
// const cursor = coll.find(filter);
// const result = await cursor.toArray();

const coll1 = client.db('artem-school').collection('classrooms');
const filter1 = {_id: userClass[msg.from.id]};
const cursor1 = coll1.find(filter1);
const result1 = await cursor1.toArray();
await client.close();
if(result1[0]){
  if(result1[0].files.length===0){
    return bot.sendMessage(msg.chat.id, 'В цьому класі ще немає файлів');
  }else{
    for(let i = 0; i<result1[0].files.length;i++){
      await
bot.forwardMessage(msg.chat.id,result1[0].files[i].chatID,result1[0].files[i].msgID);
      if(msg.chat.type ==="private" && userStatus[msg.from.id]){

```

await

```
bot.sendMessage(msg.chat.id, `${result1[0].files[i].chatID}&&${result1[0].files[i].msgID}`);
    }
}
return bot.sendMessage(msg.chat.id, 'Це всі файли в цьому класі');
}
}else{
return bot.sendMessage(msg.chat.id, 'Error');
}
}else if(text === "Завантаження файла" && userAction[msg.from.id] ===
undefined){
let replyMarkup = bot.keyboard([
["Назад"],
], {resize: true});

lastUserMessage[msg.from.id] = text;
return bot.sendMessage(msg.chat.id, 'Надішліть файл', {replyMarkup});
}else if(lastUserMessage[msg.from.id] === "Завантаження файла" &&
msg.text === undefined){
console.log(lastUserMessage[msg.from.id])
const client = await MongoClient.connect(
`mongodb+srv://${process.env.MONGO_USER}:${process.env.MONGO_PASSWORD}@${process.env.MONGO_URI}?retryWrites=true&w=majority`,
{ useNewUrlParser: true, useUnifiedTopology: true }
);
// const coll = client.db('artem-school').collection('users');
// const filter = {id: msg.from.id};
// const cursor = coll.find(filter);
// const result = await cursor.toArray();
```



```

const coll1 = client.db('artem-school').collection('classrooms');
const filter1 = {_id: userClass[msg.from.id]};
const cursor1 = coll1.find(filter1);
const result1 = await cursor1.toArray();
    const files = {files : [...result1[0].files, {chatID:msg.chat.id,
msgID:msg.message_id}]}
    console.log(result1)
    await coll1.updateOne(
        {_id: result1[0]._id},
        {
            $set: { ...files},
            $currentDate: { lastModified: true }
        }
    )
    await client.close();
    lastUserMessage[msg.from.id] = "textФайл";
    if(userStatus[msg.from.id]){
        return await bot.sendMessage(msg.chat.id, 'Файл додано',
{replyMarkup});
    }else if(userStatus[msg.from.id] === 0){
        let replyMarkup = bot.keyboard([
            ["Щоденник ✕", "Події", "Учасники"],
            ["Розклад", "Файли уроку", "Завантаження файлів для уроку"],
            ["Файли", "Завантаження файла", "Д/з", "Здати д/з"],
            ["Матеріали", "Запросити подію"],
            ["Написати учаснику", "Класи"]
        ], {resize: true});

```

```

        return await bot.sendMessage(msg.chat.id, 'Файл додано',
{replyMarkup});
    }
    return await bot.sendMessage(msg.chat.id, 'Файл додано');
}

```

Цей логічний оператор відповідає за перегляд файлів та їх завантаження користувачами в базу даних. І кожний такий логічний оператор перевіряє, яку дію зараз виконує користувач за допомогою об'єктів `userAction` та `lastUserMessage` та нинішнього повідомлення.

Функція inline кнопок

Має такий же принцип роботи, як і функція яка приймає дії, але вже очікує дій від inline кнопок. Приклад коду:

```

if(lastUserMessage[msg.from.id] === "Д/з" && userStatus[msg.from.id] ===
0){
    let newArr = userAction[msg.from.id].task.filter(arr => `${arr._id}` ===
msg.data);
    console.log(userAction[msg.from.id],msg.data, newArr)
    if(!newArr[0].type){
        userAction[msg.from.id] = undefined;
        lastUserMessage[msg.from.id] = "fgfds";
        await bot.sendMessage(msg.from.id, "Завдання:");
        for(let i = 0; i<newArr[0].task.length;i++){
            await
bot.forwardMessage(msg.from.id,newArr[0].task[i].chatId,newArr[0].task[i].msgId);
        }
    }else{
        await bot.sendMessage(msg.from.id, "Це д/з тест");
    }
}

```

```

    await bot.sendMessage(msg.from.id, `Завдання потрібно виконати до
    ${newArr[0].time} ${newArr[0].date}`)

    }

    if(lastUserMessage[msg.from.id] === "Д/з" && userStatus[msg.from.id]){
        let newArr = userAction[msg.from.id].task.filter(arr => `${arr._id}` ===
msg.data);

        console.log(userAction[msg.from.id],msg.data, newArr)
        userAction[msg.from.id] = {...userAction[msg.from.id],_id:msg.data}
        if(!newArr[0].type){
            let replyMarkup = bot.inlineKeyboard([[
                bot.inlineButton("Завдання", {callback: "Завдання"}),
            ],
            [
                bot.inlineButton("Хто виконав", {callback: "Хто виконав"}),
            ]]);
            bot.sendMessage(msg.from.id, "Виберіть дію:",{replyMarkup})
            lastUserMessage[msg.from.id] = "fgfds";
        }
    }
}

```

Цей логічний оператор відповідає за перегляд домашнього завдання та хто його виконав.

<https://github.com/Artemissssss/artem-school-bot> - посилання на код бота.

3.3. Створення сайту проєкту

Для популяризації проєкту потрібно створити сайт і оптимізувати його для пошукових систем. І якщо це сайт, то тоді потрібно зразу додати навчальну документацію по використанню боту. Отже для створення сайту згідно умов потрібно вибрати на чому його робити і з цією задачею чудово справиться фреймворк, який базується на React JS і це Docusaurus[10]. Він має можливість

швидкого і легкого створення подібний статичних сайтів. Щоб створити Docusaurus [10] проєкт потрібно в консолі прописати:

```
npx create-docusaurus@latest my-website classic
```

і тоді буде створено структуру node js проєкту і можна починати редагування і створення сайту. Для початку потрібно видалити блог і редагувати docs, де в md або mdx файлів додавати сторінки з туторіалами по використанню бота. Ну і звісно потрібно редагувати головну сторінку згідно правил чистого коду.

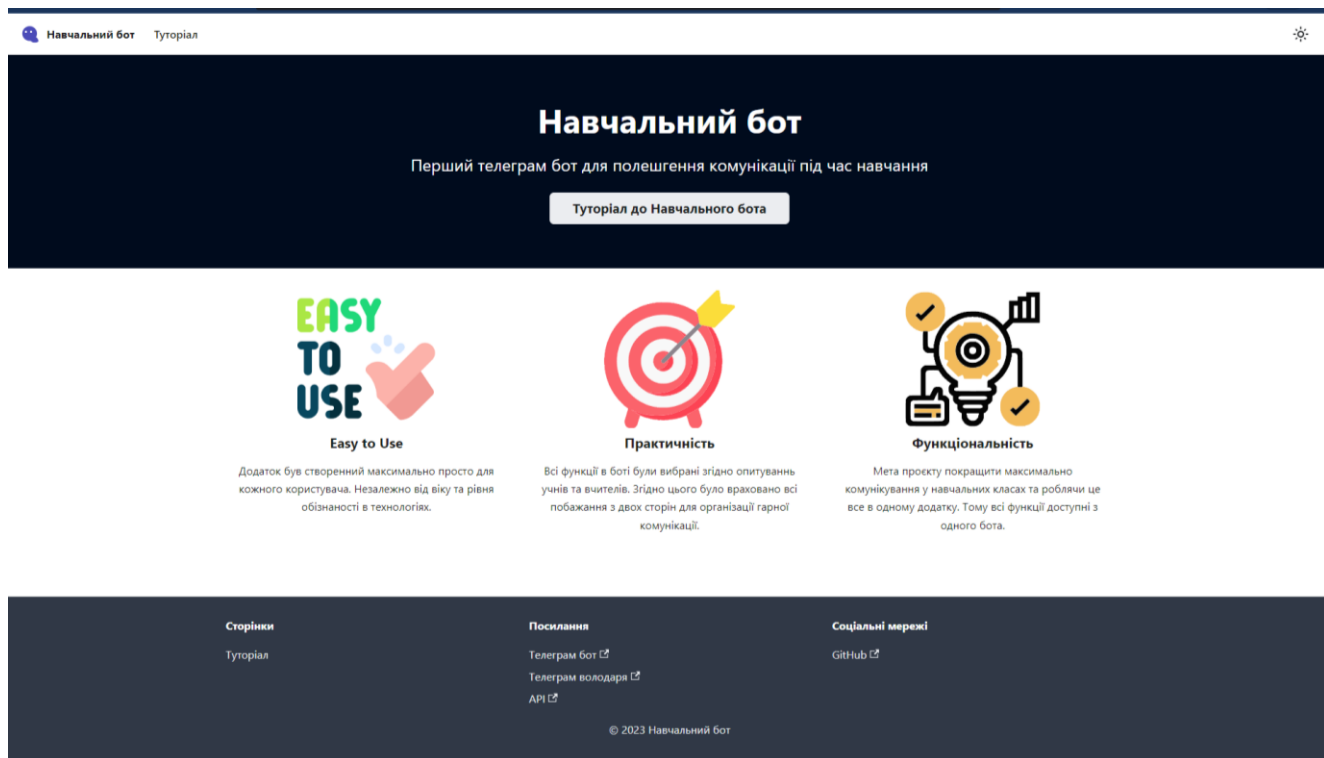


Рис. 3.1. Знімок екрану головної сторінки (рис. автора)

Щоб оптимізувати сайт для пошукових систем потрібно крім цього опублікувати його в google search console та bing webmaster, щоб їх можна було побачити в найпопулярніших пошукових системах України. Та далі потрібно проіндексувати сторінки та зробити sitemap. І через декілька годин або днів сайт можна буде знайти за запитом в пошукових системах

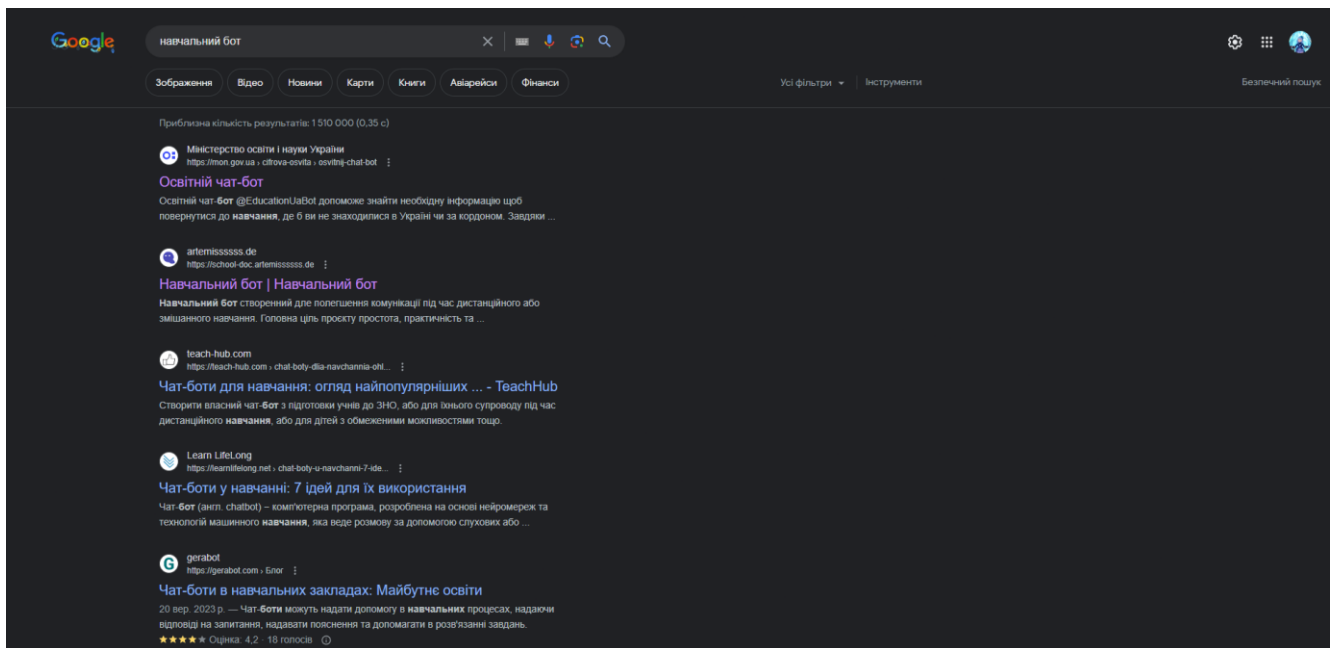


Рис. 3.2. Знімок екрану в пошуковій системі Google(рис. автора)

<https://github.com/Artemissssss/artem-school-doc> - посилання на код сайту

<https://school-doc.artemissssss.de> - посилання на сайт

РОЗДІЛ 4

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТА АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1. Методика проведення дослідження

Метою дослідження було протестувати проєкт в аудиторії 10-Б класу Харківського ліцею № 141. Та в результаті зробити опитування учасників про їх досвід використання розробленого проєкту та чи те що їм не подобається в проєкті.

Було задане опитування учням та вчителю: «Добрий вечір, прошу пройти наступне опитування пов'язане з вашим досвідом використання мого телеграм бота! Опитування анонімне, тому сильно прошу дати відгук, а ще краще якщо ви напишите побажання/недоліки через /chat або до @honkai_star_rails.», відповідями в опитуванні були з одним варіантом відповіді:

- Все подобається
- Подобається, але є недоліки
- Нейтрально
- Не подобається, але є плюси
- Не подобається
- Складно відповісти

У дослідженні взяли участь 30 учнів 10-Б класу та один вчитель.

Комунікаційна система використовувалася на уроках фізики протягом одного місяця. Через телеграм бота протягом одного місяця вчитель задавав домашні завдання, контрольні, тести, ставив уроки, події. Також вчитель заповнював матеріали, а учні в свою чергу робили самостійно свої конспекти, знімки екранів уроку та викладали в файли або в файли уроку, щоб кожен зміг їх потім побачити. Також учні та вчитель комунікували між собою для уроків фізики за допомогою телеграм бота. В останній день дослідження телеграм бот надіслав анонімне телеграм опитування з однією відповіддю учням та вчителю класу з проханням його пройти.

4.2. Результати дослідження

Результати проведення дослідження приведені на зображенні 4.1

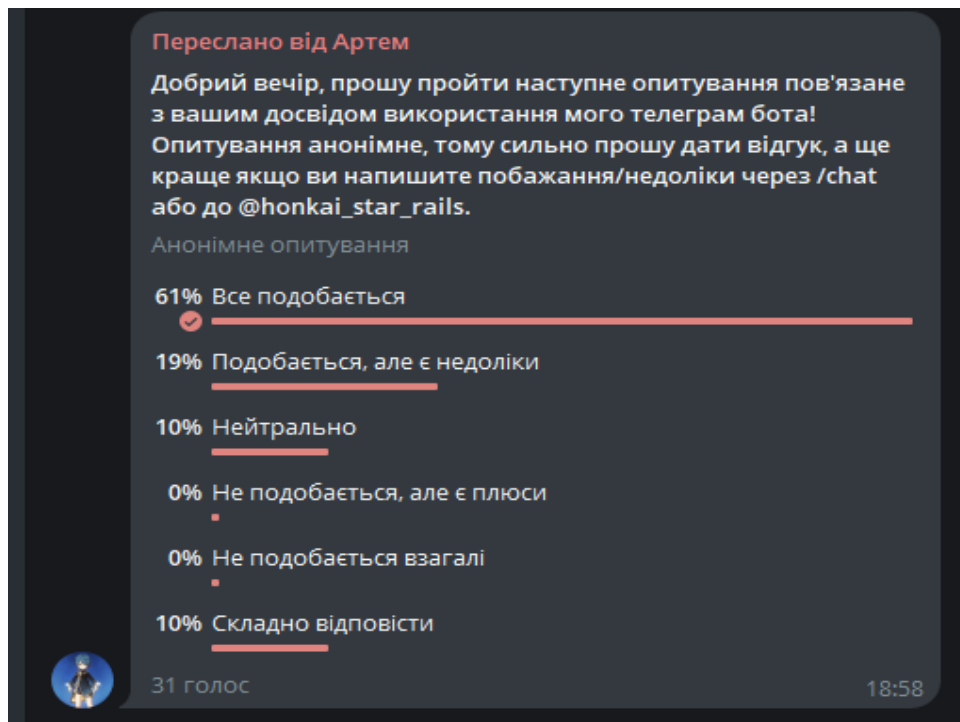


Рис 4.1. Результати отримані в рамках проведених досліджень (рис. автора)

Також окремо були надіслані такі зауваження, як повільність виходу навчальної документації по функціоналу бота та періодичні запитання з вибором в якому класі хоче бути користувач.

Отже за результатами виконаного дослідження:

- Більшість користувачів має позитивний досвід використання бота і мали змогу полегшити комунікацію на уроках фізики.
- Є недоліки пов'язані з технічною реалізацією проекту, які в майбутньому є цілями для усунення.

ВИСНОВКИ

В результаті роботи було проаналізовано необхідні функції та наявні системи комунікації між учнями та вчителями. Згідно цього аналізу було виділено всі недоліки та переваги цих систем.

Було розроблено перелік функцій згідно проведеного аналізу існуючих комунікаційних систем, необхідних для забезпечення лабораторного процесу і особливо для використання лабораторного обладнання в індивідуальному та віддаленому режимі.

Візуалізовано всі потоки інформації проєкту для створення плану реалізації проєкту за допомогою діаграми потоку даних, яка була складена за допомогою переліку функцій які були взяті при аналізі.

В роботі також було вивчено психологічні особливості сприйняття інформації учнями та викладачами.

Згідно цього було вибраний максимально простий інтерфейс для кожного користувача з максимальною легкістю використання.

Було створено комунікаційну систему з сервісною та користувацькою частиною, яка складається з 4 елементів, а саме база даних, API, телеграм бот, сайт боту.

Створену комунікаційну систему було апробовано у реальному навчанні, що дозволило вдосконалити можливості системи та запропонувати подальший розвиток проєкту на наступні роки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Юлія Назаренко, Освіта українських біжен_ок за кордоном (24 лютого – 20 червня 2022). Available: https://cedos.org.ua/wp-content/uploads/osvita-ukrayinskyh-bizhen_ok-za-kordonom-24-lyutogo-%E2%80%94-20-chervnya.pdf
2. Google Classroom. Available: <https://classroom.google.com>
3. Microsoft Teams. Available: <https://www.microsoft.com/uk-ua/microsoft-teams/group-chat-software>
4. Moodle. Available: <https://Moodle.org>
5. Schoolguide. Available: <https://Schoolguide.online>
6. MySQL. Available: <https://www.mysql.com>
7. PostgreSQL. Available: <https://www.postgresql.org>
8. MongoDB. Available: <https://www.mongodb.com>
9. Telebot. Available: <https://github.com/mullwar/telebot>
10. Docusaurus. Available: <https://docusaurus.io>
11. Про.Світ. Дослідження стану реалізації дистанційного навчання в Україні. Available: https://nus.org.ua/wp-content/uploads/2020/05/Research2020_ProSvit_MF1.pdf
12. Бондаренко, С. Г. Досвід використання сервісу Google Classroom для дистанційного навчання / Бондаренко С. Г., Шахновський А. М., Сангінова О. В. // Комп'ютерне моделювання і керування в техніці та технологіях КМКТТ-2021 : збірник наукових статей Дев'ятої міжнародної науково-практичної конференції, Київ, 12-14 травня 2021 р. – Київ : КПІ ім. Ігоря Сікорського, 2021. – С. 287–293. – Бібліогр.: 5 назв. Available: https://ela.kpi.ua/bitstream/123456789/41433/1/КМКТТ-2021_p287-293.pdf
13. Глазунова, О. Г., Корольчук, В. І., Волошина, Т. В., & Саяпіна, Т. П. (2023). ОЦІНЮВАННЯ MICROSOFT TEAMS ЯК ІНСТРУМЕНТУ ДЛЯ СИНХРОННОЇ ВЗАЄМОДІЇ ПІД ЧАС ДИСТАНЦІЙНОГО ТА ГІБРИДНОГО НАВЧАННЯ. Електронне наукове фахове видання

“ВІДКРИТЕ ОСВІТНЄ Е-СЕРЕДОВИЩЕ СУЧАСНОГО УНІВЕРСИТЕТУ”, (14), 1–11. <https://doi.org/10.28925/2414-0325.2023.141>

14. Каркач, А., & Семигіна, Т. (2022). Використання платформи moodle для навчання цифровим технологіям слухачів університетів третього віку. *Social Work and Education*, Vol. 9, No. 4. pp. 451-464. DOI: 10.25128/2520-6230.22.4.1