

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

НУБІП України

Факультет інформаційних технологій

НУБІП України

УДК 004.9:005.95

«ПОГОДЖЕНО»

Декан факультету

інформаційних технологій

«ДОПУСКАЄТЬСЯ ДО

ЗАХИСТУ»

Завідувач кафедри комп'ютерних наук

НУБІП України

Глазунова О.Г., д.п.н., професор

Голуб Б.Л., к.т.н., доцент

202_р.

202_р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

НУБІП України

на тему Система обліку та аналізу діяльності працівників підприємства

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Програмне забезпечення інформаційних систем»

Орієнтація освітньої програми освітньо-професійна

Гарант освітньої програми

НУБІП України

(науковий ступінь та вчене звання)

(підпис)

(ПІБ)

Керівник магістерської кваліфікаційної роботи

старший викладач

Панкрат'єв В.О.

доц., к.т.н.

Дудник Алла Олексіївна

НУБІП України

Виконав

(підпис)

Логвин Б.Л.

(ПІБ студента)

КИЇВ-2022

НУБІП України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет (ННІ)

ЗАТВЕРДЖУЮ

Завідувач кафедри доц.,к.т.н. _____

Голуб Б.Л.

(науковий ступінь, вчене звання) (підпис) (ШБ)
" " 20 року

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ СТУДЕНТУ

Логвина Богдану Леонідовичу

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Програмне забезпечення інформаційних систем»

Орієнтація освітньої програми освітньо-професійна

Тема магістерської кваліфікаційної роботи Система обліку та аналізу діяльності працівників підприємства затверджена наказом ректора НУБіП України від 11.10.2022р. № 1434”С”

Термін подання завершеної роботи на кафедру _____
(рік, місяць, число)

Вихідні дані до магістерської кваліфікаційної роботи

1. Аналіз даних що накопичені і процесі роботи працівників.
2. Дані про підприємства.
3. Існуючі системи аналізу роботи працівників підприємств.

Перелік питань, що підлягають дослідженню:

1. Дослідження системи аналізу роботи працівників
2. Створення моделей предметної області
3. Моделювання та проєктування системи аналізу та обліку

Дата видачі завдання " " 20 р.

Керівник магістерської кваліфікаційної роботи _____
(підпис)

Панкрат'єв В.О _____
(прізвище та ініціали)

Завдання прийняв до виконання _____

Логвин Б.Л _____
(підпис) (прізвище та ініціали студента)

ЗМІСТ

Вступ.....	4
1 Системний аналіз предметної області.....	7
1.1 Аналіз предметної області.....	7
1.2 Аналіз наявних рішень.....	11
1.3 Постановка завдання.....	18
2 Моделювання системи.....	20
2.1 Функціональне моделювання.....	20
2.1.1 IDEF0.....	22
2.2 Об'єктно-орієнтоване моделювання.....	24
2.2.1 Діаграма прецедентів.....	24
2.2.2 Діаграма активності.....	26
3 Розробка системи.....	27
3.1 Інформаційне забезпечення.....	27
3.1.1 Логічна модель даних.....	27
3.1.2 Вибір інструментів для проектування БД.....	29
3.2 Програмне забезпечення.....	31
3.2.1 Вибір інструментів розробки.....	31
4 Результати дослідження.....	33
4.1 Розрахунок Ефективності діяльності працівника.....	33
Висновки.....	37
Список використаних джерел.....	38

ВСТУП

НУБІП України

Продуктивність організації напряму залежить від ефективності роботи працівників. Поведінка співробітників дає змогу підприємству виконувати певну функцію досягати необхідних економічних показників. Однак, трудова поведінка може відрізнятися для різних людей: для одних характерна старанність і відповідальність, для інших – прагнення мінімального виконання обов'язків незалежно від якості вони будуть, чи буде оптимальним результат.

НУБІП України

Значення мотивації в рамках сучасних систем управління незаперечно велике, адже саме мотивація змушує працівника працювати з максимальною віддачою, що гарантує ефективність діяльності організації. Мотивація сприяє розвитку економіки країни, забезпечує рівень добробуту громадян, адже будь-яка людина працює для того, щоб заробити, що дасть їй змогу задовольнити власні потреби і потреби, і саме цей фактор має суттєвий вплив на прояв активності людини у трудовій діяльності. Провідним аспектом оптимального функціонування сучасного підприємства в конкурентних умовах ринку є забезпечення ефективності його господарської діяльності. Необхідність оцінки ефективності виробничої діяльності підприємства зумовлюється потребами формування нових цілей, принципів та засад господарювання орієнтованих на задоволення потреб споживачів та вимог ринку. Оцінка ефективності виробничих процесів являє собою інтегровану частину комплексної системи цілей розвитку підприємства, яка охоплює ряд факторів та заходів, що визначають рівень організації виробництва. Управлінські рішення в даному випадку мають на меті раціоналізацію виробництва за певних трудових, технічних і технологічних умов, що полягають у забезпеченні функціонування підприємства без непередбачуваних і значних витрат, виявленні та використанні в короткі терміни резервів підвищення ефективності виробництва. Оцінка ефективності займає важливе місце в системі управління діяльністю підприємства, оскільки являє собою ефективний інструмент для інформаційно-аналітичного забезпечення діяльністю підприємства. Результати оцінки складають основу для прийняття

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

управлінських рішень, при цьому вони можуть бути використані як для визначення стратегічних, так тактичних цілей. Дане дослідження має стати теоретичною базою для проведення оцінки ефективності виробничої діяльності підприємства, пошуку можливих напрямків її вдосконалення та, як наслідок, прийняття раціональних управлінських рішень в сфері виробництва.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

Актуальність дослідження полягає у покращенні ефективності роботи працівників підприємства що впливає на зріст економіки країни.

Об'єктом дослідження є показники роботи працівників.

Предмет дослідження полягає у виявленні факторів що впливають на ефективність праці.

Завдання дослідження полягає в аналізі даних з ефективності роботи працівників.

Метою дослідження є виявлення факторів що впливають на сприяння росту економіки країни

Методами дослідження є використання JavaScript, Node.js для створення веб-додатку для системи моніторингу.

Структура роботи представлена у вигляді чотирьох основних частин, висновків та переліку використаних джерел. В першому розділі міститься опис

головних критерій системного аналізу та обробки даних та аналіз наявних рішень, а також, на основі цих даних, сформовано постановку завдання дослідження. Другий розділ містить створені моделі системи за допомогою

різних методологій. Третій розділ демонструє процес створення системи та застосування різних інструментів. Останній розділ представляє результати

розробки та аналізу даних.

1 СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

Функціонування ринкових відносин на сучасному етапі розвитку України висуває перед відповідними державними органами важливі завдання щодо перебудови економіко-аналітичної роботи з вивчення зайнятості трудових ресурсів, перерозподілу робочої сили, перепідготовки кадрів, ринку праці тощо.

Це повною мірою стосується підприємств житлово-комунального господарства, де підвищення рівня та якості обслуговування міського населення потребують

значної кількості робочої сили і покращення ефективності її використання.

При цьому слід наголосити, що праця, наявність і ефективне використання трудових

ресурсів, підвищення продуктивності праці відіграють вирішальну роль у процесі виробництва продукції (послуг).

У зв'язку з цим великої ваги набуває аналіз праці, оцінка ефективності її використання у виробничому процесі, вивчення впливу окремих трудових факторів (чисельність працівників, їхня структура, тривалість робочого періоду

і робочого дня одного працівника, продуктивність праці та інше) на зміну кінцевих результатів діяльності аналізованого підприємства.

Виходячи з цього, під час аналізу праці необхідно вивчити динаміку чисельності працівників за останні періоди часу, забезпеченість ними підприємства, їхній склад і структуру за окремими ознаками, рух робочої сили, ефективність використання робочого часу, проаналізувати продуктивність

праці, визначити вплив трудових факторів на виконання виробничої програми.

Аналіз динаміки чисельності працівників за окремі періоди часу здійснюють за допомогою таких показників: середньооблікової

чисельності

працівників і облікової чисельності працівників на початок (кінець) аналізованих періодів часу. Для цих показників обчислюють відповідні показники динаміки (абсолютні й відносні прирости, темпи зростання

тощо) і

на основі їхнього порівняння виявляють закономірності та окремі тенденції у

зміні кадрового потенціалу підприємства.

Управління персоналом є складовою частиною управління підприємством загалом. Процес управління підприємством можна розбити на такі складові:

управління технікою, економікою і кадрами. Найскладнішим із трьох перелічених вище об'єктів є управління кадрами, оскільки їм належить провідна роль в управлінні персоналом.

Управління кадрами пов'язане з розробленням і реалізацією політики, що включає планування, наймання, вибір, розміщення робочої сили, навчання і підготовку працівників, просування по роботі, умови найму, методи і стандарти

оплати праці; умови праці та послуги; формальні й неформальні зв'язки; консультування як наймачів, так і тих, хто наймається: переговори про зарплату й умови роботи.

Процес управління кадрами включає в себе три основні етапи. Аналіз ситуації включає визначення потреб підприємства в робочій силі, що

впливають із планів фірми, прогнозу збуту і планів виробництва, аналізу сильних і слабких сторін наявної робочої сили. Складовою частиною цього етапу

аналіз придатності наявних кадрів, з урахуванням економічної та демографічної ситуації, що склалася, у країні та районі розташування підприємства.

Кадрова політика та її цілі мають відповідати цілям підприємства. Цілі включають заходи щодо збільшення продуктивності праці та посилення обороту робочої сили.

Контроль за виконанням рішень керівництва підприємства з кадрових питань полягає в зіставленні виконання з наміченими цілями, виконанні заходів для усунення виявлених відхилень під час найму, навчання і звільнення працівників. До обов'язків керуючого кадрами також входить консультування інших лінійних керівників з кадрових питань.

В обов'язки керуючого кадрами входить керівництво роботою відділу кадрів, консультування керівників підприємства всіх рівнів з кадрових питань, доведення прийнятих керівництвом підприємства рішень до всіх працівників. Планування робочої сили, проведення переговорів, регулювання виробничих відносин, виступ як організатора трудових відносин, виступ як представника компанії на зовнішньому і внутрішньому рівнях.

Здатність керівництва підприємства максимально ефективно використовувати наявні трудові ресурси багато в чому залежить від того, наскільки правильно будуть розраховані і контрольовані витрати на виконання трудових процесів.

Планування трудових ресурсів є одним із важливих розділів як стратегічних, так і оперативних планів підприємства. Процес планування трудових ресурсів містить у собі вирішення таких завдань:

- визначення загальних цілей на планований період;

- ув'язка цих цілей з вимогами робочої сили, враховуючи передбачувані зміни виробничого процесу, структури продукції тощо;

- оцінка рівня плинності та складання прогнозу кількості звільнених (прийнятих) працівників;

- визначення місця і часу виникнення дефіциту робочої сили;

- визначення шляхів подолання можливих проблем;

- розвиток робочої сили, адаптація до темпу технологічних, економічних і соціальних змін;

- координація та управління всіма видами політики, що впливає на

ефективність використання робочої сили.

Під час планування трудових ресурсів необхідно проводити як періодичний, так і систематичний аналіз властивостей і характеристик трудових ресурсів для виявлення змін у їхньому розподілі та складі.

Основні завдання системи аналізу:

- накопичення даних системи;
- проведення аналізу та обробки звітів працівників
- визначення коефіцієнту ефективності;
- визначення часу для виконання задачі;

Організація роботи персоналу передбачає чіткий алгоритм дій кожного співробітника офісу. Першочергове завдання керівника компанії - організація злагодженої роботи і контроль. Існує безліч програм для підрахунку неефективності праці та використуваного часу, оскільки вручну виявити проблеми у великому колективі складно, на це керівнику потрібен час.

Завдяки використанню сучасних автоматизованих підходів, процес організації контролю роботи співробітників керівником стає простим. У них автоматично формуються звітності та оповіщення щодо роботи кожного.

1.2 Аналіз наявних рішень

Способи контролю, обліку та оцінки робочого часу персоналу з часом еволюціонували, та все ж організації, особливо державні установи, дотримуються традиційних підходів. Наприклад, призначають відповідального, вахтера або чергового, за ведення журналу обліку робочого часу. Відповідальний фіксує час приходу-відходу і систематично готує звіти для керівника.

Інший варіант - ввести до штату посаду адміністратора, який перебуватиме в одному приміщенні з колегами або в окремому кабінеті та контролюватиме безперервність робочого процесу.

Ще один спосіб полягає в тому, щоб співробітники вели особисту звітність, самостійно контролювали і фіксували витрачений робочий час. Метод допомагає оцінити виконану роботу з точки зору виконавців і розвиває самостійність.

Більш звичний спосіб - встановити систему контролю та управління доступом із застосуванням перепусток або сканерів відбитка пальця. Інформація щодо кожного працівника зберігається у файлі та доступна для перегляду в будь-який момент.

Надійний, але дорогий метод моніторингу та обліку робочого часу персоналу компанії - система відеоспостереження. Буде потрібно витратити кошти не тільки на купівлю та встановлення відеокамер, а й на зарплату окремого працівника, до чийх обов'язків належить безперервний контроль діяльності персоналу та фіксація порушень правил внутрішнього трудового розпорядку. Крім того, впровадження системи відеоспостереження викликає у співробітників психологічний дискомфорт від постійного спостереження. Тому встановлення камер найчастіше - це особливий захід, який застосовується на критично важливих об'єктах.

IT-рішення для контролю робочого часу

Контролювати роботу невеликого колективу вручну - посилає завдання для безпосереднього керівника або одного виділеного фахівця. Але якщо штат підприємства налічує не одну сотню осіб, які до того ж працюють у філіях або віддалено, впоратися з таким завданням під силу тільки автоматичі.

Крім того, моніторинг роботи персоналу потрібен не тільки для фіксації результативності праці окремого працівника, а й для забезпечення безпеки підприємства. Два завдання одночасно виконують IT-рішення для контролю продуктивності протягом робочого часу. Автоматизовані системи фіксують інформацію, проводять аналіз і формують звіти.

Вартість впровадження автоматичної програми моніторингу робочого часу покривається за рахунок мінімізації витрат роботодавця, пов'язаних із відсутністю персоналу на робочому місці: прогулами, запізненнями, тривалими перервами. IT-інструменти фіксують і аналізують дані, на основі яких керівництво заохочує "передовиків" офісу, тим самим мотивуючи весь колектив працювати більш продуктивно і дисципліновано.

Автоматизовані IT-системи дають змогу виявити:

порушників дисципліни - співробітників, які систематично запізнюються або йдуть додому раніше, ніж треба, завзятих курців або любителів кави, тривалість "п'ятихвилинних" перерв у яких перевищує тривалість роботи працівників, які протягом робочого часу вирішують особисті справи, читають новини та стрічки соціальних мереж, ведуть дружне, а не службове листування в месенджерах, грають на комп'ютері.

співробітників, які настільки перевантажені робочими завданнями і змушені постійно затримуватися на роботі, що загрожує професійним вигоранням і - в крайньому разі - нервовим зривом.

співробітників, які не задоволені позицією з тих чи інших питань і зайняті пошуком нової роботи.

Автоматизований моніторинг робочого часу не обмежується встановленням спеціалізованого ПЗ і охоплює також установку контрольного обладнання при вході/виході з будівлі; обладнання пунктів пропуску; використання системи особистих ідентифікаторів для кожного працівника; фіксацію пересування персоналу під час роботи по території підприємства за допомогою GPS-маячків.

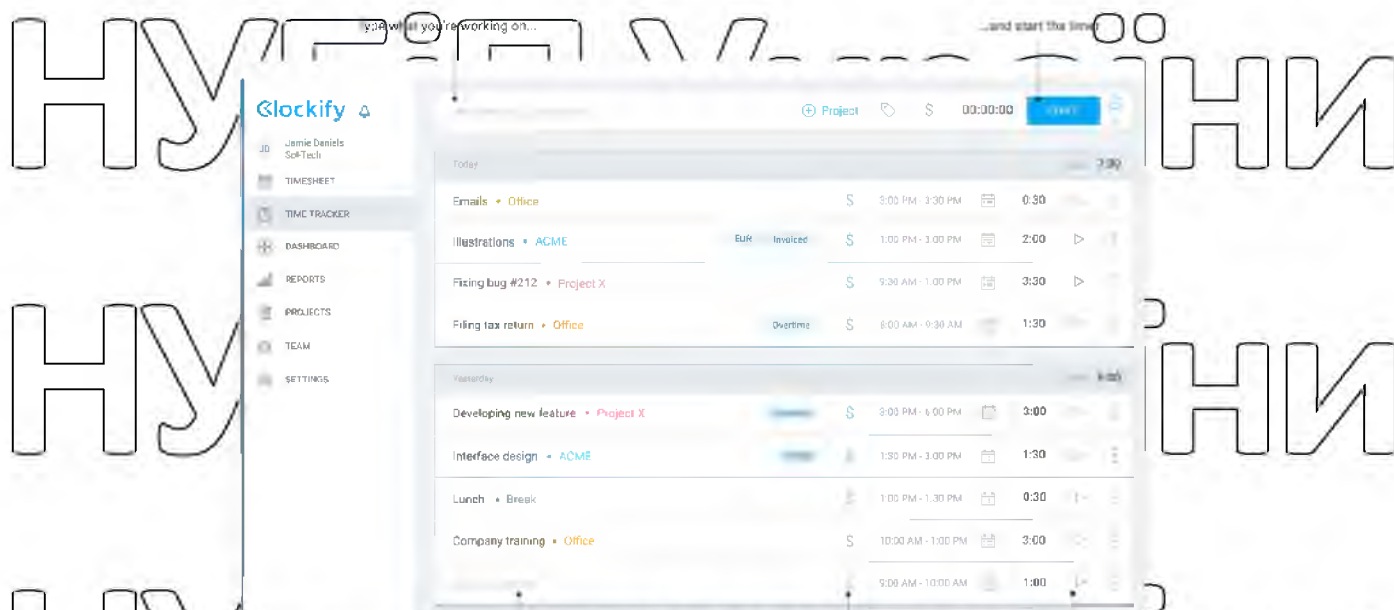
Комплексний підхід до контролю робочого часу передбачає аналіз роботи фахівців за контрольними точками: штатним розкладом, планом робіт, термінами виконання завдань, територією виконання обов'язків, виробничим графіком.

Як обрати IT-рішення для контролю робочого часу

Під час вибору програмного забезпечення для контролю робочого часу та оцінювання ефективності персоналу варто зважати на спеціалізацію рішення та одразу відповісти на запитання, з якою метою встановлюється програма: лише збирати дані чи аналізувати ефективність роботи персоналу.

Існує багато різних варіацій систем для прослідкування роботи працівника підприємства але всі вони мають деякі нюанси для інтеграції їх в різні підприємства. Нижче наведено декілька варіантів таких систем.

1. Clockify



Безкоштовна версія: доступна частина функцій

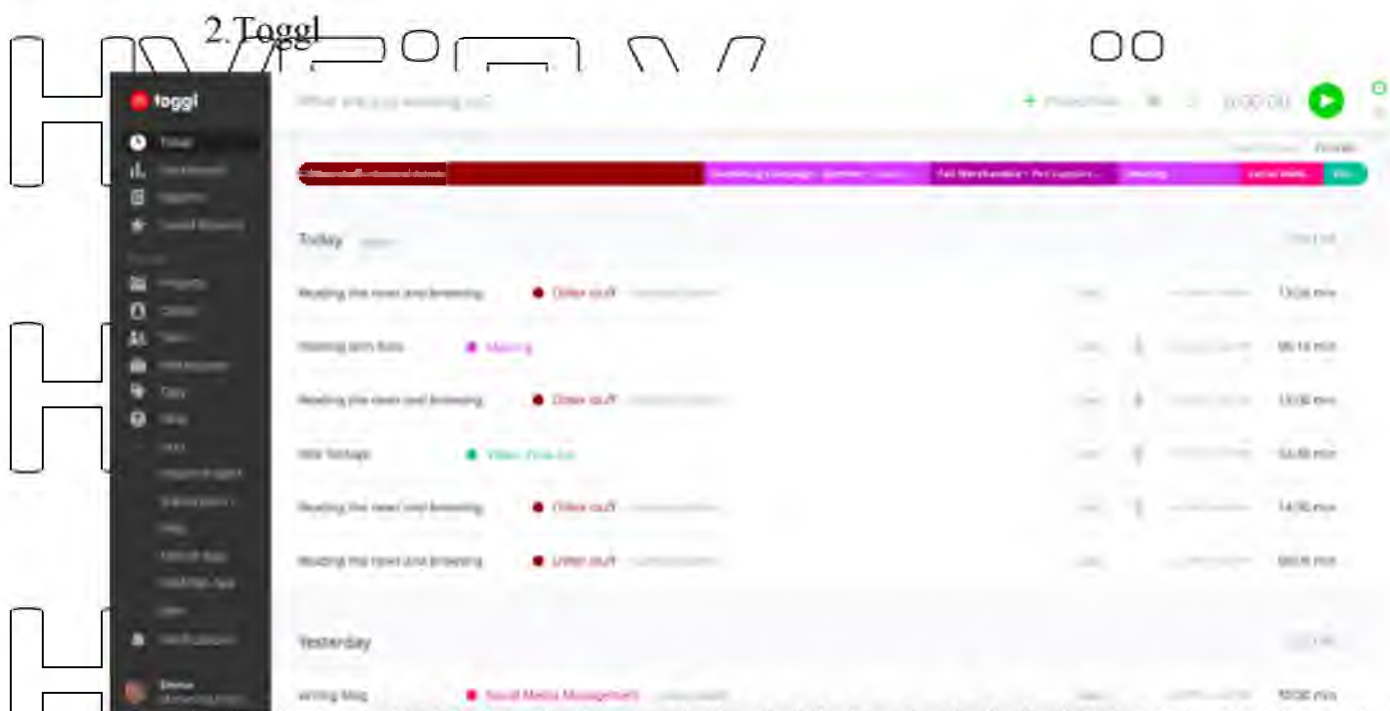
Вартість використання: від 9,99 до 29,99 доларів залежно від тарифу

У топі рішень для тайм-трекінгу Clockify заслужив особливе місце за поєднання максимально спрощеного інтерфейсу і широких можливостей з обліку відпрацьованого часу співробітників з інтеграцією з відомими додатками (від JIRA до Google Календар). Підходить фрілансерам для виставлення рахунків або підрисметвам для ведення докладних робочих розкладів.

Звіти: доступні різні види таймерів з тонким налаштуванням і великий набір звітів, єдиний нюанс - деякі функції не працюють з Windows, тільки на Mac

Функціональність: немає обмежень за кількістю проектів, є відстеження статусу, тимчасові мітки, оцінки, нагадування та сповіщення

Команда: деякі функції (SSO, формування рахунків, приватні канали) оплачуються додатково, є ієрархія, групування, редагування користувачів.



Безкоштовна версія: 5 пристроїв, тільки базові функції

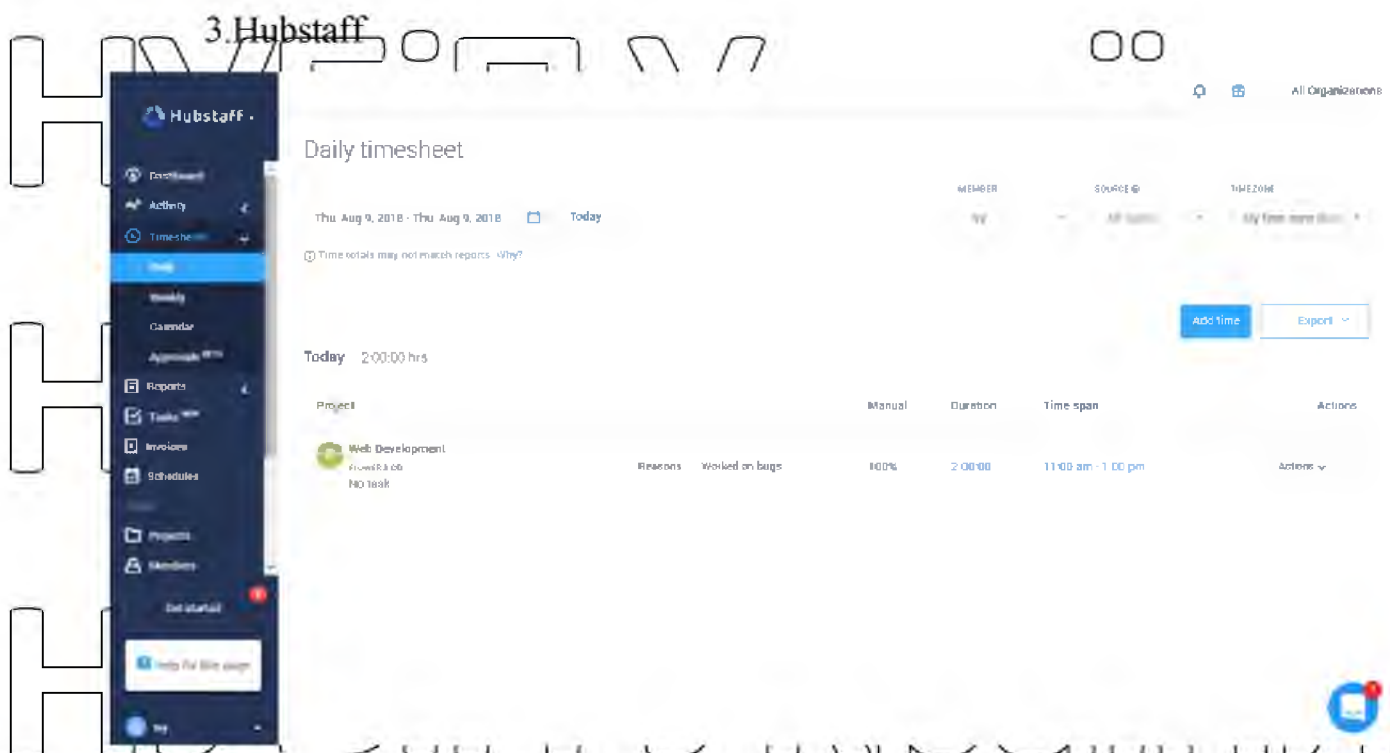
Ціна: один користувач - 10 доларів/місяць (базовий), 20 доларів/місяць (професійний)

Одне з найшвидше розвиваючихся рішень. Оптимально підходить для контролю і тайм-трекінгу фрілансерів або дисциплінування самозайнятих. З точки зору обліку відпрацьованого часу співробітників рішення підійде невеликим командам. Коли типового інтерфейсу недостатньо (персоналу об'єктивно багато і потрібні великі функції/тайм-моніторинг), вигідніше вибрати щось з альтернативних варіантів. При цьому Тоггл для своїх можливостей досить дорогий.

Складання звітів: повний спектр функцій таймера (деякі тільки в професійному тарифі, наприклад - нагадування), є шаблони і нестандартні звіти всіх типів, але більша частина також у про-версії

Функціональність: присутній повний набір сучасних функцій, але вони платні і розрізняються за тарифами, кількість проєктів не обмежена.

Команда: немає SSO і контрольних рахунків, доступні приватні канали та угруповання, відстеження командної діяльності



Безкоштовна версія: 1 користувач, базовий тариф, урізана частина функцій.

Вартість використання \$/місяць за 1 користувача (базовий), 10 (професійний), 20 (преміальний)

Рівень освоєння: дуже високий

Універсальний застосунок підійде для контролю фрилансерів та обліку часу офісних співробітників у компаніях будь-якого розміру. Програма трекінгу розроблена спеціально для Windows. Інтерфейс не найпростіший, особливо для співробітників, які раніше не працювали з подібними сервісами. Інструмент складний в освоєнні, але за підсумком керівник отримує багатоцільовий інструмент для відстеження всіх аспектів виконання фахівцями своїх посадових зобов'язань.

Складання звітів: в автономному режимі фіксує всі дії та бездіяльність, єдиний планш - немає веб-таймера, доступне вивантаження звітів у будь-яких форматах

Функціональність: немає шаблонів проектів, загальна кількість залежить від тарифу, реалізовано можливість градації за пріоритетами, відстеження статусу, групування

Команда: кількість користувачів залежить від тарифу, наявність приватних каналів зв'язку, можливість ієрархічної організації, індивідуальне відстеження

Додатково: реалізовано API та інші можливості, крім виділеного хостингу

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

1.3 Постановка завдання

Система аналізу повинна в себе включати:

- збирання даних про час за яке виконано задачу;
- аналізування і оцінювання коефіцієнту ефективності;
- інформування в разі помилок в створенні звіту

Вимоги до функціональності програмного забезпечення:

- Програмне забезпечення повинне забезпечувати можливість:
 - інтерфейс для експорту даних з інших систем;
- Інтерфейс користувача програмного забезпечення повинен бути максимально простий та зручний для використання.
- Ролі користувачів.
 - Для адміністратора, повинен бути окремий інтерфейс для керування працівниками.

Системний аналіз — науковий метод пізнання, що являє собою послідовність дій з установлення структурних зв'язків між змінними або елементами досліджуваної системи. Спирається на комплекс загальнонаукових, експериментальних, природничих, статистичних, математичних методів.

Єдиної методики системного аналізу у наукових дослідженнях поки що немає. У практиці досліджень він застосовується з використанням таких методик: процедур теорії дослідження операцій, яка дає змогу дати кількісну оцінку об'єктам дослідження;

аналізу систем дослідження об'єктів в умовах невизначеності;

системотехніки, яка включає проектування і синтез складних систем у процесі дослідження їх функціонування (проектування і оцінка економічної ефективності АСК технологічних процесів та ін.).

Важливе значення системний аналіз має в управлінні персоналом.

Системний аналіз — науковий метод пізнання, що являє собою послідовність дій з установлення структурних зв'язків між змінними або елементами досліджуваної системи. Спирається на комплекс загальнонаукових, експериментальних, природничих, статистичних, математичних методів.

НУВБІП УКРАЇНИ

Системний аналіз виник в епоху застосування комп'ютерної техніки у бізнесі та інших сферах діяльності. Успіх його застосування у вирішенні складних завдань

багато в чому визначають можливості сучасних інформаційних технологій.

Результатом системних досліджень є, як правило, вибір цілком певної

НУВБІП УКРАЇНИ

альтернативи плану розвитку регіону, параметрів конструкції тощо. Тому витоки системного аналізу, його методичні концепції лежать в тих дисциплінах, які займаються проблемами прийняття рішень: теорії операцій і загальної теорії

управління.

НУВБІП УКРАЇНИ

Системний аналіз включає здатність до аналізу, прогнозування, проектування прийняття рішень в складних системах різної природи на основі системної

методології.

НУВБІП УКРАЇНИ

Фахівці з системного аналізу проектують, створюють й експлуатують комп'ютерні системи аналізу, прогнозування, управління та проектування динамічних процесів в макроекономічних, технічних, технологічних,

екологічних і фінансових об'єктах, здійснюють аналіз бізнес-процесів з погляду

НУВБІП УКРАЇНИ

їхньої подальшої автоматизації, розробляють технічні завдання та специфікації, тестують програмне забезпечення, формують аналітичні звіти.

НУВБІП УКРАЇНИ

НУВБІП УКРАЇНИ

2 МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Функціональне моделювання

SADT – це методологія структурного аналізу та проектування (Structured Analysis and Design Technique) була розроблена в кінці 60-х рр. XX ст. для опису досить складних штучних систем, і є найпоширенішою методологією аналізу функціонального середовища, яка відображає такі системні характеристики, як управління, зворотній зв'язок, виконавців. Моделювання здійснюється за допомогою графічних елементів та спеціальних правил їх використання. Ця технологія була розроблена Дугласом Россом.

Для проектування інформаційного середовища методологія SADT являє собою сукупність методів, правил і процедур, що призначені для побудови функціональної моделі стратегічного планування на підприємстві.

Функціональна модель SADT відображає функціональну структуру об'єкта, тобто функції та дії і описує зв'язок між ними. Правила SADT включають:

1. застосування правила 3-6 блоків до формування рівнів декомпозиції, яке обмежує кількість блоків на кожному з рівнів;
2. логічність та послідовність блоків у діаграмі;
3. унікальність найменувань та міток у діаграмі;
4. застосування універсальних правил для графіки (до блоків та дуг);
5. оптимальний поділ вхідної та управлінської інформації, що включає правило визначення ролі даних;
6. чітке розділення організаційної структури та функціональної моделі.

Ці правила допомагають не відволікатися на інші об'єкти, включати в дослідження тільки головні об'єкти та обґрунтовані зв'язки між ними та зовнішнім середовищем, показувати інформацію ясно і чітко, а також відокремлення неважливих, другорядних об'єктів під час обґрунтування напрямків стратегічного планування.

Методологія SADT може використовуватися для моделювання широкого кола інформаційних систем і визначення вимог і функцій, які і включаються для стратегічного планування діяльності підприємства. Потім ця методологія може використовуватися для розробки системи, яка задовольняє ці вимоги і реалізує ці функції.

Для розроблення стратегічного планування на основі SADT-методології потрібно дотримуватися таких кроків:

1. Підготовчий: визначення об'єкту дослідження та його елементів, мети, аналіз зовнішніх і внутрішніх

факторів впливу на організацію.

2. Визначення зовнішньої точки спостереження за всіма дослідженнями (зазвичай головний директор), що

буде аналізувати та оперативно повідомляти про відхилення)

3. Моделювання: збір та систематизація даних про конкурентів, продукцію, ринок а також внутрішній аналіз

організації за допомогою звітностей бухгалтерського обліку і спостережень.

4. Вибір декомпозиції, для аналізу всього підприємства, але розгляду всіх його елементів як окремих і взаємопов'язаних. Є такі види декомпозицій: функціональна стратегія, стратегія декомпозиції ЗР (people-paperprocedures), стратегія декомпозиції на основі аналізу життєвого циклу системи, стратегія з фізичного процесу.

5. Створення дерева функцій підприємства і їх класифікація.

SADT (Structured Analysis and Design Technique) – методологія аналіза та проектування систем. Дану методологію використовується для великого спектру бізнес-процесів:

- проектування програмного забезпечення;
- проектування систем підтримки та діагностики;
- автоматизація виробництва;
- конфігурація комп'ютерних систем;

Представлена методологія має широкую підтримку, що стало ініціатором розробки стандарту IDEF0.

2.1.2 IDEF0.

IDEF0 - це методологія графічного опису систем і процесів діяльності організації як безлічі взаємозалежних функцій. Вона дозволяє досліджувати функції організації, не пов'язуючи їх з об'єктами, що забезпечують їх реалізацію.

У стандарті IDEF0 за допомогою входу показують об'єкти - інформаційні та матеріальні потоки, які перетворюються в бізнес-процесі. За допомогою

управління показуються об'єкти - матеріальні та інформаційні потоки, які перетворюються на процесі, по потрібні для його виконання. Використовуючи механізми IDEF0 можна відображати інструменти та ресурси, за допомогою яких

бізнес-процес реалізується (наприклад, технічні засоби, люди, інформаційні системи і т.д.). Вихід бізнес-процесу, описаного в стандарті IDEF0, повністю відповідає за змістом виходу процесу, описаного за допомогою DFD-схеми.

IDEF0 стандарт призначений для моделювання підприємств та їх діяльності, що дозволяє полегшити створення системи. На рисунку 2.1 наведено

діаграму IDEF0 для системи обліку та аналізу діяльності працівників підприємства.

Розглянемо кожен блок діяльності більш детально:

- Збір даних – перша діяльність системи, що має дані про виконану задачу робітником, це означає, що саме ці данні піддаються аналізу.

Вихідними потоками є дані показників, що виходить із блока діяльності.

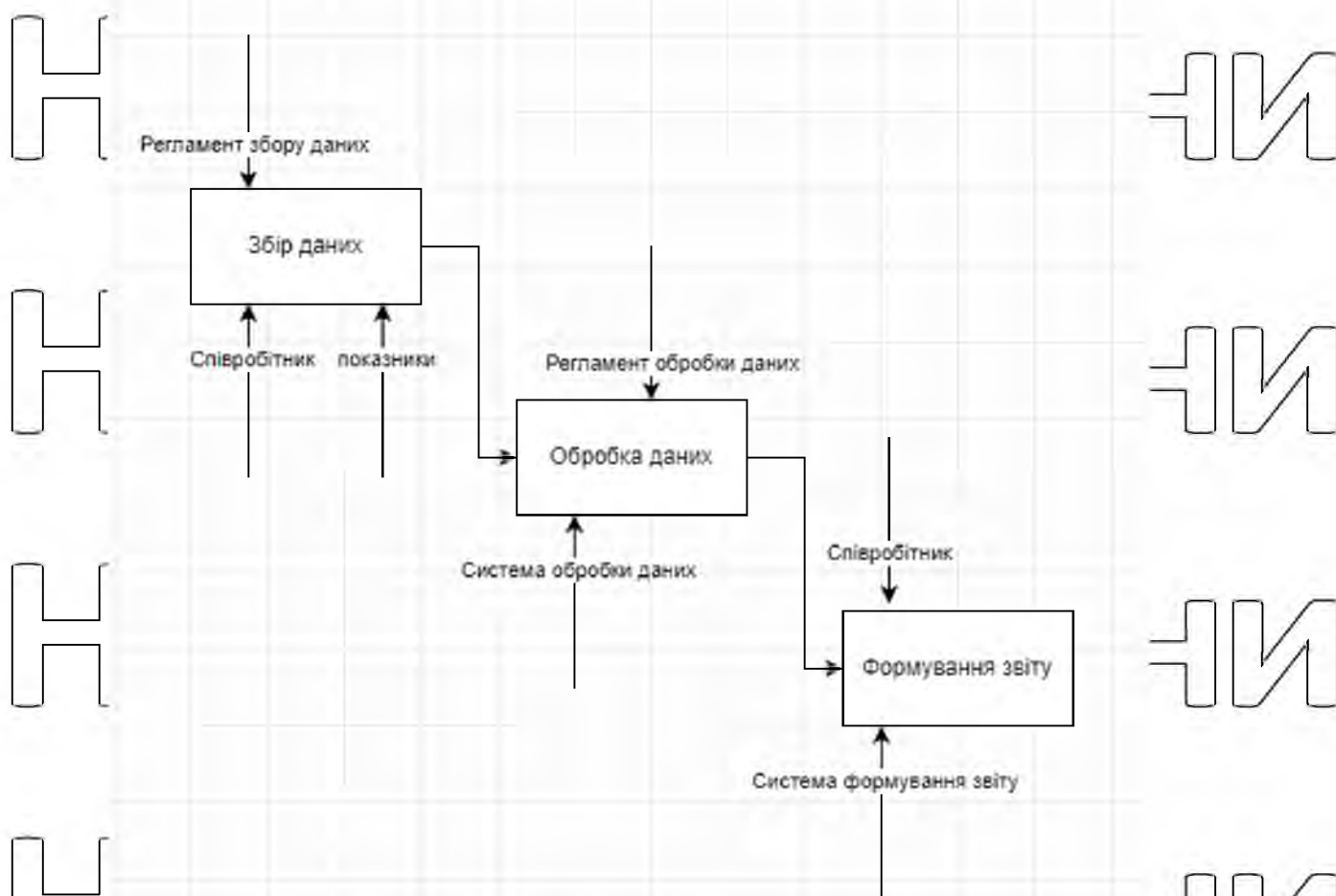


Рис. 2.1 Діаграма IDEF0

•Обробка даних – наступна робота на діаграмі, вхідний потік якої представлений вихідним потоком попереднього блока діяльності, а саме дані показників. Вихідними даними є збережені дані в системі. Керуючий потік – регламент обробки даних, що містить чіткі інструкції щодо обробки отриманих даних працівником за допомогою розроблюваної системи, адже вони являються потоками-механізмами.

•Формування звіту – третя робота представленої діаграми, що на вхід отримує збережені дані в системі, призначені формування звіту на основі вхідних даних. Завдяки звіту можна проаналізувати ефективність працівника.

2.2 Об'єктно-орієнтоване моделювання

2.2.1 Діаграма прецедентів. Уніфікована мова моделювання (Unified Modeling Language, UML) є графічною мовою для візуалізації, специфікації, конструювання та документування систем, в яких велика роль належить програмному забезпеченню. Одним із завдань UML є служити засобом комунікації всередині команди і при спілкуванні з клієнтом.

На рисунку 2.2 представлена діаграма прецедентів предметної області моніторингу стану кліматичних умов регіону. Діаграма складається із таких

компонентів:

- **Співробітник** – актор представляє собою особу, що працює на підприємстві:

- Збір даних;
- Обробка даних;
- Редагування журналу робочого часу;
- Перегляд журналу робочого часу.

- **Адміністратор** – актор представляє особу, що займається адмініструванням системи:

- Перегляд журналу робочого часу;
- Перегляд необхідних даних;
- Редагування записів;
- Перегляд активності співробітника.

- **Стороння система** – представляє систему, до якої мають доступ керівники підприємства для аналізу ефективності роботи співробітників

- Перегляд звітності;
- Внесення змін в робочий процес.

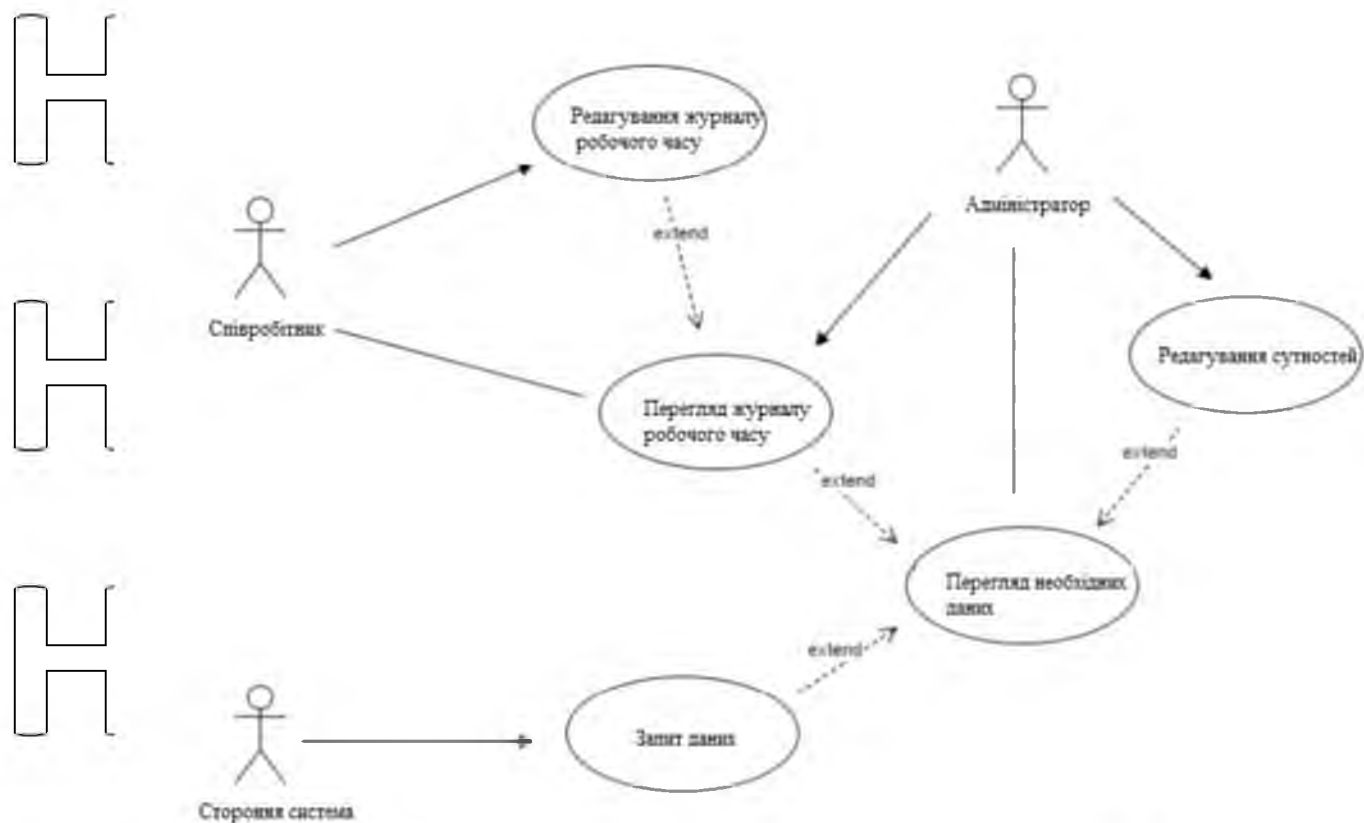


Рис. 2.2/Діаграма предедентів.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

2.2.2. Діаграма активності. Діаграма активності – схема, що фокусується на виконанні та процесі управління системи.

На рисунку 2.4 представлена діаграма активності для розробленої системи.

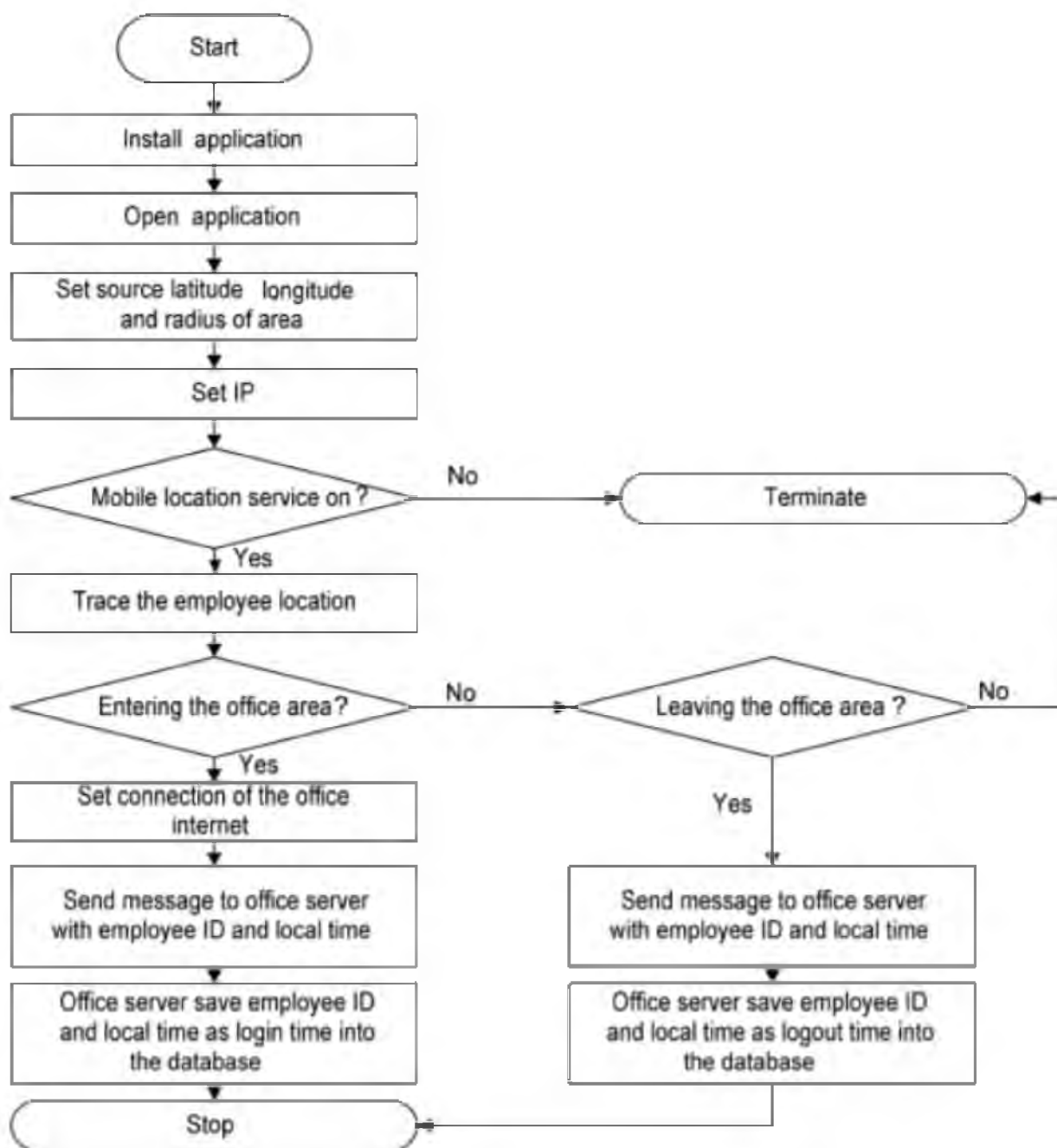


Рис. 2.4/ Діаграма активності системи

Наведена діаграма активності візуалізує порядок робіт під час збору, обробки та аналізу показників внесених співробітниками підприємства.

3 РОЗРОБКА СИСТЕМИ

НУБІП України

3.1 Інформаційне забезпечення

3.1.1 Логічна модель даних.

Логічна модель даних, як правило, складається з сутностей даних, ключів та атрибутів та зв'язків між сутностями. Це спосіб визначення даних організації та правил ведення бізнесу, що регулюють відносини між ними. Взагалі вважається реалізація концептуальної моделі даних.

LDM також допомагає у створенні фізичної моделі даних та надає дорожню карту для проектування фізичної бази даних. Зазвичай макет LDM змінюється, оскільки він має на меті визначити максимально можливі дані, незалежно від способу їх реалізації за допомогою технології.

Логічна модель даних демонструє схематичний набір даних, що присутні в системі, дозволяючи зобразити головні сутності системи та їх зв'язки між собою. Наведена на рисунку 3.1 логічна модель системи обробки та аналізу ефективності роботи працівників підприємства включає в себе:

- Працівник – сутність, призначена зберігати дані про працівника;
- Адміністратор – сутність, призначена містити інформацію про адміністратора;
- Запис – сутність, має містити дані про записи працівників;
- Звіт – сутність, що зберігає дані про звіти сформовані завдяки записам працівників;

НУБІП України

НУБІП України

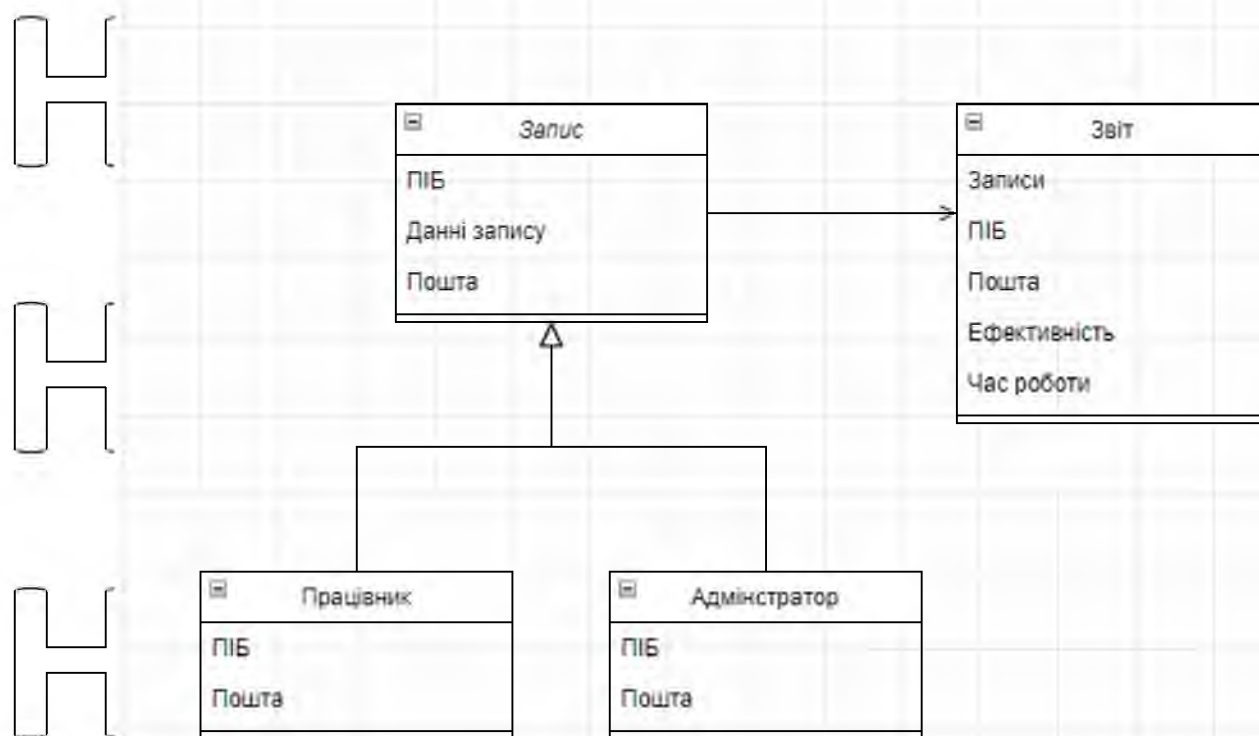


Рис. 3.1 Логічна модель даних

НУБІП України

НУБІП України

НУБІП України

НУБІП України

3.1.2 Вибір інструментів для проектування БД.

Для створення системи було обрано сервіс MongoDB.

MongoDB — документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць.

MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД функціональними і зручними у формуванні запитів.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії

AGPLv3.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій. У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збіг будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу. Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.



Система підтримує ad-hoc-запити: вони можуть повертати конкретні поля документів і призначені для користувача JavaScript-функції. Підтримується пошук за регулярними виразами. Також можна налаштувати запит на повернення випадкового набору результатів.

Є підтримка індексів.

Система може працювати з набором реплік, тобто містити дві або більше копії даних на різних вузлах. Кожен екземпляр набору реплік може в будь-який момент виступати в ролі основної або допоміжної репліки. Усі операції запису і

читання за замовчуванням здійснюються з основною реплікою. Допоміжні

репліки підтримують в актуальному стані копії даних. У разі, коли основна репліка дає збій, набір реплік проводить вибір, яка з реплік має стати основною. Другорядні репліки можуть додатково бути джерелом для операцій читання.

Система масштабується горизонтально, використовуючи техніку сегментування об'єктів баз даних - розподіл їхніх частин по різних вузлах кластера. Адміністратор обирає ключ сегментування, який визначає, за яким критерієм дані будуть рознесені по вузлах (залежно від значень хеша ключа сегментування). Завдяки тому, що кожен вузол кластера може приймати запити,

забезпечується балансування навантаження.

3.2 Програмне забезпечення

3.2.1 Вибір інструментів розробки.

Вибір інструментів для розробки системи досить складний процес, що вимагає багато уваги до деталей та вимог системи. В результаті ознайомлення із можливостями різних підходів до створення веб-додатку було обрано такі технології: JavaScript, Node.js, React.js.

JavaScript (JS) – це легковажна, інтерпретована або JIT-компільована, об'єктно-орієнтована мова з функціями першого класу. Найширше застосування знаходить мову сценаріїв веб-сторінок, але також використовується і в інших програмних продуктах, наприклад, node.js або Apache CouchDB. JavaScript це

прототипно-орієнтована, мультипарадигмна мова з динамічною типізацією, яка підтримує об'єктно-орієнтовану, імперативну та декларативну (наприклад, функціональне програмування) стилі програмування.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

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

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 їхнього виконання. Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

Node.js має наступні властивості:

- асинхронна одно-нитева модель виконання запитів;
- неблокуючий ввід/вивід,

НУБІП України

Для керування модулями використовується пакетний менеджер npm (node package manager).

НУБІП України

React (старі назви: React.js, ReactJS) — відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту вебсторінки, з якими стикаються в розробці

НУБІП України

односторінкових застосунків. Розробляється Meta (раніше Facebook) спільнотою індивідуальних розробників. React дозволяє розробникам створювати великі вебзастосунки, які використовують дані, котрі змінюються з часом, без перезавантаження сторінки

НУБІП України

Його мета полягає в тому, щоб бути швидким, простим, масштабованим. React обробляє тільки користувацький інтерфейс у застосунках. Це відповідає видові у шаблоні модель-вид-контролер (MVC), і може бути використане у поєднанні з іншими JavaScript бібліотеками або в великих фреймворках MVC, таких як AngularJS. Він також може бути використаний з React на основі надбудов, щоб

НУБІП України

підключатися про частини без користувацького інтерфейсу побудови вебзастосунків. Як бібліотеку інтерфейсу користувача React найчастіше використовують разом з іншими бібліотеками, такими як Redux.

НУБІП України

НУБІП України

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

4.1 Розрахунок Ефективності діяльності працівника

Ключові показники ефективності - обмежений набір основних параметрів, які використовуються керівництвом для відстеження і діагностики результатів діяльності підприємства і наступного прийняття на їх основі управлінських рішень. Дані показники повинні відображати всю найбільш важливу інформацію для управління підприємством. Звичайно, для підприємств різних галузей і типів у різний час та за неоднакових умов показники ефективності будуть відрізнятись.

Обсяги реалізації і величина прибутку, рівень рентабельності залежать також від виробничої, постачальницької, збутової і торгівельної діяльності підприємства.

Основними завданнями аналізу доходів від діяльності підприємства є:

- систематичний контроль за виконанням планів реалізації продукції і отриманням прибутку;

- визначення впливу як об'єктивних, так і суб'єктивних факторів на обсяги реалізації продукції і фінансові результати;

- виявлення резервів збільшення обсягів реалізації продукції і суми прибутку;

- оцінка роботи підприємства з використання можливостей збільшення обсягів реалізації продукції, прибутку та рентабельності;

- розробка заходів з використання виявлених резервів

Конверсія – показник ефективності працівника (підприємства), який допомагає організації в досягненні економічних цілей. Використання ключових показників ефективності дає компанії можливість оцінити її поточний стан і допомогти в оцінці успішності реалізації обраної методики.

Всего:3		Конверсия 67%
begdanuse Результат: + Доп.инфо:123 Создана: 09.11.2022 18:34 Последнее изменение: 09.11.2022 18:34	begdanuse Результат: + Доп.инфо:213 Создана: 09.11.2022 18:34 Последнее изменение: 09.11.2022 18:34	
begdanuse Результат: - Доп.инфо:321423 Создана: 09.11.2022 18:34 Последнее изменение: 09.11.2022 18:34		

Рис. 4.2/Розрахунок ефективності працівника

Формула конверсії продажів матиме такий вигляд: $S * 100 / K$.

S - кількість продажів за певний період часу;

K - загальна кількість клієнтів за певний період часу.

В ідеальному варіанті, оцінка ефективності діяльності персоналу повинна служити інструкцією до подальших безпомилковий дій. Це можливо, якщо вона відповідає певним загальним критеріям.

Система KPI не повинна:

- Включати показники, які не можна чітко виміряти
- Включати показники, які не можна співвіднести з прибутком.
- Включати показники, перетворені в KPI без особливого сенсу в цьому.
- Ставати формальністю.
- Заповнюватися «зі стелі».
- Ставати страшним сном для співробітника і системою покарань.
- Становитися єдиною метою роботи кожного співробітника в ущерб некасаючим показателям обов'язностям.

Система KPI повинна:

- Стати мотиватором, що підвищує ефективність роботи співробітників і всіх відділів в загальному.
- Виконувати завдання фінансового і цільового планування, оцінювати

ступінь досягнення конкретних результатів.

- Передбачати тільки метрику, на яку співробітник може вплинути, і відповідно - відповідати за результат.

НУБІП Українни

НУБІП Українни

НУБІП Українни

НУБІП Українни

НУБІП Українни

НУБІП Українни

НУБІП Українни

4.2 Інтерфейс адміністратора

Інтерфейс адміністратора включає в себе режим перегляду записів всіх користувачів чи одного окремо. Адміністратор має можливість змінювати записи, додавати нові від імені іншого користувача. Завдяки адміністратору можна зареєструвати нового користувача. Адміністратор системи має контролювати ефективність працівників.

Все записи

Менеджер

Все

Всього:5

begdanys

Результат: -
Доп.інфо:fzcsfvzc

Создана: 28.09.2022 21:13
Последнее изменение: 28.09.2022 21:13

begdanys

Результат: +
Доп.інфо:gsvksgsv

Создана: 28.09.2022 21:13
Последнее изменение: 28.09.2022 21:13

begdanys

Результат: -
Доп.інфо:sfezfs

Создана: 28.09.2022 21:13
Последнее изменение: 28.09.2022 21:13

begdanys

Результат: -
Доп.інфо:grsefvrgvfc

Создана: 28.09.2022 21:15
Последнее изменение: 28.09.2022 21:15

ref

Результат: +
Доп.інфо:123

Создана: 28.09.2022 21:47
Последнее изменение: 28.09.2022 21:47

ВИСНОВКИ

НУБІП України

В ході виконання дослідження було проведено детальний опис предметної області, проаналізовано наявні рішення та сформульовано головні завдання системи.

НУБІП України

Наступним етапом стало моделювання системи. За допомогою методології SADT було створено функціональну модель на основі контекстної діаграми IDEF0. Об'єктно-орієнтована модель була спроектована мовою UML, а саме діаграмами прецедентів та активності.

НУБІП України

В третьому розділі містяться результати наступного етапу - розробка системи, що складається з умовно двох частин. Перша частина представлена інформаційним забезпеченням, що складається із логічної моделі. Друга частина описує використані технології JavaScript, React.js, Node.js,

НУБІП України

Останній розділ демонструє отримані результати розробки, а саме її розрахунок ефективності працівника.

НУБІП України

В результаті виконання дослідження було створено просту та ефективну систему аналізу діяльності працівників підприємства, завдяки якій керівництво може слідкувати за ефективністю кадрів, та швидше приймати рішення в удосконаленні роботи працівників, для подальшого покращення економічної складової підприємства.

НУБІП України

НУБІП України

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Персонал підприємства та його класифікація

<https://buklib.net/books/24836/>

2. "Персонал підприємства: класифікація, структура

https://osvita.ua/vnz/reports/econom_pidpr/20538/

3. Випуск 1. Професії

працівників, <https://profpressa.com/dkhp/vipusk-1-profesivi-pratsivnikiv-shcho-ie-zagalnimi-dlia-vsikh-vidiv-ekonomichnovi-diiialnosti>

4. UML для бизнес-моделирования: зачем нужны диаграммы процессов. [Електронний ресурс] Режим доступу:

<https://evergreens.com.ua/ru/articles/uml-diagrams.html>

5. Диаграмма кооперации та правила її побудови. Активні і пасивні об'єкти. Зв'язки в кооперации і їх позначення. [Електронний ресурс] Режим доступу:

<http://um.co.ua/9/9-2/9-29951.html>

6. Диаграмма активности UML. [Електронний ресурс] Режим доступу:

<https://coderlessons.com/tutorials/kompiuterno-programmirovanie/uchebnik-uml/9-diagramma-aktivnosti-uml>

7. Ключевые аспекты при выборе базы данных для вашего приложения. [Електронний ресурс] Режим доступу:

<https://habr.com/en/company/otus/blog/562852/>

8. JavaScript. [Електронний ресурс] Режим доступу:

<https://developer.mozilla.org/ru/docs/Web/JavaScript>

9. MongoDB. <https://en.wikipedia.org/wiki/MongoDB>

10. Что такое Node JS. Начало работы. [Електронний ресурс] Режим доступу:

<https://metanit.com/web/nodejs/1.1.php>

11. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ОЛАП СИСТЕМ. [Електронний ресурс] Режим доступу:

<https://nsb.ihs2.ru/articles/osnovnye-kharakteristiki-olap-sistem/>

12. SSAS Tutorial: What is SSAS Cube, Architecture & Types. [Електронний ресурс] Режим доступу: <https://www.guru99.com/ssas-tutorial.html>

13. SQL Server Integration Services. [Електронний ресурс] Режим доступу: <https://docs.microsoft.com/en-us/sql/integration-services/sql-server-integration-services?view=sql-server-ver15>

14. What is SSRS? [Електронний ресурс] Режим доступу: <https://www.guru99.com/ssrs-tutorial.html>

15. Инструменты data mining. [Електронний ресурс] Режим доступу: https://studref.com/400367/menedzhment/instrumenty_data_mining

16. Decision Tree – Classification. [Електронний ресурс] Режим доступу: https://www.saeedshah.com/decision_tree.htm

17. Алгоритм наивного байесового классификатора [Електронний ресурс] Режим доступу: <https://www.javatpoint.com/machine-learning-naive-bayes-classifier>

18. Асоціаційні правила. Послідовне відображення шаблонів даних. [Електронний ресурс] Режим доступу: <https://infopedia.su/7x8af3.html>

19. Different Types of Clustering Methods and Applications [Електронний ресурс] Режим доступу: <https://www.analytixlabs.co.in/blog/types-of-clustering-algorithms>

20. Key Performance Indicators URL: <https://www.it.nu/ru/knowledge-base/technology-innovation/key-performance-indicators-kpi>

НУБІП України

НУБІП України

НУБІП ДОДАТОК А України

Код системи

```
import { useDispatch } from 'react-redux';
import { Formik } from 'formik';
import * as yup from 'yup';

import { loginUserAsyncActions } from '../../redux/actions/auth-async-actions';
import { Box, Button, FormControl, InputAdornment, TextField, Typography } from
 '@mui/material';
import { AccountCircle } from '@mui/icons-material';
import LockOpenTwoToneIcon from '@mui/icons-material/LockOpenTwoTone';
```

```
const emailRegexp = /^\\w+([.-]?\\w+)*@\\w+([.-]?\\w+)*\\.\\w{2,3}+$/;
const initialState = { email: '', password: '' };
```

```
const validationSchema = yup.object().shape({
  email: yup.string().matches(emailRegexp, 'Неверный
 формат!').required('Обязательное поле!'),
  password: yup
```

```
  .string()
  .length()
  .test('length', 'Минимум 8 символов!', val => val && val.length >= 8)
  .required('Обязательное поле!'),
});
```

```
export default function Auth() {
  const dispatch = useDispatch();
```

```
  return (
    <Box
      display={'flex'}
      flexDirection={'column'}
      sx={{ margin: '30px auto 0 auto', width: '300px' }}
    >
```

```
      <Formik
        initialValues={initialState}
        validateOnBlur
        validationSchema={validationSchema}
        onSubmit={values => dispatch(loginUserAsyncActions(values))}
      >
```

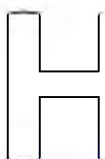
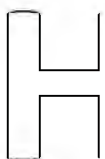
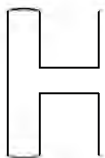
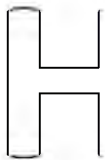
```
        {{{
          values,
          errors,
          touched,
          handleChange,
          handleBlur,
```



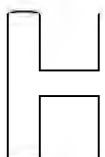
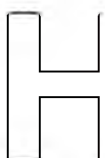
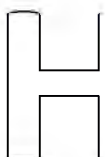
```

handleSubmit,
dirty,
isValid,
resetForm,
}) => (
  <>
    <FormControl
      sx={{ mb: '15px', position: 'relative' }}>
      <TextField
        type={'text'}
        name={'email'}
        id={'email'}
        label={'Почта'}
        value={values.email}
        onChange={handleChange}
        onBlur={handleBlur}
        variant="standard"
        InputProps={{
          startAdornment: (
            <InputAdornment position="start">
              <AccountCircle />
            </InputAdornment>
          ),
        }}
      />
      {touched.email && errors.email && (
        <Typography
          sx={{ position: 'absolute', top: 50, left: 50, fontSize: '15px'
            color="error"
          >
            {errors.email}
          </Typography>
        )}
      </FormControl>
      <FormControl sx={{ position: 'relative' }}>
      <TextField
        type={'text'}
        name={'password'}
        id={'password'}
        label={'Пароль'}
        value={values.password}
        onChange={handleChange}
        onBlur={handleBlur}
        variant="standard"
        InputProps={{
          startAdornment: (
            <InputAdornment position="start">
              <LockOpenTwoToneIcon />
            </InputAdornment>
          ),
        }}
      />

```



}}



```

    }}
  />
  {touched.password && errors.password && (
    <Typography
      sx={{ position: 'absolute', top: 50, left: 50, fontSize: '15px'
        color="error"
      >
        {errors.password}
    </Typography>
  )}
</FormControl>
<Button
  type={'submit'}
  onClick={() => {
    handleSubmit();
  }}
  sx={{ mt: '30px' }}
  variant="contained"
>
  Войти
</Button>
</>
  )}
</Formik>
</Box>
  );
}

```

```

import SaveIcon from '@mui/icons-material/Save';
import { Formik } from 'formik';
import * as yup from 'yup';
import { useSelector } from 'react-redux';
import { getManagersList } from '../../redux/selectors/ops-selectors';
import {
  Box,
  Button,
  FormControl,
  FormControlLabel,
  FormLabel,
  InputLabel,
  MenuItem,
  Radio,
  RadioGroup,
  Select,
  TextField,
  Typography,
} from '@mui/material';
import { getLogin, getUserRole } from '../../redux/selectors/auth-selectors';

```

```

export default function AddForm({ submitForm: add }) {
  const role = useSelector(getUserRole);
  const managers = useSelector(getManagersList);
  const user = useSelector(getLogin)

  const validationSchema = yup.object().shape({
    additional: yup.string().required('Обязательное поле!'),
    result: yup.string().required('Выберите ответ!'),
    owner: yup.string().required('Выберите Менеджера!'),
  });

  const validationScheme = yup.object().shape({
    additional: yup.string().required('Обязательное поле!'),
    result: yup.string().required('Выберите ответ!'),
    owner: yup.string()
  });

  const schema = role === 'admin'? validationSchema: validationScheme

  return (
    <Box
      sx={{
        padding: '10px 10px 20px 10px ',
        marginBottom: '40px',
        borderBottom: '2px solid #5cc465',
      }}
    >
      <Formik
        initialValues={{ additional: '', result: '', owner: role === 'admin'? '':
'user' }}
        validateOnBlur
        validationSchema={schema}

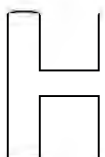
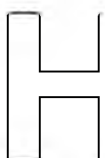
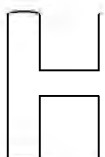
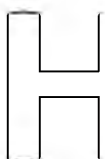
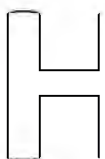
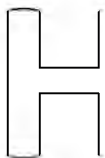
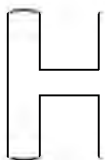
        onSubmit={(values, { resetForm }) => {
          add(values);
          console.log('suka')
          resetForm();
        }}
      >
        {{{ values, errors, touched, handleChange, handleBlur, handleSubmit,dirty,
isValid }} => (
          <Box display={'flex'}>
            <Box display={'flex'} flexDirection={'column'}>
              <FormControl sx={{ position: 'relative' }}>
                <FormLabel id="result">Результат</FormLabel>
                <RadioGroup
                  aria-labelledby="result"

```

```

    row
    defaultValue=""
    name="result"
    onChange={handleChange}
    onBlur={handleBlur}
  >
    <FormControlLabel value="+" control={<Radio />} label="Плюс" />
    <FormControlLabel value="-" control={<Radio />} label="Минус" />
  </RadioGroup>
  {touched.result && errors.result && (
    <Typography
      sx={{
        position: 'absolute',
        left: 58,
        top: 19,
        fontSize: '11px',
      }}
      color="error"
    >
      {errors.result}
    </Typography>
  )}
</FormControl>
<FormControl sx={{ position: 'relative' }}>
  <TextField
    label="Дополнительная информация"
    variant="outlined"
    name="additional"
    id="additional"
    multiline
    value={values.additional}
    onChange={handleChange}
    onBlur={handleBlur}
  >></TextField>
  {touched.additional && errors.additional && (
    <Typography
      sx={{
        position: 'absolute',
        left: 17,
        top: 57,
        fontSize: '11px',
      }}
      color="error"
    >
      {errors.additional}
    </Typography>
  )}
</FormControl>
</Box>
<Box
  marginLeft={'10px'}

```



```

display={'flex'}
flexDirection={'column'}
justifyContent={'space-between'}
>
{role === 'admin' && (
  <>
    <FormControl sx={{ position: 'relative', mt: '15px' }}>
      <InputLabel id="select-label">Имя</InputLabel>
      <Select
        labelId="select-label"
        name="owner"
        value={values.owner}
        id="owner"
        label="Имя"
        sx={{ width: '150px', height: '40px' }}
        onChange={handleChange}
      >
        {managers.length &&
          managers.map(manager => (
            <MenuItem key={manager._id} value={manager._id}>
              {manager.login}
            </MenuItem>
          ))}
      </Select>
      {touched.owner && errors.owner && (
        <Typography
          sx={{
            position: 'absolute',
            left: 15,
            top: 42,
            fontSize: '11px',
          }}
          color="error"
        >
          {errors.owner}
        </Typography>
      )}{ ' ' }
    </FormControl>
  </>
)}
<Button
  id='button'
  variant="contained"
  endIcon={<SaveIcon />}
  type="submit"
  disabled={!isValid && !dirty}
  onClick={handleSubmit}
>
  Добавить
</Button>
</Box>

```

```

    </Box>
  )}
</Formik>
</Box>
);
}

```

```

import { useDispatch } from 'react-redux';
import AddForm from '../..//components/AddForm';
import ManagersList from '../..//components/ManagersList/ManagersList';
import { addNoteAsyncActions } from '../..//redux/actions/notes-async-actions';
import { Box } from '@mui/system';
import RegisterForm from '../..//components/RegisterForm/';

```

```

export default function AdminPage() {
  const dispatch = useDispatch();

  return (
    <>
      <Box
        display={'flex'}
        flexDirection={'column'}
        // flexBasis={'calc((100% - 40px) / 2);'} flex={'no-wrap'}
      >
        <AddForm
          submitForm={data => {
            dispatch(addNoteAsyncActions(data));
          }}
        />
        <RegisterForm />
        <Box width={'100%'}>
          <ManagersList />
        </Box>
      </Box>
    </>
  );
}

```

```

{
  "name": "managers",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@emotion/react": "^11.8.2",
    "@emotion/styled": "^11.8.1",
    "@material-ui/core": "^4.12.4",
    "@mui/icons-material": "^5.5.1",
    "@mui/material": "^5.8.1",
    "@mui/x-date-pickers": "^5.0.0-alpha.6",
    "@redux-devtools/extension": "^3.2.2",
    "@testing-library/jest-dom": "^5.16.2",

```

```

    "@testing-library/react": "^12.1.4",
    "@testing-library/user-event": "^13.5.0",
    "axios": "^0.26.1",
    "date-fns": "^2.28.0",
    "firebase": "^9.6.10",
    "formik": "^2.2.9",
    "moment": "^2.29.1",
    "notiflix": "^3.2.5",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-month-picker": "^1.3.11",
    "react-redux": "^7.2.6",
    "react-router-dom": "^5.3.0",
    "react-scripts": "5.0.0",
    "redux": "^4.1.2",
    "redux-persist": "^6.0.0",
    "redux-thunk": "^2.4.1",
    "web-vitals": "^2.1.4",
    "yup": "^0.32.11"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  },
  "devDependencies": {
    "@redux-devtools/core": "^3.11.0",
    "redux-devtools-extension": "^2.13.9"
  }
}

```

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України