МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної і програмної інженерії
Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри
_____ Литвиненко О.Є.

"____"_____2022 р.

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСНИКА ОСВІТНЬОГО СТУПЕНЯ "БАКАЛАВР"

Тема: "Програмний модуль організації стартап-компанії"

Виконавець: Крюкова Л.Г.

Керівник: к.т.н., доц. Марченко Н.Б.

Нормоконтролер: к.т.н., доц. Тупота Є.В.

Київ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної і програмної інженерії
Кафедра комп'ютеризованих систем управління
Освітній ступінь бакалавр
Спеціальність 126 "Інформаційні системи та технології"
Освітньо-професійна програма "Інформаційні системи та технології"

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Литвиненко О.Є.

"___" ____2022 р

ЗАВДАННЯ

на виконання дипломної роботи

Крюкової Лілії Григорівни

1. Тема дипломної роботи: "Програмний модуль організації стартап-компанії" затверджена наказом ректора від «15» лютого 2022 р. № 251/ст.

2. Термін виконання роботи: з 16.05.2022 р. до 19.06.2022 р.

3. Вихідні данні до роботи: програмний продукт, розроблений за допомогою .NET технологій: мови програмування C#, платформи .NET 4.7, Entity Framework Core, SendGrid, Razor.

4. Зміст пояснювальної записки:
   1) Аналіз доменної області стартапів.
   2) Аналіз вимог та проектування модуля.
   3) Програмний модуль організації стартап-компанії.

5. Перелік обов'язкових слайдів презентації:
   1) Діаграми прецедентів.
   2) Опис прототипу.
   3) Схема БД.
   4) Діаграма класів.

6. Календарний план-графік

№ пор	Завдання	Термін виконання	Відмітка про виконання
1.	Ознайомлення з постановкою задачі та вивчення інформаційних матеріалів.	16.05.2022 – 17.05.2022	
2.	Складання графіку дипломного проектування та затвердження його керівником. Написання 1 розділу, представлення керівнику.	17.05.2022 – 21.05.2022	
3.	Написання 2 розділу; представлення керівнику	01.06.2022 – 01.06.2022	
4.	Написання 3 розділу; представлення керівнику	02.06.2022 – 07.06.2022	
5.	Друк ПЗ, отримання відгуку керівника, проходження нормконтролю	08.06.2022 – 10.06.2022	
6.	Розробка тексту доповіді. Оформлення графічного матеріалу для презентації. Отримання відгуку керівника. Перед захист проекту.	10.06.2022 – 14.06.2022	
7.	Отримання рецензії, підготовка презентації, здача комплекту документів (ПЗ, презентації, CD-R, рецензії, відгуку керівника, довідку об успішності) секретарю ДЕК	13.06.2022 – 14.06.2022	
8.	Захист дипломного проекту	14.06.2022 - 18.06.2022	

Дата видачі завдання 16.05.2022 р.

Керівник дипломної роботи: _____ Марченко Н.Б.

Завдання прийняв до виконання: _____ Крюкова Л.Г.

РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Програмний модуль організації стартап-компанії»: 58 с., 18 рис., 13 інформаційних джерел.

СТАРТАП, УПРАВЛІННЯ БІЗНЕС ПРОЦЕСАМИ, ІТ-КОМПАНІЯ, ВЕБ-ЗАСТОСУНОК, МЕНЕДЖЕР ПРОЕКТУ.

Об’єкт розробки – стартапи та процеси, пов’язані з ними.

Предмет – автоматизація процесів організації стартап-компанії.

Мета роботи – розробити програмний модуль організації стартап-компанії, що призведе до збільшення ефективності управління бізнес-процесами в стартап-компанії.

ПЕРЕЛІК ПРИЙНЯТИХ СКОРОЧЕНЬ

BPM – Business process management

IT – Інформаційні технології

AI – Artificial intelligence

HR – Human resources

ЗМІСТ

ВСТУП

Не має значення, яким бізнесом ви займаєтесь, регулярне підвищення ефективності процесів має важливе значення для успіху. Ігнорування управління бізнес-процесами – це певний спосіб стати стагнацією або, що ще гірше, відставати від конкурентів.

Сьогодні багато компаній використовують традиційний підхід до управління бізнес-процесами. Непослідовності, лазівки та людські помилки рясніють традиційним підходом ручки та паперу до управління бізнес-процесами. Програмне забезпечення BPM дозволяє вам краще контролювати типові операції вашої компанії. Вони можуть прискорити обробку речей, покращити спілкування та краще зрозуміти, як процес виконується в реальному часі.

Додаток для управління бізнес-процесами (BPM) – це технологія, яка допомагає компаніям керувати, автоматизувати та покращувати свої регулярні бізнес-операції. Конфігурована цифрова форма збирає вхідні дані для процесу, а автоматизований конвеєр діяльності обробляє інформацію в більшості програмного забезпечення для управління бізнес-процесами.

РОЗІДЛ 1
АНАЛІЗ ДОМЕННОЇ ОБЛАСТІ СТАРТАПІВ

1.1. Історія розвитку стартапів

За останні двадцять років ІТ-галузь стала рушійною силою конкурентоспроможності світової економіки. Розвиток інформаційних технологій змінив способи та методи управління, що призвело до переосмислення підходів до управління, набуття знань, інновацій, створення нових бізнес-моделей для компаній тощо. Разом з розвитком інформаційних технологій інтенсивно розвиваються стартапи, які гнучко реагують на потреби ринку та пропонують конкретні рішення для своєї цільової групи.

Стартап – це щойно створена компанія (може ще офіційно не зареєстрована, але планує), яка будує свій бізнес на інноваціях чи інноваційних технологіях, ще не вийшла на ринок чи розпочала, має обмежені ресурси. Стартапи часто називають «гаражними» компаніями.

Термін «стартап» найчастіше використовується для позначення інтернет-компаній та інших ІТ-компаній, але цей термін поширюється і на інші сфери бізнесу.

Інновації, на яких стартапи будують свій бізнес, можуть бути як глобальними (тобто всесвітні інновації), так і локальними (тобто інновації в одній країні, але в інших країнах ця технологія вже не є інноваційною).

Служба рейтингу стартапів на основі аналізу кількості стартапів у 137 країнах розробила рейтинг, в якому Україна посіла 42 місце (215 стартапів) у 2018 році, випередивши Литву та Естонію. Перше місце в рейтингу посідають США з 45 004 стартапами, друге – Індія з 5 203 стартапами, а третє – Великобританія з 4702 стартапами. (https://www.starturanking.com). Веб-сайт використовує оцінку SR для оцінки самих стартапів. SR Score — це число від 0 до 100 000, що розраховується як вбудований індикатор, що відображає важливість стартапу в Інтернеті та його вплив на соціальні мережі. SendPulse має найвищий рейтинг серед українських стартапів – 238

позицій із 83 313 результатів SR, також до першої 1000 увійшли ще 4 стартапи. У рейтингу 2017 року – Глобальний інноваційний індекс країн з найбільш інноваційною економікою – Україна посідає 42 місце, на одне місце нижче, ніж у попередньому році. Згідно з рейтингом, вартість коштів, виражена у відсотках від ВВП, які Україна витрачає на дослідження та розробки, має коефіцієнт 44; ККД 50; наявність високотехнологічних компаній в економіці 34; кількість поданих патентів та кількість дослідників - 27.

Людський капітал є двигуном інноваційної конкурентоспроможності України. Його ефективна реалізація є запорукою отримання конкурентної переваги.

Рис. 1.1. Сильні та слабкі сторони України у Глобальному інноваційному індексі

Стартапи, інновації та бізнес живуть своїм життям! Їм не потрібна допомога, їх не треба турбувати! Регіони, муніципалітети та університети повинні знайти своє місце в цьому процесі. Наприклад, згідно з щорічним галузевим звітом групи інвестиційних фірм, опублікованим у виданні The DealBook of Ukraine за 2018 рік: https://www.slideshare.net/YevgenSysoyev/the-

dealbook-of-ukraine-2018-edition/ 1 повністю відображають інвестиційні угоди 2016 та 2017 років в українських стартапах, розподіл інвестицій залежно від сфери діяльності та стадії розвитку стартапу. Зокрема, у дослідженні зазначається, що після рецесії 2016 року ми спостерігаємо дуже швидке зростання цифрових ринків та інновацій. Українські стартапи (або стартапи, створені українськими підприємцями) залучили $265 млн інвестицій у 2017 році, що на 231% більше, ніж у річному обчисленні. У 2017 році на різних етапах розвитку стартапу було здійснено 44 нові інвестиції. В мережі Інтернет постійно з'являється інформація про появу нових стартапів чи інноваційних рішень в Україні. Наприклад, навесні 2018 року на сайті KfundMedia була опублікована серія статей про стартапи, які створюються в різних регіонах України. Цікавий досвід Вадим Роговський і два його стартапи Clickky і 3DLOOK. Платформа для просування мобільних додатків Clickky увійшла до двадцятки найбільш швидкозростаючих європейських маркетингових компаній у 2017 році згідно з виданням профілю Inc. У 2015 році Роговський залучив 2 мільйони доларів від інвестиційної компанії iTech Capital. Каліфорнійський акселератор Boost.vc вже вклав 500 тисяч доларів у новий проект 3DLOOK. Є й інші стартапи, які успішно розвиваються у Вінниці, Києві, Дніпрі, Львові, Одесі та інших містах, успішно збираючи кошти на свій розвиток.

1.2. Організаційні форми створення стартапів

Сучасний ринок характеризується складністю та нестабільністю, що змушує компанії та підприємців різних сфер постійно шукати нові можливості та застосовувати нові підходи до ведення бізнесу. Про стартапи ви чуєте все більше з різних інформаційних мереж. Вони поєднують функції, які дозволяють компаніям оптимально функціонувати в складних умовах сучасного ринку.

У вітчизняному законодавстві відсутнє визначення терміну «стартап», яке регулює їх діяльність шляхом застосування законів і нормативних актів відповідно до специфіки діяльності таких компаній. Чинне законодавство

включає 14 нормативно-правових актів, понад 50 нормативно-правових актів уряду, близько 100 різних міністерських документів, що стосуються інновацій [1].

У науковій літературі існує багато визначень терміну «стартап». Вперше ця концепція з'явилася в Америці в 1930 році. У перекладі з англійської «start up» означає «початок» і означає нещодавно створену або все ще розвивається компанію. Вчені трактують цей термін як: новостворена організація, яка розробляє нові товари чи послуги в умовах крайньої невизначеності [2]; процес виходу на ринок стартапів з інноваційним проектом, як правило, короткостроковий і з мінімальними інвестиціями [3]; нова компанія на ранній стадії розвитку, створена для реалізації перспективної ідеї для отримання високих прибутків [4]; тимчасова структура, яка шукає масштабну, повторювану та прибуткову бізнес-модель [5]; компанія з короткою історією діяльності [6]; компанії-початківці, які перебувають у фазі розвитку та будують свій бізнес на основі нових інноваційних ідей або на основі нещодавно з'являються технологій [7]; компанія щойно створена, має прототипи, намагається організувати виробництво та вийти на ринок [8].

Слід зазначити, що в глобальній економіці місце успішних стартапів досить високе і почесне. Так, на ринках Європи, США, Японії, Росії і навіть Білорусі активно створюються венчурні фонди та «інкубатори», метою яких є фінансування та підтримка сміливих проектів капіталізації доходів [9].

Один із наших сусідів, Білорусь, також не має юридичного визначення терміну «стартап», але держава допомагає організовувати стартап-заходи, які включають навчання основам підприємництва, пошук та залучення потенційних інвесторів. Є також варіанти державної підтримки на законодавчому рівні шляхом використання інноваційних грантів для фінансування нових проектів приватних осіб або малого бізнесу [10].

Сполучені Штати вважаються лідером серед країн, де можна створити успішні стартапи. Це тому, що в Сполучених Штатах є більше відомих і

надійних каналів, за якими ви можете знайти фінансову допомогу, включно з тими, які можуть допомогти вам розпочати свою подорож. На правовому рівні влада підтримує молодих підприємців. Починаючи проект в Америці, шанси на те, що він швидше вийде на міжнародний ринок, вище.

Що стосується Азії та Європи, то вона наздоганяє Сполучені Штати, коли йдеться про динамічно розвиваються стартапи. Тут багато державних установ мають чіткі програми підтримки та побудови інноваційної системи на національному рівні, тоді як Сполучені Штати продовжують довіряти цю справу виключно приватному сектору.

Щоб краще зрозуміти, що таке стартап, потрібно окреслити сферу його функцій і зрозуміти, чим стартап відрізняється від звичайного підприємництва.

Тому основними рисами стартапу є: новизна та унікальність ідеї, адаптивність, коротка історія функціонування, мінімальна вартість авторських ресурсів, висока швидкість розробки проекту та найбільш ефективне просування компанії на ринку, здатність стати бізнес-моделлю для великого ринку, довіра до проекту та його гнучкість по відношенню до потреб ринку [12], короткий час початку проекту, адаптивність, фінансові проблеми [11].

Що стосується відмінностей між стартапом і підприємцем, то наступні:

1. У більшості випадків молодь (потенційні носії інноваційних ідей) залучається до розробки та інших видів діяльності сортапу;

2. Стартап базує свою діяльність на інноваціях та інноваційних технологіях, найчастіше це сфера IT та сфера послуг (хоча стартап можна використовувати у всіх сферах, основна ідея);

3. Стартапи не мають власного початкового капіталу. Створюється переважно інвесторами у вигляді вкладень у цей проект (інвестицій);

4. Стартап постійно змінюється в процесі створення, і зміни можуть вплинути на суть проекту. Це неминуче призводить до деяких проблем і суперечок, особливо коли автором і розробником проекту є різні люди;

5. Оскільки стартап розуміє інновації та неперевірену технологію чи послугу, такі проекти більш схильні до провалу, ніж звичайні (традиційні) компанії [13].

Юридична функція стартапу — вести бізнес у конкретній організаційно-правовій формі. Слід зазначити, що в Україні існує багато організаційно-правових форм господарської діяльності, але не всі вони є актуальними для реєстрації стартапів. Оскільки стартап зазвичай об'єднує невелику групу зацікавлених у реалізації спільної ідеї, то як перший варіант його правової структури можна запропонувати наступні форми:

- фізична особа-підприємець (ФОП);

- приватна компанія (ПП);

- Товариство з обмеженою відповідальністю (ТОВ) [14]. Кожна з цих форм має свої переваги і недоліки.

Для фізичного підприємця абсолютною перевагою є те, що стартап працює самостійно, не має зобов'язань створювати цільовий капітал і не зобов'язаний документувати будь-які свої рішення. Стартап співпрацює з підрядниками на основі договору і не зобов'язаний включати їх до штату. Однак істотним недоліком такого підприємця є те, що він обмежений власними ресурсами і якщо проект не принесе очікуваних результатів, стартап буде відповідати за борг усіма своїми активами.

При створенні стартапу у формі приватної компанії перевагою є те, що розмір статутного капіталу не визначений законом, а також можливість налагодження функціонування ЄП з дотриманням усіх відповідних положень статуту. Щодо недоліків цієї форми, то термін «приватне підприємство» лише вказує на форму власності та потребує юридичного уточнення. Цей тип компаній менш впізнаваний для іноземних інвесторів.

Перевага товариства з обмеженою відповідальністю полягає в тому, що воно несе повну відповідальність за наслідки своєї діяльності майном такого товариства. Розмір статутного капіталу може бути будь-яким від однієї копійки до мільйонів гривень. При цьому закон не обмежує право вибору

засновника. Крім того, статутний капітал установчого товариства може бути встановлений у строк, визначений самими засновниками. Ще одна перевага – спрощена система адміністрування компанії. Слід також зазначити, що для іноземних інвесторів ведення стартапу як юридичної особи є більш надійним, ніж просто підприємець.

Незважаючи на всі переваги, є і недоліки, а саме, якщо один з учасників діє всупереч інтересам суспільства, його буде важко вилучити з класу за рішенням збору учасників: він повинен бути готовий піти.

1.6. Програмний інструментарій для організації стартап-компанії

Жодна програма не існує сама по собі. Його функціональні та архітектурні особливості безпосередньо пов'язані з середовищем його використання. Тому знання систем управління проектами є неповними без попереднього обговорення типу середовища, в якому ці програми «функціонують». Середовище управління Кожну організацію можна розділити на три групи працівників, які беруть участь у процесі управління її діяльністю.

1. Вище керівництво, спеціалісти, відповідальні за постановку цілей і завдань, закріплення організаційних планів та оцінку виконання цих планів.

2. Менеджери, відповідальні за розробку детальних планів досягнення цілей, поставлених вищим керівництвом; Розподіл роботи між окремими підрядниками, планування використання ресурсів, контроль виконання планів та створення зведених звітів для вищого керівництва.

3. Фахівці галузі, які відповідають за виконання конкретної роботи згідно з графіком та складання звітів про стан виконаної роботи, її якість, наявність, використання ресурсів тощо.

Істотні відмінності виконуваних завдань визначають відмінності в вимогах цих груп користувачів до програмного забезпечення, спрямованого на підвищення ефективності їх діяльності. Welcom Software Technologies свого часу провела дослідження, яке виявило відмінності в вимогах до

програмного забезпечення на різних рівнях управління проектами (таблиця). У результаті було визначено три рівні влади: виконавчий, стратегічний та робочий. Причому до останнього в даному випадку увійшли як підрядники, так і польові працівники, які використовують програмне забезпечення для управління проектами не більше кількох годин на місяць.

В IT управління проектами може виконуватися відповідно до трьох життєвих циклів проекту:

1. Я передбачав, що це водоспад. Традиційний підхід використовується на порядок більше інших навіть у 2010 році. Покроковий лінійний алгоритм.

2. Ітеративний. Сучасний підхід, при якому функціональність програмного забезпечення розширюється з кожною новою версією в рамках проекту.

3. Адаптивний. Agile, Scrum та інші методи. Цілі та стратегія розвитку компанії можуть змінюватися незалежно від початкового плану методу PMI / PMBOK. Ініціювання, планування, реалізація, контроль та виконання. Інструкції, а не метод по суті.

Водоспад - модель водоспаду. Відповідно до моделі водоспаду, розробник строго послідовно переходить від одного етапу до наступного. По-перше, етап визначення вимог завершено, в результаті чого буде створено список вимог до програмного забезпечення.

Після повного визначення вимог відбувається перехід до проектування, де створюються документи, які детально описують програмістам, як і як реалізувати ці вимоги. Після завершення проекту розробники реалізують отриманий проект. Наступним кроком процесу є інтеграція окремих компонентів, розроблених різними командами розробників. Після завершення впровадження та інтеграції продукт тестується та налагоджується; На цьому етапі усуваються всі недоліки, що виникли на попередніх етапах розробки. Потім програмне забезпечення розгортається та підтримується – вводяться нові функції та виправляються помилки.

Scrum — це «структурний» підхід. Над кожним проектом працює ціла команда спеціалістів і ще двоє людей: власник продукту та скрам-майстер. Перший з'єднує команду з клієнтом і контролює розвиток проекту; Він не офіційний керівник групи, а куратор. Другий допомагає першому організувати бізнес-процес: проводить загальні збори, вирішує бюджетні проблеми, мотивує команду та контролює дотримання підходу Scrum. Підхід Scrum розбиває робочий процес на етапи спринту - зазвичай щотижневі або місячні періоди, залежно від проекту та команди. Перед спринтом формулюються завдання на спринт, в кінці обговорюються результати і команда починає новий спринт. Спринти дуже легко порівнювати один з одним, що дозволяє контролювати ефективність.

Канбан — це «балансовий» підхід. Його завдання — збалансувати різних спеціалістів у команді та уникнути ситуацій, коли дизайнери працюють цілодобово, а програмісти скаржаться на відсутність нових завдань. Вся команда єдина — у Kanban немає ролей Product Owner і Scrum Master. Бізнес-процес ділиться не на універсальні спринти, а на етапі виконання конкретних завдань: «Заплановано», «У розробці», «Перевірено», «Виконано», «незаплановано».

Основним показником ефективності Kanban є середній час, необхідний для подання завдання на дошці. Завдання виконали швидко – команда працювала плідно та злагоджено. Завдання затягнулося – треба подумати, на якому етапі і чому виникли затримки і чию роботу оптимізувати. Дошки використовуються для візуалізації agile-практик: фізичних та електронних. Робіть робочий процес відкритим і зрозумілим для всіх професіоналів, що важливо, коли в команді немає офіційного керівника

Kanban був розроблений в 1953 році інженерною компанією Toyota Taiichi Ono і дуже схожий на схему промислового виробництва. На вході в цей процес знаходиться шматок металу, і в результаті виходить готова деталь. Також у Kanban інкремент продукту вперше переноситься з етапу на етап і, нарешті, готовий до включення елемента.

Крім того, творця Kanban надихнули супермаркети, а точніше їхні принципи — «Тільки те, що потрібно вашим клієнтам, тримайте на полиці». Завдяки тому, що Kanban дозволяє залишити незавершене завдання на етапі зміни пріоритету та інших невідкладних завдань. Невідредагована публікація в блозі, узагальнена без дати публікації або деяких функцій коду, які можуть бути недоступні в продукті, є нормальною для Kanban.

Канбан менш суворий для нас, він не отримав Scrum — він не обмежує час спринту, немає ролей, які б відключали власників продуктів. Завдяки Kanban один член команди може навіть виконувати багатозадачність, чого не дозволяє Scrum. Також жодним чином не регулюються зустрічі щодо статусу проекту – ви можете робити це на власний розсуд або не можете робити повідомлення.

Щоб працювати з Kanban, вам потрібно визначити етапи робочого процесу. У Kanban вони представлені у вигляді таблиць, а завдання визначаються спеціальними картками. Карта рухається по землі, як деталі на заводі, переходячи від штату до штату, і відсоток станів, завершених на кожній фазі, вищий. На виході ми отримуємо елемент продукту, готовий до замовлення. Дошка з таблицями і картками може бути такою ж, як і електронна - знову ж таки Kanban не накладає ніяких обмежень на користувачів.

Історія Термін Канбан - це японський термін, який використовувався для позначення виробництва Toyota в 1960-х роках. В основі цього принципу лежить спосіб перенесення виробництва, а також різні швидкості окремих технологічних процесів у виробництві. Спробую пояснити на пальцях. У кожному виробництві є основне виробництво ("головний спонсор") і додаткове виробництво ("додатковий спонсор").

Швидкість випуску готового продукту визначається основним конвеєром, тоді як вторинні конвеєри не прискорюють швидкість випуску продукту, але можуть уповільнювати його в разі несвоєчасної видачі необхідних частин. Також під час виробництва можуть змінюватися

пріоритети. Наприклад, виявилося, що на ліводзеркальній виробничій станції було 20, а на праводзеркальній — 10, тоді як на конвеєрі 15 автомобілів і потрібно 15 дзеркал обох типів. Виникає конфлікт вимірювань – кількісне виробництво не зменшилося (додаткові конвеєри вчасно випустили 30 виробів), але виробництво все ще під загрозою зупинки.

Канбан має на меті допомогти вирішити цю проблему. У спрощеному варіанті Kanban включає два простих правила: виробнича станція має виробничий план у резерві. План для цього є, і його можна змінити в будь-який час (наприклад, якщо станція виробляє занадто багато лівих дзеркал, вона повинна мати можливість якнайшвидше перемикатися направо); Кількість одночасних завдань на станції обмежена (тобто одночасно можна виконувати не більше вказаної кількості дзеркал). Це обмеження необхідно для контролю швидкості виробництва на станції та швидкості реагування на зміни плану.

Останнім часом Kanban завойовує популярність в індустрії програмного забезпечення. Деякі команди вважають цю методологію надзвичайно корисною, інші використовують принцип «культу вантажу». Виходячи з мого емпіричного досвіду, чистий Kanban погано працює для продуктових команд (читай — «основних»), але добре працює з такими командами підтримки, як: команди підтримки програмного забезпечення, де важливий не «план», а швидкість реакції на зміни. важливо; тестові групи, що працюють окремо від команд розробників; послуги підтримки; інші приклади «непервинних галузей». Слід зазначити, що Kanban добре працює для стартапів, які не мають чіткого плану, але активно працюють над розвитком.

Пропоную розглянути приклад використання Kanban в розробці програмного забезпечення. Уявіть команду з одного розробника, яка працює над невеликим проектом. План розвитку (беклог) відсортований за пріоритетом роботи, ліміт команди на завдання в процесі - 1 шт.

Процес управління проектами в IT. Необхідно поступово розробляти

і впроваджувати PM в компанії і перевіряти кожен етап і взаємодію на практиці. Перейти на нові стандарти за один день нереально. Навіть щоб впровадити новий органайзер, потрібно: навчити команду, делегувати справи та завдання менеджеру таксі, призначити відповідальних людей і терміни, налаштувати багтрекер. Зазвичай команді важко звикнути до змін, а для інтеграції баз даних потрібен час. Хоча Worksection має довгу історію перенесення даних з інших сайтів, інструкції для новачків і сама система настільки інтуїтивно зрозумілі, що наші постійні клієнти кажуть у відгуках. Усі процеси управління проектами також є покроковими та проходять через 5 етапів:

• Розробка концепції, ініціація.

• Визначення та планування.

• Початок роботи та виконання плану.

• Контроль і моніторинг.

• Закриття проекту.

Організація управління IT-проектами Процес тверезого управління руйнує не лише функції, методи та алгоритми, а й відповідальність за результат. У кожному проекті є ролі, незалежно від його специфіки та кінцевого продукту. Найтовстіший поділ класичного зразка для наслідування:

• Власник.

• Актор.

• Споживач.

У комерційних компаніях таке поняття позначається як стейкхолдер. Все це впливає на результат і прибуток. Це перспективи.

1.7. Поняття і класифікація IT-проектів

Сьогодні в науковій та практичній літературі існує багато різноманітних визначень проектів. Ось декілька з них:

Проект — це сукупність зусиль, що здійснюються для досягнення конкретних, унікальних результатів протягом визначеного періоду часу та в

межах затвердженого бюджету, призначених для оплати ресурсів, використаних або спожитих у ході проекту [1].

Проект – це тимчасове намагання придбати новий (унікальний) продукт чи послугу за обмежені ресурси [2].

Проект - комплекс заходів, що складається із суміжних завдань, що виконуються різними функціональними організаціями (суб'єктами) з чітко визначеними цілями, графіком і бюджетом [3].

Проект – це діяльність, для якої щоразу реорганізуються людські, матеріальні та фінансові ресурси.

Основні особливості (особливості) проекту:

• Наявність конкретної цілі. Проекти створені для досягнення конкретних цілей. У деяких проектах цілі та вимоги до продукту можуть покращуватися в міру виконання проекту.

• зміна. Досягнення мети проекту завжди призводить до зміни деякої системи, елементом якої є проект.

• Обмежений час. Проект обмежений у часі – має конкретні дати початку та закінчення.

• Необхідні обмежені ресурси. Проекти реалізуються з точки зору ресурсів, організації та інших обмежень.

• Складність. При проектуванні враховується вплив зовнішніх і внутрішніх факторів.

• розмежування. Кожен проект необхідно відрізняти від інших проектів головної організації.

• Особлива організаційна структура. Часто необхідно встановити «особливу для проекту організаційну структуру» на час виконання проекту.

• Унікальність. Дизайни унікальні. Ступінь їх унікальності у них різний: це може бути пов’язано з цілями чи продуктом проекту та умовами їх досягнення.

Різноманітність реалізованих проектів величезна. Вони можуть відрізнятися за обсягом, конфігурацією спеціальності, масштабом діяльності,

складом учасників, рівнем складності і т. д. Програма - група суміжних проектів і різних видів діяльності, об'єднаних спільною метою та умовами їх виконання. На відміну від окремого проекту, програма вимагає специфічних методів координації та управління кількома проектами, щоб забезпечити загальну мету програми, дотримуючись обмежень та умов її виконання. Реалізація всієї програми забезпечує максимальну ефективність.

Портфоліо проектів – сукупність різноманітних, як правило, не пов’язаних між собою проектів, що здійснюються в інтересах однієї або кількох організацій (компаній), зазвичай піддаються загальним обмеженням ресурсів.

Вимоги проекту - формалізовані запити від клієнтів, спонсорів та інших зацікавлених сторін проекту щодо характеристик, які повинні відповідати цілям проекту, продуктів, послуг, короткострокових і довгострокових результатів, обмежень та інших умов проекту. Метою проекту є досягнення результатів (впливу, вигод) за наявності конкретних передумов та умов для їх реалізації. Стратегія проекту - спільне бачення шляхів і засобів досягнення мети проекту, що задає напрямок і основні принципи проекту. Характеризується системою (сукупністю) якісних і кількісних показників. Вимоги до проекту базуються насамперед на потребах клієнта, які формуються на основі наявних можливостей розвитку бізнесу та/або завдань, спрямованих на протидію потенційним проблемам (загрозам). Для визначення вимог необхідно визначити, ідентифікувати та пристосувати результати проекту до потреб і можливостей зацікавлених сторін, зокрема замовників та кінцевих користувачів продукту проекту.

Кінцева мета клієнта та ключових зацікавлених сторін проекту – отримати вигоду від використання результатів проекту. Мета має бути чіткою, конкретною, вимірною, зрозумілою та досяжною. Проектне завдання - операції, які необхідно виконати для отримання готового продукту проекту.

Під час реалізації проекту цілі проекту можуть бути уточнені та виправлені в результаті змін у проектному середовищі або прогресу проекту

та досягнутих проміжних результатів. Після визначення цілей проекту проводиться аналіз усіх можливих варіантів досягнення цілей проекту та вибирається той, який буде реалізований. Стратегія проекту має бути комплексною та охоплювати всі відповідні аспекти реалізації проекту. Після завершення проекту стратегію можна належним чином оновити та переглянути. Під час реалізації та після завершення проекту, досягнуті результати мають оцінюватись відповідно до цілей проекту та узгоджених критеріїв успіху.

Успіх результатів проекту можуть бути оцінені різними зацікавленими сторонами по-різному. Критерії успіху проекту - набір показників, що дозволяють оцінити ступінь успішності проекту. Успіх проекту означає отримання результатів від усіх зацікавлених сторін, які відповідають їхнім очікуванням, досягають цілей та відповідають вимогам.

Коли такі цілі та вимоги формулюються, критеріями успіху проекту можуть бути кількісні показники, що відображають ступінь досягнення цілей або вимог проекту. На першому етапі важливим завданням стає грамотне та прозоре представлення цих критеріїв. Керівник проекту відповідає за вибір та узгодження критеріїв успіху та варіантів їх оцінки із зацікавленими сторонами. Загальним критерієм успіху проекту є досягнення цілей проекту в обумовлені терміни та з обмеженими ресурсами. Критерії успіху мають бути визначені, оцінені та враховані в усіх проектах. Те, що цілі, поставлені на початку проекту, не були досягнуті, не завжди означає, що проект провалився. Якщо мета проекту змінюється, відповідні ризики та проблеми мають змінитися.

Поняття успіху в управлінні проектами пов’язане з успіхом проекту, але не однаково. Ви можете успішно контролювати реалізацію проекту, який пізніше переривається через втрату актуальності, наприклад, через зміну стратегії компанії. Якщо успіх проекту зазвичай пов’язаний з досягненням очікуваного бізнес-результату, то успіх управління проектом зазвичай вимірюється такими критеріями, як дотримання термінів і витрат на проект,

своєчасні поставки, якість зв'язку, час реагування на загрози та проблеми, тощо

Обидві категорії критеріїв – успіх проекту та успішність управління проектом – повинні бути відображені в зведеному плані проекту, який приймається всіма зацікавленими сторонами.

Організація та контроль реалізації проекту - фаза життєвого циклу управління проектом, на якій здійснюється організація. Організація проекту включає в себе основні процеси:

• Відповідно до плану управління проектом, існує розподіл функціональної відповідальності та підзвітності зацікавлених сторін.

• Експериментальна експлуатація системи управління проектом, у тому числі системи розрахунку показників фактичної реалізації проекту.

• Навчання користувачів, організація та прискорення командної роботи.

• Розподіл базових завдань та організація планової роботи. Інформаційна підтримка реалізації проекту. Оперативні заходи щодо організації проекту базуються на результатах фаз ініціації (концепції) та планування (зведений план проекту). Елементи поведінкових компетенцій менеджера та ключових членів команди управління проектом стають ключовими елементами в організації впровадження. Контроль включає моніторинг вимірювань та фіксацію виконання робіт під час їх виконання відповідно до отриманих інструкцій та фактичної розробки проекту.

Контрольну функцію в проекті виконує команда управління проектом з використанням існуючої системи контролю. Контроль змін і виконання проектних робіт відбувається в усіх функціональних областях проекту. На основі контролю готується звіт про виконання проекту, який надається зацікавленим сторонам відповідно до їх вимог та для подальшого аналізу, прогнозування та управління ходом робіт. Остаточний звіт про хід проекту зазвичай базується на інтеграції облікових даних за звітний період.

Звіти повинні бути максимально прозорими та створюватися через регулярні проміжки часу залежно від тривалості проекту, поточної фази

життєвого циклу проекту, рівня невизначеності та ризику в проекті, інтенсивності роботи, вимог зацікавлених сторін тощо.

1.8. Існуючі системи управління задачами

Trello — це платформа для підвищення продуктивності, яка дає змогу співпрацювати зі своєю командою та робити більше за обмежений проміжок часу. Його панелі та картки організують вашу роботу та допомагають визначити пріоритети невідкладних завдань. Він також відомий своїм простим інтерфейсом перетягування. Наприклад, ви можете просто перетягнути картки завдань з одного розділу в інший. Загалом це дозволяє визначити етапи проекту - від початку до кінця. Ви можете назвати кожну картку, призначити її всім членам команди, встановити дату тощо. Крім того, є вбудований майстер автоматизації робочого процесу, який дозволяє виконувати кілька завдань за вас.

Переваги:

• Пропонує кілька шаблонів для вибраних категорій, таких як технічні, маркетингові, бізнесові тощо.

• Простий процес застосування - просто скопіюйте шаблони, налаштуйте їх і почніть співпрацю.

• Можливість прикріплювати зображення та файли до кожної карти.

• Перетягування інтерфейс користувача.

• Легкий обмін і зберігання файлів

• Індикатори виконання завдань допомагають визначити пріоритети в списках справ.

Jira — чудовий варіант для зміни команд, щоб планувати, відстежувати й організовувати роботу. Ви можете створювати історії та позначати проблеми для кожного члена команди, щоб призначити окремі завдання. Повна видимість панелей дозволяє легко відстежувати хід виконання завдань. Ви також можете встановити різні фільтри карти, наприклад В. Повна, Неформульована та Неповна. Однією з найдивовижніших функцій є

те, що програмне забезпечення дозволяє вибрати робочий процес. Простіше кажучи, ви можете створити свій власний спосіб показу. Наприклад, ви можете призначити картки в такому порядку: Відкрити> Виконується >> Реєстрація >>> Остаточне схвалення >>>> Готово.

Ви також можете створити будь-який інший робочий процес. Крім того, ви можете розширити свій робочий процес шляхом інтеграції з іншими бізнес-інструментами, які ви вже використовуєте. Jira також має потужні заходи безпеки для захисту ваших даних.

Рис. 2 Статус виконання проекту

переваги:

• Пропонує панелі scrum і kanban для кращої прозорості та гнучкості.

• Створіть дорожні карти для своєї команди, щоб допомогти їм уявити завдання

• Звіти в режимі реального часу надають огляд процесу управління завданнями вашої компанії.

• Автоматизація та інтерфейс перетягування спрощують керування завданнями та економлять час.

• Мобільний додаток дозволяє зв'язатися з командою з будь-якого місця.

• Конфігуровані робочі процеси.

MeisterTask — це інструмент керування завданнями для команд, які хочуть виконати більше роботи за менший час. Інтерфейс користувача досить простий і зручний у навігації. Вам просто потрібно зосередитися на трьох основних блоках, а саме завданнях, панелях інструментів і проектах. Крім

того, є багато навчальних посібників та додаткової інформації, які допоможуть вам розпочати роботу.

Інші ключові функції включають регулярне програмне забезпечення для резервного копіювання та відновлення даних, інформаційні панелі в стилі Kanban, ідеальну інтеграцію зі сторонніми інструментами, детальні звіти та статистику тощо. Ви також можете стати «спостерігачем» і призначати завдання членам команди та залишатися в курсі подій.

Переваги:

• Користувацький інтерфейс перетягування перевірений і простий у навігації.

• Панелі в стилі Канбан спрощують керування роботою та забезпечують панорамний огляд робочих місць.

• Програмне забезпечення доступне на смартфонах, тому ви можете працювати зі своєю командою віддалено.

• Отримуйте автоматичні сповіщення, щоб залишатися в курсі.

РОЗДІЛ 2
ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ОРГАНІЗАЦІЇ СТАРТАП-КОМПАНІЇ

2.1.Технічне завдання

Ціль розробки та область використання

Метою розробки цього програмного забезпечення є розробка програмного модуля для організації стартап-компанії для управління завданнями в проектах розробки програмного забезпечення.

Цільовою групою цієї прототипу системи є програмісти та експерти з управління.

Область використання

Стартапи. Інструменти управління проектами сьогодні використовуються для вирішення складних, складних і простих завдань. Управління проектами в компаніях у цьому контексті включає виконання різноманітних завдань, вирішення проблем як всередині компанії, так і між різними компаніями. При цьому не повинно бути негативного впливу, негативного впливу на виробничі, організаційні аспекти діяльності підприємства. Область застосування методів управління проектами сильно варіюється, хоча інструмент управління проектами був розроблений в США для організації суднобудівних та авіаційних компаній. В даний час управління проектами стає все більш популярним і використовується в різних компаніях, незалежно від розміру, виду діяльності, цілей і завдань.

Користувальницький інтерфейс програми має бути розроблений зручним для користувача способом і відповідно до стандартів дизайну інтерфейсу.

Інтерфейс користувача складається з таких вікон:

- вікно авторизації;

- Реєстрація;

- Вікно для вибору активних проектів або додавання нових;

- Вікно для додавання/перегляду завдань для проекту.

Організація роботи з інтерфейсом програми повинна бути наступною:

- Запустити програму;

- Вийти з головного меню програми:

"Реєстрація"

"права"

«Створити новий проект»

«Створити завдання»;

У вікні «Реєстрація» введіть свої дані:

- Прізвище;

- Пошта;

- Пароль;

- Підтвердьте пароль

Специфікація вимог

Система має:

- Додаток дозволяє одній людині отримати доступ до кількох облікових записів з проектами

- Надайте користувачам доступ до облікового запису

- Можливість додавати завдання до окремих проектів.

- Додайте дошки для кожного проекту окремо

- Показуйте кілька проектів одночасно

- Системний програмний код повинен відповідати стандартам кодування мови програмування JS.

перспективи продукту

Ця система суттєво відрізняється від інших систем функціональною частиною та інтерфейсом в цілому. Система створена за методологією Kanban, завдяки якій ми не накладаємо жодних обмежень на користувачів.

Технічна та програмна платформа на якій повинна працювати система

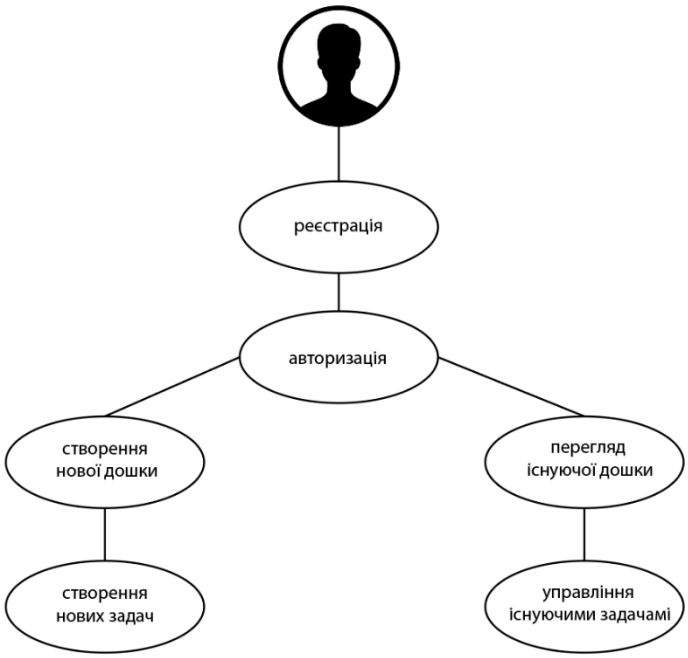

Рис. 2.1. Функціональні вимоги

Апаратні інтерфейси

Система працює на персональних комп'ютерах (ПК) з мінімальними функціями:

- 512 Мб оперативної пам'яті;

- Процесор Pentium 4 (або аналоги інших виробників) з тактовою частотою 2,0 ГГц;

програмні інтерфейси

- Операційна система: Windows, Mac OS;

-Microsoft .NET Framework 4.7;

- вузол JS (npm)

- WebStorm / VS Code

потреби бізнесу

Користувач системи повинен мати базові знання операційної системи Windows або Mac OS і знання браузера, оскільки він є веб-додатком.

Експерт-користувач повинен бути обізнаним з управління проектом, оскільки він створює завдання та контролює терміни, які впливають на результат проекту.

2.2. Концепція проектування програмного забезпечення

Проектування програмного забезпечення — це процес перетворення вимог користувача у формат, який розробник може використовувати для кодування та впровадження програмного забезпечення. Проектна документація створюється на етапі проектування програмного забезпечення на основі вимог замовника, зазначених у документі ЄСВ. Отже, метою цього етапу є перетворення документа ЄСВ у проектний документ.

У процесі проектування створюються та документуються такі елементи:

- Потрібні різні модулі.
- Модульні асоціації, якими можна керувати.
- Взаємодія між різними компонентами.
- Структура даних різних модулів.
- Алгоритми, які будуть реалізовані в кожному з різних модулів.

Діаграма класів

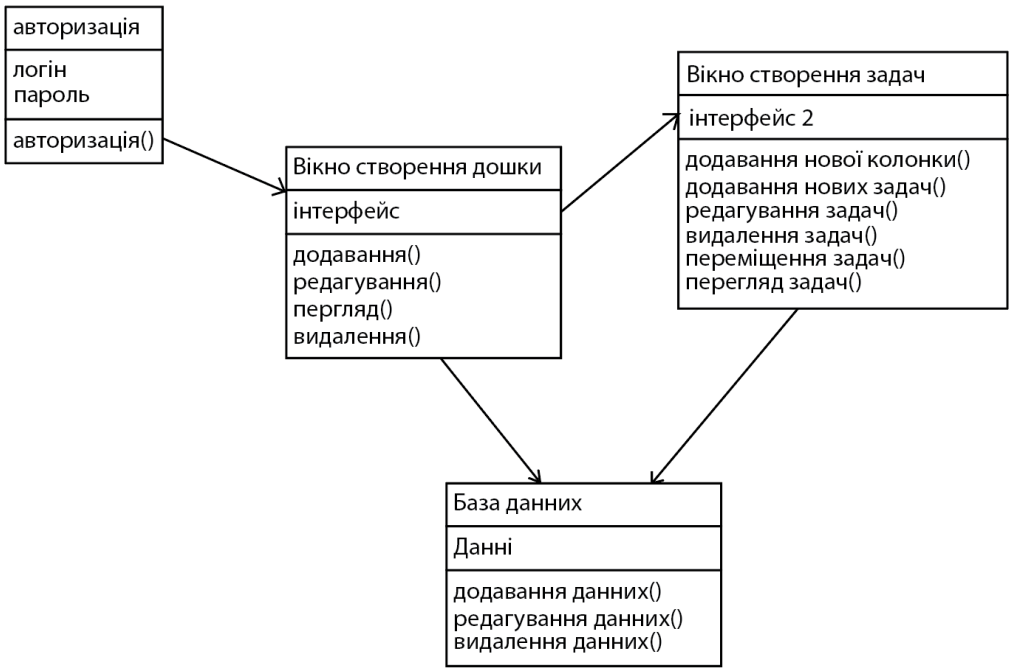

Рис. 2.2. Діаграма класів

Рис. 2.3. представляє архітектуру програмного модуля.

Крім того, для програмного модуля використовується шаблон проекту Model-View-ViewModel.

Model-View-ViewModel (MVVM) розшифровується як Model-View-ViewModel. Він заснований на шаблонах MVC і MVP з попередніх класів. Він використовується для того, щоб інтерфейс користувача не заважав бізнес-логіці програми.

Компоненти шаблону MVVM:

- Модель: як і шаблони MVC і MVP, модель містить усі дані та інформацію, необхідні для програми. Як відомо, модель не має нічого спільного з маніпулюванням або відображенням даних.

- Перегляд: Перегляд відображає дані в інтерфейсі, а також може отримувати введення користувача, тому він включає дії. Погляди в парадигмі MVVM не є пасивними. Контролер або динамік маніпулюють пасивними переглядами; Ви несете відповідальність за перегляд даних, нічого не знаючи про модель. Перегляди, з іншого боку, активні в MVVM. Вони містять

прив’язки даних, дії та події, які вимагають розуміння моделі та моделі відображення. Презентація керує своїми подіями; не повністю залежить від моделі перегляду; однак він не зберігає свого стану, якого досягає завдяки синхронізації з моделлю перегляду.

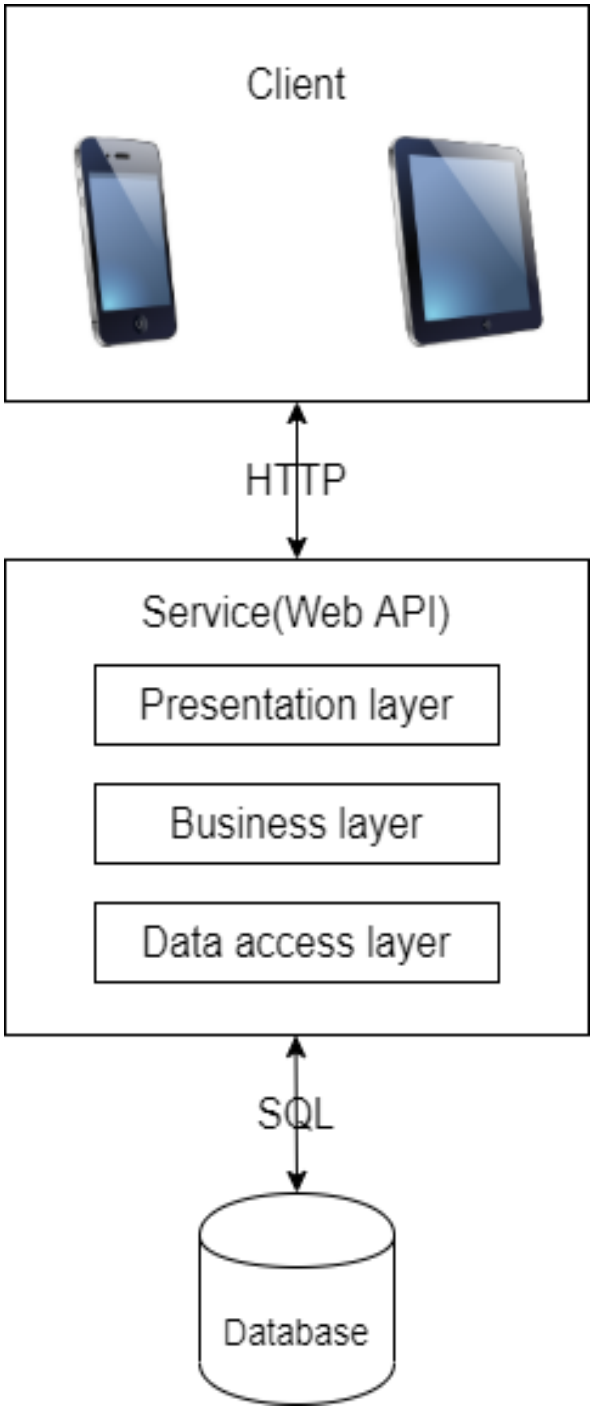

Рис. 2.3. Архітектура програмного модуля

- ViewModel: як і контролер у MVC, ViewModel служить сполучною ланкою між моделлю та представленням. Відображає інформацію,

перетворену з формату моделі у формат попереднього перегляду. Наприклад, модель може містити дату у форматі Unix, але подання може відображати її в іншому форматі. У цьому випадку ViewModel допомагає з перетворенням даних. Коли дія користувача відбувається у поданні, вона також оновлює модель; Звідси він використовується для відправки команд презентації до моделі. Він також використовується для відстеження стану подання та активації подій у ньому.

View і ViewModel

Виклики подій, даних і методів використовуються для зв’язування подання з моделлю подання. Модель браузера використовує команди для відображення подій у поданні. При двонаправленому зв’язуванні даних модель подання надає атрибути моделі, які змінюють подання.

Модель і ViewModel

ViewModel надає модель та її атрибути для прив'язки даних. Він також має API для отримання та форматування властивостей, що відображаються у поданні. Рисунок 3.3. представляє взаємодію між моделлю та ViewModel.

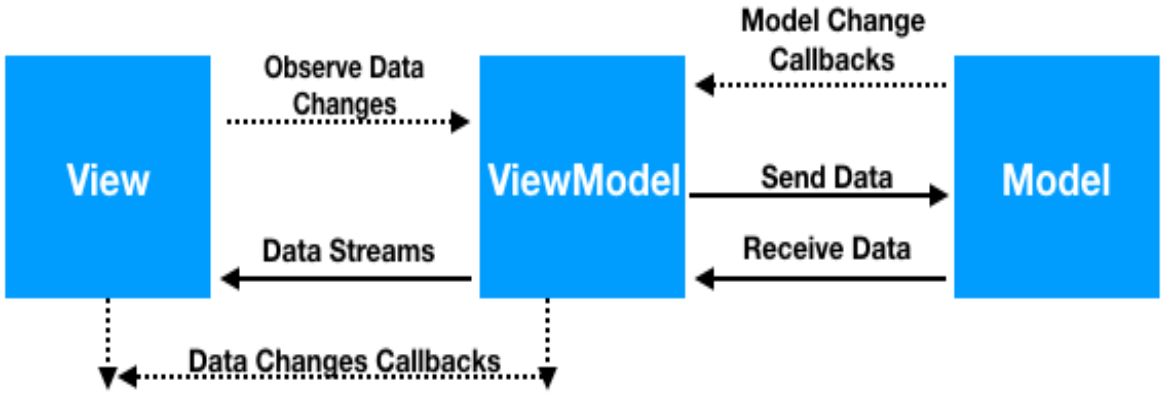

Рис. 2.4. Взаємодія між моделлю та ViewModel

Шаблон MVVM заснований на шаблоні спостерігача, як показано на малюнку. Модель у цьому випадку є суб’єктом, і саме модель сповіщає глядача (спостерігача) про зміни. Потім модель перегляду вносить необхідні

зміни до подання. З моделлю перегляду зміни у вигляді також відображаються в моделі.

Висновки

У цьому розділі здійснювався процес проектування програмного модуля. У ході цього процесу були створені його архітектура, компонент, реалізація та діаграми класів. Це було зроблено для того, щоб спростити майбутній процес впровадження програмного забезпечення.

РОЗДІЛ 3
ПРОГРАМНИЙ МОДУЛЬ ОРГАНІЗАЦІЇ СТАРТАП-КОМПАНІЇ

3.1. Технології розробки

Для розробки програмного модуля будуть використовуватися такі технології:

- ASP.NET Core MVC
- Ядро Entity Framework
- Azure SendGrid

ASP.NET Core MVC

Моделі, подання та контролери — це три основні набори компонентів архітектурного шаблону Model-View-Controller (MVC). Цей шаблон допомагає розподілити проблеми. Запити користувачів надсилаються до Контролера в цьому дизайні, який відповідає за взаємодію з Моделлю для виконання дій користувача та/або отримання результатів запиту. Контролер вибирає подання, яке буде показано користувачеві, і надає всі необхідні дані моделі. Рисунок 3.1 зображує три основні компоненти та те, як вони пов’язані один з одним.

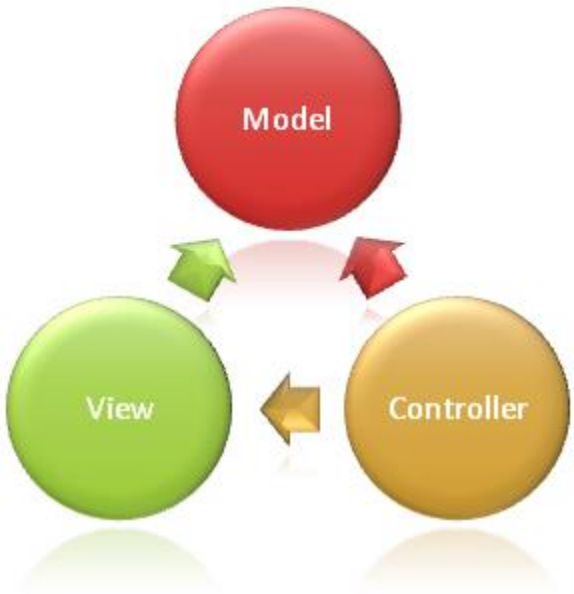

Рис. 3.1. Компоненти MVC

Оскільки створювати, налагоджувати та тестувати щось (модель, подання чи контролер) простіше за допомогою одного завдання, такий розподіл обов'язків допомагає масштабувати програму з точки зору складності. Оновлення, тестування та налагодження коду, який має залежності в двох або більше з цих трьох доменів, є більш складним. Логіка інтерфейсу користувача, наприклад, швидше за все зміниться, ніж бізнес-логіка. Коли код презентації та бізнес-логіка об'єднані в один об'єкт, об'єкт бізнес-логіки необхідно оновлювати щоразу, коли змінюється інтерфейс користувача. Це часто призводить до проблем, які вимагають повторного тестування бізнес-логіки з кожною незначною зміною інтерфейсу користувача.

У додатку MVC модель представляє стан програми, а також будь-яку бізнес-логіку або дії, які нею мають виконуватися. Бізнес-логіка, а також будь-яка логіка реалізації для збереження стану програми повинні бути загорнуті в модель. Типи ViewModel зазвичай використовуються в строго типізованих представленнях для зберігання даних, які будуть відображатися в представленні. Модель використовується для створення та заповнення цих об'єктів ViewModel контролером.

Views відповідають за відображення матеріалу в інтерфейсі користувача. Для вбудовування коду .NET у розмітку HTML вони використовують механізм перегляду Razor. У поглядах мало логіки, а будь-яка логіка повинна бути пов'язана з поданням вмісту. Подумайте про використання компонента View, ViewModel або шаблону представлення, щоб спростити перегляд, якщо вам потрібно виконати багато логіки у файлах перегляду, щоб відобразити дані зі складної моделі.

Контролери — це частини системи, які керують введенням користувача, взаємодіють з моделлю і, нарешті, вибирають, який вигляд відображати. Подання в програмі MVC просто показує дані; контролер відповідає за обробку та реагування на введення та взаємодію користувача. Контролер є першою точкою входу в архітектуру MVC, і він відповідає за

рішення, з якими типами моделей мати справу та який перегляд представляти (звідси його назва — він контролює, як програма реагує на заданий запит).

Фреймворк ASP.NET Core MVC — це легка, відкрита та високо перевірена платформа презентацій, розроблена спеціально для ASP.NET Core.

ASP.NET Core MVC — це шаблонна основа для створення динамічних веб-сайтів із чітким розподілом обов'язків. Він дозволяє вам повністю контролювати розмітку, сприяє розробці, дружній до TDD, і дотримується останніх веб-стандартів.

Ядро Entity Framework

Entity Framework (EF) Core — це кросплатформна версія популярної технології доступу до даних Entity Framework, яка є легкою, розширюваною та відкритим вихідним кодом.

Хоча EF Core чудово абстрагує багато деталей програмування, існує кілька рекомендованих практик, які можна застосувати до будь-якого O/RM, щоб запобігти типовим підводним каменям виробничих програм:

- Для створення архітектури, усунення несправностей, профілю та переміщення даних у високопродуктивних виробничих програмах потрібне розуміння базового сервера баз даних середнього або вищого рівня. Знання основних і зовнішніх ключів, обмежень, індексів, нормалізації, операторів DML і DDL, типів даних, профілювання тощо — це лише кілька прикладів.

- Для репрезентативних навантажень можна проводити тестування продуктивності та стрес-тестування. Рудиментарне застосування кількох функцій не масштабується належним чином. Наприклад, кілька колекцій. Інтенсивне використання відкладеного завантаження, умовні запити до неіндексованих стовпців, величезні оновлення та вставки з даними, створеними в магазині, відсутність керування паралельністю, великі моделі та недостатня стратегія кешу — це лише кілька прикладів.

- Обробка рядків підключення та інших секретів, дозволи бази даних для операцій без розгортання, перевірка введених даних для необробленого

SQL та шифрування конфіденційних даних – усе це приклади перевірок безпеки.

- Перевірте, чи реєстрація та діагностика є адекватними та корисними. Наприклад, відповідне налаштування ведення журналу, теги запитів і аналіз додатків.

- Виправлення помилок. Підготуйтеся до типових ситуацій збою, включаючи відкат версії, резервні сервери, масштабування та балансування навантаження, пом’якшення DoS та резервне копіювання даних, підготувавши надзвичайні ситуації.

- Міграція та розгортання додатків. Плануйте, як міграції застосовуватимуться під час розгортання; це на початку програми може спричинити труднощі з паралельністю та вимагати більше дозволів, ніж потрібно для нормальної роботи. Щоб полегшити відновлення після фатальних проблем міграції, використовуйте постановку. Додаткову інформацію див. у розділі Застосування міграцій.

- Детальна перевірка та тестування створених міграцій. Перш ніж застосовувати міграції до виробничих даних, їх слід ретельно перевірити. Після того, як таблиці містять виробничі дані, неможливо змінити форму або типи стовпців схеми. У SQL Server, наприклад, nvarchar(max) і decimal(18, 2) рідко є оптимальними типами для стовпців, зіставлених у рядкові та десяткові атрибути, але EF використовує їх за замовчуванням, оскільки не знає вашої унікальної обставини.

Двигун бритви

Компонент перегляду ASP.NET MVC ідентичний сторінці Razor. Він пропонує майже ті ж можливості, що й MVC, а також той самий синтаксис. Основна відмінність між MVC і Razor Pages полягає в тому, що код і модель контролера містяться на сторінці Razor.

Простіше кажучи, його можна порівняти з фреймворком MVVM, оскільки він забезпечує швидшу розробку та двостороннє прив’язування даних із окремими проблемами.

Хоча MVC добре працює з веб-проектами, які включають велику кількість односторінкових програм, динамічних серверних переглядів, запитів AJAX і REST API, Razor Pages найкраще підходять для простого введення даних або сторінок лише для читання.

Служби розробки ASP.NET зараз широко використовуються для створення онлайн-додатків, і вони мають ряд переваг. Як рішення MVVM, ASP.NET Web Forms спеціально створено в MVC. Razor Pages, з іншого боку, є результатом наступної еволюції веб-форм ASP.NET.

Як відомо, MVC розшифровується як Model-View-Controller. Це архітектурний шаблон, який використовується при розробці програмного забезпечення для створення інтерфейсів користувача. Незважаючи на те, що MVC є однією з найвідоміших фреймворків і що багато веб-розробників використовують її по всьому світу, вона має певні недоліки. Два найбільших недоліку:

1. Складність. У ASP.NET MVC є багато складних концепцій, таких як RouteCollection, TempData, Linq to SQL, ViewData, Lambda Expression, Controller Action, Custom Route і HTML Helpers, які допомагають підключити View, Controller і Model . Розробники не можуть використовувати ASP.NET MVC для створення веб-проекту, поки не зрозуміють усі основні ідеї. Крім того, навіть після їх розуміння вони все одно можуть зіткнутися зі складністю, особливо при розробці великомасштабних програм.

2. Витрати. Незважаючи на те, що і модель, і подання розділені в ASP.NET MVC, веб-розробники не можуть ігнорувати подання моделі взагалі. Це пов’язано з тим, що якщо модель регулярно змінюється, погляди програми можуть бути переповнені запитами на оновлення. Перегляди — це графічні дисплеї, для створення яких потрібен деякий час залежно від складності програми. Крім того, якщо модель була значно оновлена, а ваша програма є складною, представлення може відставати від будь-яких запитів на оновлення. В результаті розробникам доведеться приділити більше зусиль для усунення проблеми, можливо, збільшивши витрати.

Переваги:

1. Сторінки Razor добре організовані. Щоб все правильно назвати, спроектувати динамічні маршрути тощо, знадобиться чимало зусиль. Razor Pages, з іншого боку, набагато більш організований. У Razor Pages файли групуються ефективніше. Кожен файл має повний код, а також режим Razor View, як і старі веб-форми ASP.NET.

2. Єдина відповідальність. Якщо розробник ніколи раніше не використовував фреймворк MVC, він, можливо, помітив, що величезні класи контролерів переповнені безліччю дій. Усі ці класи можна порівняти з вірусом, який збільшується в розмірах із додаванням нових і нових елементів. У порівнянні з MVC, кожна сторінка програми в Razor Pages є автономною, а її код і подання впорядковані разом, що призводить до меншої складності в довгостроковій перспективі.

Azure SendGrid

Під час роботи над проектом розробники можуть автоматично надсилати електронні листи групі одержувачів. Зазвичай вони роблять це, надсилаючи електронні листи через SMTP-сервер, налаштований на стороні клієнта, що коштує великих грошей і вимагає тривалого обслуговування сервера. Розробники можуть використовувати службу доставки електронної пошти SendGrid, доступну в Microsoft Azure, якщо їм потрібно мати справу з низькою вартістю, мінімальним обслуговуванням, а також мати підписку на Azure.

SendGrid — це хмарне рішення SMTP для маркетингу та трансакційної доставки електронної пошти. SendGrid керує сервером електронної пошти вашої організації, забезпечуючи надсилання та доставку ваших повідомлень за потреби.

SendGrid справляється з усіма технічними потребами — від інфраструктури до роботи з Інтернет-провайдерами та моніторингу до послуг із білого списку та аналітики в режимі реального часу. Це найбільша хмарна служба доставки електронної пошти у світі.

Служби електронної пошти SendGrid можна використовувати різними способами. Однак це повністю залежить від ваших вимог і цілей.

3.2. Інтерфейси програмного модуля

Головне меню показано на рис. 3.2.

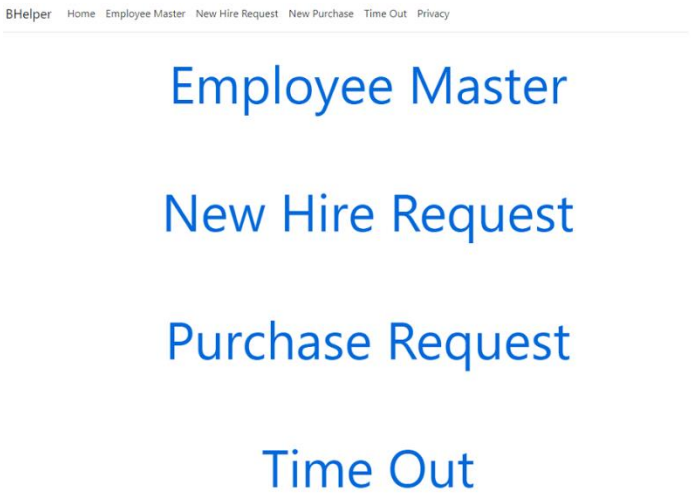

Рис. 3.2. Головне меню

Employee Master показано на рис. 3.3.

Є список співробітників компанії. Співробітник відображається з ім'ям, прізвищем, його роллю та з меню для дій.

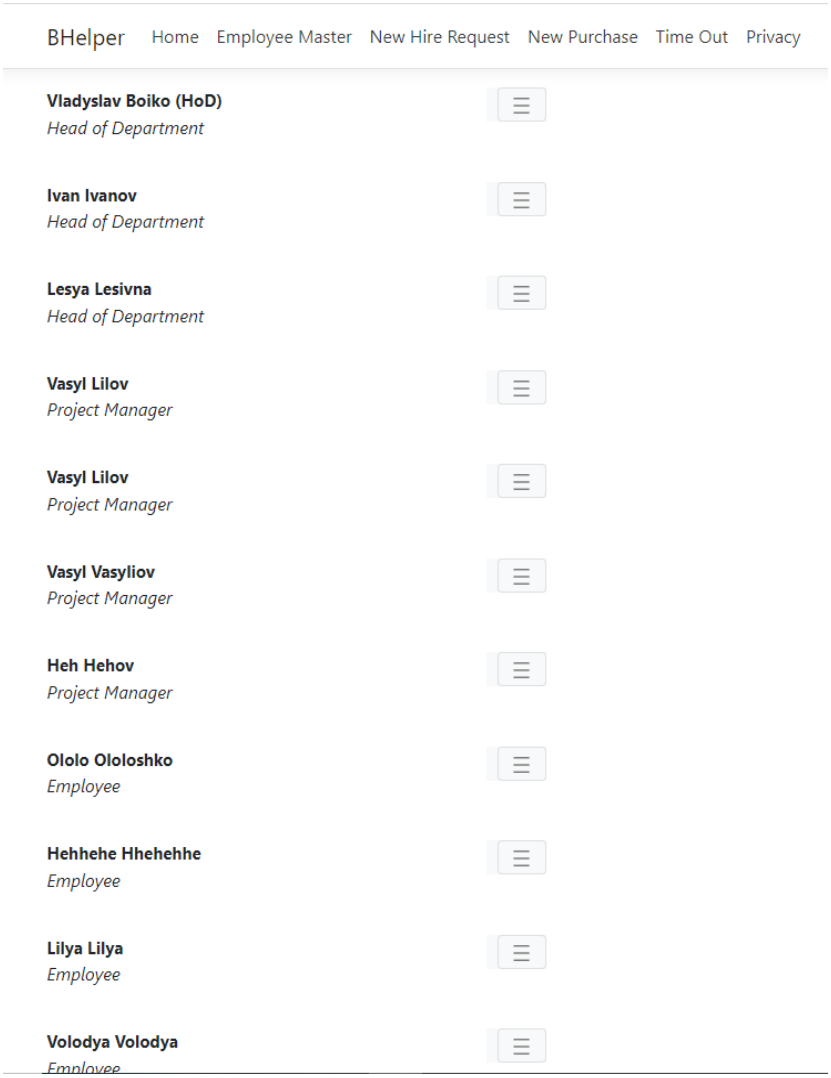

Рис. 3.3. Вкладка співробітник

Дії в меню відображені на рис. 3.4.

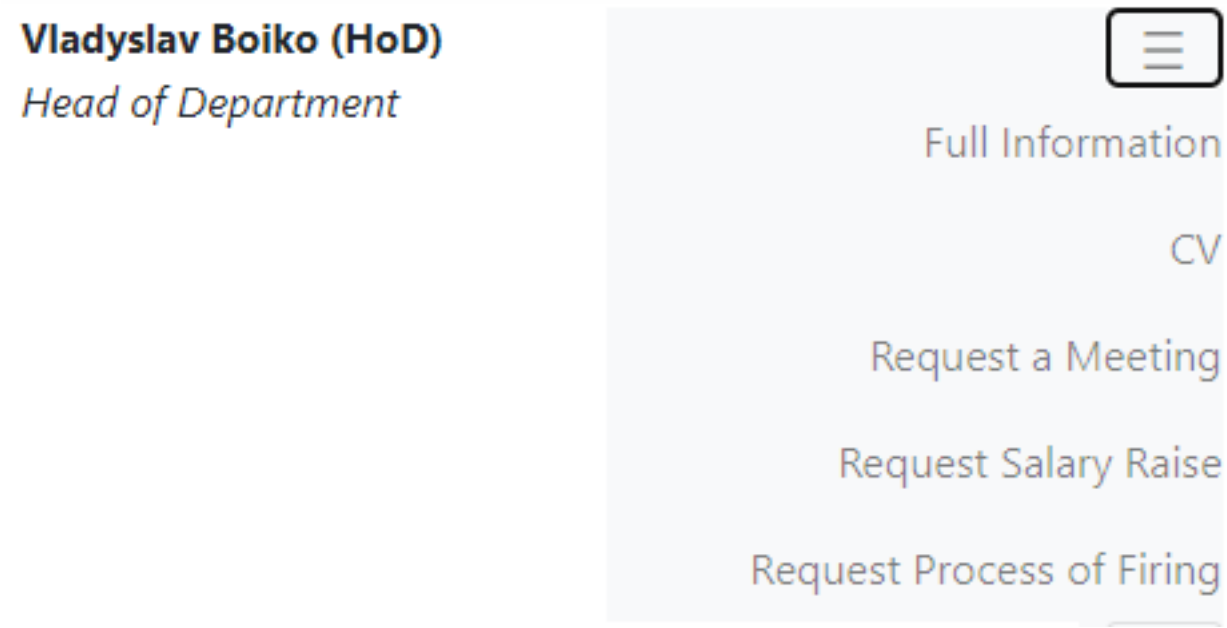

Рис. 3.4. Дії в меню

На сторінці «Новий запит на найму» міститься форма для визначення критеріїв для нового працівника, яка наведена на рис. 3.5.

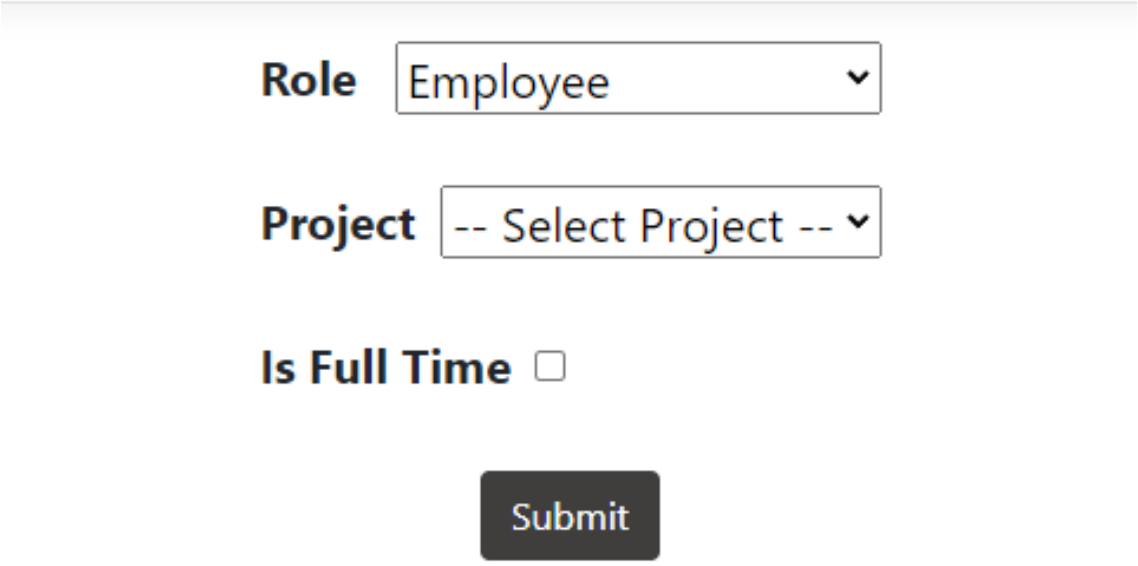

Рис. 3.5. Нова сторінка запиту на найму

Поле «Роль» містить попередньо визначені ролі, що показано на рис. 3.6.

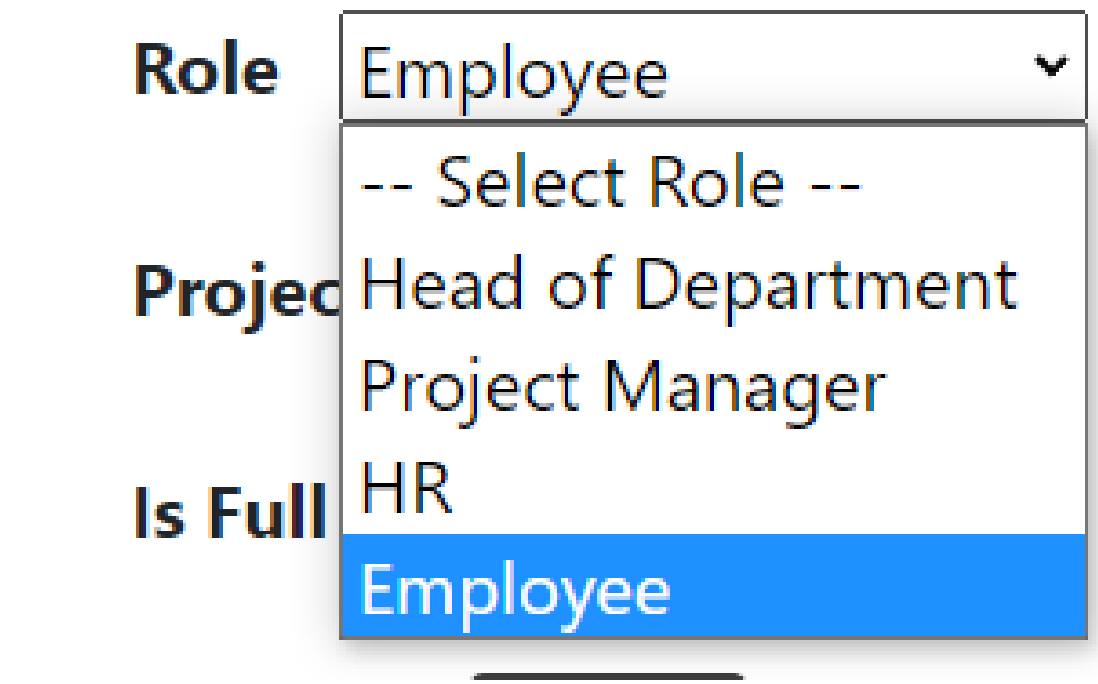

Рис. 3.6. Рольове поле

Поле Project містить проекти, які визначені в системі, що показано на рис. 3.7.

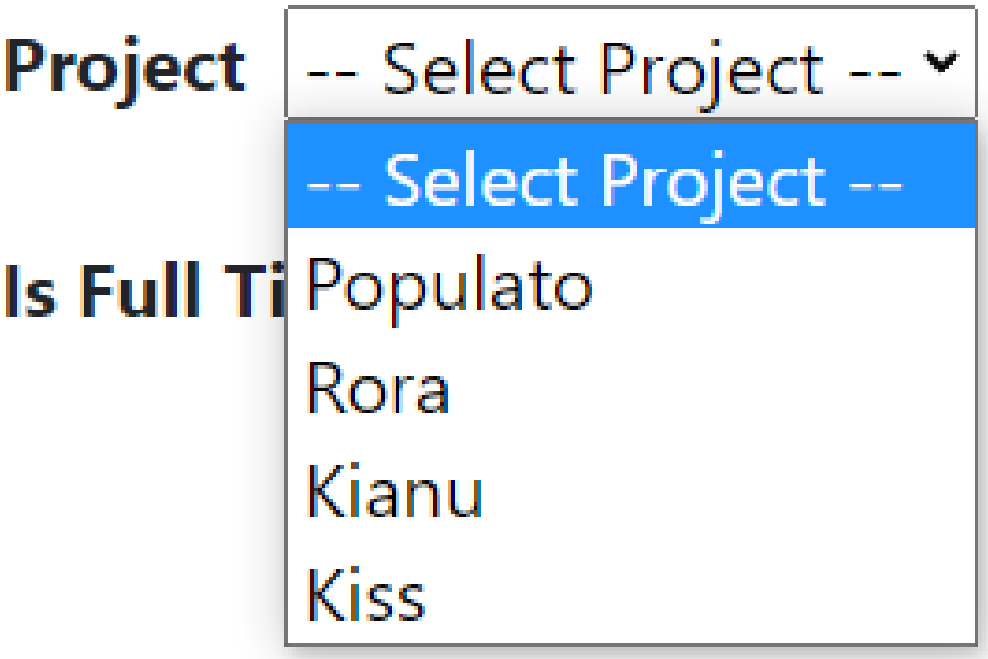

Рис. 3.7. Поле проекту

Усі вказані поля показані на рис. 3.8.

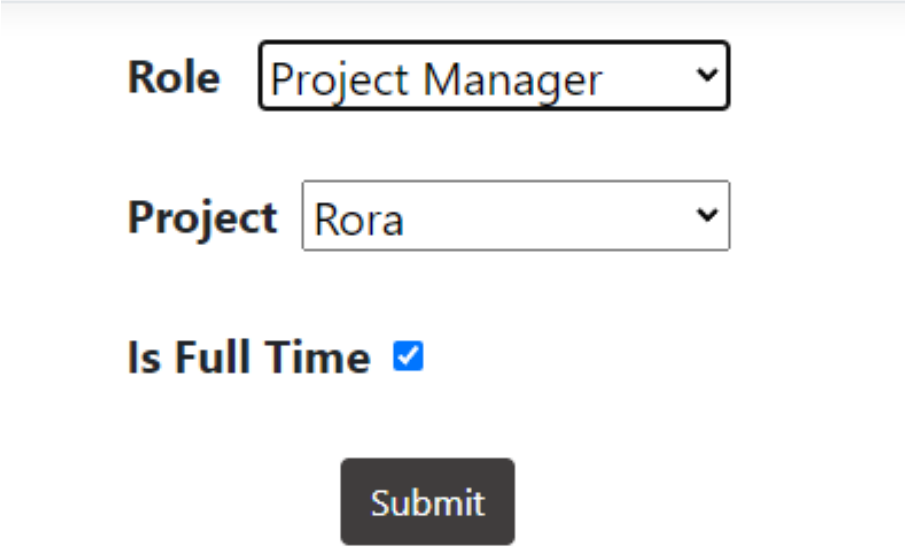

Рис. 3.8. Визначені поля форми

Сторінка Time Out показана на рис. 3.9.

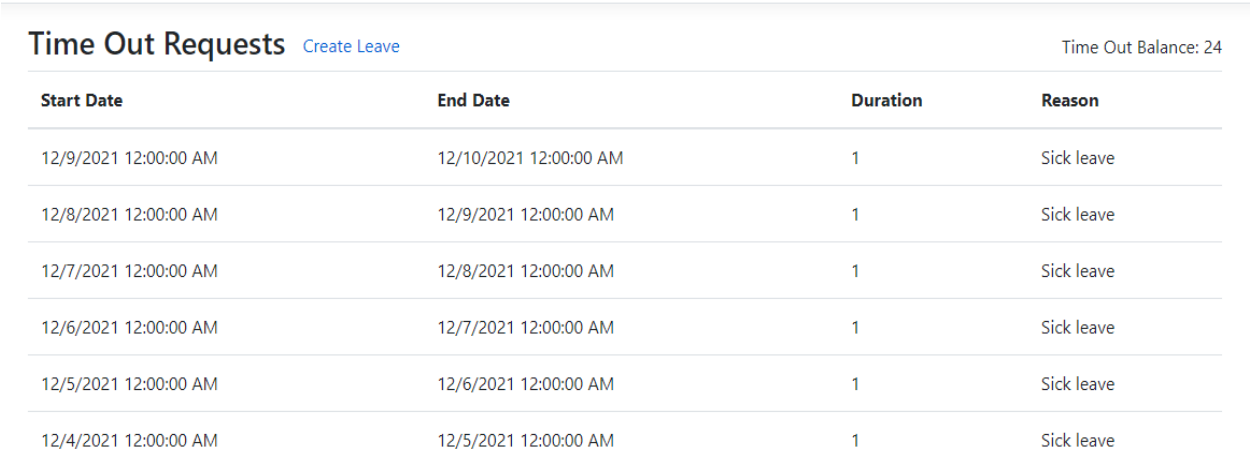

Рис. 3.9. Сторінка Time Out

Після натискання кнопки «Відправити» до списку додається новий запит на час очікування та зменшується залишок часу очікування, що показано на рис. 3.10.

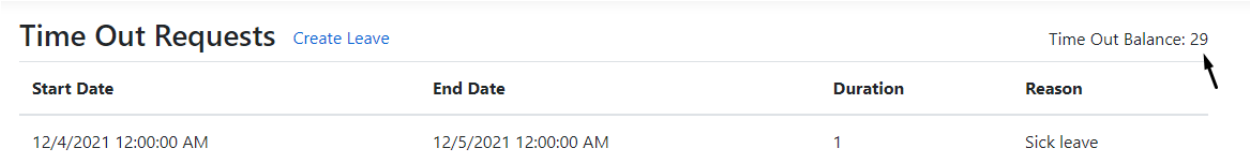

Рис. 3.10. Новий запит на час очікування

Висновки

У даному розділі були розглянуті технології, яки були використані при розробці, а також наведені скріншоти інтерфейсів розроблювано засобу.

ВИСНОВКИ

Після етапу аналізу домену було повністю зрозуміла важливість BPM в компаніях, а особливо у стартап-компаніях. Тому було досліджено, як правильно будувати такі процеси.

Після аналізу сучасних ринкових рішень було виявлено їх переваги та недоліки. Справа в тому, що майже всі вони не охоплюють усі основні процеси всередині компанії.

Після всієї виконаної роботи було отримано великий досвід аналізу домену, аналізу рішень та аналізу вимог. Було дуже корисно зрозуміти, що повинна робити програма і як вона може знайти свій успіх на ринку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розробка стартап-проектів: Конспект лекцій [Електронний ресурс] : навч. посіб. для студ. спеціальностей 151 – «Автоматизація та комп'ютерно-інтегровані технології» та 152 – «Метрологія та інформаційно-вимірювальна техніка» / О. А. Гавриш, К. О. Бояринова, К. О. Копішинська; КПІ ім. Ігоря Сікорського. – Електронні текстові данні (1 файл: Х,ХХ Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 188 с.
2. Тіль П. Від нуля до одиниці. Нотатки про стартапи, або як створити майбутнє / Пітер Тіль ; пер. з англ. Р. Обухів. – К.: Наш формат, 2015. – 232 с.
3. 9 методів оцінки вартості стартапа. URL: https://www.imena.ua/blog/valuation-forstartups/
4. Стартап проекти та їх оцінювання: конспект лекцій для студентів за спеціальністю «Інженерія програмного забезпечення» факультету інформаційних технологій УжНУ Розробник: к.т.н. Поліщук В.В. – Ужгород: 2018. – 74 с.
5. Робул Ю. Управління стартапами. URL: https://imbacademy.com.ua
6. Мрихіна О. Б. Перспективи стартап-компаній у контексті конкурентоспроможного розвитку українського ринку високих технологій / О. Б. Мрихіна, А.Р. Стояновський, Т.І. Міркунова // Актуальні проблеми економіки. – 2015. – №9 (171). – С. 215-225
7. Стартап. Вікіпедія — вільна енциклопедія: веб-сайт. URL:https://uk.wikipedia.org/wiki/Стартап
8. Колесник В.І. Управління стартапами в Україні: проблеми та перспективи Економіка харчової промисловості, 2017, 4/2, С.51-61.
9. Чазов Е.В. Стартап как новая форма ведения бизнеса. Наукові праці НУХТ. 2013. № 52. С.122-128. 3.3. Інформаційні ресурси в Інтернеті
10. Глобальний Договір ООН [Електронний ресурс] — Систем. Вимоги : Pentium-266 ; 32 Mb RAM ; Windows 98/2000/NT/XP. — Web : www.oblrada. ks.ua/index.php?id=10501.

11. Що таке стартап? Бізнес-UA: веб-сайт. URL: http://biznesua.com.ua/shho-take-startap/

12. Савін М. Що таке стартап. FORBES Україна: веб-сайт. URL: http://forbes.net.ua/ua/explain/startup_and_business/1363540-shcho-take-startap

13. Organizational project management maturity model, 2013 – 246 с.