

Національний технічний університет України
«Київський політехнічний інститут ім. Сікорського»

Кафедра обчислювальної техніки

Проект «Система управління проектами»
з дисципліни « Організація баз даних»

Виконали:
студенти 2 курсу
ФІОТ гр. ІО-82

Бригада:
Білич М.В
Галацин З.Б
Кузенний П.В
Роман О.Т
Самутін О.І
Шевчук О.Б
Ющенко А.В

Київ
2019

Зміст

1. Вступ. Система Управління Проектами	3
1.1 Мета управління проектами.....	3
1.2 Групи оцінок успішності проекту	4
1.3 Ролі в проекті.....	4
1.4 Методології управління проектами	5
1.4.1 Традиційна (Каскадна) методологія управління проектами.....	5
1.4.2 Методологія управління проектами PRINCE2.....	6
1.4.3 Гнучка методологія управління проектом (Agile Project Management)	7
1.4.4 Методологія швидкої розробки додатків (Rapid Application Development - RAD)	8
1.5 Порівняння програмних засобів.	9
2. Короткий огляд продукту	10
2.1. Основна інформація.....	10
2.2. Призначення системи управління.....	10
2.3. Політика взаємодії з користувачем	10
2.4. Характеристика ділового процесу	11
2.5. Призначення системи.....	11
2.6. Надійність	11
2.7. Функціональність.....	12
3. Ділові правила та прописи.....	12
3.1 Діаграма юзкейсів	13
3.2 Сценарії реєстрації користувача:	13
3.3 Сценарії входу у систему користувачем:	13
3.4 Сценарії для створення груп:.....	14
3.5 Сценарії додавання інших користувачів до групи:.....	14
3.6 Сценарії видалення користувачів з групи:	15
3.7 Діаграми послідовностей	16
3.7.1 Діаграма 1.1	16
3.7.2 Діаграма 1.2	17
3.7.3 Діаграма 1.3	18
3.7.4 Діаграма 1.4	19
3.7.5 Діаграма 1.5	20
4. Висновок	20

5. Список використаної літератури.....	21
--	----

1.Вступ. Система Управління Проектами

Управління проектами - область діяльності, в ході якої визначаються та досягаються чіткі цілі проекту при балансуванні між обсягом робіт, ресурсами (такими як гроші, працю, матеріали, енергія, простір і іншими), часом, якістю та ризиками.

1.1 Мета управління проектами

«Метою управління проектом (ами) є досягнення заздалегідь визначених цілей при заздалегідь відомих обмеженнях і доцільному використанні можливостей, реагуванні на ризики.»

Навіть при досягненні поставлених цілей і доцільності змін, проект може не відповідати очікуванням зацікавлених сторін. У проектах з високим рівнем змін потрібно керувати очікуваннями.

1.2 Групи оцінок успішності проекту

Успішність проекту різним чином оцінюється в різних методиках. Успішність може по-різному оцінюватися різними учасниками проекту.

Групи оцінок успішності:

- Орієнтовані на контракт з жорсткою фіксацією вимог і мінімізацією змін в ході проекту, наприклад традиційні методології, в тому числі PMBOK: «Проект успішний, якщо виконаний відповідно до затверджених критеріїв: обсягу, терміну, якості». Тобто проект успішний, якщо виконаний і закритий договір між Замовником та Виконавцем (незалежно від того, чи був він юридичним документом в разі зовнішніх проектів або визначався як-то інакше в разі внутрішніх проектів). При цьому оцінка успішності єдина як для замовника так і для виконавця.
- Орієнтовані на задоволеність замовника з гнучким управлінням вимогами, наприклад гнучкі методології SCRUM: «Проект успішний, якщо замовник задоволений»
- Орієнтовані на тривалий взаємодія з Замовником: управління програмами, Спрямоване на тривалий взаємодія, а не на один проект / контракт. Тут робиться акцент на продовження співпраці Виконавця з Замовником в рамках наступних проектів і іншої взаємодії.
- Збалансовані, наприклад PRINCE2: «Проект успішний при збалансованості принаймні за трьома категоріями - бізнесу, орієнтації на користувача і технологічної зрілості». Тут робиться акцент на фінансовій успішності проекту, задоволеності користувачів і розвитку технологій. Оцінка успішності може відрізнятися з точки зору бізнесу, користувача і виконавця. Такі методики оцінки частіше використовуються для внутрішніх проектів, коли замовник і виконавець знаходяться в одній організації.

Так, наприклад, проект, укладіть в узгоджені терміни і витрати, але не окупить за результатами проекту (витрати великі, результат неактуальний наприкінці проекту, замовник не може скористатися результатом і т. П.) Буде успішним за традиційною методологією, але не успішний по методології, орієнтованої на замовника.

Відповідальність за неуспішність такого проекту несе замовник і, в деяких випадках, проектний офіс або служба замовника.

1.3 Ролі в проекті

У багатьох випадках у проекті виділяють ролі замовника, виконавця (і іноді інвестора або спонсора). Такі ролі майже завжди є для зовнішніх проектів. Для внутрішніх проектів такий поділ ролей також бажано з метою підвищення ефективності при поділі праці і для усунення конфлікт інтересів при прийманні результатів, визначення зон відповідальності.

Замовник визначає мету і обмеження проекту і його фінансування. Виконавець виконує проект згідно із затвердженим планом.

Замовник несе відповідальність за постановку і актуальність цілей та пріоритетів, ефективність експлуатації результатів проектів. Централізацією функцій замовника і управлінням портфеля проектів займається проектний комітет. У будівельних організаціях для цього виділяють спеціальну службу єдиного замовника.

У разі чіткого поділу ролей замовник-виконавець метою управління проектом є стабілізація робіт і мінімізація відхилень від затвердженого замовником плану.

Якщо замовник і виконавець знаходяться в різних організаціях, то складається договір на виконання проекту. При зміні вимог замовника може бути підписана додаткова угода до договору в рамках обмежень сумарного бюджету програми проектів, Обумовлених основним договором.

Для пов'язання проекту з інтересами бізнесу часто вводять ролі куратора (зазвичай від виконавця) і іноді спонсора (куратора від замовника), які мають найбільшу обізнаність про інтереси бізнесу, мають право затверджувати ключові зміни в проекті.

1.4 Методології управління проектами

1.4.1 Традиційна (Каскадна) методологія управління проектами

Традиційна методологія управління проектами може бути використана в усіх галузях, але найбільш поширена в будівництві. Вона також носить назву каскадної або Водоспадної моделі, внаслідок того, що пропонується нею послідовність фаз нагадує потік. Методологія виділяє сім послідовних етапів проектного управління:

1. визначення вимог
2. проектування
3. Реалізація (будівництво, виробництво ...)
4. впровадження
5. Тестування та налагодження
6. установка
7. Експлуатація та супровід

Перехід до наступної фази проекту можливий тільки в тому випадку, якщо попередній етап завершений і прийнятий замовником. Застосування даної методології найпереважніше в проектах, результатом виконання яких є матеріальний продукт (наприклад, будівельні проекти, установка обладнання та ін.) І для реалізації яких потрібна конкретна послідовність дій. Крім того, розроблені плани можливо використовувати повторно для аналогічних проектів в майбутньому.

Однак з іншого боку, каскадна модель управління проектом вимагає інвестицій в планування. Зазвичай перші дві фази займають від 20 до 40% від усього часу виконання проекту по даній методології. Внаслідок структурованого підходу, внесення змін в зміст робіт відбувається повільно, що робить методологію не гнучкої і непотрібної в випадках, якщо клієнт не впевнений, який результат він хоче отримати.

1.4.2 Методологія управління проектами PRINCE2

PRINCE2 (Projects in Controlled Environments) так само є структурованою методологією до проектного управління. Це одна з найпопулярніших методологій управління проектами, широко використовувана в Великобританії в управлінні як в бізнесі, так в органах влади. PRINCE2 - це процесно-орієнтована проектна методологія, яка фокусується на процесах верхнього рівня (управління, організація, контроль), а не на нижчих завданнях (декомпозиція робіт, розробка графіків). Методологія PRINCE2 базується на семи принципах, семи темах і семи процесах. Принципи є центральним елементом методології: якщо хоча б один з них не виконується, то не можна говорити про те, що проект виконується в рамках PRINCE2.

Принципи методології PRINCE2:

1. Постійна оцінка економічної необхідності - залишається чи незмінною економічний зиск від проекту протягом усього життєвого циклу проекту
2. Навчання на досвіді - команда проекту повинна постійно шукати і вивчати досвід попередніх проектів
3. Визначення ролі моделі - команда проекту повинна мати ясну організаційну структуру і залучати відповідних людей для вирішення потрібних завдань
4. Управління по етапах - необхідно, щоб проекти були сплановані, а також піддавалися моніторингу та контролю на кожному етапі виконання;
5. Управління за відхиленнями - слід чітко позначити допустимі межі відхилень в проекті, щоб встановити межі відповідальності
6. Фокус на продуктах - необхідно концентруватися на визначенні та досягненні якості продуктів (результати проекту)
7. Адаптація до проектного середовища - слід адаптувати процеси і інструменти управління проектом до вимог проектного середовища, а також до масштабу робіт, їх складності, важливості, кваліфікаційним вимогам і ступеня ризику

Аспекти є напрями проектного управління, на які слід звертати увагу протягом тривалості всього проекту.

Аспекти методології управління проектами PRINCE2:

1. **Обґрунтування проекту:** яку цінність проект принесе організації?

2. **організація:** Яким чином необхідно розподілити ролі і відповідальність між членами проектної команди для того, щоб ефективно управляти проектом
3. **якість:** Які є вимоги і критерії до якості і яким чином можна їх забезпечити
4. **плани:** Кроки, необхідні для розробки плану, і інструменти PRINCE2, необхідні для використання
5. **ризики:** Яким чином менеджмент проекту буде вирішувати проблему наявності невизначеностей в плані проекту і в зовнішньому середовищі
6. **зміна:** Як керівництво проекту буде оцінювати вплив непередбачених завдань і змін і реагувати на них
7. **прогрес:** Реалізація проекту, виконання планів і подальший розвиток проекту

Нарешті, сім процесів розбивають життєвий цикл проекту на різні фази, для кожної з яких є свої рекомендовані до прийняття дії, одержувані продукти та зони відповідальності.

PRINCE2 має на увазі наступні процеси управління проектом:

1. запуск проекту
2. керівництво проектом
3. ініціація проекту
4. контроль етапів
5. управління створенням продукту
6. управління кордонами етапів
7. закриття проекту

PRINCE2 дозволяє стандартизувати процедури управління проектами, поліпшити координацію діяльності, а також допомагає зрозуміти, яким чином слід планувати проект і здійснювати моніторинг його виконання, що слід робити, якщо план проекту не виконується. Однак методологія PRINCE2 не є кращим вибором для проектів невеликого масштабу або для проектів з більшим ступенем імовірності змін обсягу робіт і вимог до них.

1.4.3 Гнучка методологія управління проектом (Agile Project Management)

Гнучке управління проектом являє собою поступальну і ітеративну проектну методологію. Її головною особливістю є те, що на початку виконання проекту точно невідомо, яким повинен бути кінцевий продукт і яким буде життєвий цикл проекту. Замість цього, проектна діяльність розбивається на кілька ітеративних фаз, званих «Спринт». Кожен спринт складається з безлічі завдань і має свій кінцевий продукт і результат. Методологія

Agile дозволяє менеджерам проектів постійно отримувати зворотний зв'язок і покращувати продукт після кожної ітерації.

Відповідно до даної методології управління проектами, відповідальність за результат ділиться між трьома ролями:

- **власник продукту** - визначає проектні цілі, розробляє оптимальний графік при заданих проектних параметрах, адаптує процес виконання проекту до зміненим вимогам і встановлює пріоритети в характеристиках продукту
- **Scrum майстер** - встановлює пріоритети у виконанні завдань командою проекту і усуває виникаючі труднощі, що перешкоджають цьому
- **члени команди** - виконують більшість поставлених завдань, здійснюють щоденний менеджмент, створюють звіти про хід виконання проекту, контролюють якість продукту

Методологія Agile є гнучкою і дозволяє легко змінити параметри проекту, що є значущим для таких сервісно-орієнтованих проектів, як розробка програмного забезпечення або графічний дизайн. Але це методологія не підходить для проектів зі строго заданими параметрами та вимогами.

1.4.4 Методологія швидкої розробки додатків (Rapid Application Development - RAD)

Швидка розробка додатків (RAD) - це проектна методологія, найчастіше використовувана в проектах по розробці ПЗ, основною метою яких є швидке і якісне створення програми. Дана методологія управління проектами виділяє 4 стадії проекту:

- планування
- користування проектування
- швидке конструювання
- перемикання

Методологія швидкої розробки додатків, з одного боку, допомагає поліпшити показники результативності проекту та підвищити якість ризик-менеджменту. Але з іншого боку, дана методологія не підходить для масштабних ІТ проектів, може привести до низької якості коду і вимагає постійного залучення клієнта в процес виконання всього проекту.

1.5 Порівняння програмних засобів.

Назва	опис	Складання звітів	створювати пріоритет	графічний інтерфейс	створення груп	Визначати робочих днів	Призначення прав доступу членам	Відстежує успіхи проекту	Створювати to-до листи	відслідковувати помилки	Виконання одного завдання групою людей
Jira	онлайн сервіс, що дозволяє командам-розробникам планувати проекти, призначати виконавців завдань, виставляти пріоритети і терміни.	+	-	+	+	+	-	+	-	+	-
Slack	web-сервіс для створення чатів і окремих груп для ведення обговорень з командою	-	-	+	+	-	-	-	-	-	-
GanttPro	дозволяє планувати і управляти проектами онлайн, візуалізувати процеси, створювати завдання і призначати їх учасникам, виставляти терміни виконання	+	-	+	-	+	+	+	-	+	+
Redbooth	web-сервіс, який дозволяє відстежувати помилки, планувати проекти і активності, управляти ресурсами і ставити завдання учасникам проекту	+	+	+	+	+	+	-	-	+	-
Basecamp	онлайн-сервісів для спільної роботи над проектами.	+	-	+	+	-	-	-	+	-	-

Asana	web-додаток,, що дозволяє призначати завдання, виставляти пріоритети і дедлайни.	-	+	+	+	+	-	-	+	-	-
Trello	популярний онлайн-сервіс	-	+	-	+	+	-	-	+	-	-
Gemini	додаток для управління проектами, що включає систему стеження за вадами	-	+	+	-	+	-	-	-	+	-

2. Короткий огляд продукту

2.1. Основна інформація

Система Управління Проектами складається з двох розділів програмного забезпечення:

1. Централізована база даних на головному сервері
2. Локальний програмний продукт на боці користувача, який міститиме в собі базу даних, оновлюватиме її з бази даних на сервері і пропонуватиме модератору додати нові записи у базу у відкладеному режимі, у разі відсутності інтернет з'єднання на момент готовності запису.

2.2. Призначення системи управління

Для швидкого і зручного управління діяльністю команди що залучена до проекту.

Головним сценарієм є надання зацікавленій особі (управляючий проектом) потрібної інформації щодо управління проектом.

2.3. Політика взаємодії з користувачем

Користувачем системи є управляючий проектом, що безпосередньо керує розподілом різних частин проекту серед членів команди що займається проектом.

Користувач системи може, згідно встановлених адміністраторами правил, отримувати інформацію з бази даних, а також редагувати і доповнювати її за необхідності.

2.4. Характеристика ділового процесу

Кожен працівник системи виконує свою роботу. Менеджер першого рівня керує працю менеджерів другого рівня. В свою чергу менеджери другого рівня відповідають за працю своїх підлеглих (воркерів). Кожен воркер виконує своє певне завдання, а також має певні ресурси для його виконання. За працездатність системи відповідають як воркери (програмісти, дизайнери та інші), так і менеджери усіх рівнів.

2.5. Призначення системи

Система призначена для полегшення управління проектами, що полягає в покращенні ефективності різних учасників проекту, та оптимізації часу виконання проекту, за допомогою певної автоматизації в управлінні та розподілі задач у команді.

2.6. Надійність

1. Захист особистих даних виконується у відповідності до закону України про захист персональних даних (**Відомості Верховної Ради України (ВВР), 2010, № 34, ст. 481**)
2. Резервне копіювання повинно проводитись для забезпечення цілісності системи та збереження даних після неочікуваних збоїв та інших обставин, що можуть призвести до їх втрати.
3. Система повинна бути добре захищена від різного роду зловмисних атак із метою заволодіння інформації або її неперифікованої зміни.
4. Система повинна витримувати великі навантаження, обслуговуючи значну кількість користувачів.

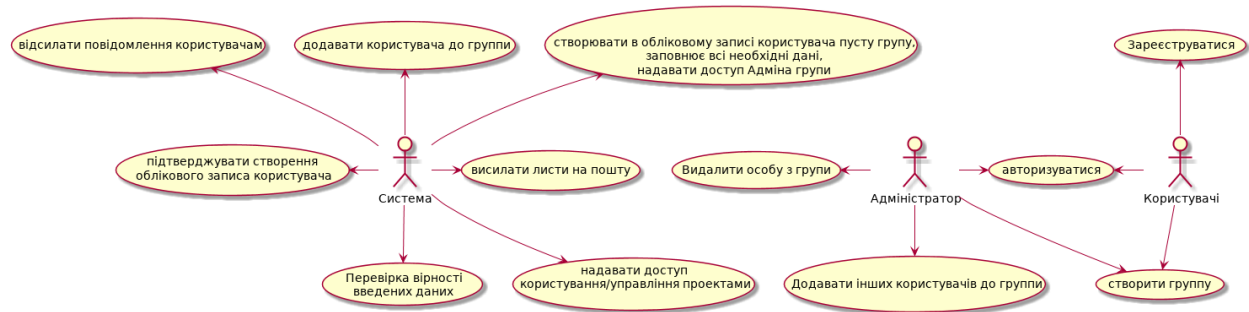
2.7. Функціональність

Серед найбільш важливих функціональних можливостей відзначимо наступні:

- наявність єдиної точки доступу до всієї інформації, що відноситься до проектів;
- створення нових документів з автоматичним зв'язуванням їх з конкретним фрагментом проекту (роботою, вузлом WBS) або з проектом в цілому;
- перехід до роботи з документами безпосередньо з середовища календарно-ресурсного планування, і навпаки;
- ініціювання процесу проходження (виконання) документа, отримання відміток про виконання, узгодження, ознайомленні або відхилення документа;
- контроль стану документа в процесі проходження ним певного бізнес-процесу, пов'язаного з управлінням проектом (наприклад, узгодження і приймання робіт).

3. Ділові правила та прописи

3.1 Діаграма юзкейсів



3.2 Сценарії реєстрації користувача:

Ідентифікатор	UC-1.1
Назва	Реєстрація нового Користувача
Учасники	Користувач, Система
Передумова	Відсутня
Результат	Створено новий обліковий запис для користувача
Виключні випадки	Користувач не підтверджує ідифікацію
Основний сценарій	<ol style="list-style-type: none"> 1) Користувач заходить на сайт та натискає кнопку «Реєстрація» 2) Користувач заповнює поля Username, password та e-mail 3) Користувач підтверджує вірність введених даних 4) Система для ідифікації відсилає користувачу письмо на пошту для підтвердження даних 5) Користувач заходить на пошту та преходить за посиланням 6) Система підтверджує створення облікового запису користувача

3.3 Сценарії входу у систему користувачем:

Ідентифікатор	UC-1.2
Назва	Входу у систему
Учасники	Користувач, Система
Передумова	Користувач має Username, password
Результат	Надання користувачу доступу для користування/управління проектами
Виключні випадки	Username або password не вірні

Основний сценарій	<ol style="list-style-type: none"> 1) Користувач заходить на сайт та натискає кнопку «Вхід» 2) Користувач заповнює поля Username, password та натискає «Log in» 3) Система перевіряє вірність Username, password 4) При наявності такого система надає доступ до користування/управління проектами
-------------------	--

3.4 Сценарії для створення груп:

Ідентифікатор	UC-1.3
Назва	Створення групи
Учасники	Користувач(Адмін), Система, інші користувачі
Передумова	Користувач має Username, password
Результат	Створення групи для об'єднання з іншими користувачами
Виключні випадки	Username або password не вірні
Основний сценарій	<ol style="list-style-type: none"> 1) Користувач заходить на сайт та натискає кнопку «Створити Групу» 2) Користувач заповнює поля назва групи, вибирає доступ приватності, вказує ціль даної групи 3) Перевіривши введені дані підтверджує створення нової групи 4) Система створює в обліковому записі користувача пусту групу, заповнює всі необхідні для цього дані та надає доступ Адміна до цієї групи

3.5 Сценарії додавання інших користувачів до групи:

Ідентифікатор	UC-1.4
Назва	Додавання користувачів до групи
Учасники	Адмін, Система, інші користувачі
Передумова	Адмін створив групу, отримання запрошення до вступу в групу
Результат	Інші користувачі додані до групи
Виключні випадки	Відмова користувачем примати запрошення
Основний сценарій	<ol style="list-style-type: none"> 1) Після створення Адміном групи він натискає кнопку «Додати інших осіб»

	<ol style="list-style-type: none"> 2) Отримує поле вводу, куди записує Username користувачей яких він хоче додати 3) При знаходженні даних осіб система відображає їх Адміну 4) Адмін натискає кнопку «Підтвердити» 5) Система після цього відсилає користувачам запрошення на e-mail та на їх обліковий запис. 6) Інші користувачі для підтвердження вступу у групу переходять за посиланням яке указане у письмі 7) Система додає користувача до групи 8) Система відсилає повідомлення Адміну, що його запрошення прийнято
--	--

3.6 Сценарії видалення користувачів з групи:

Ідентифікатор	UC-1.5
Назва	Видалення користувачів з групи
Учасники	Адмін, Система, інші користувачі
Передумова	Створена група, користувачі вступили у групу
Результат	Видалення з групи користувача
Виключні випадки	Не має
Основний сценарій	<ol style="list-style-type: none"> 1) Адміном переходить до власної групи та натискає кнопку «Видалити особу з групи» 2) Отримує поле вводу, куди записує Username користувачей яких він хоче видалити 3) При знаходженні даних осіб система відображає їх Адміну 4) Адмін натискає кнопку «Підтвердити» 5) Система після цього відсилає користувачу письмо (про те що користувача було видалено з групи) на їх обліковий запис. 6) Система видаляє користувача з групи 7) Система відсилає повідомлення Адміну, що операція пройшла успішно

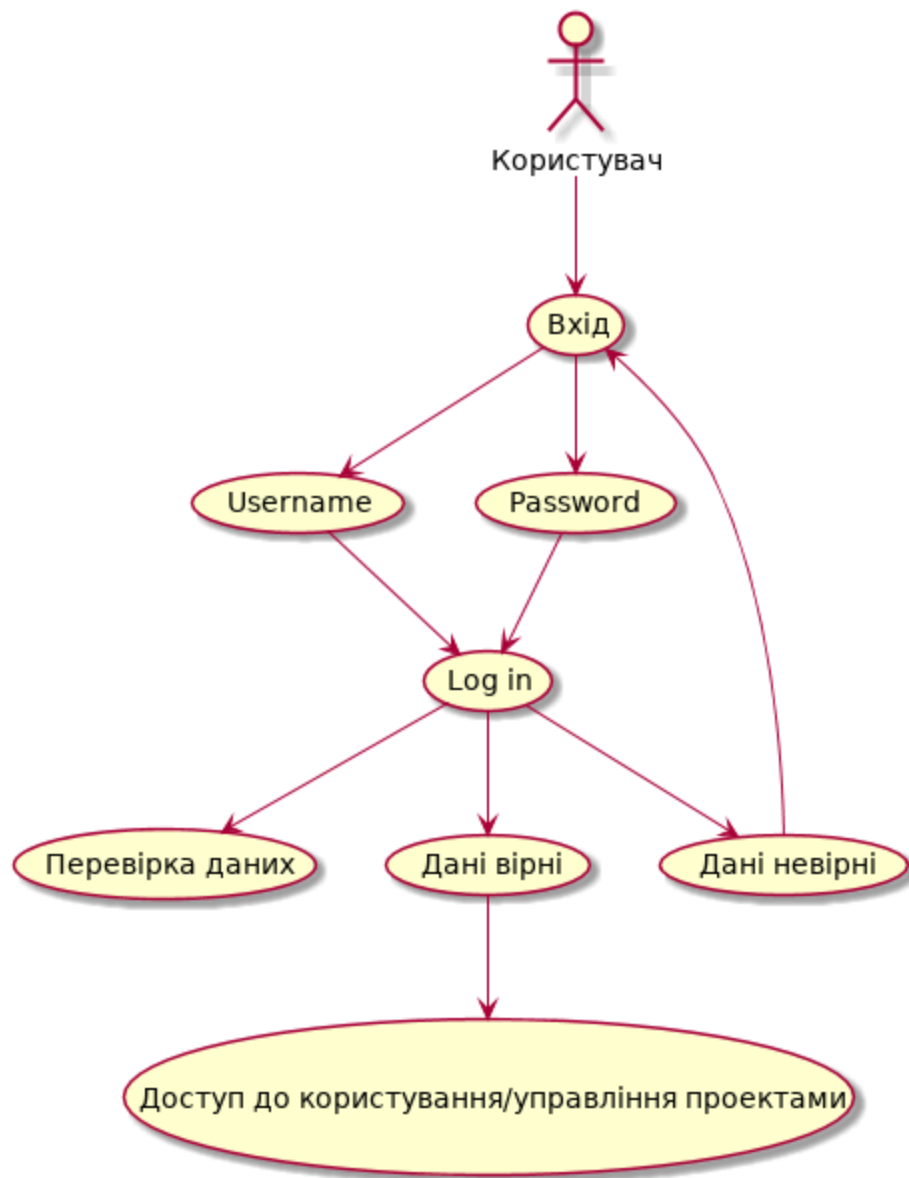
3.7 Діаграми послідовностей

3.7.1 Діаграма 1.1



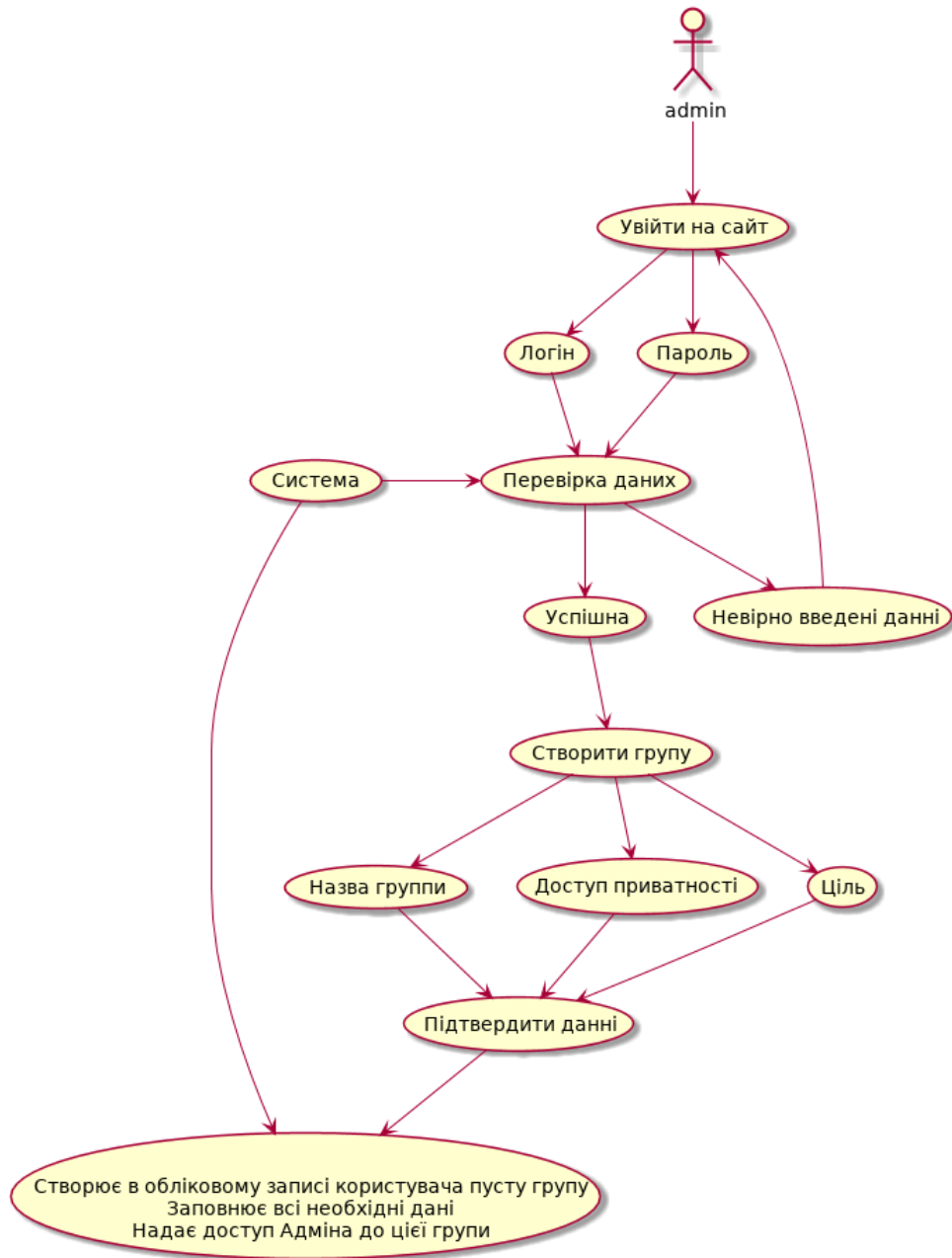
3.7.2 Діаграма 1.2

Вхід у систему



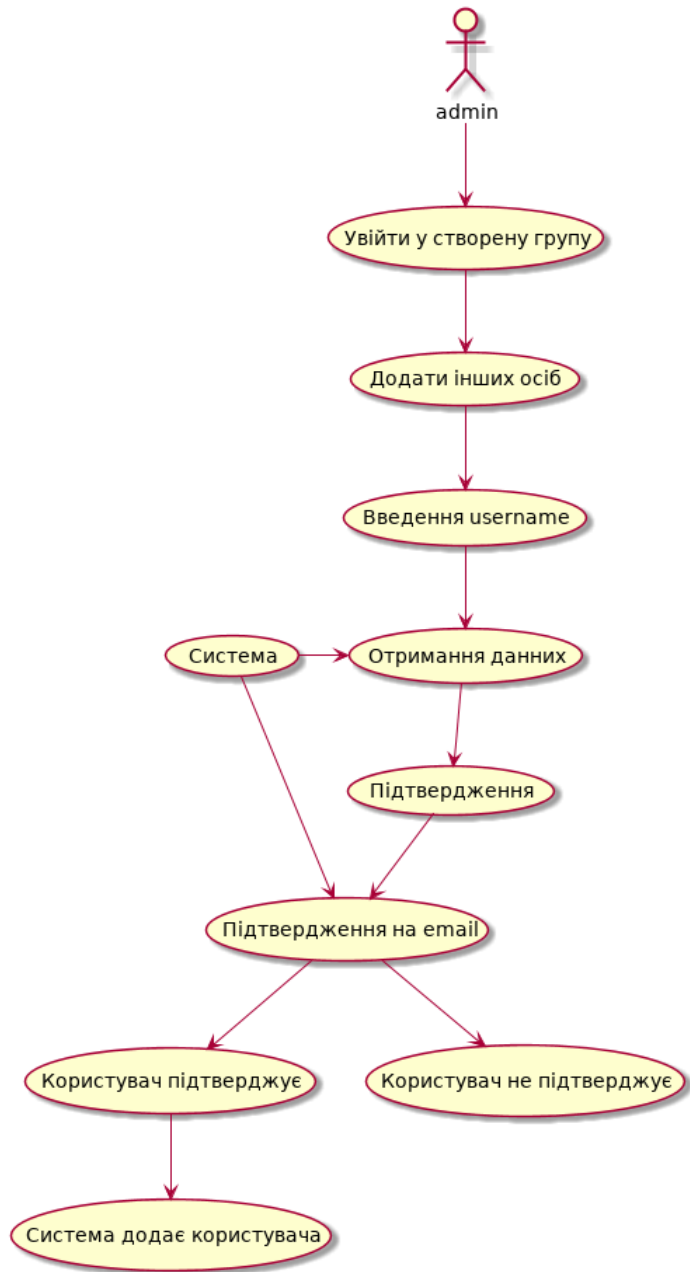
3.7.3 Діаграма 1.3

Створення групи

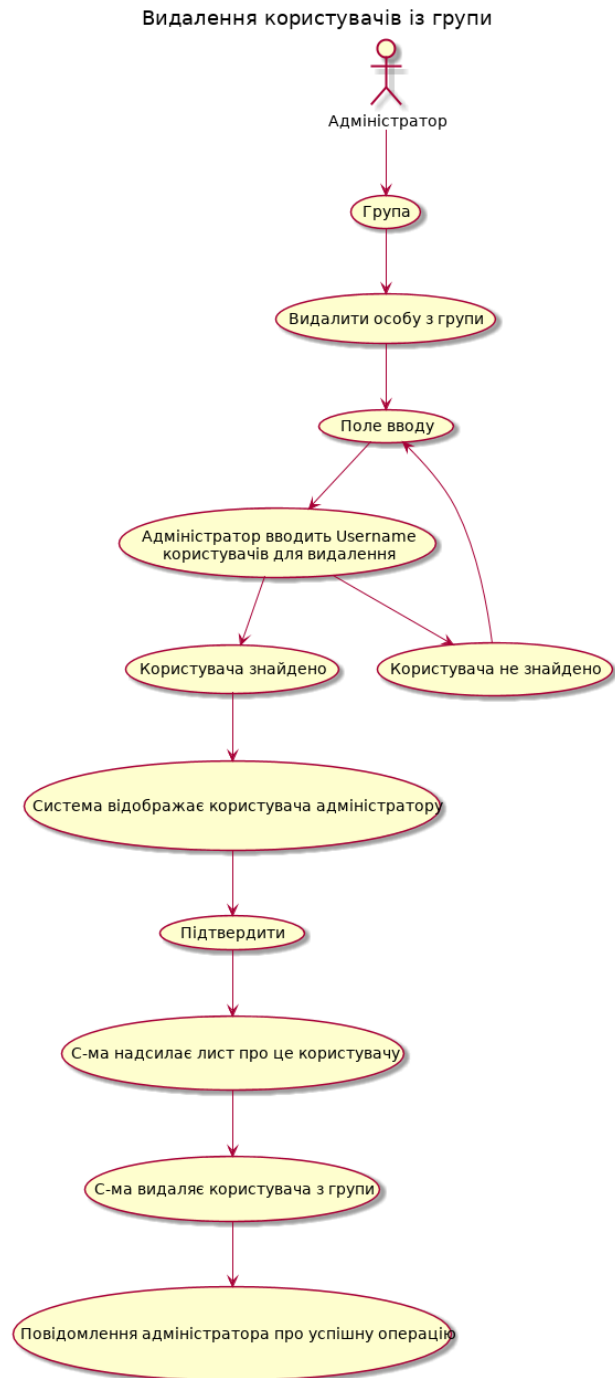


3.7.4 Діаграма 1.4

Додавання користувачей до групи



3.7.5 Діаграма 1.5



4.Висновок

Був проведений аналіз ринку. При проведенні даного аналізу було виявлено, що представлені на ринку технології не можуть повністю задовольнити потребу швидкого і зручного управління діяльністю команди,

що залучена до проекту. На основі цих факторів було прийнято рішення розробити систему управління проектами, яка б об'єднувала у собі існуючі у різних системах технології, а також була доповнена деяким функціоналом (див. П. 2.9)

5. Список використаної літератури

- https://ru.wikipedia.org/wiki/Управление_проектами
- <http://upr-proektom.ru/standarty-upravleniya-proektami>
- <https://dic.academic.ru/dic.nsf/ruwiki/48137>
- <https://www.pmservices.ru/project-management-news/top-4-metodologii-upravleniya-proektami/>
- <https://business.gov.kz/ru/business-development/tools/project-management-tools.php>
- <http://earchive.tpu.ru/bitstream/11683/40037/1/TPU392450.pdf>
- <http://pm.pacc.ru/stroydom/item/237-top4.html>