

Комп'ютерно-інформаційних систем та програмної інженерії

(повна назва факультету)

Програмної інженерії

(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

рівень вищої освіти - другий (магістерський)

(назва освітнього ступеня)

на тему: Інтеграція месенджерів в середовище ATutor.
Експорт повідомлень скриньки і чату менеджер.

Виконав: студент 6 курсу, групи СПм-61
спеціальності 121- Інженерія програмного забезпечення

(шифр і назва спеціальності)

	<u>Віглінський О. Ю</u> (підпис)	<u>Віглінський О. Ю</u> (прізвище та ініціали)
Керівник	<u>Пастух О. А.</u> (підпис)	<u>Пастух О. А.</u> (прізвище та ініціали)
Нормоконтроль	<u>Бойко І. В.</u> (підпис)	<u>Бойко І. В.</u> (прізвище та ініціали)
Завідувач кафедри	<u>Петрик М. Р.</u> (підпис)	<u>Петрик М. Р.</u> (прізвище та ініціали)
Рецензент	<u>Тиш Є. В.</u> (підпис)	<u>Тиш Є. В.</u> (прізвище та ініціали)

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ
ІВАНА ПУЛЮЯ

Факультет комп'ютерно-інформаційних систем та програмної інженерії
Кафедра Програмної інженерії
Рівень вищої освіти - другий (магістерський)
Спеціальність 121-Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« ____ » _____ 2020 р.

ЗАВДАННЯ
НА АТЕСТАЦІЙНУ РОБОТУ

студентові Віглінському Олександрю Юрійовичу

1. Тема роботи Інтеграція месенджерів в середовище ATutor. Експорт повідомлень скриньки і чату месенджер.

затверджена наказом університету від “ ____ ” _____ 2020 р. № _____

заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії

“ ____ ” _____ 2020 р.

3. Вихідні дані до роботи алгоритми шифрування повідомлень, алгоритми обробки миттєвих повідомлень, середовище об'єктно-орієнтованого проектування, середовище LMS Atutor. Використовувати ОС Windows.

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, огляд методів обміну миттєвими повідомленнями, методи забезпечення безпечності месенджера та шифрування повідомлень, методи імпорту та експорту повідомлень у найпоширеніших форматах.

5. Консультанти розділів роботи

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Охорона праці			
Безпека в надзвичайних ситуаціях			

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка *
1.	Аналіз предметної галузі	23 серпня 2020р.	
2.	Огляд існуючих методів	27 вересень 2020р.	
3.	Основна частина	15 листопад 2020р.	
4.	Підготовка пояснювальної записки	25 листопад 2020р.	
5.	Спецчастина	26 листопад 2020р.	
6.	Підготовка презентації та доповіді	28 листопад 2020р.	
7.	Попередній захист	30 листопад 2020р.	
8.	Нормоконтроль, рецензування	02 грудень 2020р.	
9.	Занесення диплома в електронний архів	03 грудень 2020р.	
10.	Допуск до захисту у зав. кафедри	04 грудень 2020р.	
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання _____ 2020 р.

Студент _____
(підпис)Керівник роботи _____
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить: 71 с., 23 рис., 22 джер.

ДИСТАНЦІЙНЕ НАВЧАННЯ, КОМУНІКАЦІЯ, МЕСЕНДЖЕР, МОДУЛЬ, НАВЧАЛЬНИЙ ПРОЦЕС, ATUTOR, JAVASCRIPT, JSXС, OPENFIRE, PHP, XMPP ТЕХНОЛОГІЯ.

Метою і завданням атестаційної роботи є розробка та інтеграція у існуюче середовище дистанційного навчання зручний, багатофункціональний і надійний спосіб обміну повідомленнями між студентами, викладачами та користувачами електронного навчання загалом.

Методи розробки базуються на інструментах розробки веб-застосувань на платформі PHP, системі керування навчанням (LMS) ATutor, протоколі передачі даних HTTP, XMPP-сервера Openfire, веб-клієнті JSXС.

В результаті роботи розглянуто методи комунікації, обґрунтовано доцільність і необхідність розробки та аргументовано користь практичного застосування. Методикою досліджень було обрано пошук, тестування та порівняння наявних способів спілкування у визначеному середовищі. Результати тестів, розбору варіантів використання різних програм і визначення їхніх переваг та недоліків висвітлено. Було розроблено програмну реалізацію модуля месенджера для системи контролю навчанням ATutor.

DISTANCE LEARNING, COMMUNICATION, MESSENGER, MODULE, LEARNING PROCESS, ATUTOR, JAVASCRIPT, JSXС, OPENFIRE, PHP, XMPP TECHNOLOGY.

The purpose and task of the certification work is to develop and integrate into the existing environment of distance learning a convenient, multifunctional and reliable way of exchanging messages between students, teachers and users of e-learning in general.

Development methods are based on web application development tools on the PHP platform, the ATutor learning management system (LMS), the HTTP data transfer protocol, the Openfire XMPP server, and the JSXС web client.

As a result of the work the methods of communication are considered, the expediency and necessity of development are substantiated and the benefit of practical application is argued. The research methodology was used to search, test and compare the available ways of communication in a certain environment. The results of tests, analysis of options for using different programs and determining their advantages and disadvantages are highlighted. A software implementation of the messenger module for the ATutor learning management system was developed.

ЗМІСТ

Вступ.....	7
1 Формулювання науково-технічної проблеми та аналіз стану вирішення поставленої проблеми. Дослідження в області існуючих способів електронної комунікації.....	9
1.1 Науково-технічна проблема	9
1.2 Стан вирішення проблеми комунікації у межах середовища ATutor	10
1.3 Дослідження існуючих способів електронної комунікації у межах середовища ATutor	11
2 Огляд існуючих технологій та інструментів розробки і вивчення способів їх використання та реалізації у поставленій задачі. Обґрунтування вибору методів та інструментів розробки програмного забезпечення	15
2.1 Сучасні методи та інструменти програмування	15
2.2 Опис технології, етапів розробки та самої розробленої програмної системи	26
3 Технічні характеристики розробленої програмної системи. Інструкції щодо її інтеграції в середовище ATutor та приклади експлуатації	30
3.1 Системні вимоги	30
3.1.1 Системні вимоги модуля Messenger	30
3.1.2 Системні вимоги системи управління навчанням ATutor v. 2.2.4	31
3.1.3 Системні вимоги XMPP сервера Openfire v. 4.6.0	32
3.1.4 Системні вимоги плагіна веб-клієнта JSXС v. 4.1.1	33
3.2 Інструкція по встановленню	33
3.2.1 Інструкція по встановленню системи управління навчанням ATutor	34
3.2.2 Інструкція по встановленню модуля Messenger	37
3.2.3 Інструкція по встановленню XMPP сервера Openfire	41
3.2.4 Інструкція по встановленню плагіна веб-клієнта JSXС	45
3.3 Інструкція по використанню	46
4 Охорона праці та безпека в надзвичайних ситуаціях	52

4.1 Охорона праці	52
4.2 Безпека в надзвичайних ситуаціях	54
Висновки	60
Перелік джерел посилання	62
Додаток А Технічне завдання	65
Додаток Б Копія тез доповіді	70
Додаток В Диск з електронними матеріалами	71

ВСТУП

На даному етапі розвитку сучасних технологій та технологічного прогресу досить важко переоцінити роль комунікації і зв'язків між суб'єктами у будь-якій сфері діяльності сучасного суспільства. Найбільшу частку усього обміну даними та інформацією становить електронна комунікація та спілкування через онлайн сервіси, файлообмінники, месенджери, чати, соціальні мережі та електронні пошти. Цей вид спілкування став невід'ємною частиною повсякденного життя, навчального процесу, бізнес процесів, ділового спілкування, наукової діяльності, засобів масової інформації, та іншого. Саме тому розвиток і оновлення такого поширеного і важливого аспекту життєдіяльності є актуальною темою і важливим об'єктом досліджень сьогодення. Інтернет-комунікації найбільшою мірою досліджені у працях українського вченого С. Г. Чемеркіна [1][2].

Враховуючи те, що Тернопільський національний технічний університет імені Івана Пулюя використовує сайт дистанційного навчання [3] як невід'ємний і обов'язковий засіб навчального процесу, можна стверджувати, що навчальний процес та обмін інформацією в університеті в значній мірі відбувається через інтернет-комунікації. Згідно з інформаційною сторінкою сайту дистанційного навчання ТНТУ імені Івана Пулюя [4] сайт розроблений на базі системи управління навчанням (LMS) ATutor, що дає змогу розширити можливості роботи з студентами та керувати курсами і навчальними матеріалами. Як вказано у документації ATutor [5], система має вбудований модуль поштової скриньки для обміну повідомленнями, але він не є надто функціональним та не відповідає сучасним вимогам якісного навчання студентів. Саме тому темою атестаційної роботи було обрано інтеграцію месенджера у середовище ATutor.

Метою і завданням атестаційної роботи є розробка та інтеграція у існуюче середовище дистанційного навчання зручний, багатофункціональний і надійний спосіб обміну повідомленнями між студентами, викладачами та користувачами електронного навчання загалом. Розширення функціоналу дистанційного навчання

можливе завдяки модульній архітектурі системи управління навчанням ATutor. Отже, бажаний кінцевий результат розробки це окремий модуль месенджера з можливістю експорту повідомлень, який можна буде налаштувати та інтегрувати у середовище ATutor.

Потреба замінити стару поштову скриньку на новий месенджер виникла вже давно, так як недоліки застарілої поштової скриньки дистанційного навчання очевидні а її використання сприймається студентами як «крайній випадок», коли інші варіанти комунікації недоступні або немає контактів реципієнта. Основними перевагами використання зручнішого і якіснішого способу комунікації між користувачами є:

- підвищення продуктивності студентів;
- швидкий обмін інформацією та розсилка навчального матеріалу і важливих повідомлень;
- доступне навчання;
- безпечність та краща захищеність особистих даних;
- зворотний зв'язок з викладачами.

Покращення якості і доступності дистанційного навчання має першочерговий рівень важливості так як це є необхідним і єдиним можливим інструментом навчання під час надзвичайних ситуацій, карантину, навчання на канікулах, а також заочному та самостійному вивченні матеріалу.

Об'єктом дослідження є розвиток і поточний стан засобів і способів обміну інформацією та спілкування у середовищі системи контролю навчанням ATutor.

Предметом дослідження є покращення засобів і способів комунікації, обміну даними та інформації, спілкування та впровадження нових функціональних можливостей у середовище системи контролю навчанням ATutor.

Основними методами дослідження є аналіз, тестування, випробовування та порівняння різних можливих і потенціальних систем інтернет-комунікації, що існують на даний момент у середовищі дистанційного навчання університету.

1 ФОРМУЛЮВАННЯ НАУКОВО-ТЕХНІЧНОЇ ПРОБЛЕМИ ТА АНАЛІЗ СТАНУ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ПРОБЛЕМИ. ДОСЛІДЖЕННЯ В ОБЛАСТІ ІСНУЮЧИХ СПОСОБІВ ЕЛЕКТРОННОЇ КОМУНІКАЦІЇ

1.1 Науково-технічна проблема

Задля кращого розуміння науково-технічної проблеми сучасних комунікацій слід розглянути і проаналізувати популярні варіанти і засоби спілкування, що використовуються у навчальних процесах університету, так як це є найоб'єктивнішим показником і оцінкою теперішньої ситуації і положення справ у даному питанні. Як серед студентів так і серед викладацького складу, найбільш поширеними способами комунікації є окремі сервіси, соціальні мережі та месенджери, не дивлячись на те що існує повністю функціональне і технологічне дистанційне навчання з усіма необхідними можливостями для якісної підготовки спеціалістів. Це свідчить про те, що на сайті дистанційного навчання відсутні доступні та/або присутні непридатні інструменти спілкування, що і призводить до розділення підготовчого процесу на суто навчальну діяльність, коли використовується обов'язкова і безальтернативна «дистанційка» (проходження тестів, завантаження матеріалів тощо) і діяльності, яка проходить поза межами дистанційного, і остання становить значно більшу частку від загального.

Науково-технічним завданням є аналіз та дослідження інтернет-комунікації як загалом, так і окремих її чинників і складових, знаходження оптимальних шляхів вирішення проблем, виявлених у процесі досліджень, проведення пошуку і підбору придатних технологій, що задовільняли би вимоги поставленої задачі, розробка, впровадження та інтеграція новітнього, альтернативного способу спілкування та обміну інформацією, а також загальна оптимізація системи в цілому.

Очевидною практичною цінністю нових технічних рішень є покращення і розвиток навчання в університеті та виведення його на новий сучасний рівень, що відповідав би вимогам розвинутого суспільства.

1.2 Стан вирішення проблеми комунікації у межах середовища ATutor

На даний момент в середовищі системи контролю навчанням (LMS) ATutor основним засобом комунікації являється вбудований модуль поштової скриньки, за допомогою якого можна відправляти листи усім користувачам дистанційного навчання. Поштова скринька хоча і є дуже корисною, але вона не дає можливості отримувати негайного результату, відсутня динаміка спілкування, яка притаманна спілкуванню у реальному часі. Надіславши лист, використовуючи даний метод, залишається невідомим для відправника питання чи досягнув лист свого призначення, чи був прочитаний лист, чи реціпієнт активний на даний момент або взагалі чи був він онлайн останнім часом. Невідомо як довго триватиме очікування відповіді. Коротко це можна назвати відсутністю зворотного зв'язку. При потребі вирішення негайних проблем або отримання своєчасної інформації це стає суттєвою проблемою.

Існує і стандартний модуль Chat доступний для користування відразу після встановлення системи, за допомогою якого можна спілкуватися миттєвими повідомленнями. Нажаль, він зовсім не придатний для користування і навіть не використовується на дистанційному навчанні. Характеризуючи його, можна сказати, що він є нефункціональним, нетехнологічним, неповноцінним, «сирим», з відсутністю мінімально прийнятного інтерфейсу користувача, незручним, неінтуїтивним, застарілим і вкінці кінців негарним. Його недоліки очевидні.

Отже, навіть при наявності цих двох варіантів, залишається багато невирішених проблем і не задоволених потреб користувачів. Головна з них це створення і впровадження комфортного, приємного, зрозумілого і миттєвого модуля переписування в реальному часі у систему керування навчанням ATutor. Саме вирішення даної проблеми і є основним завданням цієї атестаційної роботи.

1.3 Дослідження існуючих способів електронної комунікації у межах середовища ATutor

Основним методом даних досліджень було обрано пошук, тестування та порівняння наявних додаткових зовнішніх сервісів, чатів та месенджерів які дають змогу спілкуватися у середовищі LMS ATutor. У цьому підрозділі продемонстровано результати тестування та використання альтернативних програм і проаналізовано, висвітлено та обгрунтовано їхні переваги та недоліки.

У доступі є кілька готових варіантів, а саме додаткових модулів чатів і месенджерів для ATutor, що могли би допомогти вирішити поставлену проблему, але за тих чи інших обставин цього зробити не можуть. Першим з них є XMPP Chat 2.0, у описі до якого вказано, що це нова версія чату, базованого на протоколі XMPP та на технології інтерактивних областей WAI-ARIA, які забезпечують ефективніший обмін даними, більший набір функціоналу, доступний інтерфейс. Він включає переписки один-до-одного та багатокористувацькі чати між членами курсів та керівного складу курсів, керування підписками, збереження історії переписок як для приватних чатів так і для групових, надійну систему безпеки на зовнішньому XMPP сервері, підтримку офлайн повідомлень, шифрування повідомлень, інтерактивні області WAI-ARIA і дружній користувацький інтерфейс (детектор початку набору тексту, динамічне підвантаження контенту, підсвітка посилань, тощо).

При першому використанні XMPP Chat 2.0 приходиться запит на створення безкоштовного аккаунта на сервері talkr.im [6] так як його використовує модуль для обробки і відправки повідомлень. Після створення аккаунта стає можливим авторизація з подальшим автоматичним збереженням введених авторизаційних даних (Рисунок 1.1).

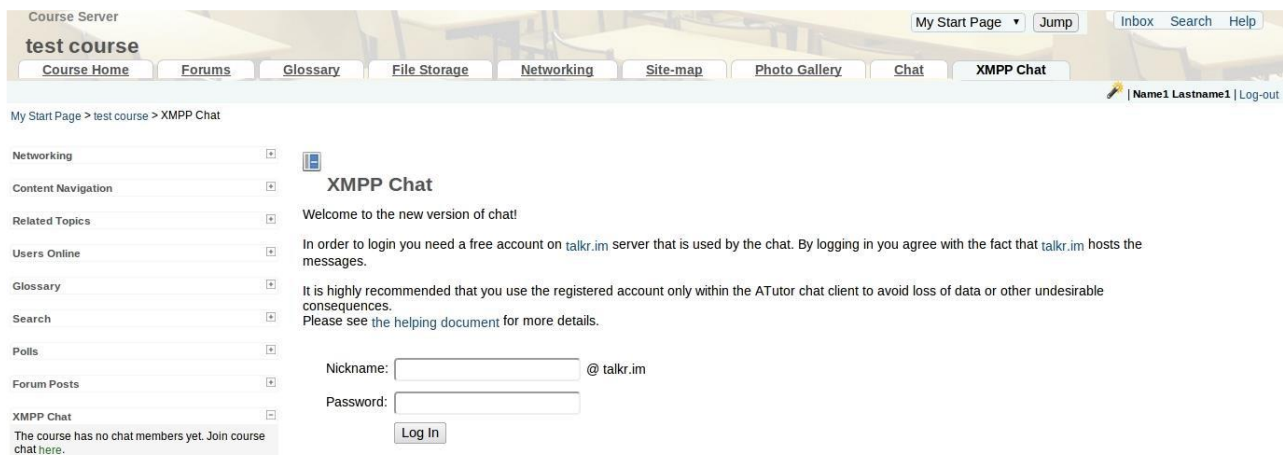


Рисунок 1.1 – Початок роботи з XMPP Chat 2.0

Задля уникнення небажаних наслідків і втрати даних рекомендується використовувати зареєстрований акаунт тільки в межах клієнту цього чату. Сам XMPP Chat створений таким чином, що назва профілю talkr.im повністю прихована а інші користувачі бачать лише імена вказані при логінуванні на курсі.

Як тільки виконано вхід і є хоча б ще один учасник, можна розпочати приватне листування, а якщо є хоча б ще два учасника, можна розпочати групову переписку MUC (multi-user conference). Якщо існують інші залоговані користувачі, то вони відобразатимуться у списку в боковій панелі курсу. Для старту приватної переписки необхідно просто натиснути на блок з бокової панелі і відкриється відповідна вкладка (Рисунок 1.2).

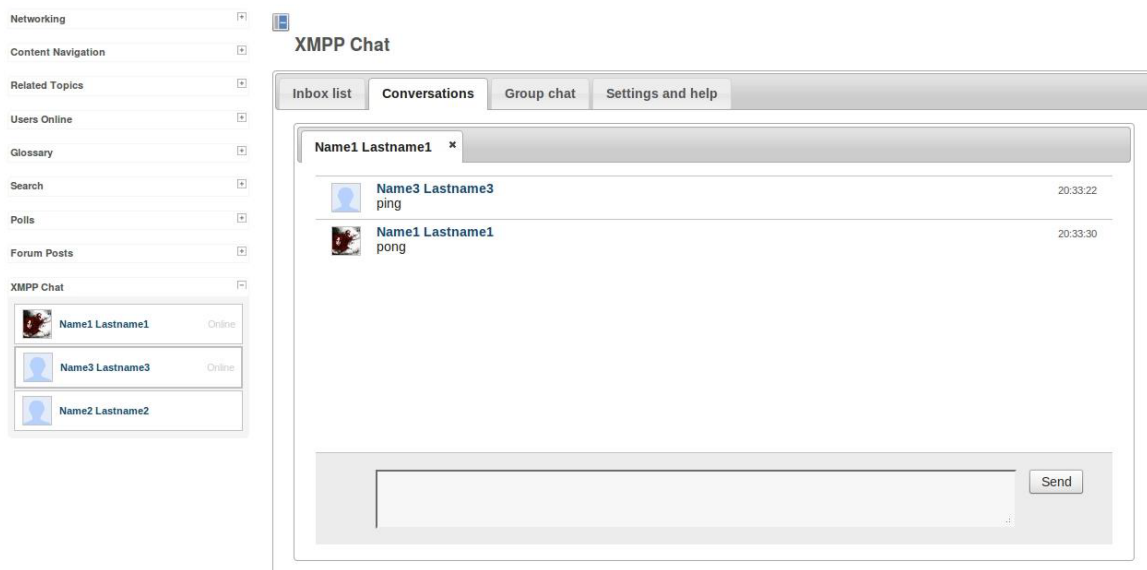


Рисунок 1.2 – Приватні переписки в чаті

Вкладка «Групові чати» використовується для створення нових групових чатів. Щоб це зробити потрібно вибрати учасників і вказати назву групи (Рисунок 1.3).

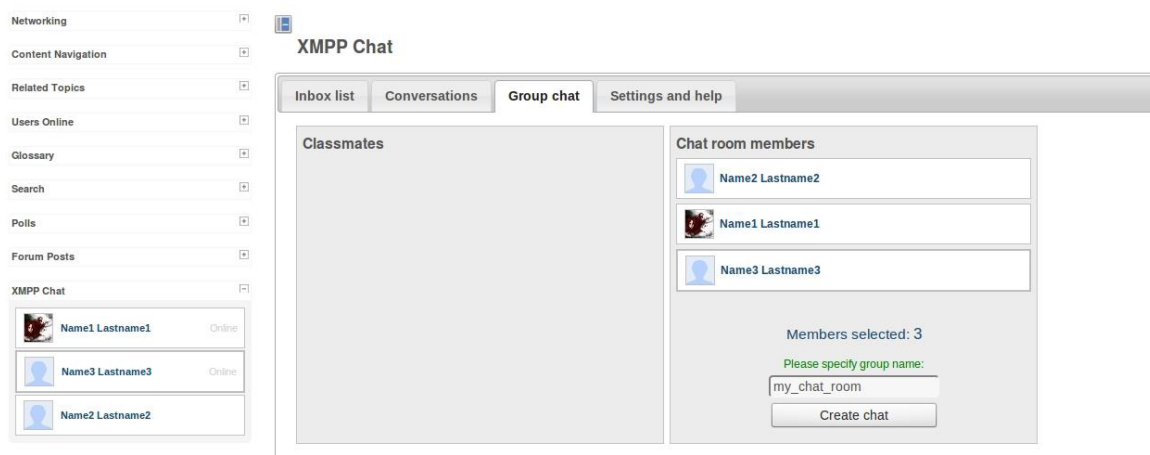


Рисунок 1.3 – Групові переписки в чаті

Створені групи є приватними, тобто люди що не були запрошені до них не зможуть приєднатися до обговорень, бачити учасників чи читати повідомлення.

Даний додаток є досить хорошим варіантом, проте у переліку модулів на офіційному сайті ATutor [7] вказано, що цей застосунок знаходиться на етапі розробки альфа, а останнє оновлення було 5 березня 2014 року. Від того часу пройшло більше шести років, з чого можна зробити висновок, що модуль більше не підтримується і не буде оновлюватися далі. XMPP сервер на якому базується програма більше не працює, а посилання на нього більше не активне. На теперішній час модуль не працює. З цього слідує, що для використання XMPP потрібно використовувати інший сервер або розгорнути власний.

У подібній ситуації знаходиться і наступний додаток, що буде розглянуто. Userplane Web Chat – модуль для ATutor, що надає користувачам можливість комунікувати між собою через текстові чати, або через аудіо/відео конференції. Включає в себе бокову панель для швидкого доступу з будь-якої сторінки курсу. Може бути активованим як інструмент користувача.

Усі користувачі (інструктори, студенти) мають доступ до кімнати Загального Обговорення, перейти на яку можна натиснувши на вкладку Userplane поза межами курсів.

У межах будь-якого курсу доступні три вкладки:

- Перша вкладка дає доступ до кімнати обговорення поточного курсу;
- Друга вкладка дозволяє використовувати попередню версію Userplane;
- Третя вкладка відповідає за Групові Чати. Будь-який студент може приєднатися до бесіди і почати обговорення з іншими учасниками чату.

Детальніший огляд даного модуля позбавлений смислу, бо ця система більше не підтримується, останнє оновлення було проведене 25 жовтня 2010 року, а офіційний сайт, вказаний у описі [8], вже довгий час знаходиться в режимі технічного обслуговування (не доступний).

Решта модулів і додатків необхідної функціональності недоступні, не підтримуються, застарілі або знаходяться у ще гіршому стані. Їхній розгляд та аналіз ніяк не вплине на прогрес дослідження та не дасть жодних результатів.

Дослідження показало, що, на сьогоднішній день, у вільному доступі не існує альтернативних систем, що задовільняли би потребу у швидкому, доступному та зручному спілкування в межах середовища управління навчанням ATutor, а також підтвердило доцільність розробки такої системи.

2 ОГЛЯД ІСНУЮЧИХ ТЕХНОЛОГІЙ ТА ІНСТРУМЕНТІВ РОЗРОБКИ І ВИВЧЕННЯ СПОСОБІВ ЇХ ВИКОРИСТАННЯ ТА РЕАЛІЗАЦІЇ У ПОСТАВЛЕНІЙ ЗАДАЧІ. ОБГРУНТУВАННЯ ВИБОРУ МЕТОДІВ ТА ІНСТРУМЕНТІВ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Сучасні методи та інструменти програмування

На сьогодні існує безліч можливих методів, технологій, інструментів, програм для розробки програмного забезпечення і способи їх застосування та використання варіюються в залежності від поставленої задачі та очікуваного результату. Теоретичне дослідження основних з них, що підходять по своїх особливостях до вирішення існуючої проблеми, дало змогу підібрати найоптимальніші варіанти методів та технологій і розробити надійний, безпечний, функціональний та якісний продукт.

На основі вже проведених досліджень було визначено можливості і переваги існуючих систем, разом з тим виявлено їхні недоробки та недоліки. Очевидно, що у розробленій програмній системі мають бути відсутні недоліки наявних засобів онлайн комунікації, а її функціональність має не поступатися функціональності доступних застосунків.

Так як Learning management system ATutor написаний на PHP, то вибір мови програмування очевидний, а напрям розглянутих технологій буде пов'язаний з цією саме з цією мовою.

Говорячи про PHP неможливо не розказати про ООП (Об'єктно спрямоване програмування) – ідеологія програмування, у якій головною сутністю під час розробки являється Об'єкт, який в свою чергу є представленням узагальненого поняття Клас. Базовими принципами ООП є:

- Поліморфізм – реалізація подібних задач одним і тим самим способом;
- Інкапсуляція – розмежування прав доступів до об'єктів і класів;
- Наслідування – здібність сутностей набувати тих самих особливостей, якими володіють сутності, що їх породили. Визначає ієрархію класів;

- Абстракція – узагальнення використання сутностей.

Перш за все, варто розглянути сучасні принципи програмування, дотримання яких, рекомендовано при розробці будь-якої системи. Їхня користь є беззаперечною, а їх слідування вважається показником якості як програмного продукту так і розробника, який розроблює це програмне забезпечення.

У контексті обговорення про принципи програмування варто зазначити, що питання стоїть не у тому, що досягається при дотримванні цих принципів під час програмування, а у тому, що втрачається при їх недотриманні. А можливі втрати значущі. У першу чергу падає якість програмного коду, його читабельність та підтримка. У другу чергу падає ефективність системи в цілому: витрачається більше ресурсів, збільшується час обробки запитів, з'являється надмірне використання пам'яті, зменшується стресостійкість програми. А найголовнішим ризиком є збільшення вразливості системи, зменшення безпечності, можлива поява непередбачуваних наслідків через неточності і необережності.

Отже, основним з них є правило SOLID.

Літера І розшифровується як Інтерфейсний Поділ. Це правило, яке відноситься до напрямку формування тексту програми. Він в основному стосується статичних мов програмування. Програмісти, що використовують мови девелопменту де немає інтерфейсів, майже не будуть стикатися із використанням даного правила на практиці. Але загальна обізнаність і розуміння цього правила в все одно сприяє покращенню якості написання коду.

Інтерфейси сприяють швидшій роботі з побудовою структури, а не з самими даними. Коректна їх реалізація дає хорошу можливість покращити програмний код.

Правило інтерфейсного поділу охоплює напрям їх використання для ізоляції програмних текстів і організації формування інтерфейсів. Як приклад, спадкування класів. Якщо властивості класу, що є вище по ієрархії, не дуже важливі і є потреба знехтувати ними, то нескладний інтерфейс у даному випадку це зробить.

О – Правило відкритості-закритості.

Дотримання цього правила вимагає, щоб програмний текст був доступний для подальших доповнень, і щоб при їх додаванні не приходилося редагувати вже

існуючий код. Це правило в основному мірою про питань побудови архітектури, ніж самого коду.

Дотримання Open-Closed не означає, необхідність писати додатковий надлишковий код, який може в майбутньому пригодитися. Мова йдеться про розробку архітектури програми і всієї кодової бази таким чином, щоб вони мали потенційну змогу розширюватися.

D – Правило інверсійності залежності. В основному стосується архітектури програмного тексту.

Правило Інверсійності залежності дещо складний до розуміння, але потрібно запам'ятати тільки два моменти. Перше, це код має бути написаний так, щоб деталі його конкретної реалізації залежали від базової логіки, а не наоборот. Друге, це всі залежності не мають бути прямими. Їх потрібно узагальнювати з використанням інтерфейсів, щоб головна логіка працювала з будь-якими сутностями, вимагаючи тільки звичайної визначного елемента.

S – Правило одиничної відповідальності.

Це правило потребує, щоб методи, класи і об'єкти потрібно розробляти так, щоб кожен з них відповідав тільки за один функціонал, дію чи сутність.

Відповідальність класів легше організувати, коли вони відображають сутності споріднені з реальними сутностями. Але ситуація стає складнішою коли іде мова про об'єкти, що мають у визначенні слова «менеджер» або «служба». Такі глибокорівневі модулі складно розробляти, так як в теорії, у них можна вкласти будь-який зміст. При цьому, кількість обов'язків таких сутностей швидко збільшується, а в результаті весь код стає складнішим і незрозумілішим. Майте на увазі, що це означає не тільки особливості, але й функції, з якими йде робота.

L – Liskov Principle (Правило Лісков). Це правило об'єктно-спрямованої розробки, що відноситься до класів, видів і підвидів.

Це правило дуже просте і логічне, але його складно використовувати на практиці новачкам. Його суть у тому, що кожен підвид повинен розширювати, а не замінювати базовий вид.

KISS.

Роби якомога простіше. Це правило велить слідкувати за тим, щоб код залишався якомога простішим. Чим код простіший, тим легше в ньому розібратися, як розробнику, так і іншим людям, які займаються його підтримкою. Під простотою головним чином мається на увазі відмова від використання хитромудрих прийомів і непотрібного ускладнення.

Як приклади порушення цього правило можна назвати написання окремої функції тільки для здійснення операції додавання або використання побітового оператора (right shift $\gg 1$) для розподілу цілих чисел на 2. Останнє, безумовно, ефективніше, ніж звичайне ($/ 2$), але при цьому дуже сильно знижується зрозумілість коду. Застосовуючи такий підхід, здійснюється clever coding (замудрений кодинг) і over-optimization (надмірна оптимізація). І те, і інше в довгостроковій перспективі не дуже добре позначається на якості коду.

YAGNI.

Тобі воно не пригодиться.

Це правило є дуже спірним у спільноті девелоперів. Готовність до майбутнього зазвичай вважається справою хорошою, але не в програмуванні. Залишати будь-який код, призначений тільки для розширюваності програми в майбутньому, це неправильно.

Проекти програмного забезпечення не мають чіткого кінця. Якщо тільки творець проекту не закидає свою ідею взагалі (і при цьому навіть не передає її комусь ще), проект, по суті, постійно розвивається. Але при цьому немає ніякої точки, де можна було б визнати проект «досить хорошим» і зупинитися. Завжди буде можливість щось покращити.

Намагатися спрогнозувати майбутнє і уявляти, як повинен виглядати код, це не погано. Але небажано залишати в продакшені точки розширення (місця, призначені тільки для того, щоб дозволити в майбутньому легко додати новий функціонал). Звичайно, ми не говоримо про випадки, коли мова йде про вже замовлений функціонал. Такі точки розширення вносять непотрібну складність і збільшують розмір кодової бази. Якщо задуматися, це також суперечить описаному вище правилу KISS.

DRY.

Не повторюйтеся. За змістом схоже на правило KISS.

Багатьом девелоперам доводиться робити копіювання і дублювання фрагментів власного коду. Загалом, в цьому немає нічого поганого. Часто часом виникає необхідність швидко перевірити що-небудь (очікувану поведінку або щось інше), щоб визначити, чи варто це писати. А ось випуск такого скопійованого коду в виробництво неприйнятний.

DRY нагадує, що кожен повторюваний фрагмент коду слід відокремлювати (наприклад, виділяти в окрему функцію) для можливості багаторазового використання. Коли у кодовій базі є два абсолютно однакові фрагменти коду, то це не добре. Це часто призводить до розсинхронізації та інших помилок, не кажучи вже про те, що від цього збільшується розмір програми.

SLAP.

Правило одного ступеню абстракції твердить, як повинен розробник організувати програмні тексти (зокрема, функції), щоб вони залишалися підтримуваними.

З довгими і складними функціями важко ужитися. У них складно розібратися непосвяченій людині, їх важко тестувати і, найчастіше, навіть щоб їх побачити повністю, доводиться користуватися прокруткою. Якщо виникають подібні проблеми, слід негайно замислитися про реструктуризацію та рефакторинг функції і створення замість неї декількох функцій поменше. Функції повинні виконувати тільки одну дію, але виконувати її добре.

Методи мають виконувати лише одну функцію, або якщо говорити у стилі цього правила, вони мають мати один рівень абстрактності. Наприклад, метод, що зчитує ввід, не має разом з тим обробляти цю інформацію. Для цього він має викликати інший метод, що визначений в окремому, зазвичай нижньому, ступені абстрактності. Чим узагальненішим є метод і чим більше окремих методів він викликає, тим вище він знаходиться у ієрархії абстракцій.

Шаблони проектування є невід'ємною частиною програмування для поважаючого себе програміста, тому не можливо не висвітлити цю тему тут.

Шаблони проектування описують типові способи вирішення поширених проблем при проектуванні програм. Шаблони відрізняються за рівнем складності, охоплення і деталізації проектованої системи. На додачу, їх можна поділити на три групи, щодо розв'язуваних проблем.

Шаблон проектування — це типовий спосіб вирішення певної проблеми, що часто зустрічається при проектуванні архітектури програм.

На відміну від готових функцій чи бібліотек, Шаблон не можна просто взяти й скопіювати в програму. Шаблон являє собою не якийсь конкретний код, а загальний принцип вирішення певної проблеми, який майже завжди треба підлаштовувати для потреб тієї чи іншої програми.

Шаблони часто плутають з алгоритмами, адже обидва поняття описують типові рішення відомих проблем. Але якщо алгоритм — це чіткий набір дій, то Шаблон — це високорівневий опис рішення, реалізація якого може відрізнитися у двох різних програмах.

Якщо провести аналогію, то алгоритм — це кулінарний рецепт з чіткими кроками, а Шаблон — інженерне креслення, на якому намальовано рішення без конкретних кроків його отримання.

Описи Шаблонів зазвичай дуже формальні й найчастіше складаються з таких пунктів:

- проблема, яку вирішує шаблон;
- мотивація щодо вирішення проблеми способом, який пропонує шаблон;
- структура класів, складових рішення;
- приклад однією з мов програмування;
- особливості реалізації в різних контекстах;
- зв'язки з іншими шаблонами.

Концепцію шаблонів вперше описав Крістофер Александер у книзі «Мова шаблонів. Міста. Будівлі. Будівництво» [9]. У книзі описано «мову» для проектування навколишнього середовища, одиниці якого — шаблони відповідають на архітектурні запитання: якої висоти потрібно зробити вікна, скільки поверхів має бути в будівлі, яку площу в мікрорайоні відвести для дерев та газонів.

Ідея видалася привабливою четвірці авторів: Еріху Гаммі, Річарду Хелму, Ральфу Джонсону, Джону Вліссідесу. У 1994 році вони написали книгу «Патерни проектування: повторно використовувані елементи архітектури об'єктно-орієнтованого програмного забезпечення» [10], до якої увійшли 23 шаблони, що вирішують різні проблеми об'єктно-орієнтованого дизайну. Назва книги була занадто довгою, щоб хтось зміг її запам'ятати. Тому незабаром усі стали називати її «book by the gang of four», тобто «книга від банди чотирьох», а потім і зовсім «GoF book».

З того часу було знайдено десятки інших об'єктних шаблонів. Шаблонний підхід став популярним і в інших галузях програмування, тому зараз можна зустріти різноманітні шаблони також за межами об'єктного проектування.

Шаблони відрізняються за рівнем складності, деталізації та охоплення проектованої системи. Проводячи аналогію з будівництвом, можна підвищити безпеку на перехресті, встановивши світлофор, а можна замінити перехрестя цілою автомобільною розв'язкою з підземними переходами.

Найбільш низькорівневі та прості шаблони — ідіоми. Вони не дуже універсальні, позаяк мають сенс лише в рамках однієї мови програмування.

Найбільш універсальні — архітектурні шаблони, які можна реалізувати практично будь-якою мовою. Вони потрібні для проектування всієї програми, а не окремих її елементів.

Крім цього, шаблони відрізняються і за призначенням. Тут буде розглянуто три основні групи шаблонів:

- Породжуючі піклуються про гнучке створення об'єктів без внесення в програму зайвих залежностей.
- Структурні показують різні способи побудови зв'язків між об'єктами.
- Поведінкові піклуються про ефективну комунікацію між об'єктами.

Породжувальні шаблони проектування.

Ці шаблони відповідають за зручне та безпечне створення нових об'єктів або навіть цілих сімейств об'єктів. Для прикладу буде наведено кілька з них.

Прототип — це породжувальний шаблон проектування, що дає змогу копіювати об'єкти, не вдаючись у подробиці їхньої реалізації. Шаблон Прототип доручає процес копіювання самим об'єктам, які треба скопіювати. Він вводить загальний інтерфейс для всіх об'єктів, що підтримують клонування. Це дозволяє копіювати об'єкти, не прив'язуючись до їхніх конкретних класів. Зазвичай такий інтерфейс має всього один метод — `clone`.

Реалізація цього методу в різних класах дуже схожа. Метод створює новий об'єкт поточного класу й копіює в нього значення всіх полів власного об'єкта. Таким чином можна скопіювати навіть приватні поля, оскільки більшість мов програмування дозволяє отримати доступ до приватних полів будь-якого об'єкта поточного класу.

Об'єкт, який копіюють, називається прототипом. Коли об'єкти програми містять сотні полів і тисячі можливих конфігурацій, прототипи можуть слугувати своєрідною альтернативою створенню підкласів.

Будівельник — це породжувальний шаблон проектування, що дає змогу створювати складні об'єкти крок за кроком. Будівельник дає можливість використовувати один і той самий код будівництва для отримання різних відображень об'єктів.

Шаблон Будівельник пропонує винести конструювання об'єкта за межі його власного класу, доручивши цю справу окремим об'єктам, які називаються будівельниками.

Шаблон пропонує розбити процес конструювання об'єкта на окремі кроки. Щоб створити об'єкт, потрібно по черзі викликати методи будівельника. До того ж не потрібно викликати всі кроки, а лише ті, що необхідні для виробництва об'єкта певної конфігурації.

Зазвичай один і той самий крок будівництва може відрізнитися для різних варіацій виготовлених об'єктів. Наприклад, дерев'яний будинок потребує будівництва стін з дерева, а кам'яний — з каменю.

У цьому випадку можна створити кілька класів будівельників, які по-різному виконуватимуть ті ж самі кроки. Використовуючи цих будівельників в одному й тому самому будівельному процесі, буде змога отримувати на виході різні об'єкти.

Абстрактна фабрика — це породжувальний шаблон проектування, що дає змогу створювати сімейства пов'язаних об'єктів, не прив'язуючись до конкретних класів створюваних об'єктів.

Для початку, шаблон Абстрактна фабрика пропонує виділити загальні інтерфейси для окремих продуктів, що складають одне сімейство, і описати в них спільну для цих продуктів поведінку.

Клієнтський код повинен працювати як із фабриками, так і з продуктами тільки через їхні загальні інтерфейси. Це дозволить подавати у класи будь-які типи фабрик і виробляти будь-які типи продуктів, без необхідності вносити зміни в існуючий код.

Структурні шаблони відповідають за побудову зручних в підтримці ієрархій класів.

Легковаговик — це структурний шаблон проектування, що дає змогу вмістити більшу кількість об'єктів у відведеній оперативній пам'яті. Легковаговик заощаджує пам'ять, розподіляючи спільний стан об'єктів між собою, замість зберігання однакових даних у кожному об'єкті.

Шаблон Легковаговик пропонує не зберігати зовнішній стан у класі, а передавати його до тих чи інших методів через параметри. Таким чином, одні і ті самі об'єкти можна буде повторно використовувати в різних контекстах. Головна ж перевага в тому, що тепер знадобиться набагато менше об'єктів, адже вони тепер відрізнятимуться тільки внутрішнім станом, а він не має так багато варіацій.

Декоратор — це структурний шаблон проектування, що дає змогу динамічно додавати об'єктам нову функціональність, загортаючи їх у корисні «обгортки».

Спадкування — це перше, що приходить в голову багатьом програмістам, коли потрібно розширити яку-небудь чинну поведінку. Проте механізм спадкування має кілька прикрих проблем.

Він статичний. Не можна змінити поведінку об'єкта, який вже існує. Для цього необхідно створити новий об'єкт, вибравши інший підклас.

Він не дозволяє наслідувати поведінку декількох класів одночасно. Тому доведеться створювати безліч підкласів-комбінацій, щоб досягти поєднання поведінки.

Одним зі способів, що дозволяє обійти ці проблеми, є заміна спадкування агрегацією або композицією.

Поведінкові шаблони вирішують завдання ефективної та безпечної взаємодії між об'єктами програми.

Стан — це поведінковий шаблон проектування, що дає змогу об'єктам змінювати поведінку в залежності від їхнього стану. Ззовні створюється враження, ніби змінився клас об'єкта.

Шаблон Стан пропонує створити окремі класи для кожного стану, в якому може перебувати контекстний об'єкт, а потім винести туди поведінки, що відповідають цим станам.

Замість того, щоб зберігати код всіх станів, початковий об'єкт, який зветься контекстом, міститиме посилання на один з об'єктів-станів і делегуватиме йому роботу в залежності від стану.

Завдяки тому, що об'єкти станів матимуть спільний інтерфейс, контекст зможе делегувати роботу стану, не прив'язуючись до його класу. Поведінку контексту можна буде змінити в будь-який момент, підключивши до нього інший об'єкт-стан.

Дуже важливим нюансом, який відрізняє цей шаблон від Стратегії, є те, що і контекст, і конкретні стани можуть знати один про одного та ініціювати переходи від одного стану до іншого.

Посередник — це поведінковий шаблон проектування, що дає змогу зменшити зв'язаність великої кількості класів між собою, завдяки переміщенню цих зв'язків до одного класу-посередника.

Шаблон Посередник змушує об'єкти спілкуватися через окремий об'єкт-посередник, який знає, кому потрібно перенаправити той або інший запит. Завдяки

цьому компоненти системи залежатимуть тільки від посередника, а не від десятків інших компонентів.

Щоб зробити код ще гнучкішим, можна виділити єдиний інтерфейс для всіх посередників, тобто діалогів програми. Наша кнопка стане залежною не від конкретного діалогу створення користувача, а від абстрактного, що дозволить використовувати її і в інших діалогах.

Таким чином, посередник приховує у собі всі складні зв'язки й залежності між класами окремих компонентів програми. А чим менше зв'язків мають класи, тим простіше їх змінювати, розширювати й повторно використовувати.

XMPP це відкритий стандарт для обміну повідомленнями, який нікому не належить. XMPP безкоштовний і доступний кожному з 1999 року. Цей стандарт активно розвивається та розширюється. Програми основані на ньому щоденно використовуються мільйонами для зв'язку між людьми та сервісами. На цьому стандарті базуються сервери, клієнти та окремі бібліотеки. Далі буде детальніше розібрано про кожен з цих пунктів.

XMPP сервери забезпечують функціональність базової переписки і можливості XML перенаправлення. У мережі інтернет існують безліч готових безкоштовних варіантів їх реалізації. Однією з головних особливостей таких серверів є масштабованість та розширюваність.

XMPP клієнти це програмні продукти або застосунки, що дають змогу під'єднуватися до XMPP серверів для отримання можливості комунікувати за допомогою миттєвих повідомлень з будь-якою кількістю людей на будь-якій відстані один від одного, використовуючи мережу інтернет. У вільному доступі є безліч готових до використання цих клієнтів для різних потреб та для різних гаджетів і систем.

XMPP бібліотеки з програмним кодом доступні для великої кількості мов розробки програмного забезпечення. Вони дають змогу девелоперам будувати свої системи багатфункціональними та насиченими та значно спрощують їм життя.

Сильними сторонами XMPP технології є:

- Розцентралізованість (будь-хто може розгорнути свій сервер і почати з'єднуватися з іншими);

- Вільна і доступна стандартизованість (технологію було формалізовано і потрібно особливих прав доступу чи дозволів для реалізації визначених специфікацій);

- Витримка часом: Ця технологія існує і успішно працює ще з 1999 року. З того часу з'явилося дуже багато реалізацій і наробок на основі специфікацій цієї технології;

- Надійність: У серверів є низка корисних налаштувань, що забезпечують їхню надійність та безпечність. Можна налаштувати доступність для інших серверів та користувачів цих серверів, можна увімкнути обов'язкове шифрування, вимкнути реєстрацію небажаних гостей;

- Еластичність: системи, розроблені на базі цієї технології, можна з легкістю кастомізувати в межах визначених специфікацій. Можливо «прикрутити» власний функціонал, встановлювати плагіни, розширення, додатки, бібліотеки.

2.2 Опис технології, етапів розробки та самої розробленої програмної системи

Для задоволення визначених вимог і потреб месенджера, базуючись на проведених дослідженнях, було проаналізовано та обрано найбільш доцільні і оптимальні методи, застосунки, інструменти програмування. Тому для розробки було використано XMPP server Openfire, плагіни JSXС (веб-клієнт), Monitoring (логування повідомлень), та Registration. Звісно, розроблялася система на мові програмування PHP.

Особливості розробленої системи:

Доступність: Система була розроблена з доступністю у пріоритеті. Широкий вибір функціоналі забезпечує дружню технологічну підтримку для користувачів.

Соціалізація: Користувачі системи можуть розгорнути мережу контактів, об'єднуватися у групи, зв'язувати тисячі гаджетів разом для спільного спілкування.

Безпечність: Усі дані шифруються з допомогою новітніх алгоритмів. Забуті логіни і паролі можуть бути відновлені всього за пару кліків. Підвищений захист до перехоплення даних.

Пошта/Переписки: Усі користувачі системи мають змогу користуватися поштою, через яку вони можуть відправляти та отримувати приватні повідомлення від інших юзерів. Історія переписок не видаляється а зберігається, а самі повідомлення можуть бути експортовані.

Адаптивна навігація: Учні можуть переміщуватися по системі за допомогою глобальних ієрархічних інструментів навігації, які опціонально можна приховати для спрощення середовища.

Робочі групи: Учні та викладачі можуть об'єднуватися у окремі робочі групи, а також співпрацювати з іншими робочими групами та користувачами, комунікуючи через форму, месенджер, пошту та чати і обмінюючись необхідними матеріалами.

Блоги груп: Кожна група має доступ до свого власного блогу, у який вони можуть публікувати статті, повідомлення, і які потім можуть переглядати учасники цієї групи і гості.

Система відгуків: Система відгуків допомагає покращувати як систему загалом, так і кожен групу чи курс окремо. Існує можливість скарг на неприйнятні або непристойні аккаунти, повідомлення, групи та блоги.

Налаштування «під себе»: Учні можуть повністю налаштовувати систему під свої потреби та смаки для ефективного навчання і підготовки. Вмикати/вимикати функціональні можливості системи та кастомізувати зовнішній вигляд інтерфейсу.

Соціальна мережа: Інструктори мають у доступі усі необхідні інструменти та можливості для організації ефективної і зручної комунікації між учасниками курсів.

Контроль доступу гостей до курсів: навіть неавторизовані користувачі (гості) можуть отримати доступ до певних курсів, якщо це передбачено налаштуваннями курсу, Є можливість запрошення нових людей на курс за посиланням.

Налаштувабельні інструменти курсу: так само як і для кожного окремого користувача, для кожного окремого курсу можна налаштувати усі можливі показники і параметри, включаючи функціональні можливості, права доступу і закінчуючи зовнішнім виглядом інтерфейсу.

Сторінка керування курсами: Усі необхідні елементи можуть бути доступні з-під вкладки «Керування».

Ведення статистики: Статистика записується як загалом, так і окремо, для кожного курсу.

Менеджер робочих груп: Інструктори можуть створювати вручну або генерувати автоматично робочі групи для низки завдань.

Файлове сховище: Сервер включає в себе розширюване файлове сховище для збереження приватних файлів і об'єктів користувачів системи.

Редактор: Інструктори можуть змінювати наповнення своїх курсів як за допомогою html тексту та і просто звичайним текстом, що надає ще більше варіантів кастомізації курсів і системи в цілому.

Менеджер резервного копіювання: Існує функціональна можливість налаштувати створення резервної копії даних користувачів, груп або курсів вручну, а також створити регулярну задачу на створення резервних копій системи.

Новини і Оголошення: Крім публікацій у Блог інструктори мають можливість робити оголошення з надсиланням відповідних сповіщень усім, кому ці оголошення були адресовані. Публікація Новин на сторінці курсів також можлива.

Менеджер проведення тестів: Запуск і проведення тестів більше не проблема! Це все тепер може відбуватися автоматично, з мінімальним втручанням адміністраторів у роботу спеціальної програми.

FAQ: Зрозуміла і повнообсяжна сторінка поширених питань та відповідей допоможе навіть тим користувачам, які не змогли розібратися у інтуїтивному інтерфейсі користування системою.

Менеджер модулів: Адміністратори можуть легко розширювати функціонал, встановлюючи додаткові модулі автоматично, з підпором потрібних версій програм та плагінів.

Мульти-адміністрування: У системі можна налаштовувати адміністраторів найрізноманітнішими способами, з розділенням прав для кожного адміністратора.

Менеджер Тем: Теми по замовчуванню можна визначити для кожного окремого користувача, кожного окремого курсу та системи в цілому.

Автоматичне оновлення: Автоматичне оновлення дає змогу підтримувати завжди найновішу версію системи з виправленими помилками.

Cron: Можливість визначати дії, які потрібно виконувати з певним інтервалом. Доступна адміністраторам.

Широкий пакет мов: Великий вибір мови відображення системи значно розширює доступність системи.

3 ТЕХНІЧНІ ХАРАКТЕРИСТИКИ РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ. ІНСТРУКЦІЇ ЩОДО ЇЇ ІНТЕГРАЦІЇ В СЕРЕДОВИЩЕ ATUTOR ТА ПРИКЛАДИ ЕКСПЛУАТАЦІЇ

3.1 Системні вимоги

У цьому підрозділі розглянуто системні вимоги розробленого програмного забезпечення та усіх його складових, а також системні вимоги усіх програм, необхідних для повноцінної і безперебійної роботи месенджера. Роботоздатність месенджера гарантується лише при задоволенні системних вимог вказаних у цьому підрозділі.

3.1.1 Системні вимоги модуля Messenger

Ця версія Messenger сумісна з версією ATutor 2.2.3 і новіше. Програма використовує інструменти які доступні у всіх сучасних браузерях: JavaScript та підтримка зображень повинні бути увімкнуті. Вимоги до мови програмування PHP такі ж самі як і для ATutor – v. 7 +.

Системні вимоги:

- операційна система Windows 10;
- HTTP веб сервер (рекомендовано Apache);
- PHP v. 7 +;
- MySQL від 4.1.10 до 5.7;
- LMS ATutor v. 2.2.3 +;
- XMPP сервер Openfire v. 4.4.0 + з установленими плагінами: Monitoring service, JSXС, Registration.
- веб-клієнт JSXС v. 4.1.1 +.

На даний момент Messenger було успішно протестовано на таких операційних системах і платформах з використанням збірки веб сервера XAMPP:

- Linux Mint 12 (Ubuntu)

Browsers: Firefox 68.11.0, Chromium 71.0.1025.168, Opera 68.12.01;

Screen readers: Orca;

- Windows 10 64-bit

Browsers: Firefox 68.11.0, Google Chrome 87.0.4280.88 (Official Build) (64-bit), OperaGX 71.0.3770.449;

Screen readers: NVDA, JAWS.

Дана система використовує такі бібліотеки:

- jQuery: попередньо включена у системі контролю навчання по замовчуванню;

- jQuery UI: надає доступ до вкладок і діалогових вікон що використовуються у проекті;

- moment.js[11]: бібліотека JavaScript зчитування, оперування, маніпулювання і форматування дат; використовується для виводу всіх дат в модулі; MIT ліцензія.

3.1.2 Системні вимоги системи управління навчанням ATutor v. 2.2.4

Месенджер розроблявся для останньої стабільної версії ATutor на даний момент – v. 2.2.4. Системні вимоги вказано згідно з офіційним сайтом цієї LMS[12], де знаходяться документація, інструкції та додаткові матеріали.

Системні вимоги:

- операційна система Windows 10;

- HTTP веб сервер (рекомендовано Apache);

- PHP v. 7 + з такими розширеннями:

- mysql;

- zlib;
- curl;
- enable-mbstring;
- gd;

- MySQL від 4.1.10 до 5.7;

3.1.3 Системні вимоги XMPP сервера Openfire v. 4.6.0

Для розробки було обрано останню стабільну версію Openfire v. 4.6.0.

Системні вимоги вказано згідно з офіційним сайтом цього XMPP сервера[13], де знаходяться документація, гайди та додаткові плагіни і матеріали.

Вимоги до обладнання варіюються в залежності від орієнтованої кількості користувачів системи та навантаження, що вони будуть створювати.

Системні вимоги:

- Операційна система (одна з):

- Windows 10, Windows 7, Windows 2000, Windows XP, Windows Vista, Windows Server 2003, Windows Server 2008
- Linux, Solaris, або інакша Unix система з підтримкою Java 8
- Mac OS X

- База даних (одна з):

- MySQL (рекомендовано 5.x або новіше)
- Oracle (рекомендовано 10g або новіше)
- Postgres (рекомендовано 7.x або новіше)
- IBM DB2 (рекомендовано v7 або новіше)
- SQL Server (рекомендовано 2000 або новіше)
- Sybase
- HSQL DB (також використовується як вбудована база даних)

- Будь-яка інша база даних, що має стабільний сумісний драйвер JDBC 2.0

А також, в результаті того, що Openfire це додаток чисто написаний на Java, для запуску і користуванням сервером необхідно щоб на платформі було встановлено JRE 8 або новіше.

Крім цього, повинні бути встановлені такі плагіни:

- Registration – плагін для реєстрації нових користувачів;
- Monitoring Service – плагін для архівування повідомлень діалогів і бесід для подальшої можливості експорту;
- JSXС – веб-клієнт.

3.1.4 Системні вимоги плагіна веб-клієнта JSXС v. 4.1.1

У процесі розробки було використано останню сумісну і стабільну версію веб-клієнта JSXС v. 4.1.1.

Системні вимоги вказано відповідно з офіційним сайтом цього веб-клієнта [14], де знаходяться документація, варіанти використання, інформація для девелоперів та додаткові матеріали.

Системні вимоги:

- встановлений і запущений веб сервер;
- XMPP сервер з підтримкою BOSH.

3.2 Інструкція по встановленню

У цьому підрозділі описано та розглянуто інструкції по встановленню розробленої програмної системи та усіх її складових, а також інструкції по

встановленню усіх компонентів, необхідних для повноцінної і безперебійної роботи месенджера. Роботоздатність месенджера гарантується лише при дотриманні чіткого слідування та послідовності кроків інструкції по встановленню вказаних у цьому підрозділі.

3.2.1 Інструкція по встановленню системи управління навчанням ATutor

Усі інструкції по установці даної системи контролю навчанням вказані з відповідністю з офіційним сайтом цієї системи [15], а також до уваги було взято особливості розробленого месенджера.

Перш за все, перед початком установки даної системи потрібно переконатися що усі системні та технічні вимоги було задоволено, а усі компоненти сумісні між собою, рекомендовано використовувати версії програм і додатків такі ж що вказані у даній роботі.

Спочатку необхідно завантажити установочний файл програми з сторінки завантажень офіційного сайту [16], за посиланням [17], або ж використати установочний файл що представлено у даному розробленому проекті.

Установка системи на машину, що працює на базі Unix системи потребує додаткових знань і компетенції в питанні прав доступу до директорій та файлів, так як в подальшому буде проведений запит на створення папки з вмістом і надання для неї відповідних прав доступу для того, щоб веб сервер мав змогу записувати та зберігати дані. Якщо ж права на запис не надано, то установка завершиться помилково і виведе відповідні повідомлення з інструкціями як виправити помилки. Тоді на цьому моменті потрібно поставити відповідні дозволи на перелічені файли/директорії і запусити установку знову.

Після запуску на вступному екрані установщик перевіряє систему на виконання усіх вимог необхідних для запуску системи. На цьому етапі необхідно

переглянути відгук і зробити поправки, якщо такі потребуються, і коли усі вимоги задоволені, розпочати процес установки.

Далі потрібно відкрити браузер як вказано у процесі встановлення і пройти необхідні кроки налаштування встановленої програми. Наступні вісім кроків описують цей процес і послідовність дій:

1) Умови використання

Використання ATutor регулюється Загальною публічною ліцензією GNU (GPL). Погодження з цією ліцензією є обов'язковим щоб мати змогу модифікувати та розповсюджувати програму.

2) Налаштування бази даних

ATutor потребує бази даних MySQL для роботи. На цьому кроці потрібно вказати необхідні деталі для підключення до цієї бази даних. Також, тут можна вказати опціональний префікс таблиць БД (по замовчуванню "AT_"). Ця опція дає можливість інтеграції таблиць програми у вже існуючу БД інших застосунків та додатків. Скрипт установки проведе спробу створити цю базу даних, що було вказано, якщо її ще не існує. Для цього потрібно надати аккаунту користувача MySQL дозволу на створення баз даних і таблиць. Якщо цей крок завершується помилкою, варто звернутися до системного адміністратора для зміни властивостей аккаунта і отримання можливості створення нових сутностей, або попросити його самого створити базу даних. Після створення бази потрібно вказати її назву у відповідне поле форми і тоді скрипт створить таблиці з даними.

3) Налаштування аккаунта адміністратора та системних особливостей

Аккаунт адміністратора використовується для керування аккаунтами користувачів та курсами. Визначені ним системні властивості впливають на створені курси і загальне користування системою.

4) Особистий аккаунт і параметри по замовчуванню

Персональний аккаунт може бути використано для запису на курси або створення цих курсів. Він не має жодних особливих можливостей керування системою. На цьому етапі можна вказати необхідність створення особистого аккаунта інструктора разом з базовим стартовим вітальним курсом.

5) Директорії

Тут відбувається створення директорії даних курсів. Її варто розмістити поза межами кореневої директорії існуючого веб сервера для більшої надійності і безпеки, а також надати їй обов'язкових прав доступу. На Unix машинах прийдеться вручну визначити і вказати ці права на папки та файли. А для систем, що працюються на базі операційної системи Windows зазвичай не потрібно проводити ніяких додаткових дій, хоча за деяких обставин можливо буде потреба налаштування властивостей деяких файлів і папок для того, щоб можна було робити у них запис. Далі потрібно вставити шлях директорії з контентом у відповідний текстбокс, попередньо запевнившись, що у цьому шляху не присутні ярлики (Windows) та символічні посилання (Unix).

6) Збереження конфігурацій

Перед завершальним етапом установки потрібно визначити файл конфігурацій з назвою «include/config.inc.php» і зробити його доступним для запису, інакше виникнуть помилки. Слідування інструкціям на екрані допоможе у разі якщо права доступу до файлу налаштувань не визначені.

7) Підтвердження на збір даних користування

Для сприяння розвитку рівня обслуговування товариства ATutor командою розробників програмного забезпечення присутній збір базової інформації про систему що запущена у момент інсталяції. Уся надана інформація залишається анонімною і приватною. Також присутня опція виключення з наданої інформації посилання на щойно встановлений сервер.

8) Завершальне вікно установки

Це завершальний етап установки. У вкладці браузера показується повідомлення про успішну операцію і можливість авторизації з використанням даних особистого аккаунта, що були вказані у кроці номер 4 цієї інструкції, або ж авторизації у аккаунт адміністратора, створеного у кроці номер 3.

3.2.2 Інструкція по встановленню модуля Messenger

Messenger є додатковим зовнішнім модулем для ATutor, тому процедура його інсталяції така ж сама як і для інших несистемних і нестандартних модулів та потребує прав адміністратора. Після авторизації у акаунт адміністратора потрібно перейти у розділ «Модулі» у головному меню на стартовій сторінці, а потім перейти за посиланням «Встановити модулі». Після цього є дві доступні опції: можна інсталювати модуль з архіву натиснувши «Select File», вибравши у діалоговому вікні архів модуля та натиснувши кнопку «Встановити» (Рисунок 3.1, зони виділені синім); або можна розпакувати уміст архіву модуля у директорію «mods» (коренева папка модуля месенджера повинна називатися «messenger»), у цьому випадку опція інсталювати месенджер буде знаходитись у списку на тому місці, що показано у Рисунку 3.1 (зони виділені червоним).

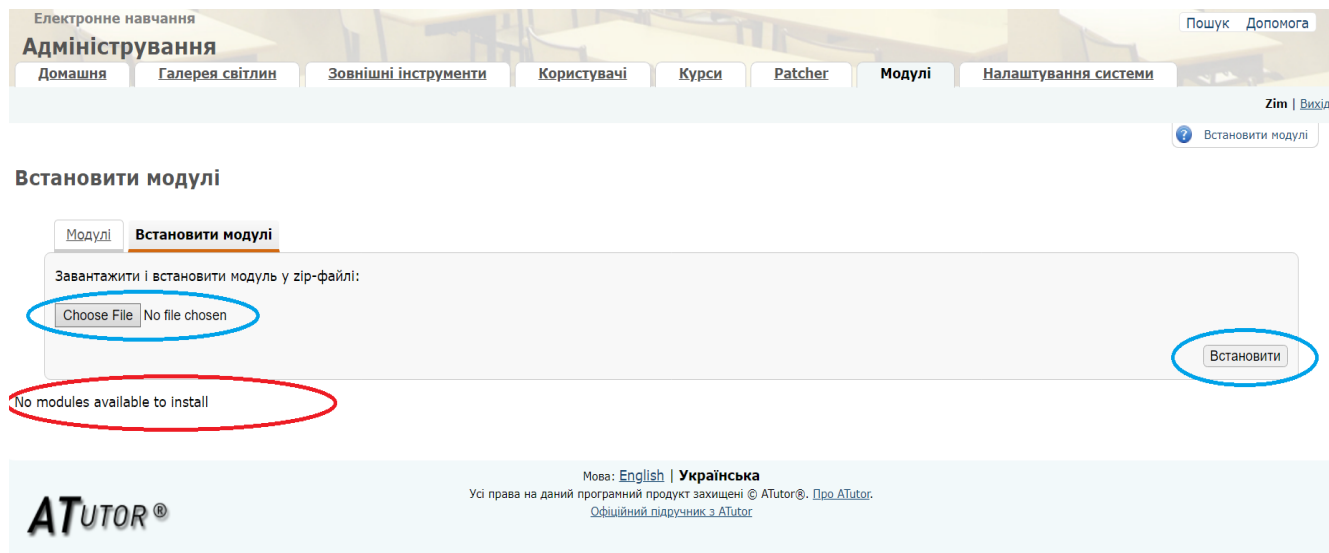


Рисунок 3.1 – Інсталяція модуля Messenger

Після підтвердження встановлення Месенджер буде додано до списку доступних модулів у вкладці «Модулі». Далі необхідно увімкнути даний модуль для того, щоб пізніше можна було додавати його на сторінки курсів. Це можливо

зробити вибравши радіо-кнопку навпроти Месенджера і натиснути «Увімкнути» так як показано на Рисунку 3.2.

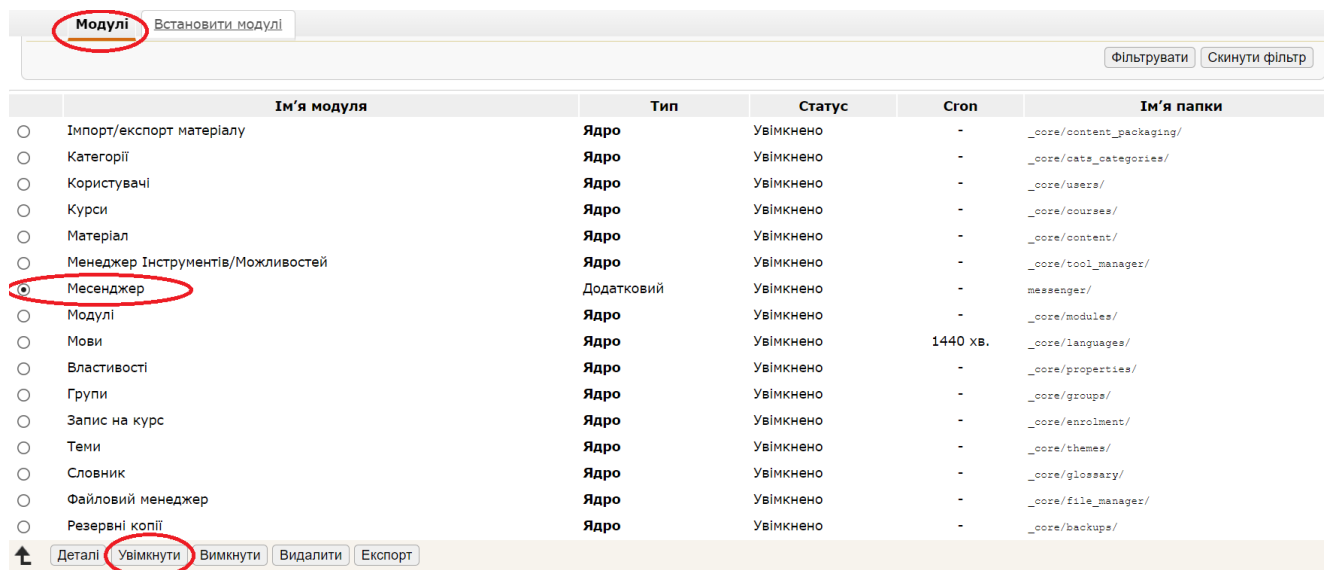


Рисунок 3.2 – Увімкнення модуля Messenger

Тепер, так як Месенджер на сервері увімкнено, він може бути доданий до будь якого курсу інструктором. Щоб це зробити слід, будучи авторизований як інструктор курсу, перейти до вкладки «Керування» на сторінці цього курсу і вибрати розділ «Інструменти курсу» (Рисунок 3.3).

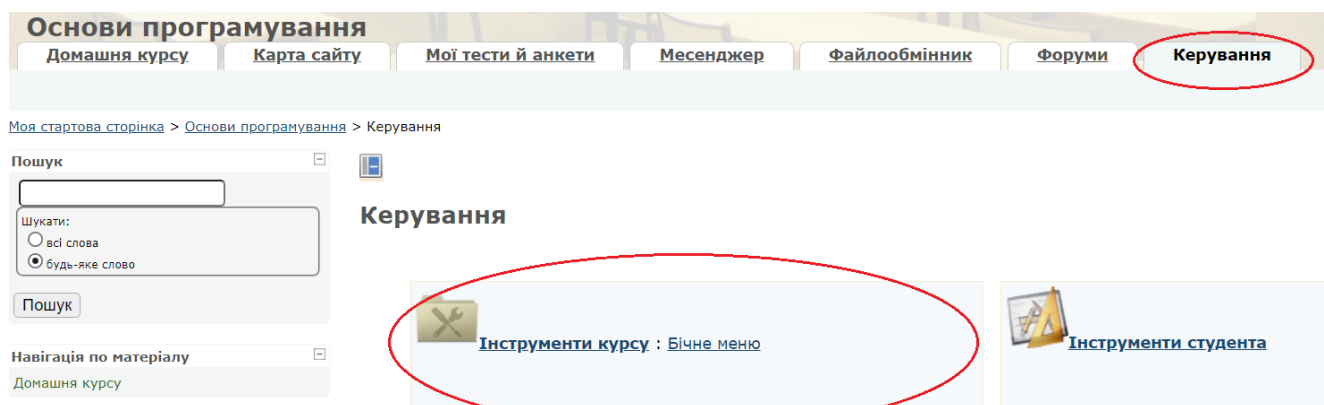


Рисунок 3.3 – Перехід до інструментів курсу

Далі увімкнути відображення месенджера на бажаних позиціях помітивши галочками відповідні чекбокси, а також налаштувати послідовність відображення

поміж інших модулів і пунктів (Рисунок 3.4). Після чого натиснути кнопку «Зберегти» внизу списку.

Інструменти курсу

Розділ	Адреса	Порядок
Карта сайту	<input checked="" type="checkbox"/> Закладки <input checked="" type="checkbox"/> Домашня	<input type="button" value="Перемістити вниз"/>
Мої тести й анкети	<input checked="" type="checkbox"/> Закладки <input type="checkbox"/> Домашня	<input type="button" value="Перемістити вгору"/> <input type="button" value="Перемістити вниз"/>
<u>Месенджер</u>	<input checked="" type="checkbox"/> <u>Закладки</u> <input checked="" type="checkbox"/> <u>Домашня</u>	<input type="button" value="Перемістити вгору"/> <input type="button" value="Перемістити вниз"/>
Файлообмінник	<input checked="" type="checkbox"/> Закладки <input type="checkbox"/> Домашня	<input type="button" value="Перемістити вгору"/> <input type="button" value="Перемістити вниз"/>
Форуми	<input checked="" type="checkbox"/> Закладки <input type="checkbox"/> Домашня	<input type="button" value="Перемістити вгору"/>
Моя робота з курсом	<input type="checkbox"/> Закладки <input checked="" type="checkbox"/> Домашня	<input type="button" value="Перемістити вниз"/>
Користувачі	<input type="checkbox"/> Закладки <input checked="" type="checkbox"/> Домашня	<input type="button" value="Перемістити вгору"/> <input type="button" value="Перемістити вниз"/>
Експорт матеріалу	<input type="checkbox"/> Закладки <input checked="" type="checkbox"/> Домашня	<input type="button" value="Перемістити вгору"/>
Словник	<input type="checkbox"/> Закладки <input type="checkbox"/> Домашня	
Чат	<input type="checkbox"/> Закладки <input type="checkbox"/> Домашня	
Посилання	<input type="checkbox"/> Закладки <input type="checkbox"/> Домашня	
Опитування	<input type="checkbox"/> Закладки <input type="checkbox"/> Домашня	
Часті запитання (FAQ)	<input type="checkbox"/> Закладки <input type="checkbox"/> Домашня	
Групи	<input type="checkbox"/> Закладки <input type="checkbox"/> Домашня	
Список літератури	<input type="checkbox"/> Закладки <input type="checkbox"/> Домашня	
Залікова книжка	<input type="checkbox"/> Закладки <input type="checkbox"/> Домашня	

Рисунок 3.4 – Увімкнення відображення Messenger

Також, існує можливість увімкнення відображення месенджера у боковому меню курсу. Для цього слід в налаштуваннях інструментів курсу перейти у вкладку «Бічне меню», де буде можливість обрати які саме компоненти і у якому порядку відображати.

Для коректної роботи модуля обов'язково необхідно визначити змінні налаштувань в конфігураційному файлі ATutor – «include/config.inc.php». Приклади конфігурацій продемонстровано нижче:

```
/* Domain of the XMPP server to connect */
define('XMPP_SERVER_PROTOCOL', 'http');
```

```

/* Domain of the XMPP server to connect */
define('XMPP_SERVER_DOMAIN', 'tntuchat');

/* Port of the XMPP server */
define('XMPP_SERVER_PORT', '9090');

/* Port of the JSXC web client of messenger */
define('JSXC_PORT', '7070');

/* Default server address for groupchats */
define('DEFAULT_GROUPCHAT_DOMAIN', 'conference.tntuchat');

```

Після виконання усіх вище перелічених дій, Месенджер успішно встановлено і додано на курс. Перейшовши на домашню сторінку курсу можна помітити вкладку Месенджера на головній панелі і пункт з іконкою в розділах курсу, натиснувши на які, можна розпочати переписуватися і спілкуватися (Рисунок 3.5).

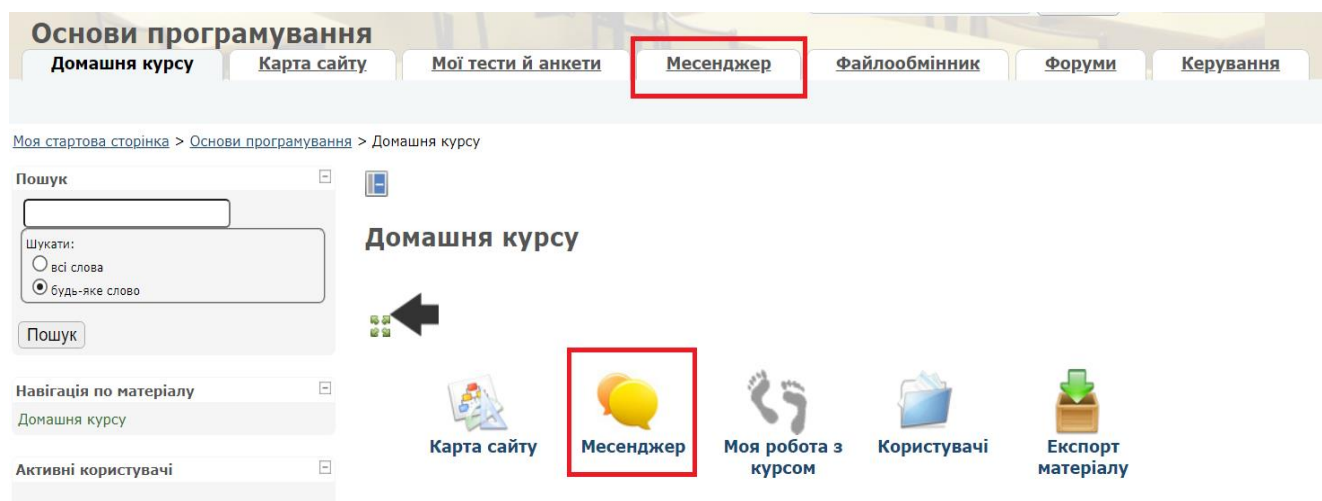


Рисунок 3.5 – Модуль успішно інстальовано та застосовано

Варіанти використання месенджера будуть детально описані і продемонстровані нижче.

3.2.3 Інструкція по встановленню XMPP сервера Openfire

Всі інструкції по інсталяції цього сервера наведені з відповідністю з його офіційним сайтом [18].

Перед початком установки даної програми варто переконатися у тому що всі технічні та системні вимоги було враховано та виконано, а всі складові сумісні між собою, рекомендовано використовувати версії застосунків такі ж самі що вказані у цій роботі.

Спочатку потрібно скачати інсталятор програми з офіційного сайту [19], за посиланням [20], або ж скористатися інсталятором який надано у цьому проекті. Сам процес встановлення сервера не є складним і не потребує ніяких додаткових дій чи обов'язкових змін в параметрах установки. Після його завершення відкриється лаунчер сервера а сам сервер автоматично запуститься (Рисунок 3.6).



Рисунок 3.6 – Лаунчер XMPP сервера Openfire

Якщо лаунчер Openfire використовується на системах з ОС Windows Vista або новіше з увімкненим UAC захистом, то він повинен бути запущений від імені

адміністратора для того, що мати змогу записувати зміни конфігурацій та змінювати вбудовані бази даних (якщо такі є). Якщо ж Openfire працює через лаунчер без прав адміністратора, то він не має належних дозволів для запису змін, у результаті чого виникають помилки у роботі сервера і текст помилок виводиться червоним кольором у лаунчері.

Після встановлення і запуску сервера слід його налаштувати, натиснувши кнопку «Launch Admin», відбудеться автоматичний перехід у вкладку браузера для вводу конфігурацій, налаштування профілів, підключення до бази даних та створення аккаунта адміністратора (Рисунок 3.7).

Це один із ключових моментів розгортання системи тому, не дивлячись на те, що деякі поля можна залишати заповненими по замовчуванню, на цьому етапі потрібно звернути увагу до введених даних і налаштувань. Дуже важливо запам'ятати логін і пароль адміністратора!

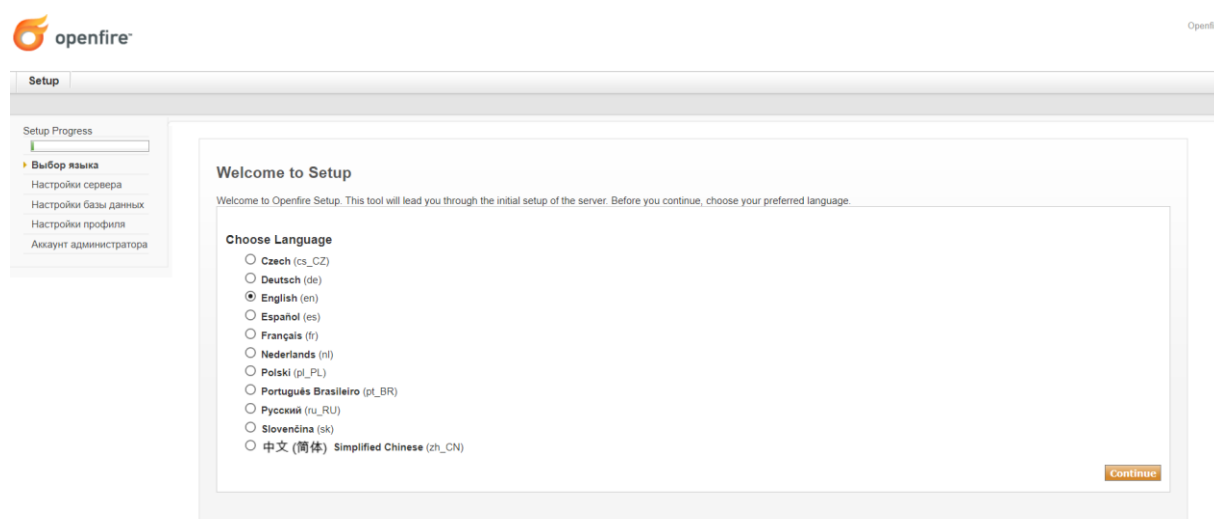


Рисунок 3.7 – Конфігурація сервера через веб-клієнт

Після вибору мови, потрібно вказати параметри і дані сервера, а саме: вказати доменне ім'я та ім'я хоста; прописати порти для звичайного і захищеного з'єднання з сервером; вибрати тип шифрування та опціонально вказати власний ключ шифрування (Рисунок 3.8).

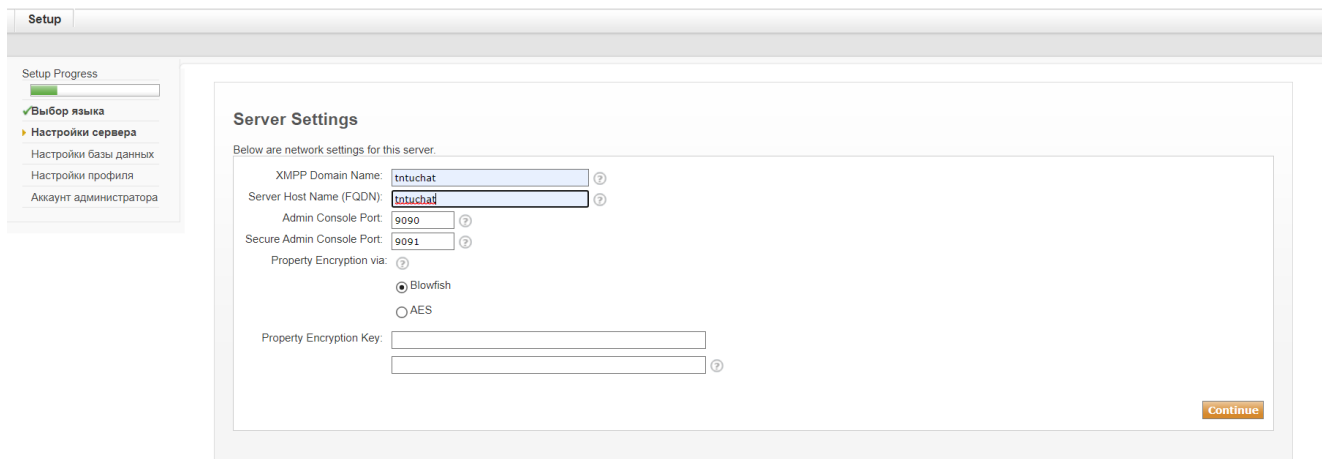


Рисунок 3.8 – Базові настройки сервера

У наступному пункті «Настройка бази даних» потрібно обрати варіант стандартного підключення БД, і у оновленому вікні вказати який саме вид бази даних буде використовуватися, вказати на неї посилання, а параметром посилання вказати конкретний ідентифікатор попередньо створеної бази у яку запишуться таблиці. Також обов'язково надати логін та пароль для доступу. Решту параметрів перевизначати за необхідністю (Рисунок 3.9).

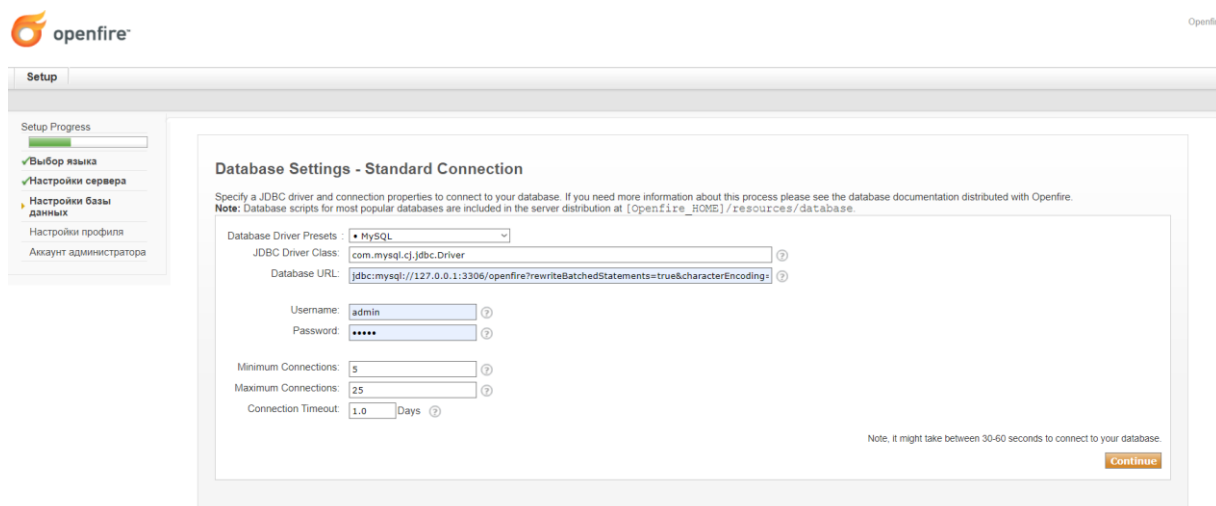


Рисунок 3.9 – Настройка бази даних сервера

У наступних вкладках відбувається налаштування параметрів профілю по замовчуванню, спосіб збереження даних користувачів, а також створення аккаунта

адміністратора. Після успішного завершення налаштування сервера, автоматично відкриється вікно авторизації у систему керування Openfire, у якому треба вказати дані аккаунта адміністратора, що були введені на останньому етапі налаштування (Рисунок 3.10).

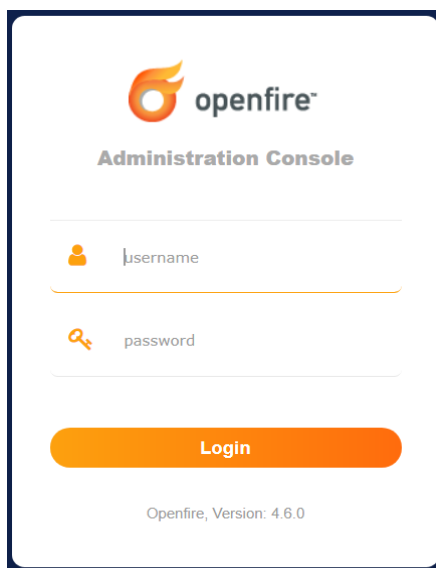
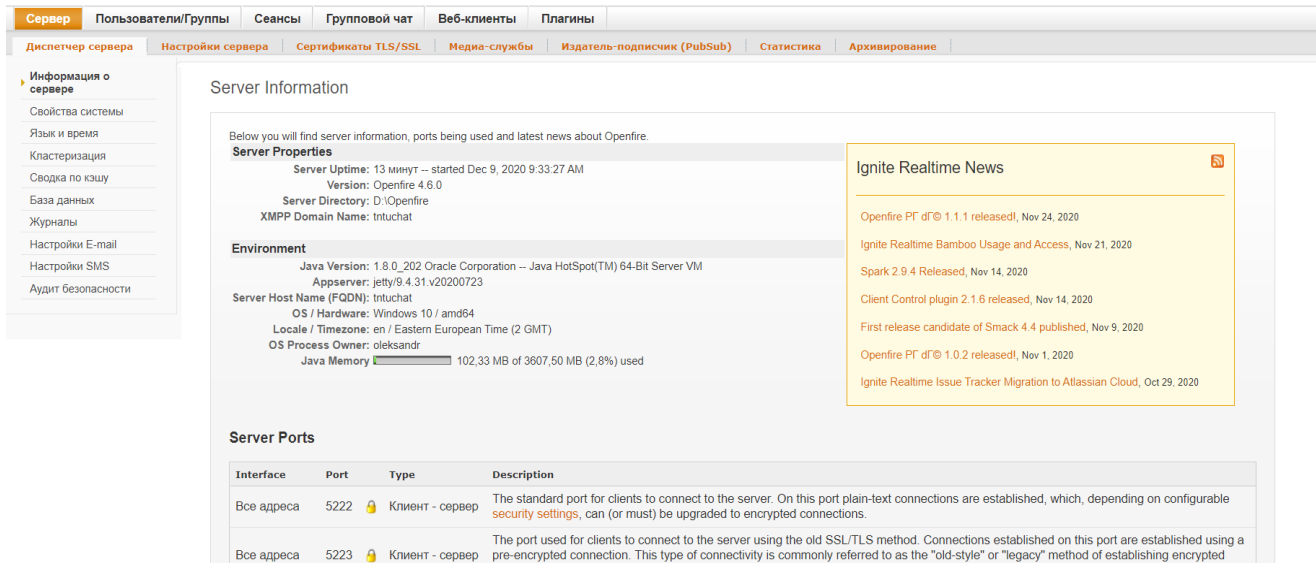


Рисунок 3.10 – Форма авторизації у систему керування сервером

Після авторизації відкриється панель управління де відображається загальна інформація і надається доступ до налаштування усіх можливих параметрів, до яких входить налаштування самого сервера і варіантів його обробки даних, з'єднання, захисту і шифрування, користувачів, груп користувачів, сеансів, групових чатів, розсилки, архівування, статистики та звітів, споживання і використання ресурсів, додаткових плагінів для розширення функціоналу, ролей користувачів та їхніх дозволів, модерування та багато іншого (Рисунок 3.11). Параметри конфігурацій, що були визначені у процесі установки сервера, а також у панелі керування, повинні бути введені у конфігураційний файл ATutor «include/config.inc.php» про який було згадано у пункті 3.2.2.




Server Information

Below you will find server information, ports being used and latest news about Openfire.

Server Properties

- Server Uptime: 13 минут -- started Dec 9, 2020 9:33:27 AM
- Version: Openfire 4.6.0
- Server Directory: D:\Openfire
- XMPP Domain Name: Intuchat

Environment

- Java Version: 1.8.0_202 Oracle Corporation -- Java HotSpot(TM) 64-Bit Server VM
- Appserver: jetty/9.4.31.v20200723
- Server Host Name (FQDN): Intuchat
- OS / Hardware: Windows 10 / amd64
- Locale / Timezone: en / Eastern European Time (2 GMT)
- OS Process Owner: oleksandr
- Java Memory  102,33 MB of 3607,50 MB (2,8%) used

Server Ports

Interface	Port	Type	Description
Все адреса	5222	Клиент - сервер	The standard port for clients to connect to the server. On this port plain-text connections are established, which, depending on configurable security settings , can (or must) be upgraded to encrypted connections.
Все адреса	5223	Клиент - сервер	The port used for clients to connect to the server using the old SSL/TLS method. Connections established on this port are established using a pre-encrypted connection. This type of connectivity is commonly referred to as the "old-style" or "legacy" method of establishing encrypted

Ignite Realtime News

- Openfire PT dГ© 1.1.1 released!, Nov 24, 2020
- Ignite Realtime Bamboo Usage and Access, Nov 21, 2020
- Spark 2.9.4 Released, Nov 14, 2020
- Client Control plugin 2.1.6 released, Nov 14, 2020
- First release candidate of Smack 4.4 published, Nov 9, 2020
- Openfire PT dГ© 1.0.2 released!, Nov 1, 2020
- Ignite Realtime Issue Tracker Migration to Atlassian Cloud, Oct 29, 2020

Рисунок 3.11 – Панель управління сервером

Якщо усе було виконано керуючись даною інструкцією, то проблем і помилок не повинно виникнути.

3.2.4 Інструкція по встановленню плагіна веб-клієнта JSXС

Для того, щоб підключити веб-клієнт JSXС у XMPP сервер Openfire достатньо зайти у панель управління (пункт 3.2.3) і перейти по вкладці головного меню «Плагіни». У завантаженому вікні відкрити список доступних плагінів клікнувши на пункт «Доступні плагіни» (Рисунок 3.12). Тут перераховані усі доступні плагіни по замовчуванню для встановлення, їхній опис, версія, документи readme та розмір пам'яті що вони займають.

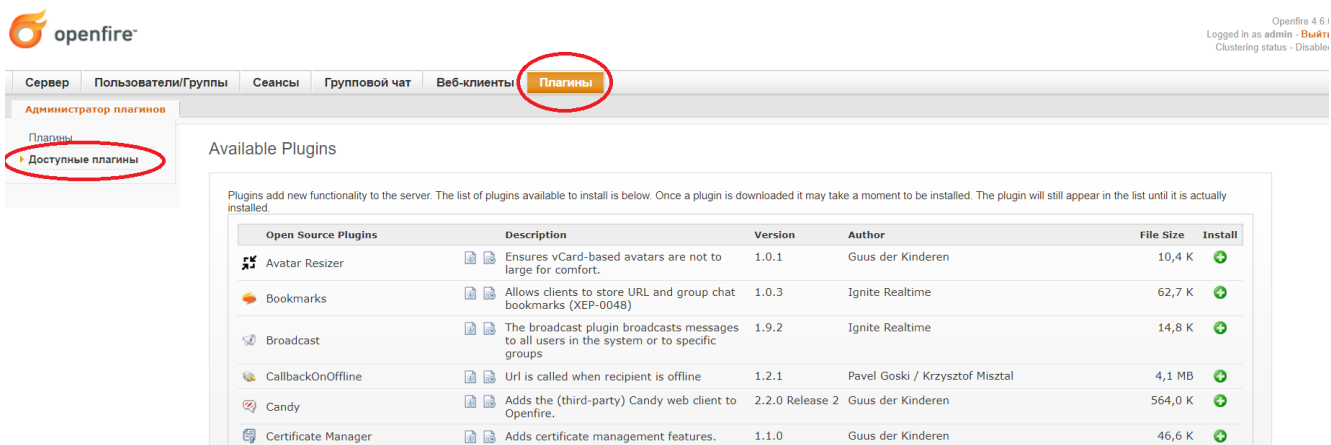


Рисунок 3.12 – підключення плагіна JSXС

Далі необхідно знайти у списку потрібний плагін і натиснути на зелений знак плюса навпроти нього, після чого розпочнеться завантаження та установка. Після успішного встановлення плагіна він з'явиться в переліку у вкладці бокового меню під назвою «Плагины», а у верхній головній панелі з'явиться вкладка «Веб клієнти» де можна налаштувати сам JSXС.

Також необхідно підключити плагін Monitoring, після встановлення якого, з'явиться можливість експортувати повідомлення переписок.

3.3 Інструкція по використанню

Для того, щоб розпочати користуватися месенджером необхідно, щоб існувало кілька користувачів, з якими можна було б з ними переписуватися, а також створений груповий чат. Переписуватися можна як і діалогами, тобто особисті переписки між двома користувачами так і приєднавшись до бесіди, тобто групова переписка. Якщо модуль Messenger добавлено на курс, то на ньому з'явиться сторінка месенджера з інформацією (Рисунок 3.13).

Електронне навчання Моя стартова сторінка [Назад до адміністру](#)

Основи програмування

Домашня курсу |
 Карта сайту |
 Мої тести й анкети |
 Месенджер |
 Файлообмінник |
 Форуми |
 Керування

[Моя стартова сторінка](#) > [Основи програмування](#) > Месенджер

Пошук

 Шукати:
 всі слова
 будь-яке слово

Месенджер

← Ласкаво просимо до месенджера!

Адреса групового чату цього курсу: **basics@conference.tntuchat**

Для того щоб розпочати Вам потрібно увійти у [Месенджер](#).
 Якщо у Вас ще не має аккаунта, то Ви можете [Зареєструватися](#).

Навігація по матеріалу

Домашня курсу

Активні користувачі

Vihlinskyi Oleksandr
Гості не показані

Месенджер

Адреса групового чату цього курсу:
basics@conference.tntuchat

Увійти у Месенджер
 Або Зареєструватися

Рисунок 3.13 – Сторінка месенджера курсу

Якщо адміністратор або інструктор курсу створив та додав груповий чат курсу то його адреса буде відображатися на сторінці. Щоб перейти безпосередньо до переписок і чатів потрібно натиснути «увійти у Месенджер». У разі якщо потрібно зареєструватися то слід натиснути «Зареєструватися».

Після переходу по посиланню реєстрації у новій вкладці браузера відкриється сторінка з формою і полями даних нового користувача (Рисунок 3.14).

Реєстрація у месенджері ТНТУ

Используйте форму ниже, чтобы создать новую учетную запись пользователя

Регистрация

Имя пользователя: *

Имя:

Электронная почта:

Пароль: *

Подтвердите пароль: *

* Обязательные поля

Рисунок 3.14 – Сторінка Реєстрації

Як тільки нового користувача успішно створено, дані, вказані при реєстрації можна вказувати при авторизації на сторінці веб-клієнта месенджера. На сторінку веб-клієнта можна перейти з сторінки курсу, натиснувши «Увійти» (Рисунок 3.15).

The image shows a login form with the following elements:

- Title: **Вхід**
- Input field for **Ім'я користувача** (Username)
- Input field for **Пароль** (Password)
- Buttons: **З'єднати** (Connect) and **Скасувати** (Cancel)

Рисунок 3.15 – Сторінка авторизації у месенджер

Авторизувавшись, відкриється і стане доступним інтерфейс користування (Рисунок 3.16), де у правому боці екрану будуть перелічені контакти і чати користувача а посередині відображатимуться самі вікна переписок з можливістю їхнього масштабування (Рисунок 3.17).

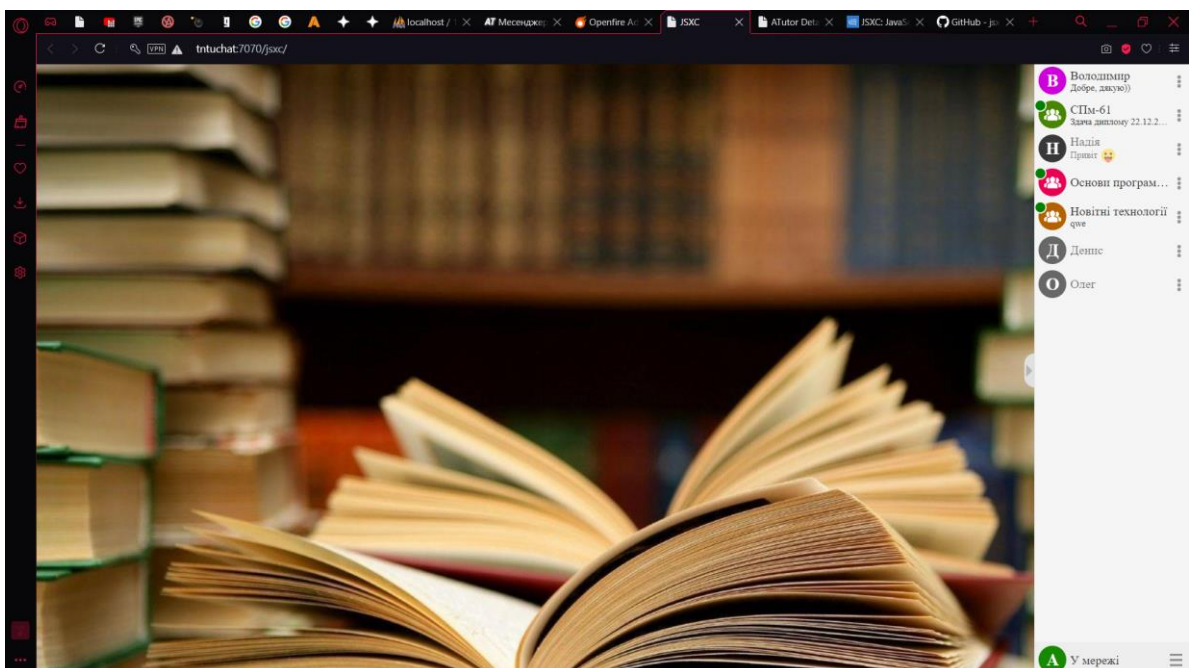


Рисунок 3.16 – Інтерфейс користувача месенджера

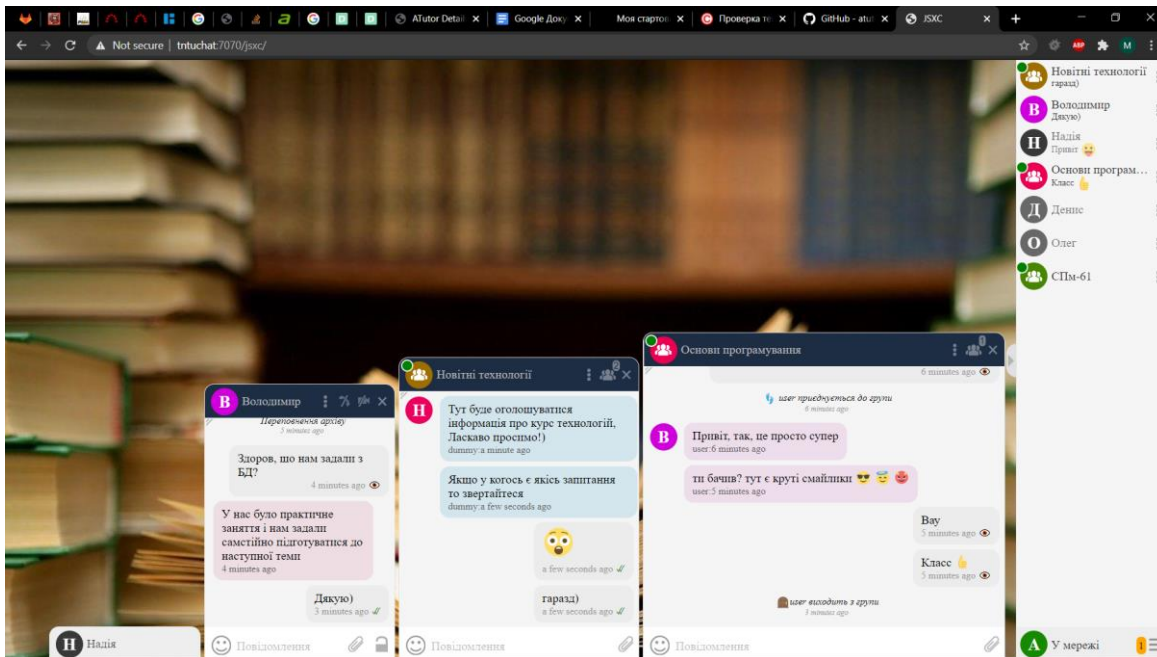


Рисунок 3.17 – Масштабовані вікна переписок

Якщо потрібно додати користувача до своїх контактів, то у меню у нижньому правому куті екрану є опція «Додати контакт», після натискання на яку відкривається вікно додавання контакту (Рисунок 3.18). У цьому вікні слід вказати ім'я користувача.

✕

Додати контакт

Введіть повне ім'я користувача та додатковий псевдонім.

Account

Ім'я користувача

Псевдонім

Закрити
Додати

Рисунок 3.18 – Опція «Додати контакт»

Після чого користувач з'явиться у списку контактів на панелі справа.

При необхідності приєднатися до нової бесіди потрібно вибрати пункт «Приєднатися до групи» у тому ж самому меню, після чого відкриється вікно приєднання до чату (Рисунок 3.19) у якому потрібно вибрати необхідну групу з випадаючого списку і натиснути «Продовжити».

Приєднатися до групи

Будь ласка, зазначте назву групи, ваше ім'я (необов'язково) та пароль для приєднання до чату

Account dummy@tntuchat

Сервер conference.tntuchat

Група

Псевдонім

- basics
Основи програмування
- new
- tech
Technology

Рисунок 3.19 – Опція «Приєднатися до групи»

За потреби можна вибрати свій поточний статус натиснувши на нього біля головного меню. Появиться випадаючий список з доступними статусами користувача (Рисунок 3.20). Стан користувача визначає його можливість отримувати повідомлення та сповіщення і дає змогу зрозуміти іншим користувачам його доступність і зайнятість у даний момент.

Для виходу чи зміни акаунт потрібно обрати пункт «Поza мережею», після цього знову появиться вікно авторизації.

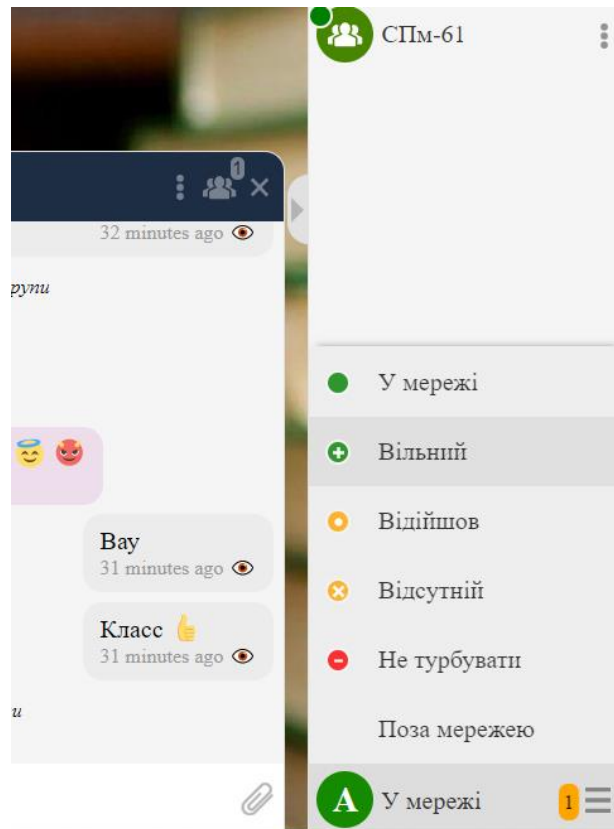


Рисунок 3.20 – Зміна поточного статусу користувача

Подальше використання месенджера є абсолютно інтуїтивно зрозумілим і не потребує подальших інструкцій застосування. А у разі виникнення питань, у головному меню є пункт «Довідка», наповнення якої може змінюватися адміністраторами сервера.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Охорона праці

Згідно з посилання на нормативні акти про безпеку, закону України «Про охорону праці» [21], закону України «Про охорону здоров'я» [22] кожна людина має природне невід'ємне і непорушне право на охорону здоров'я а суспільство і держава відповідальні перед сучасним і майбутніми поколіннями за рівень здоров'я і збереження генофонду народу України. Громадяни також мають право на безпечне для здоров'я і життя навколишнє середовище, санітарно-епідемічне благополуччя території, безпечні і здорові умови навчання, праці, побуту та відпочинку, кваліфіковану санітарно-медичну допомогу, включаючи вільний вибір лікаря і закладу охорони здоров'я, достовірну та своєчасну інформацію про стан свого здоров'я і здоров'я населення та інше.

Як частина суспільства, і, зокрема, людина, яка розробила програмний продукт, яким будуть користуватися безліч людей, розробник частково відповідальний за надійність, якість та безпечність використання розробленого програмного продукту користувачами. Саме тому, під час розробки було проаналізовано та взято до уваги можливі ризики використання розроблюваної програми, а розробка проводилася відповідно вимогам охорони здоров'я та принципам комп'ютерної етики[.]

Програмна система розроблена відповідно вимогам з охорони праці і пожежної безпеки. Програма не може призвести до перевантаження і перегріву серверів, що, у свою чергу, могло би призвести до пошкодження обладнання університету чи інших установ та закладів, нанесення шкоди працівникам, викладачам, персоналу, студентам чи іншим людям які матимуть відношення чи зв'язок з цією програмою, пожежі, надзвичайних ситуацій, порушення якісної і безперебійної роботи сайту дистанційного навчання, порушення навчального процесу студентів та викладачів.

Завдяки відповідності програми вимогам охорони здоров'я та принципам комп'ютерної етики, розроблена програмна система є абсолютно безпечною а усі ризики завдання шкоди собі, користувачам та взагалі будь-кому виключені. До чинників безпечності програми відносяться:

- кольорова палітра користувацького інтерфейсу підібрана таким чином, щоб очі користувача перебували у стані максимального спокою та не перенавантажувалися, а сам інтерфейс не переповнено графічними елементами для зручності і зрозумілості. Це допоможе користувачам не псувати свій зір користуючись месенджером;

- месенджер розроблено за принципом «чим простіше – тим краще». Таким чином кінцевий продукт вийшов максимально простим, зручним та інтуїтивним, тому месенджером можуть користуватися користувачі будь-якого контингенту та з будь-якими навичками користування новітніми технологіями. Доступність системи запобігає виникненню небажаних результатів користування через необережність та необачність;

- не дивлячись на те що розроблена система є багатофункціональною і її можна використовувати для комунікації як в навчальних цілях так і в особистих, з метою проведення дозвілля, вона не викликає залежності;

- у налаштуваннях сервера можна увімкнути фільтрування нецензурних та лайливих слів та(або) заборонених висловів, що сприяє розвитку здорового спілкування та перешкоджає булінгу;

- у налаштуваннях групових чатів адміністратор може налаштувати використання псевдонімів для анонімної комунікації, що дає змогу не розкривати свого справжнього імені коли це потрібно;

- користуючись месенджером не можливо нанести шкоду собі чи людям навколо;

- захист користувачів від зловмисників на програмному рівні, а саме: шифрування повідомлень, обмежений перехід по небезпечним інтернет посиланням, захист від перехвату повідомлень та віддаленого ініціювання відправлення форм.

- користування месенджером модерується адміністраторами як з боку сервера (загальні налаштування та керування користувачами, групами користувачів, бесідами та діалогами) так і з клієнтської частини програмного забезпечення (адміністрування бесід і групових чатів).

Програмна система не містить спеціально доданих шкідливих застосунків, вірусів, вразливостей, експлоїтів, бекдорів, що надавали би розробнику чи зацікавленим особам особливих привілеїв або можливостей для обходу системи безпеки дистанційного навчання для отримання тих чи інших переваг чи для будь-яких інших цілей. Система безпеки розробленої програми визначається та обмежується системою безпеки усіх використаних застосунків, програм та інструментів, використаних під час розробки.

Перед розгортанням та використанням розробленої програми рекомендовано провести тести її роботи в межах середовища, у яке її планують інтегрувати, та провести запуск у тестовому режимі. У разі розгортання даного програмного продукту та подальшого, як наслідок, виведення з ладу обладнання, комп'ютерів, серверів, або втрати даних відповідальність за нанесену шкоду бере на себе особа що дала наказ на розгортання та запуск даного програмного продукту, якщо така є, або ж особа що безпосередньо провела розгортання та запуск програми. Розробник, не несе відповідальності за шкоду нанесену у ході використання або неправильної інтеграції чи експлуатації даної програми.

4.2 Безпека в надзвичайних ситуаціях

Досягнення прийняттого рівня безпеки в системі “людина – середовище – мешкання” нерозривно пов'язано з необхідністю глибокого аналізу причин зростання чисельності і рівня діючих в техносфері небезпек; вивчення причин примусової втрати здоров'я та гибелі людей; розробка та широкого вжитку превентивних заходів на виробництві, в побуті і в регіонах техносфері.

Важливу роль в збереженні здоров'я та життя людей в сучасному та майбутньому закликає відігравати інформаційна діяльність держави в галузі прогнозування небезпек серед мешкання. Компетентність людей в світі небезпек і способи захисту від них - необхідна умова досягнення безпеки життєдіяльності людини на всіх етапах його життя.

Науково-технічний прогрес радикально змінив світ, породивши нові загрози для цивілізації. У житті сучасної людини все більше місце займають турботи, пов'язані з подоланням різних кризових явищ, що виникають в процесі розвитку земної цивілізації. В Україні, як і в усьому світі, в останні роки спостерігається зростання числа катастроф природного та техногенного характеру. Це обумовлено, перш за все, прогресуючої урбанізацією територій, збільшенням щільності населення Землі, і, як наслідок, збільшенням антропогенного навантаження на навколишнє середовище. Захист природних систем і населення від НС природного і техногенного характеру сформувалася в останні роки як нагальна і об'єктивна потреба суспільства і держави.

Надзвичайна ситуація (НС) - це стан, при якому в результаті виникнення джерела НС на об'єкті, визначеній території (акваторії) порушуються нормальні умови життя і діяльності людей, виникає загроза їх життю і здоров'ю, завдається шкода майну населення, народному господарству та навколишньому природному середовищу.

Будь-які НС в своєму розвитку проходять чотири стадії: зародження, ініціювання, кульмінацію і згасання.

Стадія зародження. Створюються передумови НС: активізуються несприятливі природні процеси, накопичуються технологічні неполадки, систематично порушуються вимоги заходів безпеки, наростає соціальна напруженість в державі або в міждержавних відносинах і т. д.

Стадія ініціювання. Виникають порушення, пов'язані з виходом параметрів процесу або явища за критичні значення всередині системи (внутрішні фактори). Відбуваються, наприклад, спонтанні реакції, розгерметизація трубопроводів, резервуарів і т.п. Може порушуватися робота обладнання, виявився бракованим

систем забезпечення (електропостачання, водопостачання, охолодження, вентиляції та ін.). На ініціювання впливають і зовнішні фактори, до числа яких можна віднести стихійні лиха, акти вандалізму, провокації, політичні помилки і т.п.

Стадія кульмінації. В результаті "спрацьовування" ініціюючих факторів вивільняються великі кількості енергії і маси, які і призводять до ЧС. При цьому іноді навіть невелике ініціює подія може пустити в хід ланцюгової механізм аварій з багаторазовим збільшенням потужності і масштабів ("ефект доміно"). Зовнішні чинники, такі як вандалізм і провокації, можуть привести до ЧС, пов'язаних з масовими заворушеннями, терористичними актами, військовими діями.

Стадія загасання. Триває від моменту усунення джерела НС до повної її ліквідації.

Джерелом НС може бути небезпечне природне явище, технічне подія, соціальні явища, в результаті яких сталася або може виникнути НС. Реципієнтом НС може бути природна екосистема, людина, а також технічна система.

Згідно з цією класифікацією, природними НС будуть НС, джерелом і реципієнтом яких є природа: в ролі природних джерел НС можуть виступати селі, обвали, лавини, смерчі і інші небезпечні природні явища, які будуть чинити негативний вплив на природні екосистеми.

Природно-соціальними НС слід вважати НС, джерелом яких служать природні явища (заморозки, сильна ожеледь та ін.), А об'єктом є людина (наприклад, здоров'я людини).

Природно-техногенні НС - це НС, викликані природними явищами (наприклад, повінню), які чинять негативний вплив на техногенні об'єкти (наприклад, на виробничі системи).

Техноприродні НС - це НС, викликані техногенними причинами (наприклад, аварією нафтопроводу), які завдають шкоди природним екосистемам (наприклад, водної екосистемі).

Техносоціальні НС - це НС, джерелом виникнення якої служать техногенні причини (наприклад, аварія на електроенергетичних системах), які завдають шкоди соціальній системі (людині).

Техногенні НС - це НС, джерелом виникнення яких служить техногенне явище (наприклад, аварію вантажного поїзда), яке завдає шкоди техногенному об'єкту (наприклад, будується будівлі).

Соціоприродні НС - це НС, причиною виникнення якої є людина (наприклад, недотримання ним правил техніки безпеки і порушення ходу експлуатації обладнання), а об'єктом якої є природна екосистема (наприклад, водний об'єкт).

Соціальні НС - це НС, обумовлені людським фактором (наприклад, недотримання правил контролю якості продуктів харчування), а об'єктом яких є сама людина (зокрема, здоров'я людини).

Соціотехногенними НС слід вважати НС, причиною яких служить людина і його поведінка (наприклад, недбалість), а реципієнтом яких є техногенний об'єкт (наприклад, будівля, згоріла в результаті пожежі через недбалість).

Необхідно відзначити, що джерело НС простіше віднести до однієї з трьох категорій джерел (природних, техногенних або соціальних), на відміну від реципієнтів НС. Реципієнтом (об'єктом) НС у багатьох випадках є як природна екосистема, так і техногенна система, а також осіб, беручи до уваги кумулятивні і каскадні ефекти в системі природа - техносфера - людина. Тому розгляд НС за представленою вище класифікації носить певною мірою умовний характер.

Більш зрозумілим є розгляд класифікації НС за джерелом їх виникнення: НС природного, техногенного, соціального походження (характеру). Основні прояви в навколишньому середовищі знаходять НС природного і техногенного походження (характеру). У свою чергу соціальний фактор слід вважати невід'ємною компонентою функціонування техногенних систем.

Правила безпечної поведінки в надзвичайних ситуаціях.

При загрозі ураження стрілецькою зброєю:

- закрити штори або жалюзі (заклеїти вікна паперовою стрічкою) для зниження ураження уламками скла;
- вимкнути світло, закрити вікна та двері;
- зайняти місце на підлозі в приміщенні, що не має вікон на вулицю (комора, ванна, передпокій);

- інформувати можливими засобами про небезпеку близьких і знайомих.

При загрозі бойових дій:

- закрити штори або жалюзі (заклеїти вікна паперовою стрічкою) для зниження ураження осколками скла. Вимкнути джерело живлення, закрити воду і газ, загасити пічне опалення;

- взяти документи, гроші і продукти, предмети першої необхідності, медичну аптечку;

- негайно покинути житлове приміщення, сховатись в підготовленому підвалі або найближчому укритті;

- попередити про небезпеку сусідів. Надати допомогу старим і дітям;

- без крайньої необхідності не залишати безпечного місця перебування;

- проявляти крайню обережність, не піддаватися паніці.

Якщо стався вибух:

- переконатися в тому, що не отримано значних травм;

- заспокоїтися і уважно озирнутися навколо, чи не існує загрози подальших обвалів і вибухів, чи не звисає з руїн розбите скло, не потрібна комусь допомога;

- якщо є можливість - спокійно вийти з місця події. Пам'ятати, що при низькій активності людина може протриматися без води п'ять діб;

- виконувати всі розпорядження рятувальника.

При повітряній небезпеці:

- вимкнути джерело живлення, закрити воду і газ;

- погасити пічне опалення;

- взяти документи, гроші і продукти, предмети першої необхідності, медичну аптечку;

- попередити про небезпеку сусідів і при необхідності надати допомогу старим і хворим;

- якнайшвидше дійти до захисної споруди або сховатися на місцевості. Дотримуватись спокою і порядку. Без крайньої необхідності не залишати безпечного місця перебування. Слідкувати за офіційними повідомленнями.

При загрозі масових заворушень:

- зберігати спокій і розсудливість;
- при перебуванні на вулиці негайно покинути місця масового скупчення людей, уникати агресивно налаштованих осіб;
- не піддаватися на провокації;
- надійно закрити двері. Не підходити до вікон і не виходити на балкон. Без крайньої необхідності не покидати приміщення.

При проведенні тимчасової евакуації цивільного населення з небезпечного району:

- взяти документи, гроші і продукти, необхідні речі, медикаменти;
- по можливості надати допомогу громадянам похилого віку, людям з фізичними вадами;
- дітям дошкільного віку вкладається в кишеню або пришивається до одягу записка, де вказується прізвище, ім'я, по батькові, домашня адреса, а також ім'я та по батькові матері і батька;
- дотримуватись вказаного маршруту. При необхідності звертатися за допомогою до правоохоронців та медичних працівників.

ВИСНОВКИ

Суспільство досягло значного прогресу у сфері комунікацій завдяки усвідомленню важливості якісного і доступного спілкування та завдяки серйозному і глибокому підходу до досліджень в даній сфері.

На сьогоднішній день мільярди людей по всьому світу використовують надбання та результати плідної тяжкої праці науковців, дослідників і розробників, не задумуючись про обсяги затраченого часу і сил для задоволення їхніх потреб. Приблизно так само небагато хто розуміє, що технології розробляються саме такими, якими їх потребує спільнота. А саме як наслідок прямого зв'язку між потребами і визначенням напрямку розвитку технологій і проводиться соціальна наукова та дослідницька діяльність – щоб визначити проблеми та потреби користувачів.

Варто наголосити, що обраний напрям розвитку не завжди відповідає потребам громадськості, тим не менш, яким би не був розвиток, він все одно залишає після себе цінний досвід, на основі якого можна зробити світ набагато кращим.

Беручи до уваги до уваги тенденції росту зацікавленості і потреби у інтернет-комунікаціях, можна з впевненістю сказати, що методи спілкування також не стоятимуть на місці а невпинно і стрімко рухатимуться у сторону підвищення ефективності, зручності та швидкості.

В цілому, розвиток комунікаційних технологій повинен здійснюватися за умови поєднання кількох чинників: сприяння наукових общин та університетів, врахування відгуку від користувачів та обміну результатами і надбаннями розробників. В інакшому випадку, прогрес буде повільним, неефективним або надлишковим.

Особливо уважно потрібно спостерігати за станом комунікації між студентами в університетах та під час навчального процесу, так як саме у період навчання студенти найбільш ймовірно об'єднуються у невеликі спільноти для

роботи над спільною справою, розробки стартапів та проектів, і цю ймовірність можливо значно збільшити, впровадивши якісний спосіб спілкування.

Розглядаючи світовий досвід університетів по підтримці наукової і дослідницької діяльності студентів, для досягнення результатів у цьому питанні не потрібно надто багато зусиль. Більше можливостей самореалізації і свободи вибору напрямку розвитку повинні ініціювати процес прискорення виникнення наукових спільнот серед студентів.

Варто зазначити, що стан розвитку інтернет-комунікацій під час навчання в Україні перебуває на початковій стадії. Незважаючи на це, створюються креативні простори розвитку, що стимулює ріст зацікавленості розробників і користувачів у цій галузі.

Роблячи погляд з висоти новоотриманого досвіду і знань, можна впевнено стверджувати, що напрямку спілкування і обміну інформацією ще є куди розвиватися, навіть більше, засоби і способи комунікації будуть змінюватися і модифікуватися до того часу, доки існує і розвивається суспільство.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Чемеркін С. Г. Мова інтернет-комунікації // Українська мова: Енциклопедія. Вид. 3-тє. — К.: Українська енциклопедія, 2007. — С. 365–366.
2. Чемеркін С. Г. Українська мова в Інтернеті (визначальні ознаки) // Збірник наук. праць Науково-дослідного інституту українознавства. — К., 2007. — Т. XVII. — С. 273–280.
3. Сайт дистанційного навчання Тернопільського національного технічного університету імені Івана Пулюя: [Електронний ресурс] – Режим доступу до ресурсу: <https://dl.tntu.edu.ua>
4. Інформаційна сторінка сайту дистанційного навчання Тернопільського національного технічного університету імені Івана Пулюя: [Електронний ресурс] – Режим доступу до ресурсу: <https://dl.tntu.edu.ua/about.php>
5. Документація системи управління навчанням ATutor : [Електронний ресурс] – Режим доступу до ресурсу: <https://atutor.github.io/atutor/features.html>
6. Сайт інтернет-сервісу для спілкування в режимі реального часу по всьому світу talkr.im : [Електронний ресурс] – Режим доступу до ресурсу: <https://www.talkr.im/>
7. Сторінка переліку модулів на офіційному сайті ATutor : [Електронний ресурс] – Режим доступу до ресурсу: <https://atutor.github.io/atutor/modules.html>
8. Веб сайт модуля ATutor Userplane : [Електронний ресурс] – Режим доступу до ресурсу: <http://www.userplane.com/index.cfm>
9. Крістофер А. Мова шаблонів. Міста. Будівлі. Будівництво : навч. посіб. — К., 2014. — 1096 с.
10. Еріх Г., Річард Х., Ральф Д., Джон В. Патерни проектування: повторно використовувані елементи архітектури об'єктно-орієнтованого програмного забезпечення : навч. посіб. — К., 1994. — 431 с.
11. Веб сайт бібліотеки moment.js : [Електронний ресурс] – Режим доступу до ресурсу: <https://momentjs.com>

12. Сторінка системних вимог системи управління навчанням ATutor : [Електронний ресурс] – Режим доступу до ресурсу: <https://atutor.github.io/atutor/docs/requirements.html>

13. Сторінка системних вимог XMPP сервера Openfire : [Електронний ресурс] – Режим доступу до ресурсу: <https://download.igniterealtime.org/openfire/docs/latest/documentation/install-guide.html>

14. Сторінка системних вимог веб-клієнта JSXС : [Електронний ресурс] – Режим доступу до ресурсу: <https://jsxsc.readthedocs.io/en/latest/getting-started/requirements.html>

15. Офіційним сайт системи управління навчанням ATutor : [Електронний ресурс] – Режим доступу до ресурсу: <https://atutor.github.io>

16. Сторінка завантажень офіційного сайту системи управління навчанням ATutor : [Електронний ресурс] – Режим доступу до ресурсу: <https://atutor.github.io/atutor/downloads.html>

17. Посилання на завантаження установочного файлу системи управління навчання ATutor : [Електронний ресурс] – Режим доступу до ресурсу: <https://sourceforge.net/projects/atutor/files/latest/download>

18. Офіційний сайт XMPP сервера Openfire : [Електронний ресурс] – Режим доступу до ресурсу: <https://www.igniterealtime.org>

19. Сторінка завантажень офіційного сайту XMPP сервера Openfire : [Електронний ресурс] – Режим доступу до ресурсу: <https://www.igniterealtime.org/downloads/#openfire>

20. Посилання на завантаження установочного файлу XMPP сервера Openfire : [Електронний ресурс] – Режим доступу до ресурсу: https://www.igniterealtime.org/downloadServlet?filename=openfire/openfire_4_6_0_bundledJRE_x64.exe

21. Державний сайт Верховної ради України. Закон «Про охорону праці» : [Електронний ресурс] – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12#Text>

22. Державний сайт Верховної ради України. Основи законодавства України про охорону здоров'я : [Електронний ресурс] – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2801-12#Text>

Додаток А
Технічне завдання
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

ТЕХНІЧНЕ ЗАВДАННЯ

на розробку проекту

«Інтеграція месенджерів в середовище ATutor. Експорт повідомлень скриньки і чату менеджер»

Розробник:
виконавець ст. гр. СПм-61
Віглінський Олександр Юрійович

(підпис)

керівник проекту
Пастух Олег Анатолійович

(підпис)

Тернопіль 2020

ЗМІСТ

1. ПІДСТАВИ ДО РОЗРОБКИ	67
2 ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	67
3 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ	67
3.1 Функціональні вимоги	67
3.2 Технічні вимоги	68
3.3 Програмні вимоги	68
4. ЕТАПИ РОЗРОБКИ	68
5. СУПРОВІДНА ДОКУМЕНТАЦІЯ	69
6. ПОРЯДОК ЗДАЧІ ПРОЕКТУ	69
7. ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА ЗМІНИ В ПРОЕКТІ	70

1. ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану підготовки магістрів за спеціальністю 121 «Інженерія програмного забезпечення».

Тема проекту: «Інтеграція месенджерів в середовище ATutor. Експорт повідомлень скриньки і чату менеджер».

Термін виконання: до 21.12.2020р.

2 ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Інформаційна система призначена для використання в навчальному процесі університету, на сайті дистанційного навчання.

Інформаційна система буде корисною в сферах освіти, науки, навчання та стане невід'ємною частиною у саморозвитку.

Інформаційна система дозволить здійснювати обмін повідомленнями та інформацією, спілкуватися як через приватні чати, так і приєднуватися до загальних бесід курсів і груп.

3 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Функціональні вимоги

Система повинна передбачати три ролі:

- користувач
- адміністратор
- модератор

Для користувачів система надає доступ до повного переліку функцій.

- можливість обміну миттєвими повідомленнями з іншими користувачами;
- можливість обміну файлами;
- створення аккаунта / авторизація;
- можливість додавати контакти;
- можливість адаптувати систему під себе.

Для модераторів доступні ті ж самі функції, що і для користувача, тільки з додатковою можливістю модерувати певні бесіди.

Для адміністраторів доступні ті ж самі функції, що і для модератора, тільки з додатковою можливістю повного контролю системи та адміністрування.

3.2 Технічні вимоги

Вимоги до клієнтської частини: Сучасні інтернет браузерери, інтуїтивно зрозумілий, не перевантажений інтерфейс: будь-яка бажана дія зі сторони користувача повинна досягатися в межах трьох кліків.

3.3 Програмні вимоги

Використання СУБД: MySQL

Розробка серверної частини: PHP

Розробка клієнтської частини: HTML + JS + CSS

Додаткові вимоги: використання алгоритму криптозахисту при передачі даних по відкритих каналах зв'язку.

4. ЕТАПИ РОЗРОБКИ

Розробка інформаційної системи проводиться в наступному порядку:

- аналіз предметної області, виявлення акторів та варіантів використання системи;
- вибір засобів розробки та архітектури системи;
- проектування бази даних системи;
- розробка програмного забезпечення системи;
- тестування інформаційної системи на реальних даних;
- оформлення супровідної документації;
- задача проекту.

Результати виконання кожного етапу проекту погоджуються з керівником проекту.

5. СУПРОВІДНА ДОКУМЕНТАЦІЯ

Для інформаційної системи повинні бути розроблені наступні документи:

- пояснювальна записка до проекту;
- презентація проекту;
- рецензія на проект;
- диск з проектом.

Пояснювальна записка до проекту оформляється згідно діючих вимог до нормоконтролю проектів.

6. ПОРЯДОК ЗДАЧІ ПРОЕКТУ

Розроблена інформаційна системи повинна відповідати вимогам, що складаються з перерахованих у п.3.1 цього документу характеристик.

Для задачі проекту необхідно підготувати весь перелік документів зазначений у п.5 цього документу.

Приймання проекту проводиться спеціально створеною комісією в термін зазначені в п.1 цього документу.

7. ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА ЗМІНИ В ПРОЕКТІ

Назва етапу	Відмітка*
Аналіз предметної області	
Архітектура системи	
Проектування бази даних	
Використання системи	
Супровідна документація	

* відмітки про виконання етапу ставляться керівником проекту

Додаток Б

Копія тез доповіді

УДК 004.773

Віглінський О. Ю., магістр другого року навчання кафедри програмної інженерії
Тернопільський національний технічний університет імені Івана Пулюя

Інтеграція месенджерів в середовище ATutor. Експорт повідомлень.

Vihlinsky O., second year of the master's degree in Software Engineering Department Integration of messengers into the ATutor environment. Export messages.

Згідно з інформаційною сторінкою сайту дистанційного навчання Тернопільського національного технічного університету імені Івана Пулюя [1] сайт розроблений на базі системи управління навчанням (LMS) ATutor для роботи з студентами та керування курсами і навчальними матеріалами. Як вказано у документації ATutor [2], система має вбудований модуль поштової скриньки для обміну повідомленнями, але він не є надто функціональним та не відповідає сучасним вимогам якісного навчання студентів. Саме тому мета і завдання дипломної роботи розробити та інтегрувати у існуюче середовище дистанційного навчання зручний, багатофункціональний і надійний спосіб обміну повідомленнями між студентами, викладачами та користувачами електронного навчання загалом.

Розширення функціоналу дистанційного навчання можливе завдяки модульній архітектурі системи управління навчанням Atutor. Отже, бажаний кінцевий результат розробки це окремий модуль месенджера з можливістю імпорту повідомлень, який можна буде налаштувати та інтегрувати у середовище Atutor. Можливість імпорту повідомлень дасть змогу не втратити вже досягнутого прогресу ведення діалогів та імпортувати ці діалоги з інших сервісів та месенджерів, в тому числі і вже існуючої на дистанційному навчанні скриньки повідомлень. Також, в подальшому, можна буде експортувати повідомлення месенджера в найпоширеніших форматах та використовувати їх у інших месенджерах, у яких наявний імпорт повідомлень, що дасть змогу використовувати набуті результати комунікації поза межами розроблюваного месенджера.

Потреба замінити стару поштову скриньку на новий месенджер виникла вже давно, так як недоліки застарілої поштової скриньки дистанційного навчання очевидні а її використання сприймається студентами як «крайній випадок», коли інші варіанти комунікації недоступні або немає контактів реципієнта. Основними перевагами використання зручнішого і якіснішого способу комунікації між користувачами є:

- Підвищення продуктивності студентів;
- Швидкий обмін інформацією та розсилка навчального матеріалу і важливих повідомлень;
- Доступне навчання;
- Зворотній зв'язок з викладачами.

Важливість покращення якості і доступності дистанційного навчання важко переоцінити так як воно є надзвичайно важливим інструментом навчання під час надзвичайних ситуацій, карантину, навчання на канікулах, а також заочному та самостійному вивченні матеріалу.

Література

1. Інформаційна сторінка сайту дистанційного навчання Тернопільського національного технічного університету імені Івана Пулюя: [Електронний ресурс] – Режим доступу до ресурсу: <https://dl.tntu.edu.ua/about.php>
2. Документація системи управління навчанням ATutor : [Електронний ресурс] – Режим доступу до ресурсу: <https://atutor.github.io/atutor/features.html>

Додаток В
Диск з електронними матеріалами