РЕФЕРАТ

Розрахунково-пояснювальна записка складається з 9 розділів:

- вступ в даному розділі описується сутність розробки, її актуальність.
 Складається з 2 сторінок;
- призначення, постановка задачі та огляд програмних аналогів і літератури у цьому розділі описано призначення, постановка задачі, проведений огляд аналогів і літератури, а також аналіз існуючих систем. Складається з 30 сторінок;
- зовнішнє та логічне проектування у цьому розділі проведений огляд вхідних і вихідних даних, формалізація задачі, розробка структур файлів і структури правил переходу. Складається з 5 сторінок;
- внутрішнє проектування у цьому розділі приводиться опис об'єктноорієнтованого проектування, проектування інтерфейсу користувача, ескізи форм, проектування динаміки системи, вибір мови програмування. Складається з 21 сторінки;
- тестування та налагодження включає в себе вибір об'єкту та методів тестування, результати тестування та налагодження програми.
 Складається з 5 сторінок;
- безпека праці користувача при роботі на ЕОМ. Складається з 12 сторінок;
- висновки. Складається з 2 сторінок;
- список літератури включає в себе бібліографічний список використаної літератури. Складається з 1 сторінки;
- додатки містить технічне завдання і робочий проект.

Кількість таблиць: 23.

Кількість рисунків: 56.

Інв. № подл.	dama	Замксть.інв. №	IHB.N

Підпис і дата

№ дубл.

Зм	Лист	№ докум.	Підпис	Дата

3MICT

	1 ПРИЗНАЧЕННЯ, ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ПРОГРАМНИХ
	АНАЛОГІВ ТА ЛІТЕРАТУРИ
	1.1 Призначення та область застосування
	1.2 Постановка задачі
	1.3 Огляд програмних аналогів
	1.3.1 Огляд програмного продукту GEGI
	1.3.2 Огляд програмного продукту LeaderTask.ru
	1.3.3 Огляд програмного продукту Schools.by
	1.3.4 Огляд програмного продукту Shkolnaya-Karta.ru25
	1.4 Огляд літератури
	1.4.1 Основи концепції MVC
	1.4.2 Основні принципи ООП
	1.4.3 Проектування баз даних та нормалізація
за	1.4.4. Основи роботи з AJAX
Підпис і дата	2 ЗОВНІШНЕ ТА ЛОГІЧНЕ ПРОЕКТУВАННЯ
Підпи	2.1 Зовнішнє проектування
	2.1.1 Вхідні данні
Ул.	2.1.2 Вихідні дані
№ дубл.	2.1.3 Формалізація задачі
Інв.№	3 ВНУТРІШНЄ ПРОЕКТУВАННЯ
§. №	3.1 Вибір мови програмування та середовища розробки
ть.ін	3.2 Вибір інструментів та парадигми проектування
Замксть.інв. №	3.3 Об'єктно-орієнтоване проектування
	3.4 Визначення призначень об'єктів за допомогою карток CRC51
i dame	3.5 Проектування інтерфейсу користувача
Підпис і дата	4 ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ ПРОГРАМИ
	4.1 Тестування методом «білої шухляди»
Эл. —	
в. <i>№ подп.</i>	Аркуи
89	0095.110577.ДП.2015.001

Зм.. Лист

№ докум.

Підпис Дата

4.2 Тестування методом «чорної шухляди»	67
4.3 Налагодження програми	67
5 ОХОРОНА ПРАЦІ	70
5.1 Аналіз шкідливих та небезпечних виробничих факторів	70
5.2 Організаційні і технічні заходи щодо захисту працівників від шкідливі	их
і небезпечних факторів	74
5.3 Норми пожарної безпеки	76
5.4 Вимоги безпеки праці	79
ВИСНОВКИ	82
ЛІТЕРАТУРА	85
ДОДАТКИ	86

етеб і эпібіП	ממומס ו סמוומ	
He No Aven		
Sawkrms ing No		
Підпис і дата	ממוס ו	
TOU NO AH		

Зм	Лист	№ докум.	Підпис	Дата

ВСТУП

У наші часи наука і техніка дуже швидко розвиваються та проникають в усі сфери нашого життя. Усе навколо автоматизується та тим самим спрощує життя людям. Розвинені навчальні заклади, наприклад, у США та Європі, не є винятком. Весь учбовий процес в них автоматизований. Розпочинаючи з ведення журналів, відомостей, календарів, бухгалтерських операцій, тощо, закінчуючи дистанційним навчанням.

Навіть важко уявити наскільки ці системи спрощують учбовий процес як студентам, так і співробітникам учбових закладів. Без такої системи студентам потрібно буде витрачати багато зайвого часу для того щоб знайти якусь необхідну їм інформацію (розклад занять, оцінки та відвідування, консультації викладачів, свій фінансовий баланс, тощо). А з цією системою уся необхідна їм інформація буде знаходитися завжди під рукою. Співробітникам також буде дуже зручно переглядати та опрацьовувати усю інформацію дистанційно.

Завдяки усім цим перевагам студенти та співробітники збережуть багато своїх сил та часу, та зможуть зосередитись на найголовнішому – навчальному процесі.

Саме тому було прийняте рішення розробити таку систему для наших університетів. Вона ϵ більш простою ніж її аналоги, але все одно може принести велику користь. Та коштуватиме вона значно менше.

Програма виконує наступні функції:

- створення і налаштування структури університету: факультети, кафедри, спеціальності та групи;
- створення користувачів цієї системи;
- створення і налаштування предметів, розкладу, завдань, тощо;
- створення і налаштування студентів, надання доступу до системи;
- збереження та опрацювання фінансових операцій студентів;

/HB	нв. № подл.	одл.	Підпис
Зм			
_ -			

№ докум.

Підпис

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

dama

збереження та опрацювання навчальної інформації студентів та викладачів: завдання, розклад занять, оцінки, відвідування, консультації, тощо.

Ця програма розроблена у вигляді веб-сайту. Тому працівники університету та студенти зможуть користуватися нею з будь-якого персонального комп'ютера або мобільного приладу, який має вихід в інтернет.

Підпис і дата						
Інв. № дубл.						
Замксть.інв. №						
Підпис і дата						
Інв. № подл.	Зм Лист	№ докум.	Підпис	Дата	0095.110577.ДП.2015.001	<i>Аркуш</i> 8

1 ПРИЗНАЧЕННЯ, ПОСТАНОВКА ЗАДАЧІ ТА ОГЛЯД ПРОГРАМНИХ АНАЛОГІВ ТА ЛІТЕРАТУРИ

1.1 Призначення та область застосування

Областю застосування системи ε організація учбового процесу будь-якого вищого навчального закладу. Та при необхідності можна переробити систему під школи, ліцеї, гімназії, тощо.

1.2 Постановка задачі

Необхідно розробити програмний продукт для організації учбового процесу вищого навчального закладу. Програмний продукт повинен надати користувачу можливість:

- створення і налаштування структури університету: факультети, кафедри, спеціальності та групи;
- створення користувачів цієї системи;
- створення і налаштування предметів, розкладу, завдань, тощо;
- створення і налаштування студентів, надання доступу до системи;
- збереження та опрацювання фінансових операцій студентів;
- збереження та опрацювання навчальної інформації студентів та викладачів: завдання, розклад занять, оцінки, відвідування, консультації, тощо.

1.3 Огляд програмних аналогів

Існує небагато програм, які надають такі ж самі функції, що і ця система. Та загалом вони є платними, і їх немає у відкритому доступі. Але в мене є доступ до системи «GEGI», яка впроваджена у декількох розвинутих навчальних закладах. Також ми розглянемо декілька інших програм, у яких є демо-версії: Schools.by, LeaderTask.ru, Shkolnaya-Karta.ru.

1.3.1 Огляд програмного продукту GEGI

Проект виконаний у вигляді веб-сайту.

Зм	Лист	№ докум.	Підпис	Да
	Зм	Зм Лист	Зм Лист № докум.	Зм Лист № докум. Підпис

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Тідпис і дата

0095.110577.ДП.2015.001

Щоб почати роботу з системою необхідно авторизуватись (рис. 1.1) за допомогою власного Email адресу та паролю. Користувач, який не авторизувався, не буде мати доступу ні до якої інформації у системі.

Gurnick Academy of Medical Arts

	User
Login or Email	ylukin@gurnick.edu
Password	•••••
Login For	got Password

Рисунок 1.1 – Форма авторизації у систему

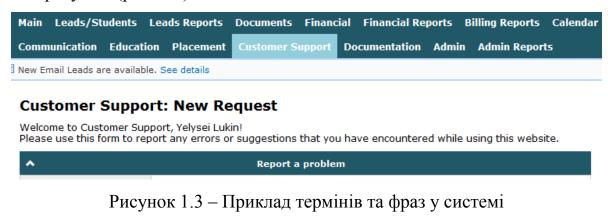
Якщо користувач забув пароль він може відновити його за допомогою функції системи та власної почти, на який зареєстровано його профіль у системі.

Введені у систему дані завжди перевіряються, а якщо користувач ввів їх неправильно, то система видає йому зрозуміле повідомлення про його помилку та навіть підсвічує поле, яке треба виправити (рис. 1.2).

Wrong login of	r password.
	User
Login or Email 🕕	ylukin@gurnick.edu
Password ①	•••••
Login Forgo	t Password

Рисунок 1.2 – Повідомлення про помилку

Інтерфейс сайту на англійській мові, але уся термінологія та фрази дуже прості та зрозумілі (рис. 1.3).



Зм	Лист	№ докум.	Підпис	Дата

Підпис і дата

Інв.№ дубл

₹

Замксть.інв.

Підпис і дата

Пе. № подл

0095.110577.ДП.2015.001

Аркуш

10

У цій системі можна зручно вести журнали оцінок та відвідувань (рис. 1.4-1.5).

^																		
ID	Student (Last, First)	Stage of Pipeline in Group	HW Ch. 16, Max: 5	HW Ch. 19, Max: 5	HW Ch. 20, Max: 5	HW CH. 7, Max: 5	HW Ch. 8, Max: 5	HW Ch. 18, Max: 5	HW Ch. 9, Max: 5	HW Ch. 17, Max: 5	HW Ch. 40, Max: 5	HW Ch. 41, Max: 5	HW Ch. 21, Max: 5	HW Ch. 14, Max: 5	HW Ch. 15, Max: 5	HW Ch. 22, Max: 5	HW Ch. 23, Max: 5	HW Ch. 37, Max: 5
S44370	☑ C Abreu, Kati 📵	ACTIVE	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
S42495	🛭 Bailey, Erica	ACTIVE	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
S13320	Beverly, Breona	ACTIVE	5	5	5	5	5	2	5	5	5	5	5	5	5	5	5	5
S29326	🛭 Biazon, Mary Ann	ACTIVE	5	5	5	5	5	5	3	5	5	5	5	5	5	5	5	5
S44042	□ □ Blake, Lauren	ACTIVE	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
S37851	🗵 Bun, Kathy	ACTIVE	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5
S35819	🛭 Carolino, Johnrey	ACTIVE	5	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5
S32683	🛭 Cuevas, Janet	ACTIVE	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
S43581	QC Diaz, Ashley	ACTIVE	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
S21647	🛭 Farrell, Heather	ACTIVE	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
S26620	🛭 Flores, Kristianne	ACTIVE	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
S42361	☑ C Goulding-Javier, Stephen	ACTIVE	5	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5

Рисунок 1.4 – Журнал оцінок

*																	Stude	nts Att	endanc	e
						Jul 2	2014							Aug	2014					
ID	Student	Stage of Pipeline in Group	08 Tue 05:00 PM Dur. 04:00	10 Thu 05:00 PM Dur. 04:00	15 Tue 05:00 PM Dur. 04:00	17 Thu 05:00 PM Dur. 04:00	22 Tue 05:00 PM Dur. 04:00	24 Thu 05:00 PM Dur. 04:00	29 Tue 05:00 PM Dur. 04:00	31 Thu 05:00 PM Dur. 04:00	05 Tue 05:00 PM Dur. 04:00	07 Thu 05:00 PM Dur. 04:00	12 Tue 05:00 PM Dur. 04:00	14 Thu 05:00 PM Dur. 04:00	19 Tue 05:00 PM Dur. 04:00	21 Thu 05:00 PM Dur. 04:00	26 Tue 05:00 PM Dur. 04:00	28 Thu 05:00 PM Dur. 04:00	02 Tue 05:00 PM Dur. 04:00	04 Thu 05:00 PM Dur. 04:00
S44370	☑ C Abreu, Kati 📵	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S42495	🗵 Bailey, Erica	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	L	Р
S13320	🗵 Beverly, Breona	ACTIVE	Р	Р	Р	Р	PM ①	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S29326	🛭 Biazon, Mary Ann	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S44042	© Blake, Lauren	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S37851	🗵 Bun, Kathy	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S35819	🙎 Carolino, Johnrey	ACTIVE	Р	Р	Р	Р	L 🕕	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S32683	🗵 Cuevas, Janet	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S43581	© □ Diaz, Ashley	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S21647	🛭 Farrell, Heather	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S26620	🛭 Flores, Kristianne	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	PM
S42361	B ⊆ Goulding-Javier, Stephen	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	L
S44747	🗵 🗆 Hernandez, Gregorio	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S43085	🗵 Isquirdo, Briana	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S25218	🛭 Kaur, Harjeet	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S34598	🛭 Kaur, Sukhvir	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р
S40990	☐ Khin, Jenny	ACTIVE	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р	Р

Рисунок 1.5 – Журнал відвідувань

Також великим плюсом програми ϵ те, що звіти про навчання можна завантажувати у форматі Excel та PDF (рис. 1.6-1.7).

Зм.	. Лист	№ докум.	Підпис	Дата

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

0095.110577.ДП.2015.001

Аркуш

11

Program: Vocational Nurse

Student Name: Nevarez, Roxana

Student ID: 12737/S306

Date Completed: 06/22/12 Official Graduation Date: 08/13/12

Prerequisites Date Entered: N/A

Official Completion Date: 06/20/11

COURSE NUMBER	TITLE	CLOCK HOURS	GRADE
VN 001	Essential Medical Bioscience	80.00	В

Program Date Entered: N/A

Official Completion Date: 06/22/12

Groups: SM-VN-2011-25

Module Dates	COURSE NUMBER	TITLE	CLOCK HOURS	CREDIT HOURS	GRADE
N/A	VN 100	Fundamental of Nursing	96.00	9.50	В
	VN 110	Anatomy and Physiology	56.00	5.50	В
	VN 120	Clinical Nutrition	32.00	3.00	A
	VN 130	Clinical Lab I	120.00	6.00	P
N/A	VN 200	Medical/Surgical Nursing I	88.00	8.50	С
	VN 210	Pharmacology I	40.00	4.00	В
N/A	VN 300	Medical/Surgical Nursing II	96.00	9.50	В
	VN 310	Pharmacology II	48.00	4.50	В
N/A	VN 400	Obstetric Nursing	44.00	4.00	В
	VN 410	Pediatric Nursing	44.00	4.00	В
	VN 420	Psychiatric Nursing	32.00	3.00	В

Program Total Clock Hours/Credit Hours: Total Quarter Hours/Credits Completed 696.00 61.50 696 61.50

Total Program GPA:

Підпис і дата

Інв.№ дубл.

⋛

Замксть.інв.

Підпис і дата

Інв. № подл.

eteu

Рисунок 1.6 – PDF документ з фінальними оцінками студента

Student: Shaw, Marissa
Group: SM-UTPR-2010-7_8
Course Type: Prerequisites

		Sessions					Hours					
Courses	Present	Absent	Late	Present "M"	Total Program Schedule	Total Scheduled as of Today		Absent	Late	Present "M"	Total Program Schedule	Total Scheduled as of Today
DMS P01 Ultrasound Basic Physics and	12	0	0	0	12	12	48:00	00:00	00:00	00:00	48:00	48:00
DMS P02 Ultrasound Cross-Sectional Anatomy	8	0	0	0	8	8	32:00	00:00	00:00	00:00	32:00	32:00
Total	20	0	0	0	20	20	80:00	00:00	00:00	00:00	80:00	80:00
Number of Hours Attended, Made-up, Outside	umber of Hours Attended, Made-up, Outside											80:00

Group: SM-UT-2010-8 Course Type: Didactic

				Sessions	1		Hours					
Courses	Present	Absent	Late	Present "M"	Total Program Schedule	Total Scheduled as of Today	Present	Absent	Late	Present "M"	Total Program Schedule	Total Scheduled as of Today
DMS 001 Ultrasound Physics and Instrumentation	10	0	-0	0	10	10	40:00	00:00	00:00	00:00	40:00	40:00
DMS 001L Ultrasound Physics and Instrumentation	15	0	0	0	15	15	60:00	00:00	00:00	00:00	60:00	60:00
DMS 003 Obstetrics and Gynecology Ultrasound	39	0	0	1	40	40	156:00	00:00	00:00	04:00	160:00	160:00
DMS 003L Obstetrics and Gynecology Ultrasound	52	0	1	2	55	55	228:00	00:00	04:00	08:00	240:00	240:00
DMS 004 Vascular Ultrasound	16	0	0	0	16	16	64:00	00:00	00:00	00:00	64:00	64:00
DMS 004L Vascular Ultrasound Laboratory	24	0	0	0	24	24	96:00	00:00	00:00	00:00	96:00	96:00
DMS PC Patient Care for the Ultrasound	4	0	0	0	4	4	12:00	00:00	00:00	00:00	12:00	12:00
Total	160	0	1	3	164	164	656:00	00:00	04:00	12:00	672:00	672:00
Number of Hours Attended, Made-up, Outside	umber of Hours Attended, Made-up, Outside											672:00

Рисунок 1.7 – PDF документ з підсумком відвідувань студента

Зм	Лист	№ докум.	Підпис	Дата	

0095.110577.ДП.2015.001

Також у системі багато звітів щодо студентів та фінансів (рис. 1.8- 1.9).



Рисунок 1.8 – Звіт про фінансовий баланс студентів

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

^		Le	ads/Months			
Month ↓	Registered New Email Leads	All Students	Enrolled Students	# of Appointments	# of No Shows	# of Drops
January	3,866	4,916	428	4,415	2,443	211
February	2,864	3,474	321	4,755	2,214	97
March	3,117	3,731	355	5,356	2,590	150
April	2,814	3,548	328	4,548	2,485	207
May	3,070	4,574	496	4,806	2,487	131
June	2,772	4,327	416	4,566	2,526	135
July	3,030	4,651	440	4,240	2,377	162
August	3,219	5,401	447	5,212	2,729	91
September	3,055	4,353	377	5,853	2,885	145
October	2,671	3,938	310	4,669	2,532	337
November	2,539	3,591	304	3,473	1,940	120
December	2,539	3,033	233	3,213	1,822	138
	35,556	49,537	4,455	55,106	29,030	1,924

Рисунок 1.9 – Звіт про кількість зареєстрованих студентів

			1			-
						Аркуш
					0095.110577.ДП.2015.001	
3м	Лист	№ докум.	Підпис	Пата	, ,	13
O.	7140111	712 00Nym.	1 1101140	дата		

① 1	lew	Subscribe for All Updates		
^				Topics
Pag	es: 1	2 3 4 5 → — Rows 1 to 50 of 247		
		Subject	Groups	Courses
•	0	NCLEX Due Dates Extended this week	MO-VN- 2013-15,	VN 300 - Medical/Surgical Nursing II
•	0	Prosthetics wired to the brain!	SM-PTA- 2014-1	
•	0	Midterm Pointers	SM-PTA- 2014-1	PTA 120 - Clinical Kinesiology with Lab
•	0	Chapter 4 Closed tipic.	SM-PTA- 2014-1	PTA 120 - Clinical Kinesiology with Lab
•	2	Wrist quiz 2 new comments. Closed tipic.	SM-PTA- 2014-1	PTA 120 - Clinical Kinesiology with Lab
•	0	Monday's quiz about elbow and forearm Closed tipic.	SM-PTA- 2014-1	PTA 120 - Clinical Kinesiology with Lab
•	0	Test email	CO-VN- 2013-25	VN 200 - Medical/Surgical Nursing I

Рисунок 1.10 – Теми форуму

У цій системі ε календар (рис. 1.11) усіх подій (нарад користувачів, зустрічей зі студентами, тощо).

	Sun 04/26/15	Mon 04/27/15	Tue 04/28/15	Wed 04/29/15	
all-day				A No Time User: Maria Alfonso (Inactive) Mclain, Maryanna Raminder	
7am					O7:00 AM User: Michellie Sanchez User Event send out time sheets
3am					
9am					O9:00 AM User: Michellie Sanchez Gomes, Guadalupe V Reminder
10am					
11am		▲ 11:00 AM		11:00 AM User: Daniel Villagomez User Event FA Meetting	
noon					
1pm				O1:00 PM O1:00 PM User: Cynthia Pukatch User Event MOD	

Рисунок 1.11 – Календар подій

При виконанні певних дій система відправляє повідомлення користувачам та студентам у вигляді Email та SMS (рис. 1.12).

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

^		Edit Notifications
Enabled	Can't Unsubscribe	Title
Custome	r Support - Requests	5
		Receive emails about new Requests (own)
		Receive emails about new Requests (students in own campuses)
		Receive emails about new Requests (students in all campuses)
		Receive emails about new Requests (employee in own campuses)
		Receive emails about new Requests (employee in all campuses)
V	V	Receive emails about Request changes (you are submitter, assignee or watcher)
		Receive emails about Request changes (students is submitter in own campuses)
		Receive emails about Request changes (students is submitter in all campuses)
		Receive emails about Request changes (employee is submitter in own campuses)
		Receive emails about Request changes (employee is submitter in all campuses)

Рисунок 1.12 – Налаштування повідомлень

Також великою перевагою цієї системи ε можливість відправляти заявки до адміністрації сайту щодо проблем та питань, що виникли під час роботи з системою (рис. 1.13).

Customer Support: New Request

Welcome to Customer Support, Yelysei Lukin! Please use this form to report any errors or suggestions that you have encountered while using this website.

^	Report a problem
Submitter	Yelysei Lukin
Campus	
Туре	Y
Priority	Y
Subject (Short Description)	
Suggestion or Issue Report	.ti
Watchers	▼ ③
Attach Files	Browse No file selected. Max files size: 10.00 MB
Send Send	

Рисунок 1.13 – Створення заявки щодо проблеми чи питання

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв. № дубл.

Замксть.інв. №

Підпис і дата

Також у системі можна зберігати документи студента та навіть підписувати їх за допомогою електронного підпису (рис. 1.14-1.16).

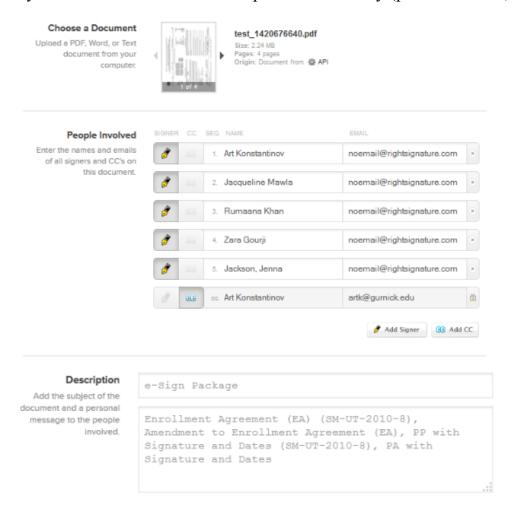


Рисунок 1.14 – Створення електронного документу

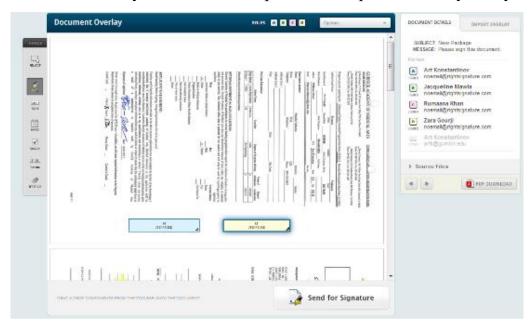


Рисунок 1.15 – Вибір місця для підписів

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв.№ дубл

윙

Замксть.інв.

Підпис і дата

№ подл.

/HB.

0095.110577.ДП.2015.001

Рисунок 1.16 – Підписання документа

За допомогою Google Maps API на сайті можна подивитися розташування офісів та кампусів університету та інших споруд (рис. 1.17).

Clinical Site

Підпис і дата

Інв.№ дубл

Замксть.інв. №

Підпис і дата

Інв. № подл.

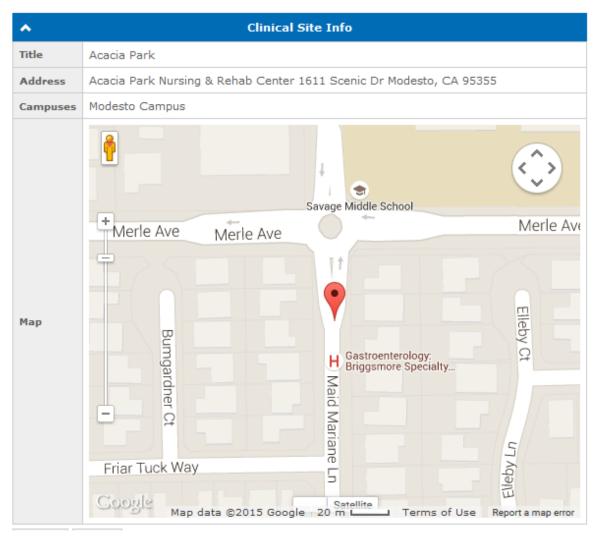


Рисунок 1.17 – Перегляд місця знаходження об'єкту на карті До недоліків цієї програми можна віднести:

- деяку різнорідність фраз та термінів та невеликі помилки;
- часткову відсутність АЈАХ (сторінку постійно доводиться перезавантажувати навіть щоб побачити помилки при вводі даних);

Зм	Лист	№ докум.	Підпис	Дата	

0095.110577.ДП.2015.001

- довгий час відклику на деяких сторінках через дуже велику кількість даних;
- дуже заплутану систему прав та ролей. По-перше, у кожного користувача може бути будь-яка кількість ролей (рис. 1.18). По-друге, у ролі декілька сотень різних прав і часом дуже важко знайти яке за що відповідає (рис. 1.19). По-третє, кожному користувачу можна переналаштувати права із ролей особисто під нього (рис. 1.20).

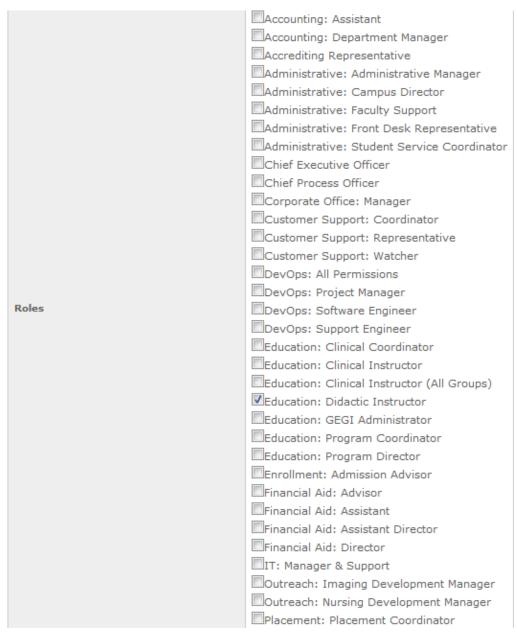


Рисунок 1.18 – Налаштування ролей користувача

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

■Vie	w Student Email Subscription Status
Edi	t Student Email Subscription Status
Car	n Ignore limitation Student Group Official Date of Withdrawal Or Expulsion = Date from Withdrawa
Car	n Ignore limitation Student Group Official Completion Date = Scheduled Completion Date from EA
Car	n Ignore limitation Student Group Official Start Date = Scheduled Official Start Date from EA
Car	n Ignore limitation Student Group Official Graduation Date = Checklist Last Passed Date
Ca	n change Program Inquire for students created from New Email Leads
Ch	ecklist:
Vie	w Checklist Template in Admin
Edi	t Checklist Template in Admin
De	lete Checklist Template in Admin
Vie	w group's checklist
V Edi	t group's checklist
√ Vie	w student's checklist
V Edi	t student's checklist in Any Stage
V Edi	t student's checklist in ACTIVE Stage
V Edi	t student's checklist in COMPLETED Stage
Edi	t student's checklist in DROPPED and DROPPED Pending Stages
VEdi	t student's checklist in GRADUATED Stage
En	rollment:
V Ada	d
Vie	w (all students)
Vie	w (own students)
V Edi	t
VDe	ete
	Рисунок 1.19 – Налаштування прав ролі
	User Permissions
Enable	Title
Custom	eer Support - Requests
V	Add Request
	View/Manage Pequests (own: you are submitter assigned watcher)

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

	User Permissions					
Enable	Title					
Custom	ner Support - Requests					
V	Add Request					
V	View/Manage Requests (own: you are submitter, assignee, watcher)					
	View/Manage Requests (students in own campuses)					
	View/Manage Requests (students in all campuses)					
	View/Manage Requests (employee in own campuses)					
	View/Manage Requests (employee in all campuses)					
	View Requests Ticket Number					
	Edit Requests Ticket Number					
	Can set Request's Assignee as Submitter					
	Can Assign Request (assignee = employee in own campuses)					
	Can Assign Request (assignee = employee in all campuses)					
	Can be Assigned to Requests					
V	Can be added to Requests as a Watcher					
	Can change Request Statuses ignoring rules					
	Can add Watchers to Request					
	Can delete Request's Watchers					
Leads/	Students					
	Add					

		Leads/	Students					
				Add				
					Рис	унок	1.20 – Налаштування прав користувача	
	Зм	Лист	Nº õ	докум.	Підпис	Дата	0095.110577.ДП.2015.001	Аркуш 19

LeaderTask.ru — клієнтське програмне забезпечення для організації процесу навчання студента. У ньому пропонуються наступні функції:

Проекти дозволять розсортирувати інформацію по необхідним критеріям (рис. 1.21);

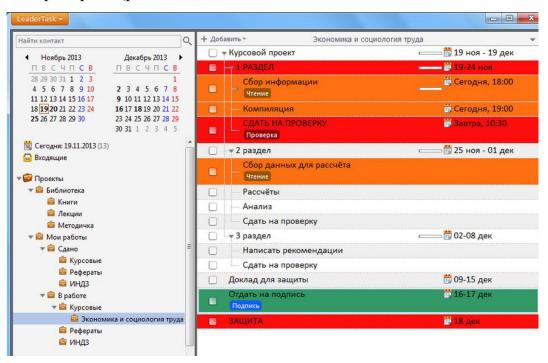


Рисунок 1.21 – Проекти студента

Категорії дозволяють ефективно систематизувати усі завдання,
 розподілив їх за визначеними критеріями (рис. 1.22);



Рисунок 1.22 – Проекти студента

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв.№ дубл

୬

Ή8.

Замксть.

Підпис і дата

№ подп.

/HB.

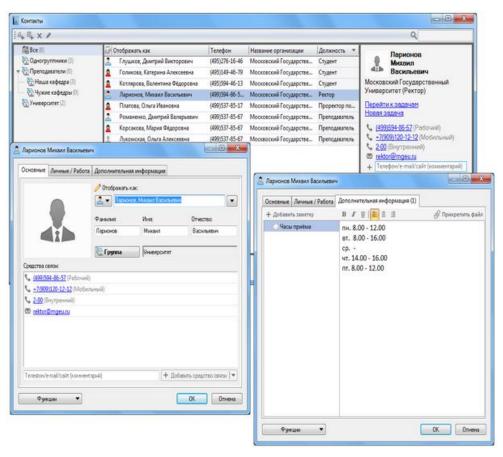


Рисунок 1.23 – Контакти

Маркери дозволять виділяти різні типи завдань різними кольорами,
 що допоможе швидше орієнтуватись у листі своїх завдань (рис. 1.24).

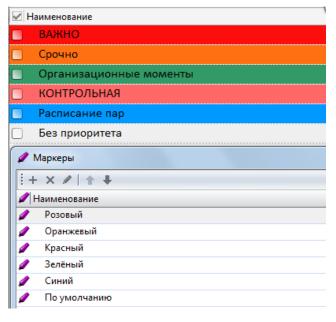


Рисунок 1.24 – Маркери

					_
Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв.№ дубл

୬

Ή8.

Замксть.

Підпис і дата

Ле подл

/HB.

0095.110577.ДП.2015.001

1.3.3 Огляд програмного продукту Schools.by

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Iнв. № подл.

Schools.by – сайт для організації учбового процесу навчального закладу. Його функції більш налаштовані для шкіл, але й для вищих навчальних закладів їх також можна використовувати.

Однією з особливостей проекту ϵ можливість батьками переглядати інформацію про навчання своїх дітей (рис. 1.25).

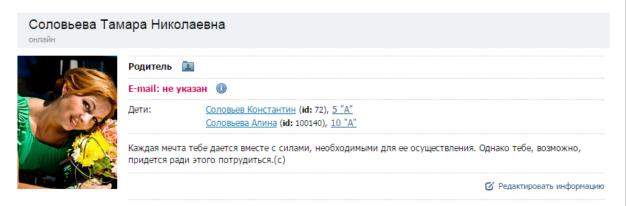


Рисунок 1.25 – Профіль батьків

Ще однією перевагою проекту ϵ зручні розклади занять, дзвінків та канікул (рис. 1.26-1.28).

Расписание звонков						
	1 смена			2 смена		
Урок	Время		Урок	Время		
1.	08:00 - 08:45		1.	12:00 - 12:45		
2.	09:00 - 09:45		2.	13:00 - 13:45		
3.	10:00 - 10:45		3.	14:00 - 14:45		
4.	11:00 - 11:45		4.	15:00 - 15:45		
5.	12:00 - 12:45		5.	16:00 - 16:45		
6.	13:00 - 13:45		6.	17:00 - 17:45		
7.	14:00 - 14:45		7.	18:00 - 18:45		
8.	15:00 - 15:45		8.	19:00 - 19:45		

Рисунок 1.26 – Розклад дзвінків						
					0005 110555 HT 2015 001	Аркуш
Зм	Лист	№ докум.	Підпис	Дата	0095.110577.ДП.2015.001	22
	'		•			

Расписание четвертей 2014-2015 учебного года

Четверть	Начало		Конец
1-я	1 сентября 2014 г.	_	25 октября 2014 г.
2-я	3 ноября 2014 г.	_	27 декабря 2014 г.
3-я	12 января 2015 г.	_	21 марта 2015 г.
4-я	30 марта 2015 г.	_	30 мая 2015 г.

Расписание каникул 2014-2015 учебного года

- осенние 8 дней с 26.10.2014 по 02.11.2014;
- зимние 15 дней с 28.12.2014 по 11.01.2015;
- весенние 8 дней с 22.03.2015 по 29.03.2015 включительно;
- летние 93 дня с 31.05.2015 по 31.08.2015 включительно.

Для учащихся, завершивших обучение на II ступени общего среднего образования, – 79 дней с 14.06.2015 по 31.08.2015 включительно.

Для учащихся I-II классов в III четверти проводятся дополнительные зимние каникулы с 16.02.2015 по 22.02.2015 продолжительностью 7 дней.

Рисунок 1.27 – Розклад четвертей та канікул

<u>5 "A"</u>

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

	Время	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
1.	08:00 - 08:45	Физ. к. и зд.	Англ. яз. (Мальчики) Англ. яз. (Девочки)	Бел. лит.	Матем.	Матем.	-
2.	09:00 - 09:45	Англ. яз. (Мальчики) Англ. яз. (Девочки)	Всемир. ист.	Бел. яз.	Всемир. ист.	Рус. лит.	_
3.	10:00 - 10:45	Матем.	Рус. яз.	Физ. к. и зд.	Англ. яз. (Девочки) Англ. яз. (Мальчики)	изо	-
4.	11:00 - 11:45	Рус. лит.	Матем.	Матем.	Рус. яз.	Труд (Мальчики) Труд (Девочки)	_
5.	12:00 - 12:45	Чел. и Мир	Бел. яз.	Рус. яз.	Бел. яз.	Физ. к. и зд.	_
6.	13:00 - 13:45	_	Бел. лит.	_	_	_	_

<u>5 "Б"</u>

Не заполнено

Рисунок 1.28 – Розклад занять

					0095.110577.ДП.2015.001	Аркуш
Зм	Лист	№ докум.	Підпис	Дата		23

Учительская

Конечно, за многолетнюю историю в школе работало множество замечательных педагогов. Чтобы получить неформальное представление о наших учителях, мы публикуем здесь их собственные заметки о себе или о своих интересах. Школа полностью обеспечена педагогическими кадрами, отсутствует текучесть кадров. Сейчас в школе творчески работает стабильный педагогический коллектив в составе 53 учителей, способный к динамичному развитию.



<u>Гашевский Георгий</u> Дмитриевич

Підпис і дата

Інв.№ дубл

⋛

Замксть.інв.

Підпис і дата

Iнв. № подл



Горький Олег Степанович



Иванова Тамара Александровна Кл. руководитель 5 "А"



Петров Алексей Николаевич

Рисунок 1.29 – Список вчителів

Гашевский Гео заходил 22 апреля в 11:2		иевич		
	Учитель (учит	ель химии и биологии.)		
60 M	Телефон: 245-87-91			
	Дата рождения	: 19 апреля 1960		
N/A	Преподает:	Всемирная История в <u>10-ом "А"</u> <u>Физика</u> в <u>5-ом "А", 10-ом "А"</u> Математика в 5-ом "А", 10-ом "А"		
		бряной медалью школу №705.Окончил с отличием Белорусский Госуд сти" Биолог-эколог. Преподаватель биологии и экологии".	дарственный Университет	

Рисунок 1.30 – Профіль вчителя

Ще однією дуже корисною функцією є архів файлів: книг, атласів, тощо. Це дуже корисно для учнів, тому що їм не потрібно покупати паперові книжки та кожен день носити їх із собою. Він може просто завантажити необхідні для цього року матеріали на свій телефон, або планшет, або ноутбук, та завжди мати їх під рукою (рис. 1.31).

Зм	Лист	№ докум.	Підпис	Дата

0095.110577.ДП.2015.001

Файловый архив

Здесь можно разместить множество полезных книг, нужных программ и т.д.

Рисунок 1.32 – Список ролей

На цьому сайті реалізовано багато функцій щодо організації навчального процесу учнів, наприклад, розклад, учбовий план, щоденник, чат, об'яви, тощо.

Зм	Лист	№ докум.	Підпис	Дата	

Директор

Підпис і дата

Iнв. № подл.

Комбинат питания

Представитель КП в школе

0095.110577.ДП.2015.001

До плюсів даного продукту можна віднести зручний перегляд відвідувань та оцінок учнів (рис. 1.33-1.34) та можливість легко робити різні звіти (рис. 1.35).

Класс:

Підпис і дата

Інв.№ дубл

⋛

Замксть.інв.

Підпис і дата

Пе. № подл

Зм... Лист

№ докум.

Підпис

Дата

Отметки

Краткий отчет за текущую неделю 22 - 28 июня

Ребенок: Жданова Арина Александровна

Вт Чт Пт Сб Пн Ср Предмет 25 мая 26 мая 27 мая 30 мая 28 мая 29 мая Английский язык 3 устн. 4 письм. <u>Биология</u> География <u>И3О</u> 4 письм. История 5 письм. Литература **Математика** 5 устн. Рисунок 1.33 – Перегляд оцінок учня Просмотр посещений Ребенок: Жданова Арина Александровна Посещаемость за 4 четверть Всего прошло учебных дней: 99 Всего пропущенных дней: 0 По болезни: 0 Без уважительной причины: 0 По уважительной причине: 0 Примечание: Согласно данных, внесенных классным руководителем Рисунок 1.34 – Перегляд відвідувань учня Аркуш 0095.110577.ДП.2015.001

26

Nº	Название			
1.	Список учащихся	Просмотреть шаблон	Скачать пустой шаблон	Скачать заполненый шаблон
2.	Обходной лист	Просмотреть шаблон	Скачать пустой шаблон	Скачать заполненый шаблон
3.	Занятия в кружках и секциях	Просмотреть шаблон	Скачать пустой шаблон	Скачать заполненый шаблон
4.	График дежурств	Просмотреть шаблон	Скачать пустой шаблон	Скачать заполненый шаблон
5.	Занятость в летний период	Просмотреть шаблон	Скачать пустой шаблон	Скачать заполненый шаблон
6.	Выписка текущих отметок	Просмотреть шаблон		
7.	Характеристика классного коллектива	Скачать шаблон		
8.	Занятость учащихся во внеурочное время	Просмотреть шаблон	Скачать пустой шаблон	Скачать заполненый шаблон
9.	Журнал классного руководителя	Скачать шаблон		
10.	Занятия учащихся второй половины дня	Скачать шаблон		
11.	Отчет классного руководителя по воспитательной работе	Просмотреть шаблон	Скачать пустой шаблон	Скачать заполненый шаблон

Рисунок 1.35 – Генерація форм для звітів

1.4 Огляд літератури

Підпис і дата

Інв. № дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

1.4.1 Основи концепції MVC

MVC - це не шаблон проекту, це конструкційний шаблон, який описує спосіб побудови структури нашого застосування, сфери відповідальності та взаємодію кожної з частин в даній структурі [3].

Вперше вона була описана в 1979 році, звичайно ж, для іншого оточення. Тоді не існувало концепції веб додатку. Тіт Berners Lee (Тім Бернерс Лі) посіяв насіння World Wide Web (WWW) на початку дев'яностих і назавжди змінив світ. Шаблон, який ми використовуємо сьогодні, є адаптацією оригінального шаблону для веб розробки.

Шалена популярність даної структури у веб додатках склалася завдяки її включенню в два середовища розробки, які стали дуже популярними: Struts і Ruby on Rails. Ці два середовища розробки визначили шляхи розвитку для сотень робочих середовищ, створених пізніше.

Зм	Лист	№ докум.	Підпис	Дата	
					_

Ідея, яка лежить в основі конструкційного шаблону MVC, дуже проста: потрібно чітко розділяти відповідальність за різне функціонування в наших додатках:

- модель обробка даних та логіка програми;
- вид представлення даних користувачу у необхідному форматі;
- контролер обробка запитів користувача та виклик необхідних ресурсів.

Розберемо усі складові детальніше.

Контролер (Controller)

Контролер керує запитами користувача (отримуються у вигляді запитів НТТР GET або POST, коли користувач натискає на елементи інтерфейсу для виконання різних дій). Його основна функція — викликати і координувати дію необхідних ресурсів та об'єктів, потрібних для виконання дій, що задаються користувачем. Зазвичай контролер викликає відповідну модель для задачі і вибирає відповідний вид.

Модель (Model)

Модель - це дані і правила, які використовуються для роботи з даними, які представляють концепцію управління додатком. У будь-якому додатку вся структура моделюється як дані, які обробляються певним чином. Що таке користувач для програми — повідомлення або книга? Тільки дані, які повинні бути оброблені відповідно до правил (дата не може вказувати в майбутнє, Email повинен бути в певному форматі, ім'я не може бути довшим X символів, і так далі).

Модель дає контролеру представлення даних, які запросив користувач (повідомлення, сторінку книги, фотоальбом, тощо). Модель даних буде однаковою, незалежно від того, як ми хочемо представляти їх користувачеві. Тому ми вибираємо будь-який доступний вид для відображення даних.

Модель містить найбільш важливу частину логіки нашого застосування, логіки, яка вирішує завдання, з яким ми маємо справу (форум, магазин, банк,

Підпис і дата

Інв.№ дубл.

8. **N**º

тощо). Контролер містить в основному організаційну логіку для самого додатка (дуже схоже на ведення домашнього господарства).

Вид (View)

Вид забезпечує різні способи подання даних, які отримані з моделі. Він може бути шаблоном, який заповнюється даними. Може бути кілька різних видів, і контролер вибирає, який підходить якнайкраще для поточної ситуації.

Веб додаток зазвичай складається з набору контролерів, моделей і видів. Контролер може бути влаштований як основний, який отримує всі запити і викликає інші контролери для виконання дій в залежності від ситуації.

На рис. 1.36 можна побачити принцип дії шаблону MVC.

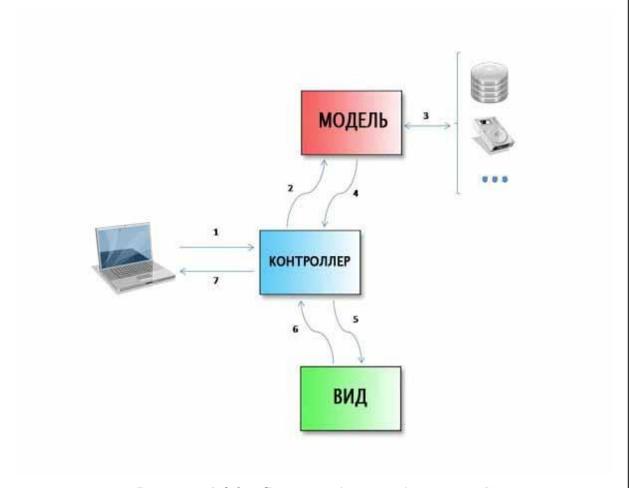


Рисунок 1.36 – Схема роботи шаблону MVC

Найбільш очевидна перевага, яку ми отримуємо від використання концепції MVC - це чіткий поділ логіки подання (інтерфейсу користувача) і логіки програми.

g					
№ подл					
/HB.					
±	Зм	Лист	№ докум.	Підпис	Дата

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Підтримка різних типів користувачів, які використовують різні типи пристроїв є спільною проблемою наших днів. Наданий інтерфейс повинен відрізнятися, якщо запит приходить з персонального комп'ютера або з мобільного телефону. Модель повертає однакові дані, єдина відмінність полягає в тому, що контролер вибирає різні види для виведення даних.

Крім ізолювання видів від логіки додатка, концепція MVC істотно зменшує складність великих додатків. Код виходить набагато більш структурованим, і, тим самим, полегшується підтримка, тестування та повторне використання рішень.

1.4.2 Основні принципи ООП

Абстракція - в об'єктно-орієнтованому програмуванні це додання об'єкту характеристик, які відрізняють його від усіх інших об'єктів, чітко визначаючи його концептуальні межі. Основна ідея полягає в тому, щоб відокремити спосіб використання складових об'єктів даних від деталей їх реалізації у вигляді більш простих об'єктів, подібно до того, як функціональна абстракція розділяє спосіб використання функції і деталей її реалізації в термінах більш примітивних функцій, таким чином, дані обробляються функцією високого рівня за допомогою виклику функцій низького рівня. Такий підхід є основою об'єктноорієнтованого програмування. Це дозволяє працювати з об'єктами, не вдаючись у особливості їх реалізації. У кожному конкретному випадку застосовується той чи інший підхід: інкапсуляція, поліморфізм або спадкування. Наприклад, при необхідності звернутися до прихованих даних об'єкта, слід скористатися інкапсуляцією, створивши так звану функцію доступу або властивість. Абстракція даних [4] - популярна і в загальному невірно визначена техніка програмування. Фундаментальна ідея полягає в поділі несуттєвих деталей реалізації підпрограми i характеристик істотних ДЛЯ використання. Цей поділ може бути вираженим через спеціальний «інтерфейс», зосереджують опис всіх можливих застосувань програми. З точки зору теорії

Iнв. № подл. Підпис і дата Замксть.інв. №

Підпис і дата

Інв.№ дубл.

множин, процес являє собою організацію для групи підмножин своєї множини. Також Закон зворотного відношення між змістом і обсягом поняття.

властивість Інкапсуляція МОВИ програмування, ЩО дозволяє користувачеві не замислюватися про складність реалізації використовуваного програмного компонента (що у нього всередині?), а взаємодіяти з ним за допомогою наданого інтерфейсу (публічних методів і членів), а також об'єднати і захистити життєво важливі для компонента дані. При цьому користувачеві тільки специфікація (інтерфейс) об'єкта. Користувач надається взаємодіяти з об'єктом тільки через цей інтерфейс. Реалізується за допомогою ключового слова: public. Користувач не може використовувати закриті дані і методи. Реалізується за допомогою ключових слів: private, protected, internal. Інкапсуляція - один чотирьох найважливіших механізмів об'єктно-3 орієнтованого програмування (поряд абстракцією, 3 поліморфізмом спадкуванням). Приховування реалізації доцільно застосовувати в наступних локалізація змін при необхідності випадках: гранична прогнозованість змін (які зміни в коді треба зробити для заданого зміни функціональності) і прогнозованість наслідків змін.

Спадкування - один з чотирьох найважливіших механізмів об'єктноорієнтованого програмування (поряд з інкапсуляцією, поліморфізмом і
абстракцією), що дозволяє описати новий клас на основі вже існуючого
(батьківського), при цьому властивості і функціональність батьківського класу
запозичуються новим класом. Іншими словами, клас-спадкоємець реалізує
специфікацію вже існуючого класу (базовий клас). Це дозволяє звертатися до
об'єктів класу-спадкоємця точно так само, як до об'єктів базового класу. Просте
спадкування: Клас, від якого відбулося спадкування, називається базовим або
батьківським (англ. Base class). Класи, які походять від базового, називаються
нащадками, спадкоємцями або похідними класами (англ. Derived class). У
деяких мовах використовуються абстрактні класи. Абстрактний клас – це клас,
що містить хоча б один абстрактний метод, він описаний в програмі, має поля,

Зм	Лист	№ докум.	Підпис	Дата	
					Τ

Підпис і дата

Інв. № дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

методи і не може використовуватися для безпосереднього створення об'єкта. Тобто від абстрактного класу можна тільки наслідувати. Об'єкти створюються тільки на основі похідних класів, успадкованих від абстрактного. Наприклад, абстрактним класом може бути базовий клас «співробітник вузу», від якого успадковуються класи «аспірант», «професор» і т.д. Так як похідні класи мають спільні поля і функції (наприклад, поле «рік народження»), то ці члени класу можуть бути описані в базовому класі. У програмі створюються об'єкти на основі класів «аспірант», «професор», але немає сенсу створювати об'єкт на основі класу «співробітник вузу».

Множинне спадкування

При множинному спадкуванні у класу може бути більше одного предка. У цьому випадку клас успадковує методи всіх предків. Переваги такого підходу в більшій гнучкості. Множинне спадкування реалізовано в С ++. З інших мов, які надають цю можливість, можна відзначити Python і Ейфель. Множинне спадкування підтримується в мові UML. Множинне спадкування - потенційне джерело помилок, які можуть виникнути через наявність однакових імен методів у предках. У мовах, які позиціонуються як спадкоємці С ++ (Java, С # та ін.), від множинного спадкоємства було вирішено відмовитися на користь інтерфейсів. Практично завжди можна обійтися без використання даного механізму. Однак, якщо така необхідність все-таки виникла, то, для вирішення використання успадкованих методів з однаковими іменами, можливо, наприклад, застосувати операцію розширення видимості — «::» — для виклику конкретного методу конкретного батька. Спроба вирішення проблеми наявності однакових імен методів у предках була зроблена в мові Ейфель, в якому при описі нового класу необхідно явно вказувати імпортовані члени кожного з успадкованих класів та їх іменування в дочірньому класі. Більшість сучасних об'єктно-орієнтованих мов програмування (С #, Java, Delphi та ін.) підтримують можливість одночасно успадковуватися від класу-предка і реалізувати методи декількох інтерфейсів одним і тим же класом. Цей механізм

Інв. № подл. Підпис і дата Замксть.інв. №

Підпис і дата

Інв.№ дубл.

Зм	Лист	№ докум.	Підпис	Дата	

дозволяє багато в чому замінити множинне спадкування - методи інтерфейсів необхідно перевизначати явно, що виключає помилки при спадкуванні функціональності однакових методів різних класів-предків.

Поліморфізм - можливість об'єктів з однаковою специфікацією мати різну реалізацію. Мова програмування підтримує поліморфізм, якщо класи з однаковою специфікацією можуть мати різну реалізацію – наприклад, реалізація класу може бути змінена в процесі спадкування. Коротко зміст поліморфізму можна виразити фразою: «Один інтерфейс, безліч реалізацій». - один з чотирьох найважливіших механізмів об'єктно-Поліморфізм інкапсуляцією програмування (поряд 3 абстракцією, орієнтованого спадкуванням). Поліморфізм дозволяє писати більш абстрактні програми і підвищити коефіцієнт повторного використання коду. Загальні властивості об'єктів об'єднуються в систему, яку можуть називати по-різному - інтерфейс, клас. Спільність має зовнішнє і внутрішнє вираження: зовнішня спільність проявляється як однаковий набір методів з однаковими іменами і сигнатурами (ім'ям методів і типами аргументів та їх кількістю); внутрішня спільність однакова функціональність методів. Її можна описати інтуїтивно або виразити у вигляді суворих законів, правил, яким повинні підкорятися методи. Можливість приписувати різну функціональність одному методу (функції, операції) (перевантаженням функцій, називається перевантаженням методу перевантаженням операцій).

1.4.3 Проектування баз даних та нормалізація

Реляційні бази даних розроблені для швидкого збереження і отримання великих обсягів інформації. Нижче наведені деякі характеристики реляційних баз даних і реляційної моделі даних [2].

Використання ключів (рис. 1.37).

Кожен рядок даних у таблиці ідентифікується унікальним "ключем", який називається первинним ключем. Найчастіше, первинний ключ це автоматично збільшуване (автоінкрементне) число (1,2,3,4 і т.д). Дані в різних таблицях

інв. № подл.					
\ <u>o</u> ⊓					
18. /					
11	Зм	Лист	№ докум.	Підпис	Дата

Підпис і дата

Інв. № дубл.

Замксть.інв. №

Підпис і дата

0095.110577.ДП.2015.001

Інв. № подл.

можуть бути пов'язані разом при використанні ключів. Значення первинного ключа однієї таблиці можуть бути додані в рядки (записи) іншої таблиці, тим самим, пов'язуючи ці записи разом.



Рисунок 1.37 – Використання ключів у БД

Використовуючи структуровану мову запитів (SQL), дані з різних таблиць, які пов'язані ключем, можуть бути обрані за один раз.

Стовпець іd в даній таблиці ϵ первинним ключем. Кожен запис має унікальний первинний ключ, часто число. Стовпець usergroup (групи користувачів) ϵ зовнішнім ключем.

Відсутність надмірності даних.

У проекті бази даних, яка створена з урахуванням правил реляційної моделі даних, кожен шматочок інформації, наприклад, ім'я користувача, зберігається тільки в одному місці. Це дозволяє усунути необхідність роботи з даними в декількох місцях. Дублювання даних називається надмірністю даних і цього слід уникати в якісному проекті бази даних.

Обмеження введення (рис. 1.38).

Використовуючи реляційну базу даних ви можете визначити який вид даних дозволено зберігати в стовпці. Ви можете створити поле, яке містить цілі числа, десяткові числа, невеликі фрагменти тексту, великі фрагменти тексту, дати і т.д.

	Field Name	Datatype		Len	Def
	id	int	•	11	
	username	varchar	•	50	
	email	varchar	•	50	
	password	varchar	•	100	
	usergroup	int	•	10	
	birthday	date	•		
*			•		

Рисунок 1.38 – Обмеження введення даних у БД

Зм	Лист	№ докум.	Підпис	Дата

Підтримка цілісності даних.

Налаштовуючи властивості полів, пов'язуючи таблиці між собою і налаштовуючи обмеження, ви можете збільшити надійність ваших даних.

Вказівки для правильного проектування реляційних баз даних зібрані в 5 груп, які називаються нормальними формами.

Нормальні форми - це рекомендації з проектування баз даних. Ви не зобов'язані дотримуватися всіх п'яти нормальних форм при проектуванні баз даних. Тим не менш, рекомендується нормалізувати базу даних в деякій мірі тому, що цей процес має ряд істотних переваг з точки зору ефективності та зручності поводження з вашою базою даних:

- у нормалізованої структурі бази даних ви можете робити складні вибірки даних відносно простими SQL-запитами;
- цілісність даних. Нормалізована база даних дозволяє надійно зберігати дані;
- нормалізація запобігає появі надмірності збережених даних. Дані завжди зберігаються тільки в одному місці, що робить легким процес вставки, оновлення та видалення даних. Є виняток з цього правила. Ключі, самі по собі, зберігаються в кількох місцях тому, що вони копіюються як зовнішні ключі в інші таблиці;
- масштабованість це можливість системи справлятися з майбутнім зростанням. Для бази даних це означає, що вона повинна бути здатна працювати швидко, коли число користувачів і обсяги даних зростають.
 Масштабованість це дуже важлива характеристика будь-якої моделі бази даних і для РСУБД.

Перша нормальна форма (1НФ).

Перша нормальна форма свідчить, що таблиця бази даних — це подання суті вашої системи, яку ви створюєте. Кожен запис в базі даних — один екземпляр сутності.

Підпис і да	Інв. № дубл.	Замксть.інв. №	Підпис і дата	№ подл.

ma

Зм	Лист	№ докум.	Підпис	Дата	

Первинний ключ: кожна таблиця має первинний ключ, що складається з найменшої можливої кількості полів.

Атомарність: поля не мають дублікатів в кожному записі і кожне поле містить тільки одне значення.

Порядок записів не повинен мати значення.

Друга нормальна форма (2НФ).

Для того, щоб база даних була нормалізована згідно з другою нормальною формою, вона повинна бути нормалізована згідно з першою нормальною формою. Друга нормальна форма пов'язана з надмірністю даних.

Надмірність даних: поля з не первинним ключем не повинні бути залежні від первинного ключа. В таблиці повинні зберігатися тільки дані, які напряму пов'язані з нею і не мають відношення до іншої сутності. Завдання другої нормальної форми — це знаходження даних, які часто дублюються в записах таблиці і які можуть належати іншій сутності.

Третя нормальна форма (3НФ).

Третя нормальна форма пов'язана з транзитивними залежностями. Транзитивні залежності між полями бази даних існують тоді, коли значення не ключових полів залежать від значень інших не ключових полів. Щоб база даних була в третій нормальній формі, вона повинна бути в другій нормальній формі.

Транзитивні залежності: не може бути транзитивних залежностей між полями в таблиці.

Третя нормальна форма свідчить, що ви не повинні зберігати дані в таблиці, які можуть бути отримані з інших (ключових) полів таблиці.

Третя нормальна форма не завжди використовується при проектуванні баз даних. Коли розробляється база даних завжди потрібно порівнювати переваги від більш високою нормальної форми в порівнянні з обсягом робіт, які потрібні для застосування третьої нормальної форми і підтримки даних у такому стані.

Замк	
Підпис і дата	
Інв. № подл.	

Підпис і дата

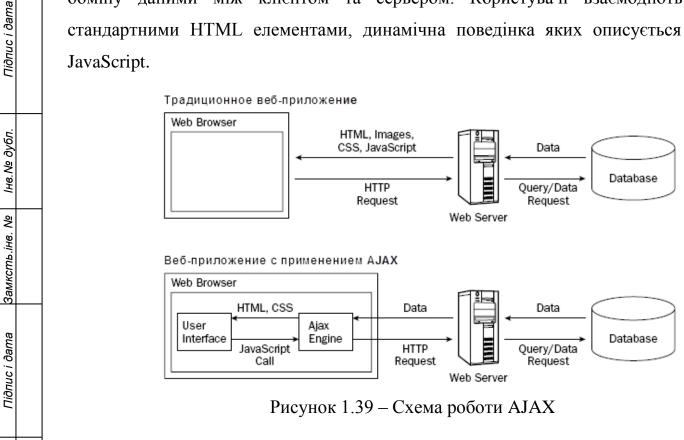
Інв. № дубл.

сть.інв. №

АЈАХ (англ. Asynchronous Javascript and XML) - спосіб побудови інтерактивних користувацьких web-додатків за допомогою фонового обміну інформацією браузера з сервером [1]. Термін АЈАХ позначив Джессі Джеймс Гаррет в 2005 році. Першими додатками, що використовують дану технологію, став сервіс карт Google Maps і поштовий клієнт Gmail.

Використання АЈАХ для розкрутки сайту дозволяє поліпшити його юзабіліті (додатки стають більш зручними і швидкими для відвідувачів), функціональність і зовнішній вигляд сторінок.

Принцип роботи (рис. 1.39) АЈАХ базується на технології звернення до сервера без перезавантаження сторінки (XTMLHttpRequest, створення дочірніх фреймів або тега <script>) або використанні DHTML, що дозволяє динамічно змінювати вміст. Формат передачі даних — XML або JSON. АЈАХ можна реалізувати в різних мовах програмування: PHP, Ruby on Rails, ASP.NET та інших. У коді web-сторінок широко використовується JavaScript для прозорого обміну даними між клієнтом та сервером. Користувачі взаємодіють зі стандартними HTML елементами, динамічна поведінка яких описується на JavaScript.



0095.110577.ДП.2015.001

Аркуш

37

Iнв. № подл

Зм..

Лист

№ докум.

Підпис

Дата

Переваги:

- економія трафіку користувача (замість оновлення всієї сторінки,
 завантажується її невелика частина, що змінилась);
- зниження навантаження на сервер. Наприклад, на сторінці особистих повідомлень форуму при виділенні користувачем прочитаних листів сервер вносить зміни в БД і відправляє скрипту клієнта відповідь про виконання операції без повторного створення сторінки та її передачі;
- прискорення реагування інтерфейсу на команди користувача.

Недоліки:

- не завжди можлива інтеграція зі стандартним набором інструментів браузера. У деяких випадках немає можливості додати у закладки потрібний матеріал;
- контент, що завантажується динамічно, не доступний пошуковим системам, тому необхідно забезпечити альтернативний доступ до вмісту ресурсу;
- неправильний облік статистики переміщення користувача на сайті;
- ускладнення контролю цілісності типів і форматів, тому що процеси форматування даних частково переносяться на сторону клієнта;
- в браузері користувача повинен бути включений JavaScript.

Підпис і дата	
Інв.№ дубл.	
Замксть.інв. №	
Підпис і дата	
. № подл.	

Зм	Лист	№ докум.	Підпис	Дата

2 ЗОВНІШНЕ ТА ЛОГІЧНЕ ПРОЕКТУВАННЯ

2.1 Зовнішнє проектування

2.1.1 Вхідні данні

Вхідними даними програми ϵ наступні дані, які користувачі вводять до системи за допомогою різних форм та таблиць:

- інформація про користувачів: ім'я, Етаіl, пароль, роль у системі, посада, запис видалено чи ні, він є викладачем чи ні;
- інформація про консультації викладачів: день тижня, час початку;
- інформація про університет: назва, скорочена назва, адрес, ім'я ректора;
- інформація про факультети: назва, скорочена назва, унікальний номер, ім'я декану, запис видалено чи ні;
- інформація про кафедри: назва, скорочена назва, назва факультету,
 ім'я завідуючого кафедрою, запис видалено чи ні;
- інформація про спеціальності: назва, назва кафедри, унікальний номер, запис видалено чи ні;
- інформація про ролі: назва та список прав;
- інформація про посади: назва;
- інформація про заняття: номер, час початку, час завершення;
- інформація про предмети: назва, загальна кількість годин, тип зарахування предмету (екзамен чи залік);
- інформація про платежі: назва, напрям платежу;
- інформація про групи: назва, рік початку навчання, поточний рік навчання, загальна кількість років навчання, назва спеціальності, ім'я куратора, навчання завершено чи ні, запис видалено чи ні;
- інформація про предмети у групі: назва предмету;
- інформація про заняття у предметі групи: дата, номер заняття, тип
 заняття (лекційне заняття, практичне заняття, лабораторне заняття);

Інв. № подл.	Підпис і дата	Замксть.інв. №	Інв. № дубл.	1

Підпис і дата

Зм.. Лист № докум. Підпис Дата

0095.110577.ДП.2015.001

- інформація про завдання у предметі групи: повний опис завдання, термін здачі завдання, максимальна оцінка;
- інформація про студентів: повне ім'я, Email, пароль, номер телефону, адреса, номер паспорту, номер ІНН, кількість балів при вступі, запис видалено чи ні;
- інформація про групи студентів: назва групи, статус у цій групі (навчається, завершив навчання, виключено);
- інформація про предмети студента: назва предмета у групі;
- інформація про платежі студента: назва, напрям платежу, назва групи, дата проведення платежу, сума;
- інформація про відвідування студента: заняття, оцінка про відвідування (був присутній, був відсутній, запізнився, відпрацював пропуск заняття);
- інформація про оцінки студента: завдання, оцінка за виконання.

2.1.2 Вихідні дані

Вихідними даними програми ϵ наступні дані, які користувачі отримують за допомогою різних форм та таблиць:

- інформація про користувачів: ім'я, Етаіl, пароль, роль у системі, посада, запис видалено чи ні, він є викладачем чи ні;
- інформація про консультації викладачів: день тижня, час початку;
- інформація про університет: назва, скорочена назва, адрес, ім'я ректора;
- інформація про факультети: назва, скорочена назва, унікальний номер,
 ім'я декану, запис видалено чи ні;
- інформація про кафедри: назва, скорочена назва, назва факультету,
 ім'я завідуючого кафедрою, запис видалено чи ні;
- інформація про спеціальності: назва, назва кафедри, унікальний номер, запис видалено чи ні;
- інформація про ролі: назва та список прав;

.п∂уб № ви.	
Замксть.інв. №	
Підпис і дата	
Інв. № подл.	

Підпис і дата

- інформація про предмети: назва, загальна кількість годин, тип зарахування предмету (екзамен чи залік);
- інформація про платежі: назва, напрям платежу;
- інформація про групи: назва, рік початку навчання, поточний рік навчання, загальна кількість років навчання, назва спеціальності, ім'я куратора, навчання завершено чи ні, запис видалено чи ні;
- інформація про предмети у групі: назва предмету;
- інформація про заняття у предметі групи: дата, номер заняття, тип
 заняття (лекційне заняття, практичне заняття, лабораторне заняття);
- інформація про завдання у предметі групи: повний опис завдання, термін здачі завдання, максимальна оцінка;
- інформація про студентів: повне ім'я, Email, пароль, номер телефону, адреса, номер паспорту, номер ІНН, кількість балів при вступі, запис видалено чи ні;
- інформація про групи студентів: назва групи, статус у цій групі (навчається, завершив навчання, виключено);
- інформація про предмети студента: назва предмета у групі;
- інформація про платежі студента: назва, напрям платежу, назва групи, дата проведення платежу, сума;
- інформація про відвідування студента: заняття, оцінка про відвідування (був присутній, був відсутній, запізнився, відпрацював пропуск заняття);
- інформація про оцінки студента: завдання, оцінка за виконання.

2.1.3 Формалізація задачі

Формалізація задачі на рівні зовнішнього проектування представлена у вигляді діаграми варіантів використання програми (рис. 2.1-2.2).

Користувачі представлені у вигляді акторів «Студент» та «Викладач», що

Інв. № подп. Підпис і дата Замксть.інв. № Інв.№ дубл

Підпис і дата

Зм	Лист	№ докум.	Підпис	Дата	

взаємодіє з системою за допомогою варіантів використання. Варіанти використання надають опис можливостей, які система надає акторам.

На діаграмах використані наступні типи відношень між варіантами використання та акторами:

- відношення асоціації відображає зв'язок між акторами та варіантом використання. Відображається лінією зі стрілкою між акторами і варіантами використання;
- відношення включення між двома варіантами використання вказує,
 що деяка задана поведінка для одного варіанту використання включається як складова компонента в послідовність поведінки іншого варіанту використання. Позначається пунктирною стрілкою з поміткою «include».

Актор «Студент» може виконувати наступні дії (рис. 2.1):

- перегляд та редагування власних даних (контактних даних, а також зміна паролю);
- перегляд контактних даних викладачів та списку їх консультацій;
- перегляд інформації про структуру університету (університет, факультети, кафедри, спеціальності, групи);
- перегляд власної інформації щодо навчання (завдання та оцінки, розклад занять, та відвідування) та фінансів (список отриманих та відправлених платежів).

Актор «Викладач» може виконувати наступні дії (рис. 2.2):

- перегляд та редагування власних даних (контактних даних, а також зміна паролю);
- перегляд та редагування ролей;
- перегляд та редагування користувачів;
- перегляд та редагування консультацій викладачів;
- перегляд та редагування структури університету;
- перегляд та редагування даних групи;

Дата

Інв. № подп. Підпис і дата Замксть інв.

Лист

№ докум.

Підпис

Підпис і дата

Інв.№ дубл

S

- перегляд та редагування списку завдань;
- перегляд та редагування розкладу занять;
- перегляд та редагування даних про студента;
- перегляд та редагування даних про групи студента;
- перегляд та редагування даних про предмети у групі студента;
- перегляд та редагування оцінок студента;
- перегляд та редагування відвідувань студента.

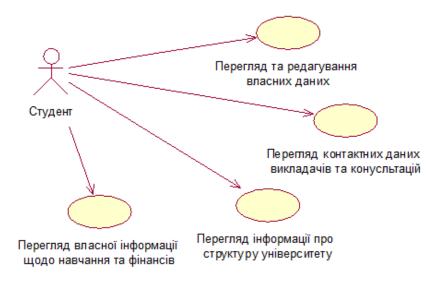


Рисунок 2.1 – Діаграма варіантів використання для актора «Студент»

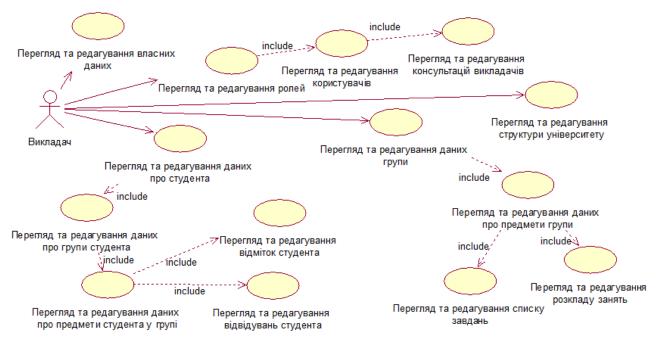


Рисунок 2.2 – Діаграма варіантів використання для актора «Викладач»

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв.№ дубл

윙

Замксть.інв.

Підпис і дата

№ подл

/HB.

0095.110577.ДП.2015.001

3 ВНУТРІШНЄ ПРОЕКТУВАННЯ

3.1 Вибір мови програмування та середовища розробки

Було прийнято рішення розробляти програму у вигляді веб-сайту, виходячи з багатьох причин:

- легко розробляти клієнт-серверну взаємодію та зручний інтерфейс користувача;
- великій вибір інструментів для розробки продукту;
- потрібно зручність користування. Користувачам не нічого Веб-сайтом завантажувати та нічого встановлювати. можна користуватися з будь-якого сучасного електронного пристрою (смартфон, планшет, ноутбук, ПК), який має доступ до мережі інтернет.

Інтерфейс користувача розроблятиметься за допомогою мови гіпертекстової розмітки HTML 5 та каскадної таблиці стилів CSS 3.

Для розробки серверної частини існує багато мов, наприклад:

- PHP:
- Python;
- Java;
- C#.

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Серед них був обраний РНР, тому що він ε найпопулярнішою мовою для розробки серверної частини веб-сайту, отже, для такого сайту буде не складно знайти дешевий хостинг, та такий сайт буде легко підтримувати та доповнювати.

У якості СУБД було вибрано MySQL, тому що її легко інтегрувати із вебсайтом, а також у PHP ϵ вбудовані функції для роботи з СУБД MySQL: підключення, вибір бази даних, відправка запитів та обробка отриманої інформації.

Інтерактивність клієнтській частині надається за рахунок JS та бібліотеки JQuery, а також технології AJAX для асинхронної взаємодії с сервером.

подп.					
Ne n					
IH8. I					
1	Зм	Лист	№ докум.	Підпис	Дата

0095.110577.ДП.2015.001

Існує досить багато середовищ розробки для веб-сайтів, наприклад:
JetBrains PHPStorm;
JetBrains WebStorm;
Sublime Text;

- Eclipse;

NetBeans;

Zend Studio.

JetBrains PHPStorm більше інших відповідала вимогам цього проекту, тому була вибрана саме вона у якості середовища розробки.

3.2 Вибір інструментів та парадигми проектування

Парадигма програмування - це сукупність ідей і понять, що визначають стиль написання комп'ютерних програм (підхід до програмування). Це спосіб концептуалізації, що визначає організацію обчислень і структурування роботи, що виконується комп'ютером.

Основними моделями програмування є:

- імперативне програмування;
- декларативне програмування;
- функціональне програмування;
- об'єктно-орієнтоване програмування.

У нашому проекті було вибрано об'єктно-орієнтоване програмування, так як воно дає нам багато переваг при розробці продукту.

Об'єктно-орієнтований підхід засновано на представлені програми у вигляді сукупності об'єктів, кожен з яких є екземпляром певного класу, а класи утворюють ієрархію зі спадкуванням властивостей.

Об'єктно-орієнтований підхід володіє наступними перевагами:

- опис системи у вигляді об'єктів більше відповідає змісту предметної області;
- сутності реального світу, як правило, володіють поведінкою, що в об'єктно-орієнтованому проектуванні відображаються за допомогою

Інв. № подл. Підпис і дата Замксть.інв. №

Підпис і дата

Інв.№ дубл

Зм.. Лист № докум. Підпис Дата

0095.110577.ДП.2015.001

- визначення методів класу. У структурному підході дані (атрибути) і алгоритми (методи) існують окремо один від одного;
- об'єднання атрибутів і методів в об'єкті (класі), а також інкапсуляція дозволяють домогтися більшої внутрішньої і меншою зовнішньої зв'язності між компонентами системи. Це полегшує вирішення проблем:
 - адаптації системи до зміни існуючих або появи нових вимог;
 - супроводу системи на різних стадіях життєвого циклу;
 - повторного використання компонентів;
- дозволяє легше організувати паралельні обчислення, так як кожен об'єкт володіє власними значеннями характеристик (атрибутів) і поведінкою, за рахунок чого можна домогтися його автономної роботи.

Також для реалізації програми був використаний шаблон MVC.

Ідея конструкційного шаблону MVC полягає у чіткому розділу відповідальності за різні функції у програмі:

- контролер керує запитами користувача (отримуються у вигляді запитів НТТР GET або POST, коли користувач натискає на елементи інтерфейсу для виконання різних дій). Його основна функція викликати і координувати дію необхідних ресурсів та об'єктів, потрібних для виконання дій, що задаються користувачем. Зазвичай контролер викликає відповідну модель для задачі і вибирає підходящий вид;
- модель це дані і правила, які використовуються для роботи з даними, які представляють концепцію управління додатком. У будьякому додатку вся структура моделюється як дані, які обробляються певним чином. Модель містить найбільш важливу частину логіки застосування, логіки, яка вирішує завдання, з яким ми маємо справу.

i dan				певн
Підпис і дап				заст
Інв. № подл.				
₹				
φ				
*	Зм	Лист	Nº	докум.

Підпис

Дата

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Контролер містить в основному організаційну логіку для самого додатка;

 вид забезпечує відображення даних, які отримані з моделі, користувачу. У нашому випадку дані передаються до браузеру клієнта та відображаються у ньому.

Для того, щоб реалізувати шаблон MVC було прийнято рішення використовувати шаблонізатор Smarty, так як він найбільш надійний та перевірений.

Smarty — це компілюючий обробник шаблонів для PHP, один з інструментів, що дозволяють відокремити прикладну логіку і дані від подання у виді концепції Model-view-controller.

Мова шаблонів Smarty розширює HTML smarty-тегами, вбудованими в документ. Ці теги можуть являти собою PHP-змінні (можливо, забезпечені модифікаторами, напр. {\$ Variable | nl2br}), функції або базові конструкції структурного програмування, такі як {if ...} ... {else} ... {/ if}.

Одне з призначень Smarty — це відділення логіки додатка від подання. Шаблони, тим не менш, можуть містити в собі логіку, але це має бути логіка представлення даних. Вона повинна вирішувати такі завдання, як підключення інших шаблонів, забарвлення рядків у таблиці що чергується, приведення букв до верхнього регістру, циклічний прохід по масиву для його відображення і т.п. Сама по собі бібліотека Smarty не примушує розділяти логіку програми та подання — коректна дисципліна використання веб-шаблонів залишається завданням розробника.

Smarty дозволяє:

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

- створювати користувача функції і модифікатори;
- використовувати настроюються роздільники тегів шаблону: {} і т.д.;
- можливість включення РНР-коду прямо в шаблон;
- користувацькі функції кешування;
- використання компонентної архітектури.

					ĺ
Зм	Лист	№ докум.	Підпис	Дата	

Для спрощення створення масштабованих веб-додатків в Smarty вбудований механізм кешування.

Для більш зручної роботи з cookie було підключено бібліотеку jquery-cookie.js.

Плагін jQuery cookie надає простий і зручний інтерфейс для роботи з соокіе на стороні клієнта. Зазвичай соокіе використовують на стороні сервера, проте в деяких випадках, більш доречно працювати з ними саме на стороні клієнта. Наприклад, якщо в javascript-плагіні необхідно організувати зберігання стану, то щоб уникнути додаткової серверної частини для плагіна, можна використовувати cookie на стороні клієнта.

Для більш зручної роботи з AJAX було підключено бібліотеку jquery.js.

Функція jQuery.ajax() завантажує віддалену сторінку використовуючи НТТР запит. Це реалізація АЈАХ в jQuery на низькому рівні. Для функцій більш високого рівня абстракції зверніться до \$.get, \$.post і т.д., які, найчастіше, легше зрозуміти і використовувати, але які все ж надають обмежену функціональність у порівнянні з даним методом.

\$.ajax() повертає об'єкт XMLHttpRequest. У більшості випадків Вам не знадобиться працювати безпосередньо з цим об'єктом, але він все ж доступний, у випадку, якщо Вам необхідно перервати запит вручну.

Як аргумент функції \$.ajax() передається об'єкт, що складається з пар ключ / значення, які використовуються для ініціалізації та управління запитом.

3.3 Об'єктно-орієнтоване проектування

На рис. 3.1 показана діаграма класів та зв'язків між ними. Як ми бачимо із діаграми усі класи є рівноправними та єдиний зв'язок, який є між ними, це агрегація. Агрегація — це проста асоціація між двома класами, яка відображає структурні відношення між рівноправними сутностями, коли обидва класи знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший.

Підпис і дата	,	СТ 3Н Ні
Iнв. № подл.		
⋛		
φ		
±	Зм	Л

Підпис і дата

Інв. № дубл.

Замксть.інв. №

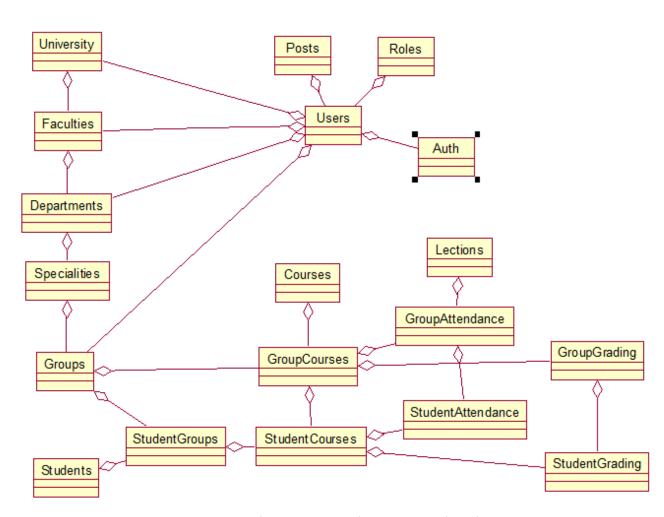


Рисунок 3.1 – Діаграма класів та зв'язків між ними

Діаграма класів дозволяє описувати статичну структуру класів і взаємозв'язки між ними, не відображаючи динамічну поведінку об'єктів зображених на ній класів. На діаграмах класів показуються класи, інтерфейси і відносини між ними.

Діаграми класів служать для наступних цілей:

- для моделювання даних. Аналіз предметної області дозволяє виявити основні характерні для неї сутності і зв'язки між ними. Це зручно моделюється за допомогою діаграм класів. Ці діаграми є основою для побудови концептуальної схеми бази даних;
- для представлення архітектури ПЗ. Можна виділити архітектурно значущі класи і показати їх на діаграмах, що описують архітектуру ПЗ;

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв.№ дубл.

₹

Замксть.інв.

Підпис і дата

Інв. № подл.

- для моделювання навігації екранів. На таких діаграмах показуються граничні класи та їх логічний взаємозв'язок. Інформаційні поля моделюються як атрибути класів, а керуючі кнопки – як операції і відносини;
- для моделювання логіки програми;
- для моделювання логіки обробки даних.

Далі представлено список усіх класів з їх описанням:

- клас University. Надає можливість переглядати та редагувати інформацію про університет;
- клас Faculties. Надає можливість переглядати, доповнювати,
 редагувати, видаляти та відновлювати інформацію про факультети;
- клас Departments. Надає можливість переглядати, доповнювати,
 редагувати, видаляти та відновлювати інформацію про кафедри;
- клас Specialties. Надає можливість переглядати, доповнювати редагувати, видаляти та відновлювати інформацію про спеціальності;
- клас Users. Надає можливість переглядати, доповнювати, редагувати, видаляти та відновлювати інформацію про користувачів, а також відновлювати пароль та перевіряти правильність вводу унікального Етаіl адресу;
- клас Roles. Надає можливість переглядати, доповнювати, редагувати та видаляти інформацію про ролі;
- клас Posts. Надає можливість переглядати, доповнювати,
 редагувати, та видаляти інформацію про посади;
- клас Auth. Надає можливість перевіряти правильність введення даних авторизації, авторизувати користувача у системі та вийти із неї, перевіряти наявність доступу у правах користувачів та перевести на сторінку «Немає доступу»;
- клас Groups. Надає можливість переглядати, доповнювати,
 редагувати, видаляти та відновлювати інформацію про групи;

I				
	Підпис і дата	Замксть.інв. №	Інв.№ дубл.	Підпис і дата

Ив. № подл

Зм.. Лист № докум. Підпис Дата

0095.110577.ДП.2015.001

- клас Courses. Надає можливість переглядати, доповнювати,
 редагувати та видаляти інформацію про предмети;
- клас GroupCourses. Надає можливість переглядати, доповнювати та видаляти інформацію про предмети групи;
- клас StudentCourses. Надає можливість переглядати, доповнювати та видаляти інформацію про студентів групи;
- клас Lections. Надає можливість переглядати, доповнювати,
 редагувати та видаляти інформацію про пари;
- клас GroupAttendance. Надає можливість переглядати, доповнювати,
 редагувати та видаляти інформацію про заняття у групі;
- клас GroupGrading. Надає можливість переглядати, доповнювати,
 редагувати та видаляти інформацію про завдання у групі;
- клас StudentAttendance. Надає можливість переглядати, доповнювати,
 редагувати та видаляти інформацію про відвідування студента;
- клас StudentGrading. Надає можливість переглядати, доповнювати,
 редагувати та видаляти інформацію про оцінки студента;
- клас Payments. Надає можливість переглядати, доповнювати,
 редагувати та видаляти інформацію про платежі;
- клас StudentPayments. Надає можливість переглядати, доповнювати,
 редагувати та видаляти інформацію про платежі студента.
- 3.4 Визначення призначень об'єктів за допомогою карток CRC

Для оцінки узгодженості об'єктів потрібно досліджувати кожен клас і його взаємодію з іншими класами. Щоб спростити таке дослідження зручно

Інв. № подл. Підпис і дата Замксть.інв. № Інв.№ дубл. Підпис і дата

Зм.. Лист № докум. Підпис Дата

0095.110577.ДП.2015.001

University		
Відповідальність	Зв'язки	
Робота з даними про університет	Використовується в Faculties	
	Використовує Users	

Таблиця 3.2 – CRC-картка на клас Faculties

Facı	ılties
Відповідальність	Зв'язки
Робота з даними про факультети	Використовується в Departments,
	Groups
	Використовує Users

Таблиця 3.3 – CRC-картка на клас Departments

Depar	Departments		
Відповідальність	Зв'язки		
Робота з даними про кафедри	Використовується в Specialities		
	Використовує Users, Faculties		

Таблиця 3.4 – CRC-картка на клас Specialities

Specialities	
Відповідальність Зв'язки	
Робота з даними про специальности	Використовується в Groups Використовує Departments

Інв. № подл. Підпис і дата

Лист

№ докум.

Підпис

Дата

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

0095.110577.ДП.2015.001

Users	
Відповідальність	Зв'язки
Робота з даними про користувачів	Використовується в Faculties,
	Departments, University, Groups, Auth
	Використовує Roles, Posts

Таблиця 3.6 – CRC-картка на клас Roles

Roles		
Відповідальність	Зв'язки	
Робота з даними про ролі	Використовується в Users	

Таблиця 3.7 – CRC-картка на клас Posts

Posts		
Відповідальність	Зв'язки	
Робота з даними про посади	Використовується в Users	

Таблиця 3.8 – CRC-картка на клас Auth

Auth		
Відповідальність	Зв'язки	
Авторизація та перевірка пра користувачів	В Використовує Users	

Таблиця 3.9 – CRC-картка на клас Groups

Groups	
Відповідальність	Зв'язки
Робота з даними про групи	Використовується в StudentGroups,
	GroupCourses
	Використовує Users, Specialities

Таблиця 3.10 – CRC-картка на клас Students

Students		
Відповідальність	Зв'язки	
Робота з даними про студентів	Використовується в StudentGroups,	
	StudentPayments	
	Використовує Users	

Таблиця 3.11 – CRC-картка на клас StudentGroups

StudentGroups		
Відповідальність Зв'язки		
Робота з даними про групи студента	Використовується в StudentCourses	
Використовує Students, Groups		

Підпис і дата		Роб
Інв. № подл.		
Š		
9		
=	Зм	Лист

№ докум.

Підпис

Підпис і дата

Інв. № дубл.

Замксть.інв. №

0095.110577.ДП.2015.001

Courses		
Відповідальність	Зв'язки	
Робота з даними про предмети	Використовується в GroupCourses	

Таблиця 3.13 – CRC-картка на клас GroupCourses

GroupCourses	
Відповідальність Зв'язки	
Робота з даними про предмети групи	Використовується в StudentCourses,
	GroupAttendance, GroupGrading
	Використовує Courses

Таблиця 3.14 – CRC-картка на клас StudentCourses

StudentCourses		
Відповідальність	Зв'язки	
Робота з даними про предмети	Використовується в	
студента	StudentAttendance, StudentGrading	
	Використовує StudentGroups,	
	GroupCourses	

Таблиця 3.15 – CRC-картка на клас Lections

Lections				
Відповідальність	Зв'язки			
Робота з даними про пари	Використовується в GroupAttendance			

Таблиця 3.16 – CRC-картка на клас GroupAttendance

GroupAttendance				
Відповідальність	Зв'язки			
Робота з даними про заняття у групі	Використовується в StudentAttendance			
	Використовує GroupCourses, Lections			

Таблиця 3.17 – CRC-картка на клас GroupGrading

GroupGrading					
Відповідальність	Зв'язки				
Робота з даними про завдання у групі	Використовується в StudentGrading				
	Використовує GroupCourses				

Таблиця 3.18 – CRC-картка на клас StudentAttendance

StudentAttendance						
Відповідальність	Зв'язки					
Робота з даними про відвідування	Використовує StudentCourses,					
студентів	GroupAttendance					

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв. № дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

0095.110577.ДП.2015.001

StudentGrading					
Відповідальність	Зв'язки				
Робота з даними про оцінки студентів	Використовує StudentCourses,				
	GroupGrading				

Таблиця 3.20 – CRC-картка на клас Payments

Payments				
Відповідальність	Зв'язки			
Робота з даними про платежі	Використовується в StudentPayments			

Таблиця 3.21 – CRC-картка на клас StudentPayments

StudentPayments						
Відповідальність					Зв'язки	
Робота з даними про платежі		платежі	Використовує Students, Payments			
студенті	В					

3.5 Проектування інтерфейсу користувача

Інтерфейс – це механізм, що забезпечує взаємодію користувачів з ПЗ (програмне забезпечення). Раціональна конструкція інтерфейсу користувача, що забезпечує простоту роботи з ним, має першорядне значення при створенні ПЗ.

Інтерфейс перш за все повинен враховувати потреби кінцевих користувачів ПЗ. Раціонально сконструйований інтерфейс полегшує освоєння і застосування програми, а невдалий викликає у клієнтів відчуття безсилля і знижує ефективність роботи з ПЗ.

Існує чотири основних критерії якості будь-якого інтерфейсу, а саме:

- швидкість роботи користувачів;
- кількість людських помилок;
- швидкість навчання;

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

- суб'єктивне задоволення користувачів.

Швидкість виконання роботи ϵ важливим критерієм ефективності інтерфейсу. У чистому вигляді цей критерій цінують досить рідко, але майже завжди він ϵ вкрай бажаною складовою цілого. Будь-яка спроба якось збільшити продуктивність праці завжди зустрічається позитивно.

Тривалість виконання роботи користувачем складається з наступних

						Арк
					0095.110577.ДП.2015.001	<u> </u>
Зм	Лист	№ докум.	Підпис	Дата		5:

- тривалості сприйняття вихідної інформації;
- тривалості інтелектуальної роботи (користувач думає, що він повинен зробити);
- тривалості фізичних дій користувача;
- тривалості реакції системи.

Тривалість сприйняття вихідної інформації особливих коментарів не потребує, користувач повинен уявити собі, яка інформація про виконуване завдання у нього існує, і в якому стані знаходяться засоби, за допомогою яких він буде вирішувати це завдання. Основний час тут витрачатиметься на зчитування показань системи.

Тривалість інтелектуальної роботи оцінюється тим, що взаємодія користувача з системою (не тільки комп'ютерною) складається з семи кроків:

- формування мети дій;
- визначення загальної спрямованості дій;
- визначення конкретних дій;
- виконання дій;
- сприйняття нового стану системи;
- інтерпретація стану системи;
- оцінка результату.

З даного списку можна побачити, що процес роздумів займає майже весь час, протягом якого користувач працює з комп'ютером, у всякому разі, шість з семи етапів повністю зайняті розумовою діяльністю. Відповідно, підвищення швидкості цих роздумів приводить до істотного поліпшення швидкості роботи.

На жаль, істотно підвищити швидкість власне мислення користувачів неможливо. Тим не менш, зменшити вплив чинників, які ускладнюють і, відповідно, уповільнюють процес мислення, цілком можливо. Наприклад, застосуванням методу безпосереднього маніпулювання і застосування в інтерфейсі ефективних методів при втраті фокусу уваги.

Інв. № дубл.	
Замксть.інв. №	
Підпис і дата	
Інв. № подл.	

Підпис і дата

Людські помилки

Вчинення помилок є природним для людини. А раз помилки природні, значить система має бути здатна сама їх виявити і виправити. Таким чином, людських помилок не буває. Бувають помилки в проектуванні систем. Під словосполученням «людська помилка» треба розуміти «дія користувача, не збігалася з метою дій цього користувача».

Типи помилок:

- помилки, викликані недостатнім знанням предметної області;
- помилки друку;
- не зчитування показань системи;
- моторні помилки.

При боротьбі з помилками потрібно спрямовувати зусилля на:

- плавне навчання користувачів в процесі роботи;
- зниження вимог до надмірної уваги;
- підвищення розбірливості і помітності індикаторів;
- зниження чутливості системи до помилок.

Для цього ϵ три основних способи, а саме:

- блокування потенційно небезпечних дій користувача до отримання підтвердження правильності дії;
- перевірка системою всіх дій користувача перед їх прийняттям;
- самостійний вибір системою необхідних команд або параметрів, при яких від користувача потрібно тільки перевірка.

Засоби навчання користувачів роботі з системою:

- паперова документація;
- «Оперативна довідка»;
- загальна «зрозумілість» системи;
- навчальні матеріали.

Iнв. № подл. Підпис і дата Замкст	N <u>е</u> подл. Підпис і д

Підпис і дата

Інв.№ дубл

b.iH8. №

Навчальні матеріали:

- базова довідка пояснює користувачеві сутність та призначення системи. Зазвичай повинна спрацювати лише один раз, пояснюючи користувачеві, навіщо система потрібна;
- оглядова довідка рекламує користувачеві функції системи. Також зазвичай спрацьовує один раз. Потрібна тим більше, ніж більш функціональна система;
- довідка предметної області відповідає на питання «Як зробити добре?»;
- процедурна довідка відповідає на питання «Як це зробити?». Кращим місцем для процедурної довідки є виділена довідкова система;
- контекстна довідка відповідає на питання «Що це робить?» і «Навіщо це потрібно?». Для контекстної довідки заслужено використовують спливаючі підказки (ToolTip) і, останнім часом, бульбашки;
- довідка стану відповідає на питання «Що відбувається зараз?».

Суб'єктивне задоволення:

- увага до деталей;
- інтерфейс передає інформацію своєму користувачеві;
- інтерфейс зазвичай призначений для тривалого використання;
- інтерфейс функціональний.

Розробка інтерфейсу користувача.

При вході до програми користувач буде бачити сторінку авторизації (рис. 3.2). Доки правильний Email та пароль не буде введено, користувач не зможе відкрити жодну іншу сторінку.

У студентів та викладачів одна сторінка авторизації.

Інв. № подл. Підпис і дата Замксть.інв. № Інв. № дубл

Підпис і дата

Зм.. Лист № докум. Підпис Дата

0095.110577.ДП.2015.001



Органайзер Учебного Процесса

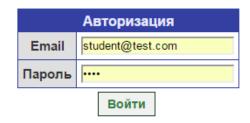


Рисунок 3.2 – Сторінка авторизації

На кожній сторінці програми ϵ навігаційне меню (рис. 3.3), яке допоможе користувачу пересуватись по сайту.

Мой профиль	Студенты	Группы	Университет	Преподаватели	Админ	
					Пользователи	
					Роли	
					Должности	
					Платежи	
					Занятия	
					Предметы	

Рисунок 3.3 – Навігаційне меню

У меню будуть такі пункти:

- «Профіль користувача» перегляд та налаштування персональної інформації;
- «Студенти» перегляд та налаштування студентів, їх груп, платежів,

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Iнв. № подл.

0095.110577.ДП.2015.001

Аркуш

59

- «Групи» перегляд та налаштування груп, предметів, занять та завдань;
- «Університет» перегляд та налаштування структури університету;
- «Викладачі» перегляд та налаштування викладачів та їх консультацій;
- «Користувачі» перегляд та налаштування користувачів;
- «Ролі» перегляд та налаштування ролей;
- «Посади» перегляд та налаштування посад;
- «Платежі» перегляд та налаштування платежів;
- «Пари» перегляд та налаштування пар;

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

«Предмети» – перегляд та налаштування предметів.

Уся інформація виводиться у вигляді форм та таблиць (рис. 3.4-3.5).

	Пользователи						
ld	Полное Имя	Email	Роль	Должность	Активен	Преподаватель	Действия
2	Администратор	admin2@test.com	Все права	Администратор	②	②	Ø 🔑 🖨
3	Пшинько А.М.	rector@test.com	Все права	Администратор	②	②	0 P 🖨
4	Скалозуб В.В.	dean@test.com	Все права	Декан	②	②	0 P 🖨
5	<u>Шинкаренко В.И.</u>	head@test.com	Образование главный	Заведующий кафедрой	②	②	0 P 🖨
6	Андрющенко В.А.	instructor@test.com	Образование	Преподаватель	②	②	Ø 🔑 🖨
9	Швец О.М.	curator@test.com	Образование	Преподаватель	②	②	0 P 🖨
7	Тищенко Л.М.	accounting@test.com	Финансы	Бухгалтер	②	②	0 P 🖨
8	Павленко А.И	enrollment@test.com	Регистрация	Приемная комиссия	②	②	0 P 🖨
Ст	Страницы: 1						
Д	Добавить						

Рисунок 3.4 – Приклад таблиці

Профиль Пользователя				
ld	2			
Полное Имя	Администратор			
Email	admin2@test.com			
Роль	Все права			
Должность	Администратор			
Активен	②			
Преподаватель	&			
Все Пользователи Редактировать Удалить				

						Рисунок 3.5 – Приклад форми	
					0005 110577 HH 2015 001	Аркуш	
	Зм	Лист	№ докум.	Підпис	Дата	0095.110577.ДП.2015.001	

У системі буде впроваджено декілька іконок з підписами для позначення певних дій:

- помилка у введених даних ⁽¹⁾;
- редагування –
- видалення ⊚;
- відміна, ні [™];
- додавання [™];
- підтвердження, так ∅;
- перегляд [●];
- вхід до системи ≤
- відновлення паролю №.

У системі є три типа кнопок:

- звичайна кнопка. Використовується якщо при натисненні на кнопку не відбувається ніяких спеціальних дій. Наприклад, просто відкривається нова сторінка. (рис. 3.6);
- кнопка-підтвердження. Використовується коли треба підтвердити якусь дію, або якщо дія не приносить шкоди. Наприклад, вхід до системи, додавання або зберігання. (рис. 3.7);
- кнопка-відмова. Використовується коли треба відмінити якусь дію, або якщо дія приносить шкоду. Наприклад, вихід із системи, видалення (рис. 3.8);

Все Пользователи

Рисунок 3.6 – Приклад звичайної кнопки

|нв. № подл. Підпис і дата Замксть.інв. № | Інв.№ дубл. Підпис і дата

Рисунок 3.7 – Приклад кнопки-підтвердження

Удалить

Рисунок 3.8 – Приклад кнопки-відміни

Графічний інтерфейс дуже простий:

- усі кути прямі, без закруглень;
- немає тіней;
- немає анімацій;
- немає дуже яскравих елементів;
- тощо.

Іншими словами все спроектовано так, щоб користувач зміг просто та швидку знайти необхідну йому інформацію, щоб його нічого не відволікало та не заважало працювати з системою.

Вибрано нейтральну кольорову гаму: синій, білий, сірий.

Фон буде майже всюди білий, а текст чорний. Це ϵ найкращою комбінацією для підвищення читабельності тексту. Меню та заголовки форм та таблиць синього кольору. Строки та стовпці, що ϵ заголовками мають сірий фон.

Зеленим та червоним кольором завдяки контрасту звертається увага на важливі моменти.

Повідомлення у системі мають бути повними та зрозумілими. Є декілька видів повідомлень:

- повідомлення про помилку. Використовується, коли користувач ввів не правильні дані. (рис. 3.9);
- інформаційне повідомлення. Використовується, коли потрібно про щось проінформувати користувача. У такому повідомлення звичайне є лише кнопка «ОК», тому що користувач повинен підтвердити, що він прочитав повідомлення (рис. 3.10);

|--|

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

з

повідомлення-питання. Використовується, користувачу коли необхідно щось вибрати або підтвердити якусь дію (рис. 3.11).

Полное Имя должно быть введено. Полное Имя должно содержать только буквы кирилицы. Етмаі должен быть уникальным.				
Редактиро	Редактирование Пользователя			
Полное Имя				
Роль	Все права 🔻			
Должность	Администратор ▼			
Email	dean@test.com			
Преподаватель				
Сохранить	ИТЬ			

Рисунок 3.9 – Приклад повідомлення про помилку

Ошибка! Доступ запрещен!

При необходимости доступа, пожалуйста, обратитесь к администратору.

Вернуться назад

Вернуться в профиль

Рисунок 3.12 – Сторінка викладача (користувач без права)

Консульт	Консультации			
День	Время			
Понедельник	12:20:00			
Четверг	11:00:00			

Рисунок 3.13 – Сторінка викладача (користувач з правом тільки перегляду)

Консультации				
День	Время	Действия		
Понедельник	12:20:00	Ø 🖨		
Четверг	11:00:00	<i>Q</i> 🖨		
Добавить				

Рисунок 3.14 – Сторінка викладача (користувач з правом перегляду і редагування)

Підпис і дата	
Інв. № дубл.	
Замксть.інв. №	
Підпис і дата	
Інв. № подл.	

Зм	Лист	№ докум.	Підпис	Дата

0095.110577.ДП.2015.001

4 ТЕСТУВАННЯ ТА НАЛАГОДЖЕННЯ ПРОГРАМИ

Тестування програми – це процес виконання програми з метою виявлення помилок. Тестом називається набор вхідних і вихідних даних, які відповідають специфікації.

До найбільш ефективних методів тестування можна віднести комбінацію стратегій чорної шухляди і білої шухляди. Стратегія «чорної шухляди» передбачає вивчення і тестування зовнішньої поведінки функціонуючої програми; стратегія «білої шухляди» передбачає вивчення і тестування внутрішньої архітектури і логіки програми на основі її вихідних кодів.

4.1 Тестування методом «білої шухляди»

Найбільш ефективними методами тестування «білої шухляди» ϵ метод покриття умов і метод покриття рішень. Метод покриття умов поляга ϵ в такому підборі тестів, коли кожна умова (елементарне судження в умовних операторах) прийма ϵ як істинне так і хибне значення.

Метод покриття рішень вимагає такої кількості тестів, щоб при виконанні їх усіх по кожній траєкторії, що з'єднує сусідні елементи блок-схеми обчислення пройшло хоча б один раз. Це означає, що кожне рішення повинно приймати як істинні, так і помилкові значення. Саме це забезпечує використання всіх шляхів, що виходять з точок розгалуження.

Вхідними даними для тестування програми методами білого ящика ϵ специфікації функції та текст програми. Таким чином далі буде надаватися:

- текст методу, який тестуємо;
- специфікація;
- тестування методом покриття умов;
- тестування методом покриття рішень.

Метод: canAuth().

Специфікація: метод перевіряє чи можна авторизувати користувача (студента або викладача).

Іне. № подл. Підпис і дата Замксть.інв. № Інв. № дубл

Підпис і дата

Зм	Лист	№ докум.	Підпис	Дата	

0095.110577.ДП.2015.001

```
рядок: email (email-адреса користувача);
         рядок: password (пароль користувача).
     Вихідні дані: цифра (1 – можна зайти як викладач, 2 – можна зайти як
студент, 3 – не можна зайти)
     Текст методу:
public function testAuth ($email, $password)
{
    $this->assertEquals(1,Auth::canAuth($email, $password));
    $this->assertEquals(2,Auth::canAuth(($email, $password));
    $this->assertEquals(0,Auth::canAuth(($email, $password));
}
      Тест 1.
     Bxiд: $email = "admin@test.com", $password = "1234"
     Вихід: true,false,false.
     Тест 2.
     Bxiд: $email = "student@test.com", $password = "1234"
     Вихід: false, true,false.
     Тест 3.
     Bxiд: $email = "admin@test.com", $password = "0000"
     Вихід: false,false,true.
     Таблиця 4.1 Матриця обліку умов методу testAuth ()
     Тест
                                             Умови
                 mysql_num_rows($users) >= 1
                                                  mysql_num_rows($studs) >= 1
                       +
                                                        +
      1
                       +
                                                                        +
      2
      3
                                       +
                                                                        +
```

0095.110577.ДП.2015.001

Аркуш

66

Вхідні дані:

Підпис і дата

Інв.№ дубл

Замксть.інв. №

Підпис і дата

Інв. № подл.

Лист

№ докум.

Підпис

Дата

4.2 Тестування методом «чорної шухляди»

Метою тестування ставиться з'ясування обставин, в яких поведінка програми не відповідає специфікації. Для тестування попередніх методів використаємо наступні методи стратегії: метод припущення о помилці і метод еквівалентного розбиття. Метод припущення о помилці полягає в інтуїтивному передбаченні ймовірних помилок і наступній розробці тестів для їх виявлення.

Метод еквівалентного розбиття полягає у розбитті всіх тестів на класи еквівалентності. Якщо один тест виявляє помилку, то і інший виявить, та навпаки.

Специфікація: редагування власних даних користувачем.

Тест 1. Користувач ввів ім'я латинськими літерами.

Вхід: Повне Ім'я = «Ivanov I.I.»

Очікується: інформація про користувача у БД зміниться.

Вихід: повідомлення про помилку «Повне Ім'я повинно містити лише букви кирилиці».

Тест 2. Користувач не ввів ім'я.

Bхід: Повне Ім'я = «»

Очікується: інформація про користувача у БД зміниться.

Вихід: повідомлення про помилку «Повне Ім'я повинно бути введено».

Тест 3. Користувач ввів Етаіl, який вже використовує інший студент або викладач.

Bxiд: Email = «admin@test.com»

Очікується: інформація про користувача у БД зміниться.

Вихід: повідомлення про помилку «Етаіl має бути унікальним».

Тест пройдено успішно.

4.3 Налагодження програми

Налагодження – це процес виявлення помилок програми та їх усунення. Налагодження спрямоване на встановлення точного місця відомої помилки, а

Підпис і дата

Інв.№ дубл.

Зм	Лист	№ докум.	Підпис	Дата

потім на виправлення цієї помилки. Тестування та налагодження дуже пов'язані – результати тестування ϵ вихідними даними для налагодження.

Налагодження програми буде проводитись з використанням методу індукції та методу просування від місця виникнення помилки до місця помилки. Таким чином, отримаємо налагоджену програму.

Налагодження виконується за допомогою спеціальних утиліт, які вбудовані у браузер Google Chrome:

- watch дозволяє дивитися зміст деяких змінних під час прогонки програми;
- stack дозволяє дивитися зміст стеку у деякий момент часу;
- step дозволяє виконувати програму крок за кроком.

Треба зазначити, що, оскільки до браузера, надходять вже скомпільовані тексти програми, щоб налагодити програми нам необхідно додати до скомпільованого тексту так звані source maps. Це спеціальним образом закодовані дані які свідчать браузеру про те, що він виконує скомпільований файл який має вихідні коди.

До місця виникнення ймовірної помилки ставиться точка зупину Breakpoint (рис. 4.1), на якій програма зупиниться і чекатиме покрокового виконання. На кожному наступному кроці існує можливість поглядання змісту змінних у watch.

Таким чином легко відстежити, в якому місці програми виникає помилка.

Iнв. № подп. Підпис і дата Замксть.інв. № Інв.№ дубл.

Підпис і дата

Зм	Лист	№ докум.	Підпис	Дата

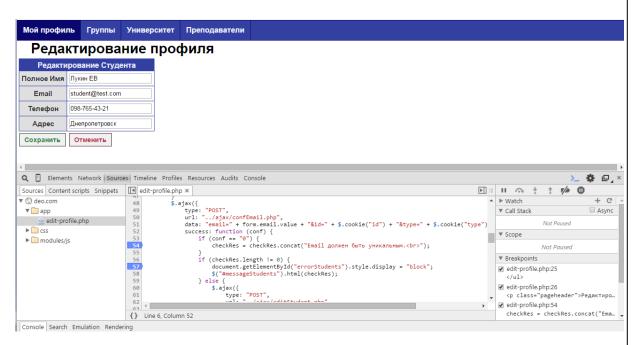


Рисунок 4.1 – Налагодження програми

Підпис і дата	
Інв. № дубл.	
Замксть.інв. №	
Підпис і дата	
Інв. № подл.	 <i>Аркуш</i> 69

Програма-органайзер учбового процесу університету є програмним продуктом, який буде робити учбовий процес студентів та співробітників університету більш зручним та швидким. Студенти зможуть мати віддалений доступ до інформації про свої предмети, завдання, строки, оцінки, відвідування, розклад занять, розклад консультацій викладачів, власні фінансові операції, контактні дані викладача, тощо. А співробітники зможуть більш швидко та зручно зберігати та опрацьовувати навчальні завдання, журнали відвідувань та оцінок, фінансові операції, тощо. Під час роботи з даною системою, користувач буде працювати з персональним комп'ютером, клавіатурою, монітором, або ноутбуком, TOMY слід зазначити правила користування електроннообчислювальними машинами та відео-дисплейними терміналами.

5.1 Аналіз шкідливих та небезпечних виробничих факторів

Небезпечні виробничі чинники — це виробничі чинники, вплив яких на працівника в певних умовах призводить до травм, гострого отруєння або іншого раптового різкого погіршення здоров'я або до смерті [1].

До різкого погіршення здоров'я можна віднести отруєння, опромінення, удар електрострумом, тепловий удар та ін.

До небезпечних факторів відносяться електронебезпека та пожежонебезпека.

Шкідливі виробничі фактори — фактори, тривалий вплив яких на працюючого у визначених умовах приведе до захворювання, зниження працездатності і (або) негативного впливу на здоров'я нащадків [1]. У залежності від рівня і тривалості впливу шкідливі фактори можуть класифікуватися і як небезпечні.

Згідно санітарно-гігієнічним вимогам [2] умови праці робітника, який працює з програмою-органайзером учбового процесу на ЕОМ, повинні відповідати І або ІІ класу.

Інв. № подл.	Підпис і дата	39

Підпис і дата

Інв.№ дубл.

<u>имксть.інв. №</u>

Зм	Лист	№ докум.	Підпис	Дата

Робота на ЕОМ пов'язана з наступними шкідливими факторами:

- недолік освітлення природним світлом;
- відблиски на екрані монітора;
- іонізуюче випромінювання електронно-променевого монітору;
- електромагнітне поле.

Недостатнє освітлення приводить до швидкої стомленості очей, що у свою чергу, призводить до зниження продуктивності праці і росту кількості прийнятих помилкових рішень.

Відблиски на екрані електронно-променевого монітора, що виникають при неправильному освітленні, приводить до погіршення зору, а у випадку тривалого впливу даного небезпечного фактора, може призвести до повної втрати зору. З метою зниження рівня впливу на робітника даного шкідливого фактора, варто дотримуватись вимог [3] або застосовувати рідкокристалічні монітори, які в силу своєї конструкції і використовуваних матеріалів мають менший коефіцієнт відбиття світла, ніж електронно-променевий монітор, тому відблисків на них практично не буває.

Рівні електромагнітного випромінювання і магнітних полів на робочому місці повинні відповідати вимогам [4].

5.1.1 Небезпечні чинники пов'язані з пожежами

Пожежі становлять особливу небезпеку для життя людини, і можуть призвести до великих матеріальних втрат. Джерелами загоряння можуть виявитися електронні схеми ЕОМ, елементи, що перегрілися, які здатні викликати загоряння пальних матеріалів.

Для всіх споруджень та приміщень, у яких експлуатуються система управління процесами програмної інженерії на ЕОМ, повинна бути визначена категорія по вибухонебезпечній і пожежній безпеці відповідно до [5], і клас зони згідно ППЕ [6]. Відповідні позначення повинні бути нанесені на вхідні двері приміщення.

Будинки та ті їх частини, у яких розміщуються ЕОМ, повинні мати

 В
 3м.. Лист
 № докум.
 Підпис
 Дата

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

0095.110577.ДП.2015.001

ступінь вогнестійкості не нижче II. Приміщення для обслуговування, ремонту і налагодження ЕОМ повинні відноситися:

- по пожежонебезпеки до категорії В пожежонебезпечні приміщення,
 де розташовуються тверді пальні речовини (ТПР);
- по класу приміщення до категорії П II [6].

Неприпустимим є розташування приміщень категорії А и Б [5], а також виробництв із мокрими технологічними процесами поруч із приміщеннями, де розташовуються ЕОМ, виконується їхнє обслуговування, налагодження і ремонт, а також над такими приміщеннями під ними.

Приміщення з ЕОМ повинні бути оснащені:

- системою автоматичної пожежної сигналізації відповідно до вимог переліку однотипних за значенням об'єктів, що підлягають устаткуванню автоматичними установками пожежогасіння і пожежної сигналізації;
- димовими пожежними оповіщувачами;
- переносними вуглекислотними вогнегасниками з розрахунку 2 шт, на кожні 20 м² площі приміщення з обліком гранично припустимих концентрацій вогнегасної рідини відповідно до вимог Правил пожежної безпеки в Україні.

Стіни приміщень з ЕОМ виготовляються з не пальних матеріалів. Підходи до засобів пожежогасіння повинні бути вільними.

5.1.2 Електронебезпечні чинники

При роботі з системою управління процесами програмної інженерії на ЕОМ можуть трапитися нещасні випадки, зв'язані з ураженням електричним струмом, викликаним дотиком до оголених місць струмоведучих частин устаткування, або частин, що знаходяться під напругою.

Вплив струму на людину проявляється по-різному, в залежності від його величини:

- струм до 0,6 мА не відчувається людиною;

	T			
	+			
	+			
Зм.	. Jlucm	№ докум.	Підпис	Дата

Підпис і дата

Інв. № дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

0095.110577.ДП.2015.001

Інв. № подл.

- струм завбільшки 6 мА призводить до скорочення м'язів тієї частини,
 тіла, що піддалася його впливу;
- струм 50 мА вражає органи дихання та серцево-судинну систему;
- при досягненні струму порога 100мА настає фібриляція серця і, потім, його зупинка;
- при впливі на тіло людини струму в 3-4 А виникає обвуглювання ділянок тіла.

За способом захисту людини від ураження електричним струмом відео термінали, ЕОМ, периферійні пристосування ЕОМ і оснащення для обслуговування, ремонту і налагодження ЕОМ повинні відповідати І класу захисту. Вимоги електричної і механічної безпеки і методи іспитів чи повинні бути заземлені.

Лінія електромережі для живлення ЕОМ, периферійних устроїв ЕОМ і устаткування для обслуговування, ремонту і налагодження ЕОМ виконується як окрема групова, трьох провідна мережа, шляхом прокладки фазового, нульового робочого і нульового захисного провідників.

Підключення на розподільному щиті до одного контактного затиску нульового робочого і нульового захисного провідників заборонено.

Площа перетину нульового робочого і нульового захисного провідника в груповій трьох провідній мережі повинна бути не менш площі перетину фазового провідника.

У приміщенні, де одночасно експлуатується чи обслуговується більш п'яти персональних ЕОМ, на видному доступному місці встановлюється аварійний вимикач, за допомогою якого можливо зробити знеструмлення приміщення (за винятком освітлення).

5.1.3 Фізично шкідливі і небезпечні чинники

До фізичних шкідливих і небезпечних чинників відносяться:

підвищені рівні електромагнітного, рентгенівського,
 ультрафіолетового і інфрачервоного випромінювання;

Зм	Лист	№ докум.	Підпис	Дата	

- підвищений вміст позитивних аеронів і понижений вміст негативних аеронів в повітрі робочої зони.
- 5.1.4 Хімічно шкідливі і небезпечні чинники

Хімічні шкідливі і небезпечні чинники наступні:

- підвищений вміст в повітрі робочої зони двоокису вуглецю, озону;
- бідність повітря робочої зони на кисень.
- 5.2 Організаційні і технічні заходи щодо захисту працівників від шкідливих і небезпечних факторів.

Для якісної і зручної роботи з системою управління процесами програмної інженерії на ЕОМ, необхідне проведення проектних заходів: правильна облаштованість, належне дотримання ергономічних характеристик основних елементів робочого місця [7], санітарно-гігієнічних вимог [8,9] і т.п.

Облаштованість робочого місця користувача при роботі з підсистемою повинне забезпечувати відповідність всіх елементів робочого місця і їхнього розташування.

Площа, яка виділена для одного робочого місця з ВДТ чи ЕОМ, повинна складати не менш 6 m^2 , а обсяг – не менш 20 m^3 .

Робочі місця з ВДТ щодо світлових прорізів повинні розташовуватися так, щоб природне світло падало з боку, переважно ліворуч.

При розташуванні робочих місць із ВДТ необхідно дотримувати наступних вимог:

- робочі місця з ВДТ розташовуються на відстані не менш 1 м від стін зі світловими прорізами;
- відстань між бічними поверхнями ВДТ повинне бути не менш 1,2 м;
- відстань між тильною поверхнею ВДТ і екраном іншого ВДТ повинне бути не менш 2,5 м;
- прохід між рядами робочих місць повинний бути не менш 1 м.

Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв. № дубл.

Замксть.інв. №

Підпис і дата

Інв. № подл.

0095.110577.ДП.2015.001

Висота робочої поверхні столу для ВДТ повинна бути в межах 680 - 800 мм, а ширина - забезпечувати виконання належних операцій.

Розміри столу, що рекомендуються: висота — 725 мм, ширина ~ 600 - 1400 мм, глибина — 800-1000 мм.

Робочий стіл для ВДТ повинний мати простір для ніг висотою не менш 600 мм, шириною не менш 500 мм, глибиною на рівні колін не менш 450 мм, на рівні витягнутої ноги \sim не менш 650 мм,

Можливість обертання екрана ВДТ навколо горизонтальної і вертикальної осі.

Вимоги до організації робочого місця

При роботі з системою управління процесами програмної інженерії на ЕОМ необхідно виконувати вимоги до організації робочого місця, що наведені нижче.

До самостійної роботи допускаються особи, які пройшли медичний огляд, навчання за професією, вступний інструктаж з охорони праці на робочому місці.

Конструкція робочого місця робітника при роботі з ЕОМ [10] (при роботі сидячи) повинна забезпечувати підтримку оптимальної робочої пози з наступними ергономічними характеристиками:

- стопи ніг на підлозі чи на підставці для ніг;
- стегно у горизонтальній площині;
- передпліччя вертикально;
- лікті під кутом 70°-90° до вертикальної площини;
- зап'ястя зігнуті під кутом не більш 20° щодо горизонтальної площини;
- нахил голови 15°-20° відносно вертикальної площини.

Дисплей та клавіатура мають розташовуватися на оптимальній відстані від очей користувача, але не ближче 600 мм, з урахуванням розміру алфавітноцифрових знаків та символів.

Замксть.інв. №	
Підпис і дата	
Інв. № подл.	

Підпис і дата

Iнв.№ дубл.

Розташування екрану відео терміналу має забезпечувати зручність зорового спостереження у вертикальній під кутом ±30° від лінії зору працівника.

Клавіатуру слід розміщувати на поверхні столу або на спеціальній, робочій і такій, яка регулюється за висотою поверхні окремо від столу на відстані 100-300 мм від краю, ближчого до працівника. Кут нахилу клавіатури має бути в межах 5-15°.

5.3 Норми пожежної безпеки

Вимоги безпеки праці перед початком роботи на ЕОМ

Перед початком роботи з системою управління процесами програмної інженерії на **EOM** робітник інструктується первинним інструктажем. Первинний інструктаж завжди проводиться на робочому місці з безпосереднім показом робіт (стажування 1 місяць). Потім, через кожні 6 місяців проводиться повторний інструктаж. Результати інструктажу заносяться в "Журнал реєстрації інструктажів з питань охорони праці". У журналі після проходження інструктажу повинен бути підпис людини, яка інструктувала та робітника. Далі підготовка робочого місця повинна виконується відповідно нижче приведених пунктів:

- увімкнути систему кондиціонування повітря в приміщенні;
- оглянути робоче місце і привести його в порядок: переконатися, що на ньому немає сторонніх предметів; все устаткування і блоки ЕОМ з'єднані із системним блоком за допомогою сполучних шнурів;
- перевірити надійність установки апаратури на робочому столі. Монітор повинен стояти не на краю столу. Повернути монітор так, щоб було зручно дивитися на екран – під прямим кутом (а не збоку) – та ледь зверху вниз, при цьому екран повинен бути ледь нахилений – нижній його край ближче до робітника;
- перевірити загальний стан апаратури, або перевірити чи справні електропроводка, сполучні шнури, штепсельні вилки, розетки,

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

перевірити заземлення захисного екрана;

- відрегулювати освітлення робочого місця;
- відрегулювати і зафіксувати висоту крісла, зручний для робітника нахил його спинки;
- включити апаратуру комп'ютера перемикачами на корпусі в послідовності: стабілізатор напруги, монітор, системний блок;
- відрегулювати яскравість світіння екрана до яскравості навколишніх його поверхонь у робочій зоні – і не більше 3:1;
- при виявленні будь-яких несправностей роботу не починати, повідомити про це керівника робіт.

Вимоги безпеки праці під час роботи на ЕОМ

Під час роботи з системою управління процесами програмної інженерії на ЕОМ необхідно дотримуватися наступних правил:

- при виконанні роботи на комп'ютері слід сидіти прямо, не напружуватися;
- не дозволяються сторонні розмови, подразнюючі шуми;
- постійно слідкувати за функціонуванням систем опалення і кондиціювання;
- для зниження напруженості праці на комп'ютері необхідно рівномірно розподілити і чергувати характер робіт відповідно до їх складності.

Для зменшення негативного впливу на здоров'я працюючих виробничих факторів необхідно застосовувати регламентні перерви. У таблиці 5.1 приведений час регламентованих перерв робітника у залежності від категорії і групи робіт.

Інв. № подл.	Підпис і дата	Замксть.інв. №	Інв.№ дубл.	Підпис і дата

Зм.. Лист № докум. Підпис Дата

0095.110577.ДП.2015.001

Категорія	Група роботи			Час перерви при
робіт	А, кількість	Б, кількість	В, година	8-годинній зміні,
	знаків	знаків		XB.
Оптимальна-І	до 20 000	до 15000	до 2	20
Припустима-	21 000-40 000	до 30 000	до 4	40
II				
Важка-III	понад 40 000 не більш 60 000	понад 30 000 не більш 45 000	понад 4 не більш 6	60

Група A – читання інформації з попередні запитом (діалоговий режим роботи).

Група Б – введення інформації.

Група В – творча робота в режимі діалогу ЕОМ (налагодження програм, переклад і редагування текстів і т, д.).

При восьмигодинній робочій зміні регламентована перерва повинна бути такою:

- для I категорії робіт через 2 години від початку зміни і через 2 години після обідньої перерви (кожен тривалістю 10 хв.);
- для II категорії робіт через 2 години від початку зміни (тривалістю 15 хв.), через 1,5 і 2,5 годин після обідньої перерви (15 і 10 хв. відповідно чи тривалістю 5-10 хв. Через щогодини роботи, в залежності від характеру технологічного процесу);
- для III категорії робіт через 2 години від початку зміни (тривалістю 15 хв.), через 1,5 і 2,5 годин після обідньої перерви (тривалістю 20 хв. Кожний тривалістю 5-15 хв. Щогодини роботи, у залежності від характеру технологічного процесу);
- під час нічної зміни, незалежно від групи і категорії робіт, тривалість

№ подл.						
Ne⊓						
/H8.						
-	Зм	Лист	№ докум.	Підпис	Дата	

Підпис і дата

Інв. № дубл.

Замксть.інв. №

Підпис і дата

0095.110577.ДП.2015.001

регламентованих переривів збільшується на 60 хв.

Під час роботи забороняється:

- використовувати рідинні або аерозольні засоби ДЛЯ чишення поверхонь комп'ютера, крім передбачених для цього засобів (для чищення пластмасових поверхонь використовують чисту бавовняну ганчірку, зволожену мильним розчином; екрани та захисні екрани протирають ватою, змоченою спиртом);
- самостійно ремонтувати апаратуру;
- класти будь-які речі на вузли комп'ютера;
- закривати вентиляційні отвори апаратури, що може привести до її перегрівання і виходу з ладу.

Після закінчення роботи необхідно:

- записати в пам'ять комп'ютера файл, що знаходився в роботі;
- вийти з програмної оболонки і повернутися в середовище операційної системи;
- вимкнути системний блок, принтер, інші периферійні пристрої, монітор;
- вимкнути стабілізатор живлення, якщо комп'ютер підключений до мережі через нього;
- штепсельні вилки витягнути з розетки;
- накрити клавіатуру кришкою для запобігання потрапляння на неї пилу;
- прибрати робоче місце.

5.4 Вимоги безпеки праці

Перед початком роботи з ПК працівник зобов'язаний: провітрити робоче приміщення; перевірити стійкість положення устаткування на робочому столі; перевірити відсутність видимих пошкоджень обладнання, дискет в дисководі системного блоку; перевірити справність і цілісність живлять і сполучних кабелів, роз'ємів і штепсельних з'єднань, захисного заземлення (занулення);

Інв. № подл.			_		
11	Зм	Лист	№ докум.	Підпис	Дата

Підпис і дата

Інв.№ дубл.

Замксть.інв. №

Підпис і дата

0095.110577.ДП.2015.001

перевірити справність.

Відрегулювати положення столу, стільця (крісла), підставки для ніг, клавіатури, екрана монітора; освітленість на робочому місці. При необхідності включити місцеве освітлення; протерти поверхню екрану монітора, захисного фільтра (за його наявності) сухою м'якою тканинною серветкою.

Переконатися у відсутності відображень на екрані монітора, зустрічного світлового потоку. Включити обладнання ПК в електричну мережу, дотримуючись наступну послідовність: стабілізатор напруги (якщо він використовується), блок безперебійного живлення, периферійні пристрої (принтер, монітор, сканер та інші пристрої), системний блок.

Забороняється приступати до роботи при: вираженому тремтінні зображення на моніторі; виявленні несправності обладнання; наявності пошкоджених кабелів або проводів, роз'ємів, штепсельних з'єднань; відсутності або несправності захисного заземлення (занулення) обладнання.

У період роботи за ПЕОМ необхідно передбачати перерви для відпочинку, які мають бути оптимальної тривалості (надто тривалі ведуть до порушення готовності до дії та розладу динамічного стереотипу). Тому, через кожні 40-45 хвилин необхідно робити перерву на 3-хвилини, а через 2 години — на 15 хвилин. Середня сумарна тривалість роботи за ПЕОМ на день не повинна перевищувати 4 годин, а на тиждень — 20 годин.

При використанні захисного екрана або монітора із зниженим випромінюванням час роботи за ПЕОМ може бути збільшено.

Шкідливою ϵ одна й та сама поза протягом тривалого часу. Тому в положенні сидячі треба час від часу змінювати фіксовані робочі пози, робити короткочасні перерви.

Під час роботи на ПЕОМ напружуються м'язи рук. Для підтримки їх тонусу під час перерви рекомендується проводити гімнастичні вправи.

3 метою зниження негативного впливу монотонної діяльності доцільно чергувати операції введення тексту і цифрових даних, редагування тексту.

Замксть.інв. №	
Підпис і дата	
Інв. № подл.	

Підпис і дата

.№ дубл.

Забороняється залишати ввімкнені ПЕОМ і пристрої без нагляду. Підключення і відключення роз'ємних кабелів пристроїв ПЕОМ робити тільки при відключенні їх з мережі.

Не можна користуватися біля ПЕОМ аерозолями (дезодорантами, тощо). Не допускати попадання води та інших рідин в середину пристроїв комп'ютера. При наявності електроструму на корпусі припинити роботу, вимкнути ПЕОМ від електромережі, сповістити керівника.

У випадку виникнення у співробітника зорового або дискомфорту інших неприємних суб'єктивних відчуттів, що настають, незважаючи на проходження санітарно-гігієнічних і ергономічних вимог, режимів роботи і відпочинку, варто застосувати індивідуальний підхід в обмеженні часу робіт і корекції тривалості перерв для відпочинку, або проводити заміну іншими видами робіт (не пов'язаних з використанням комп'ютера).

По закінченні роботи працівник зобов'язаний: коректно закрити всі активні завдання; вимкнути живлення системного блоку; вимкнути живлення усіх периферійних пристроїв; відключити блок безперебійного живлення; відключити кабель живлення від мережі; привести в порядок робоче місце; про несправність обладнання та інших зауважень по роботі з ПК повідомити безпосереднього керівника або особам, що здійснюють технічне обслуговування обладнання.

Iнв. № подп. Підпис і дата Замксть.інв. № Інв. № дубл. Підпис і дата

Зм	Лист	№ докум.	Підпис	Дата

ВИСНОВКИ

У наші часи наука і техніка дуже швидко розвиваються та проникають в усі сфери нашого життя. Усе навколо автоматизується та тим самим спрощує життя людям. Розвинені навчальні заклади, наприклад, у США та Європі, не є винятком. Весь учбовий процес в них автоматизований. Розпочинаючи з ведення журналів, відомостей, календарів, бухгалтерських операцій, тощо, закінчуючи дистанційним навчанням.

У цьому дипломному проекті була розроблена перша версія такої системи, яка буде робити процес навчання більш легким та зручним для студентів та викладачів завдяки надання можливості дистанційного доступу до необхідної їм інформації.

Ця програма розроблена у вигляді веб-сайту. Тому працівники університету та студенти зможуть користуватися нею з будь-якого персонального комп'ютера або мобільного приладу, який має вихід в інтернет.

Студенти зможуть виконувати в цій системі такі функції:

- переглянути інформацію про університет, факультети, кафедри, спеціальності та групи;
- побачити контактну інформацію викладачів, та розклад їх консультацій;
- переглянути інформацію щодо навчання (завдань та оцінок, занять та відвідувань) та фінансів (список платежів) у своїх групах.

Викладачі зможуть:

- вести журнали відвідувань та оцінок зі своїх предметів;
- побачити контактну інформацію студентів;
- переглянути інформацію про університет, факультети, кафедри, спеціальності та групи.

Завдяки усім цим перевагам студенти та співробітники збережуть багато своїх сил та часу, та зможуть зосередитись на найголовнішому – навчальному процесі.

нв. № подл. Підпи	ıc i дата	Замксть.інв. №	H8.

Підпис і дата

№ дубл.

Звичайно, як і будь-яка інша система, даний програмний продукт може розвиватися та доповнюватися новими функціями, для того щоб ще більше полегшити процеси щодо навчання та усі інші процеси пов'язані з ним. Наприклад, можна покращити систему такими функціями:

- мобільний додаток під Android, iOS чи Windows Phone, для того, щоб користувачам мобільних телефонів та планшетів було більш зручно користуватися системою;
- впровадження технології AngularJS або її аналогів для того, щоб зробити клієнтський додаток більш інтерактивним та швидким;
- покращення інтерфейсу користувача з метою зробити його більш зручним та зменшити час роботи із системою;
- автоматизація роботи бібліотеки та ведення електронної бібліотеки;
- впровадження можливості генерації документів у програмі та їх електронного підписання за допомогою перевірених та надійних API, наприклад, RightSignature;
- впровадження календаря подій студентів та викладачів;
- впровадження форуму для спілкування між студентами та викладачами, та передачі файлів щодо навчального процесу;
- впровадження зворотного зв'язку з адміністрацією сайту з приводу проблем чи питань, що виникнули під час використання системи;
- відправка повідомлення з системи при різних подіях у вигляді Email та SMS;
- генерація різних звітів та статистик щодо викладачів, студентів, груп, фінансових та навчальних даних;
- проведення фінансових операції через API банків та інших платіжних систем;
- ведення журналів дій користувачів, відправлених Email та SMS.

Незважаючи на таку велику кількість можливих покращень, дана система вже на даному етапі ϵ повність працездатною та готовою до

Iнв. № подл. Підпис і дата Замксть.інв. № Інв.№ дубл. Підпис і дата

Зм.. Лист № докум. Підпис Дата

0095.110577.ДП.2015.001

	використання в університеті. У ній присутні всі базові функції, необхідні для полегшення організації учбового процесу, та вже зараз вона може принести багато користі як студентам, так і викладачам.	
Підпис і дата		
Інв. № дубл.		
Замксть.інв. №		
Підпис і дата		
Інв. № подл.	Даркуш м Лист № докум. Підпис Дата 0095.110577.ДП.2015.001	_ , _

Γ

ЛІТЕРАТУРА

- 1. Що таке AJAX, його переваги та недоліки [Електронний ресурс] // Режим доступа до сайту: http://promo.ingate.ru/seo-wikipedia/ajax/
- 2. Керівництво з проектування баз даних [Електронний ресурс] // Режим доступа до сайту: http://habrahabr.ru/post/193136/
- 3. Основи MVC. З чого розпочати? [Електронний ресурс] // Режим доступа до сайту: http://ruseller.com/net.tutsplus.com/tutorials/other/mvc-for-noobs/
- 4. Основні принципи ООП [Електронний ресурс] // Режим доступа до сайту: http://gos-it.wikia.com/wiki/OOP
- Буч Г., Якобсон И. Язык UML. Руководство пользователя / Г. Буч,
 И. Якобсон М.: ДМК Пресс, 2006. 496 с.
- 6. ДСТ 12.0.003-74. «ССБТ. Небезпечні і шкідливі виробничі фактори. Класифікація»
- 7. СанПіН 2.2.2/2.4. 1340-03 «Гігієнічні вимоги до персональних електроннообчислювальних машин та організації роботи»
- 8. ДСанПіН 3.3.2.007-98 «Державними санітарними правилами і нормами роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»
- 9. НПАОП 0.00-4.12-05 «Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці»НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів»
- 10. СанПиН 2.2.2.542-96 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы»

Підпис і дата	
Інв.№ дубл.	
Замксть.інв. №	
Підпис і дата	
в. № подл.	

Зм.. Лист № докум. Підпис Дата

0095.110577.ДП.2015.001

ДОДАТКИ

Інв. № подл. Підпис і дата Замксть.інв. № Інв.№ дубл. Підпис і дата

Зм	Лист	№ докум.	Підпис	Дата

0095.110577.ДП.2015.001

Аркуш

86