

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«СТВОРЕННЯ UI/UX ДЛЯ LEARNING-ДОДАТКУ «CAMPUS»
ДЛЯ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ВЗАЄМОДІЇ З КІНЦЕВИМ
КОРИСТУВАЧЕМ»**

Виконав: студент 4 курсу, групи ПД-42
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Конішевський В.І.
(прізвище та ініціали)

Керівник Коба А.Б.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Київ –2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти -«Бакалавр»
Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри
Інженерії програмного
забезпечення

Негоденко О.В.

“ ____ ” _____ 2021 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

КОНШЕВСЬКОМУ ВЛАДИСЛАВУ ІГОРОВИЧУ

(прізвище, ім'я, по батькові)

1. Тема роботи: «створення ui/ux для learning-додатку «campus» для підвищення ефективності взаємодії з кінцевим користувачем»

Керівник роботи: Коба А.Б., старший викладач кафедри ІІЗ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» березня 2021 року №65.

2. Строк подання студентом роботи «1» червня 2021 року

3. Вхідні дані до роботи

Методи обробки зображень;

Технічна література з питань, пов'язаних з проектуванням, оцінюванням, вимірюванням користувацького інтерфесу для систем управління навчанням (LMS)

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Поняття ці/их дизайну та його роль в сучасних веб-додатках.

4.2 Вимоги до інтерфейсу користувача.

4.3 UX-проектування та .реалізація дизайну інтерфейсу користувача.

4.4 Тестування користувацького інтерфейсу.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Актуальність проблеми

2. User Interface (користувацький інтерфейс)

3. User Experience (користувацький досвід)

4. Transitions basemap (Базова карта переходів) додатку

5. Користувацькі сценарії

6. Тестування інтерфейсу користувача

7. Інструменти аналізу інтерфейсу користувача

6. Дата видачі завдання «19» квітня 2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір технічної літератури	20.04-25.04	Виконано
2	Вимоги до дизайн-системи	26.04-30.04	Виконано
4	Розробка та проектування користувацького інтерфейсу	1.05-15.05	Виконано
5	Побудова дизайн-концепції продукту	16.05-20.05	Виконано
6	Реферат, Вступ, висновки	21.05-23.05	Виконано
7	Розробка демонстраційних матеріалів	24.05-29.05	Виконано
8	Попередній захист роботи	30.05	
9	Здача роботи	1.06	

Студент _____
(підпис)

(прізвище та ініціали)

Керівник роботи _____
(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 54с. 20 рис., 18 джерел

Об'єкт дослідження: централізована система організації учбового процесу студентів вищих навчальних закладів інженерних спеціальностей та слухачів зовнішніх курсів для підготовки майбутніх ІТ спеціалістів.

Предмет дослідження: дизайн система веб застосунку для обміну та аналізу інформації стосовно навчального графіку, організації управління завданнями/сторонніми подіями студентів освітніх установ

Мета роботи: Дослідити сучасні системи управління навчанням, спроектувати та реалізувати повну дизайн концепцію веб-додатку «Campus». Здобути практичні навички у проектуванні дизайн-систем сучасних веб-додатків

Методи дослідження – методи теорії інформації, методи проектування інтерфейсу користувача , методи тестування, реєстрації, верифікації та валідації поведінкових технік користувача. Визначено оптимальний склад, положення ,візуальне оформлення інтерфесу, яким буде користуватися кінцевий користувач додатку

Здійснено ряд робіт зв'язаних з дослідженням сучасних тенденцій розвитку популярних навчальних веб-додатків та їх візуальної складової

На основі результатів виконаних досліджень розроблена схема та план по реалізації дизайн системи яка згодом буде імплементована у візуальну частину веб-додатку

Упровадження розробленої системи дозволяє зрозуміти та покращити існуючі підходи до розробки/проектування UI/UX дизайну що мають відповідати сучасним тенденціям щодо розробки , експлуатації а в подальшому - підтриманні такої системі в належному(зрозумілому та зручному) для користувача стані

Ключові слова: UI/UX DESIGN, DESIGN SYSTEM,WIREFRAME, PROTOTYPING, USER INTERFACE ANALYSIS

ЗМІСТ

1. ПОНЯТТЯ UI/UX ДИЗАЙНУ ТА ЙОГО РОЛЬ В СУЧАСНИХ ВЕБ-ДОДАТКАХ	12
1.1 Концепції сучасного веб-дизайну	12
1.2 User Interface (UI)	14
1.3 User Experience	16
1.4. Огляд та аналіз існуючих додатків LMS	20
1.4.1 Moodle.	20
1.4.2 – OpenOlat.	21
1.4.3 – Google Classroom.	22
1.5. Вибір середовища розробки/проектування UI/UX.....	24
2. ВИМОГИ ДО ІНТЕРФЕЙСУ КОРИСТУВАЧА	26
2.1 Загальний опис проекту.....	26
2.2 Вимоги до дизайну «Campus»	27
2.3 Вимоги до дизайн-концепції продукту.....	28
2.3.1 Дослідження і постановка гіпотез:	28
2.3.2 Ідея концепту	29
2.3.3 Модулі	30
3. UX ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ДИЗАЙНУ ІНТЕРФЕЙСУ КОРИСТУВАЧА	32
3.1 Побудування User Flow	32
3.2 Моделювання Customer Journey Map (Карта подорожі клієнта)	34
3.3 Основні етапи створення дизайну інтерфейсу користувача	40
3.4 Реалізація дизайну та прототипування	42
3.4.1 Модуль «Dashboard».....	43
3.4.2 Модуль «Calendar»	46
3.4.3 Модуль «ClassRooms»	48
4. ТЕСТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА	51
4.1 Основні етапи тестування.....	51

4.2 Проведення тестування користувацького інтерфейсу на основі розробленої дизайн-системи.....	53
4.2.1 Дослідницьке тестування	53
4.2.2 Оцінкове тестування.....	54
4.2.3 Валідаційне тестування.....	57
4.2.4 Порівняльне тестування:	57
5. РЕАЛІЗАЦІЯ ІНСТРУМЕНТУ КІЛЬКІСНОГО АНАЛІЗУ ІНТЕРФЕЙСУ КОРИСТУВАЧА	59
5.1 Опис використаних технологій	59
5.2 Вид обчислень та схема роботи інструменту	60
5.3 Процес збору даних та обчислень	62
ВИСНОВКИ	65
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	66
ДОДАТКИ	68

ВСТУП

Стрімкий розвиток технологій по всьому світу привів неперевершених змін до в історії людства. Сьогодні, ми, через зрозумілий нам програмний інтерфейс можемо в один клік замовити потрібну послугу, дізнатися та передати вмиль необхідну інформацію. Сьогоднішні технології є доволі складними в розумінні, але тим не менше робота з ними нам здається легкою адже ми в першу чергу ітеруємо саме з інтерфейсом користувача, який максимально допомагає виконати поставлену користувачем ціль в зрозумілій та доступній для нього формі.

Але так було не завжди: наприклад, коли з'явилися найперші в світі комп'ютери та принтери, вони не були призначені для масового користування, були громіздкими, дорогими і не зручними. Тому користувалися ними лише самі інженери, або вузькі фахівці своєї справи, і необхідності в UX дизайні не було.

Час минав, ситуація на ринку технологій змінювалася, в життя звичайної людини увійшли нові девайси, дешевші комп'ютери і інша організаційна техніка. Так як оргтехніка стала доступна звичайному користувачеві, який не має досвіду спілкування з нею, з'явилася необхідність в проектуванні простих і доступних для розуміння інтерфейсів.

На ранній стадії розвитку комп'ютерної ери, користувачі на собі відчули ті незручні віконців в Windows, незрозумілі сайти для покупок і т.д. З'явився попит, тому стало з'являтися і пропозиції. Незручні сайти / девайси стали причиною втрати часу і грошей для людей, тому що при роботі з ними було не зрозуміло як оптимально ними користуватися. З іншого боку згодом отримавши гарні, прості і приємні інтерфейси люди перестали думати, що це "комп'ютер", стали більше довіряти, приділяючи час роботи на "комп'ютері". У них виходило вивести гроші з банкомату, послухати свою музику в плейлісті і вони перестали відчувати почуття ніби треба щось програмувати. І тому метою UX

дизайну стало зробити досвід користувача зрозумілим, а не незграбним і складним.

Специфіка джерельної бази: Більшість матеріалів, що стосуються UI/UX на сьогоднішній день знаходяться в мережі інтернет. Коли мова йде про проектування дизайн-системи важливо підібрати «правильну» стилізацію продукту – тобто, щоб стильова система тільки допомагала користувачеві дійти до своєї цілі, а не відштовхнути користувача від наміченого чи призупиняти його рух до наміченого. Основу джерельної бази складають матеріали таких веб-ресурсів :

- dribbble.com;
- behance.net;
- lawsofux.com.

Об'єкт дослідження: Дистанційне навчання та централізована система організації учбового процесу студентів вищих навчальних закладів інженерних спеціальностей та слухачів зовнішніх курсів для підготовки майбутніх ІТ спеціалістів.

Предмет дослідження: візуальна частина(UI/UX) веб застосунку для обміну та аналізу інформації стосовно навчального графіку, організації управління завданнями/сторонніми подіями студентів освітніх установ

Мета: Дослідити сучасні системи управління навчанням, спроектувати та реалізувати повну дизайн концепцію веб-додатку «Campus». Здобути практичні навички у проектуванні дизайн-систем сучасних веб-додатків

Завдання: В процесі розробки дизайн-концепції продукту вирішувалися наступні завдання:

- Час ітерації користувача(на 1 функціональність) на робочому екрані
- Ідентифікація індивідуальної стилізації веб-додатку
- UX проектування інтерфейсу користувача

Методи дослідження: методи теорії інформації, методи проектування інтерфейсу користувача, методи тестування, реєстрації, верифікації та валідації поведінкових технік користувача.

Наукова новизна роботи полягає у наступному:

1. Був проведений повний цикл робіт щодо розробки UI/UX дизайну веб-додатку - дослідження та призначені для користувача сценарії, розробка структури інтерфейсу, прототипування інтерфейсу, визначення стилістики, дизайн концепція, оформлення всіх екранів, підготовка матеріалів для розробників)

2. На підставі загальної теорії сучасних веб-сервісів було проведено аналіз аналогів систем управління навчанням як об'єктів проектування, які характеризуються певними виконуваними функціями і багатокomпонентною структурою.

Практична значущість результатів дослідження:

1. Показано основні етапи створення користувацького інтерфейсу: починаючи з ескізів, прототипів низької якості, закінчуючи повноцінним(фінальним) компонентом дизайн системи

2. Показано UX проектування інтерфейсу користувача використовуючи для цього такий перелік інструментів: user flow(користувацькі сценарії), Customer Journey Map (Карта подорожі клієнта), user research (дослідження користувача)

1. ПОНЯТТЯ UI/UX ДИЗАЙНУ ТА ЙОГО РОЛЬ В СУЧАСНИХ ВЕБ-ДОДАТКАХ

1.1 Концепції сучасного веб-дизайну

UX - це User Experience (Користувацький досвід). Це про те, який досвід та враження отримує користувач від роботи з вашим інтерфейсом. Чи вдається йому досягти мети і на скільки просто або складно це зробити [1].

А UI - це User Interface (Інтерфейс Користувача) – це про те, як виглядає інтерфейс і те, які фізичні характеристики набуває. Визначає, якого кольору буде ваш «виріб», чи зручно буде людині взаємодіяти з певними механізмами (кнопка, віджет, контекстне меню і т.д), читабельним чи буде текст тощо.[1]

UX/UI дизайн - це проектування будь-яких призначених для користувача інтерфейсів в яких зручність використання так само важливо як і зовнішній вигляд [3].

Фундаментальні концепції сучасного UI/UX

1. Візуальний дизайн. Вдало підібраний візуальний дизайн сприяє створенню позитивного досвіду для користувачів. Кінцевий користувач не бачить усіх нюансів інформаційної архітектури, але може помітити певні візуальні ефекти. Візуальний дизайн у додатку означає все, що користувач бачить на сторінці. Це може бути що завгодно - від зображень та макета до ілюстрацій типографіки.

2. Інтерфейс користувача. Коли справа стосується UI та UX, їх часто плутають і використовують як взаємозамінні. Найпростіший спосіб розрізнити його - сказати, що користувацький інтерфейс - це те, що користувач може бачити та взаємодіяти. Незважаючи на те, що взаємодія з користувачами полягає в орієнтованих на користувача аспектах системи, користувацький інтерфейс відіграє важливу роль у примушуванні користувачів залишатися у вашому додатку.

3. Юзабіліті. Юзабіліті - це те, що може стати вашим додатком для виграшу чи програшу. Наприклад, якщо аудиторія з самого початку не користується додатком належним чином спочатку, то надалі вона не захоче використовувати його взагалі. Перший виклик юзабіліті, - перший досвід користувача. Тому зручність використання додатків має бути належним чином запроектованою.

4. Інформаційна архітектура. Інформаційна архітектура програми полягає у створенні структури програми та організації інформації таким чином, щоб допомогти користувачеві зрозуміти програму/додаток. Це неймовірно важлива частина проектування, оскільки вона дозволяє користувачам легко шукати всю необхідну інформацію. Користувачам потрібна ясність, а інформаційна архітектура допомагає це отримати. Якщо дизайнерам не вистачає належної інформаційної архітектури програми, користувач залишиться розгубленим.

5. Функціональність. Функціональність програми допомагає користувачам виконати бажані дії. Кожен мобільний додаток має свої власні та унікальні функціональні можливості відповідно до платформи, для якої створений додаток. Близько 88% користувачів програми рідше повертаються до програми після поганого досвіду. Тому функціональність програми повинна бути унікальною та зручною для користувача. Ідеально розроблена функціональність програми підвищує ефективність і зменшує часові затрати на взаємодії додатку і кінцевого користувача.

6. Контент. Контент допомагає залучити більше користувачів на додатку/веб-сайті та надати їм значущу інформацію. Вміст може бути будь-яким: путівниками, подкастами, блогами, соціальними медіа чи інформацією про товари. Вміст може створити велике враження на думку користувача про ваш бренд.

7. Дизайн взаємодії. Дизайн взаємодії - це все про дії, клацання тапи та прокрутки. Якби дії користувачі не вживали під час використання вашого додатку, вони становлять невід'ємну частину користувацького досвіду. Цей концепт розширює межі функціональності, створюючи захват від кожної взаємодії користувача.

1.2 User Interface (UI)

Інтерфейс користувача, який також називається "UI" - це засіб, за допомогою якого людина керує програмним додатком або апаратним пристроєм. Хороший користувацький інтерфейс забезпечує "зручний" досвід, дозволяючи користувачеві взаємодіяти з програмним або апаратним забезпеченням природним і зрозумілим способом.

Майже всі програми мають графічний користувацький інтерфейс, або графічний інтерфейс користувача (GUI). Це означає, що програма включає в себе графічні елементи управління, які користувач може вибрати за допомогою миші або клавіатури. Типовий графічний інтерфейс програми включає в себе рядок меню, Панель інструментів, вікна, кнопки та інші елементи управління.

Операційні системи Macintosh і Windows мають різні інтерфейси користувача, але вони мають багато спільних елементів, таких як робочий стіл, вікна, значки і т.д. ці загальні елементи дозволяють людям використовувати будь-яку операційну систему без необхідності заново вивчати інтерфейс. Аналогічно, такі програми, як текстові процесори і веб-браузери, мають досить схожі інтерфейси, забезпечуючи узгоджений користувацький інтерфейс між декількома програмами [5].

Більшість апаратних пристроїв також включають користувацький інтерфейс, хоча він зазвичай не такий складний, як програмний інтерфейс. Поширеним прикладом апаратного пристрою з призначенням для користувача інтерфейсом є пульт дистанційного керування.

Типовий пульт дистанційного керування телевізором має цифрову клавіатуру, кнопки гучності і каналу, кнопки відключення звуку і харчування, селектор введення і інші кнопки, що виконують різні функції. Цей набір кнопок і те, як вони розташовані на контролері, складають користувальницький інтерфейс. Інші пристрої, такі як цифрові камери, мікшерні пульти і стереосистеми, також мають користувальницький інтерфейс.

Хоча користувальницькі інтерфейси можуть бути розроблені як для апаратного, так і для програмного забезпечення, більшість з них є комбінацією обох. Наприклад, для управління програмним забезпеченням зазвичай потрібно використовувати клавіатуру і мишу, кожна з яких має свій власний користувальницький інтерфейс. Аналогічно, щоб керувати цифровою камерою, вам може знадобитися переміщатися по екранним меню, яке є програмним інтерфейсом. Незалежно від програми, мета хорошого інтерфейсу користувача - бути зручним для користувача. Зрештою, ми всі знаємо, як неприємно використовувати пристрій, який працює не так, як ми хочемо.

Користувальницький інтерфейс створюється в сферах взаємодії, які звертаються до людських почуттів (зір, дотик, слух і багато іншого). Вони включають в себе як пристрої вводу, такі як клавіатура, миша, трекпад, мікрофон, сенсорний екран, сканер відбитків пальців, електронне перо і камеру, так і пристрої виведення, такі як монітори, динаміки і принтери. Пристрої, що взаємодіють з декількома органами чуття, називаються "мультимедійними інтерфейсами". Наприклад, повсякденний користувальницький інтерфейс використовує комбінацію тактильного введення (Клавіатура і миша) і візуального і слухового виведення (монітор і динаміки).

Інші типи користувальницьких інтерфейсів можуть включати:

- Користувальницький інтерфейс на основі форм;
- Графічний інтерфейс користувача;
- Користувальницький інтерфейс, керований меню;
- Сенсорний користувальницький інтерфейс;

- Голосовий користувальницький інтерфейс.

Користувальницький інтерфейс на основі форм використовується для введення даних в Програму або додаток, пропонуючи обмежений вибір варіантів. Наприклад, меню налаштувань на пристрої засноване на формі [5];

Графічний інтерфейс користувача - тактильний введення користувальницького інтерфейсу з візуальним «висновком» користувальницького інтерфейсу (клавіатура і монітор);

Користувальницький інтерфейс, керований меню - користувальницький інтерфейс, який використовує список варіантів для навігації по програмі або веб-сайту. Наприклад, банкомати використовують користувальницький інтерфейс, керований меню, які є простими у використанні для більшості користувачів;

Сенсорний користувальницький інтерфейс - користувальницький інтерфейс за допомогою гаптики або дотику. Більшість смартфонів, планшетів і будь-яких пристроїв, що працюють з сенсорним екраном, використовують тактильний ввід;

Голосовий користувальницький інтерфейс - взаємодія між людиною і машиною за допомогою слухових команд. Приклади включають в себе віртуальні помічники, розмови з текстом, GPS і багато іншого.

1.3 User Experience

Користувальницький досвід (UX) - це те, що користувач конкретного продукту відчуває при використанні цього продукту. Таким чином, завдання UX-дизайнера полягає в створенні продукту, що забезпечує найкращий користувальницький досвід. Але як це відбувається?

Все починається з великої кількості досліджень. Неможливо створити нічого цінного для користувача, якщо не розуміти, які проблеми треба вирішити і як їх треба вирішити, щоб користувач захотів ,або, ще краще, потребував у

вашому рішенні. Отримати це розуміння можна тільки взаємодіючи з користувачами.

T-модель і стратегії практиків ІА: у 2004 році фахівець з UX-дизайну Пітер Бурсма припустив, що Інформаційна архітектура є однією з багатьох дисциплін, які об'єднуються для формування міждисциплінарної практики проектування користувацького досвіду для Інтернету. Він назвав діаграму, яку використовував для вираження цієї концепції, T-моделлю, показаною на малюнку. До того, як Бурсма сформулював цю точку зору, багато інформаційних архітекторів розглядали практику інформаційної архітектури як всеосяжний парасольку, називаючи її великою ІА.

З тих пір як Бурсма створив свою діаграму, багато шанованих фахівців в області інформаційної архітектури зайняли аналогічну позицію. У своєму пленарному засіданні на Summit 2008 року Ендрю Хінтон описав практику інформаційної архітектури як одну з багатьох дисциплін, які вносять свій внесок у більш широку практику проектування користувацького досвіду. І в своїй останній книзі "Проникна Інформаційна архітектура: проектування міжканального досвіду" Андреа Ресміні та Лука Розаті погодилися з Бурсмою, посилаючись на практику інформаційної архітектури як на необхідну частину процесу проектування користувацького досвіду.

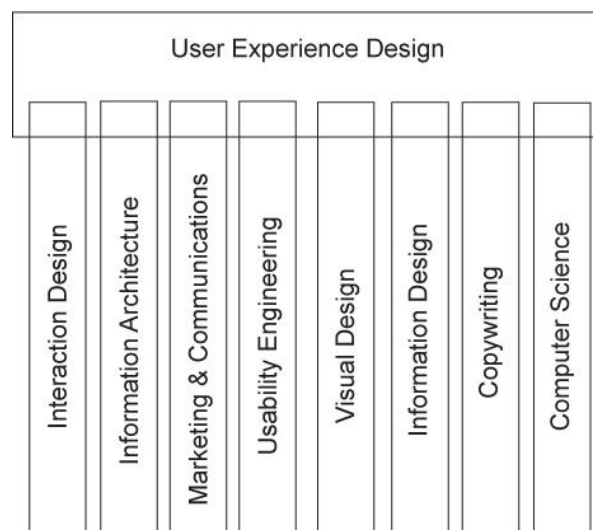


Рисунок 1 – Складові дизайну користувацького досвіду

Важливість користувацького досвіду проявляється як при розробці так і при оптимізації веб-сайту, орієнтованого на певну аудиторію.

З іншого боку, UX-це створення цифрового інтерфейсу, який працює саме так, як очікують користувачі програмного продукту. Добре спроектований користувацький інтерфейс -це коли взаємодія інтуїтивно зрозуміла і настільки добре продумана, що користувачам навіть не потрібно думати.

Коли справа доходить до веб-дизайну, відвідувачі є потенційними клієнтами . Тому треба бути впевненим, що програмний продукт відображає всі елементи і пропонує такий рівень взаємодії, який дозволить відвідувачам завжди повертатися за новими.

Інші концепції, про які потрібно думати, коли ви розробляєте свій сайт для максимальної зручності користувачів:

Лояльність - якщо у ваших відвідувачів є поганий досвід на вашому сайті, вони не повернуться найближчим часом. Цей втрачений користувач міг стати клієнтом.

З іншого боку, якщо програмний продукт вражає аудиторію, ви почнете виховувати лояльність тільки з перших кількох взаємодій з сайтом. Чим довше ви будете утримувати відвідувачів на сайті за допомогою кращих практик UX, тим більш лояльними стануть ваші клієнти, що являє собою найкращий сценарій довгострокового успіху бізнесу.

Крім того, є всі ті рекомендації, які ви отримаєте, оскільки лояльні клієнти, як правило, стають найгучнішими захисниками бренду.

Рентабельність інвестицій і коефіцієнти конверсії - коли ваш користувацький досвід настільки ефективний, наскільки це можливо, ваша норма прибутку обов'язково зростає. По суті, це означає, що ви вкладаєте в свій

бізнес рівно стільки грошей, щоб зробити його улюбленим Користувачем, і тепер ви пожинаєте плоди з більш високим прибутком.

Звичайно, ваш успіх вимірюється не тільки матеріально, покращений UX може зробити вашу подорож покупців більш впорядкованим, скоротивши час від початку до продажу. Це, у свою чергу, підвищує якість ваших потенційних клієнтів, оскільки тільки ті, хто має сильний інтерес, перейдуть до конверсії, і ви, швидше за все, отримаєте більше прибутку від щасливих і задоволених клієнтів.

Конверсія, є найважливішою метрикою, яку може мати веб-сайт, оскільки саме стільки потенційних клієнтів перетворилося на реальних клієнтів. Розрахуйте коефіцієнт конверсії вашого сайту, щоб виміряти ефективність UX вашого сайту, і спостерігайте, як ця цифра зростає в міру того, як ви продовжуєте фокусуватися на наданні відвідувачам максимального користувацького досвіду.

Ефективність та / або продуктивність - коли ваш UX знаходиться на місці, відвідувачі можуть знайти те, що вони шукають на вашому сайті набагато швидше і легше, сприяючи меншій кількості помилок.

Якщо ваш UX використовується для поліпшення бізнес-інтрамережі, наприклад, підтримка ефективності процесів допомагає співробітникам підтримувати високий рівень продуктивності. Але навіть відвідувачі сайту оцінять можливість зробити більше за менший час.

Подумайте про це з точки зору електронної комерції. Чим швидше ви зможете перетворити потенційних клієнтів на клієнтів, тим більше у вас буде шансів щодня заробляти більше продажів.

Задоволеність клієнта - думайте про відвідувачів вашого сайту як про клієнтів, які хочуть легкого і приємного вирішення своїх проблем. Коли ваш сайт пропонує ці рішення у швидкій і інтуїтивно зрозумілій формі, вони, швидше за все, залишаться, затримуються і повернуться за великим.

1.4. Огляд та аналіз існуючих додатків LMS

1.4.1 Moodle.

Система управління навчанням (LMS) - це програмне забезпечення, розроблене спеціально для створення, розповсюдження та управління доставкою освітнього контенту. LMS може розміщуватися як окремий продукт на сервері компанії, або це може бути хмарна платформа, яку розміщує фірма програмного забезпечення.

Moodle - це навчальна платформа, призначена для надання викладачам, адміністраторам та учням єдиної надійної, безпечної та інтегрованої системи для створення персоналізованих навчальних середовищ

Moodle розшифровується як Модульне об'єктно-орієнтоване динамічне навчальне середовище. Заснований і розроблений Мартіном Дугіамасом у 2002 році, Moodle був розроблений, щоб надати викладачам, адміністраторам та учням відкриту, надійну, безпечну та безкоштовну платформу для створення та забезпечення персоналізованих навчальних середовищ. Moodle - це зручна система управління навчанням (LMS), яка підтримує потреби у навчанні та навчанні для широкого кола установ та організацій по всьому світу.

На сьогодні Moodle є найбільш широко використовуваною системою управління навчанням у світі, в якій зареєстровано понад 100 000 реалізацій у всьому світі, що підтримують понад 150 мільйонів учнів. Проектом Moodle з відкритим кодом керує спеціалізована команда у штабі Moodle із головним офісом у Перті, Австралія та супутниковими офісами по всьому світу.

Модульний характер і властива Moodle гнучкість роблять його ідеальною платформою як для академічного, так і для корпоративного рівня будь-якого розміру.

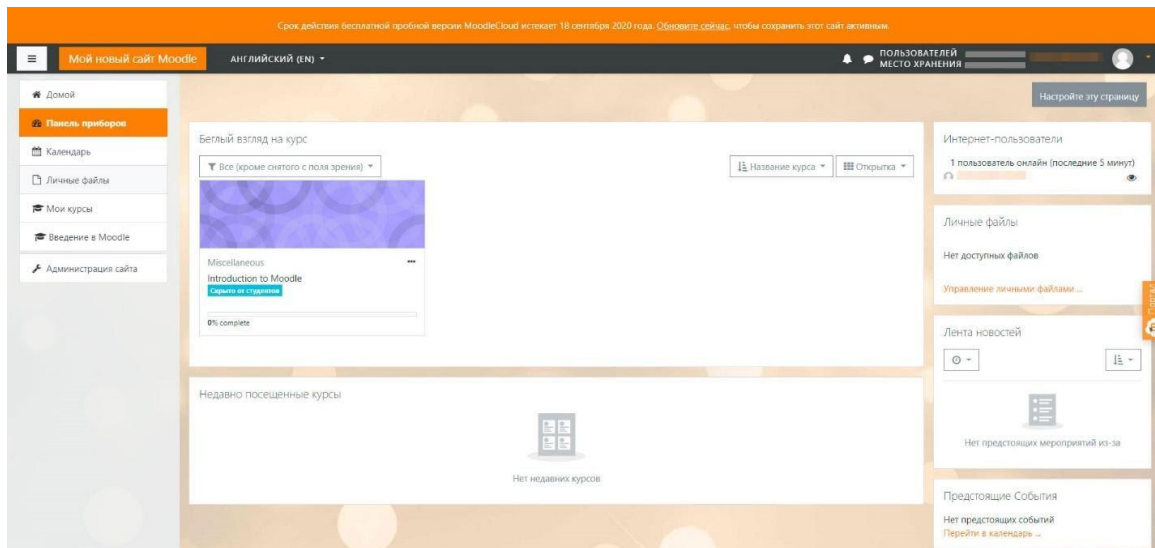


Рисунок 1.1 - Moodle LMS

1.4.2 – OpenOlat.

OpenOlat - це навчальна платформа, що базується на Інтернеті, для викладання, навчання, оцінювання та спілкування, LMS, система управління навчанням. OpenOlat вражає своєю простою та інтуїтивно зрозумілою роботою.

OpenOlatT не є системою управління вмістом (CMS). Оскільки OLAT означає "Інтернет-навчання та тренування"(Online Learning and Training), основною метою не є створення навчального матеріалу. Тому додаткові інструменти слід використовувати під час створення або редагування більших обсягів навчального контенту (наприклад, eLML). Навчальний вміст, створений зовні, можна імпортувати в OpenOLAT за допомогою стандартизованих інтерфейсів. Однак для того, щоб мати можливість створювати прості сторінки в OpenOLAT, існують відповідні інструменти, такі як редактор HTML.

Складна модульна система пропонує авторам курсів широкий спектр дидактичних можливостей. Кожну установку OpenOlat можна індивідуально розширити і, таким чином, адаптувати до організаційних потреб та інтегрувати в існуючі IT-структури. Архітектура, розроблена для мінімального споживання ресурсів, масштабування та безпеки, гарантує надзвичайно надійну роботу.

The screenshot displays the 'Overview' page of the OpenOlat LMS. At the top, there are navigation tabs for 'Overview', 'Users / course structure', and 'Groups'. Below this, a blue header reads 'Overview'. A grey box contains the text: 'By means of this tool you can assess your course participants. You can see test or assessment results and modify them.'

On the left, under 'List of assessable items', it shows: 5 users, 5 group(s), and 23 course element(s). On the right, 'Quick links' shows: 0 users passed and 0 users failed.

The 'Pending reviews' section contains a table with the following data:

User name	Last name	First name	Elements
cmuster	Muster	Christoph	▲ 2 pending
emuster	Muster	Elisabeth	▲ 2 pending
dmuster	Muster	Daniel	▲ 2 pending
amuster	Muster	Andreas	▲ 5 pending
bmuster	Muster	Beate	▲ 5 pending

The 'Statistics overview' section shows:

Number of registered participants	5
Number of not-registered users	3
Opened resource at least once	● 5 / 3
Average	
Passed	

At the bottom, the 'Notification service' section includes: 'Notification regarding new test results of this course:' with a 'Modifications ON' button, and 'Generated certificates' with a 'Modifications ON' button.

Рисунок 1.2- OpenOlat LMS

1.4.3 – Google Classroom.

Google Classroom дозволяє викладачам створити онлайн-класну кімнату, в якій вони можуть управляти усіма документами, які потрібні їх студентам. Документи зберігаються на Google Диску та можуть редагуватися в додатках Диска, таких як Документи Google, Таблиці тощо. Але що відрізняє Google Classroom від звичайної роботи на Google Drive, це інтерфейс викладача / учня, який Google розробив для того, як вчителі та студенти думають і працюють.

Google Classroom знаходиться між вами та Google Drive і забезпечує зручний для викладачів / студентів спосіб управління документами в класі. Ось що ви можете зробити з Google Classroom(Рис 1.3):

Виконуйте завдання: Головне, що ви, як вчитель, будете робити з Google Classroom - це робити домашні завдання для своїх учнів. Коли ви створюєте завдання, ви можете завантажити необхідні документи, щоб учні могли читати їх або працювати над ними. Студенти отримують електронною поштою повідомлення про нове завдання. Студенти “здають” завдання, коли закінчують, а потім ви можете оцінювати завдання.

Здійснюйте оголошення: Якщо у вас є швидке повідомлення для всього класу, ви можете швидко ввести оголошення, яке буде надіслано електронною поштою всім вашим учням класу.

Зберігайте матеріали в класі: матеріали, які ви додаєте до завдання, - це не єдині документи, які ви можете зберігати. Ви можете зберігати будь-які інші необхідні документи для студентів на Google Drive.

Дозволити студентам взаємодіяти: студенти мають можливість коментувати завдання та оголошення, а також надсилати електронною поштою один одному через інтерфейс Класу.

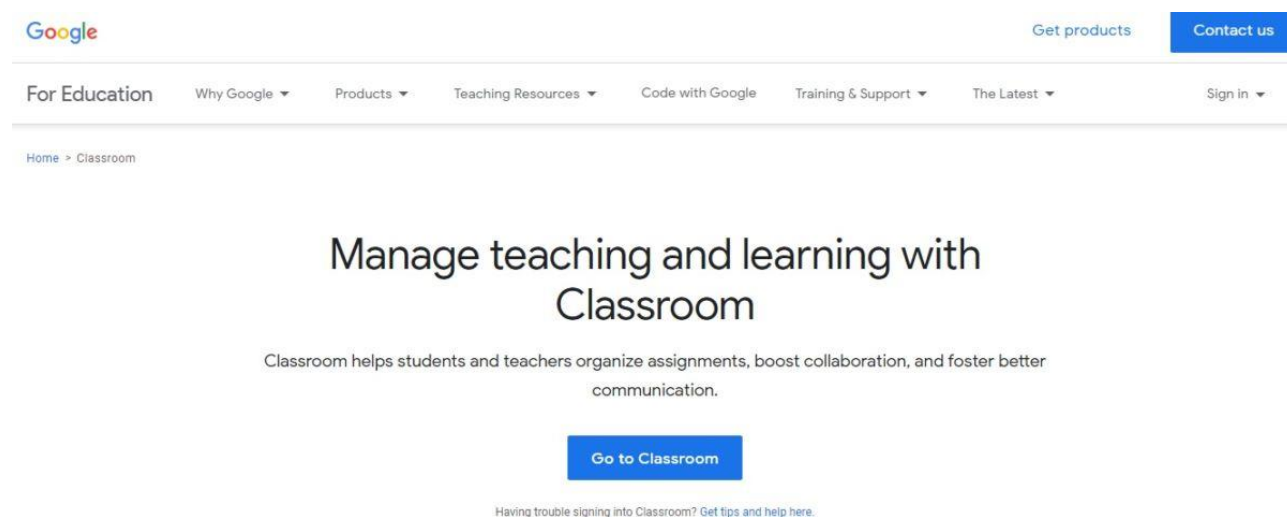


Рисунок 1.3 - Веб-додаток Google Classroom

В результаті аналізу аналогів learning-додатку «Campus» було складено наступну таблицю з порівнянням за наступними властивостями, притаманним системам управління навчанням:

Таблиця 1.1 Порівняння аналогів додатку

Властивість	Campus	Google Classroom	Open Olat	Moodle
Система push повідомлень	Присутня	Не знайдено	Присутня	Присутня
Запуск програмного коду інтерактивно у браузері	Присутня	Не знайдено	Не знайдено	Не знайдено
Розподіл дисциплін по віртуальним кімнатам	Присутня	Присутня	Присутня	Присутня
Авто присвоювання оцінки	Присутня	Не знайдено	Присутня	Не знайдено
Виявлення фальшивого зображення профілю	Присутня	Не знайдено	Не знайдено	Не знайдено

1.5. Вибір середовища розробки/проектування UI/UX

Розглянемо 3 найбільш популярних інструментів для комплексного дизайну та анімації інтерфейсів (UI/UX).

Sketch - це векторний графічний редактор, розроблений голландською компанією Bohemian Coding. Використовується для проектування інтерфейсів мобільних додатків і веб-сайтів. Основним мінусом є те що дане програмне забезпечення доступне лише для macOS, та не має дистрибутивів для інших операційних систем.

Sketch, однак, не призначений для малювання - він насправді призначений для створення прототипів. Одні з можливостей його можливостей: визначати гарячі точки; призначати області екрана як прокручувані / непрокручувальні; а

також екземпляри посилань та перехресних посилань, меню, сторінки, розділи та символи. Ви також є можливість імпортувати зображення або текст, щоб заповнити прототип більш реалістичними даними, ніж прості заповнювачі. Sketch особливо корисний, коли мова йде про дизайн з високим рівнем точності:

Adobe Experience Design (Adobe XD) - програма для розробки інтерфейсів від Adobe Systems. Підтримує векторну графіку і веб-верстку, створює невеликі активні прототипи. Має дистрибутиви для macOS так і для Windows.

Як і подібні інструменти, Adobe XD обробляє складні системи дизайну та бібліотеки символів. Він також має добре спроектовані та зрозумілі інструменти для легкого підключення екранів та створення інтерактивних прототипів, які можна використовувати під час тестування користувачів без необхідності введення коду. Нещодавній випуск функції автоматичної анімації робить прототипи багатих взаємодій ще простішими завдяки автоматичній анімації мікро-взаємодій на монтажних областях в Adobe XD.

Figma - крос-платформний онлайн-сервіс для дизайнерів інтерфейсів і веб-розробників. Розробка інтерфейсів може відбуватися як повноцінному нативному додатку так і в онлайн-додатку.

Що стосується інструментів дизайну, то Figma також пропонує кілька функцій, які можна вважати кращими за аналоги. Сітки та обмеження забезпечують повну гнучкість коли мова заходить про створення змінних, адаптивних макетів. Це також можливо зробити в Sketch за допомогою функції зміни розміру групи, але тільки за обмеженням в декілька опцій.

Figma є дуже зручним рішенням особливо для колаборації дизайнера з розробниками, де дві сторони можуть погоджувати зміни в продукті на «льоту». Для роботи з Figma навіть не обов'язково завантажувати додаток на операційну систему, працювати можна прямо з браузера, саме тому і було обрано цей інструмент розробки/проектування UI/UX.

2. ВИМОГИ ДО ІНТЕРФЕЙСУ КОРИСТУВАЧА

2.1 Загальний опис проекту

На сьогоднішній день існує чимало онлайн сервісів для відтворення навчального процесу онлайн. Особливо в зв'язку з епідемічним станом в світі збільшився попит на додатки такого типу. Загалом такі сервіси допомагають частково відтворити, автоматизувати, та покращити процес навчання, навіть якщо передавач і реципієнт інформації не мають безпосереднього зорового контакту та знаходяться на значній відстані один від одного. Такі умови ускладнюють процес передачі інформації (знань).

«Campus» – це веб-додаток, основна мета якого є допомогти студенту та викладачу побудувати якісний процес навчання.

«Campus» представляє собою набір інструментів які мають покращити взаємодію викладача та студента, таким чином полегшити поширення та обмін необхідної інформації, що має позитивно вплинути на хід дистанційного навчального процесу.

Розглянемо внутрішню структуру «Campus»:

Backend частина – це «задня» частина сайту, яку не бачить користувач, але сама вона відповідає за внутрішню роботу додатку. В якості Backend-прошарку Campus використовує мову C# на базі фреймворку .Net Core 3.1

Frontend частина – «лицьова» частина додатку яку безпосередньо бачить користувач, саме з нею він безпосередньо взаємодіє. Frontend-прошарок побудований на основі фреймворку, написаному на мові TypeScript - Angular JS 10.

2.2 Вимоги до дизайну «Campus»

Технічно, існує ISO 9241-210, який описує ергономіку взаємодії людини і системи, визначає користувальницький досвід як сприйняття і реакцію індивіда, що виникають в результаті використання або очікуваного використання продукту, послуги або системи. Але такий стандарт не покритиє в повній мірі всі вимоги, що складуються до сучасного веб-додатку, тому, в результаті аналізу вимог, які присвоюють до сучасних проектів які користують

Цільова аудиторія – студенти та викладачі, бажаючи полегшити процес пошуку поширення та обміну інформації під час навчального процесу (в основному в дистанційному форматі). Студент – соціально активна особа віком від 17 до 23 років, приймає інформацію від викладача. Викладач – особа, яка передає, поширює інформацію до студента, приблизний віковий діапазон 25-80 років.

Загальний перелік робіт:

1. Розробка фірменної символіки та стилістики;
2. Розробка та проектування робочих екранів веб-додатку;
3. Проектування UX.

Характеристики, яким має відповідати веб-додаток:

1. Інформативність – кожен робочий екран містить лише потрібну кінцевому користувачу інформацію та функціонал;
2. Зручність – взаємодія з додатком має бути легкою, всі візуальні «навантаження» під час роботи з додатком мають бути мінімізованими. UX додатку має відповідати як мінімум трьом принципам користувацького досвіду: Принцип Pareto, Закон Міллера, KISS;
3. Легкість – з кожним наступним користуванням певною функціональністю середній час, який витрачає користувач на неї має зменшуватися на як мінімум 20% від часу першої спроби користування цим функціоналом;

Лозунг додатку, який має використовуватися на стартових екрана (екран входу або реєстрації): Learn! Act! Upgrade! (Вивчай! Дій! Вдосконалюйся!)

Головні пункти меню

- Панель приладів (Dashboard) ;
- Календар (Calendar);
- Класи (Classrooms);
- Налаштування (Settings).

Кожна сторінка має складати в собі — шапку, меню, логотип.

Розробка/Проектування UI/UX передбачає в собі як макети для настільних (ПК, ноутбук, моноблок тощо.) так і мобільних пристроїв (смартфон, планшет).

2.3 Вимоги до дизайн-концепції продукту

Дизайн-концепція – сукупність дизайнерських рішень при розробці продукту , яка дає загальне уявлення про нього, модель майбутнього проєкта.

Завдання дизайн-концепції: Визначити напрямлення в якому має діяти продукт, визначити загальну ідею , створити Концепт-карту як результат дослідження напрямлення.

2.3.1 Дослідження і постановка гіпотез:

Початком дослідження будемо вважати позначення основних проблем з якими стикаються користувачі LMS в процесі навчання.

За особистим досвідом та під час спостереження за навчальним процесом протягом дистанційного навчання, було виявлено наступні проблеми:

Із за відсутності безпосереднього контакту з викладачем студент не завжди мав нагоду знати стан зданих робіт (практичних, лабораторних, тестів тощо), оскільки не всі роботи надсилалися і перевірялися через систему Moodle.

- Під час змін в навчальному графіку студент не завжди знав/міг дізнатися про дати поточних подій (перездача, модульний контроль, строки здачі робіт)

- У викладачів не було змоги одразу ж перевірити і отримати результат надісланих студентами лабораторних робіт, які написані на комільованій/інтерпренованій мові програмування без використання сторонніх онлайн сервісів або IDE.

- Студенти не мають єдиного каналу інформування, новин, наголошень тощо щодо навчального процесу. Зазвичай інформація подається через різні джерела і не завжди вчасно потрапляє до студента.

На основі даних проблем виведемо гіпотезу: у викладачів та у студентів під час дистанційного формату навчання існує/постає різка проблема вчасної сумісної синхронізації певної інформації та подій.

2.3.2 Ідея концепту

Оскільки основною проблемою постає некоректний взаємообмін інформацією, сфокусуємося саме на цьому. Вирішивши таку проблему буде простіше планувати події, які можуть не зходитися зі звичним графіком, зручніше підбирати дедлайни для різних видів робіт.

Аби давати користувачу максимальну кількість інформації його діяльності в мінімальній кількості сторінок, в реалізації продукту буде присутньою спеціальна сторінка зі збіркою визначених компонентів, що містять в собі інформацію з інших сторінок, такі як: поточні класи, розклад на день зі сторінки «Calendar», нещодавні оцінки зі сторінки «Classrooms»

Було вирішено зібрати розклад, події, новини, оголошення, дедлайни робіт в один компонент – календар. Таким чином ми звуємо кількість інформаційних

джерел до мінімуму, це означає, що більшість потрібної інформації для студента та викладача відтепер буде знаходитися в одному місці.

Для того щоб користувач не мав потреби переключатися між сервісом з його розкладом та сервісом, що відтворює управління навчального процесу, а саме роботу з управлінням та збереженням завдань та робочих документів зв'язаних з дисципліною було вирішено додати функціонал що буде знаходитися на сторінці «Classrooms» Сутність роботи цієї сторінки полягатиме в наступному: модуль містить в собі абстракції дисциплін що знаходяться в навчальному плані користувача, ці дисципліни містять собі перелік завдань та оцінок, виставлених за їх виконання. Абстракції в «Classrooms» є ресурсом що розділяється між користувачами, є приватним, тобто доступним лише визначеному колу осіб -тільки за запрошенням (окрім адміністраторів).

Також, не менш важливим для викладачів, що викладають дисципліни в які включене вивчення мови програмування буде великою перевагою перевірити результат роботи програми одразу ж додатку, тобто не користуючись іншими засобами компіляції або інтерпретування мови.

2.3.3 Модулі

Щоб коректно описати кожен функціональний/інформаційний частину Camrus доцільно розбити дизайн веб-додатку на модулі.

Модулями будемо вважати частину веб-додатку яка відповідає за певний функціонал, з яким працює користувач.

Розкладемо Camrus на модулі:

- Панель приладів(Dashboard)
- Календар подій(Calendar)
- Збірник Класів (Classrooms)

Розглянемо кожен модуль окремо:

Панель приладів(Dashboard) - цей модуль здебільшого має функцію інформування користувача про найближчі події, новини, оголошення. В ньому має знаходитися тільки важлива та актуальна інформація в скороченому вигляді (не має в собі містити повний перелік подій, розкладу або повну інформацію про Python-блокноти). Цей модуль є стартовою сторінкою , яка зустрічає користувача після авторизації.

Календар подій (Calendar) - модуль який містить в собі перелік всіх подій студента/викладача. Модуль має бути виповнений у вигляді календаря з такими часовими поділками: місяць, неділя, день. Кожна подія має виділятися кольором в залежності від її типу. Приклади подій: Оголошення, Лекція, Практична робота, Лабораторна робота, Модульний контроль, Тестування.

Збірник Класів (Classrooms). Модуль Збірнику має містити в собі так звані класи – абстракція ,що побудована на всій інформації, що стосується дисципліни. Також в даному модулі можна виконувати завдання шлягом запуску коду прямо в браузері або завантажувати файли в виді Notebook-документів. Notebook-документи – це документи згенеровані за допомогою додатку Jupyter Notebook, які містять в собі комп'ютерний код та текстову частину (абзаци, рівняння, посилання, тощо).

Такі документи як R-Notebooks можуть читатися як людиною та машиною. Вони містять в собі опис аналізу , результати , які можуть бути змодельованими в виді таблиць графіків і діаграм а також виконувані документи, які можна запускати, в основному для аналізу даних.

3. UX ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ДИЗАЙНУ ІНТЕРФЕЙСУ КОРИСТУВАЧА

3.1 Побудування User Flow

User flow (користувацький сценарій) - це візуальне представлення послідовності дій, які користувач виконує для досягнення своєї мети. Користувацькі сценарії можуть охоплювати як певну функцію, так і продукт в цілому.

Користувацький сценарій є одним з основних вимог при проектуванні користувацького досвіду. Розробка user flow починається з розуміння потреб користувачів, він допомагає побудувати такий маршрут переходів, де користувач, досягає своєї мети не витрачаючи зайвого часу, уваги та сил. Кожна ітерація, перехід та дія мають спонукати тому щоб увага користувача якомога більше була повернута до користування продукту, аби весь шлях до його мети був у найбільш привабливому і комфортному вигляді.

Спочатку розробимо базову карту переходів.

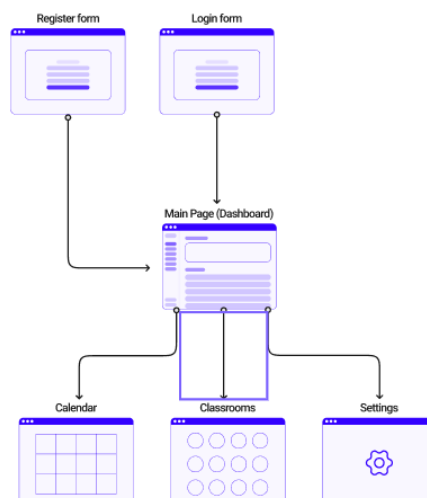


Рисунок 3.1 - Базова карта переходів «Campus»

На ній видні лише прямі переходи по робочим екранам(сторінкам) додатку. На рисунку явно видно, що початком «шляху користувача» з моменту входу є сторінка «Dashboard».

Доповнимо існуючу діаграму кроками та кінцевим функціоналом яким володіє кожен робочий екран, через який проходить користувач з моменту входу або реєстрації(Рис. 3.2)

Оцінимо основні переваги розробки User Flow

- Це швидко - створення User Flow – це один з найшвидших способів візуалізації процесів. Для його створення не потрібно мати додаткових інструментів, збудувати таку діаграму можна прямо в своєму середовищі роботи(додатку)
- Покращена візуалізація процесі – Хоча візуальні прототипи імітують остаточний результат роботи з дизайном, користувацькі сценарії відображають всі можливості користувача одразу на одному рисунку.
- Більше, ніж макети(мокапи) – простий набір мокапів – не найкращий спосіб показати карту процесів користувача. Сам додаток є більше ніж просто набір робочих екранів. User flow допомагає усунути ці пробіли, таким чином це може допомогти уникнути непорозумінь між розробником та дизайнером під час переводу дизайну додатку в програмний код
- Легко для розуміння – наявне зображення блоків з переходами дає легке розуміння майбутніх процесів які мають міститися в додатку.
- Процес перед користувацьким інтерфейсом – одна з самих сильних переваг. Перед тим як почати створення візуальної частини важливо заздалегідь мати бачення продукту, хоча б приблизну структуру екранів, функціоналу та переходів, це значно допоможе проектувати користувацький інтерфейс з мінімальними змінами при імплементації дизайну в додатку.

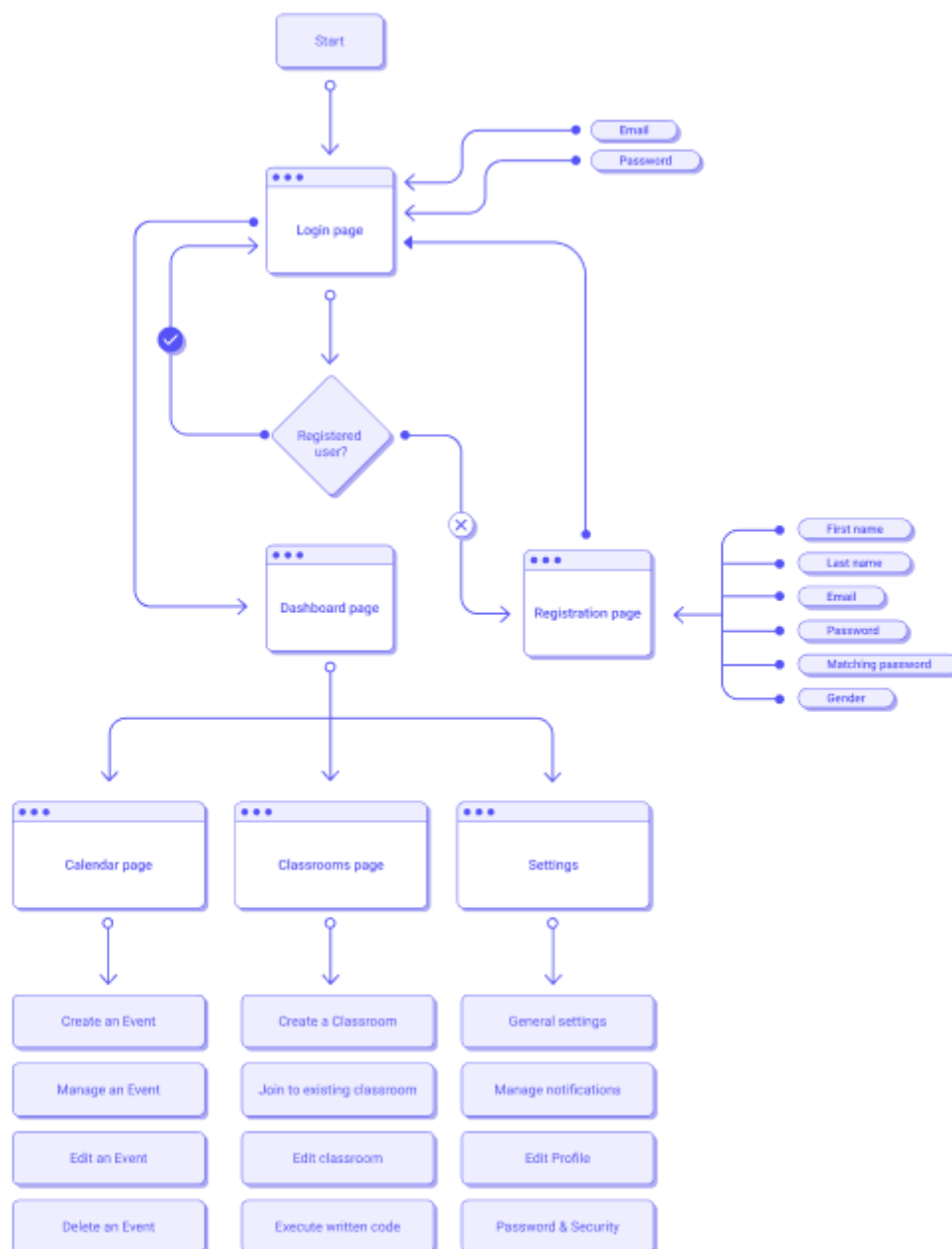


Рисунок 3.2 – Користувацький сценарій «Campus»

3.2 Моделювання Customer Journey Map (Карта подорожі клієнта)

Карты подорожей клієнтів - це інструменти, засновані на дослідженнях, які команди дизайнерів використовують для виявлення типового досвіду споживача з часом та візуалізації багатьох вимірів та факторів, які впливають на нього. Вони

дозволяють брендам дізнатися більше про цільових користувачів. Члени команди вивчають завдання та запитання щодо того, як дизайн відповідає або не відповідає потребам клієнтів за час, за який вони витрачають на взаємодію з товаром чи послугою. Отже, ці карти повинні бути насиченими деталями термінами, що відображають найважливіші кроки та події користувацького шлягу. За допомогою цієї часової шкали ми додаємо уявлення про те, що думають і відчувають клієнти, рухаючись по часовій шкалі.

Щоб створити ефективні візуальні карти, що відображають подорожі клієнтів за цими каналами, карти подорожей повинні бути укорінені в дослідженнях, орієнтованих на дані, і візуально відображати різні фази аналізу досвіду клієнтів на основі різноманітних вимірів, включаючи настрої клієнтів, цілі та точки дотику.

Картографування подорожей - це один із найпотужніших та найпоширеніших інструментів, які призначені для відображення або опису того, як користувачі досягають своїх цілей, схожий прикладом є User Flow.

Правильне розуміння значення відображення подорожі надає такі переваги:

Розуміння картографування процесу-подорожі змушує нас думати про досвід набагато більш структурованим та цілісним чином, визначати послідовність подій, ключові етапи та переходи між ними.

Формування емпатії - коли ви переживаєте досвід клієнтів у форматі історії, ви переживаєте емоції та відчуваєте біль і розчарування користувачів, це допомагає зрозуміти їх мотивацію.

Визначення больових точок - це надзвичайно важливо, оскільки кожен продукт повинен полегшити життя користувачів. Якщо ви розумієте біль, ви маєте змогу покращити досвід.

Структура Customer Journey map:

- Етап подорожі- визначений період подорожі(часовий або поетапний), що включає вибрані області від обізнаності до конверсії та не тільки.
- Сценарії - контекст і послідовність подій, в яких користувач/клієнт повинен досягти мети (наприклад, користувач хоче купити квиток по телефону), від перших дій (розпізнавання проблеми) до останніх дій (наприклад, поновлення передплати).
- Точки дотику - що роблять клієнти під час взаємодії та як вони це роблять.
- Канали - де вони виконують дії (наприклад, Facebook).

Аби правильно побудувати майбутню карту подорожі клієнта будемо використовувати сервіс, який представляє собою гнучкий інструмент візуалізацій користувацького досвіду – UXPressia.

Uxpressia містить в собі готові шаблони карти подорожі клієнта – починаючи від етапів подорожі та закінчуючи діаграмами настою замовника/користувача на певних етапах користування продуктом , а також, окрім картографування подорожі клієнта дозволяє створювати, описувати персон(англ. Personas – людина яка має певні цілі проблеми які має спонукати досягненню/вирішити продукт) та карт впливу.

На основі вищевказаних елементів структури карти подорожі клієнта побудуємо наступну карту для студента:

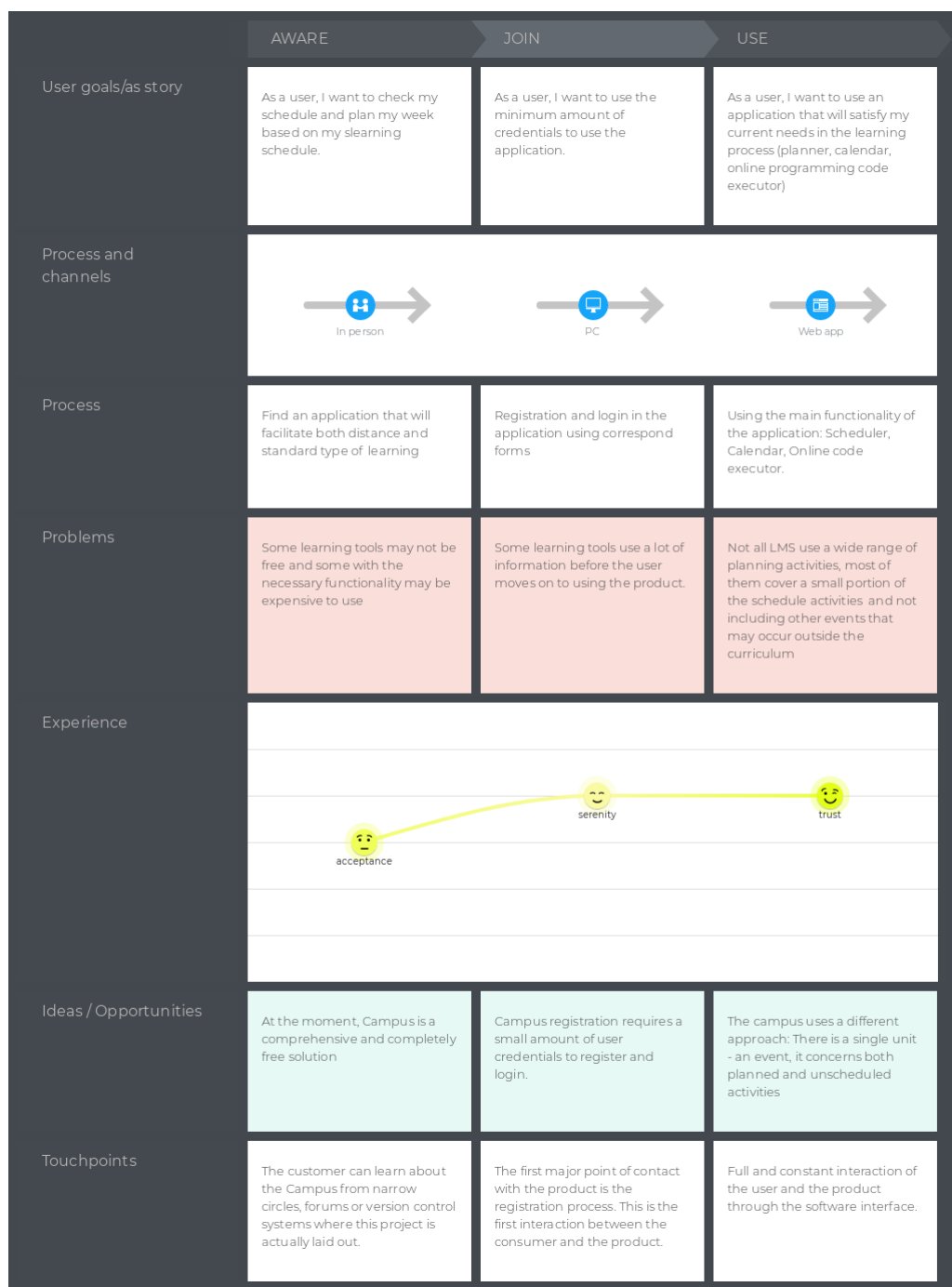


Рисунок 3.3 - Карта подорожі для Студента

Розглянемо основні етапи, зображені на карті: усвідомлення, приєднання, користування(Рис 3.3)

Усвідомлення – етап, коли користувач стикається з якоюсь проблемою та має свідомий намір вирішити її, саме тут починається знайомство з продуктом, про який він може дізнатися з різних соціальних джерел.

Приєднання – процес активного знайомства з продуктом. На цьому етапі користувач починає напряму взаємодіяти з додатком. Як правило фаза прямого знайомства з продуктом починається зі сторінок входу та реєстрації.

Користування – активна взаємодія користувача з додатком. На цьому етапі користується з функціоналом який має вирішити його проблеми які були визначені на етапі усвідомлення.

Тепер розглянемо типового користувача за даними карти подорожі клієнта(студента) – це дасть нам явну картину складових цільової аудиторії додатку

Таблиця 3.1 Картка користувача (Студент)

Ім'я	Григорій
Опис/цілі	Проактивний, енергійний студент, який поважає і цінує свій час. Хочє зробити процес планування свого часу передбаченим ґрунтуючись на навчальному графіку університету, в якому він навчається. Також Григорій хотів би завчасно отримувати інформацію про студентські активності з офіційного джерела в одному місці, при цьому не використовуючи сторонній завантажуваний додаток, лише браузер його пристроїв
Речі, що мотивують	<ul style="list-style-type: none"> - Гарні оцінки - Передбачене планування вільного часу - Прикладати мінімум зусиль, досягати максимум результатів
Речі, що від відштовхують	- Застарілі та повільні інтерфейси користувача

	- додатки, ітерація з якими займає багато часу аби дійти до певного функціоналу
Очікування від додатку	Сподіваюсь, ця програма закрий проблеми зв'язані з несподіваними змінами в навчальному процесі та допоможе якісно побудувати систему взаємодії викладача та студента саме в етапах перевірки та оцінювання робіт через один сервіс, таким чином викладачі зможуть значно зменшити час та зусилля, витрачені на перевірку робіт.
Технології, якими користується	Смартфон – IOS Планшет – IOS Ноутбук/Лептоп – Windows Персональний комп'ютер – MacOS
Браузери, якими користується	Смартфон, Планшет, Ноубук/Лептоп, Персональний комп'ютер - Google Chrome

За цим же підходом будуюмо карту подорожі клієнта для ролі викладача.

Таблиця 3.2 Картка користувача (Викладач)

Ім'я	Ніна
Опис/цілі	Викладачка кафедри комп'ютерних наук. Має досвід роботи викладання різних мов програмування, має стаж роботи понад 20 років. Вважає, що в навчальному процесі потрібно поступово давати навантаження даючи знання студентам, переходячи від простого до складного. Віддає перевагу простоті і мінімалізму у всьому. Знаходиться в пошуку

	інструменту, який одночасно допоможе їй керувати непростим навчальним процесом, а також буде простий в освоєнні.
Речі, що мотивують	- Успіхи студентів - віддача/фідбек від колег з кафедри
Речі, що від відштовхуть	- Складність, неоднозначність, непередбачуваність процесів, які в подальшому тільки гальмують той чи інший аспект викладання
Очікування від додатку	Сподіваюсь, що додаток не буде складним в освоєнні і я зможу якомога швидше повноцінно користуватися ним. Очікую ,що такий інструмент допоможе мені зекономити час на плануванні подій щодо робочого графіку, щоб при будь яких змінах у навчальному графіку студенти одразу ж бачили їх в одному місці, і мені не потрібно було б оголошувати ці зміни окремо через пошту або месенджери, якими користуються студенти.
Технології, якими користується	Смартфон – Android Ноутбук/Лептоп – Windows
Браузери, якими користується	Смартфон, Ноутбук/Лептоп - Google Chrome

3.3 Основні етапи створення дизайну інтерфейсу користувача

Дизайн інтерфейсу програмного продукту має вирішальне значення для успішного запуску продукту. Незалежно від того, яку чудову ідею створила

команда і які першокласні технології використовувала команда розробників програмного забезпечення, продукт не матиме успіху, якщо користувачі розгублені і не можуть зрозуміти, як ним користуватися. Програмний продукт слід створювати для своїх користувачів, а не для команди розробки.

На сьогоднішній день існує безліч ресурсів, які описують те, як має будуватися процес створення інтерфейсу користувача, тому, проаналізувавши кожен з них виділимо лише основні етапи створення інтерфейсу користувача:

1. Дослідження користувачів.

Мета полягає в тому, щоб визначити, хто є споживачами вашого продукту і як вони збираються його використовувати. На цьому етапі нам слід дізнатися:

- Потреби, проблеми та проблеми користувача;
- Типи користувачів, їх досвід, рівень знань та вмінь;
- Які дії вони можуть робити за допомогою вашої системи.

Цей етап необхідний у всіх проектах з розробки програмного забезпечення та у всіх методологіях.. Якщо інформація про всі пункти, наведені вище, суперечлива або неповна, тоді існує багато ризиків, що ваша спроба розробити інтерфейс користувача не буде успішною.

2. Дизайн та прототипування(Каркасні моделі, Прототипи).

Цей процес можна починати з простих ескізів та вайрфреймів з низькою точністю, далі доповнити їх до поступово високої точності.

Каркас - це ескіз побудованої системи. Каркасні моделі повинні чітко пояснювати, які елементи реалізують різні функції на всіх сторінках або екранах майбутнього продукту, але без повної деталізації. Фактичний дизайн екрану буде виготовлений пізніше шляхом посилання на каркас.

Етап каркасного проектування не вимагає багато часу та зусиль, але він може стати в нагоді, коли дизайнер спілкується із замовниками, користувачами, командою проектів, оскільки дозволяє їм легше візуалізувати перші ідеї, коли він просто описує їх усно. Рами можна легко переглянути або змінити. Це допомагає

зробити весь процес більш рентабельним, тому каркаси є ваговою частиною в процесі дизайну

3. Прототипи.

Прототип, у цьому випадку, є напівфункціональним макетом, який може надати точний попередній перегляд фактичної функціональності користувацького інтерфейсу програми або веб-сайту (інтерфейс). Хоча прототип може не мати повноцінних функціональних можливостей, він, як правило, надає клієнтам та / або кінцевим споживачам можливість переходити між елементами інтерфейсу та моделювати, як програма дійсно працюватиме.

Прототипи можуть включати або не включати завершені елементи конструкції. На цьому етапі можна продемонструвати та обговорити логіку роботи компоненти та елементи інтерфейсу користувача . Це допомагає перевірити ідеї та внести деякі зміни на ранніх стадіях проекту.

4. Оцінка створеного дизайну.

Оцінка повинна бути включена на всіх етапах процесу проектування. Його основна мета - оцінити якість дизайну, проаналізувати, як він підходить для користувачів, відповідає їхнім потребам .

Після декількох серій ітерацій, ескізів, каркасів, макетів, прототипів, обговорень, мозкових штурмів із зацікавленими сторонами, кінцевими користувачами та командою проектів, ймовірність створення привабливого інтерфейсу користувача буде значно зростати.

3.4 Реалізація дизайну та прототипування

Оскільки вся інформація про користувача вже зібрана ,одразу переходимо до безпосереднього створення дизайну інтерфейсу користувача

Спочатку, аби окреслити загальну структуру веб додатку скористуємося каркасними моделями – прототипами низької якості. В даному контексті низька якість означає неповну, тестову форму модуля, його шаблон, панелі будь-якої складової інтерфейсу, в якій знаходиться лише основий контент без уклону на візуальну складову.

Прототипи розроблені за такою схемою: Прототип низької якості – Каркасна модель високої якості – Елемент дизайн системи

3.4.1 Модуль «Dashboard»

Dashboard – Представляє собою сторінку яка збирає в собі декілька компонентів, які можуть бути цікавими/корисними для користувача. В основному, Dashboard – збірка інформації яка є первинною для користувача і звільняє його від негайного переходу до інших сторінок сервісу.

Такий вигляд Прототип сторінки «Dashboard» низької якості

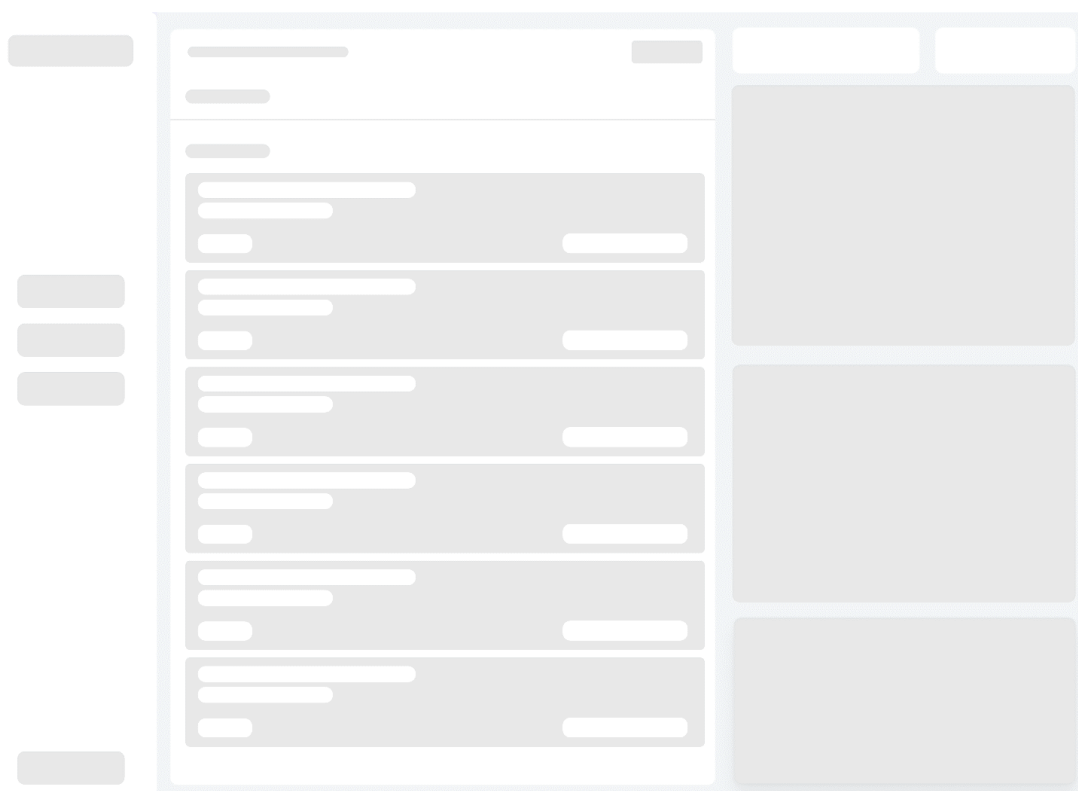


Рисунок 3.4 – Прототип низької якості сторінки «Dashboard»

На Рис. 3.4 віділено основні компоненти, які повинні міститися в кінцевому ПЗ. Окрім цього також розмежено самі компоненти між собою а подібні – згруповані.

Візуально сторінка поділена на 3 частини: головне меню, таблиця нещодавніх подій або класів(classrooms), група додаткових вкладок(розклад, заборгованості, нещодавно виставлені оцінки,)

Наступний крок – розробка каркасної моделі. Така модель має містити в собі більш детальну реалізацію попереднього прототипу.

В більш детальному прототипі слід зазначити що мають містити в собі кожна частина сторінки:

Таблиця 3.2 Визначений вміст модулів «Campus»

Частина сторінки «Dashboard»	Перелік складових частини сторінки
Головна частина(нещодавні події/класи)	<ul style="list-style-type: none"> - Назва предмета; - Короткий опис предмета; - Поточна оцінка; - Викладач предмету; - Перемикач діапазону часу (день, неділя, місяць).
Група додаткових вкладок	<p>Верхній рівень:</p> <ul style="list-style-type: none"> - Пошук дисциплін; - Профіль користувача; - Кнопка створення події/класу. <p>Розклад:</p> <ul style="list-style-type: none"> - Години події;

	<ul style="list-style-type: none"> - Опис події. <p>Заборгованість:</p> <ul style="list-style-type: none"> - Назва дисципліни; - Кількість заборгованостей. <p>Нещодавні оцінки:</p> <ul style="list-style-type: none"> - Назва дисципліни; - Дата(дд/мм/рррр); - Оцінка.
<p>Меню</p>	<ul style="list-style-type: none"> - Логотип; - Пункт «Dashboard»; - Пункт «Calendar»; - Пункт «Classrooms»; - Пункт «Налаштування».

На основі виведених вимог до більш детального прототипу(WireFrame) отримуємо наступний шаблон:

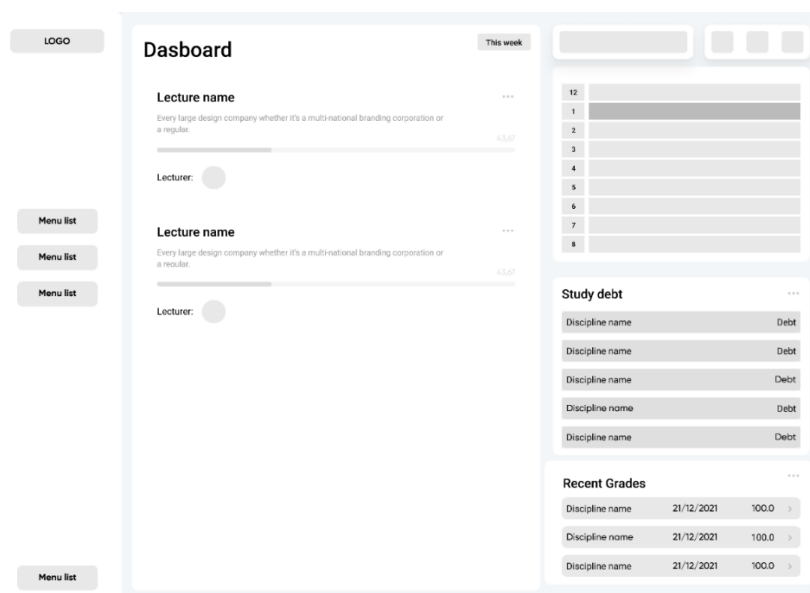


Рисунок 3.5 – Каркасна модель сторінки «Dashboard»

3.4.2 Модуль «Calendar»

Calendar – сторінка, яка містить в собі планувальник у вигляді каледара. Такий календар містить в собі різний набір подій, які входять в розклад студента/викладача.

Вигляд побудованого прототипу низької якості та визначено розположення компонентів , які будуть міститися на сторінці

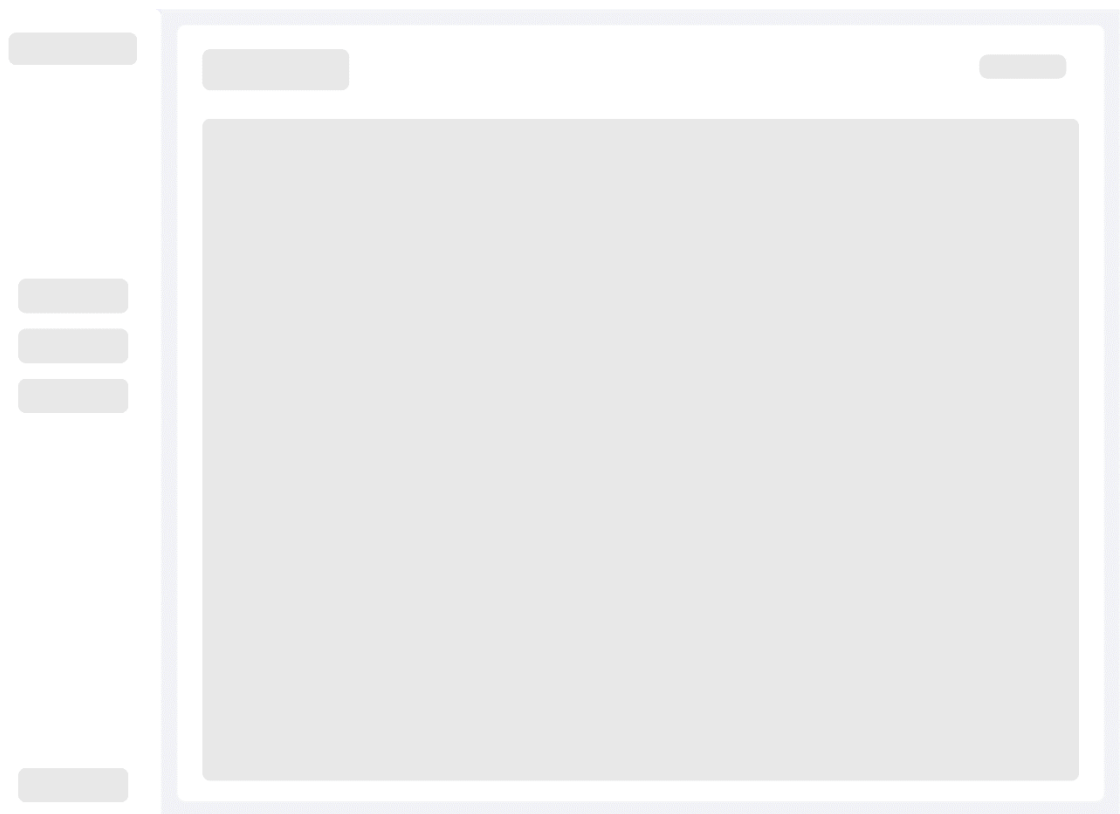


Рисунок 3.6 - Прототип низької якості сторінки «Calendar»

В результаті побудування виділено такі елементи сторінки «Calendar»

- Ліве меню, яке служить для переходу до інших сторінок сервісу
- Центральна частина: містить в собі компонент календарю та логіку створення нової події.

Визначившись з елементами сторінки створюємо початкову каркасну модель (Рис 3.7)

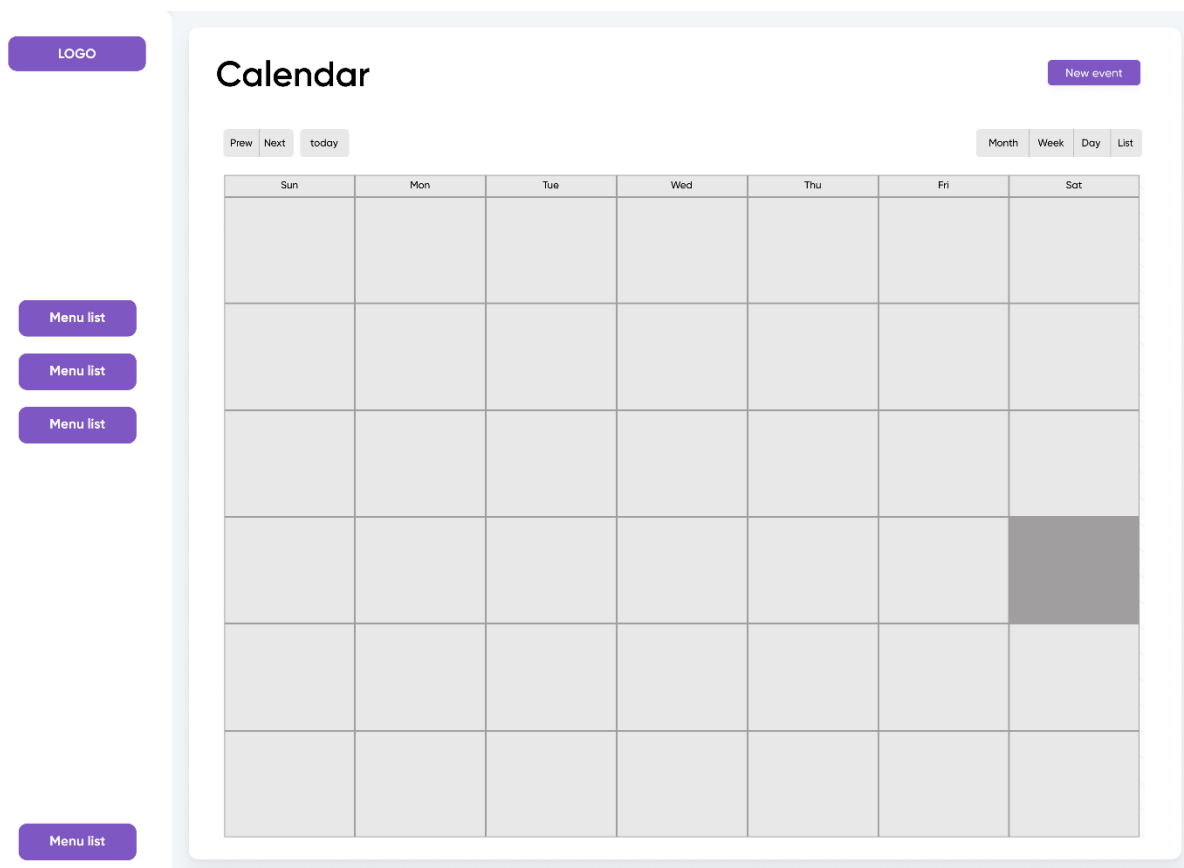


Рисунок 3.7 – Каркасна модель сторінки «Calendar»

Тепер каркасна модель містить головні контролери центрального компоненту каледару (кнопки оперування календарем та зміни його виду), більш детальний вид компоненту «Календар».

Календар буде реалізовуватись на основі проекту відкритого доступу «FullCalendar», тому на основі цього проекту будемо проектувати подальший вигляд сторінки.

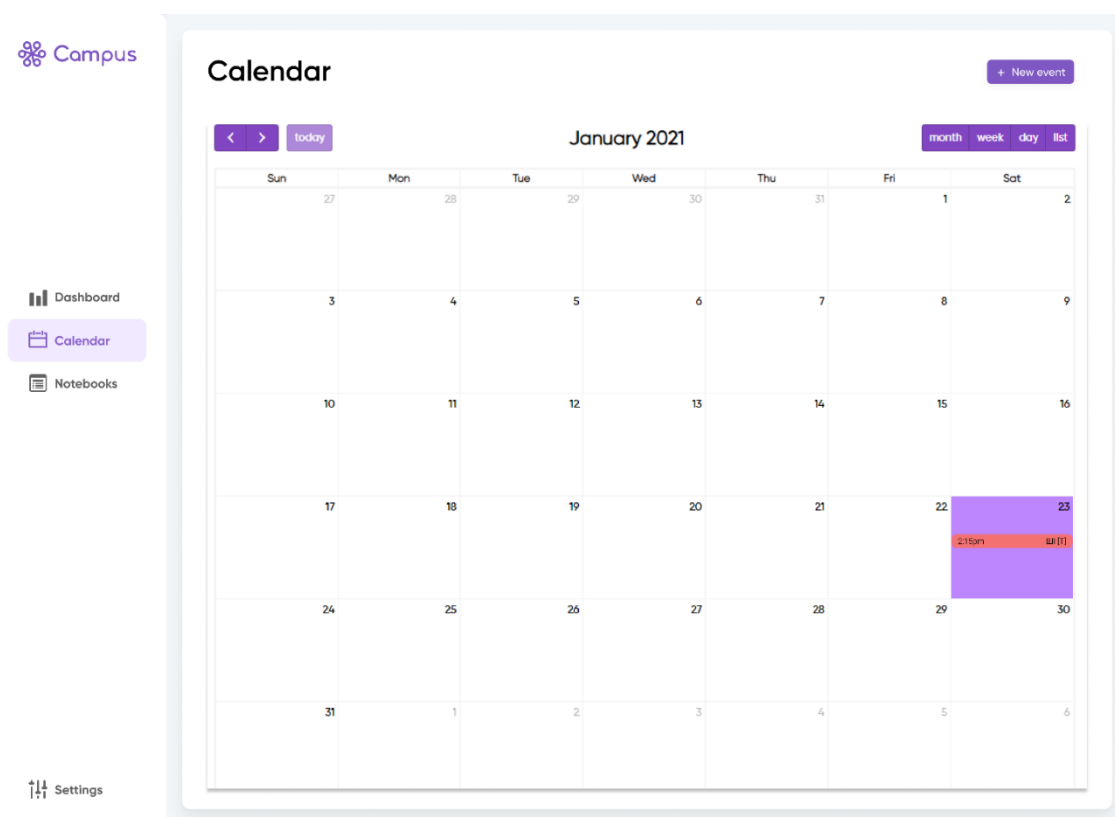


Рисунок 3.8 – Кінцевий прототип сторінки «Calendar»

3.4.3 Модуль «ClassRooms»

Classrooms – сторінка, яка містить в собі функціонал модулю що допомагає викладачу керувати навчальним процесом. Такий модуль містить в собі набір как званих класів – певна дисципліна з власним набором навчальної інформації.

Згідно з визначеним порядком створення прототипів, починаємо з прототипу низької якості, де мають бути відділені основні зони роботи користувача: ліва стрічка меню та модуль «Classrooms».

Оскільки зони роботи користувача сторіки «ClassRooms» та сторінки «Calendar» подібні, є сенс перейти до розгляду наступного виду прототипу – каркасної моделі.

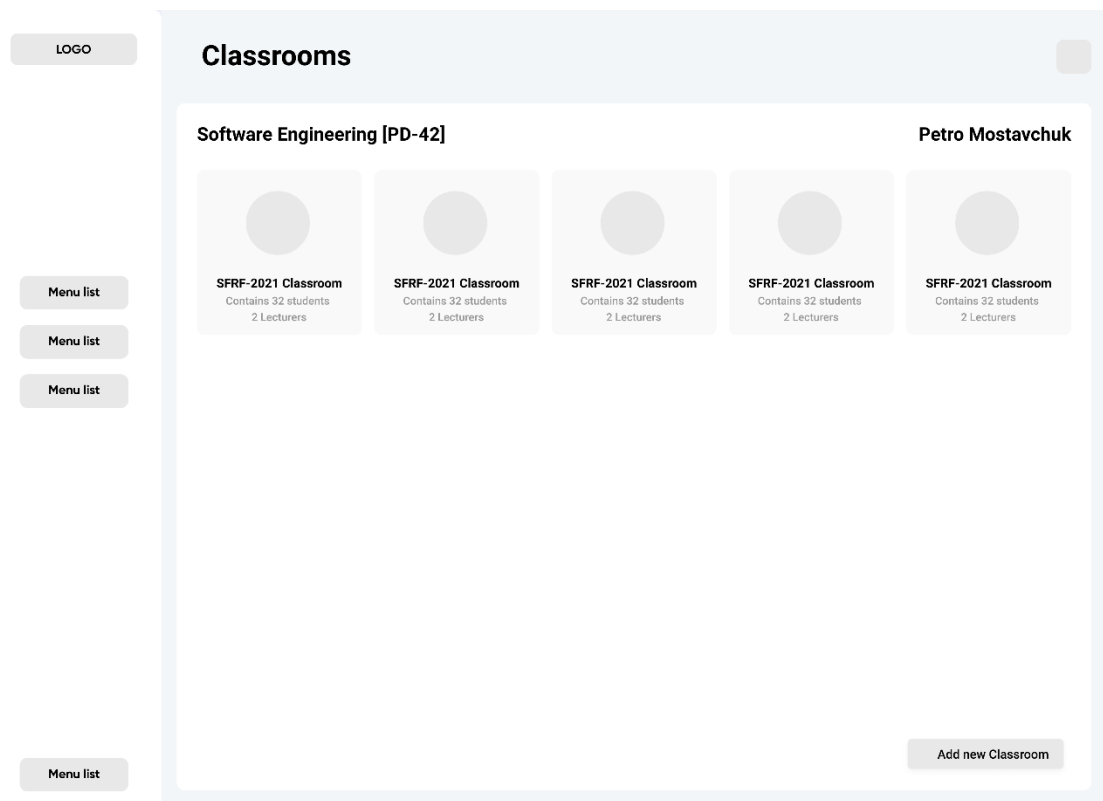


Рисунок 3.9 - Каркасна модель сторіки «Classrooms»

Кожна карта Класу містить в собі:

- Назву класу;
- кількість учасників;
- кількість викладачів;
- зображення дисципліни.

Така каркасна модель орієнтовно виділяє компоненти та їх складові що мають міститися в елементі дизайн системи, єдине , що залишилось – додати фінальні стилі, що притаманні даному додатку

Кінцева варіація дизайну сторінки «Classrooms» буде виглядати наступним чином:

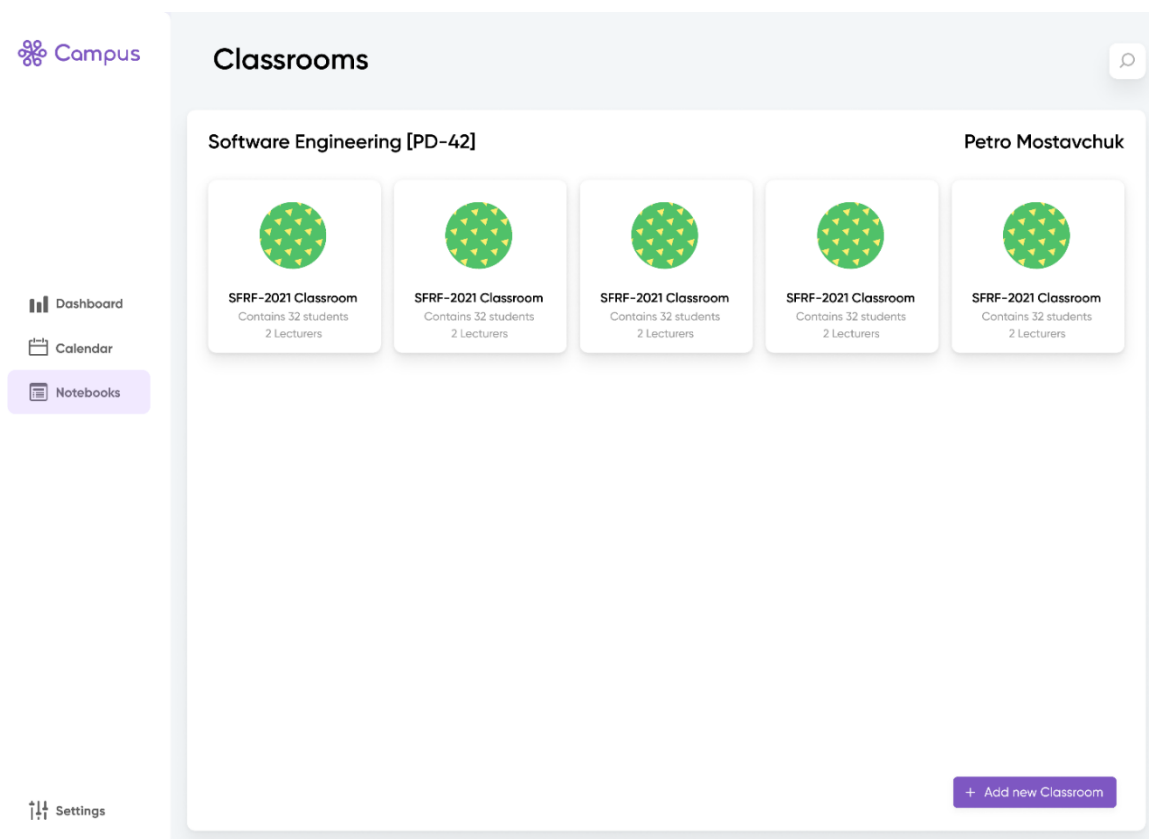


Рисунок 3.10 - Кінцевий прототип сторінки «Classrooms»

Це були основні екрани, з якими має взаємодіяти користувач більшу частину часу користуванням додатку. Для виготовлення подібних шаблонів використовувалась поступова візуалізація компонентів кожного екрану. Такий спосіб проявив себе ефективно, адже вже на перших етапах побудови цей підхід прискорює первинне проектування компонента, ми не тратимо додаткового часу на роботу з дрібними деталями а лише освітлюємо орієнтовні розміри та положення поступово збільшуючи якість до кінцевого вигляду – елементу дизайн системи

4. ТЕСТУВАННЯ ІНТЕРФЕЙСУ КОРИСТУВАЧА

4.1 Основні етапи тестування

Тестування інтерфейсу користувача здійснюється разом з наступними видами тестування (UI):

- Тестування на відповідність стандартам графічних інтерфейсів.
- Тестування з різними дозволами екрану.
- Тестування кросбраузерності або сумісності з різними інтернет браузерами і їх версіями.
- Тестування локалізованих версій: точність перекладу (багатомовність, мультивалютність), перевірка довжини назв елементів інтерфейсу і т.д ..
- Тестування графічного інтерфейсу користувача на цільових пристроях (смартфони, кпп, планшети).

Основні елементи графічного інтерфейсу:

- Вікно (вікно браузера, діалогове вікно, модальне вікно, плаваюче вікно).
- Меню (головне, спливаюче, контекстне, системне).
- Віджети / елементи управління / контролю (акордеон, кнопка, радіо-кнопка, чек-бокс, значок (іконка), список, панель інструментів, дерево, смуга прокрутки, повзунок, рядок стану, зринаючі підказки (підказка) і ін.).
- Вкладка.
- Елементи взаємодії: курсор миші, текстовий курсор, поінтер ("долонька"), курсор перетягування і ін.

Можна виділити етапи тестування зручності використання інтерфейсу користувача:

1. Дослідницьке - проводиться після формулювання вимог і специфікацій до системи, а також після розробки прототипу інтерфейсу. Основна мета на цьому етапі - з'ясувати, чи дозволяє він з достатнім ступенем ефективності вирішувати завдання користувача.

2. Оцінкове - проводиться після розробки низькорівневих вимог детального прототипу призначеного для користувача інтерфейсу. Оціночне тестування поглиблює дослідне і має ту ж мету. На даному етапі вже проводяться кількісні вимірювання характеристик призначеного для користувача інтерфейсу: вимірюються кількість звернень до системи допомоги по відношенню до кількості здійснених операцій, кількість помилкових операцій, час усунення наслідків помилкових операцій і т.п.

3. Валідаційне - проводиться ближче до етапу завершення розробки. На цьому етапі проводиться аналіз відповідності інтерфейсу програмної системи стандартам, що регламентують питання зручності інтерфейсу, проводиться загальне тестування всіх компонентів для користувача інтерфейсу (з точки зору кінцевого користувача). Під компонентами інтерфейсу тут розуміється як його програмна реалізація, так і система допомоги і керівництво користувача. Також на даному етапі перевіряється відсутність дефектів зручності використання інтерфейсу, виявлених на попередніх етапах.

4. Порівняльне - даний вид тестування може проводитися на будь-якому етапі розробки інтерфейсу. В ході порівняльного тестування порівнюються два або більше варіантів реалізації призначеного для користувача інтерфейсу.

4.2 Проведення тестування користувацького інтерфейсу на основі розробленої дизайн-системи

Розглянувши основні види тестування інтерфейсу користувача, звернемо увагу на те, як були проведені ці етапи для «Campus».

4.2.1 Дослідницьке тестування

Після формулювання вимог в складі повної команди розробки було розглянуто перелік рішень стосовно оптимальності та важливості визначених функціональних можливостей, які в подальшому мав би кінцевий веб-додаток та які згодом заносилися до Roadmap сервісу Atlassian - Jira. Тому, після декількох ітерацій розробки з подальшим тестуванням та плануванням після кожної ітерації життєвого циклу додатку Roadmap виглядав наступним чином:

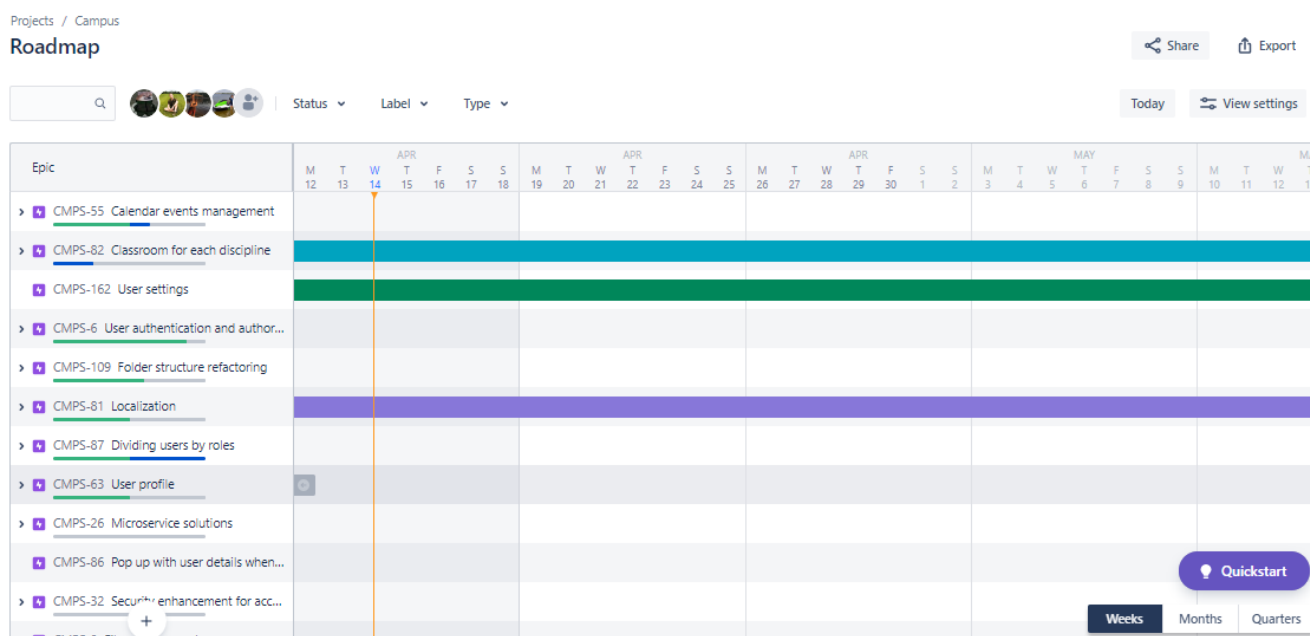


Рисунок 4.1 – «Дорожня карта» Campus

4.2.2 Оцінкове тестування

Оцінкове дослідження полягало у збірці кількісних характеристик програмного інтерфейсу для цього було використано закони Хіка та Фітса

При проектуванні інтерфейсів закон Хіка допомагає визначити оптимальну кількість об'єктів в однорідному масиві, наприклад, в меню. Зазвичай він застосовується в зв'язці з законом Фітса, який допомагає визначити оптимальний з позиції швидкості реакції розмір елемента.

Закон Хіка також тісно пов'язаний з іншими принципами сприйняття і психологічними особливостями прийняття рішень. Його можна однаково ефективно розглядати в контексті теорії близькості і правила 7 ± 2 , а також інших моделей поведінки користувача на сайті.

Поза інтернет-середовища принципи закону Хіка з психології реалізовані практично в будь-якому дизайні інтерфейсу, які взаємодіють з користувачем: починаючи від панелі управління мікрохвильовкою і закінчуючи розташуванням і кількістю кнопок на пульті для телевізора.

Формула Хіка-Хаймана(4.2.1)

Вчені вивели формулу, яка описує логарифмічну залежність між часом реакції і кількістю об'єктів, з яких потрібно вибрати.

$$RT = a + b * \log_2(n + 1) , \quad (4.2.1)$$

де RT - це загальний час реакції,

a - загальний час, який не бере участі у прийнятті рішень,

b - це емпірично отримана константа, заснована на часі когнітивного процесу для кожного варіанту, який для людини становить 0.155 ,

n - кількість рівнозначних альтернативних варіантів, з яких потрібно вибрати.

Оскільки веб-додаток в процесі його проектування як з боку серверної так і з боку візуальної частини поступово зазнавав змін, дизайн система Campus була

поділена за версіями (v.1.0, v.1.1), таким чином, порівнявши кількісні характеристики продукту до і після значних змін можна буде оцінити, наскільки добре було оптимізовано додаток під роботу для кінцевого користувача

Під час тестування для вирахування RT було складено наступну таблицю:

Таблиця 4.2.1 – Порівняння розрахунків версії 1.0 та 1.1 додатку за законом Хіка

	Campus v.1.0 (a,n,RT)			Campus v.1.1 (a,n,RT)		
	a	n	RT	a	n	RT
Сторінка «Dashboard»	3.34	6	3.77	3.11	6	3.54
Сторінка «Tasklist/Calendar»	2.45	4	2.8	2.31	3	2.62
Сторінка Notebooks /Classrooms (головна)	3.1	3	3.42	2.7	3	3.01

При створенні дизайну і проектуванні інтерфейсу закон Фітса допомагає визначити оптимальні розміри і взаємне розташування кнопок, посилань і інших елементів на сторінці в залежності від того, чого саме ми хочемо домогтися від користувача. Також закон дозволяє задати оптимальні для користувача зони натискання для всіх клікабельних елементів інтерфейсу

Для початку наведемо формулювання закону Фітса - загальний закон, що стосується сенсорно-моторних процесів, що зв'язує час руху з точністю руху і з відстанню переміщення: чим далі або точніше виконується рух, тим більше корекції необхідно для його виконання, і відповідно, більше часу потрібно для внесення цієї корекції.

$$T = a + b * \log_2\left(\frac{D}{W} + 1\right) \quad (4.2.2)$$

де T - середній час, що витрачається на вчинення дії,

a - час запуску/зупинки пристрою (дослідницьким шляхом виявлено,

як 56 мс),

b - величина, що залежить від типової швидкості пристрою, (дослідницьким шляхом виявлено, як 155 мс),

D - відстань від точки старту до центру об'єкта,

W - ширина об'єкта, виміряна вздовж осі руху.

Закон Фіттса (Fitts' Law) дозволяє визначити кількісно той факт, що чим далі знаходиться об'єкт від поточної позиції курсора або чим менше розміри цього об'єкта, тим більше часу буде потрібно користувачеві для переміщення до нього курсора.

Складемо наступну таблицю для розрахунку даних ,користуючись формулою закону Фіттса [10]:

Таблиця 4.2.2 – Порівняння розрахунків щодо додатку v1.0 та v1.1 за законом Фіттса.

	Campus v.1.0			Campus v.1.1		
	$D(px)$	$W(px)$	$T(мс)$	$D(px)$	$W(px)$	$T(мс)$
Компонент нещодавних Класів - Сторінка «Dashboard»	510	195	3343.3	604	415	2560.87
Сторінка «Tasklist/Calendar»	596	333	2285.42	682	426	2069.75
Елемент компоненту Класів – Сторінка «Classrooms»	552	345	2069.67	359	212	2770.56

4.2.3 Валідаційне тестування

Розпочнемо валідаційне тестування з переліку систем стандартів графічного інтерфейсу, яких дотримується веб-додаток «Campus» :

- ISO 10741-1995 - курсор;
- ISO 9241-14-1997 - меню;
- ISO 9241-12-1998 регулює візуальне представлення інформації, вікна, списки, таблиці, мітки, поля та ін.;
- стандарт ISO 9241-16-1998 - прями маніпуляції;
- стандарт ISO 12581- (1999-2000) - піктограми.
- ISO/IEC JTC 1/SC 35/WG 2 - Графічний користувальницький інтерфейс та взаємодія
- ISO/IEC JTC 1/SC 35/WG 4 - Інтерфейси користувача для мобільних пристроїв

За системами, що підтримують діалоговий інтерфейс «Campus» є системою з «жорстким сценарієм діалога», тобто мають представлений шаблон сценаріїв діалогів, з якими взаємодіє користувач, адже має такі характеристики:

- меню-діалог, який пропонує користувачеві вибір певних альтернативних функцій обробки з фіксованого перегляду;
- дії запит-відповідь з фіксованим переліком можливих значень, вибраних із списку або відповідей типу Так/Ні (Прийняти/Скасувати);
- запрошення за форматом, який задається за допомогою ключових слів, фраз або за допомогою заповнення екранної форми з регламентованим за складом та структурою набором реквізитів здійснюється підготовка повідомлень.

4.2.4 Порівняльне тестування:

Оскільки під час процесу розробки веб-додаток поступово змінювався під впливом дослідницького тестування, після кожного етапу, було вирішено розділити візуальну частину додатку на версії – v1.0 та v.1.1

Під час проведення оцінкового дослідження окремо взятих модулів додатку були визначені наступні зміни в метриках:

Різниця визначеного середнього часу реакції за законом Хіка(порівняння між Campus v.1.0 та v.1.1):

- Сторінка «Dashboard»: -6.1%
- Сторінка «Tasklist/Calendar»:-6.4%
- Сторінка (головна) Notebooks/Classrooms: -12%

Різниця визначеного середнього часу для вчинення дії за законом Фітса

- Компонент нещодавніх Класів - Сторінка «Dashboard»: -25 %
- Сторінка «Tasklist/Calendar»: -5.5%
- Елемент компоненту Класів – Сторінка «Classrooms»: +3%

Порівнявши організацію, оформлення та розміщення компонентів що належать модулям, в деяких місцях було оптимізовано на рівні часу, затраченого для виконання будь-якої дії, та середнього часу реакції, а в цілому – на переході до певного функціоналу, який пропонує веб-додаток.

Хоча і зміни в кількісному аналізі інтерфейсу користувача були мінімальними, в більшій мірі змінився сам зовнішній вигляд сторінок. Рис. 4.2 ілюструє вигляд компонентів сторінки, які також брали участь у вимірюваннях

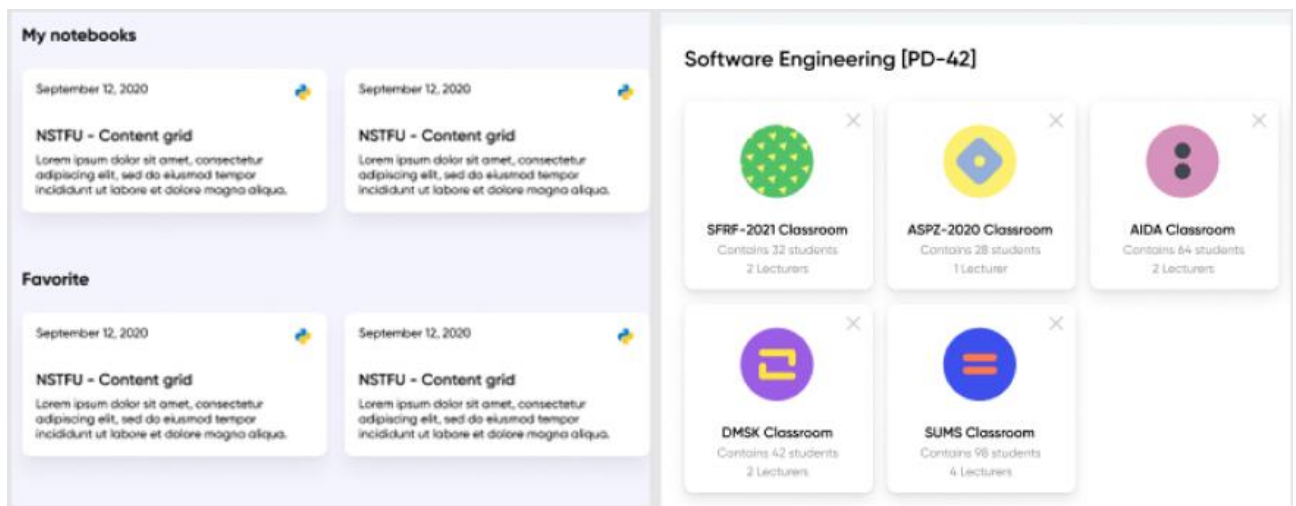


Рисунок (4.2) Вигляд компонентів сторінки «Classrooms» до та після оновлення

5. РЕАЛІЗАЦІЯ ІНСТРУМЕНТУ КІЛЬКІСНОГО АНАЛІЗУ ІНТЕРФЕЙСУ КОРИСТУВАЧА

5.1 Опис використаних технологій

В якості бази для інструмента аналізу було використано стек технологій, що складається з таких мов програмування, як Java та R. Java в свою чергу буде відповідати за основну логіку та буде представляти програмний інтерфейс для зчитування дій користувача, а також виведення даних в окремий файл для подальшого аналізу.

R буде служити для збору та виведення статистики, основаної на зібраних даних додатку. Результатом обробки зчитаних даних буде діаграма з зображеними даними.

Java – це строго типізована, об'єктно-орієнтована мова програмування загального призначення, на основі класів. Це обчислювальна платформа для розробки додатків. Java швидка, безпечна та надійна. Вона широко використовується для розробки додатків на центрах обробки даних, ігрових приставках, наукових суперкомп'ютерах, стільникових телефонах тощо.

Для створення інтерфейсу користувача інструменту буде використовуватися бібліотека для Java-платформи – Swing.

Swing - це інструментарій графічного інтерфейсу користувача (GUI), що включає компоненти графічного інтерфейсу. Swing надає багатий набір віджетів та пакетів для створення витончених компонентів графічного інтерфейсу для програм Java. Swing є частиною Java Foundation Classes (JFC), що є API для програм Java, що забезпечують графічний інтерфейс [12].

R - мова програмування та середовище вільного програмного забезпечення для статистичних обчислень та графіки, що підтримується Фондом статистичних обчислень R. Мова R широко використовується серед статистиків та майнерів

даних для розробки статистичного програмного забезпечення та аналізу даних. Опитування, обстеження даних та дослідження баз даних наукової літератури демонструють значне зростання популярності, станом на квітень 2021 р. R посідає 16 місце в індексі ТЮВЕ, показник популярності мов програмування.

5.2 Вид обчислень та схема роботи інструменту

Даний інструмент відтворює вимірювання часу переміщення між цілями для створення діапазону відстаней за законом Фітса. Окрім часу, закон Фітса може обчислювати Індекс Складності(5.2.1), для цього використовується формула Шенона:

$$ID = \log_2 \left(\frac{D}{W} + 1 \right) \quad (5.2.1)$$

Також в результаті таких обчислень ми можемо визначити так звану «Пропускню здатність» - Середня швидкість інформації(5.2.2), що генерується в результаті серії рухів, є середньою інформацією, поділеною на час за рух, що обчислюється за формулою:

$$TP = \frac{1}{y} \sum_{i=1}^y \left(\frac{1}{x} \sum_{j=1}^x \frac{IDe_{ij}}{MT_{ij}} \right) \quad (5.2.2)$$

Або, опираючись на дані, що обчислюються за законом Фітса отримаємо наступну формулу:

$$TP = \frac{ID}{MT} \quad (5.2.3)$$

Де ID – Індекс складності

MT – час, що витрачається на вчинення дії

За допомогою цих обчислень і зібраних даних можна буде складати графіки, що будуть наявно виводити пропускню здатність та індекс складності.

Основне джерело збору даних, які будуть використовуватися для виведення графіків буде Java-додаток, він в свою чергу збирає ширину, та дистанцію від

точки старту (поточне положення курсора миші на осі X,Y) до об'єкта. Після цього всі дані записуються до файлу з розширенням «.log».

Далі, залоговані дані переходять до обробки R-скриптом, який в свою чергу обробляє, проводить розрахунки та генерує графік з ними у форматі «.png».

Загальна схема роботи інструменту виглядає наступним чином:

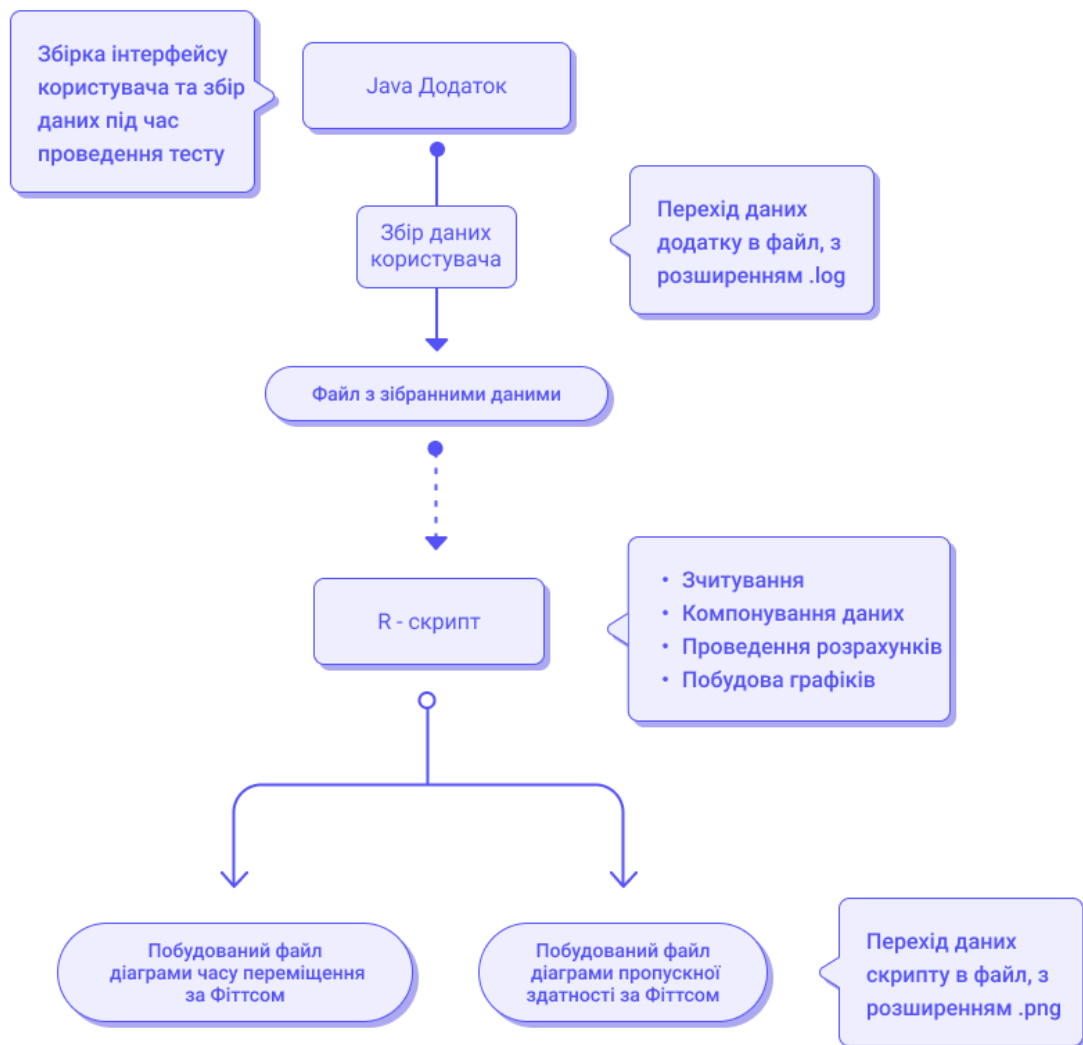


Рисунок 5.1 – Схема роботи інструменту

5.3 Процес збору даних та обчислень

Точкою входу до Java програми є метод `main()` – метод що ініціалізує роботу всього додатку. Після цього будується графічний програмний інтерфейс, створений за допомогою бібліотеки `Swing`, в якому користувачу пропонується пройти тест, для зняття метриків, що знадобляться в подальшому для розрахунків. Система зняття метриків пропонує невелику серію перевірок реакції, на основі якої перевіряється дистанція між об'єктами, що в подальшому служить одним з параметрів розрахунку часу переміщення між цілями та розрахунку індексу складності для кожного запису (Рис. 5.2).

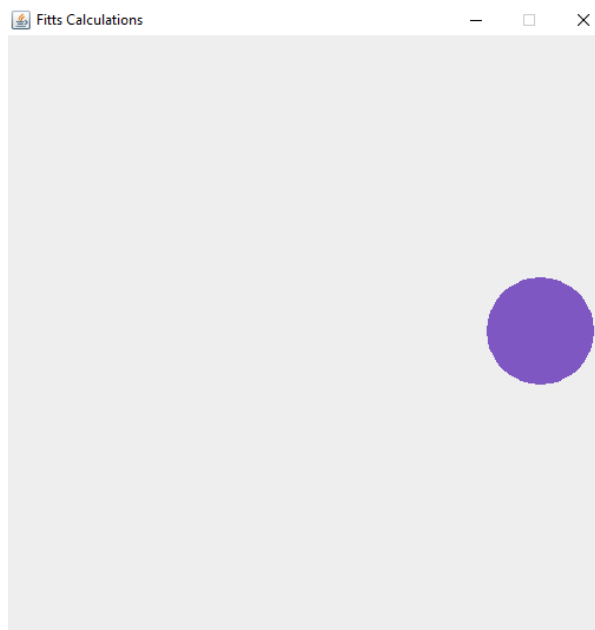


Рис. 5.2 – Вигляд інструменту для збору даних

В якості сховища первинних даних виступає `.log` файл, в якому містяться записи, що відповідають шаблону – `<розмір_цілі> :<дистанція>: <інтервал_між_кліком>`. Таким чином відбувається логування дій користувача.

Такі дані проходять первинну обробку і на основі їх `R`-скрипт спочатку створює запис в пам'яті для кожної зчитуваної одиниці даних. Після цього

розраховуються індекс складності та пропускна здатність. Після таких розрахунків будуються графіки.

Аби побудувати графік, зв'язати дані на ньому або виставити дані в зручному вигляді, не потрібно писати свою реалізацію, такий функціонал все наявний в мові R [14]:

- *plot* (): використовується для створення лінійної діаграми У найпростішій функції *plot* () просто будується два вектори один проти одного.

- *abline* (): функція, що вимальовує пряму, що відповідає закономірності, зображеній на графік

- *aggregate* (): дозволяє розділяти об'єкт на підмножини, обчислювати сумарну статистику для кожного, і повертати результат у зручній формі.

- *lm* (): функцію для створення моделі регресії, заданої деякою формулою, у вигляді $Y \sim X + X^2$.

- *summary* (): загальна функція, яка використовується для отримання підсумків результатів різних функцій підбору моделі

Наприклад, графік залежності інтервалу між кліком та Індексом Складності (TP~ID) виглядає так:

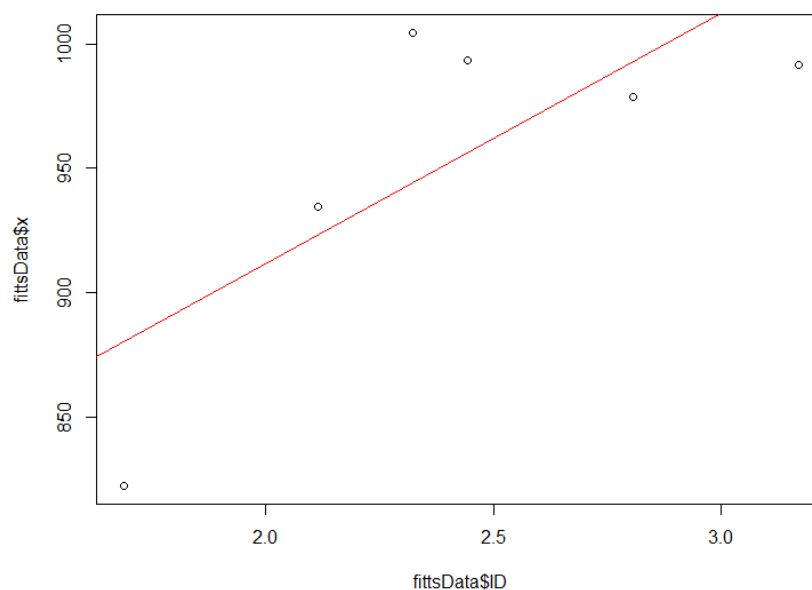


Рисунок 5.2 – Графік МТ~ІД

Таким же чином будується графік залежності пропускної здатності та Індексу складності (TP~ID)(Рис.5.2).

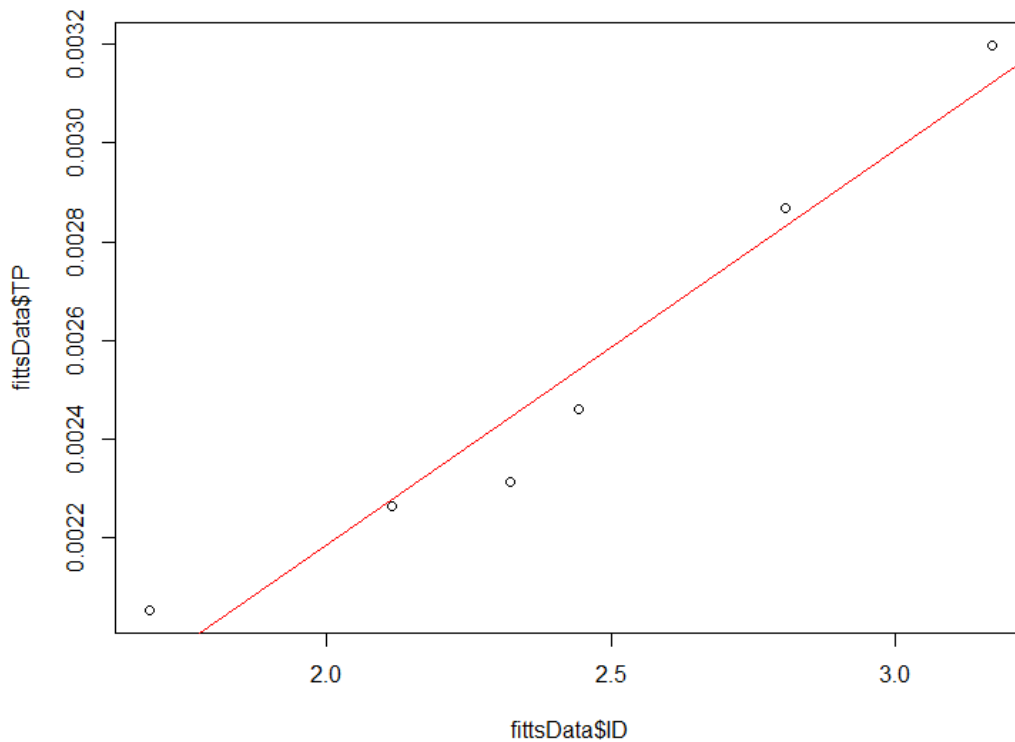


Рисунок 5.2 – Графік TP~ID

Файли з розширенням png зберігаються на робочу директорію проекту у розмірі 480x480 px.

Даний додаток є прикладом, на основі якого можна створювати інструменти аналізу, як інтерфейсу користувача, так і його вподобання (наприклад, створення інструменту що аналізує «гарячі точки» модулів). Даний технічний стек було обрано швидкого та надійного процесу збирання, зберігання, запису інформації (Java) для мінімізації реалізацій функціоналу розробником для обрахунку та графічного виведення аналітичних даних (R).

ВИСНОВКИ

У ході роботи було розглянуто та реалізовано підходи, щодо проектування користувацького інтерфейсу. Результатом роботи стала дизайн система – набір графічних прототипів, шаблонів гайдлайнів , які допоможуть Front-end розробнику втілити дизайн-концепт в Front-end частину додатку. Також такий набір буде корисним при розширенні функціоналу додатку, адже дизайн основних компонентів вже готовий, єдине, що залишається - додати контент.

Розробка таких систем також передбачає спостереження за поведінкою користувача на етапах експлуатації. Також при таких умовах проектувальник дизайн системи має бути готовим до постійних змін, адже неможливо одразу зрозуміти всі вподобання користувача, тому елементи дизайн системи корегуються у відповідності до якості ітерації кінцевого користувача стосовно існуючого вигляду компоненту

Невід’ємною частиною розробки став аналіз поточних версій додатку – аби впевнитися , що наступні генерації дизайн системи дійсно забезпечують ефективну роботу користувачеві з веб-додатком. Такий підхід допоміг дізнатися проблеми кожній версії в порівнянні один з одним, та визначив напрямлення, яким потрібно було приділити більше уваги в контексті наповнення шаблонів компонентів, з якими має ітерувати кінцевий користувач.

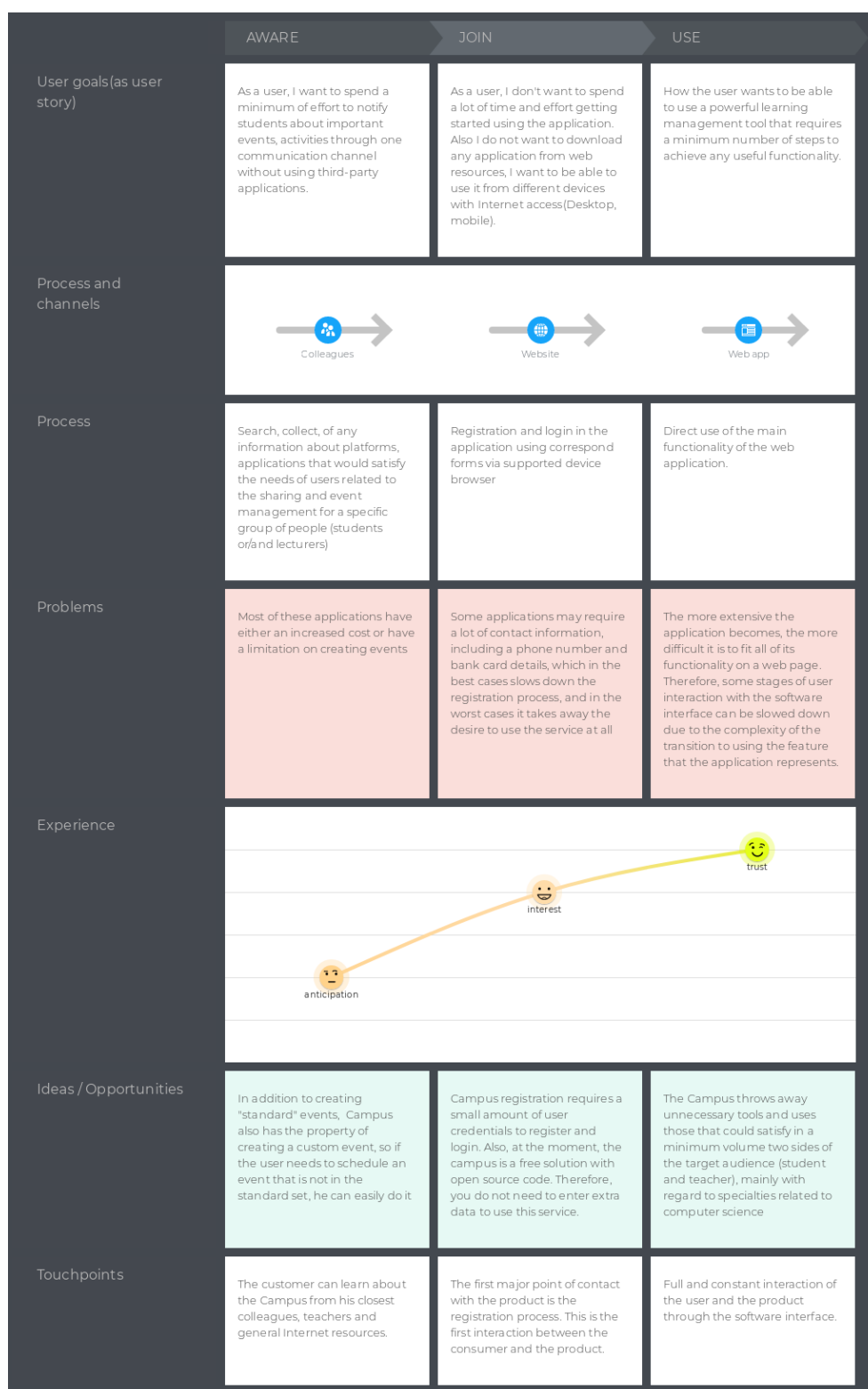
Дипломна робота має практичну цінність, що полягає тому, що кожен бажаючий зможе спроектувати справжню дизайн систему, що має задовольняти потреби користувача, користуючись планом, вказаним в основній частині. Також UI/UX проектувальник підвищить свої знання в області оцінювання готового дизайн продукту та аналізу проблемних точок, що можуть ускладнювати взаємодію програмний інтерфейс – людина.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Vishal Unjiya/ Fundamental Concepts of UI/UX To Provide User-friendly Experience [Електронний ресурс] - 2020 – Режим доступу до ресурсу :[://yourstory.com/mystory/best-ui-ux-design-concepts](https://yourstory.com/mystory/best-ui-ux-design-concepts)
2. Sharpened productions / The User Interface -2021 - [Електронний ресурс] – Режим доступу: https://techterms.com/definition/user_interface#:~:text=A%20user%20interface%2C%20also%20called,a%20natural%20and%20intuitive%20way.
3. Impressum, UserTesting / UI vs. UX: What’s the difference between user interface and user experience? 2021 - [Електронний ресурс] – Режим доступу до ресурсу: <https://www.usertesting.com/blog/ui-vs-ux>
4. Impressum, UserTesting / UX design , user experience design experts weigh in 2019 - [Електронний ресурс] – Режим доступу до ресурсу: <https://www.usertesting.com/blog/what-is-ux-design-15-user-experience-experts-weigh-in>
5. Resmini, Andrea, Luca Rosati/ Pervasive information Architecture: Designing Cross-Channel Experiences. Burlington, 2011 – 335 с. - (Morgan Kaufmann Publishers).
6. Jon Yablonski/ Laws of UX: Using Psychology to Design Better Products & Services, 2020 – 270 с. - (O'Reilly Media, Inc.)
7. Jon Yablonski/ Collection of best practices that designers can consider when building user interfaces – 2021 -[Електронний ресурс] – Режим доступу до ресурсу : <https://lawsofux.com>
8. George A. Miller/ The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, Psychological Review 63, no. 2 – 1996 -: pp.81–97.
9. Jakob Nielsen / Card Sorting: Pushing Users Beyond Terminology Matches -2009 - [Електронний ресурс], <https://www.nngroup.com/articles/card-sortingterminology-matches.>

10. Heiko Drewes/ A Lecture on Fitts' Law – 2013 -[Электронный ресурс] – Режим доступа до ресурсу : <https://www.cip.ifi.lmu.de/~drewes/science/fitts/A%20Lecture%20on%20Fitts%20Law.pdf>
11. Herbert Schildt / Java The Complete Reference, 8th Edition– 2011 - [Электронный ресурс] – Режим доступа до ресурсу : <https://srikarthiks.files.wordpress.com/2016/07/java-the-complete-reference-8th-edition.pdf>
12. Kishori Sharan/Beginning Java 8 APIs, Extensions and Libraries: Swing, JavaFX, JavaScript, JDBC and Network Programming APIs, 2018 – pp. 85-204 (Apress)
13. Robert Martin / Clean Code: A Handbook of Agile Software Craftsmanship 1st Edition, 2010 – 1168 с.
14. The R Foundation / R language Documentation – 2021 -[Электронный ресурс] – Режим доступа до ресурсу : <https://www.r-project.org/other-docs.html>
15. Masaaki Kurosu / International Conference on Human-Computer Interaction: HCI 2015: Human-Computer Interaction: Interaction Technologies, 2015 – pp. 238-249, (SPRINGER)
16. Journal Dev / R programming – 2021 -[Электронный ресурс] – Режим доступа до ресурсу : <https://www.journaldev.com/r-programming>
17. Bernard J. Jansen/ The Graphical User Interface: An Introduction – 2008 -[Электронный ресурс] – Режим доступа до ресурсу : <https://faculty.ist.psu.edu/jjansen/academic/pubs/chi.html>
18. ISO/IEC JTC 1/ STANDARDS BY ISO/IEC JTC 1/SC 35 - [Электронный ресурс] - 2020 – Режим доступа до ресурсу : <https://www.iso.org/committee/45382/x/catalogue/>

ДОДАТКИ



Додаток «А» Карта подорожі Викладача

ДИПЛОМНА РОБОТА

«Створення UI/UX для learning-додатку
«Campus» для підвищення ефективності
взаємодії з кінцевим користувачем»

Виконав
студент
Конішевський В.І.
Керівник:
Коба А.Б.

Київ 2021

Основні відомості роботи

- **Мета роботи** – розробка дизайн системи для learning-додатку «Campus» Покращення якості взаємодії між програмним інтерфейсом та кінцевим користувачем.
- **Об'єкт дослідження** – Дистанційне навчання та централізована система організації учбового процесу студентів вищих навчальних закладів інженерних спеціальностей та слухачів зовнішніх курсів для підготовки майбутніх ІТ спеціалістів.
- **Предмет дослідження** – візуальна частина(UI/UX) веб застосунку для обміну та аналізу інформації стосовно навчального графіку, організації управління завданнями/сторонніми подіями студентів освітніх установ

Актуальність проблеми

На сьогоднішній день як ніколи актуальний формат саме дистанційного навчання, де одними з головних каналів обміну інформацією виступають системи конференц зв'язку та системи управління навчанням.

Оскільки практика проведення конференц дзвінків поширюється корпоративний навчальний сектор вільного використання, такі системи часто оновлюють, причому не тільки технічно але і візуально. На відміну від систем зв'язку, системи управління навчанням обмежені, в основному, на навчальний сектор, тобто такий прошарок ПЗ не передбачує абсолютно вільного використання та розробляється під окремі вимоги навчальних закладів за стандартами і набором технологій, які як правило тривалий період не зазнають ніяких змін.

Те ж саме стосується візуальної частини таких систем - з часом актуальність зовнішнього вигляду та зручності таких додатків знижується, а оскільки зміни в системі залишаються тривіальними постає питання зручності взаємодії з такими системами, особливо під час дистанційного навчання, коли користувач користується такими додатки більшу частину навчального часу.

User Interface (користувацький інтерфейс)

Інтерфейс користувача, який також називається "UI" - це засіб, за допомогою якого людина керує програмним додатком або апаратним пристроєм. Хороший користувацький інтерфейс забезпечує "зручний" досвід, дозволяючи користувачеві взаємодіяти з програмним або апаратним забезпеченням природним і зрозумілим способом.

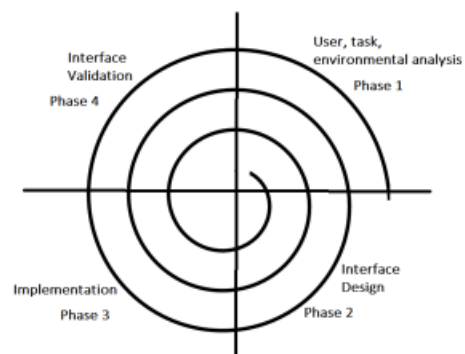


Рисунок 1 – Процес розробки користувацького інтерфейсу

User Experience(користувацький досвід)

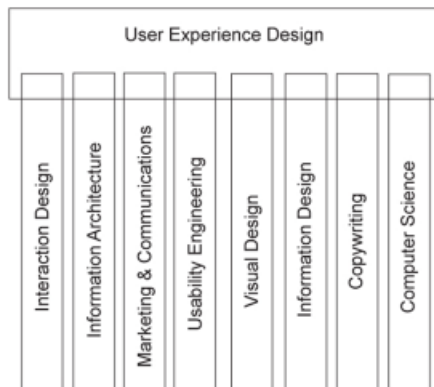


Рисунок 2 – складові проектування користувацького досвіду

Користувальницький досвід (UX) - це те, що користувач конкретного продукту відчуває при використанні цього продукту. Таким чином, завдання UX-дизайнера полягає в створенні продукту, що забезпечує найкращий користувацький досвід.

Transitions basemap (Базова карта переходів)

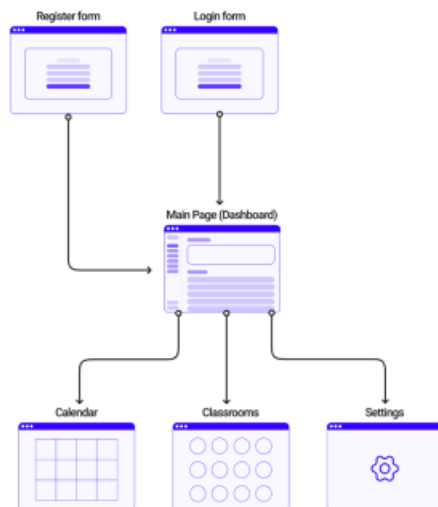


Рисунок 3 – Базова карта переходів додатку

Базова карта переходів представляє собою відображення основних модулів, на які може переходити користувач

Learning-додаток «Campus» містить в собі 4 головних модулі + сторінки реєстрації/авторизації:

- Dashboard (панель приладів)
- Calendar (Модуль Календару)
- Classrooms (Модуль з навчальними класами)
- Settings (Модуль налаштувань)

Користувацькі сценарії

User flow (користувацький сценарій) - це візуальне представлення послідовності дій, які користувач виконує для досягнення своєї мети. Користувацькі сценарії можуть охоплювати як певну функцію, так і продукт в цілому.

Дані користувацькі сценарії охоплюють процеси, через які проходить користувач від реєстрації до користування функціоналом додатку.

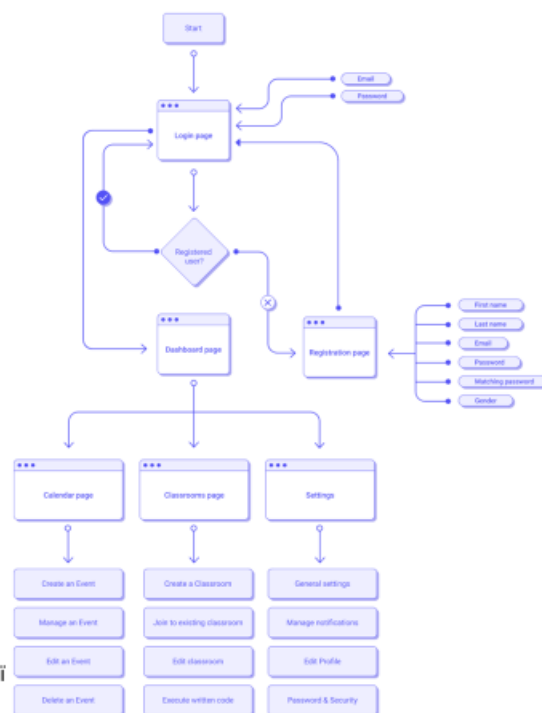


Рисунок 4 – Користувацькі сценарії додатку «Campus»

Реалізація карт подорожей (викладач)

Окрім етапів користувацького шляху карта містить в собі набір процесів, цілей, емоцій які з'являються під час проходження цих етапів. Для додатку було визначено наступні характеристики:

1. Цілі користувача – цілі, написані в форматі користувацьких сценаріїв, яких хоче досягнути користувач, та труднощі з якими, на його думку він може зіткнутися.
2. Канали – засоби, оточення, та інші процеси, з якими взаємодіє користувач для отримання інформації
3. Процес – описання набору дій, які притаманні під час проходження етапу

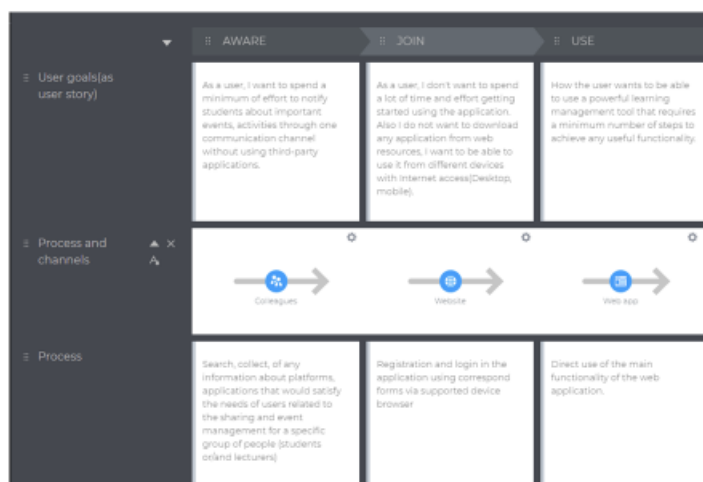


Рисунок 5 – Користувацькі сценарії додатку «Campus» для викладача (частина 1)

Реалізація карт подорожей (студент)

4. Досвід – діаграма, що зображує емоційний стан користувача за час проходження кожного етапу (за час набуття досвіду роботи з додатком)
5. Ідеї/можливості – перелік особливостей продукту, що позитивно впливають на користувацький досвід
6. Точки дотику - що роблять клієнти під час взаємодії та як вони це роблять



Рисунок 6 – Користувацькі сценарії додатку «Campus» для студента (частина 2)

Побудування шаблонів інтерфейсу користувача

Спочатку, аби окреслити загальну структуру веб додатку скористуємося прототипами низької якості. В даному контексті низька якість означає неповну, тестову форму модуля, його шаблон, панелі будь-якої складової інтерфейсу, в якій знаходиться лише основний контент без уклону на візуальну складову.

Було вирішено розробляти прототипи за такою схемою:

1. Прототип низької якості
2. Каркасна модель високої якості
3. Елемент дизайн системи



Рисунок 7 – Схема розробки прототипів

Прототип низької якості

Такий вид прототипу розробляється на початковому етапі розробки шаблонів інтерфейсу користувача. Він задає початковий вид сторінки, розмежує всі її елементи на групи та окреслює приблизні розміри і відстані між потенційними елементами системи

На рисунку зображено прототип сторінки «Dashboard», в цьому випадку сторінка поділена на 3 частини: головне меню, таблиця нещодавніх подій або класів(classrooms), група додаткових вкладок(розклад, заборгованості, нещодавно виставлені оцінки,)

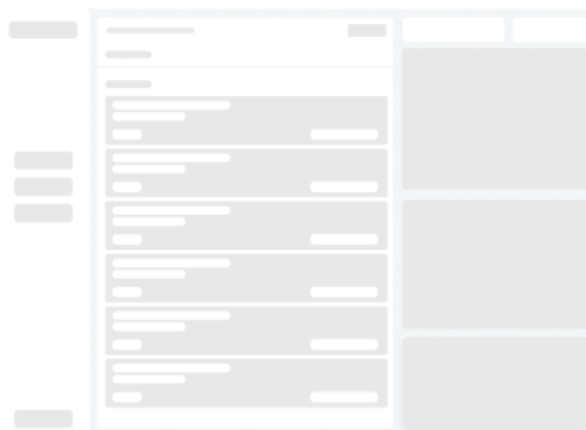


Рисунок 8 – Прототип низької якості сторінки «Dashboard»

Каркасна модель (Wireframe)

З самого початку проектування будь якого прототипу потрібно визначити загальні за якими буде побудований майбутній елемент дизайн системи

Для кожного шаблону learning-додатку «Campus» визначений такий перелік правил що мають в собі ціль підвищення якості взаємодії додатку з кінцевим користувачем:

Майбутній елемент дизайн-системи має відповідати таким принципам та законам:

1. Закон Міллера –кожен компонент має містити елементи в кількості 7 ± 2 (4 ± 2)
2. Принцип простоти (Бритва Оккама) – якщо є декілька відповідей, чи рішень, правильний варіант, де користувач задіює мінімум зусиль та часу
3. Закон Хіка – відношення розмірів та відстаней між елементами має спонукати користувача витратити менше зусиль для пошуку потрібного йому функціоналу (Визначений інтервал: 3-5 секунд)

Каркасна модель високої якості (Wireframe)

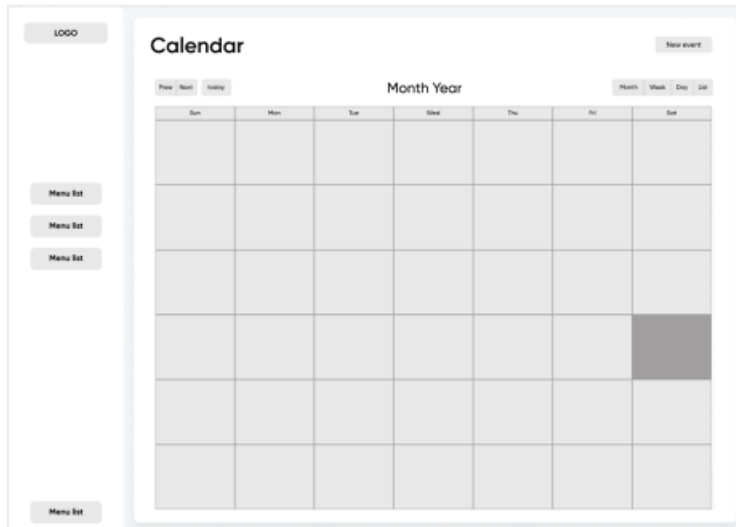
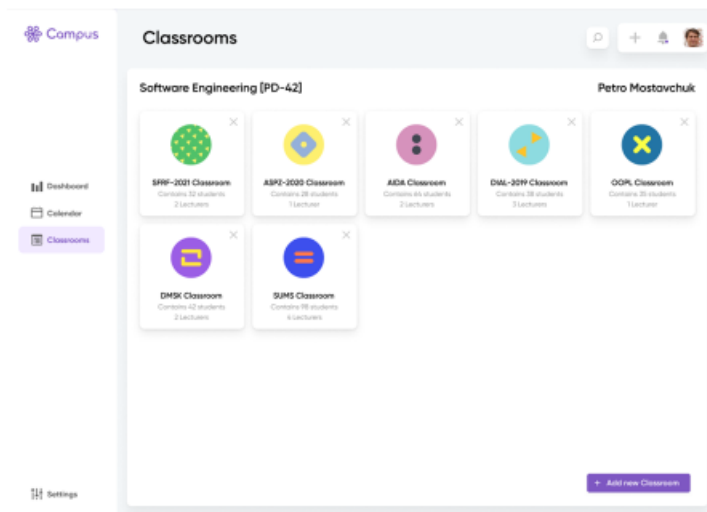


Рисунок 8 – Каркасна модель високої якості сторінки «Calendar»

Каркасна модель – деталізований прототип, що чітко виділяє компоненти та їх складові що мають міститися в елементі дизайн системи, не має кінцевих стилей.

Елемент дизайн системи



Такий вид шаблону сторінки є кінцевою реалізацією, що має бути відображена в додатку. Містить кінцеві стилі, розмір, оформлення елементів, відстань між ними, їх контент та атрибутику продукту

Оцінкове тестування

Оцінкове дослідження полягало у збірці кількісних характеристик програмного інтерфейсу, для цього було використано закони Хіка та Фіттса

Формула Хіка-Хаймана

Вчені вивели формулу, яка описує логарифмічну залежність між часом реакції і кількістю об'єктів, з яких потрібно вибрати.

$$RT = a + b * \log_2(n + 1)$$

де **RT** - це загальний час реакції,
a - загальний час, який не бере участі у прийнятті рішень,
b - це емпірично отримана константа, заснована на часі когнітивного процесу для кожного варіанту, який для людини становить 0.155 ,
n - кількість рівнозначних альтернативних варіантів, з яких потрібно вибрати.

Формула Фіттса

Закон Фіттса - загальний закон, що стосується сенсорно-моторних процесів, що зв'язує час руху з точністю руху і з відстанню переміщення

$$T = a + b * \log_2\left(\frac{D}{W} + 1\right)$$

де **T** - середній час, що витрачається на вчинення дії,
a - час запуску/зупинки пристрою (дослідницьким шляхом виявлено, як 56 мс),
b - величина, що залежить від типової швидкості пристрою, (дослідницьким шляхом виявлено, як 155 мс),
D - відстань від точки старту до центру об'єкта,
W - ширина об'єкта, виміряна вздовж осі руху.

Порівняння результатів розрахунків за законом Хіка

Campus v.1.0	a(с)	n	RT(с)
Сторінка «Dashboard»	3.34	6	3.77
Сторінка «Tasklist/Calendar»	2.45	4	2.8
Сторінка (головна) Notebooks/Classrooms	3.1	3	3.42

Час реакції, отриманий з розмірів та числових характеристик, які були зібрані з компонентів додатку, версії 1.0

Campus v.1.1	a(с)	n	RT(с)
Сторінка «Dashboard»	3.11	6	3.54
Сторінка «Tasklist/Calendar»	2.31	3	2.62
Сторінка (головна) Notebooks/Classrooms	2.7	3	3.01

Час реакції, отриманий з розмірів та числових характеристик, які були зібрані з компонентів додатку, версії 1.1

Порівняння результатів розрахунків за законом Фіттса

	Campus v.1.0		
Компонент нещодавніх Класів - Сторінка «Dashboard»	D(px)	W(px)	T(мс)
Сторінка «Tasklist/Calendar»	596	333	285.42
Елемент компоненту Класів - Сторінка «Classrooms»	552	345	269.67

Мінімальний час для вчинення дії, отриманий з числових характеристик, які були зібрані з компонентів додатку, версії 1.0

	Campus v.1.1		
Компонент нещодавніх Класів - Сторінка «Dashboard»	D(px)	W(px)	T(мс)
Сторінка «Tasklist/Calendar»	682	426	269.75
Елемент компоненту Класів - Сторінка «Classrooms»	359	212	277.56

Мінімальний час для вчинення дії, отриманий з числових характеристик, які були зібрані з компонентів додатку, версії 1.1

Інструмент аналізу інтерфейсу користувача

Результати аналізу, отримані під час тестування та розраховані за законом Фіттса

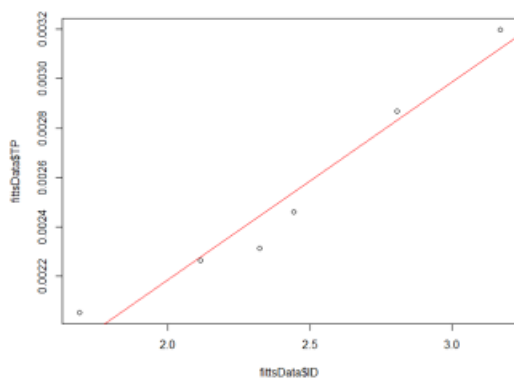


Рисунок 9 - Графік TD~ID (Пропускна здатність – Індекс складності)

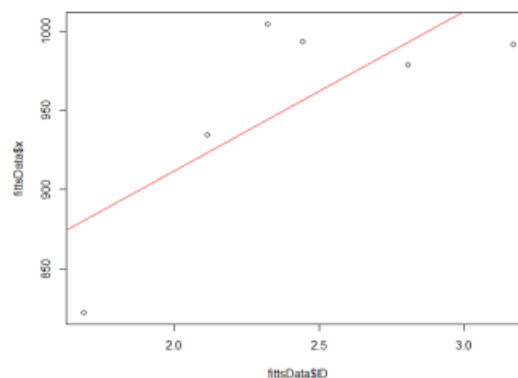


Рисунок 10 - Графік MT~ID (Час на ітерацію – Індекс складності)

Висновки

1. У ході роботи було розглянуто та реалізовано підходи, щодо проектування користувацького інтерфейсу. Результатом роботи стала дизайн система – набір графічних прототипів, шаблонів гайдлайнів , які допоможуть front-end розробнику втілити дизайн-концепт в front end частину додатку. Також такий набір буде корисним при розширенні функціоналу додатку, адже дизайн основних компонентів вже готовий, єдине, що залишається - додати контент.
2. Невід'ємною частиною розробки став аналіз поточних версій додатку – аби впевнитися , що наступні генерації дизайн системи дійсно забезпечують ефективну роботу користувачеві з веб-додатком. Такий підхід допоміг дізнатися проблеми кожній версії в порівнянні один з одним, та визначив напрямлення, яким потрібно було приділити більше уваги в контексті наповнення шаблонів компонентів, з якими має ітерувати кінцевий користувач.