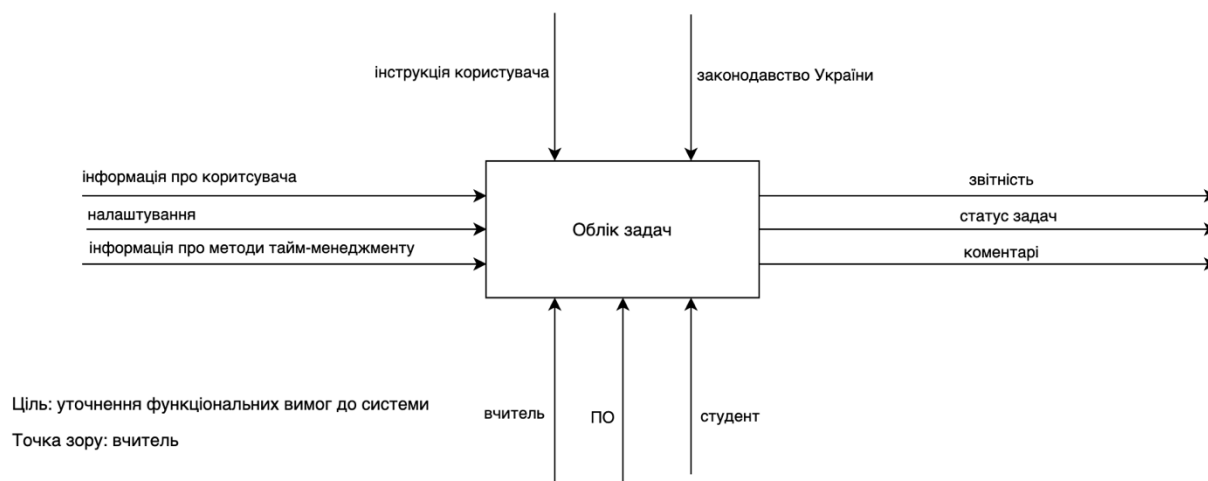


Додаток А
Графічні матеріали

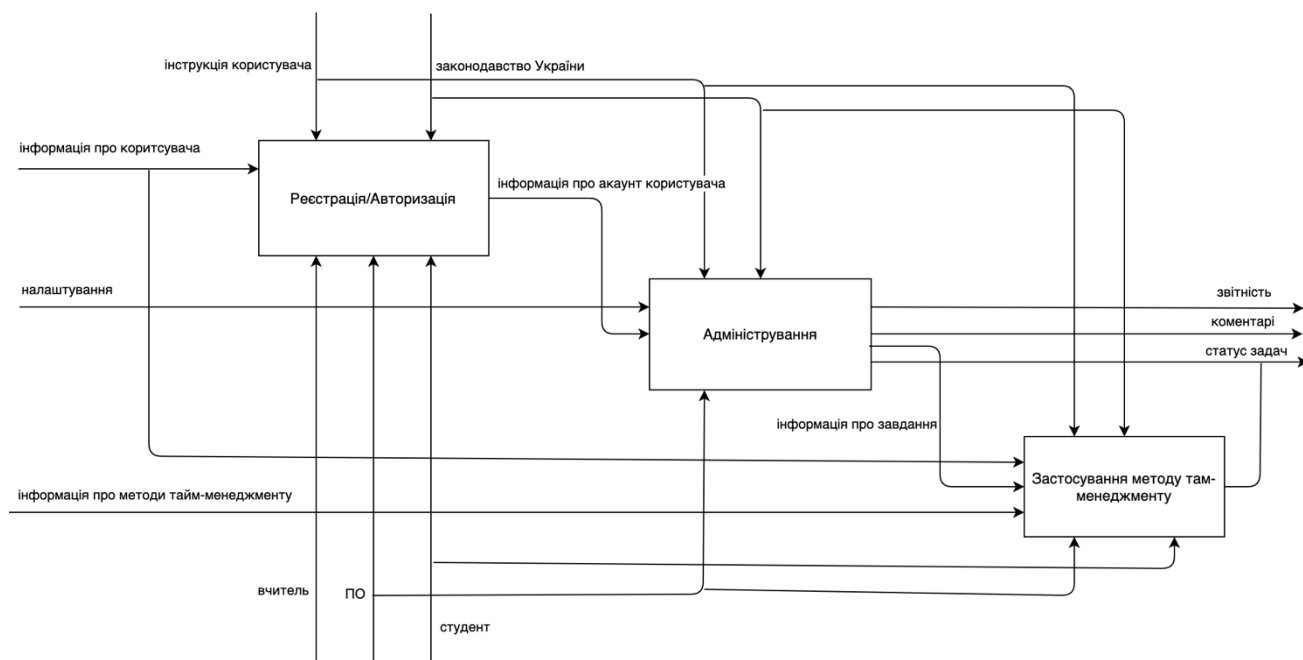
ГЮИК.507200.010

КОНТЕКСТНА ДІАГРАМА СИСТЕМИ



					ГЮИК.507200.010			
					Контекстна діаграма системи	Лім.	Маса	Масштаб
Змін.	Лист	№ докум.	Підпис	Дата				
Розроб.		Селіванова К.І.						
Перевір.		Ситнікова П.Е.						
Т. Контр.						Лист 1		Листів 1
Реценз.						ХНУРЕ Кафедра СТ		
Н. Контр.		Ситнікова П.Е.						
Затверд.		Гребеннік І.В.						

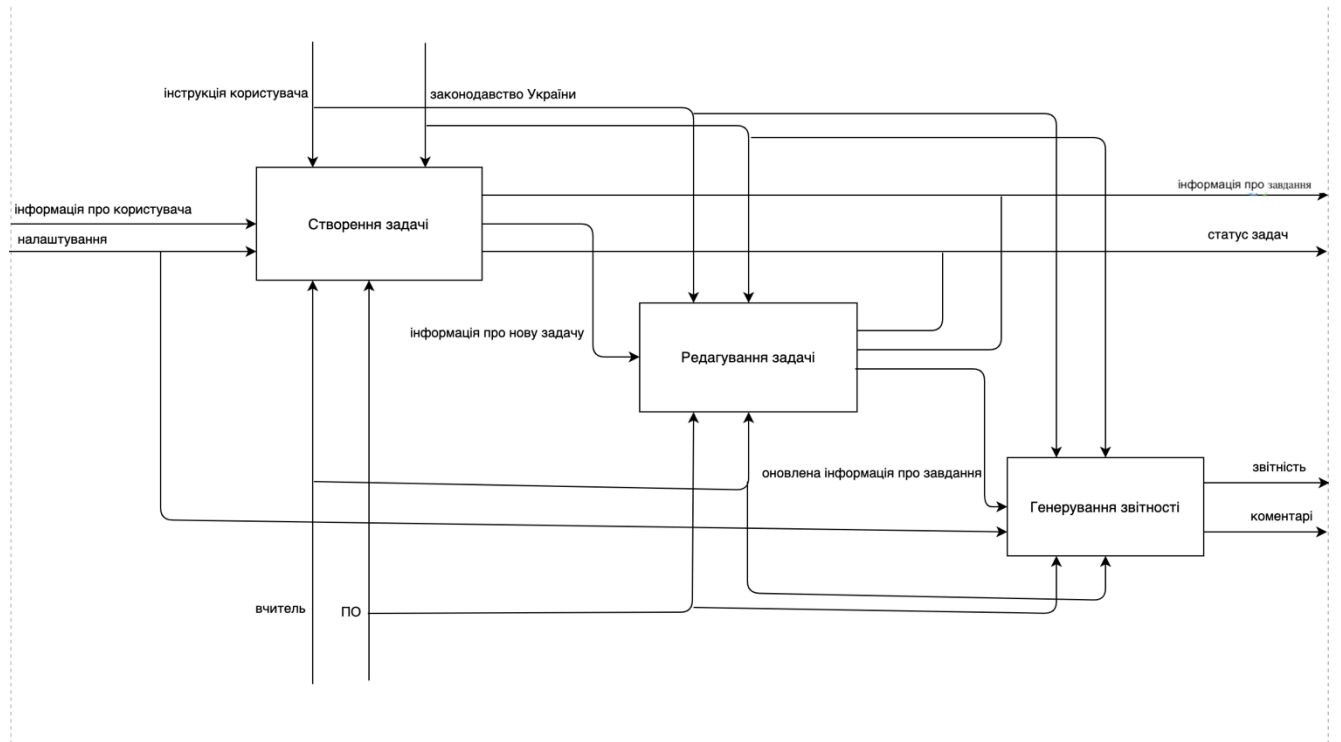
ДЕКОМПОЗИЦІЯ КОНТЕКСТНОЇ ДІАГРАМИ СИСТЕМИ



					ГЮИК.507200.010					
					Декомпозиція контекстної діаграми системи	Лім.			Маса	Масштаб
Змін.	Лист	№ докум.	Підпис	Дата						
Розроб.	Селіванова К.І.									
Перевір.	Ситнікова П.Е.									
Т. Контр.										
Реценз.						Лист 1			Листів 1	
Н. Контр.	Ситнікова П.Е.					ХНУРЕ Кафедра СТ				
Затверд.	Гребеннік І.В.									

ГЮИК.507200.010

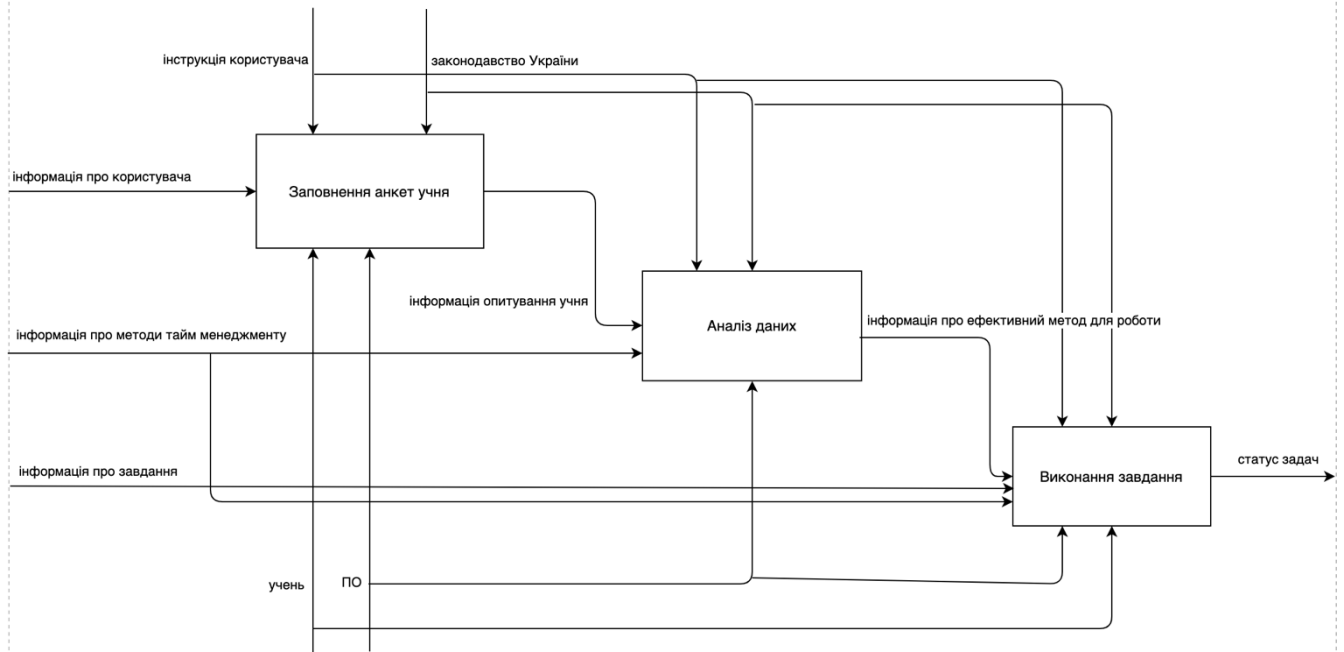
ДЕКОМПОЗИЦІЯ КОНТЕКСТНОЇ ДІАГРАМИ СИСТЕМИ



					ГЮИК.507200.010			
					Декомпозиція контекстної діаграми системи	Лім.	Маса	Масштаб
Змін.	Лист	№ докум.	Підпис	Дата				
Розроб.		Селіванова К.І.						
Перевір.		Ситнікова П.Е.						
Т. Контр.						Лист 1		Листів 1
Реценз.						ХНУРЕ Кафедра СТ		
Н. Контр.		Ситнікова П.Е.						
Затверд.		Гребеннік І.В.						

ГЮИК.507200.010

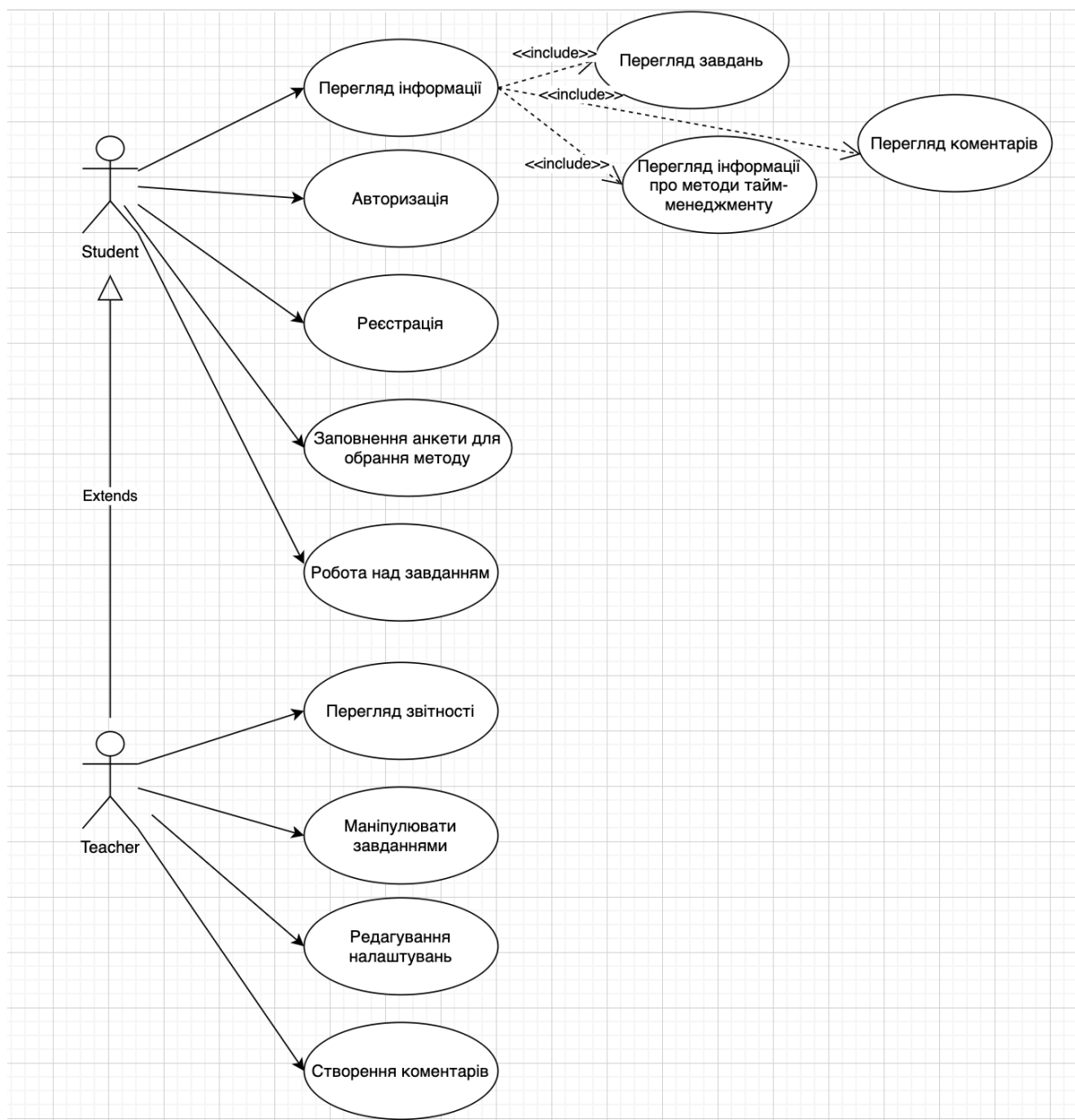
ДЕКОМПОЗИЦІЯ КОНТЕКСТНОЇ ДІАГРАМИ СИСТЕМИ



					ГЮИК.507200.010			
					Декомпозиція контекстної діаграми системи	Літ.	Маса	Масштаб
Змін.	Лист	№ докум.	Підпис	Дата				
Розроб.	Селіванова К.І.							
Перевір.	Ситнікова П.Е.							
Т. Контр.						Лист 1		Листів 1
Реценз.						ХНУРЕ Кафедра СТ		
Н. Контр.	Ситнікова П.Е.							
Затверд.	Гребеннік І.В.							

ГЮИК.507200.010

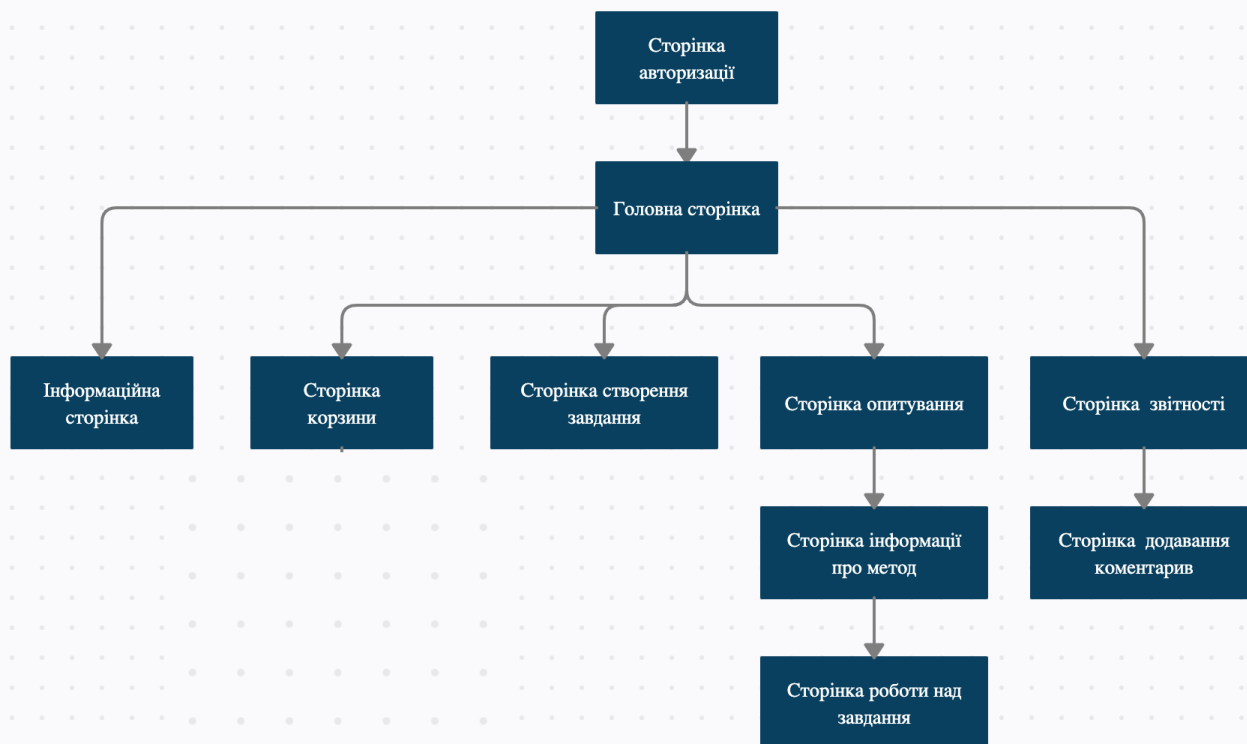
USE CASE ДІАГРАМА



					ГЮИК.507200.010			
					Use Case діаграма	Лім.	Маса	Масштаб
Змін.	Лист	№ докум.	Підпис	Дата				
Розроб.	Селіванова К.І.							
Перевір.	Ситнікова П.Е.							
Т. Контр.						Лист 1		Листів 1
Реценз.						ХНУРЕ Кафедра СТ		
Н. Контр.	Ситнікова П.Е.							
Затверд.	Гребеннік І.В.							

ГЮИК.507200.010

СТРУКТУРНА СХЕМА



					ГЮИК.507200.010			
					Структурна схема	Лім.	Маса	Масштаб
Змін.	Лист	№ докум.	Підпис	Дата				
Розроб.		Селіванова К.І.						
Перевір.		Ситнікова П.Е.						
Т. Контр.						Лист 1		Листів 1
Реценз.						ХНУРЕ Кафедра СТ		
Н. Контр.		Ситнікова П.Е.						
Затверд.		Гребеннік І.В.						

ГЮИК.507200.010

РЕАЛІЗОВАНА ВЕРСІЯ ІНТЕРФЕЙСУ

Fill out the form to determine the appropriate method

Course

Approximate amount of free time after school

Need a timer to perform techniques or not

Self-assessment

How much time can a student spend planning

Free work schedule or not

CANCEL

SUBMIT

to a specific date and time. It is the importance and urgency considered together, that influence the prioritization.

Type I: "Important and urgent"
These are cases, the late execution of which will lead to significant damage to your business (for example, renewing licenses, submitting tax reports, etc.). A certain proportion of such cases will inevitably be present in the life of every person. However, planning ahead (Type I cases - "important but not urgent") can prevent many crises from occurring (for example, by studying legislation, building good relationships with influential people). It can also be projects with a "burning" deadline, urgent. For example, visiting a doctor for health problems, submitting an article to a journal by a tough deadline, or submitting a research report. We have no choice here. The efforts of the group must be done, period. Otherwise, there will be serious problems.

Type II: "Important but not urgent"
These are things that are focused on the future: training, studying promising areas of business development, improving equipment, restoring health and working capacity. Cases leading to your strategic goal. For example, learn a foreign language in order to move to work in another, more promising organization. It is also a prevention of problems: to keep yourself in good physical shape. Unfortunately, we often neglect such matters, we postpone their decision on the back burner. As a result, the language has not been learned yet, incomes are not growing, but declining health is felt. These matters have an interesting feature: if they are neglected for a long time, they become "important" (urgent). After all, if you do not appear at the dentist at least once a year, then sooner or later an urgent visit to him will become inevitable.

Type III: "Not important, but urgent"
Many of these activities are not really of great benefit to life. We do them only because they tell us (a long telephone conversation in the study of an advertisement that came by mail), or out of habit (visiting exhibitions at which there is nothing new anymore). This is exactly the same bad routine that takes a lot of time and effort from you.

Type IV: "Not important and not urgent"
These are all sorts of ways of "killing time": alcohol abuse, "light reading", watching more (films, etc.). We often resort to this when we do not have the strength for productive work (we are confused with real rest and communication with loved ones and friends - very important matters). This is the "waste" stage of our time.

Method "ABCD"

The ABCD method is an effective prioritization method that you can use on a daily basis. This method is simple and so effective that it can be used by anyone, even you, the rank of the most productive and productive people in your field of activity. The strength of the method lies in its simplicity. This is how it works. You start by making a list of everything you have to do for the coming day. Think to yourself: "After that, you put the letter A, B, C, D or E in front of each item on your list."

A Type A task is defined as something that matters most at this small amount of time you have to do. Think to yourself: "After that, you put the letter A, B, C, D or E in front of each item on your list."

A Type B task is defined as something that matters less at this small amount of time you have to do. Think to yourself: "After that, you put the letter B, C, D or E in front of each item on your list."

A Type C task is defined as something that matters even less at this small amount of time you have to do. Think to yourself: "After that, you put the letter C, D or E in front of each item on your list."

A Type D task is defined as something that matters even less at this small amount of time you have to do. Think to yourself: "After that, you put the letter D or E in front of each item on your list."

A Type E task is defined as something that matters even less at this small amount of time you have to do. Think to yourself: "After that, you put the letter E in front of each item on your list."

Technique "Timing"

Technique "Timing" is a method of time for any task, work for each primitive unit as moving resources or lunch. You write every task on a separate card, then write a word, you can solve the following tasks:

- determine when you will begin;
- to identify chronological, time, time order;
- develop a sense of efficiency and "sense of time".

We all counted for you

A red card means that the time is up and a student has missed the task.

A green card means student completed assignment.

A white card means that the task has not yet been started and the student still has time to complete it.

A orange card means student failed assignment.

					ГЮИК.507200.010			
					Реалізована версія інтерфейсу			
Змін.	Лист	№ докум.	Підпис	Дата				
Розроб.		Селіванова К.І.						
Перевір.		Ситнікова П.Е.						
Т. Контр.								
Реценз.								
Н. Контр.		Ситнікова П.Е.			ХНУРЕ Кафедра СТ			
Затверд.		Гребеннік І.В.						

Додаток Б
Посібник користувача
ГЮИК.507200.010 34 01 ИЗ

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»

Керівник кваліфікаційної роботи

_____ доц. Ситнікова П.Е.
(підпис)

«___» _____ 2021р.

ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ МЕТОДІВ ТАЙМ-МЕНЕДЖМЕНТУ
В СИСТЕМІ ДИСТАНЦІЙНОЇ ОСВІТИ

Посібник користувача

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК.507200.010 34 01 ИЗ – ЛЗ

Виконавець:

студент групи ІТПм 20-1
(Назва групи)

Селіванова Карина Іванівна
(Підпис, прізвище, ім'я, по батькові)

«___» _____ 2021р.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

ЗАТВЕРДЖЕНО
ГЮИК.507200.010 34 01 ИЗ – ЛЗ

ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ МЕТОДІВ ТАЙМ-МЕНЕДЖМЕНТУ
В СИСТЕМІ ДИСТАНЦІЙНОЇ ОСВІТИ

Керівництво користувача

ГЮИК.507200.010 34 01 ИЗ – ЛЗ

Листів 8

ЗМІСТ

Вступ	3
1 Призначення та умови використання	4
2 Підготовка до роботи	4
3 Опис функцій	5
4 Аварійні ситуації	5
5 Рекомендації з освоєння.....	6

Вступ

Розроблений програмний продукт розрахований на студентів та викладачів середньої школи або університету. Система що розробляється має багато переваг над своїми аналогами. Вона надає можливість скористатися пробним періодом на пів року, призначена для школи або університету та легка у використанні, не потребує додаткового налаштування. Перейшовши за посиланням на web сайт, що розробляється клієнт повинен зареєструватися або авторизуватися, якщо він вже має обліковий запис в БД.

Отже, система буде корисна середнім закладам освіти, закладам совіти з маленьким бюджетом та для приватних уроків, якщо учень не може бути присутнім на уроці. За допомогою методів тайм-менеджменту та функціоналу звітності, викладач легко зможе контролювати виконання завдань, а студент правильно розподіляти свій час.

Головним завданням системи є наданням можливості дистанційного навчання для учнів середніх шкіл та університетів. Адже на сьогоднішній день в Україні немає інформаційної системи дистанційного навчання саме для учнів шкіл. Якщо учень має фізичні вади, учитель проводить уроки вдома в учня. Дана система надасть можливість робити це за допомогою Інтернету.

Програма розрахована не лиш для учнів шкіл. Вона легка у освоюванні та налаштуванні, отже нею можуть користуватися університети, заклади позашкільної освіти або індивідуальні курси.

В свою чергу користувачі повинні мати базові навички роботи в інтернеті та з web ресурсами.

Перед початком роботи викладач повинен зареєструватися у системі, отримати ідентифікатор групи та розіслати його своїм студентам. Студенти у свою чергу також повинні зареєструватися на web сайті, але при заповненні форми обрати роль користувача Student і у додатковому полі ввести ідентифікатор групи.

1 Призначення та умови використання

Розроблений web сайт використовується для дистанційного навчання учнів з використанням методів тайм-менеджменту.

Для роботи з інформаційною системою, необхідно мати комп'ютер або смартфон з можливістю підключення Інтернету. Стабільний сигнал буде гарантувати легкість та безперебійність роботи системи.

На комп'ютері користувача необхідно встановити web браузер (якщо він ще не встановлений). Вимог до браузера немає, це може бути, як Opera, Chrome, Firefox так Safari. Адже інформаційна система працює у всіх браузерах та на всіх розширеннях екрану.

Зареєструвавшись у системі необхідно перейти на інформаційну сторінку та ознайомитись з інструкцією користувача. Знайти web сайт можна знайти у пошуковому вікні Google за посиланням.

2 Підготовка до роботи

Дистрибутив програми немає фізичного носія. Інформаційна система розташована на хостингу Heroku і доступна за посиланням.

Першим кроком необхідно впевнитись, що на смартфон або комп'ютер встановлено web браузер (Opera, Chrome, Firefox так і Microsoft Edge). Для Mac OS або iPhone це Safari. Наступна дія це перевірити Інтернет підключення, та стабільний сигнал.

Web сайт, що розробляється знаходить у мережі Інтернет. Тому отримати доступ до нього дуже просто, необхідно ввести у пошукове поле <https://distans-learning-pomodoro.herokuapp.com/>. Таким чином доступ до web ресурсу можна отримати по всьому світу. Якщо девайс на якому відкрили web сторінку має

нестабільний Інтернет сигнал, користувачу прийде сповіщення та попередження про це, але останні дії збережуться.

Першою ознакою, що система працює це завантаження сторінки авторизації при переході за посиланням.

3 Опис функцій

Метою роботи інформаційної системи є забезпечення роботи головної функції web сайту. Головною функцією системи є облік задач, або ж дистанційне навчання з використанням методів тайм-менеджменту. Для забезпечення роботи основної функції можуть використовуватись допоміжні функції системи.

Web сайт дозволяє виконувати такі функції:

- перегляд інформаційної сторінки, щоб ознайомитись з інструкцією використання;
- пройти анкетування для підбору найкращого метод тайм-менеджменту;
- перегляд завдань;
- створення нових задач;
- редагування створених завдань;
- видалення завдань;
- перегляд звітності по кожному студенту у вигляді діаграм;
- додавання коментарів студенту;
- робота над завданням за допомогою одного з чотирьох методів тайм-менеджменту;
- для доступу до всіх функцій необхідно авторизуватись.

4 Аварійні ситуації

При виникненні проблем в роботі web сайту необхідно виконати такі дії:

- перезавантажити сторінку;
- перезавантажити браузер або девайс;

- перевірити Інтернет підключення;
- видалити кеш або куки;
- заново авторизуватися;

Ці дії необхідно виконувати до того моменту, поки робота системи заново не відновиться.

Якщо не вдалось налагодити роботу системи, слід звернутися до служби підтримки. Всі останні дії та інформація будуть збережені і при новому запуску web сторінки будуть відображатися.

Клієнтські web додатки не дають можливість несанкціонованого доступу до даних.

В разі несанкціонованого доступу до серверної частини в першу чергу необхідно змінити всі облікові дані, що використовуються для роботи сервера. Користувач обов'язково буде повідомлений про необхідність зміни своїх облікових даних.

5 Рекомендації з освоєння

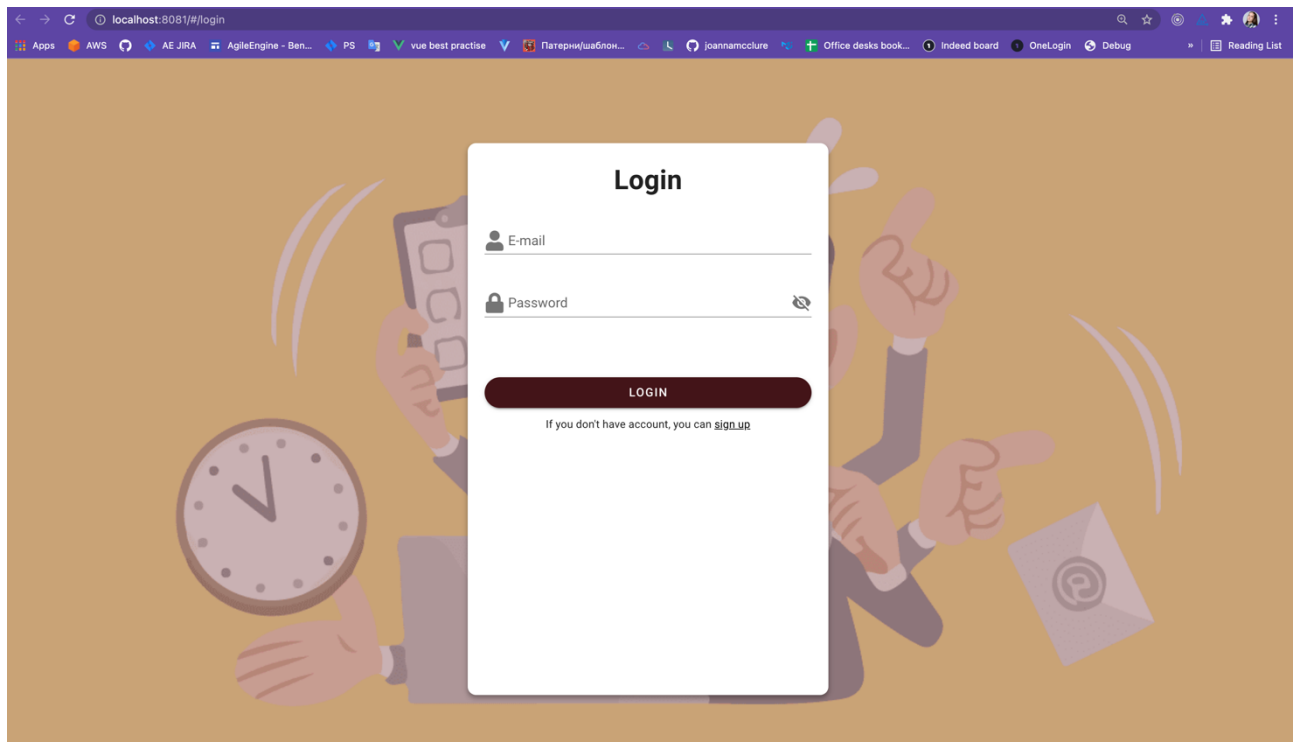
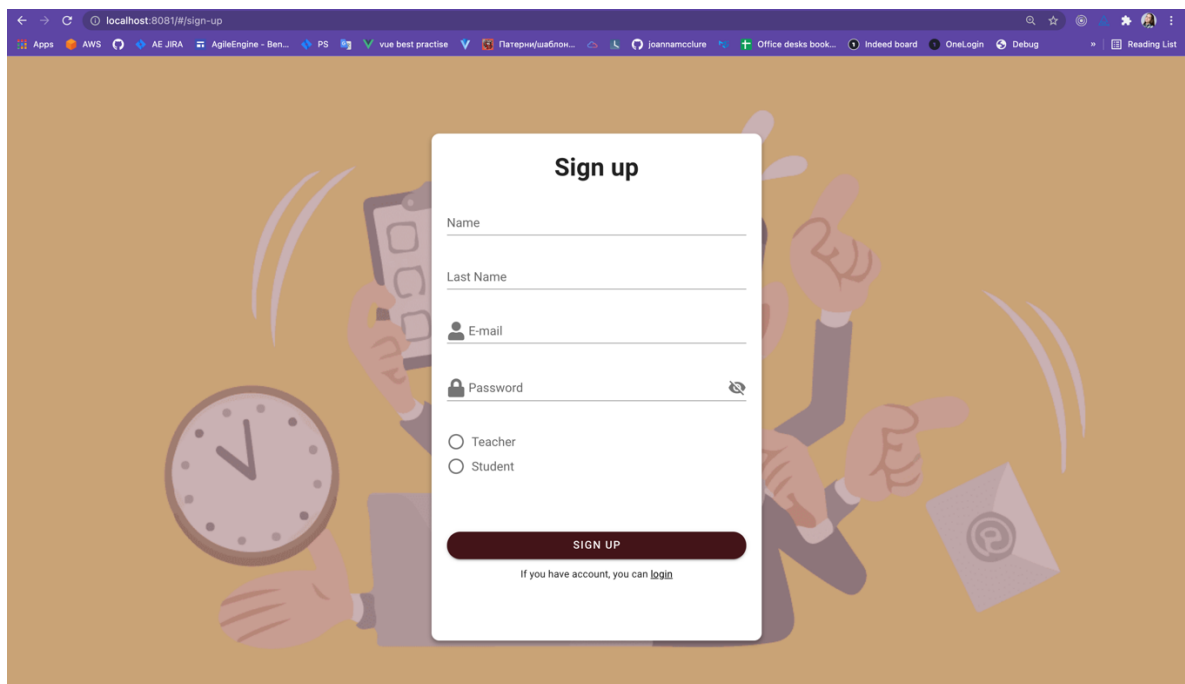


Рисунок 1 – Сторінка авторизації

Після переходу за посиланням та відкриття web сайту, користувач повинен бути автоматично перенаправлен на сторінку авторизації з формою, яка складається з полів e-mail та password, яка зображена на рисунку 1. Для користування повним функціоналом системи користувач повинен авторизуватися або зареєструватися. Форма реєстрації представлена на рисунку 2.



The image shows a web browser window with the address bar displaying 'localhost:8081/#/sign-up'. The browser's tab bar shows several open tabs, including 'Apps', 'AWS', 'AE JIRA', 'AgileEngine - Ben...', 'PS', 'vue best practise', 'Патерни/шаблон...', 'joannamcclore', 'Office desks book...', 'Indeed board', 'OneLogin', 'Debug', and 'Reading List'. The main content area features a 'Sign up' form with the following fields: 'Name', 'Last Name', 'E-mail' (with a user icon), 'Password' (with a lock icon and a toggle for visibility), and two radio buttons for role selection: 'Teacher' and 'Student'. Below the form is a dark 'SIGN UP' button. Underneath the button is a link that says 'If you have account, you can login'. The background of the page is a solid brown color with stylized illustrations of hands holding a smartphone, a clock, and a document.

Рисунок 2 – Сторінка реєстрації

Одразу після вище перерахованих дій, користувачу системи стане доступний весь функціонал, в залежності від його ролі. Перша сторінка куди автоматично перенаправляє користувача після авторизації або реєстрації це головна сторінка, рисунок 3, на якій можна проводити маніпуляції над завданнями.

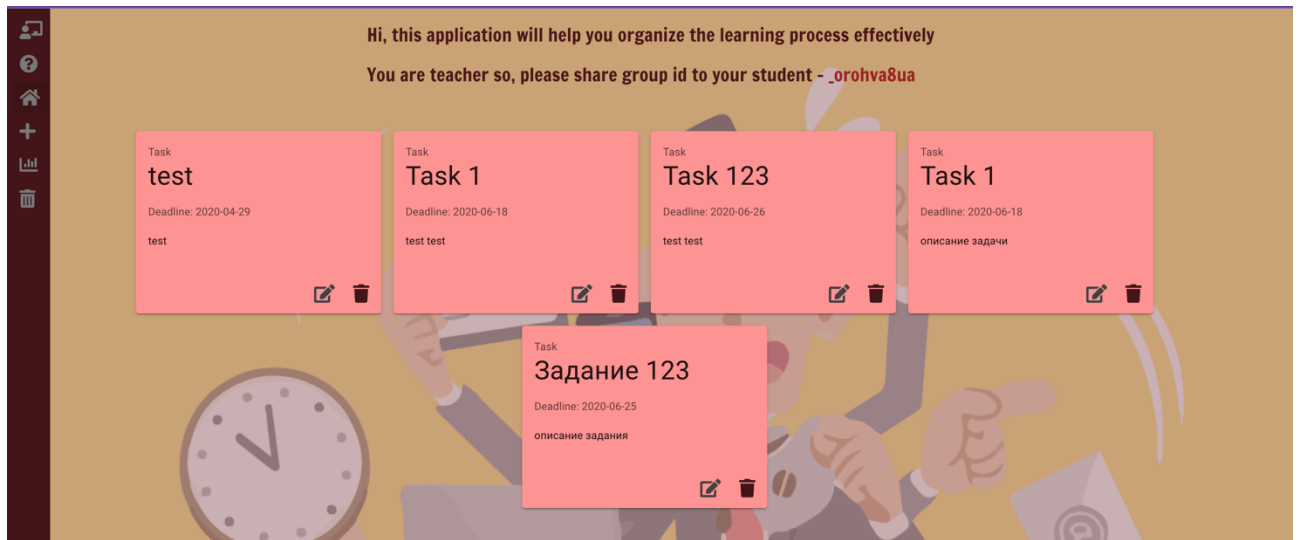


Рисунок 3 – Головна сторінка

На боковій панелі меню, користувач у ролі вчителя може натиснути на пункт «Додати нове завдання» та створити нове завдання для учня, рисунок 4.

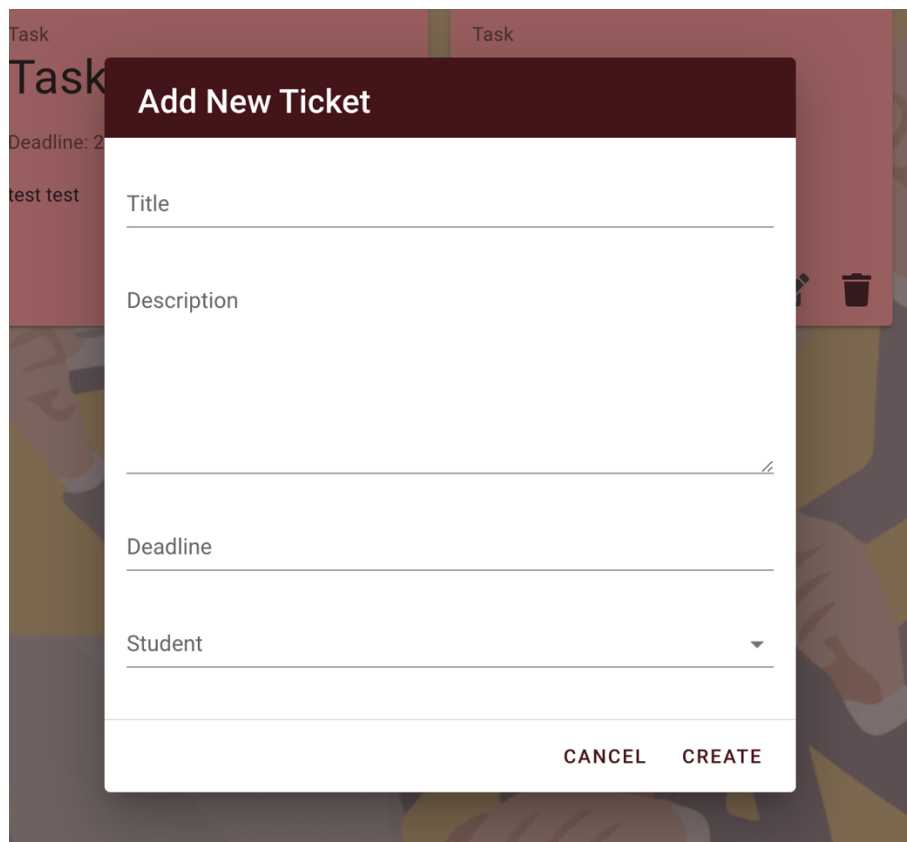


Рисунок 4 – Модальне вікно створення нового завдання

Модальне вікно для редагування вже створеного завдання відповідає рисунку 4. Наступний крок це виконання завдання, яке створив учитель. Перед початком роботи студент проходить опитування для аналізу його ритму життя та для надання рекомендацій щодо методу тайм-менеджменту. Форма опитування представлена на рисунку 5.

Fill out the form to determine the appropriate method

Course

Approximate amount of free time after school

☒ Need a timer to perform techniques or not

Self-assessment

How much time can a student spend planning

☒ Free work schedule or not

CANCEL SUBMIT

Рисунок 5 – Форма опитування

Результат опитування представлений у вигляді інформаційної сторінки рисунок 6.

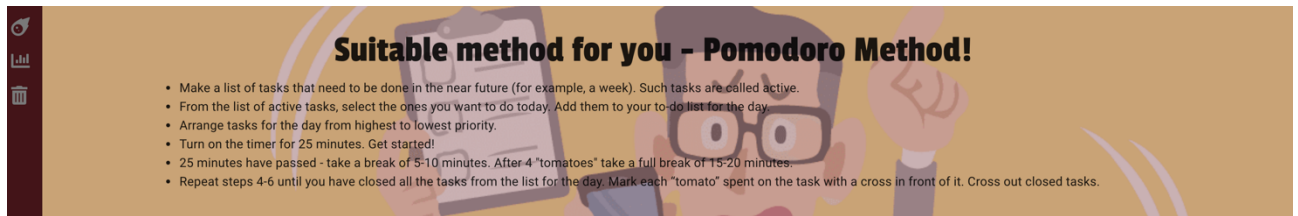


Рисунок 6 – Сторінка результату опитування

Наступний крок – це робота над завданням за допомогою методу тайм-менеджменту, рисунок 7.

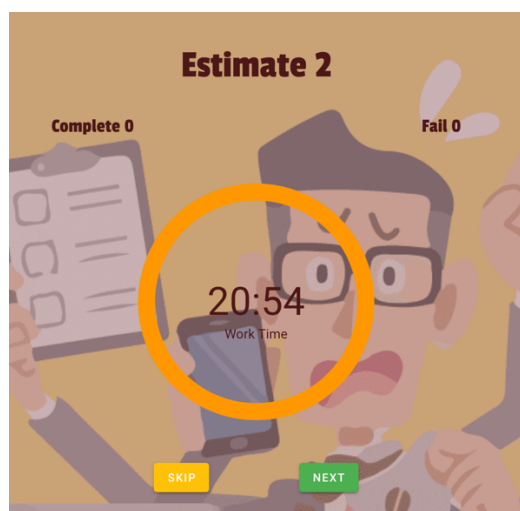


Рисунок 7 – Сторінка роботи над завданням

Кожен студент і викладач може переглянути сторінку звітності для аналізу активності студента рисунок 8.

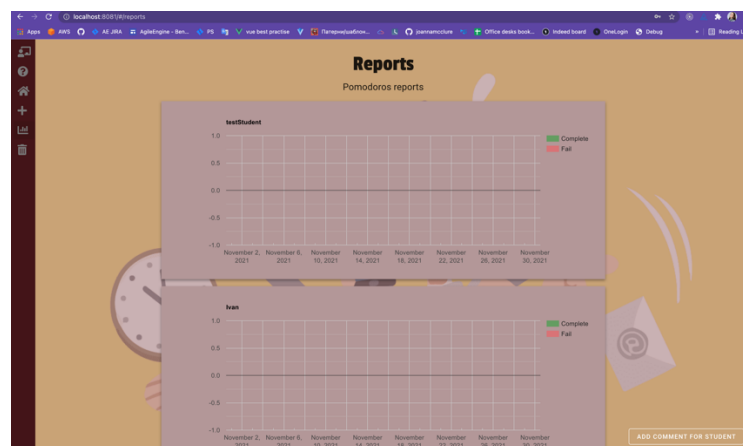


Рисунок 8 – Сторінка звітності

Додаток В
Текст програми

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»

Керівник кваліфікаційної роботи

_____ доц. Ситнікова П.Е.
(підпис)

« ____ » _____ 2021р.

ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ МЕТОДІВ ТАЙМ-МЕНЕДЖМЕНТУ
В СИСТЕМІ ДИСТАНЦІЙНОЇ ОСВІТИ

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ
ГЮИК.507200.010 – 01 12 01– ЛЗ

Виконавець:

студент групи ІТПм 20-1
(Назва групи)

Селіванова Карина Іванівна

(Підпис, прізвище, ім'я, по батькові)

« ____ » _____ 2021р.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Затверджено
ГЮИК.507200.010 – 01 12 01

ДОСЛІДЖЕННЯ ТА ЗАСТОСУВАННЯ МЕТОДІВ ТАЙМ-МЕНЕДЖМЕНТУ
В СИСТЕМІ ДИСТАНЦІЙНОЇ ОСВІТИ

Текст програми

ГЮИК.507200.010 – 01 12 01

Листів 12

App.vue

```

<template>
  <v-app id="app">
    <v-content>
      <router-view/>
    </v-content>
  </v-app>
</template>

<script>
export default {
  name: 'App',
};
</script>

<style lang="stylus">
  @import "../assets/styles/main.styl"
  #app
    background url("../assets/images/tomato-bg.png")
    background-attachment fixed
</style>

```

Home.vue

```

<template>
  <div class="w-100 page">
    <SideBar/>
    <div class="flex-column w-100">
      <WelcomeNotification/>
      <div class="flex-wrap mt-10 px-10 justify-center">
        <Ticket
          v-for="ticket in displayTickets"
          :key="ticket.id"
          :ticket="ticket"
        />
      </div>
    </div>
    <CommentNotification v-if="displayCommentsNotification"/>
  </div>
</template>

<script>
import SideBar from '../components/side-bar/SideBar';
import WelcomeNotification from '../components/WelcomeNotification';
import Ticket from '../components/Ticket';
import CommentNotification from '../components/CommentNotification';
import { mapGetters } from 'vuex'

export default {
  components: {
    SideBar,
    WelcomeNotification,

```

```

    Ticket,
    CommentNotification
  },
  computed: {
    ...mapGetters(['tickets', 'role', 'comments']),
    displayTickets() {
      return this.tickets.filter(ticket => !ticket.trash)
    },
    displayCommentsNotification() {
      return this.role === 'STUDENT' && this.comments.length > 0
    }
  }
}
</script>

```

```

<style lang="stylus">
  @import "../assets/styles/main.styl"
</style>

```

Login.vue

```

<template>
  <div class="custom-form">
    <h1 class="title-form">
      Login
    </h1>
    <v-form
      ref="loginForm"
      v-model="valid"
      lazy-validation
      @submit.prevent="login">
      <v-text-field
        v-model="loginForm.email"
        :rules="rules.email"
        id="email-field"
        prepend-inner-icon="fas fa-user"
        label="E-mail"/>
      <v-text-field
        class="mt-2"
        v-model="loginForm.password"
        :rules="rules.password"
        id="password-field"
        :append-icon="showPassword ? 'mdi-eye' : 'mdi-eye-off'"
        :type="showPassword ? 'text' : 'password'"
        label="Password"
        hint="At least 8 characters"
        prepend-inner-icon="fas fa-lock"
        @click:append="showPassword = !showPassword"/>
      <div class="buttons">
        <v-btn
          type="submit"
          color="primary"

```

```

      id="log-in-button"
      @click="login"
      rounded
      dark>
      Login
    </v-btn>
    <p class="notification">
      If you don't have account, you can
      <router-link
        to="/sign-up"
        class="link sign-up-link"
        color="primary">
        sign up
      </router-link>
    </p>
  </div>
  <p class="error-msg">{{ notification }}</p>
</v-form>
</div>
</template>

<script>
  import { emailRules, passwordRules } from '../utils/validation'
  import { mapActions } from 'vuex'

  export default {
    data: () => ({
      valid: true,
      showPassword: false,
      loginForm: {
        email: '',
        password: ''
      },
      notification: '',
      rules: {
        email: emailRules,
        password: passwordRules
      }
    }),
    methods: {
      ...mapActions(['loginUser']),
      login (e) {
        e.preventDefault()
        if (this.$refs.loginForm.validate()) {
          this.loginUser(this.loginForm)
            .then(() => this.$router.push('/'))
            .catch(error => this.notification = error)
        }
      }
    }
  }
}

```

```

</script>

<style lang="stylus">
  @import "../assets/styles/main.styl"
</style>

TicketAddPopup.vue
<template>
  <v-card>
    <v-card-title
      class="headline red white--text"
      primary-title
    >
      Add New Ticket
    </v-card-title>

    <v-form
      ref="form"
      class="px-4 py-4"
      lazy-validation
    >
      <v-text-field
        v-model="addTicketForm.title"
        id="title-task"
        label="Title"
        required
      ></v-text-field>
      <v-textarea
        label="Description"
        id="task-description"
        v-model="addTicketForm.description"
      ></v-textarea>
      <v-menu
        v-model="menu"
        :close-on-content-click="false"
        max-width="290"
      >
        <template v-slot:activator="{ on }">
          <v-text-field
            :value="computedDateFormattedMomentjs"
            clearable
            label="Deadline"
            id="task-deadline"
            readonly
            v-on="on"
            @click:clear="date = null"
          ></v-text-field>
        </template>
        <v-date-picker
          id="task-deadline-picker"
          v-model="addTicketForm.deadline"

```

```

      @change="menu = false"
    ></v-date-picker>
  </v-menu>
  <v-select
    v-model="addTicketForm.student"
    :items="studentsNames"
    :rules="[v => !!v || 'Item is required']"
    label="Student"
    id="task-student"
    required
  ></v-select>
</v-form>
<v-divider></v-divider>
<v-card-actions>
  <v-spacer></v-spacer>
  <v-btn
    color="primary"
    text
    @click="$emit('close')"
  >
    Cancel
  </v-btn>
  <v-btn
    color="primary"
    text
    id="create-ticket-button"
    @click="createTicket"
  >
    Create
  </v-btn>
</v-card-actions>
</v-card>
</template>

<script>
  import moment from 'moment'
  import { mapGetters, mapActions } from 'vuex'
  import { v4 as uuidv4 } from "uuid";

  export default {
    data: () => ({
      addTicketForm: {
        id: uuidv4(),
        title: '',
        description: '',
        deadline: '',
        student: '',
        trash: false,
        estimate: 0,
        complete: 0,
        fail: 0
      }
    })
  }

```

```

    },
    date: new Date().toISOString().substr(0, 10),
    menu: false,
  )),
  computed: {
    ...mapGetters(['students']),
    computedDateFormattedMomentjs () {
      return this.addTicketForm.deadline ?
moment(this.addTicketForm.deadline).format('dddd, MMMM Do YYYY') : ''
    },
    studentsNames () {
      let studentsNames = []
      this.students.map(student => {
        if(student.isAdmin === 'false') {
          studentsNames = [...studentsNames, {
            text: `${student.name} ${student.lastName}`,
            value: student.id
          }]
        }
      })
      return studentsNames
    }
  },
  methods: {
    ...mapActions(['addTicket']),
    refreshPopupData () {
      this.addTicketForm = {
        id: uuidv4(),
        title: '',
        description: '',
        deadline: '',
        student: '',
        trash: false,
        estimate: 0,
        complete: 0,
        fail: 0
      }
    },
    async createTicket () {
      await this.addTicket(this.addTicketForm)
      this.refreshPopupData()
      this.$emit('close')
    }
  }
}
</script>

<style lang="stylus">
  @import "../assets/styles/main.styl"
  .v-picker__body.theme--light div
    flex-direction column

```

```

    align-items center

    .v-date-picker-header.theme--light
      flex-direction row
</style>

Timer.vue
<template>
  <div class="w-100">
    <SideBar/>
    <div class="flex-column w-100">
      <h1 class="text-center mt-12 mb-5 timer-title">Estimate
{{ticket.estimate}}</h1>
      <div class="mx-auto w-30 justify-space-between flex-wrap">
        <div class="timer-description">Complete {{complete}}</div>
        <div class="timer-description">Fail {{fail}}</div>
      </div>
      <div class="base-timer mt-12">
        <svg
          class="base-timer__svg"
          viewBox="0 0 100 100"
          xmlns="http://www.w3.org/2000/svg">
          <g class="base-timer__circle">
            <circle
              class="base-timer__path-elapsed"
              cx="50"
              cy="50"
              r="45">
            </circle>
            <path
              :stroke-dasharray="circleDasharray"
              class="base-timer__path-remaining"
              :class="remainingPathColor"
              d="
                M 50, 50
                m -45, 0
                a 45,45 0 1,0 90,0
                a 45,45 0 1,0 -90,0
                "
              ></path>
          </g>
        </svg>
        <span class="base-timer__label">{{ formattedTimeLeft }}</span>
        <span class="base-timer__description">{{ minuteLabel }}</span>
      </div>
      <div class="mx-auto w-15 mt-12 justify-space-between timer-buttons">
        <v-btn
          color="warning"
          @click="onTimesUp(true)">
          Skip
        </v-btn>

```

```

        <v-btn
          color="success"
          @click="onTimesUp(false)">
          Next
        </v-btn>
      </div>
    </div>
  </div>
</template>

<script>
  import SideBar from '../components/side-bar/SideBar'
  import { mapGetters, mapActions } from 'vuex'

  const FULL_DASH_ARRAY = 283
  const FINISH = 'Finish'
  const WORK_TIME = 'Work Time'
  const LONG_BREAK_TIME = 'Long Break Time'
  const SHORT_BREAK_TIME = 'Short Break Time'

  export default {
    props: {
      ticket: Object
    },
    components: {
      SideBar,
    },
    data: () => ({
      currentCircle: 1,
      timePassed: 0,
      timerInterval: null,
      fail: 0,
      complete: 0
    }),
    computed: {
      ...mapGetters(['settings']),
      initialValue() {
        return this.conditionalForTime(0, this.settings.workTime*60,
this.settings.longBreakTime*60, this.settings.shortBreakTime*60)
      },

      minuteLabel() {
        return this.conditionalForTime(FINISH, WORK_TIME, LONG_BREAK_TIME,
SHORT_BREAK_TIME)
      },

      circleDasharray() {
        return `${(this.timeFraction * FULL_DASH_ARRAY).toFixed(0)} 283`
      },

      formattedTimeLeft() {

```

```

    const timeLeft = this.timeLeft
    const minutes = Math.floor(timeLeft / 60)
    let seconds = timeLeft % 60

    if (seconds < 10) {
      seconds = `0${seconds}`
    }

    return `${minutes}:${seconds}`
  },

  timeLeft() {
    return this.initialValue - this.timePassed
  },

  timeFraction() {
    const rawTimeFraction = this.timeLeft / this.initialValue
    return rawTimeFraction - (1 / this.initialValue) * (1 -
rawTimeFraction)
  },

  remainingPathColor() {
    return this.conditionalForTime('grey', 'orange', 'blue', 'green')
  },
},
watch: {
  timeLeft(newValue) {
    if (newValue === 0) {
      this.onTimesUp()
    }
  }
},
},

mounted() {
  this.startTimer()
},
methods: {
  ...mapActions(['editTicketAction']),
  async checkAction(skip) {
    if (this.currentCircle % 2 !== 0 && !skip) {
      this.complete = this.complete + 1
      this.ticket.complete = this.complete
    } else if (this.currentCircle % 2 !== 0 && skip) {
      this.fail = this.fail + 1
      this.ticket.fail = this.fail
    }
    this.ticket.timerDate = new Date()
    await this.editTicketAction(this.ticket)
  },

  addNewCircle() {

```

```

    if (this.currentCircle < (this.ticket.estimate * 2) - 1) {
      this.currentCircle = this.currentCircle + 1
      this.startTimer()
    } else if (this.currentCircle === (this.ticket.estimate * 2) - 1) {
      this.currentCircle = this.currentCircle + 1
      this.$router.push('/')
    }
  },

  onTimesUp(skip=false) {
    clearInterval(this.timerInterval)
    this.timePassed = 0

    this.checkAction(skip)

    this.addNewCircle()
  },

  startTimer() {
    this.timerInterval = setInterval(() => (this.timePassed += 1),
1000)
  },

  conditionalForTime(lastCircle, workTime, longBreak, shortBreak) {
    if (this.currentCircle === (this.ticket.estimate * 2)) {
      return lastCircle
    } else {
      if (this.currentCircle % 2 !== 0) {
        return workTime
      } else {
        if (this.currentCircle === this.ticket.estimate) {
          return longBreak
        } else {
          return shortBreak
        }
      }
    }
  }
}
}
}
</script>

<style lang="stylus">
  @import "../assets/styles/main.styl"
  .base-timer
    position relative
    width 300px
    height 300px
    margin 0 auto

  & _svg

```

```
transform scaleX(-1)
width 100%

&__circle
  fill none
  stroke none

&__path-elapsed
  stroke-width 7px
  stroke rgba(128, 128, 128, 0.44)

&__path-remaining
  stroke-width 7px
  stroke-linecap round
  transform rotate(90deg)
  transform-origin center
  transition 1s linear all
  fill-rule nonzero
  stroke currentColor

&.green
  color $short-break-time

&.orange
  color $work-time

&.blue
  color $long-break-time

&.grey
  color rgba(128, 128, 128, 0.44)

&__label
  position absolute
  width 300px
  height 300px
  top 0
  display flex
  align-items center
  justify-content center
  font-size 48px
  color $dark-red

&__description
  position absolute
  width 300px
  height 300px
  top 40px
  display flex
  align-items center
  justify-content center
```

```
    font-size 1rem
    color $dark-red

.timer-title
    font-size 3rem
    color $dark-red
    font-family 'Passion One', cursive

.timer-description
    font-size 1.5rem
    color $dark-red
    font-family 'Passion One', cursive

@media (min-width: 426px) and (max-width: 768px)
    .timer-buttons
        width 30%!important

@media (min-width: 376px) and (max-width: 425px)
    .timer-buttons
        width 70%!important
</style>
```