

ДОДАТОК А

ГРАФІЧНИЙ МАТЕРІАЛ КВАЛІФІКАЦІЙНОЇ РОБОТИ

ГЮИК.506160.025

(позначення документу)

ЗАТВЕРДЖЕНО

ГЮИК.506160.025 – ЛЗ

Дослідження методів формування проектних команд в ІТ-індустрії

Графічний матеріал

ГЮИК. 506160.025 – ЛЗ

ЛИСТІВ 14

2021 р.

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник кваліфікаційної роботи
проф. Овезгельдієв А.О.

Дослідження методів формування проектних команд в ІТ-індустрії

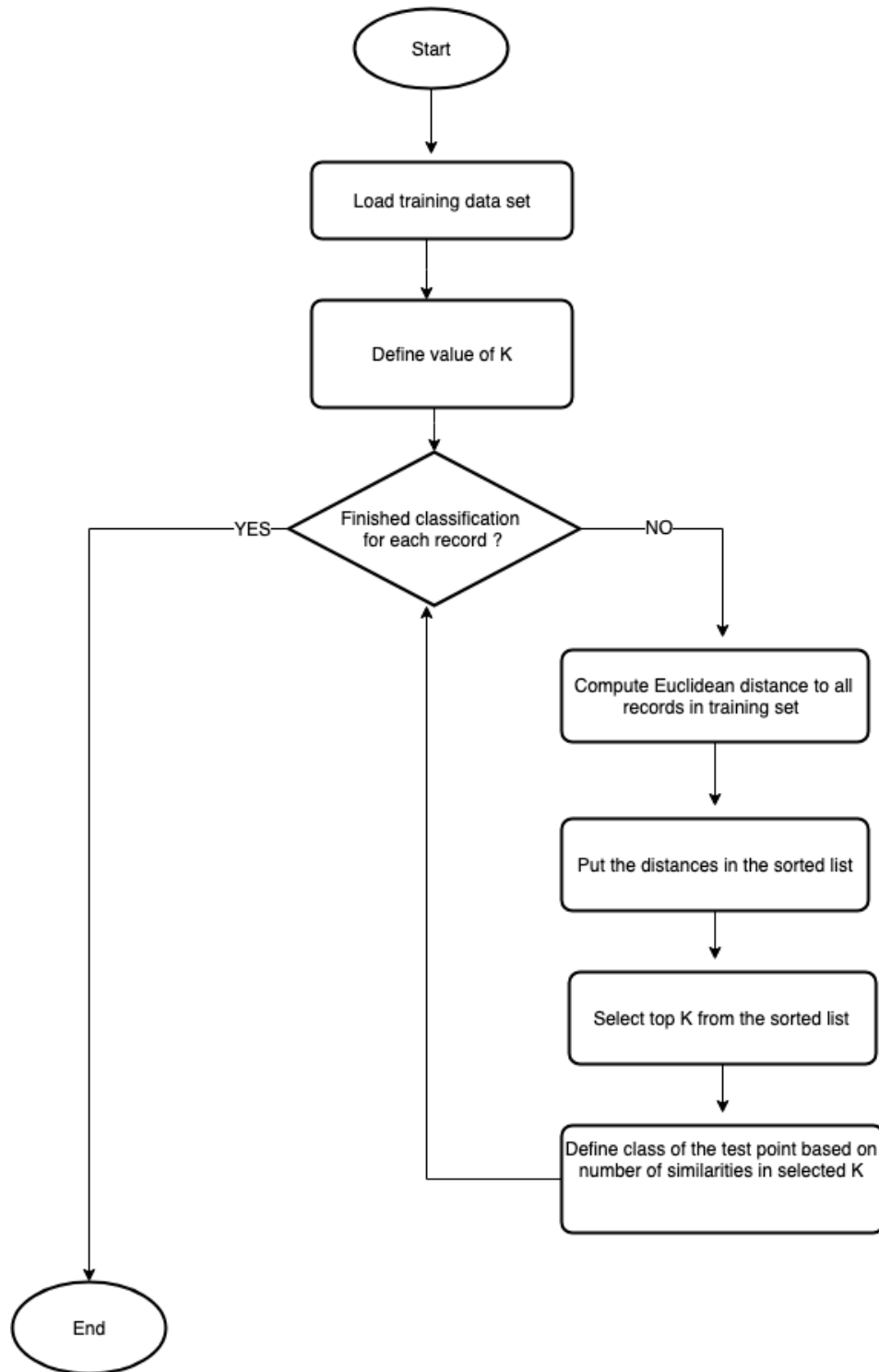
Графічний матеріал

ЛИСТ ЗАТВЕРДЖЕННЯ
ГЮИК.506160.025 ЛЗ

РОЗРОБИВ:
ст. гр. ІТІм-20-1
Івановська К.А.

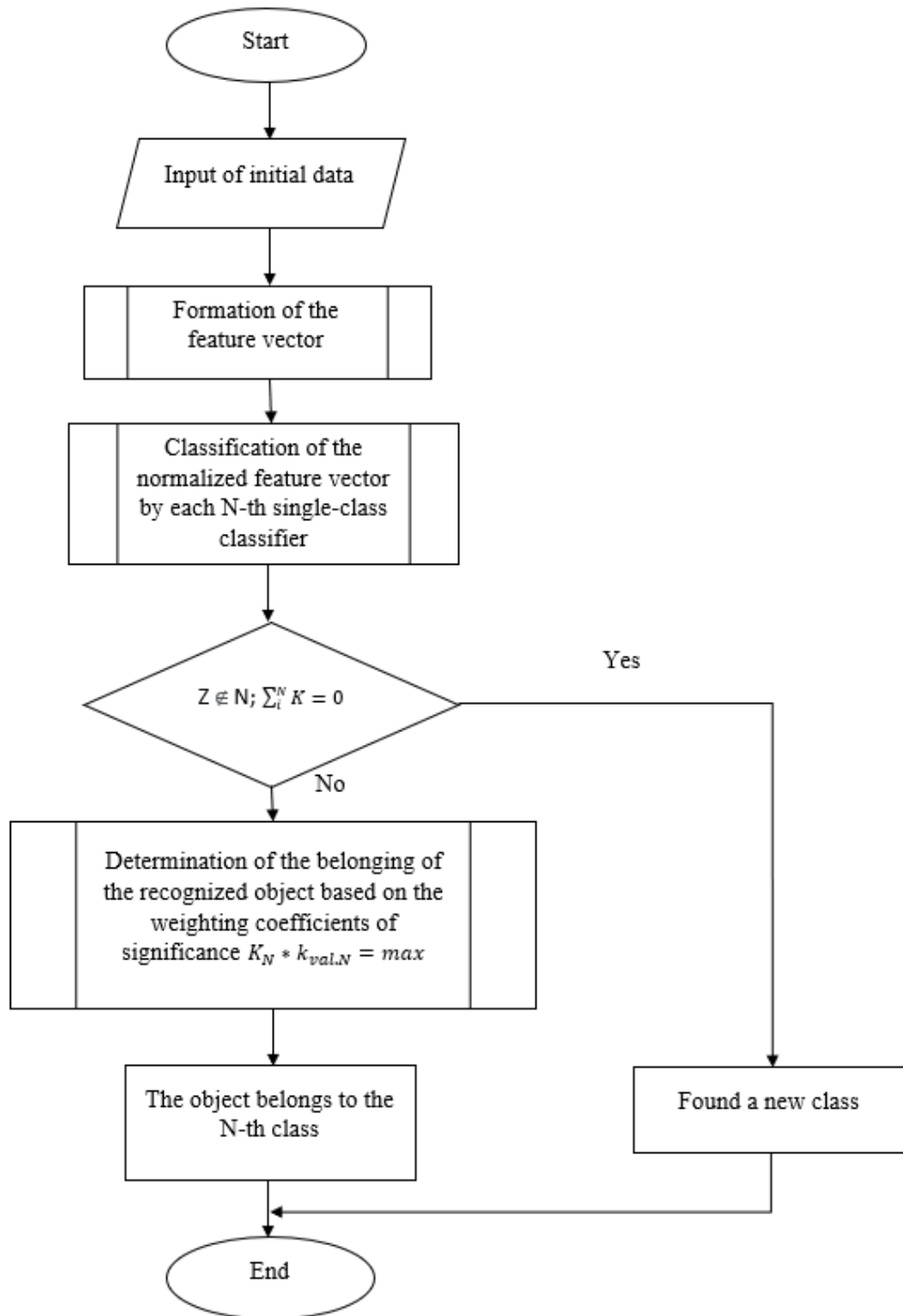
2021 р.

ЗАГАЛЬНА СХЕМА АЛГОРИТМУ К-БЛИЖНІХ СУСІДІВ



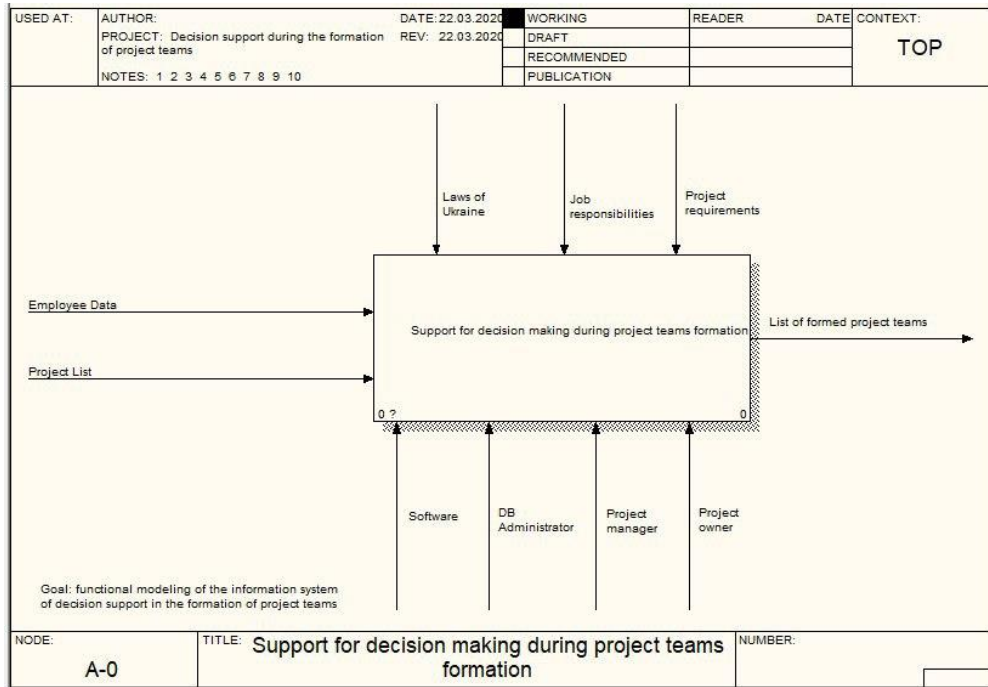
<i>Розробив</i>	<i>Івановська К.А.</i>			<i>Дослідження методів формування проектних команд в ІТ-індустрії</i>	
<i>Перевірив</i>	<i>Овезгельдієв А.О.</i>				
<i>Н.контр.</i>	<i>Овезгельдієв А.О.</i>			<i>ІТІМ-20-1</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

ЗАГАЛЬНА СХЕМА АЛГОРИТМУ МЕТОДУ ОПОРНИХ ВЕКТОРІВ



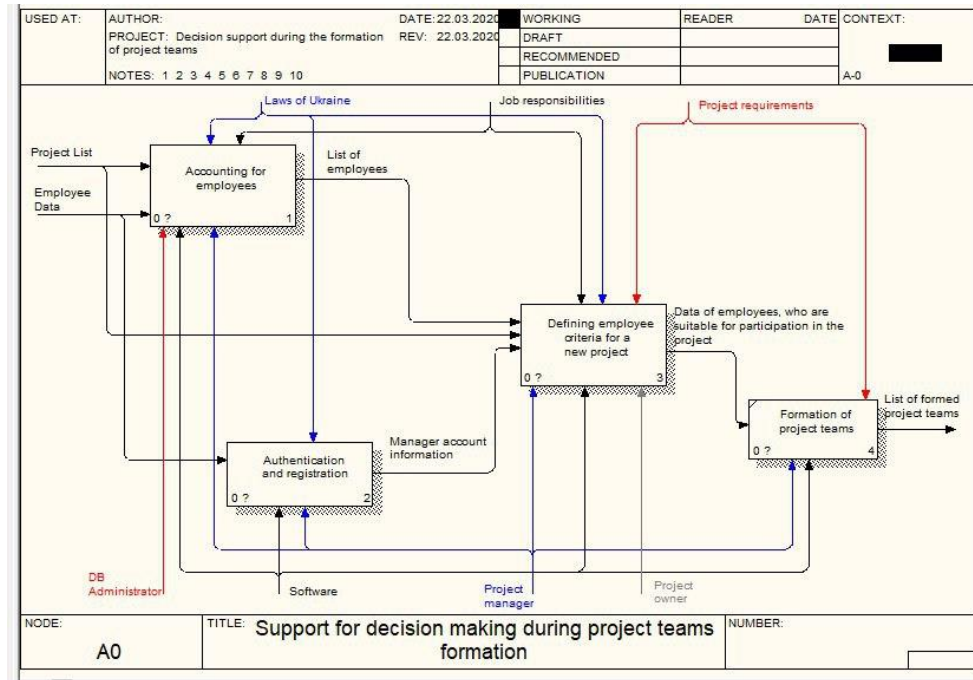
<i>Розробив</i>	<i>Івановська К.А.</i>			<i>Дослідження методів формування проектних команд в ІТ-індустрії</i>	
<i>Перевірив</i>	<i>Овезгельдієв А.О.</i>				
<i>Н.контр.</i>	<i>Овезгельдієв А.О.</i>			<i>ІТІМ-20-1</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

КОНТЕКСТНА ДІАГРАМА



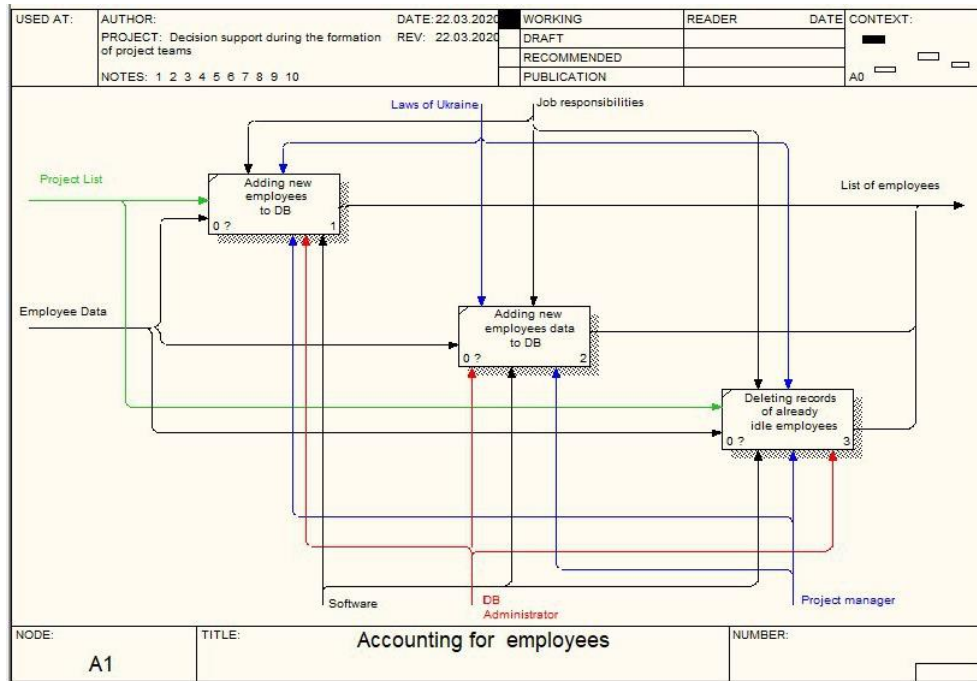
<i>Розробив</i>	<i>Івановська К.А.</i>			<i>Дослідження методів формування проектних команд в ІТ-індустрії</i>	
<i>Перевірів</i>	<i>Овезгельдиєв А.О.</i>				
<i>Н.контр.</i>	<i>Овезгельдиєв А.О.</i>			<i>ІТІМ-20-1</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

ДІАГРАМА ДЕКОМПОЗИЦІЇ ОСНОВНОГО БІЗНЕС-ПРОЦЕСУ КОНТЕКСТНОЇ ДІАГРАМИ



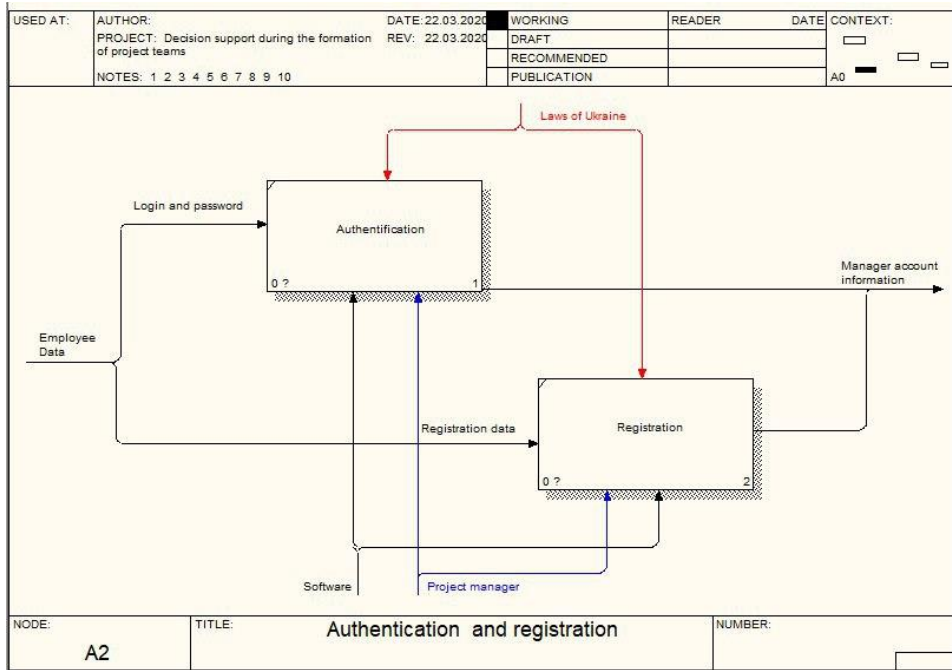
Розробив	Івановська К.А.			Дослідження методів формування проектних команд в ІТ-індустрії	
Перевірів	Овезгельдієв А.О.				
Н.контр.	Овезгельдієв А.О			ІТМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

ДІАГРАМА ДЕКОМПОЗИЦІЇ БІЗНЕС-ПРОЦЕСУ «ACCOUNTING FOR EMPLOYEES»



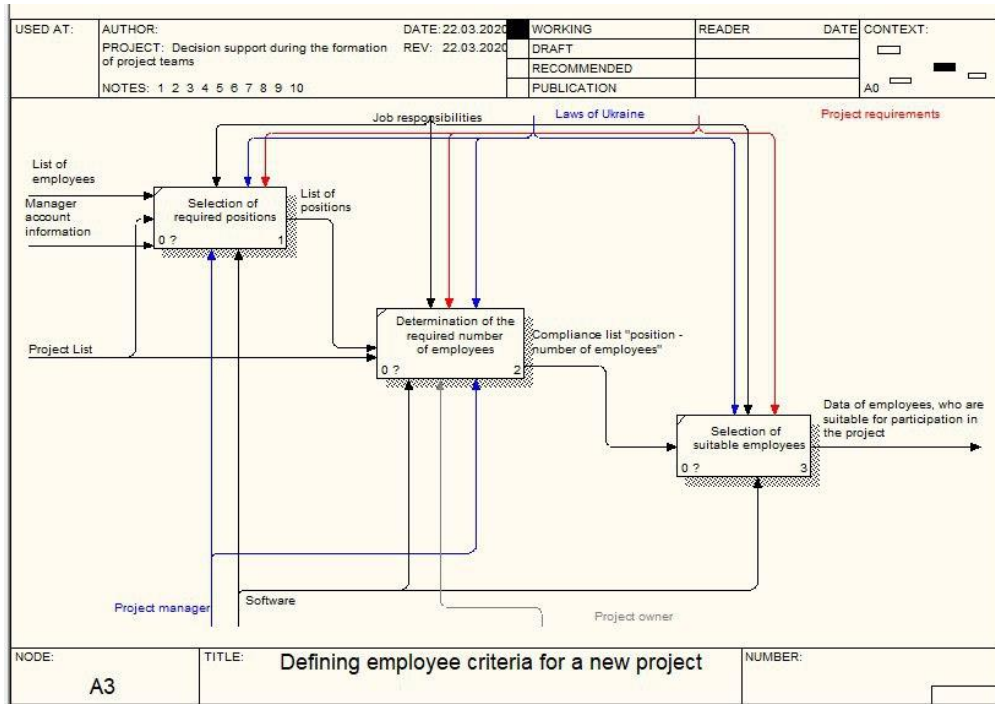
Розробив	Івановська К.А.			Дослідження методів формування проектних команд в ІТ-індустрії	
Перевірів	Овезгельдієв А.О.				
Н.контр.	Овезгельдієв А.О			ІТІМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

ДІАГРАМА ДЕКОМПОЗИЦІЇ БІЗНЕС-ПРОЦЕСУ «AUTHENTICATION AND REGISTRATION»



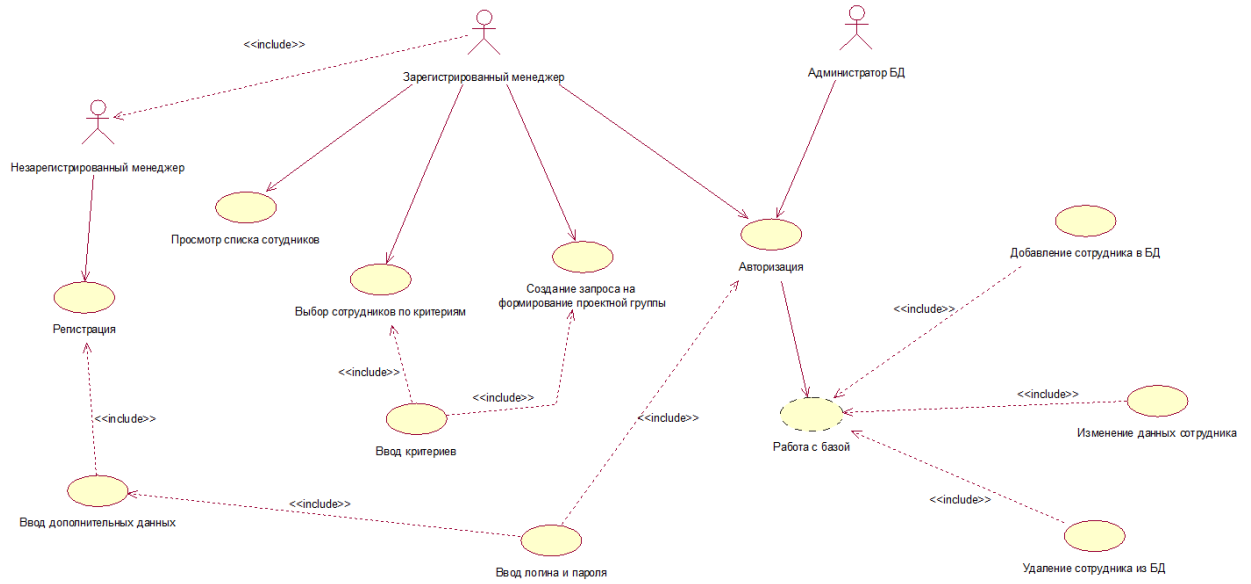
Розробив	Івановська К.А.			<i>Дослідження методів формування проектних команд в ІТ-індустрії</i>
Перевірів	Овезгельдієв А.О.			
Н.контр.	Овезгельдієв А.О		ІТІМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.		СТ	Аркушів 1

ДІАГРАМА ДЕКОМПОЗИЦІЇ БІЗНЕС-ПРОЦЕСУ «DEFINING EMPLOYEE CRITERIA FOR A NEW PROJECT»



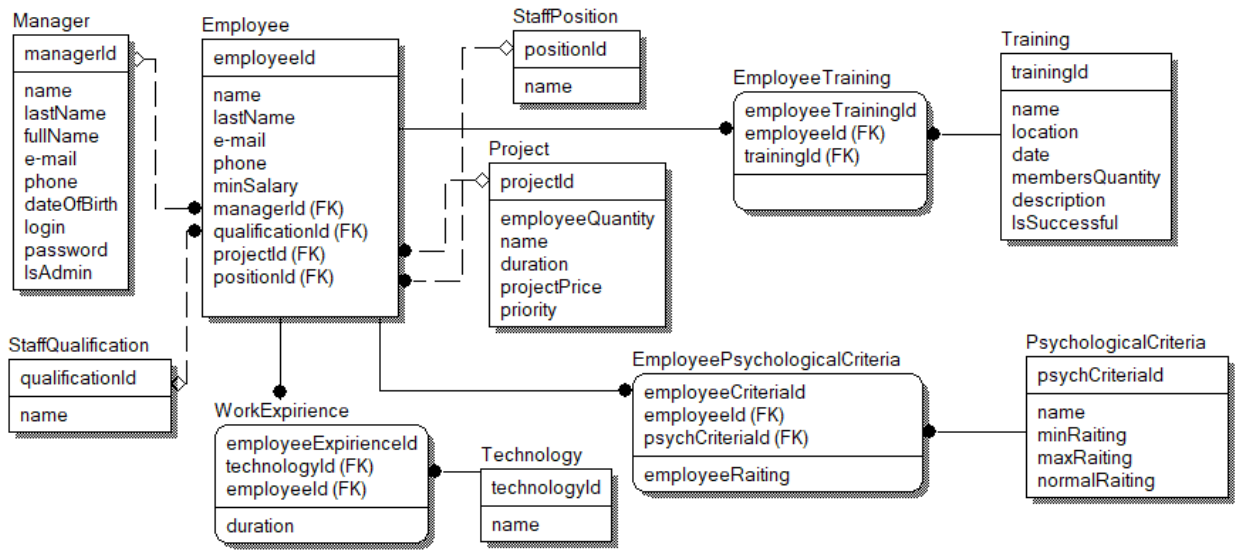
Розробив	Івановська К.А.			<i>Дослідження методів формування проектних команд в ІТ-індустрії</i>	
Перевірів	Овезгельдієв А.О.			<i>ІТІМ-20-1</i>	<i>Аркуш 1</i>
Н.контр.	Овезгельдієв А.О			<i>СТ</i>	<i>Аркушів 1</i>
Затвердив	Гребеннік І.В.				

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ СИСТЕМИ



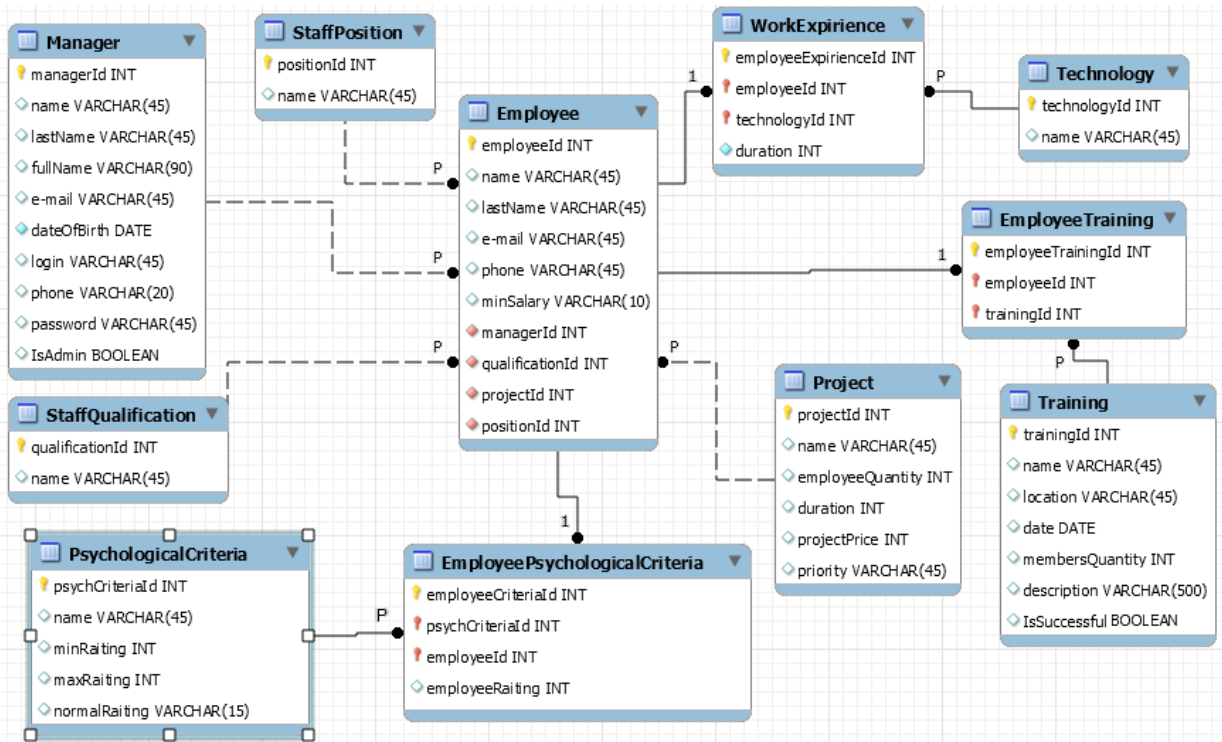
Розробив	Івановська К.А.			Дослідження методів формування проектних команд в ІТ-індустрії	
Перевірів	Овезгельдиев А.О.				
Н.контр.	Овезгельдиев А.О			ІТІМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

СХЕМА ЛОГІЧНОЇ МОДЕЛІ ДАНИХ ДЛЯ БД



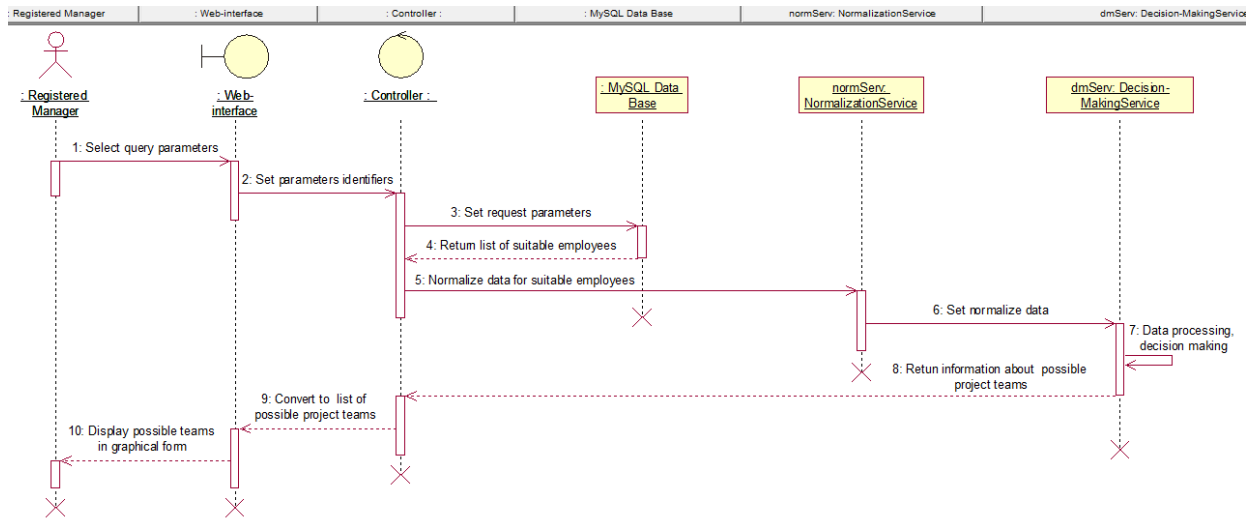
Розробив	Івановська К.А.			Дослідження методів формування проектних команд в ІТ-індустрії	
Перевірів	Овезгельдієв А.О.				
Н.контр.	Овезгельдієв А.О.			ІТІМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

СХЕМА ФІЗИЧНОЇ МОДЕЛІ ДАНИХ ДЛЯ БД



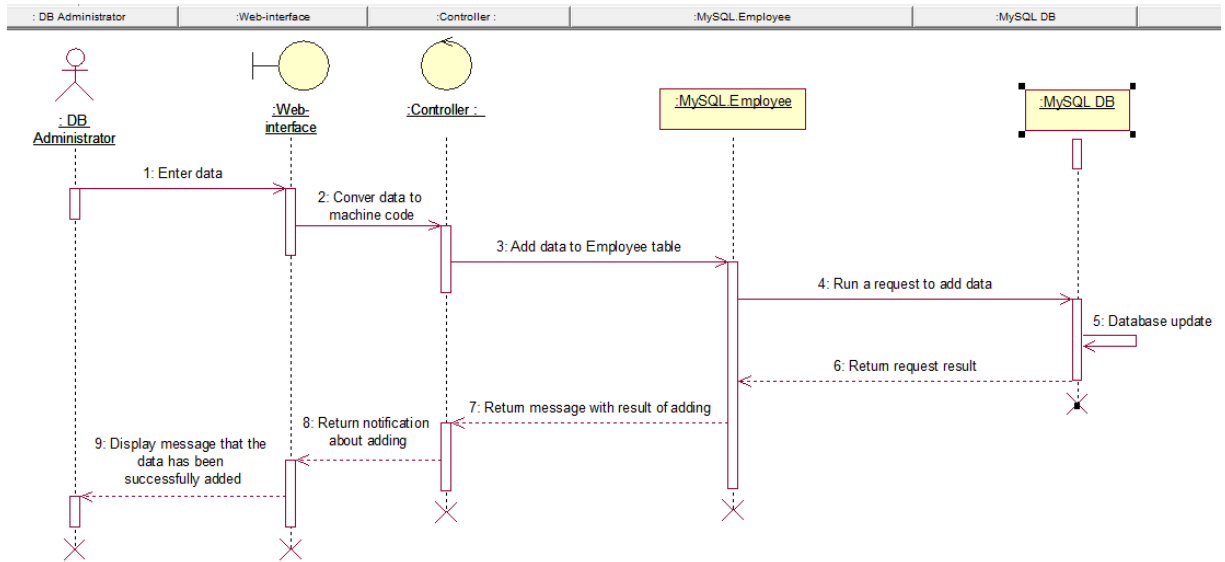
Розробив	Івановська К.А.			Дослідження методів формування проектних команд в ІТ-індустрії	
Перевірів	Овезгельдієв А.О.				
Н.контр.	Овезгельдієв А.О.			ІТІМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

ДІАГРАМА ПОСЛІДОВНОСТЕЙ ДЛЯ ПРЕЦЕДЕНТУ «ФОРМУВАННЯ КОМАНД ЗА ПЕВНИМИ КРИТЕРІЯМИ»



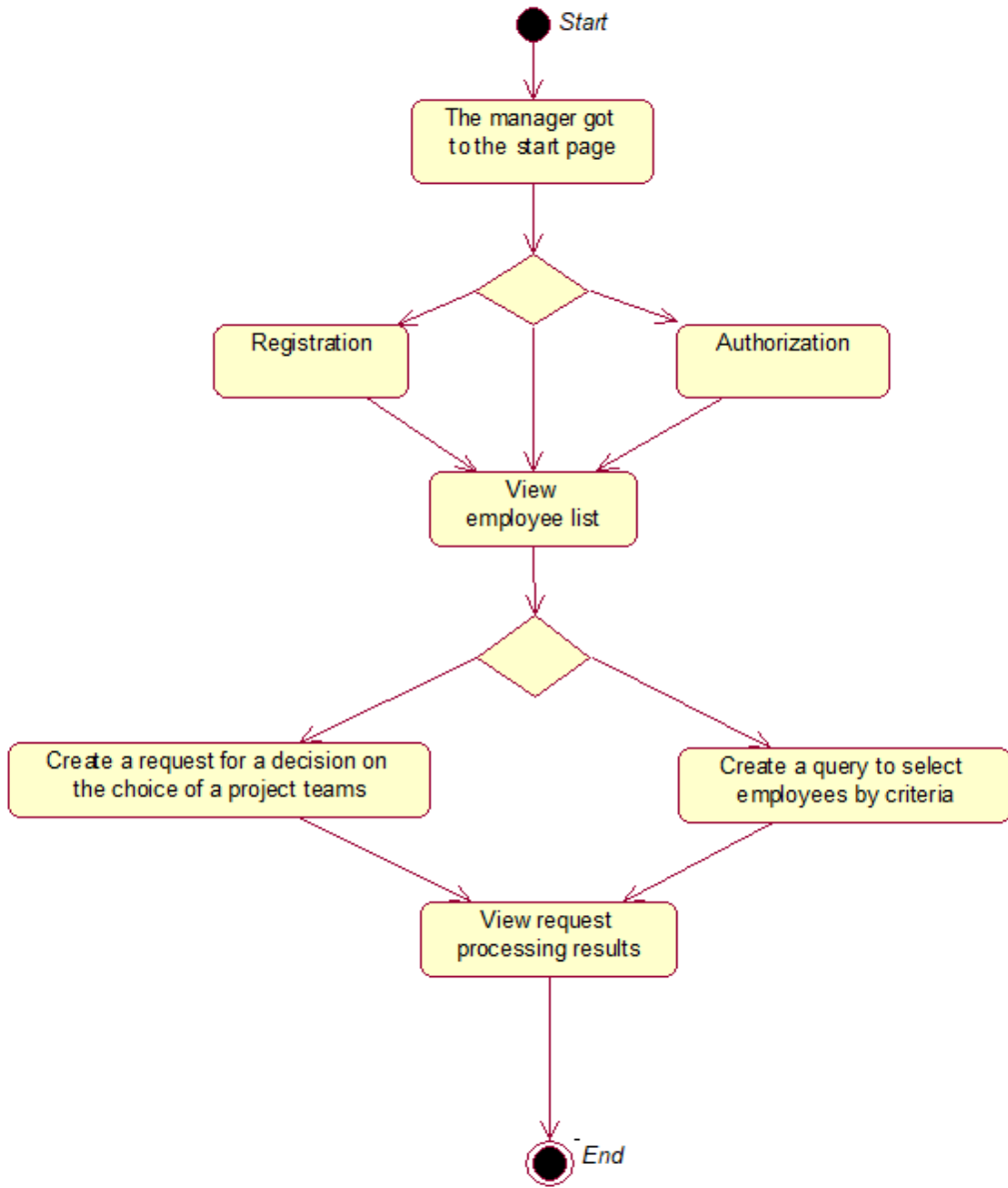
Розробив	Івановська К.А.			Дослідження методів формування проектних команд в ІТ-індустрії	
Перевірів	Овезгельдиєв А.О.				
Н.контр.	Овезгельдиєв А.О.			ІТІМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

ДІАГРАМА ПОСЛІДОВНОСТЕЙ ДЛЯ ПРЕЦЕДЕНТУ «ДОДАВАННЯ ІНФОРМАЦІЇ ПРО ПРАЦІВНИКА ДО БАЗИ ДАНИХ»



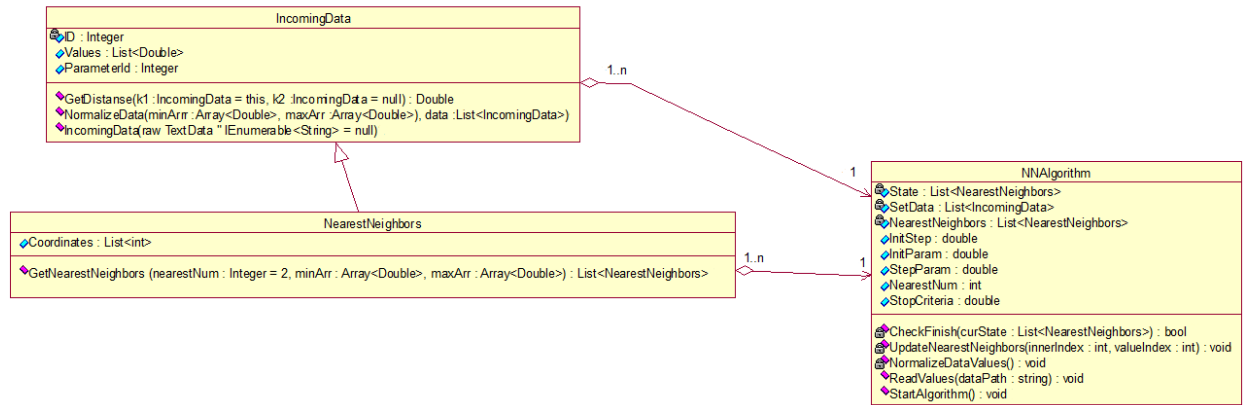
Розробив	Івановська К.А.			Дослідження методів формування проектних команд в ІТ-індустрії	
Перевірів	Овезгельдієв А.О.				
Н.контр.	Овезгельдієв А.О			ІТІМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

ДІАГРАМА СТАНІВ СИСТЕМИ ПІД ЧАС РОБОТИ ПРОЕКТНОГО МЕНЕДЖЕРА



<i>Розробив</i>	<i>Івановська К.А.</i>			<i>Дослідження методів формування проектних команд в ІТ-індустрії</i>	
<i>Перевірів</i>	<i>Овезгельдієв А.О.</i>				
<i>Н.контр.</i>	<i>Овезгельдієв А.О.</i>			<i>ІТМ-20-1</i>	<i>Аркуш 1</i>
<i>Затвердив</i>	<i>Гребеннік І.В.</i>			<i>СТ</i>	<i>Аркушів 1</i>

ДІАГРАМА КЛАСІВ ПІДСИСТЕМИ АНАЛІЗУ ДАНИХ



Розробив	Івановська К.А.			Дослідження методів формування проектних команд в ІТ-індустрії	
Перевірів	Овезгельдієв А.О.				
Н.контр.	Овезгельдієв А.О.			ІТІМ-20-1	Аркуш 1
Затвердив	Гребеннік І.В.			СТ	Аркушів 1

ДОДАТОК Б

ТЕКСТ ПРОГРАМИ

ГЮИК.506160.025 – 01 12 01

(позначення документу)

ЗАТВЕРДЖЕНО

ГЮИК. 506160.025 – 01 12 01 – ЛЗ

Дослідження методів формування проектних команд в ІТ-індустрії

Текст програми

ГЮИК.506160.025 – 01 12 01 – ЛЗ

ЛИСТІВ 11

2021 р.

Міністерство освіти і науки України
Харківський Національний Університет Радіоелектроніки

«ЗАТВЕРДЖУЮ»
керівник кваліфікаційної роботи
проф. Овезгельдієв А.О.

Дослідження методів формування проектних команд в ІТ-індустрії

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ

ГЮИК. 506160.025 – 01 12 01 – ЛЗ

РОЗРОБИВ:
ст. гр. ІТПм-20-1
Івановська К.А.

2021 р.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace kNN
{
    class NNAlgorithm
    {
        int kNN;
        int totalTrainset;
        List<Customer> trainset;
        List<Customer> customerset;

        Distance [] distances;

        int maxAge;
        int maxIncome;
        int maxParam;

        public NNAlgorithm( int k, List <Customer> train, List <Customer> customer )
        {
            this.kNN = k;//k neighbor

            this.trainset = train;//trainset
            this.customerset = customer;//customer
            this.totalTrainset = train.Count;//total of customer

            distances = new Distance [this.totalTrainset];

            //get max value of each column need to normalize
            MaximumValue maximum = new MaximumValue(train);
            maximum.findAllMax();

            maxAge = maximum.getMaxAge();
            maxIncome = maximum.getMaxIncome();
            maxParam = maximum.getMaxParam();
        }

        public void setResponse(Customer cus)
        {
            //normalize cus
            Normalize ncus = new Normalize(cus, maxAge, maxIncome, maxParam);

            //calculate all distances
            for (int i = 0; i < this.totalTrainset; i++)
            {
                distances[i] = new Distance();
            }
        }
    }
}
```

```

distances[i].distance = 0;
distances[i].index = i;

//normalize element
Normalize tmp = new Normalize(this.trainset[i], maxAge, maxIncome, maxParam);

//distance between two age normalized
distances[i].distance = distances[i].distance + getDistance( ncus.age , tmp.age);

//distance between two gender
distances[i].distance = distances[i].distance + getDistance(cus.getGender(),
this.trainset[i].getGender());

//distance between two incoming normalized
distances[i].distance = distances[i].distance + getDistance(ncus.incoming, tmp.incoming);

//distance between two number of card normalized
distances[i].distance = distances[i].distance + getDistance( ncus.param, tmp.param);

} // end loop

//sort
for (int i = 0; i < totalTrainset - 1; i++)
{
    for (int j = i + 1; j < totalTrainset; j++)
    {
        if (distances[i].distance > distances[j].distance)
        {
            Distance tmp = distances[i];
            distances[i] = distances[j];
            distances[j] = tmp;
        } //swap
    } //end j loop
} //end i loop

//select k nearest neighbor
int yesCount = 0;
int noCount = 0;

for (int i = 0; i < kNN; i++)
{
    Customer tmp = trainset[ distances[i].index ];
    if (tmp.getResponse() == 0)
    {
        noCount = noCount + 1;
    }
}

```

```
else if ( tmp.getResponse() == 1 )
{
    yesCount = yesCount + 1;
}
}

//set response value for unknown customer
if (yesCount > noCount)
{
    cus.setResponse(1);
}
else if (yesCount < noCount)
{
    cus.setResponse(0);
}
}

public float getDistance(float a, float b)
{
    return (a - b) * (a - b);
}

public void runkNN()
{
    for (int i = 0; i < this.customerset.Count; i++)
    {
        setResponse(this.customerset[i]);
    }
}

public List<Customer> getCustomerList()
{
    return this.customerset;
}
}
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace kNN
{
    class Customer
    {
        private string name;
        private int age;
        private int gender;
        private int incoming;
        private int param;
        private int response;

        public Customer()
        {

        }

        public void setName(string name)
        {
            this.name = name;
        }

        public string getName()
        {
            return this.name;
        }

        public void setAge(int age)
        {
            this.age = age;
        }

        public int getAge()
        {
            return this.age;
        }

        public void setGender(int gender)
        {
            this.gender = gender;
        }

        public int getGender()
        {
```

```
        return this.gender;
    }

    public void setIncoming(int income)
    {
        this.incoming = income;
    }

    public int getIncoming()
    {
        return this.incoming;
    }

    public void setParam(int param)
    {
        this.param = param;
    }

    public int getParam()
    {
        return this.param;
    }

    public void setResponse(int res)
    {
        this.response = res;
    }

    public int getResponse()
    {
        return this.response;
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace kNN
{
    class Distance
    {
        public int index;
        public float distance;
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace kNN
{
    class
    {
        int maxAge = 0;
        int maxIncome = 0;
        int maxParam = 0;

        List<Customer> lCustomer = new List<Customer>();

        public MaximumValue( List <Customer> lc )
        {
            this.lCustomer = lc;
        }

        public void findAllMax()
        {
            for (int i = 0; i < this.lCustomer.Count; i++)
            {
                //find maxAge
                if (this.lCustomer[i].getAge() > maxAge)
                {
                    maxAge = this.lCustomer[i].getAge();
                }

                //find maxIncome
                if (this.lCustomer[i].getIncoming() > maxIncome)
                {
                    maxIncome = this.lCustomer[i].getIncoming();
                }

                //find maxParam
                if (this.lCustomer[i].getParam() > maxParam)
                {
                    maxParam = this.lCustomer[i].getParam();
                }
            }
        }

        public int getMaxAge()
        {
            return this.maxAge;
        }
    }
}
```

```
public int getMaxIncome()
{
    return this.maxIncome;
}

public int getMaxParam()
{
    return this.maxParam;
}
}
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace kNN
{
    class Normalize
    {
        public float age;
        public float incoming;
        public float param;

        public Normalize( Customer cus, int maxAge, int maxIncome, int maxParam )
        {
            age = (float) cus.getAge() / (float)maxAge;
            incoming = (float) cus.getIncoming() / (float)maxIncome;
            param = (float) cus.getParam() / (float)maxParam;
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace kNN
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
```

```

static void Main()
{
    Application.EnableVisualStyles();
    Application.SetCompatibleTextRenderingDefault(false);
    Application.Run(new frmMain());
}
}
}

public List<Customer> getCustomerList()
{
    return this.customerset;
}
}
}

public class StringProcessor : IStringProcessor
{
    private readonly SvmModelSettings _settings;

    public StringProcessor(SvmModelSettings settings)
    {
        if (settings == null) throw new ArgumentNullException("settings");
        _settings = settings;
    }

    public string Normalize(string text)
    {
        var str = text.Replace('\n', ' ');
        return _settings.IgnoredPatterns.Aggregate(str,
            (current, pattern) => Regex.Replace(current, pattern, "", RegexOptions.IgnoreCase));
    }

    public IEnumerable<string> GetWords(string text)
    {
        return
            text.Split(_settings.Delimiters, StringSplitOptions.RemoveEmptyEntries)
                .Select(w => w.ToLower())
                .Where(w => !_settings.IgnoredWords.Contains(w));
    }
}

public enum Emotion
{
    PositiveOrNeutral = 1,
    Negative = -1
}

public class ClassifiedItem //Classified sample
{
    public Emotion Emotion { get; set; } //Sample tonality
}

```

```

public string Text { get; set; } //
}
public class SvmModelBuilder //The class is for creating a model
{
private readonly IStringProcessor _stringProcessor;

public SvmModelBuilder(IStringProcessor stringProcessor)
{
_stringProcessor = stringProcessor;
}

public virtual SvmTrainedModel Train(IEnumerable<ClassifiedItem> items)
{
if (!items.Any())
throw new InvalidOperationException("No data to train the model");

var emotionArr = new List<double>();
var vocabularySet = new HashSet<string>();
var linewords = new List<string[]>();

foreach (var classifiedItem in items) //build a dictionary of words from the full input set
{
var words = GetWords(classifiedItem.Text).ToArray();
vocabularySet.UnionWith(words);
linewords.Add(words);
emotionArr.Add((double)classifiedItem.Emotion);
}

var vocabulary = new Dictionary<string, int>(vocabularySet.Count);
var sorted = vocabularySet.OrderBy(w => w).ToArray();

// sort the words in the dictionary and add indices
// so that later the original string can be turned into a vector of features
for (var i = 0; i < sorted.Length; i++)
{
vocabulary.Add(sorted[i], i);
}

var problem = CreateProblem(linewords, emotionArr, vocabulary);

// get model using library classes libsvm.net
var model = new C_SVC(problem, KernelHelper.LinearKernel(), 1);

// return the model ready for classification
return new SvmTrainedModel(model, vocabulary, _stringProcessor);
}

private static svm_problem CreateProblem(IReadOnlyCollection<string[]> lines,
List<double> emotionArr, IReadOnlyDictionary<string, int> vocabulary)
{
return new svm_problem()
{

```

```

l = lines.Count, //total number of classified comments

//turns strings into feature vectors
x = lines.Select(line => NodeUtils.CreateNode(line, vocabulary).ToArray()).ToArray(),

y = emotionArr.ToArray() //comment ratings vector
};
}

//returns a list of words from the string, cleared of "noise"
protected virtual IEnumerable<string> GetWords(string text)
{
    var normalized = _stringProcessor.Normalize(text);
    return _stringProcessor.GetWords(normalized);
}
}
public class SvmTrainedModel
{
    private readonly SVM _model;
    private readonly IReadOnlyDictionary<string, int> _vocabulary;
    private readonly IStringProcessor _stringProcessor;

    public SvmTrainedModel(SVM model, IReadOnlyDictionary<string, int> vocabulary,
IStringProcessor stringProcessor)
    {
        if (model == null) throw new ArgumentNullException("model");
        if (vocabulary == null) throw new ArgumentNullException("vocabulary");
        if (stringProcessor == null) throw new ArgumentNullException("stringProcessor");
        _model = model;
        _vocabulary = vocabulary;
        _stringProcessor = stringProcessor;
    }

    public Emotion Classify(string text) //performs string classification
    {
        return (Emotion)Model.Predict(NodeUtils.CreateNode(GetWords(text).ToArray(),
Vocabulary).ToArray());
    }

    //returns a list of words from a string, stripped of "noise"
protected virtual IEnumerable<string> GetWords(string text)
{
    var normalized = StringProcessor.Normalize(text);
    return StringProcessor.GetWords(normalized);
}
}

```

ДОДАТОК В

ВІДОМІСТЬ АТЕСТАЦІЙНОЇ РОБОТИ

ГЮИК.506160.025

(позначення документа)

№	Позначення				Найменування	Дод. відомості	
					Текстові документи		
1.	ГЮИК.506160.025 ПЗ				Пояснювальна записка	72 с.	
2.	ГЮИК.506160.025 – 01 12 01				Текст програми	11 с.	
					Графічні матеріали		
3.					Загальна схема алгоритму k-ближніх сусідів.	1 аркуш	
4.					Загальна схема алгоритму методу опорних векторів	1 аркуш	
5.					Контекстна діаграма	1 аркуш	
6.					Діаграма декомпозиції основного бізнес-процесу контекстної діаграми	1 аркуш	
7.					Діаграма декомпозиції бізнес-процесу «Accounting for employees»	1 аркуш	
8.					Діаграма декомпозиції бізнес-процесу «Authentication and registration»	1 аркуш	
9.					Діаграма декомпозиції бізнес-процесу «Defining employee criteria for a new project»	1 аркуш	
10.					Діаграма варіантів використання системи	1 аркуш	
11.					Схема логічної моделі даних для БД	1 аркуш	
12.					Схема фізичної моделі даних для БД	1 аркуш	
13.					Діаграма послідовностей для прецеденту «Формування команд за певними критеріями»	1 аркуш	
					ГЮИК.506160.025 ДЗ		
Змін.	Арк.	№ докум	Підпис	Дата	Дослідження методів формування проектних команд в ІТ-індустрії	Аркуш	Аркушів
Розробив		Івановська К.А.				1	2
Перевірів		Овезгельдієв А.О.				ХНУРЕ Кафедра СТ	
Н. Контр.		Овезгельдієв А.О.					
Затвердив		Гребеннік І.В.					

