

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
„ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”
КАФЕДРА „СИСТЕМНОГО ПРОЕКТУВАННЯ”

ЛАБОРАТОРНА РОБОТА №2

з курсу: “Проектування інформаційних систем ”

на тему: “Розробка опису передпроектної документації (Software Requirement Specifications) ”

Виконав: студент IV курсу
групи ДА-72
Охота Д.Л.
Варіант №21

Мета роботи:

Вивчити основні етапи створення перепроектної документації (SRS).

Задача:

Використати приклад SRS для створення перепроектної документації згідно з індивідуальною темою для виконання лабораторних робіт.

Завдання:

1. Вивчити вимоги до передпроектної документації.
2. Скласти опис передпроектної документації для об'єкта проектування.
3. Скласти 5-7 приймальних тестів для ПО об'єкта проектування.
4. Оформити технічне завдання згідно опис передпроектної документації (використовувати рекомендації IEEE 830).

Software Requirements Specification

Version 1.0

November 02, 2020

City alarm system

Okhota Denys

Revision History

| Date | Description | Author | Comments |
|------------|-------------|-----------|----------|
| 02.11.2020 | Version 1.0 | Okhota D. | |
| 12.11.2020 | Version 1.1 | Okhota D. | |
| | | | |
| | | | |

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

| Signature | Printed Name | Title | Date |
|-----------|--------------|-------|------------|
| | Okhota D. | | 02.11.2020 |
| | | | |
| | | | |

1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the City Alarm System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

1.2. Scope of Project

This software system will be a City Alarm System for a townspeople. System is designed to inform users of some situation that happened not so far, by providing news feed and notification system.

More specifically, this system is designed to create post about emergency situation, traffic and meetings. User will be able watch posts, get notifications and add new information. All the information, users and administrators will be stored in database.

1.3. Glossary

| Term | Definition |
|-------------------------------------|---|
| Post | The report that is tracked by the system; it is a narrative that is planned to be posted to the public website. |
| Author | Person submitting an post. |
| Database | Collection of all the information monitored by this system. |
| Administrator | Person who receives post, sends posts for review, and makes final judgments for publications. |
| Field | A cell within a form. |
| Reader | Anyone visiting the site to read posts. |
| Reviewer | A person that examines an post and has the ability to recommend approval of the article for publication or to request that changes be made in the post. |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Reader or Author. |

1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications*. IEEE Computer Society, 1998.

Методичні вказівки до виконання лабораторних робіт для студентів напряму підготовки 6.050101 «Комп'ютерні науки», спеціальностей 8.05010102 «Інформаційні технології проектування» та 8.05010103 «Системне проектування» денної та заочної форм навчання / Укл. Харченко К.В. – К. : НТУУ «КПІ», 2019 р. – 14 с.

1.5 Overview

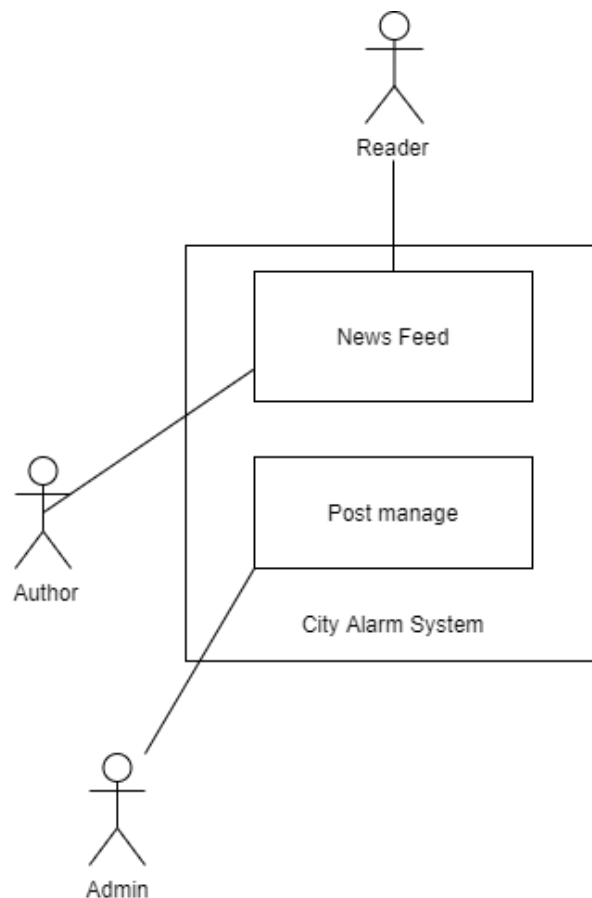
The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. General Description

2.1 Product Perspective



The City Alarm System has only three type of users – Author, Reader, Admin. Admin communicates with database via web app and changes information which is stored in system. Reviewer communicates with Admin and request to edit posts or delete them. Author, Reader communicates with app via Internet and reads or adds new information. All they needs - is any device connected to the internet.

2.2 Product Functions

Product will contain 3 main functions. Notify users about emergency situation, searching lost things, post news about city or town .

2.3 User Characteristics

System divides into 2 categories:

Management staff (Admin)

Users (Author, Reader)

Management staff decides to post certain news or discard it.

Users post news and read them.

2.4 General Constraints

System will be written in Python programming language with the aid of Django web frame-work. HTML, CSS, JS for frontend part.

Users need to have good internet connection ,email and password to get authorized.

Also possible phone version for such OS:

Android

IOS

Windows

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

All pages of the UI should employ an organizational method that separates the major sections of the application for easy access. The user should be able to determine the sections of the UI currently being viewed using visual cues, such as bold fonts, arrows, and highlighting. The application's major sections should be available from any page of the application and include the following pages: News, Add, Administration, Settings, and Log in/out. The user must be presented with a login page when accessing any page, except a landing or front page, as an anonymous user. Administrative tasks and running queries must have a UI. The UI for administrative tasks must included the ability to manage Users, manage posts, manage permissions, manage data templates. All pages should use a consistent visual theme. Pop-ups displaying notifications, error messages, or warnings should use the same theme and template as the rest of the application.

3.1.2 Hardware Interfaces

All server-side components must execute on server-class computers. All client-side components must execute on workstation-class and personal-class computers.

3.1.3 Software Interfaces

Service: All browsers besides IE6 and less.

Server: Python3.7 with all required packages + PostgreSQL + Windows10

3.1.4 Communications Interfaces

System shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

3.2 Functional Requirements

This section describes major functional requirements of the system. It is divided into two subsections as wearable device and server.

Login: input user login and password.

Password restoration: input user email address, input security code from email.

Logout: button at the right top corner.

Change user info: form with (name, surname, email, login) fields.

Change user password: form with (old password, new password, repeat password) fields.

Delete account: form with (account password field).

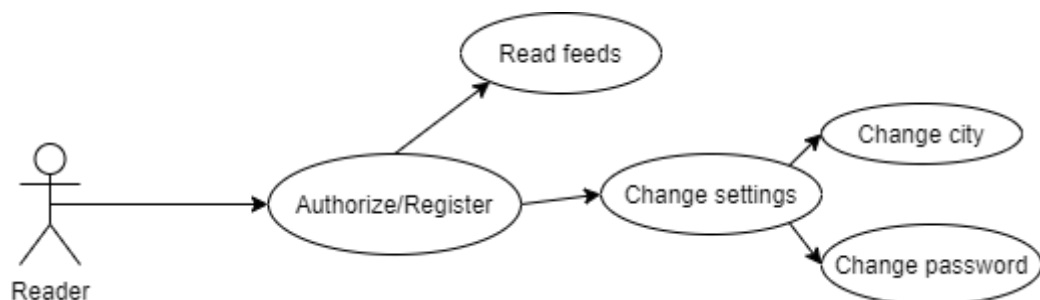
Edit post: form with pre-filled data.

Delete post: button with confirmation alert.

Add post, notification: form with fields.

3.3 Use Cases

3.3.1 Reader Use Case



Use case: Authorize/Register



Brief Description

The reader authorizes or registers.

Initial Step-By-Step Description

1. The Reader enters username and password.
2. The Reader clicks Log in button

In step 1, if the Reader clicks Sign Up button

2. The Reader enters username, password, and re-enter password for verification.
3. The Reader clicks Sign Up button.

Use case: Read feeds



Brief Description

The reader/author reads news feed.

Initial Step-By-Step Description

1. The System displays news.
2. The Reader presses “Read” placeholder in one of them.

Use case: Change settings



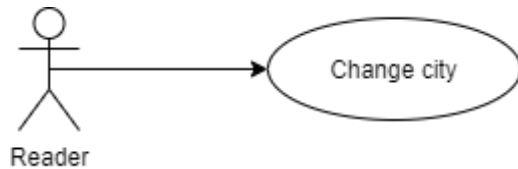
Brief Description

The reader wants to change account settings.

Initial Step-By-Step Description

1. Reader/Author presses “Settings” icon
2. System displays a list of settings.

Use case: Change city



Brief Description

The reader wants to change city .

Initial Step-By-Step Description

1. Reader/Author presses “Settings” icon
2. System displays a list of settings.
3. User presses city`s name
4. And choose or enters new one
5. And presses “Save” button

Use case: Change password



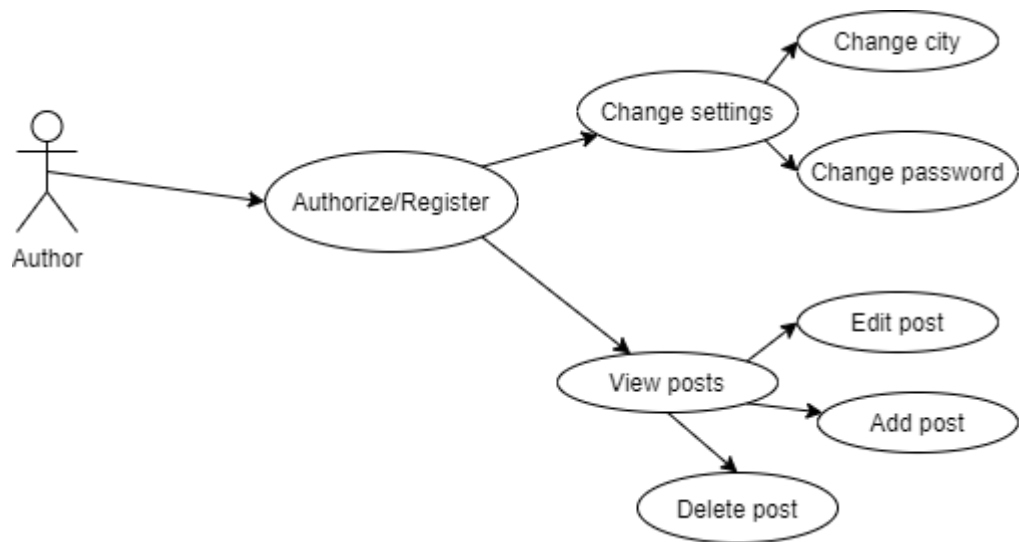
Brief Description

The reader wants to change password .

Initial Step-By-Step Description

1. Reader/Author presses “Settings” icon
2. System displays a list of settings.
3. User presses “Privacy Settings”
4. Enters old password
5. And enters new password
6. Confirms it by entering once more
7. And presses “Save” button

3.3.2 Author Use Case



Use case: Authorize/Register



Brief Description

The author authorizes or registers.

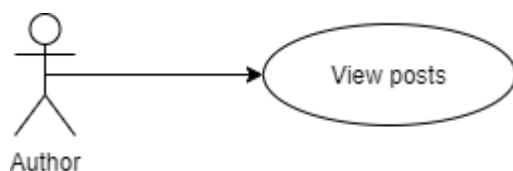
Initial Step-By-Step Description

3. The Reader enters username and password.
4. The Reader clicks Log in button

In step 1, if the Reader clicks Sign Up button

2. The Reader enters username, password, and re-enter password for verification.
3. The Reader clicks Sign Up button.

Use case: View posts



Brief Description

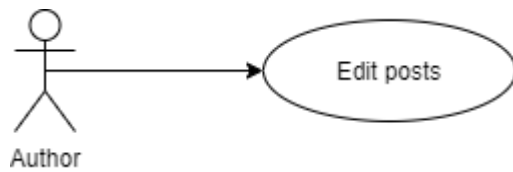
The author can see all created posts.

Initial Step-By-Step Description

1. Author presses "Posts" placeholder

2. System displays a list of Author's posts.

Use case: Edit post



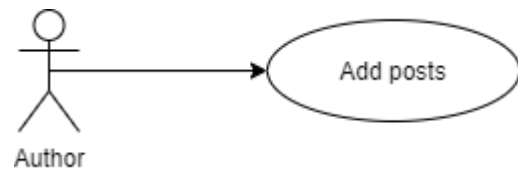
Brief Description

The author wants to edit post.

Initial Step-By-Step Description

1. The System displays author's posts list.
2. The Author executes long press of the post that wants to edit.
3. Edit Pop up is displayed.
4. Then makes changes.
5. Then clicks "Save" button.

Use case: Add post



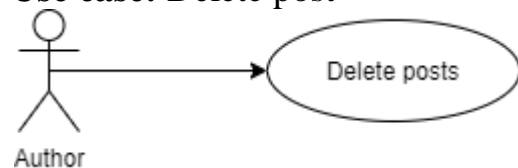
Brief Description

The author wants to add post.

Initial Step-By-Step Description

1. The System displays author's posts list.
2. The Author executes press on add button.
3. Add Pop up is displayed.
4. Then fills fields.
5. Then clicks "Save" button.

Use case: Delete post



Brief Description

The author wants to delete post.

Initial Step-By-Step Description

1. The System displays author's posts list.
2. The Reader executes long press of the post that wants to delete.
3. Delete Pop up is displayed.
4. Then clicks "Agree" button.

Use case: Change settings



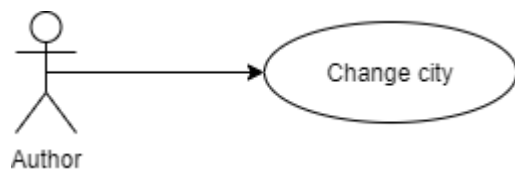
Brief Description

The author wants to change account settings.

Initial Step-By-Step Description

3. Author presses "Settings" icon
4. System displays a list of settings.

Use case: Change city



Brief Description

The author wants to change city .

Initial Step-By-Step Description

6. Author presses "Settings" icon
7. System displays a list of settings.
8. User presses city's name
9. And choose or enters new one
10. And presses "Save" button

Use case: Change password



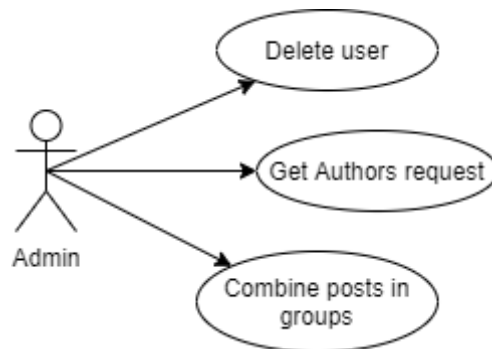
Brief Description

The author wants to change password .

Initial Step-By-Step Description

8. Author presses “Settings” icon
9. System displays a list of settings.
10. User presses “Privacy Settings”
11. Enters old password
12. And enters new password
13. Confirms it by entering once more
14. And presses “Save” button

3.3.2 Admin Use Case



Use case: Get Reviewers request



Brief Description

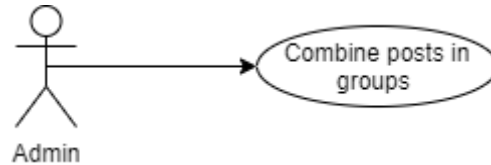
Admin receives request from Reviewer add decides next steps in post lifecycle.

Initial Step-By-Step Description

1. The System displays request.
2. The Admin executes press of the post that reviewr wants to edit/delete.
3. Pop up is displayed.

4. Admin decides what to do
5. Then clicks “Save ” button.

Use case: Get Authors request



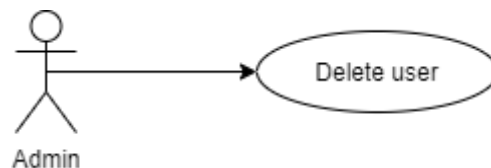
Brief Description

Admin see a huge amount of similar posts and combines them in one.

Initial Step-By-Step Description

1. The System displays list of posts.
2. The Admin executes long press of the posts that admin wants to combine.
3. Pop up is displayed.
4. Then clicks “Save ” button.

Use case: Delete user



Brief Description

Admin sees spamming user and decides to ban him.

Initial Step-By-Step Description

1. The System displays list of posts.
2. The Admin executes long press of the posts than admin picks author.
3. Pop up is displayed.
4. Then clicks “Delete” button.

3.4 Non-Functional Requirements

3.4.1 Performance

Speed of system execution depends mostly on Internet connection, and server processor.

3.4.2 Availability

Mobile app will be available to download from the App Store, Google Play or Windows Store respectively on device you are using.

3.4.3 Security

All your personal information is secured and encrypted. The server on which the Mobile App and Desktop app reside will have its own security to prevent unauthorized *write/delete/edit* access. There is no special protection built into this system other than to provide the admin with *write/edit* access to the DB to publish an article.

3.4.4 Maintainability

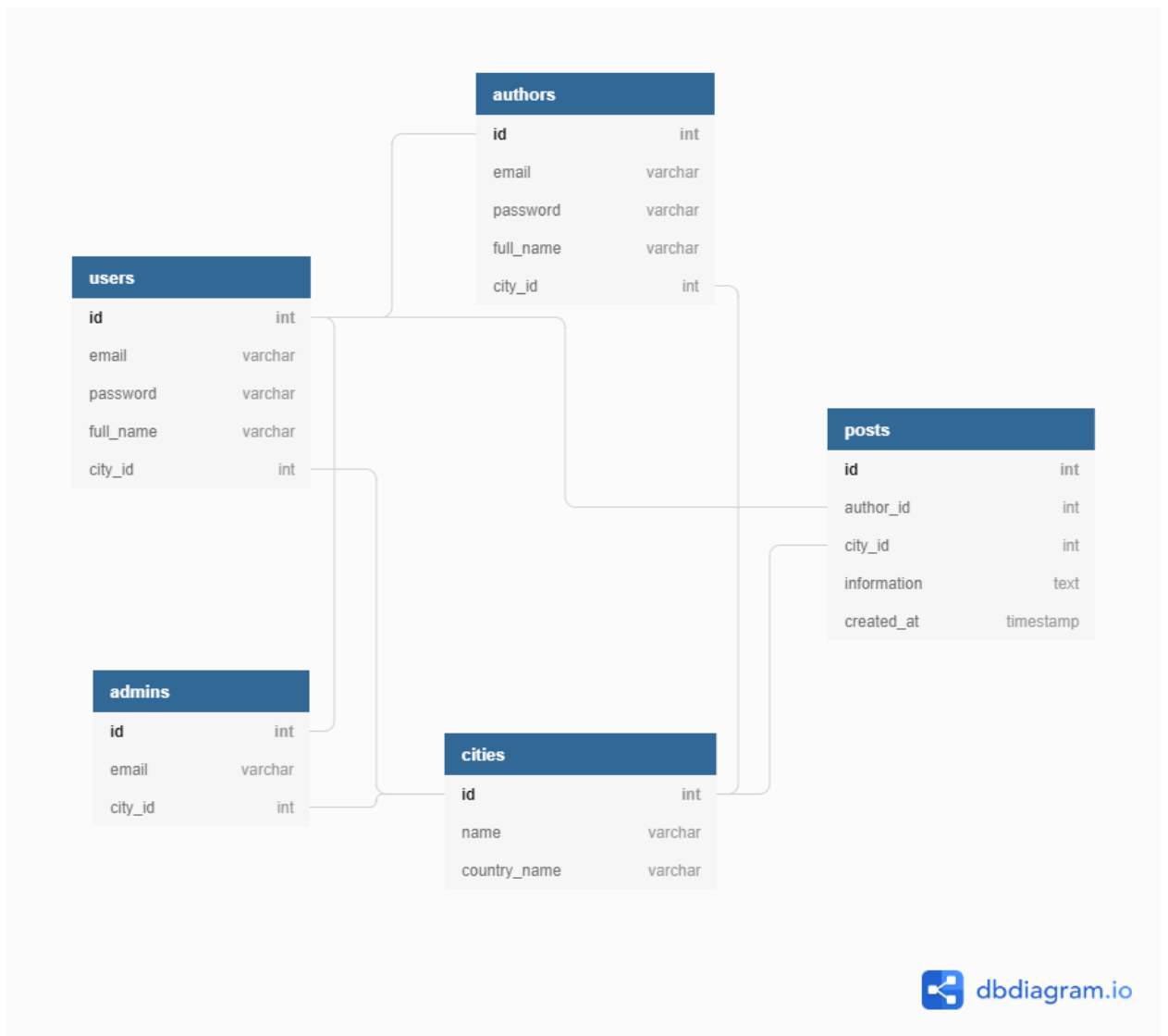
System will have regular updates that fix bugs and performance improvements. Besides that could be added new features.

3.4.5 Portability

System targets wide range of devices:

- Android;
- IOS;
- Windows Phones;

3.5 Logical Database Requirements



4. Tests

| The main goal | Preliminary requirements | Required Actions in order to perform a test | System`s state after test | Actions which reverts system`s state | Comments |
|---------------|--|--|--|--------------------------------------|---------------------------|
| To add a post | The user must be authorized | User press the add button fills fields in pop-up and clicks “Save” | System will get new one record in database | To delete post | |
| To Edit post | The user must be authorized and has a post | long press is executed of the post that | System will update information | Again edit post to previous | Author can edit temporary |

| | | | | | |
|----------------|--|--|---------------------------------------|-----------------------------|---|
| | | wants to edit. Edit Pop up is displayed. Then makes changes. Then clicks “Save” button. | in post | version | post not current. And then admin decides to change current post to temporary. |
| Authorize | The user must have email and password | Enter email and password and press “Log in” button | - | - | |
| To delete post | The user must be authorized and has a post | long press is executed of the post that wants to delete. Confirm deleting by clicking “Delete” button. | System will delete post from database | Admin should revert changes | |
| Read post | The user must be authorized and there should be post | executes press of the post that wants to read. Enter some reaction . | - | - | |