Silesian
University
of Technology

FINAL PROJECT

Interactive security training platform based on CTF concept

Krzysztof Marek DZIEMBAŁA
Student identification number: 293671

Programme: Control, Electronic, and Information Engineering
Specialisation: Informatics

SUPERVISOR
dr inż. Dominik Samociuk
DEPARTMENT Department of Computer Networks and Systems
Faculty of Automatic Control, Electronics and Computer Science

Gliwice 2023

Thesis title

Interactive security training platform based on CTF concept

Abstract

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in English.)

Keywords

(2-5 keywords, separated with commas)

Tytuł pracy

Interaktywna platforma do nauki bezpieczeństwa wykorzystująca zadania typu CTF

Streszczenie

(Thesis abstract – to be copied into an appropriate field during an electronic submission – in Polish.)

Słowa kluczowe

(2-5 keywords, separated by commas, in Polish)

Contents

Chapter 1

Introduction

- introduction into the problem domain
- settling of the problem in the domain
- objective of the thesis
- scope of the thesis
- short description of chapters
- clear description of contribution of the thesis's author – in case of more authors table with enumeration of contribution of authors

Chapter 2

[Problem analysis]

- problem analysis
- state of the art, problem statement
- literature research (all sources in the thesis have to be referenced [2, 1, 3, 4])
- description of existing solutions (also scientific ones, if the problem is scientifically researched), algorithms, location of the thesis in the scientific domain

Mathematical formulae

$$y = \frac{\partial x}{\partial t} \tag{2.1}$$

and single math symbols x and y are typeset in the mathematical mode.

Chapter 3

Requirements and tools

- functional and nonfunctional requirements
- use cases (UML diagrams)
- description of tools
- methodology of design and implementation

3.1 Functional requirements

Functional requirements list functionality available in the system. Each of the requirements contains a description detailing the desired behaviour.

- **Account creation**: Users must be able to register an account in the system. This operation requires username and password submission from an HTML form in a POST request. Registration is allowed only using unique username. If there already exists an account in the system with the same username, the action must be refused. As a result of a successful registration, a document with the username, cryptographic hash of the user's password and a default role `"user"` is inserted into a collection storing user accounts. After the registration succeeds, the user is redirected to the login page.
- **Signing in**: Users must be able to log into their accounts. Username and password must be sent in a POST request as HTML form data. The operation must fail if there is no account in the database with the provided username or if the password hash does not match the one stored in the database. If there has been no failure, a session is created and the user is redirected to the home page.
- **Logging out**: Users must be able to log out of their accounts. It is expected that the user is signed in when they log out. This operation destroys the session (if any) and redirects to the home page.

- **Changing password**: Users must be able to change their account password. New password must be sent in a POST request as HTML form data. User must be logged in in order to change the password. As a result of this operation, user's password hash in the database is updated.
- **Listing categories**: Users must be able to see a list of categories that exist in the system. List of categories must link to category pages.
- **Displaying category**: Users must be able to see category details on a category page. The details should include category name, description and a list of related tasks.

3.1.1 Displaying task

3.1.2 Solving challenge

3.1.3 Answering quiz

3.1.4 Administrator panel

3.1.5 Listing users

3.1.6 Changing user permissions

3.1.7 Deleting user

3.1.8 Creating category

3.1.9 Editing category

3.1.10 Adding tasks

3.1.11 Starting challenges

3.2 Non-functional requirements

3.2.1 Responsiveness

UI should properly scale across different display sizes. It must be mobile-friendly.

3.2.2 Accessibility

There should be no errors in the Accessibility section of a webhint scan.

3.2.3 Visual consistency

A single set of styling rules, such as colours, fonts and icons should be used across whole user interface.

3.2.4 Page load performance

The system should have a score of over 90 in PageSpeed Insights report for mobile.

3.2.5 Compatibility

User interface should work in latest (as of January 2023) versions of Firefox, Chrome and Safari browsers for desktops and mobile devices. Basic system functionality, except for the administrator panel, should be available in browsers with JavaScript disabled.

3.3 Use cases

The use case diagram presented on Fig. 3.1 shows actions offered to actors using the system. There are two actors, with different sets of allowed interactions. The actor *User* represents anyone with access to the system. Users with special account role `"admin"`, which allows them to perform operations related to management of the system.

3.4 Tools

3.4.1 Node.js + Express

3.4.2 MongoDB

3.4.3 Docker

3.4.4 nginx

3.4.5 Bootstrap

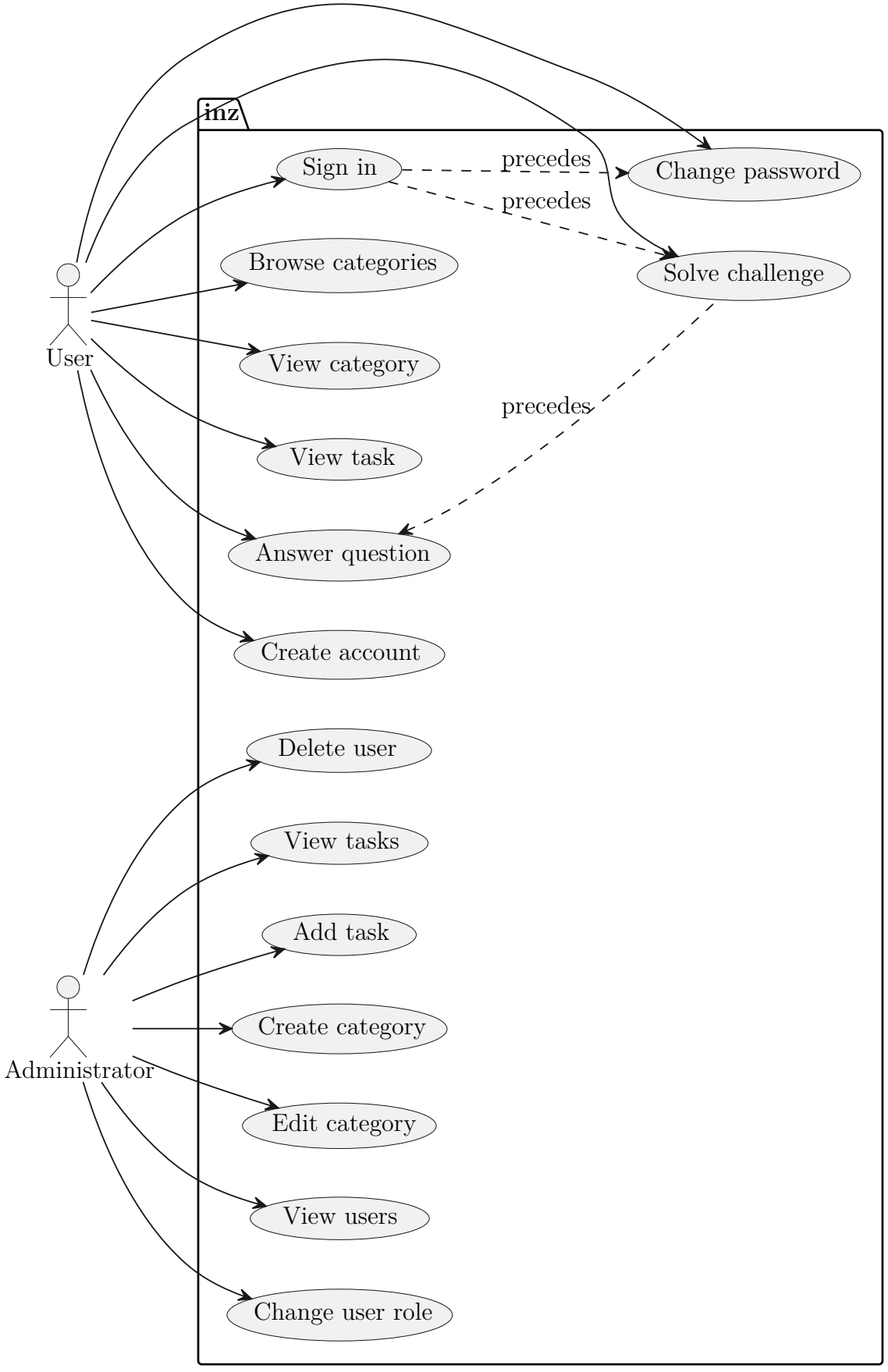

Figure 3.1: Use case diagram

Chapter 4

External specification

- hardware and software requirements
- installation procedure
- activation procedure
- types of users
- user manual
- system administration
- security issues
- example of usage
- working scenarios (with screenshots or output files)

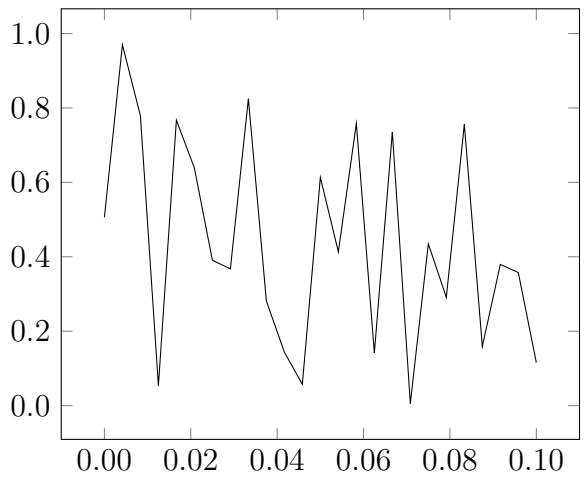

Figure 4.1: Figure caption (below the figure).

Chapter 5

Internal specification

- concept of the system
- system architecture
- description of data structures (and data bases)
- components, modules, libraries, resume of important classes (if used)
- resume of important algorithms (if used)
- details of implementation of selected parts
- applied design patterns
- UML diagrams

Use special environments for inline code, eg `int a;` (package `minted`) . Longer parts of code put in the figure environment, eg. code in Fig. 5.1 . Very long listings–move to an appendix.

```
class test : public basic
{
   public:
     test (int a);
     friend std::ostream operator<<(std::ostream & s,
                                    const test & t);
   protected:
     int _a;

};
```

Figure 5.1: Pseudocode in `minted`.

Chapter 6

Verification and validation

- testing paradigm (eg V model)
- test cases, testing scope (full / partial)
- detected and fixed bugs
- results of experiments (optional)

Table 6.1: A caption of a table is **above** it.

	method						
			alg. 3			alg. 4, $\gamma = 2$	
ζ	alg. 1	alg. 2	$\alpha = 1.5$	$\alpha = 2$	$\alpha = 3$	$\beta = 0.1$	$\beta = -0.1$
0	8.3250	1.45305	7.5791	14.8517	20.0028	1.16396	1.1365
5	0.6111	2.27126	6.9952	13.8560	18.6064	1.18659	1.1630
10	11.6126	2.69218	6.2520	12.5202	16.8278	1.23180	1.2045
15	0.5665	2.95046	5.7753	11.4588	15.4837	1.25131	1.2614
20	15.8728	3.07225	5.3071	10.3935	13.8738	1.25307	1.2217
25	0.9791	3.19034	5.4575	9.9533	13.0721	1.27104	1.2640
30	2.0228	3.27474	5.7461	9.7164	12.2637	1.33404	1.3209
35	13.4210	3.36086	6.6735	10.0442	12.0270	1.35385	1.3059
40	13.2226	3.36420	7.7248	10.4495	12.0379	1.34919	1.2768
45	12.8445	3.47436	8.5539	10.8552	12.2773	1.42303	1.4362
50	12.9245	3.58228	9.2702	11.2183	12.3990	1.40922	1.3724

Chapter 7

Conclusions

- achieved results with regard to objectives of the thesis and requirements
- path of further development (eg functional extension ...)
- encountered difficulties and problems

Bibliography

[1] Name Surname and Name Surname. *Title of a book*. Hong Kong: Publisher, 2017. ISBN: 83-204-3229-9-434.

[2] Name Surname and Name Surname. 'Title of an article in a journal'. In: *Journal Title* 157.8 (2016), pp. 1092–1113.

[3] Name Surname, Name Surname and N. Surname. 'Title of a conference article'. In: *Conference title*. 2006, pp. 5346–5349.

[4] Name Surname, Name Surname and N. Surname. *Title of a web page*. 2021. URL: `http://somewhere/on/the/internet.html` (visited on 30/09/2021).

Appendices

Index of abbreviations and symbols

CTF	Capture The Flag
UI	User Interface
SQL	Structured Query Language

Listings

(Put long listings here.)

```
if (_nClusters < 1)
        throw std::string ("unknown number of clusters");
if (_nIterations < 1 and _epsilon < 0)
        throw std::string ("You should set a maximal number of iteration
 ↪  or minimal difference -- epsilon.");
if (_nIterations > 0 and _epsilon > 0)
        throw std::string ("Both number of iterations and minimal epsilon
 ↪  set -- you should set either number of iterations or minimal
 ↪  epsilon.");
```

List of additional files in electronic submission (if applicable)

Additional files uploaded to the system include:

- source code of the application,
- test data,
- a video file showing how software or hardware developed for thesis is used,
- etc.

List of Figures

List of Tables