POLITECHNIKA GDAŃSKA | WYDZIAŁ ELEKTRONIKI, TELEKOMUNIKACJI I INFORMATYKI

**Group Project Documentation**

# Schedule and Requirements Specification

Faculty of Electronics, Telecommunications and Informatics

Gdansk University of Technology

| | | |
|---|---|---|
| **Project name and acronym:**<br>{name of the project, e.g.: Port security system against terrorist threats - SZP}<br>**Redundant coding visualization app** | **Principal:**<br>{customer name}<br>Bartosz Czaplewski, PhD | |
| **Order number:**<br>{number of the project team within the Group Project according to the SPG system, e.g. 13@KSSR'2022}<br>**5@KSTI'2023/24** | **Project manager:**<br>{project team leader}<br>**Bartosz Kołakowski** | **Client:**<br>{Client's name}<br>**Bartosz Czaplewski, PhD** |

| | |
|---|---|
| **Document name/code:**<br>**Schedule and specification of requirements – HiSW** | **Version No.:**<br>{document version e.g. 1.00}<br>**6.00** |
| **Responsible for the document:**<br>{surname, first name}<br>**Kołakowski Bartosz** | **Date of first drafting:**<br>{date of first version of the document}<br>**31.10.2023** |
| | **Last updated date:**<br>{date of execution of the current version of the document}<br>**28.05.2024** |
| | **Semester of the Group Project:**<br>{enter 1 or 2} **2** |

**Historia dokumentu**

| Version | Description of the modification | Chapter / page | Author of the modification | Date |
|---|---|---|---|---|
| 1.00 | {description, e.g. draft version}<br>Draft version | {e.g. whole}<br>whole | {surname, first name}<br>Kołakowski Bartosz | {date modified}<br>31.10.2023 |
| 2.00 | {description, e.g. draft version}<br>Add dates to your work schedule | (e.g. the whole}<br>item 2 | (surname, first name}<br>Kołakowski Bartosz | {date modified}<br>8.11.2023 |
| 3.00 | {description, e.g. draft version}<br>More detailed descriptions of project stages | {e.g. points 2, 2.3}<br>point 2.1 | (surname, first name}<br>Kołakowski Bartosz | {date modified}<br>21.11.2023 |
| 4.00 | {description, e.g. draft version}<br>Change of dates in the schedule, refinement of the third point | {e.g. points 2, 2.3}<br>points 2 and 3 | (surname, first name}<br>Jastrzębski Paweł | {date modified}<br>01.01.2024 |
| 5.00 | {description, e.g. draft version}<br>Adding information in the third point | {e.g. points 2, 2.3}<br>point 3 | (surname, first name}<br>Kołakowski Bartosz | {date modified}<br>04.01.2024 |
| 6.00 | {description, e.g. draft version}<br>Update information | {e.g. points 2, 2.3}<br>whole | {surname, first name}<br>Kołakowski Bartosz | {date modified}<br>28.05.2024 |
| {version} | {description, e.g. adding stage C} | {e.g. points 2, 2.3} | {surname, first name} | {date modified} |

{NOTE: in the second semester, the documentation can be an extension of the documentation from the first semester (new version of the document), it can also be a new file
NOTE: the schedule can be planned in the first semester for 2 semesters at once – however, the file from the first semester must be updated in the second semester; it may also contain additional provisions established after the end of the first semester
NOTE: The schedule created in the SPG system does not have to be the same as in this document}

# Table of contents

# 1 Introduction - About the document

## 1.1 Full Document

{do not change}

The purpose of the document is to document the planned schedule of project implementation in the semester, the planned division of tasks in the project team, indicate and describe the tasks and roles of the responsible persons, as well as specify the requirements for the project along with the acceptance criteria imposed by the supervisor and the client.

## 1.2 Customer

{specification of the addressees of the document, it can be the type of recipient; here: the contractor (Department), members of the project team and named persons to whom the document is to reach}

Principal - Bartosz Czaplewski, Ph.D. (KSTI)
Team Members:

Bartosz Kołakowski
Michał Mróz
Paweł Jastrzębski
Maxim Novak
Piotr Noga

## 1.3 Terminology

{explanation of terms and abbreviations used in the document, designations used inside the document, e.g. requirements designations}

Abbreviations of people:
Bartosz Kołakowski - BK
Michał Mróz - MM
Paweł Jastrzębski - PJ
Maksym Novak - MN
Piotr Noga – PN

GUI – Graphical User Interface

# 2 Schedule of the project team's work

{main stages, investigators, beginning, end - in the form of a table; at least 3 main stages of project implementation per semester should be defined}

Creating a project database, implementing Hamming code, implementing Red-Solomon code, creating a GUI database, visualizing Hamming code, visualizing Red-Solomon code, writing tests for Hamming code, writing tests for R-S code

| Name | BK | MM | PJ | MN | PN | Inception | Conclusion |
|---|---|---|---|---|---|---|---|
| creating a project database | | X | | | | 31.10.2023 | 5.11.2023 |
| creating a GUI database | | X | | | X | 31.10.2023 | 15.11.2023 |
| implementation of Hamming code | | X | X | | | 03.11.2023 | 01.12.2024 |
| implementation of Reed-Solomon code | | | X | X | | 21.02.2024 | 30.04.2024 |
| visualization of Hamming code | X | X | | X | X | 05.11.2023 | 10.01.2024 |
| Reed-Solomon code visualization | | X | X | X | X | 01.03.2024 | 31.05.2024 |
| writing tests for Hamming code | X | X | | | | 05.11.2023 | 22.11.2024 |
| writing tests for the Reed-Solomon code | X | | X | X | | 01.03.2024 | 31.05.2024 |
| writing an English translation | | | | | X | 01.04.2024 | 31.05.2024 |

## 2.1 Description of the stages of development (project management)

{main tasks in individual stages)

**creating a project database**

{goals, products, acceptance criteria, main tasks, etc.}

Goals: To select a technology, to prepare the technology for use, to create a project on Github
Products: a repository on github that can be accessed by every member of the group, the base configuration of the QT framework (c++) has been uploaded to the repository
Acceptance criteria: access of group members to the repository with push permissions (except for branch main) and the ability to launch the project

**creating a GUI database**
{goals, products, acceptance criteria, main tasks, etc.}
Goals: Basic user interface, creating options for choosing code visualization, options for inserting code
Products: code in QT presenting a GUI in which you can choose what code you want to visualize (without implementing this step yet)
Acceptance criteria: aesthetic GUI, runs on the computer of each group member, available on Linux and Windows

**implementation of Hamming code**
{goals, products, acceptance criteria, main tasks, etc.}
Goals: To create a Hamming code algorithm
Products: c++ code that allows you to encode data with Hamming code, function to encode information and function to decode
Acceptance criteria: correctly implemented algorithm, consistent with coding theory

**implementation of Reed-Solomon code**
{goals, products, acceptance criteria, main tasks, etc.}
Goals: To create a Reed-Solomon code algorithm
Products: c++ code that allows you to encode data with Reed-Solomon code, function for encoding information and function for decoding
Acceptance criteria: correctly implemented algorithm, consistent with coding theory

**visualization of Hamming code**
{goals, products, acceptance criteria, main tasks, etc.}
Goals: To create a visualization of the Hamming code with the possibility of user input
Products: code in c++ (QT framework) that allows to display the operation of Hamming code for the input entered
Acceptance criteria: the visualization in the GUI shows the operation of the Hamming code in an understandable and aesthetic way, the correct acceptance of user input. First, it encodes the entered sequence of bits, then allows the encoding result to be changed, and then performs the decoding

**Reed-Solomon code visualization**
{goals, products, acceptance criteria, main tasks, etc.}
Goals: To create a visualization of the Reed-Solomon code with the possibility of user input
Products: code in c++ (QT framework) that allows to display the operation of the Reed-Solomon code for the input entered
Acceptance criteria: the visualization in the GUI shows the operation of the Reed-Solomon code in an understandable and aesthetic way, the correct acceptance of user input. First, it encodes the entered sequence of bits, then allows the encoding result to be changed, and then performs the decoding

**writing tests for Hamming code**
{goals, products, acceptance criteria, main tasks, etc.}
Objectives: To write tests to check the correctness of Hamming code execution
Products: Tests that run automatically when building applications to check whether the functions for encoding and decoding information work correctly
Acceptance criteria: after a minimum of 5 coding and decoding tests

**writing tests for the Reed-Solomon code**
{goals, products, acceptance criteria, main tasks, etc.}
Objectives: To write tests to check the correctness of the execution of the Reed-Solomon code
Products: Tests that run automatically when building applications to check whether the functions for encoding and decoding information work correctly
Acceptance criteria: after a minimum of 5 coding and decoding tests

**writing a translation of the program into English**
{goals, products, acceptance criteria, main tasks, etc.}
Objectives: To write a translation for the existing Polish version of the program into English
Products: The content of the program translated into English and properly integrated into the content of the program
Acceptance Criteria: Fully translated program into understandable English

# 3 Planned division of tasks and roles in the project in the project team

## 3.1 Description of the tasks planned to be carried out with the indication of responsible persons

{main tasks in individual stages, indicated in the schedule from point 2)

### Creating a repository on GitHub

{Specific tasks/roles, Responsible/Executor(s)
Creating a repository, giving all members access rights, adding a description and typical settings such as not being able to push to the main branch, the ability to accept a Pull Request only after a code review. This allows you to control the version and revert to the previous one if necessary.
Michał Mróz

### Code Review

{Specific tasks/roles, Responsible/Executor(s)
Performing code review for Pull Requests on GitHub – checking if the code contains any errors and if it performs all the established functionalities, accepting the Pull Request.
Paweł Jastrzębski, Michał Mróz, Piotr Noga

### User interface design

{Specific tasks/roles, Responsible/Executor(s)
Creating the concept of the general appearance of the application and implementing this appearance. The interface must meet the aesthetic requirements.
Michał Mróz, Piotr Noga

### Writing tests

{Specific tasks/roles, Responsible/Executor(s)
Writing tests to check the correctness of the written functions, the tests concern both the coding process and the decoding process.
Bartosz Kołakowski, Michał Mróz, Paweł jastrzębski, Maksym Nowak

### Team management

{Specific tasks/roles, Responsible/Executor(s)
Organizing work, setting deadlines, discussing changes in the project, organizing group communication, taking care of positive relationships in the group.
Bartosz Kołakowski

### Implementing redundant code

{Specific tasks/roles, Responsible/Executor(s)
The actual implementation of the codes, in accordance with their theoretical algorithms.
Bartosz Kołakowski, Michał Mróz, Paweł jastrzębski, Maksym Nowak

### Create a redundancy code visualization

{Specific tasks/roles, Responsible/Executor(s)
Creating a visualization of how the code works, showing the result to the project supervisor and making corrections indicated by the supervisor.
Bartosz Kołakowski, Michał Mróz, Paweł jastrzębski, Maksym Nowak, Piotr Noga

### Writing an English translation

{Specific tasks/roles, Responsible/Executor(s)
Translating the program so that it is understandable for people who speak English. It is assumed that the functionality of the program will be equally available in both Polish and English.
Piotr Noga

### Document Writing

{Specific tasks/roles, Responsible/Executor(s)
Writing all documents required to pass the group project; schedule, documentation, report, poster, project information
Bartosz Kołakowski, Paweł Jastrzębski

### Creating a presentation

{Specific tasks/roles, Responsible/Executor(s)
Creation of a presentation on the progress of the project, which will be presented at the cathedral seminar
Bartosz Kołakowski, Paweł Jastrzębski, Maksym Nowak

# 4 Product requirements and acceptance criteria

## 4.1 General description of the planned product

{draw up a general description of the product, what it is supposed to be, what it is for, what main functions it is to perform, etc.; in semester 2, the information should be updated if there have been changes}

The aim of the project is to write a tool to support the teaching process. The tool is to be an application that visualizes issues related to redundant coding. The application is to vividly present the classification of redundant codes, the process of encoding and decoding selected redundant codes, the gain of coding, various issues related to the topic, and allow you to conduct experiments. The application is to be available in many languages.

## 4.2 Minimum requirements for the product

{describe what are the minimum quality requirements for the product, try to determine the methods of testing the minimum requirements, provide verification methods; in semester II, the information from semester I should be verified and updated, if there have been changes}

You need to visualize at least two redundant codes – the Hamming code and the Reed-Solomon code
The application must have an aesthetically pleasing graphical user interface and must be run on MS Windows.
The application must be translated into English. Your app must provide the ability for the user to provide data.

## 4.3 Reception conditions

{from the point of view of the contract - when we consider the project to be successful - qualification tests, meeting the requirements, (technical, legal, financial conditions; in semester II, information from semester I should be verified and updated, if there have been changes}

Meeting the requirements of the project assumed by the supervisor by the end of the time allocated for implementation, i.e. the end of the second semester.
Approval of the project by the project supervisor.

# 5 Provisions

## 5.1 Provisions on changes to the original plan and scope of work

{if there are any changes in relation to the original plan, they should be indicated, e.g. if the schedule was created entirely in the first semester, but immediately for 2 semesters and there are some changes, then they should be indicated, if there are no resolutions, then "not applicable" should be entered}At
the beginning, Reed-Solomon coding (and visualization and writing tests for them) were to be completed in the first semester, however, due to its focus on execution quality, Reed-Solomon coding was postponed to the second semester.

## 5.2 Other provisions

*{if any}*