Computer Science Department

Software Engineering & Business Analysis

Bachelor's Thesis

Capstone Project

BeatRate Web Application Paper

Yaroslav Khomych

Supervisor
KSE, Isaac Newton, PhD, inewton@kse.org.ua

Expert
Company, Expert Name, expert@domain.ua

Submission date
14 August 2025

TODOS

Academic Integrity Statement

I, undersigned, hereby declare that this capstone project is the result of my own work.

- All ideas, data, figures and text from other authors have been clearly cited and listed in the bibliography.
- No part of this project has been submitted previously for academic credit in this or any other institution.
- All code, diagrams, and third-party materials are either my original work or are used with permission and properly referenced.
- I have not engaged in plagiarism or any form of academic dishonesty.
- Any assistance received (e.g. from peers, tutors, or online forums) is acknowledged in the acknowledgements section.

I understand that failure to comply with these declarations constitutes academic misconduct and may lead to disciplinary action.

Place, date	Kyiv, 22.05.2025
Signature	____________________

Contents

Acknowledgements

While optional, acknowledgements provide an opportunity to express gratitude to individuals, institutions, or organizations that have supported you throughout your academic journey.

Despite not impacting the evaluation, acknowledgements contribute to the overall tone and appreciation within your thesis.

Abstract

The abstract serves as a concise summary of your entire thesis, encapsulating key elements on a single page such as:

- *General background information*
- *Objective(s)*
- *Approach and method*
- *Conclusions*

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Keywords:

KSE, *Software Engineering*, *Thesis*, *Template*

1 Introduction

Your introduction serves to introduce the topic of your Bachelor thesis and to arouse the reader's curiosity with an overview. Why it is important and how it is structured, we explain here.

You can consider an introduction as a teaser for your bachelor thesis. You arouse interest and give a foretaste by presenting your motivation, your method and the state of research in your introduction.

Convince your examiners already in the introduction that your Bachelor thesis will be exciting. If your professor starts reading your thesis with anticipation and interest, the chances of getting good grades are higher.

Pay particular attention to the following in your introduction:

- ***Introduce the topic*** *- What characterizes the topic?*
- ***Introduce the goal*** *- What do you want to achieve with your thesis?*
- ***Make the reader curious*** *- What motivates the reader to read on?*
- ***Describe the relevance*** *- Why is this bachelor thesis scientifically relevant?*

The introduction should have the following content:

- ***Initial situation presentation of the topic*** *- You introduce the topic with an exciting 'bait'. You provide initial information on the topic and the object of research and explain the current state of research.*
- ***Relevance of the topic motivation*** *- You justify the relevance of your topic (scientifically) and place it in the context of your field. In addition, it is often required that you disclose your personal motivation.*
- ***Objectives*** *- Your introduction should clearly state what the goal of your paper is and what outcome you hope to achieve upon completion of the bachelor thesis.*
- ***Method*** *- You explain the approach and justify the choice of method.*
- ***Structure of the Bachelor's thesis*** *- Finally, you give the reader a general overview of your Bachelor's thesis by explaining the structure, showing the red thread and how the research question is answered.*

> Welcome to the template's introductory chapter! Instead of boring you with lorem ipsum, here's a quick guide to what you can do in Typst and, more specifically, in this template.
>
> Need more? Check out the Guide to Typst.

1.1 Basic markup

Typst lets you create bold, italic, or monospaced text with ease. You can also sprinkle in equations like $e^{i\pi} + 1 = 0$ or even inline code like `fn main() { println!("Hello, World!") }`. And because life is better in color: pink, blue, yellow, orange, green, and more! Boldly colorize!

You can also write numbered or unnumbered lists:

- First item
- Second item
  1. First Subitem

   2. Second Subitem
- Third item

Need equations? Sure! They look great as blocks too:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

1.2 Images

As they say, a picture is worth a thousand words. Let's add one:

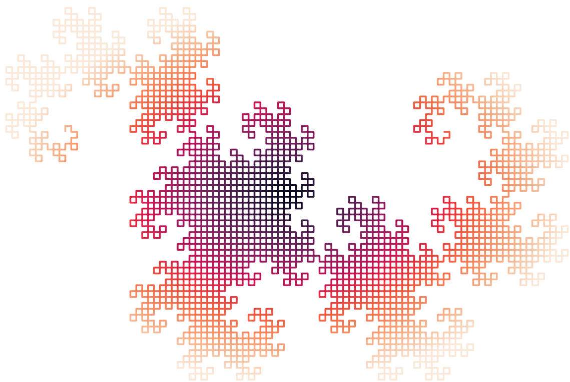

Figure 2: Project logo

1.3 Tables

Tables are great for organizing data. From simple to complex, Typst handles them all:

Name	**Age**	**City**
Albert Einstein	25	Bern
Marie Curie	22	Paris
Isaac Newton	30	London

Table 1: Simple table

[31:27]			[24:20]	[19:15]	[14:12]	[11:7]	[6:0]
funct5	aq	rl	rs2	rs1	funct3	rd	opcode
5			5	5	3	5	7

Table 2: Complex table

1.4 Boxes

Highlight key points with these fun boxes (and more):

Infobox: For highlighting information.

Ideabox: Share a brilliant idea.

Warningbox: Proceed with caution!

Firebox: This is 🔥 !

Rocketbox: Shoot for the moon!

Todobox: Just do it!

TODO
Personnal todo before marking this thesis as final

1.5 Citations, Acronyms and Glossary

Add citations with `@` like [1] or [1, p.7ff] (stored in `/tail/bibliography.bib`).

Acronym terms like Infotronics (IT) expand on first use and abbreviate after IT. Glossary items such as Rust Programming Language (Rust) can also be used to show their description as such: Rust is a modern systems programming language focused on safety, speed, and concurrency. It prevents common programming errors such as null pointer dereferencing and data races at compile time, making it a preferred choice for performance-critical applications.. Acronyms and glossary entries auto-generate at the document's end (defined in `/tail/glossary.typ`).

1.6 Code

Besides writing inline code as such `fn main() { println!("Hello World") }` you can also write code blocks like this:

```rust
fn main() {
  let ship = Starship::new("USS Rustacean", (0.0, 0.0, 0.0));
  let destination = (42.0, 13.0, 7.0);
  let warp = ship.optimal_warp(ship.distance_to(destination));

  println!("🖐 {} traveling to {:?} at Warp {:.2}", ship.name, destination, warp);
  if warp <= 9.0 {
    println!("✨ Warp engaged!");
  } else {
    println!("⚠ Warp failed!");
  }
}
```

Listing 1: First part of the USS-Rustacean code

or directly from a file

```rust
struct Starship {
  name: String,
  position: (f64, f64, f64),
}

impl Starship {
  fn new(name: &str, position: (f64, f64, f64)) -> Self {
    Self {
      name: name.into(),
      position,
    }
  }
  fn distance_to(&self, dest: (f64, f64, f64)) -> f64 {
    ((dest.0 - self.position.0).powi(2)
      + (dest.1 - self.position.1).powi(2)
      + (dest.2 - self.position.2).powi(2))
    .sqrt()
  }
  fn optimal_warp(&self, distance: f64) -> f64 {
    (distance / 10.0).sqrt().min(9.0)
  }
}
```

Listing 2: Second part of the USS-Rustacean code from `/resources/code/uss-rustacean.rs`

1.7 Context Problem

Haute École d'Ingénierie (HEI) Rust Rust programs

[1], [1, p.7ff]

```rust
fn main() {
  println!("Hello World!");
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

1.8 Objectives

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

1.9 Structure of this report

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

2 Research

In this section, you perform a State of the Art investigation of your domain to understand what's been built, how it works, and where it falls short. You'll survey existing solutions—tools, frameworks and approaches—then compare them against your project's requirements to justify why a new tool (and its specific feature set) is needed.

A focused Domain Area Research unfolds in these steps:

1. *Clarify your research questions and functional requirements*
2. *Collect and review candidate solutions*
3. *Evaluate each alternative's strengths, weaknesses, maturity and architectural style*
4. *Group similar approaches into meaningful categories (for example, monolithic vs. microservice, commercial vs. open-source)*
5. *Pinpoint gaps or missing features that motivate your own design*

This process carries your initial concept through systematic research all the way to a clear, actionable set of requirements for your proposed tool.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Contents

2.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

2.2 Section 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

2.3 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

3 Design

In this section you turn your requirements into a concrete engineering blueprint. You'll justify every major architectural choice, visualize structure with C4 diagrams for the first three layers, and map out your runtime topology so that peers can understand—and you can defend—every aspect of your system.

1. *Clarify how functional and non-functional requirements drive your high-level architecture*
2. *List each architectural decision (for example, "We chose microservices to enable independent scaling and deployment") and explain why it best meets your goals*
3. *Include a C4 Context diagram showing your system in its environment (users, external systems, data sources)*
4. *Include a C4 Container diagram breaking the system into deployable units (APIs, web front end, background workers, databases) and annotate communication styles and protocols*
5. *Include a C4 Component diagram for your core container(s), illustrating key modules, services or libraries and their interactions*
6. *Describe your deployment topology: physical or cloud hosts, network zones, load-balancing, failover and backup strategies*
7. *Summarize your technology stack, mapping each tool or framework back to a specific container or component and noting any trade-offs (performance, community support, learning curve)*
8. *Outline how data flows through the system—including storage models, messaging patterns or API contracts—and note any schema or interface versioning plans*
9. *Address cross-cutting concerns (Security, Logging, Monitoring, Scalability) and show where they sit in your topology*

By walking through Requirements → Decisions → Diagrams → Topology, your Design section becomes a rigorous, evidence-backed foundation for the implementation that follows.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Contents

3.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

3.2 Section 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

3.3 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

4 Implementation

In this section you translate your component-level designs into working code and systems. Focus on the C4 Component layer and on the details needed to show how your design was realized. Include only the most important code snippets that illustrate key patterns or algorithms, rather than full listings.

1. *Describe the development methodology (for example, Agile or test-driven development) used to guide your implementation*
2. *Explain any prototyping or iterative strategies you applied to refine components before full-scale coding*
3. *Summarize coding standards, naming conventions and architectural patterns followed in your codebase*
4. *Present critical code snippets or configuration templates that highlight how core components were implemented (for example, key classes, interfaces or algorithms)*
5. *Detail your testing approach and quality assurance measures (unit tests, integration tests, coverage metrics)*
6. *Note any performance optimizations or profiling results for components that were bottlenecks*
7. *Outline your deployment and configuration management process for component artifacts (containerization, CI/CD pipelines)*
8. *Highlight documentation deliverables (API references, inline comments, architecture decision records) that support future maintenance*

This section demonstrates how each component specification becomes actual, maintainable code—closing the loop from design to implementation.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Contents

4.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

4.2 Section 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

4.3 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

5 Validation

Validation (Requirements Verification and Testing)

In this section you demonstrate how your implementation satisfies the initial requirements through clear testing methods and concise examples—suitable for a bachelor-level project:

1. *Restate each key functional and non-functional requirement from your Analysis and Design sections*
2. *Describe the testing approach for each requirement (for example, unit tests, manual acceptance checks or scenario walkthroughs)*
3. *Provide concrete test cases or usage examples that show how you verify each requirement in practice*
4. *Summarize actual versus expected outcomes, indicating pass/fail status for each test*
5. *Include brief snippets of test code or sample console outputs to illustrate your procedures*
6. *Note any gaps or deviations and suggest simple fixes or areas for future improvement*
7. *If a feature wasn't intended for specific scenarios (e.g. high-load), omit unrealistic stress tests and clearly document its current limitations*

This focused structure ties every requirement directly to validation results, using examples and methods you can realistically carry out at the bachelor level.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Contents

5.1 Section 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

5.2 Section 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

5.3 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

6 Conclusion

Conclusion

In this final section you bring together your work and reflect on its impact. Keep it concise, restating key points without introducing new information:

1. *Project Summary: Briefly recap objectives, methodology and principal results*
2. *Alignment with Objectives: Discuss how outcomes meet initial goals, referencing requirements and design aims*
3. *Lessons Learned and Challenges: Note any obstacles and how they informed improvements*
4. *Limitations: Acknowledge features or scenarios beyond this scope and clearly state current system boundaries*
5. *Future Work: Suggest practical enhancements or research directions building on your findings*

Avoid introducing new concepts here; refer readers to the Discussion for deeper analysis.

6.1 Project summary

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

6.2 Comparison with the initial objectives

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

6.3 Encountered difficulties

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

6.4 Future perspectives

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Glossary

Programming Language

***Rust* – Rust Programming Language**: Rust is a modern systems programming language focused on safety, speed, and concurrency. It prevents common programming errors such as null pointer dereferencing and data races at compile time, making it a preferred choice for performance-critical applications. 5, 6

University

***HEI* – Haute École d'Ingénierie** 6

***IT* – Infotronics** 5

Bibliography

[1] S. Zahno *et al.*, “Dynamic Project Planning with Digital Twin,” *Frontiers in Manufacturing Technology*, vol. 3, May 2023, doi: 10.3389/fmtec.2023.1009633.