

SFWR ENG 3A04 Deliverable #2

Tutorial Number: T02

Group Number: G03

Group Members:

- Takhtar, Tarnveer (Captain)
- Hannoufa, Akram
- Patel, Aryan
- Damjanovski, Alexander
- Bradbury, Matthew

1 Introduction

This section should provide a brief overview of the entire document.

1.1 Purpose

This document is intended to provide a high-level overview of the *LockTalk* system architecture. This includes discussing the architectural style and design decisions/considerations for this system.

The intended audience of this document is both the stakeholders as well as those working on developing the system. This includes project managers, developers, requirement engineers, and domain experts with the intention of giving a concrete plan for the system and the context behind design decisions. This is also intended to be shown to internal stakeholders for a progress update and more information regarding our system.

1.2 System Description

The system will use Model-View-Controller (MVC) architecture with select subsystems following a repository architecture design style. MVC is the best option for this system as it promotes interaction between the system and user, and allows for better implementation of logic needed to interface with the repository subsystems.

The repository was chosen for the subsystems as it was the best option to enforce multiple views while also being able to store large amounts of data and provide necessary security measures.

1.3 Overview

Section 2 shows an analysis class diagram, displaying a high-level overview of the system and its interactions. In section 3, we outline the overall architectural design, considering different design alternatives and going over the reasons behind sticking with our choice of architecture. Finally, section 4 includes Class Responsibility Collaboration (CRC) Cards to provide a closer look into each individual class's responsibilities and interactions.

2 Analysis Class Diagram

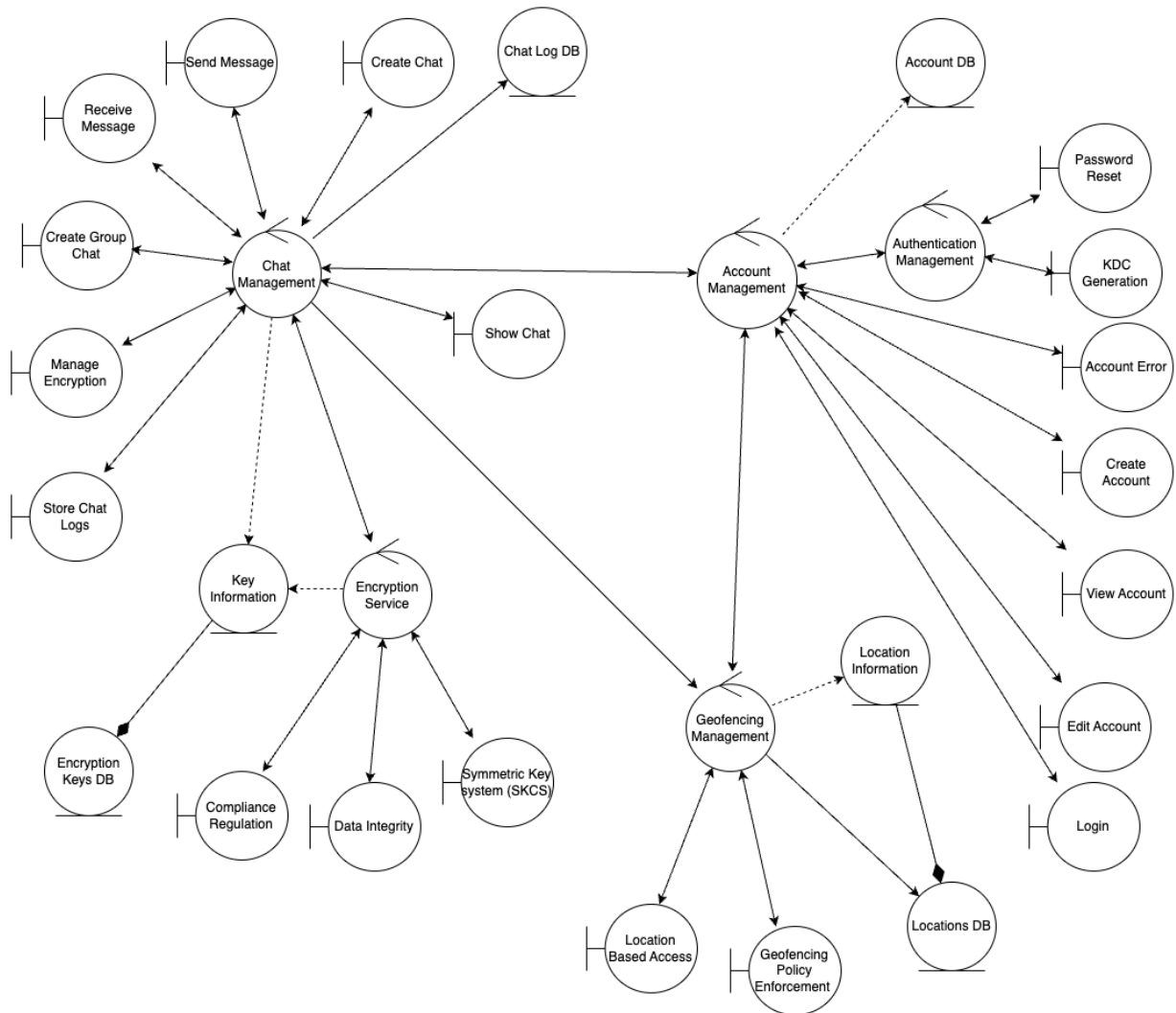


Figure 1. Analysis Class Diagram

3 Architectural Design

This section should provide an overview of the overall architectural design of your application. Your overall architecture should show the division of the system into subsystems with high cohesion and low coupling.

3.1 System Architecture

LockTalk uses a Model-View-Controller (MVC) software architecture. LockTalk needs to be able to respond to extensive user input and display specific and different information to each user depending on which conversations and messages they should be able to see. Thus, an Interaction-Oriented software architecture is essential.

Additionally, the controller and model represent the logic necessary to interface with the repository-based subsystems, as this will ensure that employees can only message other employees of their own company, can only view their own messages, and maintain other important features and security measures. MVC architecture is well suited for enforcing this kind of logic when accessing data, constructing different views for the users based on permissions, and changing these views in response to user input.

The following subsystems are defined to be within the controller:

| Subsystem | Use | Architectural Style |
|-----------------------|---|---------------------|
| Chat Management | Create and Send Messages | Repository |
| Account Management | Create and manage an account | Repository |
| Geofencing Management | Keeps track of when and where chat was accessed | Repository |
| Encryption Management | Generates and manages encryption keys | Repository |

Details on each subsystem and their relations can be found in section 3.2

A key component of the application is that for each user, different views will need to be visible. Each user will need to have only their own conversations displayed to them, and message data needs to be displayed differently even for two people in the same conversation (one's own messages should be displayed on the right side of the screen and in a different color than their conversation partner's). As such, the MVC architecture's ability to synchronize data across different views makes it ideal for the high-level architecture of our system. Constant user input/interaction is also feasible with this architecture, meaning its views can quickly respond to new messages being sent/received, and to aesthetic personalization from the user, which is essential.

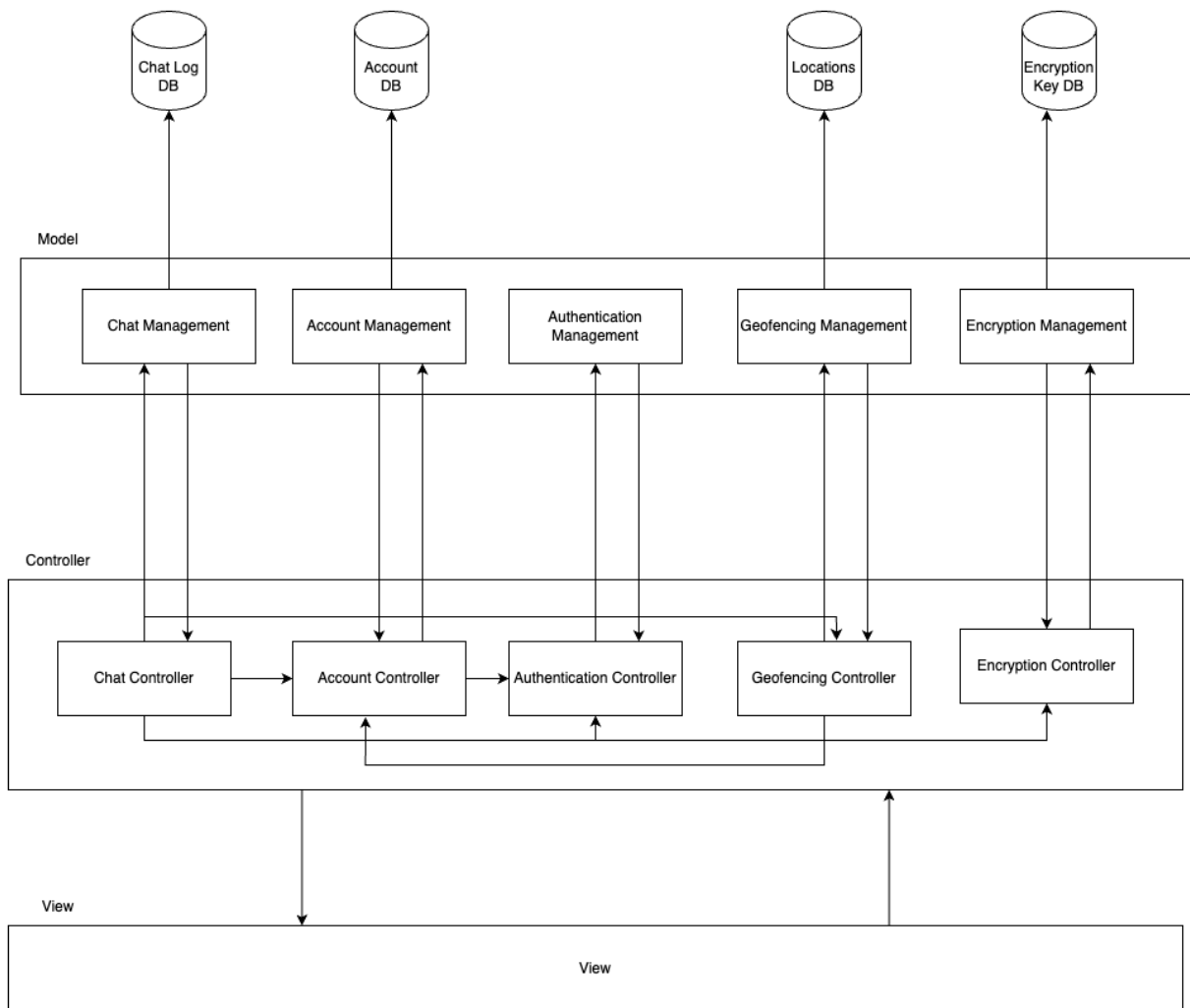


Figure 2. System Architecture Diagram

Before reaching our conclusion of choosing the Model View Controller (MVC) architecture, the following architectures were also considered:

Pipe and Filter

This style was initially considered, however, due to the dynamic and varied user interactions that will occur within the system, the rigid structure of the pipe and filter architecture will not accommodate this dynamic setting very well. For example, users can send messages, receive messages, start new conversations, and join existing conversations, there is not a simple single-order workflow that can be put together to handle these dynamic interactions. User

messages may even contain various characters, along with files and so wouldn't be very flexible for this purpose.

Batch Sequential

The Batch Sequential style was also considered, but seeing as the purpose of this system is to be a fast and secure real-time messaging platform for the company, the architecture is not ideal. Data being batched could affect the ability of the system to handle messages in real-time and could become a bottleneck of the system, so was eliminated.

Blackboard

Due to there being no real decision-making component in our system, and remaining simple when it came to the message-sending aspect of our system, the need for a knowledge agent-based subsystem over a repository subsystem was deemed unnecessary and added complexity to a system that requires clear decision-making processes and synchronization of messages between users.

Process Control Architecture

As this architecture was deemed to be better suited for embedded systems, it was concluded that the architecture wasn't suited for the system being designed.

3.2 Subsystems

Chat Management Subsystem

This subsystem is responsible for creating and sending messages within the LockTalk app. It handles the core functionality related to messaging, including composing messages, sending them to their designated recipient selected from the company directory, and managing message history. This subsystem will heavily interact with the account management subsystem to ensure that messages are only sent between current employees of the company (the company directory that is contained in the account database). Additionally, the subsystem will also interact with the encryption management and geofencing subsystems for encrypting and decrypting data transmissions and updating the access location of chats.

Account Management Subsystem

This subsystem is responsible for handling the creation and management of user accounts within the LockTalk app. It allows users to register for an account, log in securely, and manage their account security settings. This subsystem will interact with the encryption management subsystem for secure login purposes, and the geofencing management subsystem to retrieve relevant data.

Geofencing Management Subsystem

This subsystem is responsible for tracking the geographical locations and timestamps when users access the LockTalk app. It enables features such as geographically restricted messaging points or monitoring employee attendance based on location, and logs when and where chats

were accessed. This subsystem will be interacting with the chat management and account management systems, providing geographical data for the other subsystems to track and log.

Encryption Management Subsystem

This subsystem is responsible for generating and managing encryption keys used to secure communications within the LockTalk app. It ensures that messages exchanged between users are encrypted while in transmission to safeguard against unauthorized access or interception by another party. The encryption management subsystem plays a critical role in securing communications between users handled by the chat management subsystem and encrypting login information in transmission prior to authentication with the account management subsystem.

4 Class Responsibility Collaboration (CRC) Cards

| | |
|---|--|
| Class Name: Chat Management Controller | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none">• Knows Store Chat Logs• Knows Manage Encryption• Knows Create Group Chat• Knows Receive Message• Knows Send Message• Knows Create Chat• Knows Show Chat• Knows Chat Log DB• Knows Encryption Service• Knows Key Information DB• Knows Geofencing Management• Knows Account Management | <ul style="list-style-type: none">• Store Chat Logs• Manage Encryption• Create Group Chat• Receive Message• Send Message• Create Chat• Chat Log DB• Show Chat• Encryption Service• Key Information• Geofencing Management• Account Management |

| | |
|---|--|
| Class Name: Store Chat Logs (Boundary) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none">• Knows Chat Management Controller• Handles storing chat log history | <ul style="list-style-type: none">• Chat Management Controller |

| | |
|--|--|
| Class Name: Manage Encryption (Boundary) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Handles message encryption | <ul style="list-style-type: none"> • Chat Management Controller |

| | |
|---|--|
| Class Name: Create Group Chat (Boundary) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Handles group chat creation | <ul style="list-style-type: none"> • Chat Management Controller |

| | |
|---|--|
| Class Name: Receive Message (Boundary) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Handles message reception | <ul style="list-style-type: none"> • Chat Management Controller |

| |
|--|
| Class Name: Send Message (Boundary) |
|--|

| Responsibility: | Collaborators: |
|--|--|
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Handles sending messages | <ul style="list-style-type: none"> • Chat Management Controller |

| Class Name: Receive Message (Boundary) | |
|---|--|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Handles message reception | <ul style="list-style-type: none"> • Chat Management Controller |

| Class Name: Create Chat (Boundary) | |
|---|--|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Handles chat creation | <ul style="list-style-type: none"> • Chat Management Controller |

| Class Name: Show Chat (Boundary) | |
|----------------------------------|----------------|
| Responsibility: | Collaborators: |
| | |

| | |
|--|--|
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Handles message display and output | <ul style="list-style-type: none"> • Chat Management Controller |
|--|--|

| | |
|---|---|
| Class Name: Chat Log DB (Entity) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Knows Store Chat Logs • Knows chat messages sent on server • Knows Sent Timedate • Knows Message content • Knows Sender • Knows Receiver | <ul style="list-style-type: none"> • Chat Management Controller • Store Chat Logs |

| | |
|---|---|
| Class Name: Encryption Service Controller | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Knows Key Information | <ul style="list-style-type: none"> • Chat Management Controller • Key Information |

| |
|--|
| Class Name: Symmetric Key System (Boundary) |
|--|

| Responsibility: | Collaborators: |
|--|---|
| <ul style="list-style-type: none"> • Knows Encryption Service Controller • Handles symmetric-key crypto system for encryption/decryption | <ul style="list-style-type: none"> • Encryption Service Controller |

| Class Name: Data Integrity (Boundary) | |
|---|---|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Encryption Service Controller • Handles data integrity | <ul style="list-style-type: none"> • Encryption Service Controller |

| Class Name: Compliance Regulation (Boundary) | |
|--|---|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Encryption Service Controller • Handles compliance regulation | <ul style="list-style-type: none"> • Encryption Service Controller |

| Class Name: Key Information (Entity) | |
|---|-----------------------|
| Responsibility: | Collaborators: |
| | |

| | |
|---|---|
| <ul style="list-style-type: none"> • Knows Encryption Service Controller • Knows Chat Management Controller • Knows Encryption Keys DB • Knows Key value • Knows Key expiration date | <ul style="list-style-type: none"> • Encryption Service Controller • Chat Management Controller • Encryption Keys DB |
|---|---|

| | |
|---|---|
| Class Name: Encryption Keys DB (Entity) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Key Information | <ul style="list-style-type: none"> • Key Information |

| | |
|--|--|
| Class Name: Account Management Controller | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Knows Authentication Management Controller • Knows Geofencing Management Controller • Knows Account DB • Knows Account Error • Knows Create Account • Knows View Account • Knows Edit Account • Knows Login | <ul style="list-style-type: none"> • Chat Management Controller • Authentication Management Controller • Geofencing Management Controller • Account DB • Account Error • Create Account • View Account • Edit Account • Login |

| | |
|--|--|
| | |
|--|--|

| | |
|--|---|
| Class Name: Account DB (Entity) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Account Management Controller • Knows Company Directory DB • Knows Username • Knows Password • Knows Email • Knows First Name • Knows Last Name • Knows Account Management Controller • Knows Authentication Management Controller • Knows Account DB | <ul style="list-style-type: none"> • Account Management Controller • Company Directory DB • Authentication Management Controller |

| | |
|---|---|
| Class Name: Authentication Management Controller | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Account Management Controller • Knows Password Reset • Knows KDC Generation | <ul style="list-style-type: none"> • Account Management Controller • Password Reset • KDC Generation |

| |
|--|
| Class Name: Password Reset (Boundary) |
|--|

| Responsibility: | Collaborators: |
|--|--|
| <ul style="list-style-type: none"> • Knows Authentication Management Controller • Handles password resetting | <ul style="list-style-type: none"> • Authentication Management Controller |

| Class Name: KDC Generation (Boundary) | |
|--|--|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Authentication Management Controller • Handles key generation from the Key Distribution Centre • Handles KDC updates | <ul style="list-style-type: none"> • Authentication Management Controller |

| Class Name: Account Error (Boundary) | |
|--|---|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Account Management Controller • Handles account creation errors • Handles login errors and timeouts • Handles incorrect login information | <ul style="list-style-type: none"> • Account Management Controller |

| Class Name: Create Account (Boundary) |
|---------------------------------------|
|---------------------------------------|

| Responsibility: | Collaborators: |
|---|---|
| <ul style="list-style-type: none"> • Knows Account Management Controller • Handles account creation • Handles account registration | <ul style="list-style-type: none"> • Account Management Controller |

| Class Name: View Account (Boundary) | |
|---|---|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Account Management Controller • Handles admin account detail viewing • Handles user viewing own account details • Handles general account view | <ul style="list-style-type: none"> • Account Management Controller |

| Class Name: Edit Account (Boundary) | |
|--|---|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Account Management Controller • Handles admin account detail updates • Handles user updates to own account | <ul style="list-style-type: none"> • Account Management Controller |

| Class Name: Login (Boundary) |
|-------------------------------------|
|-------------------------------------|

| Responsibility: | Collaborators: |
|--|---|
| <ul style="list-style-type: none"> • Knows Account Management Controller • Handles account login | <ul style="list-style-type: none"> • Account Management Controller |

| Class Name: Geofencing Management Controller | |
|---|---|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Chat Management Controller • Knows Account Management Controller • Knows Locations DB • Knows Location Information • Knows Geofencing Policy Enforcement • Knows Location Based Access | <ul style="list-style-type: none"> • Chat Management Controller • Account Management Controller • Locations DB • Location Information • Geofencing Policy Enforcement • Location Based Access |

| Class Name: Location Information (Entity) | |
|--|--|
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Geofencing Management Controller • Knows Locations DB • Knows Location Latitude • Knows Location Longitude • Knows Location Radius • Knows Location Name • Knows Location Country • Knows Location Central City | <ul style="list-style-type: none"> • Geofencing Management Controller • Locations DB |

| | |
|--|--|
| Class Name: Locations DB (Entity) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Geofencing Management Controller • Knows Location Information | <ul style="list-style-type: none"> • Geofencing Management Controller • Location Information |

| | |
|---|--|
| Class Name: Location Based Access (Boundary) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Geofencing Management Controller • Handles app access depending on user location • Handles display to user of valid/invalid geographic location | <ul style="list-style-type: none"> • Geofencing Management Controller |

| | |
|---|--|
| Class Name: Geofencing Policy Enforcement (Boundary) | |
| Responsibility: | Collaborators: |
| <ul style="list-style-type: none"> • Knows Geofencing Management Controller • Handles parameters of the geofencing policy • Handles updates to the policy • Handles notifications of breach of policy | <ul style="list-style-type: none"> • Geofencing Management Controller |

A Division of Labour

Include a Division of Labour sheet which indicates the contributions of each team member. This sheet must be signed by all team members.

Hannoufa, Akram

- Initial and final drafts of the Class Responsibility Collaboration Cards
- Formatting, edits, final overview
- Section 4 final version



Patel, Aryan

- Section 2 Analysis Class Diagram
- Section 3 System Architecture
 - Section 3 description assistance
- Section 4 review and edits
- Update use case from D1



Bradbury, Matthew

- Initial and final drafts of section 1
- Edits, revisions, overview of final draft



Takhtar, Tarnveer

- Section 3.1 (Alternative Architecture Considerations), Section 3.2
- Formatting and review of final draft

A handwritten signature in black ink on a light gray background. The signature consists of the first name 'Tarnveer' written in a cursive script, followed by the last name 'Takhtar' written in a more stylized, bold cursive script.

Damjanovski, Alexander

- Section 3.1
- Formatting, edits, final overview

A handwritten signature in black ink on a white background. The signature is highly stylized and cursive, appearing to read 'Alexander Damjanovski'.