

Technical Report






Target release	03 Mar 2023
Document status	 This document is completed.
Product Manager	 @ Keopitou Doung
Developers	 @ samedy.phin @ Raksa Kun @ Sereysothirich Peang
QA	 Quality Assurance has not been selected for this project.
NOTE	 This document was generated by Confluence and subsequently imported as a PDF file . As a result, some of the content may have been lost in the process. For optimal viewing of the document, please visit this link below https://docs.pitou.live/content/63123/technical_report_telegram_automata

Table of Contents

- [Table of Contents](#)
- [About Us](#)
- [Introduction](#)
- [Functions and Features](#)
- [Milestones](#)
- [Data Structure](#)
 - [Built-In Data Structure](#)
 - [User-Defined Data Structure](#)
- [Database Design](#)
- [Implementation](#)
 - [?? Technology](#)
 - [Architecture and Application](#)
 - [Project Setup](#)
- [Result](#)
- [Conclusion and Perspective](#)

About Us

If orderly_database macro is not available:



Introduction

Automata theory is a branch of computer science that studies abstract models of computation such as finite automata, regular expressions, and languages. These models are useful for describing and analyzing various aspects of computation such as pattern matching, lexical analysis, parsing, and cryptography.

Automata Telegram Bot is a small application that runs entirely within the Telegram app. It allows users to create and manipulate finite automata machines using simple commands. Users can design, edit, delete and view their own machines, as well as convert them between different types and perform operations on them. For example, users can check if a string belongs to a language recognized by a machine, transform a nondeterministic finite automaton (NFA) into a deterministic one (DFA), or minimize a DFA. Automata Telegram Bot is an educational and fun tool for anyone interested in learning more about finite automata theory and practice.

Functions and Features

	Function	Responsible	Priority	Effort	Status	Description
1	start	@ Keopitou Doung	LOW	LOW	SHIPPED	<ul style="list-style-type: none"> Lists down all available commands Introduces to the bot
2	my_machines	@ Keopitou Doung	MEDIUM	LOW	SHIPPED	<ul style="list-style-type: none"> Lists down all machines the current user has
3	design	@ Keopitou Doung	HIGH	MEDIUM	SHIPPED	<ul style="list-style-type: none"> Asks the current user 5 consecutive questions: Q, X, Delta, q0, and F Stores the inputs in the database
4	delete	@ Keopitou Doung	MEDIUM	LOW	SHIPPED	<ul style="list-style-type: none"> Lists down all machines the current user has and delete any selected machine
5	edit	@ Keopitou Doung	MEDIUM	HIGH	IN PROGRESS	<ul style="list-style-type: none"> Lists down all machines the current user has and edit any selected machine with certain conditions
6	type	@ Sereysothirich Peang @ Raksa Kun	HIGH	HIGH	SHIPPED	<ul style="list-style-type: none"> Lists down all machines the current user has and returns the type of any selected machine

7	check_string	@ samedy.phin	HIGH	HIGH	SHIPPED	<ul style="list-style-type: none"> Lists down all machines the current user has and asks for input string for all selected machine(s) then returns back the result
8	nfa_to_dfa	@ Sereysothirich Peang @ Raksa Kun	HIGH	HIGH	SHIPPED	<ul style="list-style-type: none"> Lists down all NFA machines the current user has and converts the selected machine to an equivalent DFA
9	minimize_dfa	@ samedy.phin	HIGH	HIGH	SHIPPED	<ul style="list-style-type: none"> Lists down all DFA machines the current user has and minimizes the selected machine to as minimal as possible.

Milestones

Gantt chart is not supported to export.

If gantt-chart-macro is not supported, visit this [link](#).

Data Structure

We are using a combination of both Built-In Data Structure and User-Defined Data Structure of Python.

Built-In Data Structure

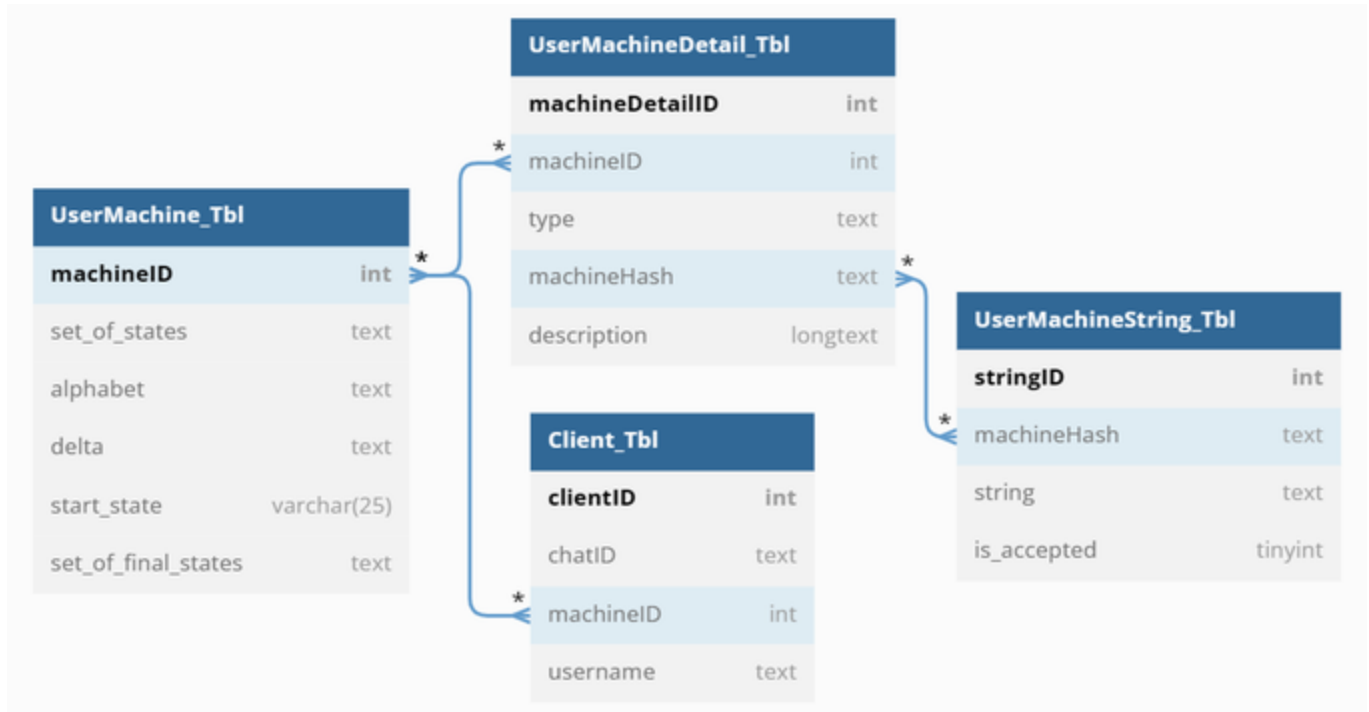
- ✓ List
- ✓ Dictionary
- ✓ Tuple
- ✓ Set

User-Defined Data Structure

- ✓ Tree

- ✗ Queue
- ✗ Stack
- ✗ Linked List
- ✗ Graph
- ✗ HashMap


Database Design




Implementation

Technology





 **Python** is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented, and functional programming.




 **MySQL** is an open-source relational database management system (RDBMS) based on Structured Query Language (SQL). It is the most popular database system used for web-based applications and is used to store, organize, and retrieve data stored in databases. MySQL is used by many web-based applications, including WordPress, Drupal, and Joomla. It is also used in many large-scale web applications, such as Facebook, Twitter, and YouTube. MySQL is highly scalable and can handle large amounts of data. It is also highly secure and reliable, making it an ideal choice for mission-critical applications.




 **GitHub, Inc.** is an Internet hosting service for software development and version control using Git. It provides the distributed version control of Git plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis for every project.

 **Telegram Messenger** is a globally accessible freemium, cross-platform, encrypted, cloud-based, and centralized instant messaging service. The application also provides optional end-to-end encrypted chats, popularly known as secret chat and video calling, VoIP, file sharing, and several other features.




 **BotFather** is a special bot in the Telegram messaging app that helps users create and manage their own bots. It provides users with a set of commands that can be used to create new bots, change their settings, and manage existing bots. BotFather also provides users with a list of available bots and allows them to search for bots by keyword. BotFather also allows users to customize their bots with custom commands and settings.




 **PlanetScale** is a cloud-native database-as-a-service (DBaaS) platform that provides a fully managed, distributed, and highly available database solution for applications. It is designed to help developers and DevOps teams quickly and easily deploy and manage distributed databases in the cloud. PlanetScale's platform is built on top of the open-source Vitess project, which provides a sharding and replication layer for MySQL and other relational databases. PlanetScale's platform provides a range of features, including automated sharding, automated failover, and automated scaling. It also provides a range of monitoring and management tools, including a web-based dashboard, an API, and a command-line interface.

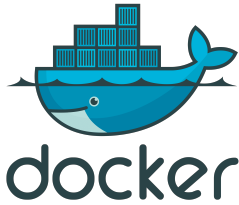


 **Railway** is a cloud-based platform that simplifies the process of shipping software. It provides a streamlined workflow for developers to quickly and easily deploy their applications to the cloud. Railway automates the process of setting up and managing cloud infrastructure, allowing developers to focus on building their applications instead of worrying about the underlying infrastructure. It also provides a unified dashboard for monitoring and managing applications, as well as a library of pre-built components for quickly deploying applications. Railway is designed to make the process of shipping software easier and more efficient, allowing developers to get their applications to market faster.



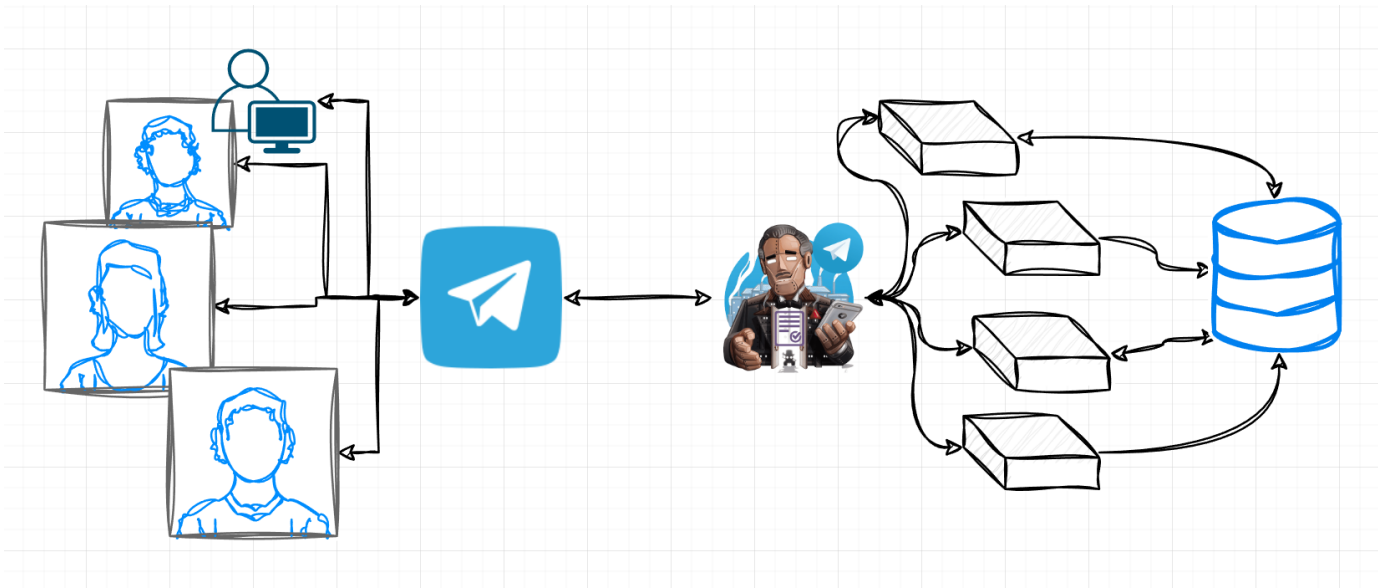
 **Confluence** is a collaboration software developed by Atlassian. It is designed to help teams work together more effectively by providing a platform for creating, sharing, and organizing content. It allows users to create, edit, and comment on documents, wikis, and other content. It also provides tools for managing tasks, tracking progress, and collaborating with other teams. Confluence also integrates with other Atlassian products, such as Jira and Bitbucket, to provide a comprehensive suite of tools for teams to work together.

Docker is an open-source platform that enables developers to build, deploy, run, update, and manage containers. Containers are standardized components that combine application source code with the operating system libraries and dependencies required to run that code in any environment. Docker allows you to separate your applications from your infrastructure so you can deliver software

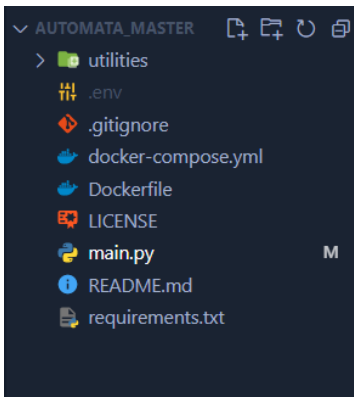


quickly and reliably. Docker also provides a comprehensive end-to-end platform that includes UIs, CLIs, APIs, and security for containerized applications.

Architecture and Application



Project Setup



- The folder “**utilities**” contains all required files for processing the backend of the application such as **database.py**, **FiniteAutomaton.py**, **fa_controller.py**,...etc.
- The **dot env** file contains all environments for the application, the database's credentials, and the telegram bot token.
- The **main.py** file contains the frontend code processing the user's requests **from the telegram server to our server and to the telegram server from our server**.
- The **requirements.txt** file contains the list of libraries to install and set up before starting the application.

Result

<https://clipchamp.com/watch/NC3j3ON1flp>

Conclusion and Perspective

Automata Telegram Bot is an interactive way of exploring automata theory using a popular messaging platform. It enables users to create and manipulate finite automata machines with ease and convenience. [It also demonstrates the power and versatility of finite automata models in various domains of computation such as pattern matching, lexical analysis, parsing, and cryptography¹.](#)

The project has several potential directions for future work and improvement. Some of them are:

- Adding support for other types of automata such as pushdown automata, Turing machines, cellular automata, etc.
- Implementing more operations on finite automata such as union, intersection, complementation, reversal, etc.
- Enhancing the user interface and experience by adding graphical visualization of machines, voice input/output, natural language processing, etc.
- Extending the application to other platforms such as web browsers or desktop applications.

- Evaluating the educational impact and effectiveness of the application on students and teachers.

Automata Telegram Bot is a valuable contribution to the field of computer science education and research. It showcases how automata theory can be made accessible and engaging for anyone interested in learning more about it.