CIS6550 Final Technical Report

Name: Yiwei Guo Email: yguo27@uoguelph.ca
Name: Pingfan Xu Email: pingfan@uoguelph.ca

1. Objectives

As the toughest privacy and security law in the world, the General Data Protection Regulation (GDPR) is applied on a large scale of web applications to protect the EU citizens. Due to the detailed and complicated contents, programmers need to spend a lot of time on following these regulations which increases the development costs. In this project, we decide to implement a backend template of web applications for those who have demands on web application development compliant with the GDPR. Our goal is to build an automated backend to handle all users' data during the data lifecycle (capture, store, use, destroy), which strictly follows the regulations. Using our template, developers can save the time cost of studying the GDPR and manage the users' data by simple API calls.

Out expected outcome for this project is to

1. Establish a generalized web application backend template compliant with the GDPR (Node.js & MongoDB)
2. Implement real backend codes for two specific scenarios as the test cases to validate our designed template actually works

2. Tools

1. **Node.js**

We select Node.js as our backend technology and Express as the framework. Since Node.js can handle multiple requests simultaneously and provide prompt responses, it has been widely used on client-focused web applications, which indicates that our template will have good adaptability.

2. **MongoDB**

As for our consideration of the database selection, both MongoDB and MySQL are good candidates because they both have practical web application usability. Due to our project's time limitation, we decide to implement only with MongoDB rather than implement with both. MongoDB provides a higher speed of the data I/O, and it has excellent scalability that is suitable for using our templates. Besides, the storage format of MongoDB is the JSON format, which provides a better universality of data management.

3. Approach

Our approach is mapping a subset of the GDPR rules (which are the rules that can be reflected on a backend template of a web application) to the API handlers in our template. The specific mapping will be shown in the Template part of next section. The template plays the role of a controller that translates any data-related request from the frontend to actual operations in the database. Additionally, the template does not just process the request but automates the corresponding operation under the GDPR requirements. In our actual creation of the template, we choose to use MongoDB as the database technology.

4. Results

As our final project's outcome, we created a Node.js backend template managing collected data storing in MongoDB under a subset of the GDPR requirements. To complete the template, we mapped the selected GDPR rules to API handlers before our actual implementations. With the template, we also created two concrete implementations of scenarios. In the following subsections, we will describe the template implementation, validate compliance through the mapping from GDPR rules to API handlers in our template, and discuss the scenario actions' corresponding GDRP rules and how they are satisfied implementation of each scenario.

4.1 Template

Our template contains eight handlers that satisfy nine GDPR rules from the capture stage to the destroy stage in the GDPR information life cycle. The template maintains one central collection (Figure1.1) and several service collections (Figure1.2).

4.1.1 API Handlers

1. **Handler For Adding New Record To Service Collection (POST /record)**: When any new record needs to be added to the database, for example a new customer joins a service, this handler will be executed. This handler ensures the target database name, collection name, data needs to insert, data consent expiration date, and data owner's unique identifier are provided and valid. After the completion of inserting the new data to the target collection, this handler automatically creates a new record in the central collection to keep track of the usage of the newly inserted data like the owner's contact info, consent expiration date, etc.
2. **Handler For Getting One User's Record In A Collection (GET /record)**: When the data controller needs to get one user's record in a collection, for example displaying a user's personal info on the profile page, this handler will be called. As long as the valid database and collection name is passed in, the record corresponding to the username in the target collection will be returned to the frontend.
3. **Handler For Getting All Data Of One User (GET /record/all)**: When the data controller needs to get all data belongs to one user, for example, when a user asks for all data the organization collected about him/her with the rights under the GDPR, this handler will be executed. This handler's execution result is the user's data stored in all collections across the whole database. There is an option to choose to return the result as a JSON object or a downloadable JSON file.
4. **Handler For Getting Contact Info Of Data Retention/Consent Expiring Soon Users (GET /contact/expire)**: Under the GDPR, the data controller cannot use the data if the corresponding consent was expired. If the data controller wishes to

continue to use the expiring data, they need to contact the data subject for renewals. A list of soon expiring consents with the data subject's unique identifier, email address, and data collection under these consents will be returned by calling this handler. With the result list, it will be easy for the data controller to format emails of renewal if necessary.

5. **Handler For Executing Database Queries For Internal Usage (GET /record/query)**: The ultimate purpose of collecting data is to use it. This handler provides the capability of executing queries fetching data for internal usage purposes. The internal use of this handler can customize the query and options for execution. The result of this handler would be the result of executing the query with the given options.
6. **Handler For Updating Data Stored In A Collection (PATCH /record)**: In case some stored data needs to be corrected, or the consent expiration date needs to be updated, this handler can be used.
7. **Handler For Deleting All Info Belongs To One User Within A Collection (DELETE /record)**: The GDPR gives the data subject the right to ask for revoking consents and erasing the corresponding records. This handler can be used to revoke consent and erase the data owner's record within a specific collection.
8. **Handler For Deleting All Info Belongs To One User Across The Database (DELETE /record/all)**: Similar to the above handler, this handler is in charge of data erasion. The difference between this handler and the above one is that this one allows the user to erase all of the data owner's information from all collections across the database from a single API call. This handler is suitable for situations like erasion after export all data or revoke all consents as required under the GDPR.

4.1.2 Validate Compliance

1. **Capture (Obtain Consent)**

At the first stage (capture) of data lifecycle, the data controllers are required to **obtain consent** from the data subject from the GDPR. The GDPR (2018) explains in Recital 40 that "In order for processing to be lawful, personal data should be

processed on the basis of the consent of the data subject concerned or some other legitimate basis." As a backend template, we are not in charge of the detailed consent content, but we ask the developers to provide the consent expiration date from their users once they want to insert any users' personal information. Without the consent expiration date, the insert request (handler1) will be rejected. By doing so, we achieve the backend's requirement of obtaining consent from the data subject.

2. **Store (Safe And Secure, Restricted Access, Subject Access Requests)**

For the second stage (store) of the data lifecycle, the data controllers need to have **safe and secure** ways of saving the data subject's information. Under the GDPR (2018), organizations are supposed to take the responsibility to ensure that they provide enough security measures to protect their users' data. So in our template, we choose the MongoDB Cloud service (called MongoDB Atlas) as the database to store information. It offers one of the most sophisticated security controls built in to the cloud development, which reduces the potential risk of data breaches. In this way, we enhance security and undertake this responsibility with the third party.

Besides, the GDPR (2016) specifies in Article 18 that only the authorized people are allowed to access the stored data. Our backend is for the organizations' internal use, and they could implement different authentication methods according to their situation, which could also meet the security requirement as well as **restricted access** to the data.

In addition, the GDPR (2018) describes the **subject access requests** in Article 15. Once the users ask for the data that the organizations have collected on them, the organizations need to provide the held information within a reasonable time. We map this regulation to our (handler3) call, which is used to get all the stored data from all the organizations' services of a specific user. The developer could directly make this API call to hand the users' information over by sending a downloadable JSON file to the frontend, which stratifies the subject access requests.

3. **Use (Consent, Manage Consent)**

As the third stage (use) of the data lifecycle, **consent** is indispensable when data controllers are going to process the data subject's information from Article 7 (GDPR, 2018). According to our method of getting the consent expiration date within a

given period (handler4) from the central database, the developers are able to write scripts to determine whether the users' data could be processed based on the consents' period of validity.

Besides, the GDPR also specifies the requirements for **manage consent**. In Article 7, it declares that the data subject has the right to revoke the consent for all of the processing at any time (GDPR, 2018). Our template provides two calls (handler7 and handler8) to withdraw the users' consent. Once they revoke it, the developers can delete the users' related data according to the different situations by these two API calls. Additionally, we also offer a PATCH call (handler6) for the developers to update the consents in the request database when receiving new consents' expiration date.

4. **Destroy (retention period, right to erasure, portability)**

To the fourth stage (destroy) of the life cycle, the GDPR limits the **retention period**, which has to be "no longer than is necessary for the purposes which the personal data are processed" in Article 5 (GDPR, 2016). When the retention period has expired, all the related data must be destroyed. As our default settings, we suppose the retention period is the same as the consent expiration date (may be different in real situations). As we mention above, we provide the developers with a API call (handler4) to get the records whose consent is about to expire. They could use this together with another delete call (handler7) to achieve the goal of destruction.

Another regulation in this stage is the **right to erasure** (also called the right to be forgotten). In the GDPR (2017) Article 17, it explains that "The data subject shall have the right to obtain from the controller the erasure of personal data concerning him or her without undue delay and the controller shall have the obligation to erase personal data without undue delay." Similar to how we handle the expired retention period, once users ask for clear their personal information, the developers can use the two delete calls (handler7 and handler8), which provides the functionality to erase the related data in single or multiple collections.

Portability is also a regulation that needs to follow. In Article 20, the GDPR (2018) illustrates that the data subjects have the right to obtain their data, which has been provided to the data controllers in a standard format. The developers can combine the ways they have implemented in "subject access requests" of the store

stage as well as the “right to erasure” of the destroy state if need. The idea is first to call (handler3) to gather a downloadable JSON file of all stored data belongs to those users. Then, in some scenarios, the users also request to delete the information after the exporting process. So the developers can use the second call (handler8) to erase the data that has been exported.

4.2 Scenario One

The first scenario is a use case of a French customer Kyle interacts with a telecom carrier Bogers, which has cellphone and credit card services. We assume that Bogers is using our template for its backend data management to comply with the GDPR. In Bogers’ implementation, there are four collections, central (Figure 2.1), profile (Figure 2.2), cellphone (Figure 2.3), and credit-card (Figure 2.4). This section will discuss how the actions in scenario one reflect one or several GDPR rules and how our handlers ensure compliance and automate the management process.

When Kyle is asked for personal information, Bogers needs to ask for his consent to satisfy the GDPR. Once Kyle grants the consent, handler1 will be called to insert his data and record the consent expiration info. Similar API calling will be performed when Kyle is setting up his phone service and registering for his Bogers credit card. The only difference is that different target collections will be passed into the handler. After Kyle got his Bogers credit approved, he updates his payment info for the cellphone service. Then, handler6 will be executed to finish the updates.

When Kyle tries to export all his information from Bogers, the GDPR requires Bogers to export data subject’s all info through a standard format. With the help of the template, handler3 will be called to accomplish this exporting task. After exporting Kyle’s data, he asks for revoking consent and erasing all his data. To complete consent revoking and entire data erasion, handler8 can be called for an automatic finish.

4.3 Scenario Two

The second scenario is a German student’s use case interacts with a research group, SRGP, which has three ongoing studies. We assume that SRGP is using our

template for its backend data management to comply with GDPR. In the SRGP's implementation, there are three collections, central (Figure 3.1), COVID-19-research (Figure 3.2), sleeping-quality-research (Figure 3.3), and height-research (Figure 3.4). This section will discuss how the actions in scenario one reflect one or several GDPR rules and how our handlers ensure compliance and automate the management process.

When Sam decides to contribute to the three researches that require his personal information, the SRGP group needs to ask for three consents from him to comply with the GDPR. Once Sam grants the consent, handler1 will be called to insert his data and record the consent expiration date for all of the three collections separately. The difference among the three insertions is the different parameters from the requirements of each research. After finding that Sam's foot size might contain a typo, the staff will need to find his contact information using handler5 and then contact him to make a correction using handler6. This followed the data accuracy principle of the GPDR (2016) in article 5.

When Sam asks the SRGP group to erase all his information, the GDPR requires the SRGP group to immediately delete all his information. With the help of the template, handler8 will be called to accomplish this deletion task.

Since this COVID-19 research may have a delay, the SRGP group first needs to query all volunteers' contact information from the COVID-19-research collection using handler5. After that, the researchers can contact them and ask for new consents, which will update the consents expiration as well as the retention period using handler6 for those who agree with the new consents and also will delete the personal information for the volunteers who disagree with the new consents using handler7. During the three research processes, the SRGP group is supposed to implement a script to check if the consent has expired or the retention period has passed using handler4. When the condition happens, the corresponding data should be deleted immediately, which can be achieved by handler7.

Reference

General Data Protection Regulation (GDPR). (2016, August 30). Art. 5 GDPR – Principles relating to processing of personal data. Retrieved December 06, 2020, from https://gdpr-info.eu/art-5-gdpr/

General Data Protection Regulation (GDPR). (2018, March 28). Art. 7 GDPR – Conditions for consent. Retrieved December 06, 2020, from https://gdpr-info.eu/art-7-gdpr/

General Data Protection Regulation (GDPR). (2018, March 28). Art. 15 GDPR – Right of access by the data subject. Retrieved December 06, 2020, from https://gdpr-info.eu/art-15-gdpr/

General Data Protection Regulation (GDPR). (2017, June 12). Art. 17 GDPR – Right to erasure ('right to be forgotten'). Retrieved December 06, 2020, from https://gdpr-info.eu/art-17-gdpr/

General Data Protection Regulation (GDPR). (2016, August 30). Art. 18 GDPR – Right to restriction of processing. Retrieved December 06, 2020, from https://gdpr-info.eu/art-18-gdpr/

General Data Protection Regulation (GDPR). (2018, March 28). Art. 20 GDPR – Right to data portability. Retrieved December 06, 2020, from https://gdpr-info.eu/art-20-gdpr/

General Data Protection Regulation (GDPR). (2018, November 14). Recital 40 - Lawfulness of data processing. Retrieved December 06, 2020, from https://gdpr.eu/Recital-40-Lawfulness-of-data-processing/

Appendix

Scenario One

Kyle is an international student from France, who is new to Canada. After settling down, he wants to find a great-value phone plan. There are two carriers Bogers and Rell, which offer good price plans that satisfy Kyle's needs. With his friend's suggestion, he chooses Bogers. He goes to the official website of Bogers and starts creating his profile.

The website first asks for his personal information and the consent for it, for example, name, gender, date of birth, address, etc. After the website shows the success message of collecting his personal information and the corresponding consent, he is then asked to add his payment information and consent. And as the one last step, he confirmed his phone plan, which is a 5GB per month plan named New Customer Great Value Plan with a one-year contract. By far, Kyle successfully finishes his phone service setup.

After several weeks, Kyle saw an advertisement for Bogers' credit card. He is attracted by the big promotion of paying Bogers' bills by Bogers' credit card. Kyle then goes to the official website of Bogers and starts applying for the credit card. The application process requires Kyle's personal information, employment, and income information, and each information's consents correspondingly.

After one week of waiting, Kyle gets his Bogers' credit card approved. In order to get the credit promotion, he goes to the Bogers website and updates his payment information from his previously used debit card to his new Bogers credit card.
Time flies and Kyle has enjoyed using his credit card and the past year's promoted phone plan. However, as time goes by, his one-year promotional phone contract ends. The price of the phone plan goes back to twice the promotional price. He starts querying other carriers for a better price of the phone plan. The back-to-school promotion of Rell gives Kyle a considerable discount on transferring his phone service from Bogers. Kyle thinks it is not hard to transfer his profile and other information from Bogers to Rell because both of these two carriers claim that every stage of their information life cycle is under GDPR regulation. The guaranteed portability of collected

data makes the transferring process as simple as a standard exporting request from Kyle.

Additionally, since the benefits of Boger's credit card is specifically applying to Boger's service bills, Kyle decides to cancel his Bogers credit card as well. Thus, he submitted an exporting information request and an information consent revoking request on Bogers' website. Kyle receives all of his information in a standard format through the webpage. Kyle now can happily continue his phone service with Rell.

Scenario Two

SRGP (Scientific Research Guaranteeing Privacy) group is a German group that does scientific researches and promises research participants' privacy under GDPR regulations. The SRGP group is recently conducting three separate studies in parallel: the relationship between blood types and COVID-19 infection possibilities, the impacts of electronic device usage time on sleeping quality, and the relationship between people's height and foot size.

All three researches collect participants' personal information, including their names, genders, ages, living places, contacting emails, etc. Each research gathers some unique information as well. Since the first study looks at the relationship between different blood types and COVID-19 infection possibilities, it requires the blood types of volunteers who have been tested positive. The second research gathers the number of hours the participant uses electronic devices per day and the number of sleepless nights per month. And the last study collects the volunteers' heights and foot sizes. People can participate in one or several researches by providing their information on the corresponding webpage on the SRGP group's official website. The consent for the SRGP group to access the provided info is granted once the volunteers submit their information. The gathered information has a default consent and retention period expiration date, which both matches each research topic's planned ending date.

After publicly recruiting volunteers for several weeks, each of the three studies receives sufficient valid input data and gradually starts researching. Sam is a fourth-year student at the University of Munich who studies computer science. He contributed his real information to all of the three researches. In a study examining the relationship

between height and foot size, one of the researchers found that Sam had filled his own size as 95, possibly due to a typo. So the investigator contacted him to make the correct sizing change.

In this semester, Sam takes a course that recently talks about privacy invasion. After knowing how serious privacy invasion is nowadays, he becomes very concerned about the information he shared with the SRGP group. He struggled with whether he should quit the researches for several days. In the end, to be safe, he requested the SRGP group to erase all information related to him.

When all these researches are about to end, all retention periods and consent are about to expire as well. The SRGP team had to wipe out the data in the three databases once the expiration date passed. However, the first research related to COVID-19 cannot finish on time because of the rapid change of the pandemic situation. The team of this study needs to have the email list of all volunteers and asks them for an extension of consent and retention periods.

Figures

Figure1.1

_id	uid	email	serviceCollection	consentExp

Figure1.2

_id	field_1	field_2	field_3	...

Figure 2.1

_id	username	email	serviceCollection	consentExp

Figure 2.2

_id	firstName	LastName	gender	dateOfBirth	address	email

Figure 2.3

_id	phoneNum	phonePlan	dateContractExp	paymentInfo

Figure 2.4

_id	employmentInfo	incomeInfo	phoneNum	email

Figure 3.1

_id	userId	email	serviceCollection	consentExp

Figure 3.2

_id	firstName	LastName	gender	bloodType	address	email

Figure 3.3

_id	firstName	LastName	gender	sleeplessNightPerMonth	hoursPerDay	address	email

Figure 3.4

_id	firstName	LastName	gender	address	email	height	footSize