Integrated Platform for Delivering SCG Corporate Administration Services

05/12/2023

Alex Siu, Aria Zhou, Tana Na Nakorn

A project report submitted as part of the Research, Entrepreneurship and Innovation course at CMKL, in partnership with Siam Cement Group (SCG)

Table of Contents

Executive Summary

The Siam Cement Public Company Limited (SCG), like many other large organizations, faces a significant challenge with its current service processes inside the company, which are hindered by redundant manual processes and lack of visibility in service delivery, leading to inefficiency and higher operational costs. The fragmented data storage system makes it difficult to examine and analyze the effectiveness and efficiency of processes across the large organization.

To improve its service processes, SCG needs to engage in process re-engineering to ensure that all service tasks are able to be completed on a one-stop platform such that the process can be quick, accurate, and efficient as well as be monitored to identify any issues and inefficiencies. Corporate administration services should be consolidated into a singular platform where service requests can be made and completed, and the data of this service delivery is stored for management to analyze.

While many platforms for electronic corporate administration services exist today, their services are hardcoded where an adjustment to the workflow process or an addition of a new service process comes with additional fees. This discourages the organization from adding new services and optimizing existing ones which defeats the purpose of shifting towards a consolidated platform for continuous improvement.

Our eCA platform will help SCG to stay on top of fast-changing business trends to recognize new growth chances and encourage continuous improvement. This will allow SCG to accomplish its short-term and long-term objectives all while remaining competitive and creating sustainable growth. By condensing all service processes into one platform, SCG can leverage the abundance of information it produces to remain competitive in today’s economy.

Problem Statement

The Siam Cement Public Company Limited (SCG) is a leading cement and building materials company in Thailand and Southeast Asia. Currently, the company faces a significant challenge with its current service processes inside the company, which are hindered by redundant manual processes and lack of visibility in service delivery, leading to inefficiency and higher operational costs. The performance and delivery of service tasks are not recorded in a centralized database, which makes it challenging to access, analyze and utilize this information for improvement in service delivery. To improve its service processes, SCG needs to engage in process re-engineering to ensure that all service tasks are able to be completed on a one-stop platform such that the process can be quick, accurate, and efficient as well as be monitored to identify any issues and inefficiencies. This focuses on streamlining workflows and consolidating service delivery on a singular platform.

Without effective service processes, SCG can endure serious repercussions, such as higher operational costs, inefficient processes, and lost chances to maximize performance. Service requesters are required to provide redundant information and have longer wait times while service providers have to perform redundant tasks to complete the service. Furthermore, the current system prevents them from swiftly responding to emerging trends in their industry or identifying potential risks as well as overlooked expansion opportunities.

Another issue faced by SCG is the fragmented data storage system which makes it difficult to examine and analyze the effectiveness and efficiency of processes across the large organization. Ideally, SCG should have a platform that can access all the data generated by their different departments. This would allow for efficient data sharing, collaboration, and analysis across the company. However, in reality, data generated by different departments is stored in separate databases, and without an integrated platform, it requires service users and providers to access different data from multiple sources.

Currently, the fragmented service processes implemented by SCG create significant challenges for its employees, who are the primary users of the company's internal systems. This lead to delays and inefficiencies in completing tasks, which can reduce productivity and job satisfaction. Service requesters must spend time searching for necessary information and waiting for responses from other departments, which can cause frustration and negatively impact the overall service delivery. By implementing a consolidated service platform, SCG can streamline service processes and improve the efficiency and effectiveness of its service delivery. With an integrated platform, service requesters will have access to all necessary tools and data to complete their tasks quickly and efficiently. This will not only eliminate time-consuming tasks, but also reduce the risk of errors during manual data entry.

For service providers, the inefficiency of service processes due to the lack of a consolidated service system can have a ripple effect on the entire organization. When service providers spend extra time and effort on data gathering and communication, it leads to delays in response times, missed deadlines, and errors. In addition, the lack of collaboration between different

teams can lead to misunderstandings, duplication of efforts, and even conflict within the organization. The deficiency of an integrated service platform also implies that service providers may have restricted access to pertinent information which could be used to streamline their work processes, resulting in lags as they wait for feedback from one team prior to advancing onto the next phase. This can result in more mistakes being made due to staff making decisions based on insufficient or incorrect details, leading to further delays and lack of productivity.

The fragmentation of service processes causes major problems for SCG management and units charged with continuous improvement: it prevents them from detecting patterns or trends across departments. When tasks are done manually with no records of service delivery and performance, there is no data collected that is able to be analyzed by management. Consequently, the group cannot make informed decisions based on this data which could be a huge asset when making important business choices and streamlining the service delivery process. Furthermore, there is a lack of visibility in the workplace, as managers are unable to clearly assess an employee's performance. Without an official and user-friendly platform, employees tend to communicate through unofficial means such as LINE, which cannot be assessed for service delivery performance as well as carrying the risk of a security breach. The current service delivery process thus limits SCG's growth potential as it is difficult to identify areas of potential improvement.

Corporate administration services should be consolidated into a singular platform where service requests can be made and completed, and the data of this service delivery is stored for management to analyze.

While many platforms for electronic corporate administration services exist today, their services are hardcoded where an adjustment to the workflow process or an addition of a new service process comes with additional fees. Scaling such a platform to provide all services required by a large organization such as SCG will be costly due to the number of services required and the complexity of departments involved. Moreover, this discourages the organization from adding new services to the portal as well as adjusting current service workflows for increased efficiency. A platform such as this will soon become obsolete as it may not cover new services and some current service processes may be seen as redundant as more efficient workflows are developed. This then defeats the purpose of shifting towards a consolidated platform. SCG should be able to create new services and adjust current service processes on the platform without extensive coding.

Our eCA platform will help SCG to stay on top of fast-changing business trends to recognize new growth chances and encourage continuous improvement. This will allow SCG to accomplish its short-term and long-term objectives all while remaining competitive and creating sustainable growth. By condensing all service processes into one platform, SCG can leverage the abundance of information it produces to remain competitive in today's economy. There will be corporate learning in the process of transitioning to the portal, with new understanding of processes that will need to be represented and managed by the portal. The work outlined in this

proposal is essential for boosting performance and efficacy, resulting in a more successful and enduring business.

Technical Background, Literature Review, and Related Work

The idea of a single digital ecosystem based on a software platform, as proposed by Klochan *et al* [1], can be applied in the SCG project to address the challenges of fragmented data storage and inefficient service processes. By implementing a digital platform that can integrate and consolidate data access from different departments, SCG will improve its service processes, facilitate collaboration between teams, and make informed decisions based on data analysis. Additionally, the use of a digital platform can streamline workflows and improve the efficiency of service tasks, leading to cost savings and better employee satisfaction.

Microservice architecture and API gateway are pivotal technologies in the work of Bauer *et al.* [2] on building and operating a large-scale data processing platform for corporate data analytics at scale on an OpenStack private cloud instance. In the context of the SCG platform, microservice architecture and API gateway can be used to facilitate the integration and communication of different service modules and data sources. Microservice architecture involves breaking down the system into smaller, modular services that can be developed, deployed, and scaled independently, allowing for greater flexibility and agility in responding to changes. On the other hand, an API gateway acts as a centralized entry point that routes requests to the appropriate microservices and provides additional functionalities such as authentication, authorization, and caching. We implemented API gateway so SCG can effectively streamline its service processes and consolidate data access from different departments into a cohesive and efficient platform.

Braubach et al.[3] provide a comprehensive analysis of the challenges and requirements for the development of a distributed computing infrastructure for private enterprise clouds. Their approach emphasizes the importance of building an infrastructure that requires minimal installation and administration effort, while supporting independently operated nodes in a dynamic environment. Additionally, the infrastructure provides an intuitive programming model for distributed applications, allowing for transparent application administration and dynamic reconfiguration. The proposed solution, JadexCloud, consists of three layers, including the daemon layer for managing basic cloud resources, the platform layer for application management tasks, and the application layer for supporting application development through APIs and tools. We took this infrastructure into consideration as we build our portal as we aim to facilitate the scalability and robustness as modules can be replaced or reinvented to suit the needs of the SCG in the future.

Bonomi *et al.* [4] created The Creawelfare platform, which is a user-friendly and completely digital solution that doesn't require any local software installation. It leverages the spatial proximity of mutual aid societies and the wide outreach of local cooperative banks, enabling a large number of small and medium-sized enterprises (SMEs) to access welfare plans and involve their workers. This inclusive approach allows SMEs to benefit from corporate welfare (CW) opportunities that were once exclusive to larger companies. The SCG platform will adopt a

similar approach by leveraging digital and spatial proximity strategies to increase accessibility to its services for a broader range of users. This will include incorporating user-friendly digital interfaces and robustness to enhance engagement with the community.

The URRAN system by Shubinsky and Zamyshlaev [5] is a risk-based management methodology for the technical maintenance of railway transport facilities that relies heavily on automation and informatization of data collection and processing. With about 30% of all regulatory documents already automated, the URRAN system is an example of how automation can improve efficiency in asset management. The SCG platform will learn from the URRAN system's approach by exploring opportunities for automation and informatization of data collection and processing to enhance the platform's efficiency in managing technical assets.

Considine [6] discusses the corporate management framework that has been implemented in Australian public administration, which includes program budgeting, corporate planning, performance contracts, program evaluation, and efficiency scrutiny. He stated that the framework has fundamental problems, including inappropriate use of the product format, ignoring political dimensions, denying the value of decentralization, and displaying an unwarranted optimism about technical rationality under central direction. The SCG platform can learn from this article that implementing a corporate management framework alone may not guarantee genuine and lasting reform. Instead of solely focusing on control and efficiency, the platform should also consider other factors such as staff involvement and decentralization. It is important to address the root causes of inefficiency rather than just imposing new management techniques. The SCG platform should also be cautious of the political attractiveness of new management frameworks and ensure that they prioritize administrative effectiveness and policy-making competence.

Enterprise Collaboration Platform (ECP) by Pustylnick [7] shifts services in a company from a system-centric to a collaboration-centric approach, borrowing concepts from social network collaboration to align real employee interactions with the systems that facilitate them. ECP allows for a clear separation of system administrators and their functions from key communication processes within the enterprise. It retains all enterprise communications from email to instant messaging, eliminates the need for external communications, and allows users to administer portions of the content. SCG can leverage this approach by implementing the service platform to align more towards collaboration-centric. This can bring several advantages such as improved communication and collaboration among employees, better control, and increased efficiency and productivity. Additionally, the ability for users to administer portions of the content can allow for greater participation and contribution to important projects and initiatives. However, there may be disadvantages such as the need for training and implementation costs, potential resistance to change from employees, and the risk of information overload if not managed properly.

Ma, Chung and Thorson [8] discuss how China's e-government initiatives aim to foster administrative reform, streamline procedures, and enhance transparency to support economic development. The authors argue that understanding the concept of administrative reform is key

to resolving the apparent contradiction between administrative decentralization and government oversight. The paper provides evidence that administrative reform has been a driving force behind many of China’s e-government applications at both the national and local levels. Downscaling to a company level, SCG can learn from China's e-government initiatives in terms of fostering administration through the use of technology. By streamlining procedures and enhancing transparency, SCG can help improve the efficiency and accountability of services for its employees, which in turn can support its business. However, the drawbacks could include concerns around data privacy and security, as well as barriers to switch.

Methods

Solution Process

Sample Scenarios

We started our process by considering scenarios in which our portal will be able to most benefit the service process. We started with four main scenarios as a basis for our problem. The scenarios we came up with were replacing a staff ID card, department database failure, office equipment maintenance, and employee onboarding. For each of the scenarios, the current service protocol was considered, namely what departments are involved in the service, the type of data classified by each of the departments, and the workflow of the service. Our sample scenario for employee ID replacement is shown below, with the other services given in Appendix A.

Scenario: New staff ID card for employees
Departments involved:

- Corporate Human Resources (CHR)
- Workplace Solutions Office (WSO)

Data Classified by Department:
CHR:

- Personal Information: Name, title, employee number, employee status (full-time, part-time, contract), employee photo, signature, etc.
- Department Information: Department name, location, contact information
- Security Information: Access levels, security clearance, etc.
- Signature Authority: Approval from authorized personnel

WSO:

- Card Information: card expiration date
- Issuance Information: Date of issuance, location of issuance, employee signature confirming receipt

Flow:

1. Issue Request (input) - An employee needs a new staff ID card and submits a request to CHR.
2. CHR - CHR receives the request and verifies the employee's personal information, department information, and security information. CHR then approves the request and forwards the information to WSO.
3. WSO - WSO receives the approved request and uses the information to design and produce the new staff ID card with the appropriate card technology and expiration date. WSO then contacts the employee to arrange for the physical pickup of the new ID card.
4. Physical Card Acquired (output) - The employee goes to the designated location to receive the new ID card, signs for it as confirmation of receipt, and returns the signed

document to WSO. WSO then updates its records with the issuance information, including the date and location of issuance.

Terminology

We define a service as a complete process consisting of individual tasks. For example, in the ID card scenario, the service is creating a new ID card for an employee and the tasks involved are verifying the employee request by CHR and the printing of the ID card by WSO.

We identified four main roles for the users of our portal: requester, provider, manager, and administrator. Each of these roles will have different uses for the portal. A requester is an employee who wants a service performed, a provider completes service tasks, manager is an employee who holds a managerial position and supervises other employees, and lastly an administrator overlooks multiple departments. Employees can have multiple roles. All employees have a requester role so they can use the services provided by the portal. Employees may also have provider, manager, and/or administrator roles based on their job position. For example, an HR officer can be both a requester and provider since they will also be completing HR tasks that are requested on the portal.

Workflow Diagrams

With the sample scenarios in mind, we considered a generalized workflow process that can be easily implemented for multiple scenarios and services. Our main goal was to make the workflow for both the requester and provider feel as organic as possible while also giving them the ability to check the progress of their previous requests. The data of requests being made and completed will also be recorded so that it can be used by managers and administrators for analysis and evaluation to continuously improve service delivery. From that, we created the following diagrams for the workflow of the service requester, provider, and manager.

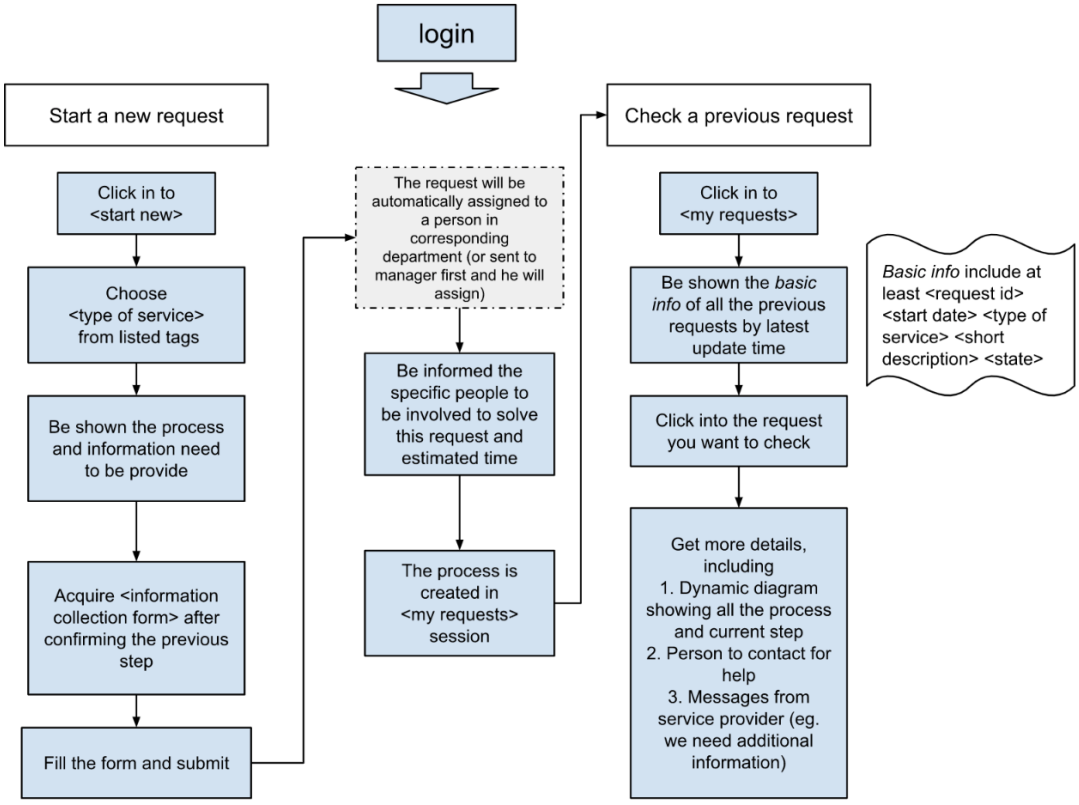


Figure 1: Workflow diagram for requesters

The workflow diagram for the requester given in Fig.1 shows the general procedure for a requester from start to finish. This workflow will require the requester only provide necessary information for the service provider only once without having to reach out to multiple departments by themself. Because of this, the number of steps for the requester remains the same regardless of the complexity of the task, ie. the number of departments involved. The requester will be notified on the platform once a request has been completed and they can also check the status of their previous requests. Once a request has been completed, they can provide feedback to the provider on their service delivery performance.

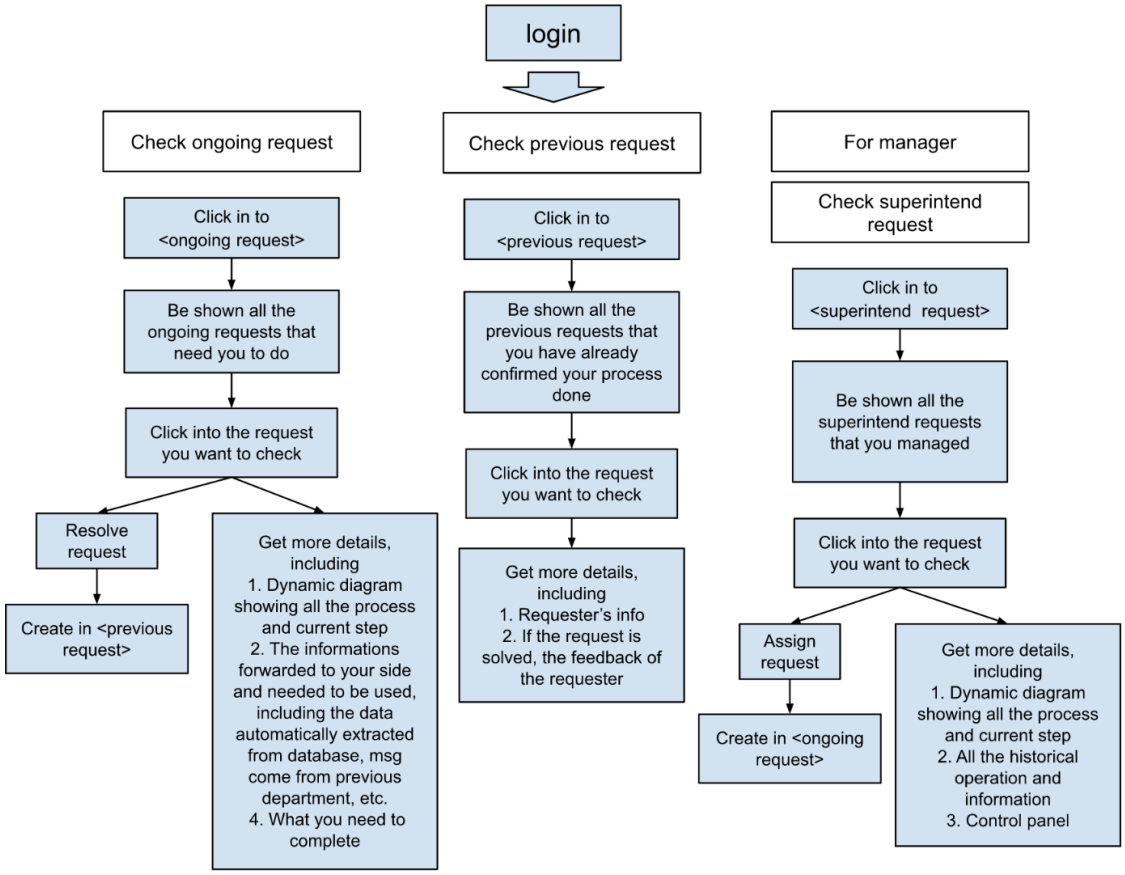


Figure 2: Workflow diagram for providers and managers

The workflow diagram for the providers and managers in Fig.2 shows the general procedure for a provider or manager to check an ongoing or previously completed tasks as well as a manager assigning tasks or checking a request status. Providers can check a request, resolve it once the task is completed, and can see feedback given by the requesters. Only managers can view and assign tasks to providers under their supervision. The system will notify providers and managers when they have a task that needs to be completed.

We considered many approaches to how tasks will be assigned to individual providers in a department. Our first idea was to wait for the manager to assign a task to an appropriate provider. However, this adds an additional step to the request completion process for each task and therefore the system will assign the task to an appropriate provider in the department. Nonetheless, we recognized that some tasks may require the manager to individually assign a provider to the task, therefore these specific tasks will have the additional step of manager assignment.

Use Cases

We made use case narratives for each of the roles of the requester, provider, and the administration based on our workflow diagrams. We use the use case narratives to get a better

understanding of the user experience and capabilities of the portal. From our narratives, we created a use case diagram that expands on the cases for each of the roles. Two use case narratives for the requester portal are given below.

Case 1: Start new request
Main Success Scenario:

1. Requester selects start new request
2. System provides list of types of services
3. Requester selects service
4. System asks for information needed for service
5. Requester enters needed information
6. System asks to confirm
7. Requester confirms
8. System creates request inquiry in previous requests

Ext (a) requester cancels
6a. System asks to confirm
7a. Requester selects cancel instead of confirm
8a. Inquiry is not created
9a. Return to step 2

Ext (b) requester provides incorrect/incomplete information
4b. System asks for information needed for service
5b. Requester enters incorrect/incomplete information
6b. System returns error message
7b. Return to step 4

Case 2: Check ongoing request
Main success scenario

1. Requester selects check previous requests
2. System provides list of ongoing and completed request inquiries by latest update time with basic information such as request id, request date, type of service, request status
3. Requester selects a request inquiry
4. System displays more details of request inquiry, such as current step, contact person, messages from provider
5. System asks to accept completed request
6. Requester accepts completed request
7. System asks to close inquiry
8. Requester confirms
9. System inquiry status changed from ongoing to completed

Extension (a) follow up
5a. System asks to accept completed request
6a. Requester selects follow up

7a. System prompts requester for information needed for service
8a. Requester enters needed information
9a. System asks to confirm
10a. Requester confirms
11a. System sends the request back to the same provider along with the request history
12c. Return to step 4 of use case “start new request”

Extension (b) request is already completed
4b. System displays more details of request inquiry, such as current step, contact person, messages from provider
5b. Inquiry status is displayed as completed

Extension (c) provide feedback
9c. System inquiry status changed from ongoing to completed
10c. Requester selects provide feedback
11c. System prompts user to write feedback
12c. Requester write feedback
13c. System sends feedback to provider

The use case diagram for the requestor portal is shown in Fig.3. The main success scenarios of creating a new request and checking a previous request corresponds to the use case diagram with the extensions covering other possible cases. The diagram then extends beyond our narratives to cover other cases as well.

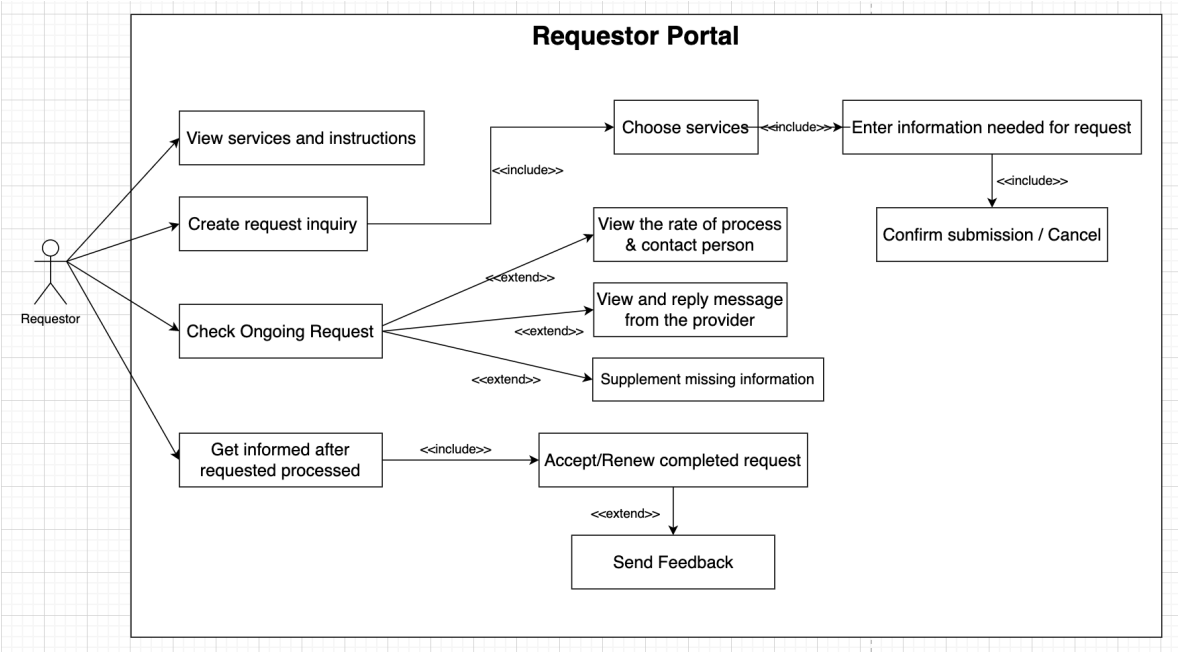


Figure 3: Requester portal use case diagram

We follow a similar process to create a use case narrative and diagram for the provider portal. A use case narrative for the provider portal is given below and the use case diagram is shown in Fig.4.

Case 1: complete task
Main success scenario

1. System sends the request inquiry notification to the service provider
2. Provider accepts request inquiry
3. System prompts provider to complete task
4. Provider completes task
5. System asks to confirm
6. Provider confirms
7. System sends completed task to requester

Ext (a) inquiry requires more than one provider (each provider completes their part before it gets sent on to the next provider)
6a. Provider confirms
7a. System sends request inquiry with current completed task to next provider
8a. Return to step 10

Ext (b) provider does not accept inquiry
1b. System sends the request inquiry notification to the service provider
2b. Provider does not open the inquiry within a given time limit
3b. System sends second request inquiry notification to provider
4b. Return to step 10

Ext (c) provider does not accept inquiry more than once
1c. System sends second request inquiry notification to provider
2c. Provider does not open the inquiry within a given time limit
3c. System sends third request inquiry notification to provider and the provider's manager
4c. Return to step 10

Ext (d) provider cancels
5d. System asks to conform
6d. Provider selects cancel rather than confirm
7d. Return to step 11

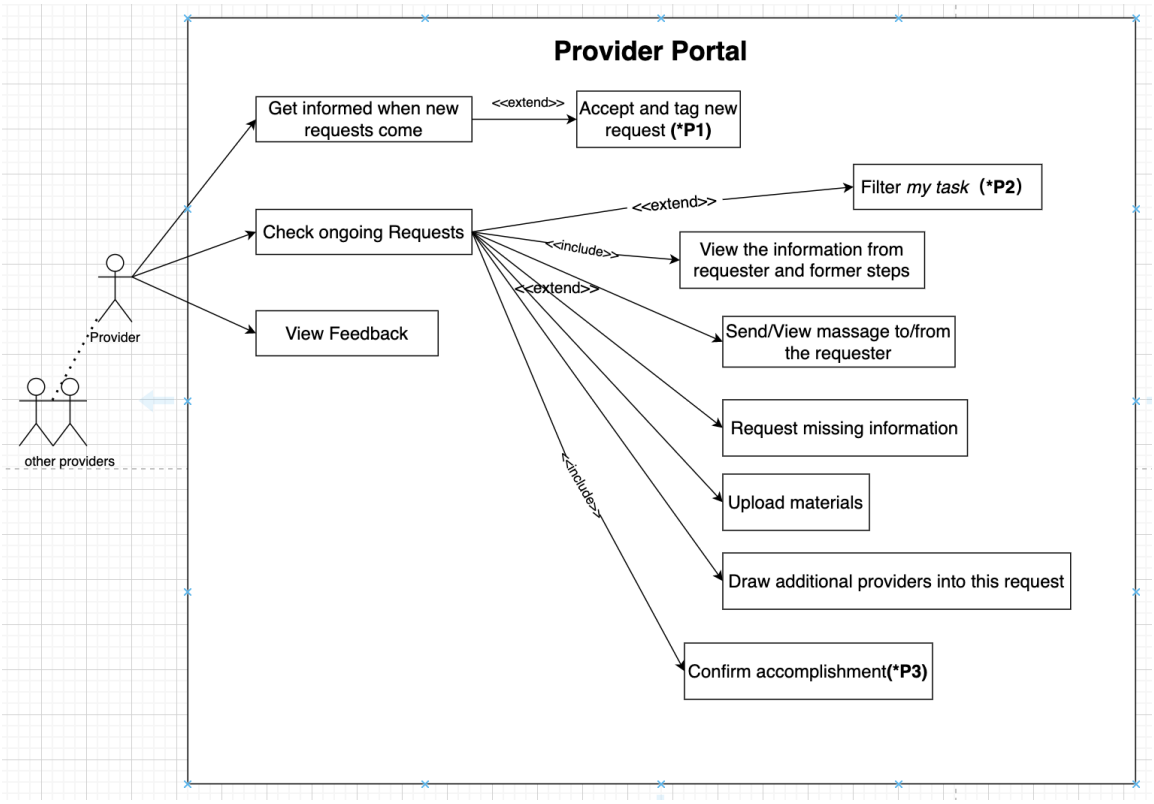
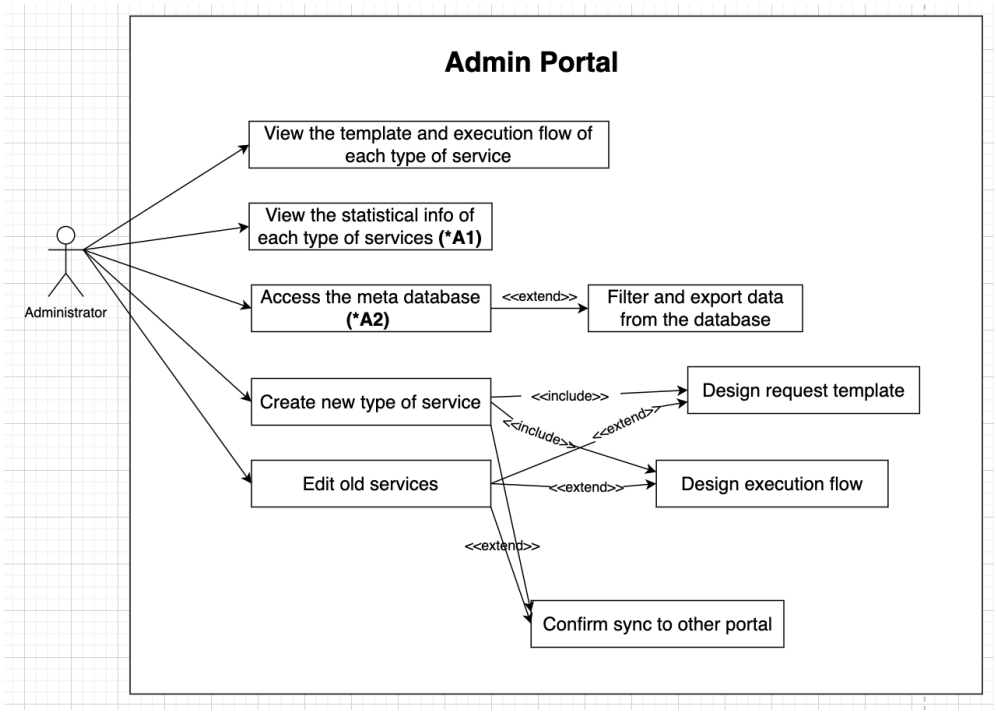


Figure 4: Provider portal use case diagram

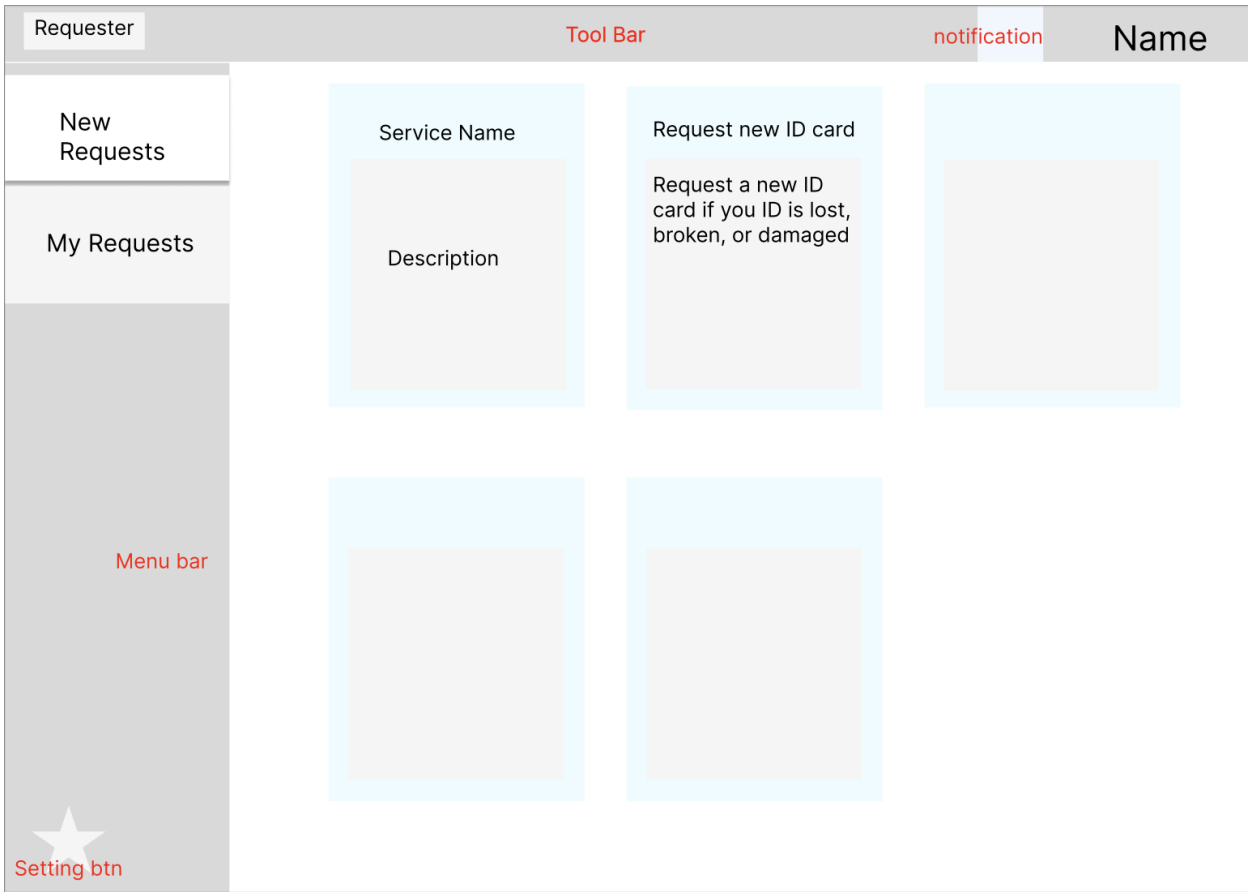
Use case narratives were also written as a basis for the admin portal use case diagram, however many adjustments have been made to the main success scenarios since then. The use case diagram is given in Fig.5. and the use case narratives for the admin portal are given in Appendix B.

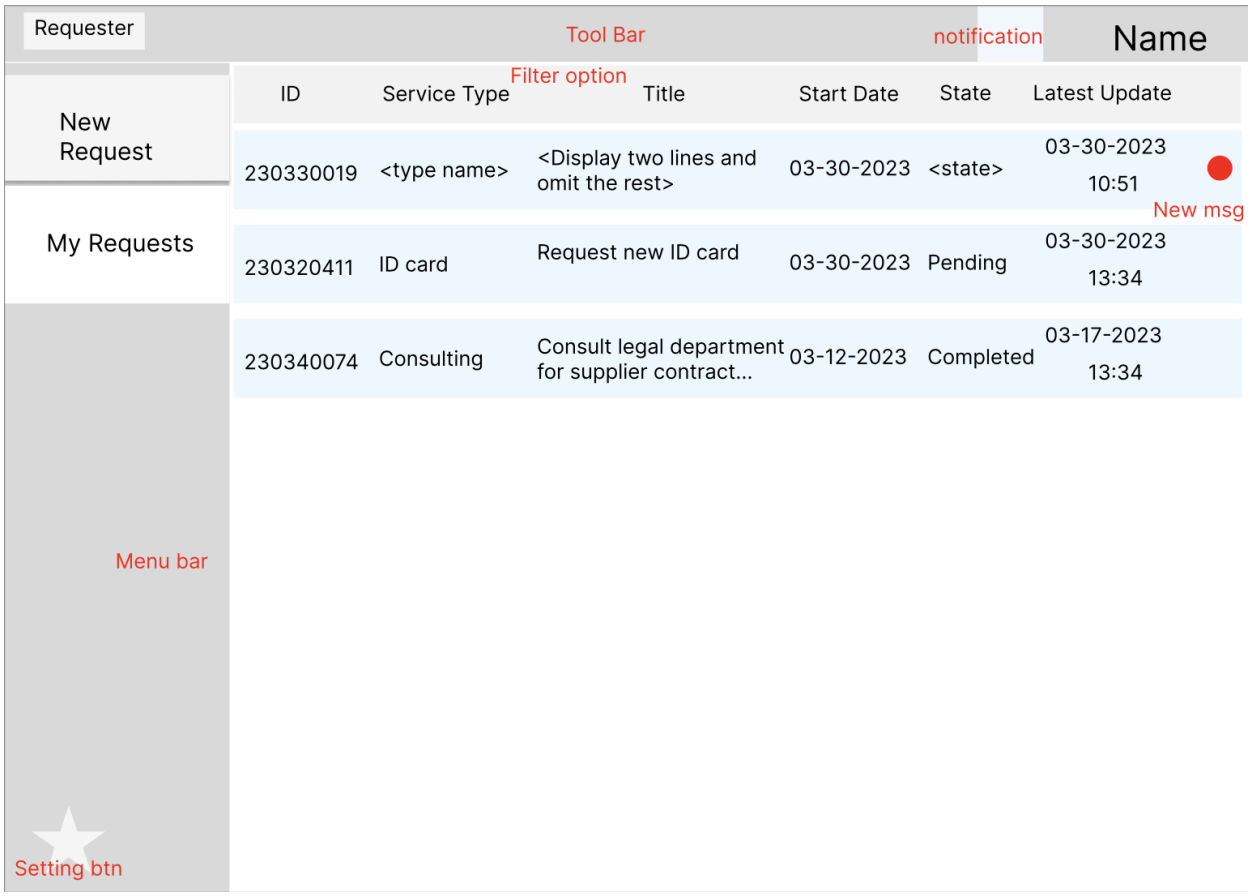


A1: The information includes <request amount>, <average processing time>, <request peak time> etc.
A2: Contain the details of previous requests, including all historical information
Figure 5: Admin portal use case diagram

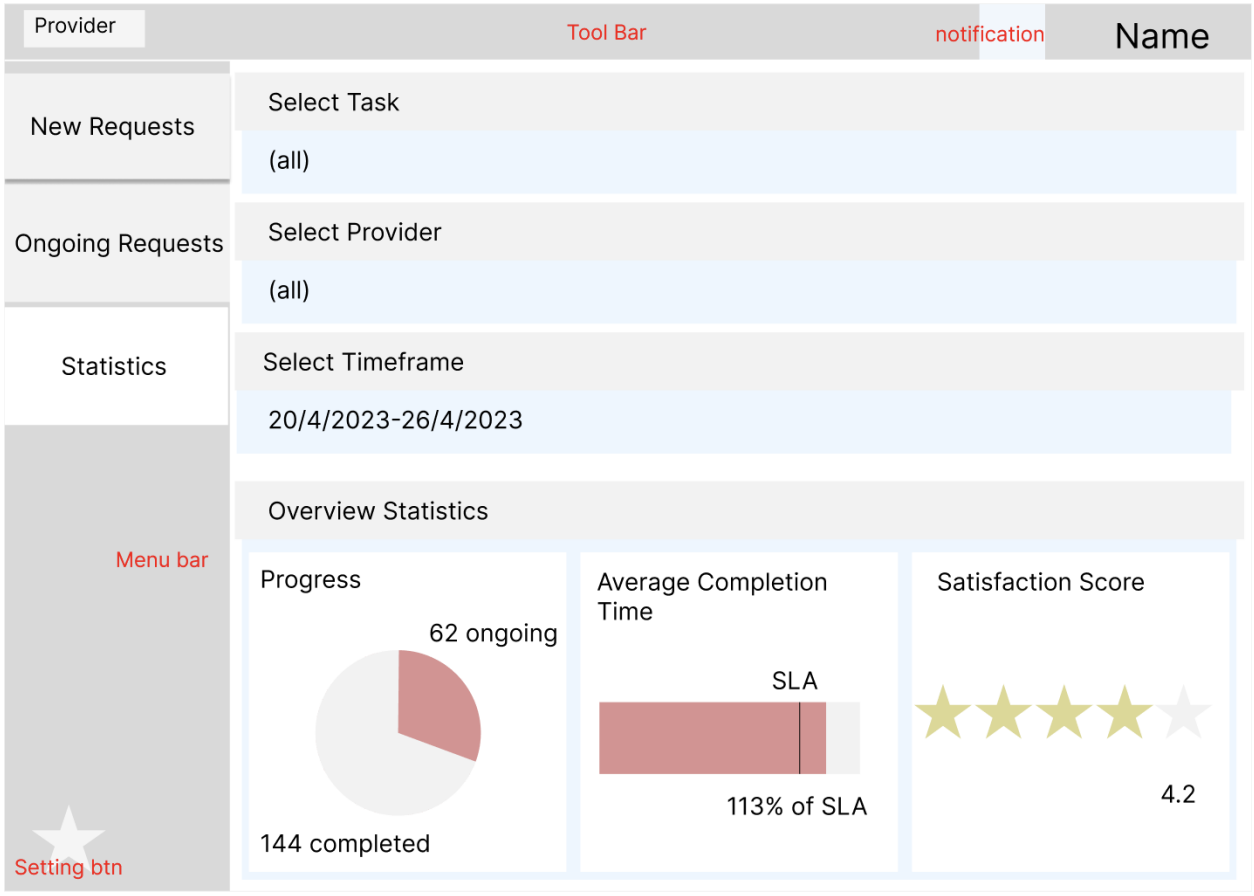
Portal UI for Requester, Provider, Management, and Admin

Figma was used to sketch out a sample UI for the requester and the provider. The UI design we implemented on figma was largely based on the functionality we have defined with our use case diagrams. The main pages for requesters to start a new service request and check previous requests are shown below. The full figma UI portals can be found in Appendix C.

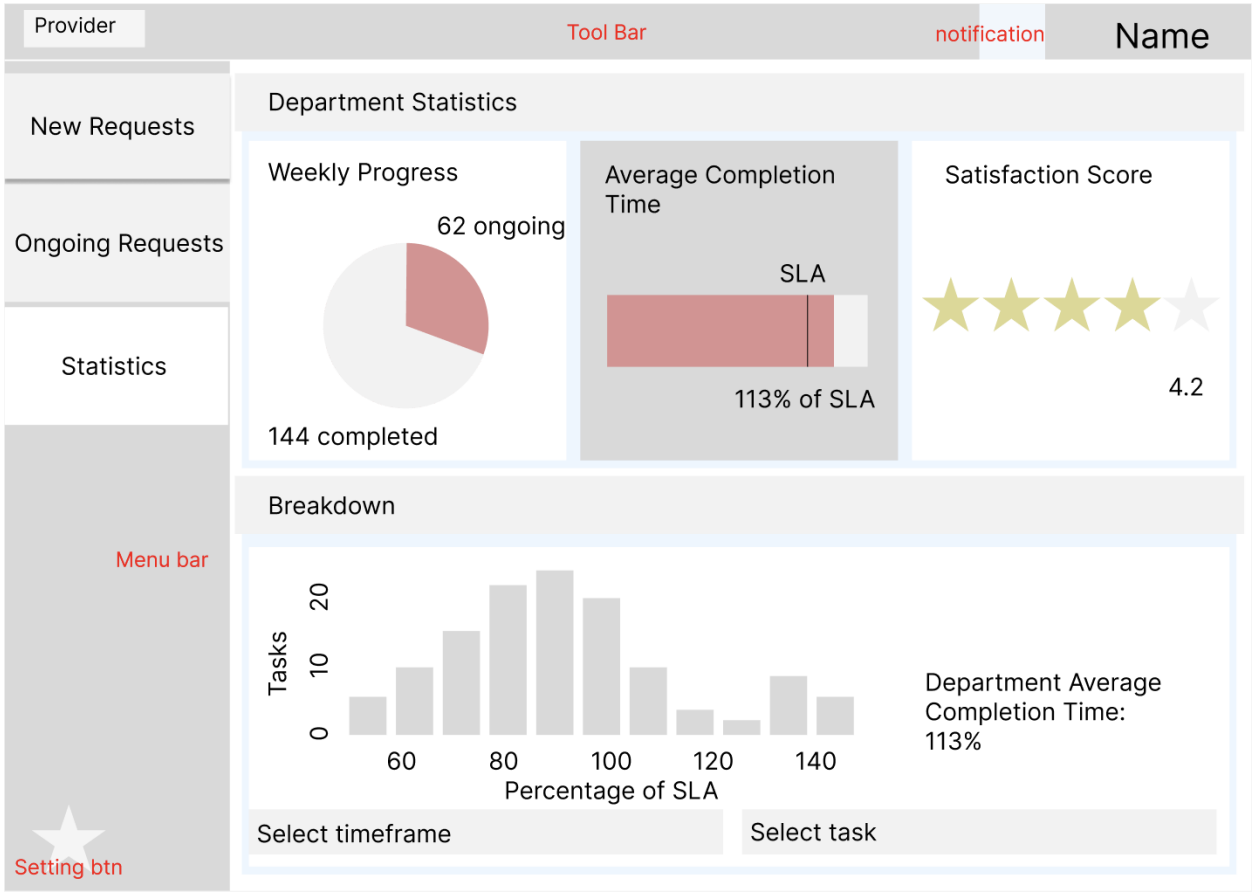

Requester
Tool Bar
notification
Name
New Requests
My Requests
Menu bar
Setting btn
Service Name
Description
Request new ID card
Request a new ID card if you ID is lost, broken, or damaged



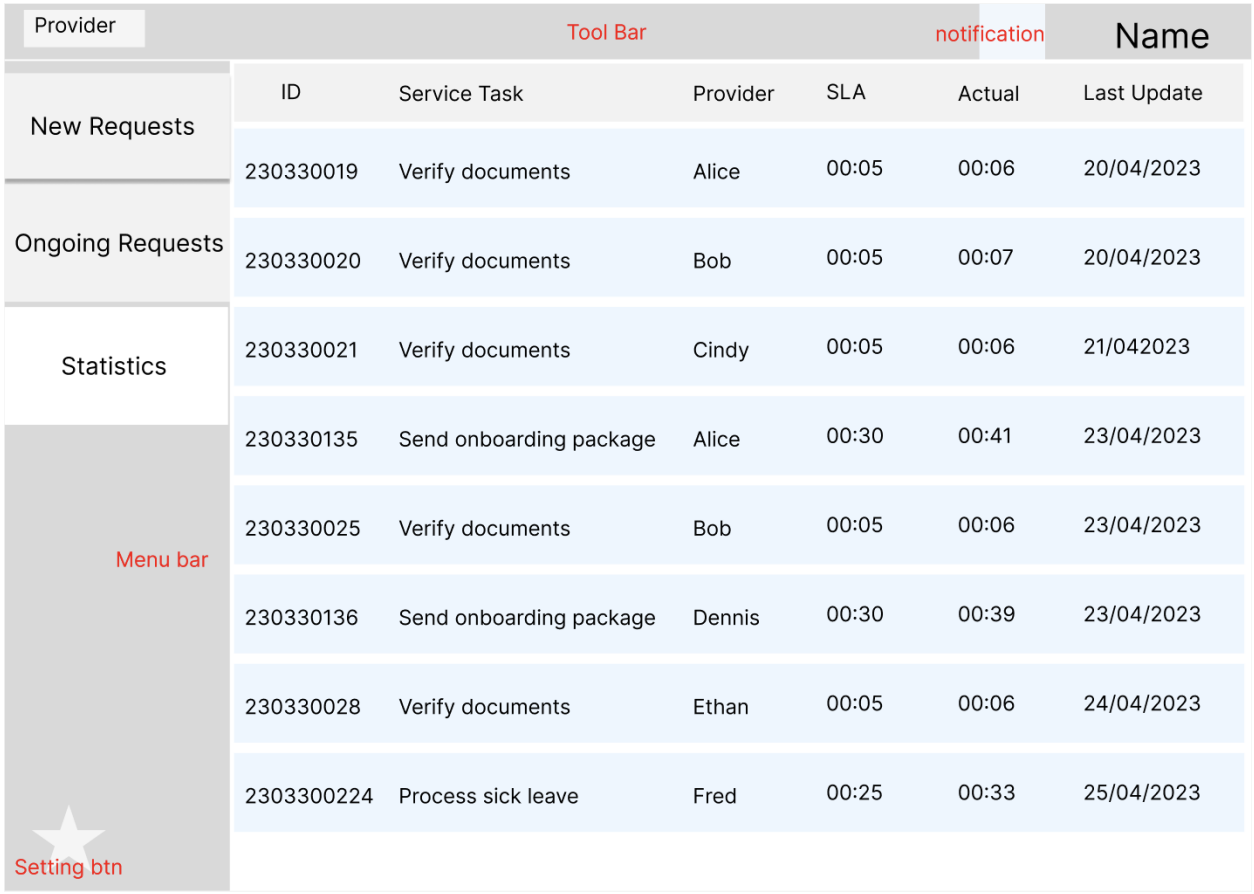
We created a dashboard for management and administration to be able to see an overview of a department's performance in fulfilling service requests. The main functionality of this dashboard is to offer a visual summary of key performance statistics such as the percentage of uncompleted versus completed tasks, average completion time as a percentage of expected completion time (that is, Service Level Agreements), and average feedback ratings. The data that each manager and admin will have access to will be determined by who they are supervising. A sample dashboard UI for management is shown below.



The manager will be able to filter tasks shown on the dashboard as well as click on the diagrams for more in depth details.



Looking at the dashboard completion time provides a histogram of tasks and their completion time as a percentage of the expected completion time. When they click on a histogram bar, they will be presented with a list of tasks corresponding to that bar.

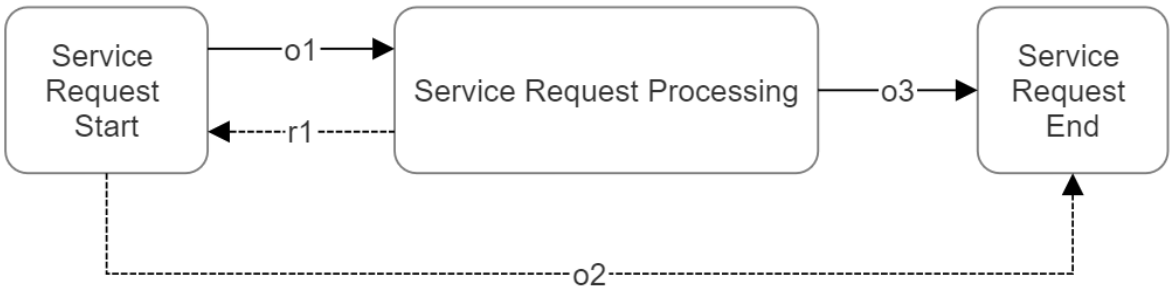


This can help management identify any issues with task completion.

All figma UI portals are given in Appendix C.

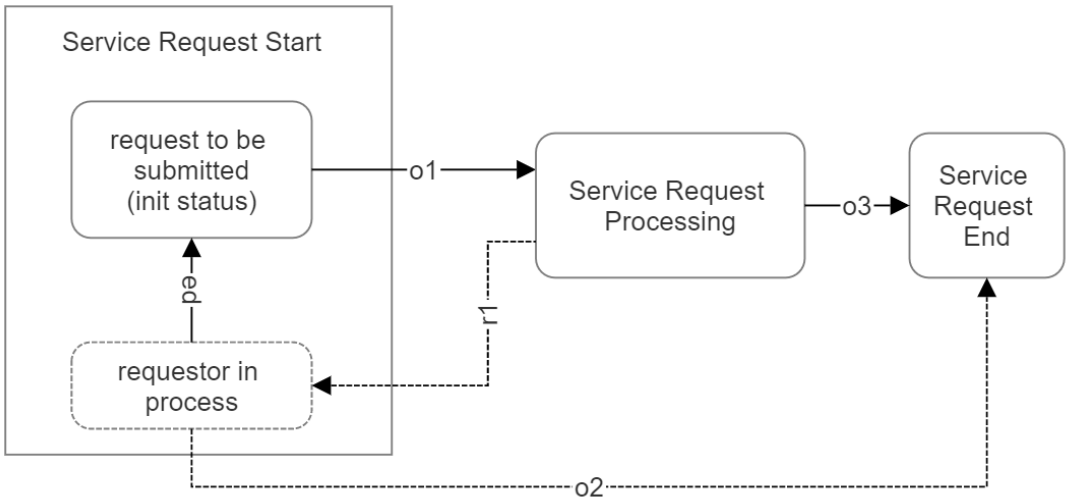
Low Code Solution

A low code solution will enable administrators to create new services without extensive coding, therefore it can be done internally. To achieve a low code solution, we made a generalization for the service flow structure in state diagrams.

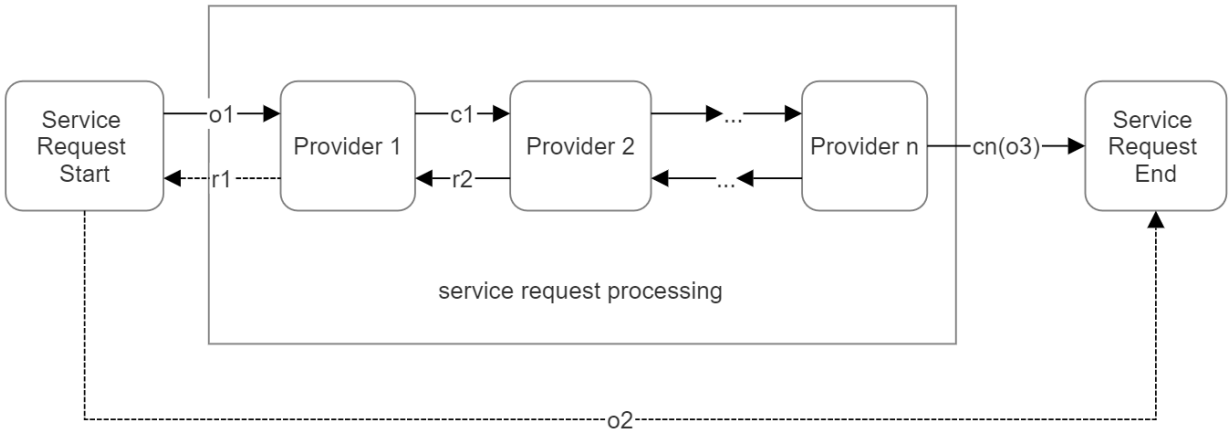


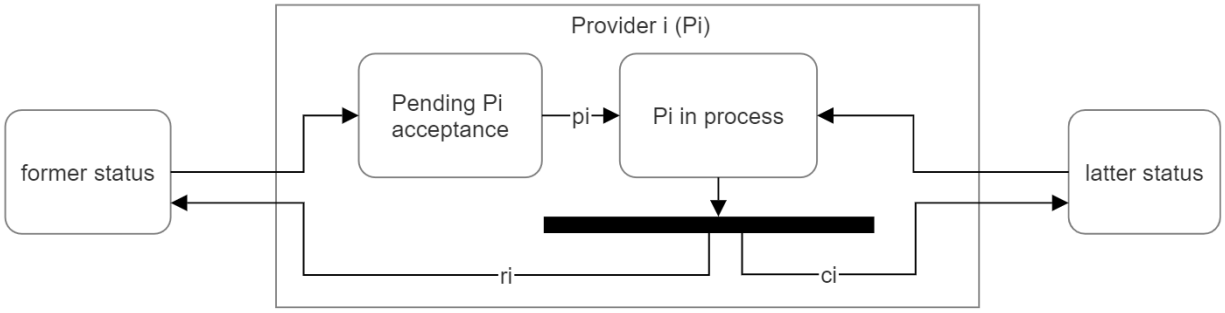
From the broadest view, a service consists of three main states: Service Request Start, Service Request Processing, and Service Request End. Their relation is shown in the following diagram.

In the Start state, the requestor is the only role to make action. When it comes to the Processing state, the providers will be the actor and push the progress. After the request is finished by providers, it enters the End state and waits to be closed. Normally, the state of a request changes from Start to Processing to End, but we allow a reverse from Processing to Start in case that provider detected any mistakes from the receiving request and wanted the requestor to modify it. We also considered the situation that the requestor doesn't want to continue this request, in which case the state will change from Start to End directly.

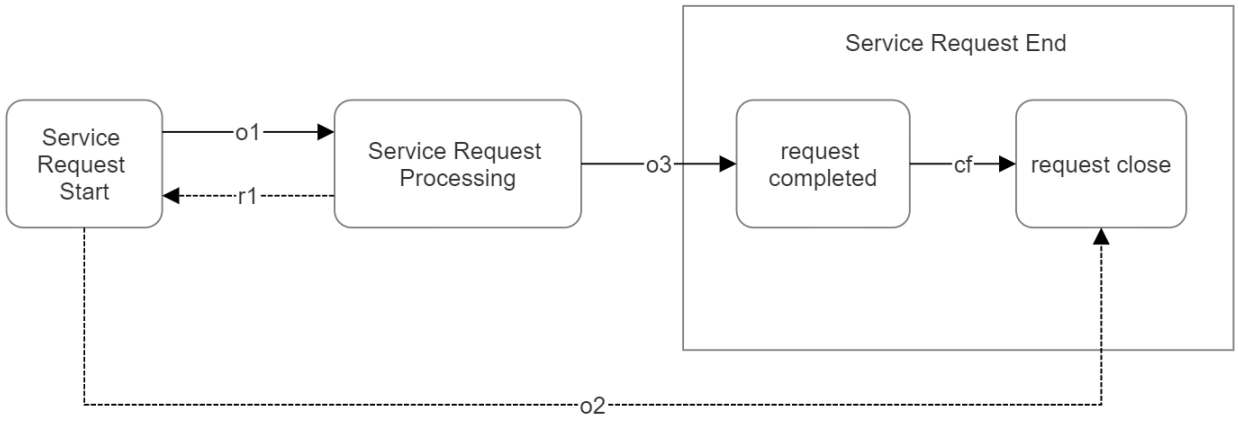


To be specific, the Service Request Start state consists of two inner statuses. When a new request is started for the first time, it goes into the init status and waits for the requestor filling the required information and submitting. <o1> denotes the action that the requestor submits (or resubmits) the request. For the administrator to create a service, he or she needs to define what information the requestor needs to provide. If one request is returned by the requestor, it holds the requestor in process status and waits for the requestor to edit. <ed> denotes the action that the requestor finishes editing.





The processing state contains a series of different providers. When a request is forwarded to a provider Pi, it gets into the status of Pending Pi Acceptance. Pi needs to accept this processing task to activate <pi> and then start operation. If Pi somehow cannot process it, for example, spotting insufficient or wrong information from former providers or requestor, Pi can return this request back (<ri>). Otherwides, if Pi completes its processing step, Pi will forward it to the next status(<ci>). For an administrator creating a new service, he or she can define what a provider should do in its process and easily insert this provider into the queue using APIs.



When the last provider completes processing, the request will reach the End state. In this state, it is waiting for <cf>, which means the requestor's confirmation or a time up. The request will close after then.

The sample backend for our low-code solution is shown below.

1. service_define()
create a new table for the service, containing a <public_properties> field that includes common arguments of all services. (see the service data structure)
define some basic elements of the service, including <service_name>, <service_description>, <processing_flow>, etc.

2. request_define()
create a <requester> field in the above table

define what information is needed from requestor to submit this request

3. provider_define()
create a new <requester> field in the above table
define the required processing operations of a provider

4. provider_insert()
insert a defined provider into a certain position of the processing flow

5. end_action()
define the close action of a completed request

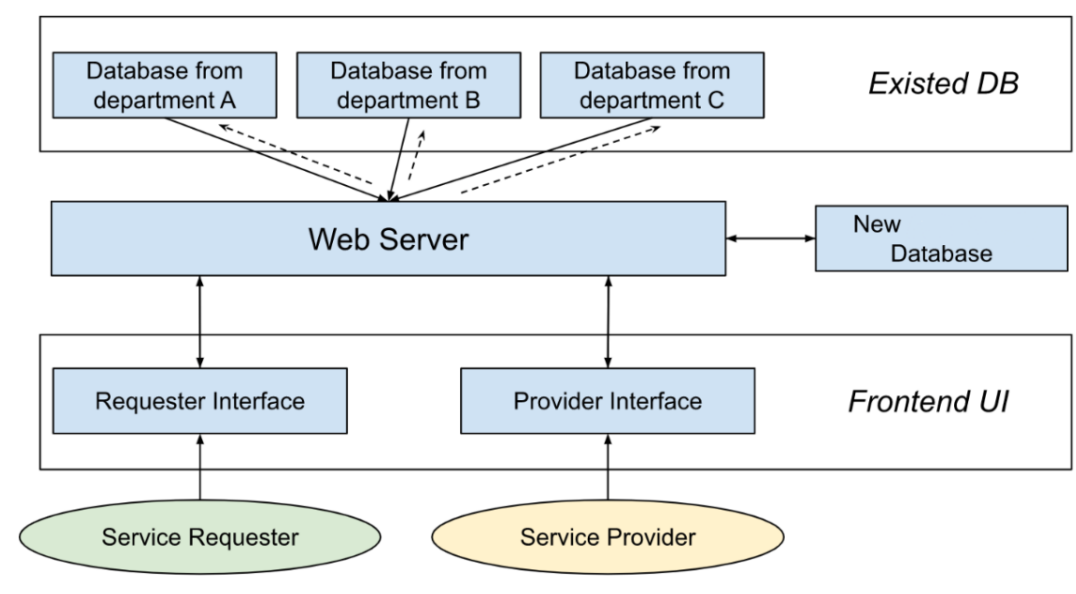
6. return_action()
define the return action (ri) of a provider

7. complete_action()
define the complete action (ci) of a provider

Discontinued Approaches

Consolidating Databases

At the start of the project, we considered migrating all the databases from their existing silo into our new portal by using the eCA portal as a ledger to keep track of services completed as well as save a copy of the data that is used for each of the services.

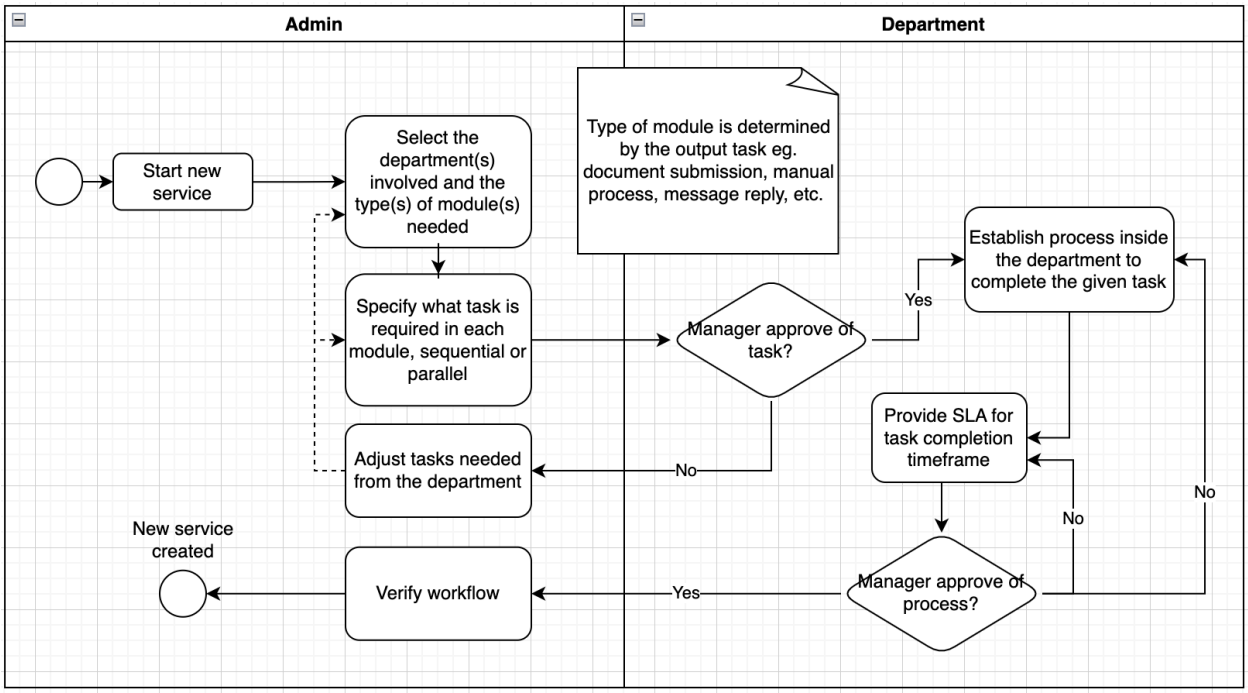


This hinges on the hope that we will be able to directly access the data in the various departments involved according to the above diagram. However, the databases for each of the different departments contain vastly different types of data as well as having different storage methods. Eventually, this approach was abandoned in favor of our current approach.

In the end, we decided to build our portal using the CHR database as a starting point since they already have all the necessary information about all of the employees in the company. Then, when a request is initiated, the portal will only prompt the providers in various departments to complete their task as opposed to accessing their database directly.

No Code Solution and Workflow

Our first approach was to have a no code solution. We envisioned that administrators will be able to create new services by selecting modules for each of the tasks required for the service. Then, the request for approval of this task will be sent to their respective department manager. The department will establish their own internal process for the completion of that task. The use case narrative for the admin is given in Appendix B1 and incomplete use case narratives for the department manager are given in Appendix E. The process workflow is shown in the diagram below.



After more consideration, we found that this workflow would not be realistic and is also potentially redundant. Administrators are likely not working in a department that is providing services on this portal, therefore they will not know what the service delivery process is like for the service they want to create.

It is more likely that the department manager will approach the administrator with a plan to create a new service for their department. The manager will know all the necessary information and tasks required for the completion of the service and convey that to the admin. Therefore, the department side of the workflow will become redundant as the manager will have preapproved of the service because reaching out to the admin in the first place.

Our no code solution was discontinued because of time constraints. Implementation of the no code solution would require an extensive amount of time to complete. Given our limited time for this project, we ultimately decided on a low code solution as it is more feasible to achieve in the given time frame.

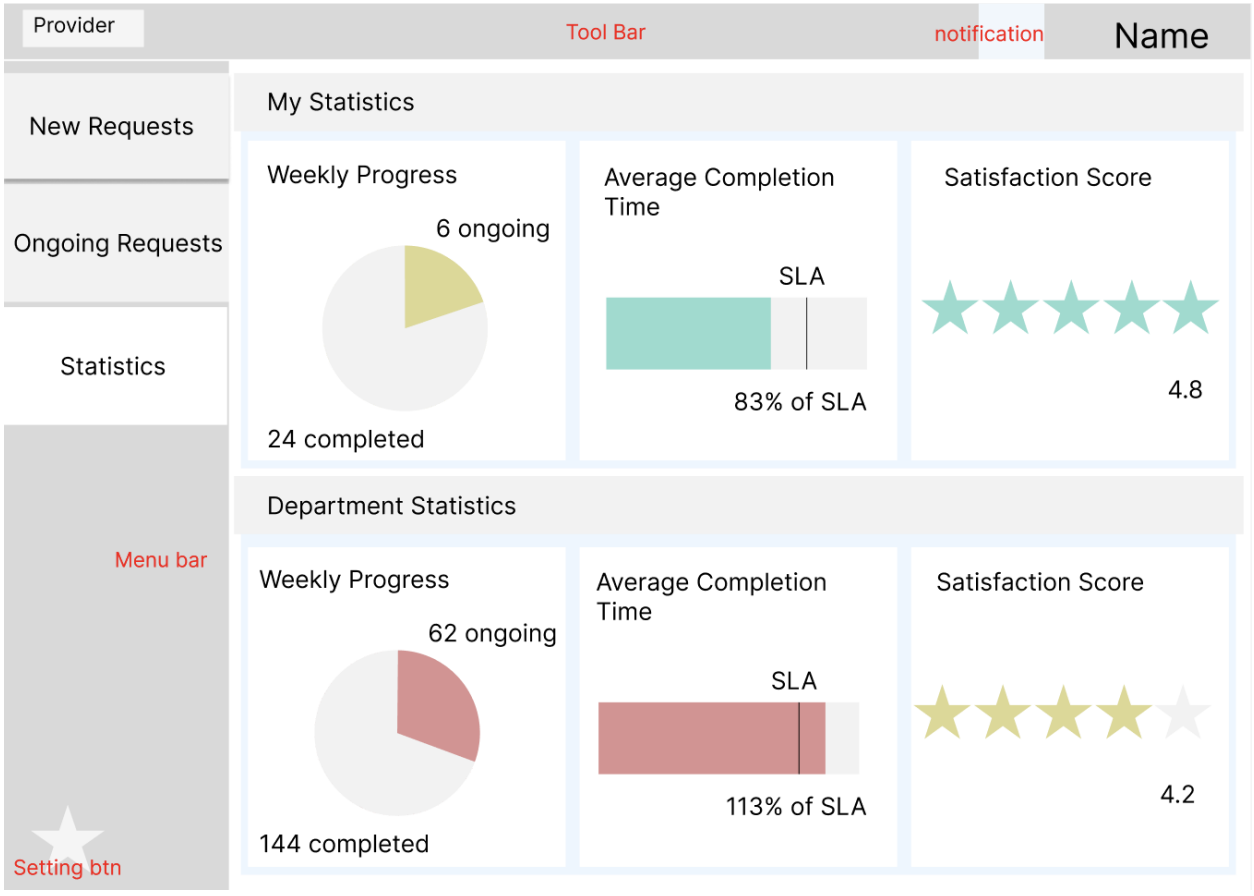
Solution Description

Big Picture

Our solution is to create an eCA portal prototype that will increase productivity through more efficient workflow processes and improve the workplace environment by tracking and assessing employees' performance. We have also created an overview of a gradual transition process of SCG to shift towards this platform.

Requesters will have a user-friendly portal to submit service requests with redundant tasks removed. They will receive faster service with a reliable estimated service delivery time and are able to keep track of what process their service request is in.

Providers will have a single platform to complete various tasks as well as a decreased workload from the removal of redundant tasks. They can keep track of their previous tasks as well as see a statistics dashboard of their service delivery performance compared to the average performance in their department. This provides greater awareness of personal performance as well as visibility within the department.



Managers will be able to easily track and assess their employees' performance using the statistics dashboard. With all service communications done in the portal, managers can also monitor chat logs to assess employee performance. These features can be used to identify choke points in service delivery which will make it easier to improve service delivery.
This also promotes a performance based culture in the organization as employees.

Administrators will be able to achieve continuous improvement using this portal. They will have all the capabilities of the managers while also being able to create no service without extensive coding with the low code solution. This will encourage the optimization of workflow processes as they are able to adjust the workflow of existing processes at any time. The portal's recording of service delivery data helps lessen issues stemming from organizational silos by providing admins with data on all the departments' service delivery. They are also ensured that employees are less likely to use informal means of communication to request and complete services, leading to greater data security.

Technical

This technical summary of the prototype product highlights the development of a SCG service portal using Angular as the frontend technology and AWS as the primary platform for backend services, security, and deployment. The portal was successfully hosted using AWS S3 bucket, while AWS Cognito, AWS API Gateway, and a PostGreSQL database connected to a GoLang backend formed the crux of the backend services.

1. Development in Angular and Hosting on AWS S3

The web portal was developed using Angular, a platform for building web applications. Angular's robustness and versatility made it an ideal choice for the development of the portal. Its modular architecture allowed for the creation of reusable components, increasing the efficiency of the development process.

Once the development phase was complete, the web portal was hosted on an Amazon S3 bucket. Amazon S3 provides a highly scalable, reliable, and low-latency data storage infrastructure at a low cost. The bucket was configured for static website hosting, with access permissions managed via AWS Identity and Access Management (IAM), allowing for a highly secure and easily accessible web portal.

2. User Credentials Control via AWS Cognito

The portal's user credentials are managed via AWS Cognito, a powerful user directory service that enables user sign-up, sign-in, and access control. The service allows for the creation of User Pools, essentially user directories, that provide a secure and scalable solution for user management.

The User Pools have a built-in feature called Tags, which allow for the assignment of metadata to users. These tags can then be used to control different levels of authorization when calling the same API endpoint. For instance, different user roles (admin, editor, viewer) can be tagged with different access rights, ensuring users can only perform actions they're authorized for.

Moreover, these User Pools aid in mapping the relationship between members inside a company, allowing administrators to manage users and their access seamlessly.

3. AWS API Gateway

The APIs for the portal are managed using AWS API Gateway, a fully managed service for creating, deploying, and managing APIs. This service acts as a "gatekeeper" for the backend services, handling traffic management, API version control, and security.

The integration of AWS API Gateway with AWS Cognito provides a secure environment for the APIs. It allows for the authentication and authorization of API requests based on the role assigned to a user in AWS Cognito. This ensures only authorized users can access the services.

Benefits of using AWS API Gateway include its scalability to handle any amount of incoming API traffic, the ability to monitor API usage, the provision of API lifecycle management capabilities, and the capability to run multiple versions of the same API simultaneously.

4. Backend Implementation in GoLang and PostGreSQL

The backend services of the portal are implemented using GoLang, known for its simplicity, efficiency, and high performance. GoLang's inherent concurrency handling makes it well-suited for the development of scalable applications.

The GoLang backend is connected to a PostGreSQL database, a powerful, open-source object-relational database system. PostGreSQL's robustness and advanced features like Multi-Version Concurrency Control (MVCC) and point-in-time recovery make it a suitable choice for the web portal.

The combination of GoLang and PostGreSQL allows for a highly efficient and reliable backend. GoLang's performance benefits, combined with PostGreSQL's data integrity features, enable the creation of a fast and secure data storage and retrieval system, ensuring a smooth user experience for the web portal users.

In conclusion, the use of Angular, AWS services, GoLang, and PostGreSQL in the development of the web portal has resulted in a secure, efficient, and scalable web application that is well-equipped to handle the needs of its users.

Final Results

We have achieved a prototype portal that is capable of scaling up to solve SCG's main issues with corporate administration services. Our final deliverables are the portal prototype and transition process guideline.

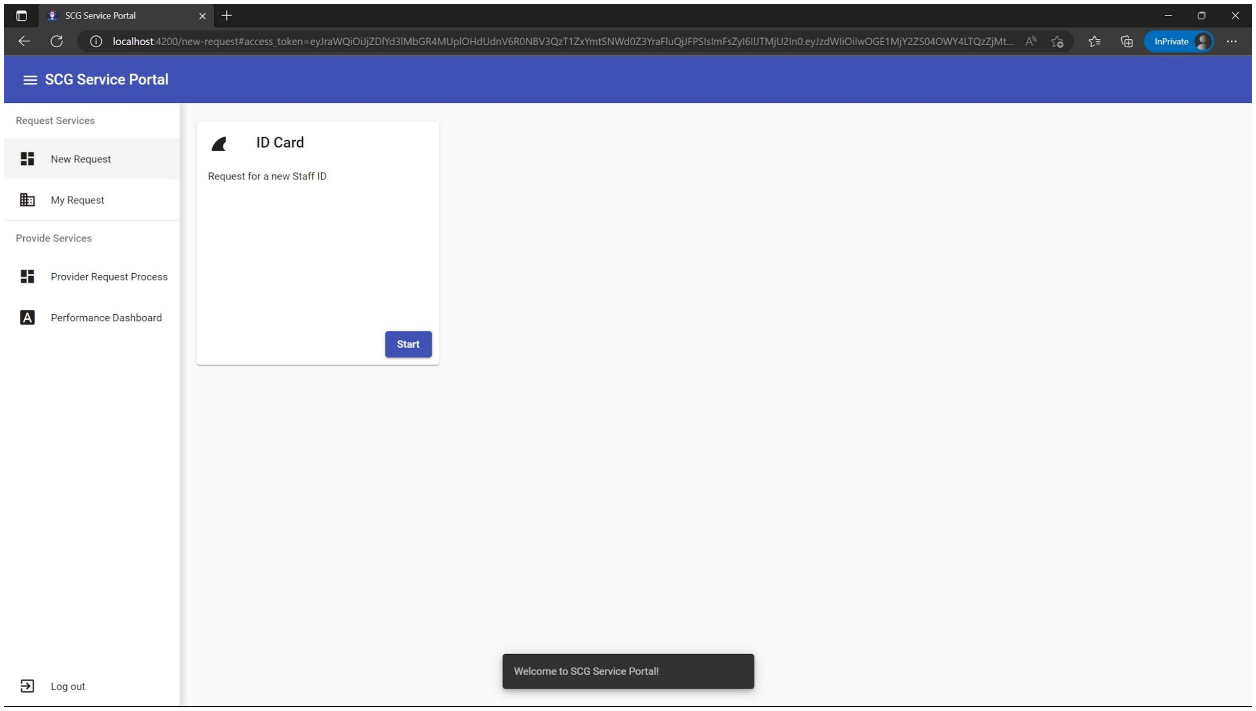
The prototype is able to help requesters and providers improve their work efficiency by reducing redundant tasks and increase visibility by tracking their service requests and performance. Although we were not able to consolidate the databases of the different departments, our portal still allows managers and administrators access to service delivery data through the use of the statistics dashboard. This data will help them analyze and assess current workflows and performance to better identify ways to improve service delivery.

Prototype Walkthrough

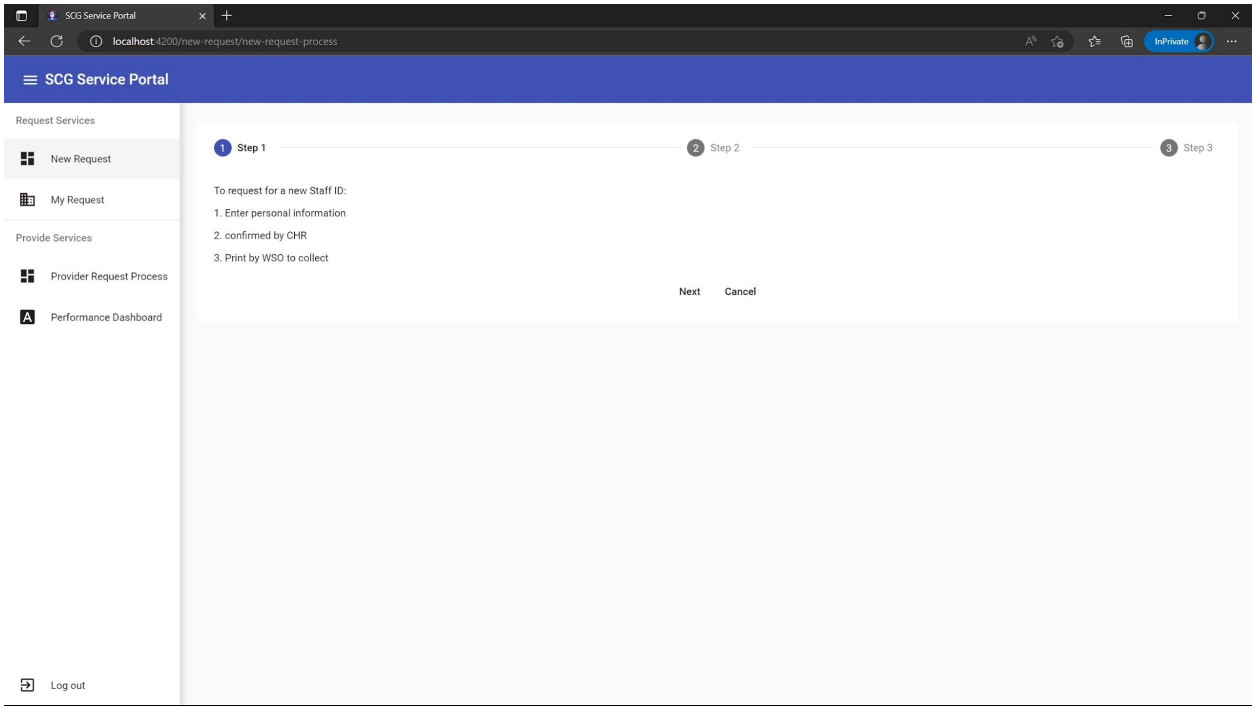
The API and backend codes for our prototype can be found in Appendix F.

Requester

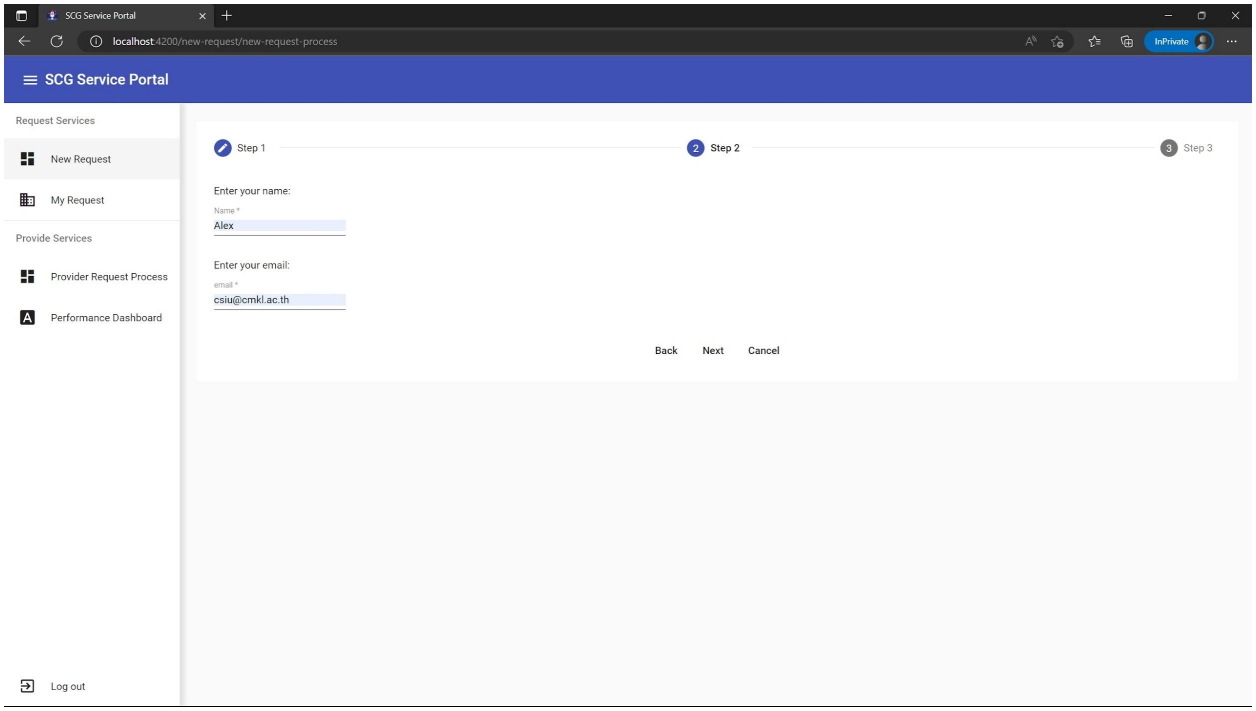
The requester can start a new request by selecting one of the services provided.



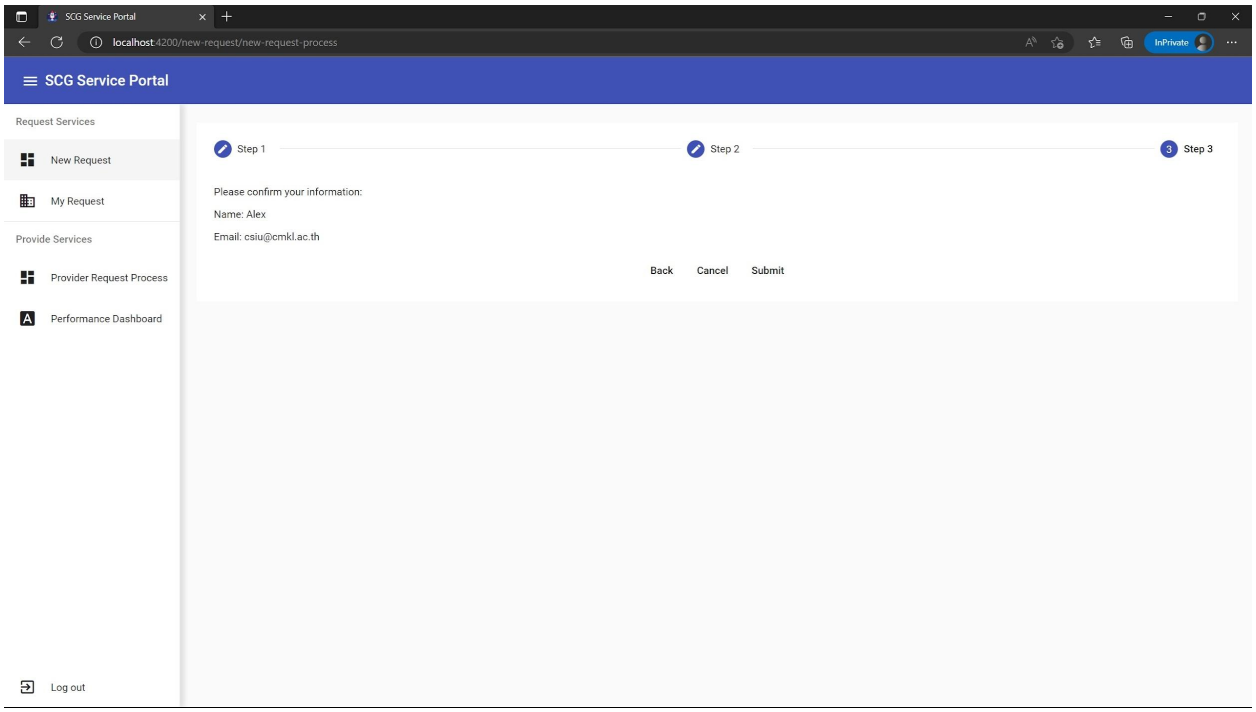
The portal will then provide an overview of the information and steps required to complete the service.



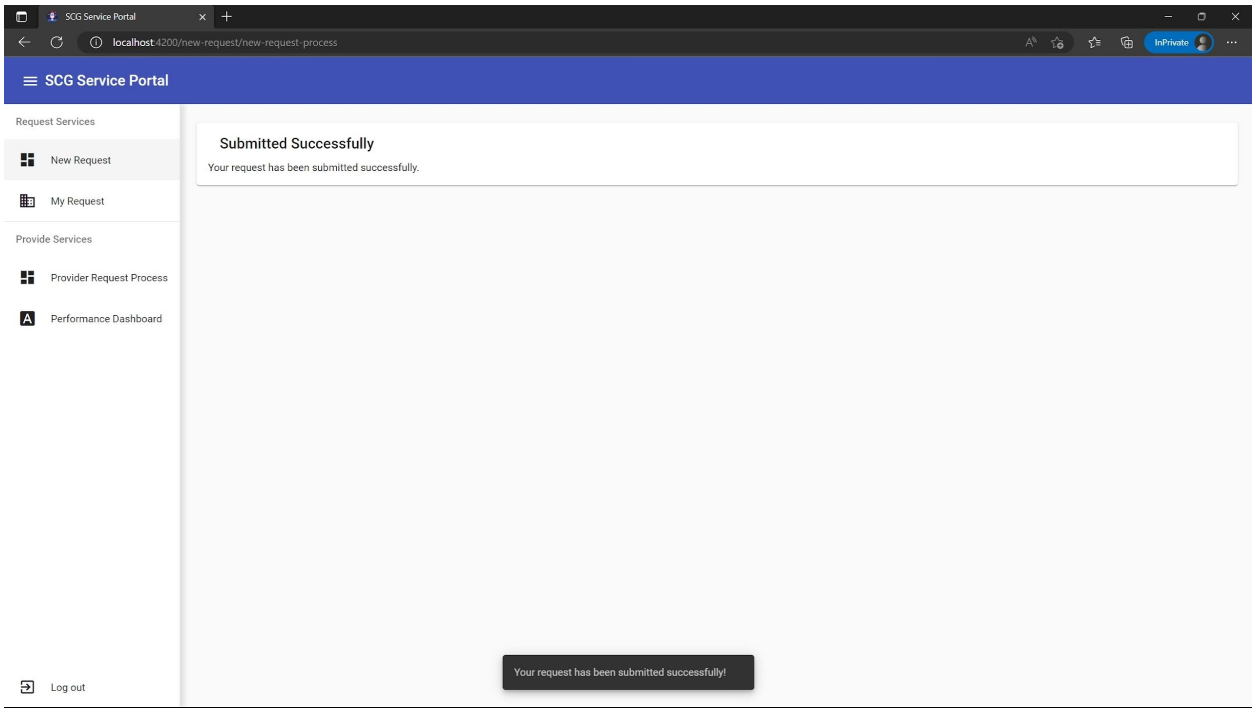
Once the requester now has to fill in all the necessary information to complete the service. This will be the only time the requester needs to provide any information unless the information they provided is incorrect. Also note that since the requester is logged into their account, their information is also saved on the portal, therefore they can autofill basic information such as their name and email address. This reduces the number of redundant manual tasks required of the requester.



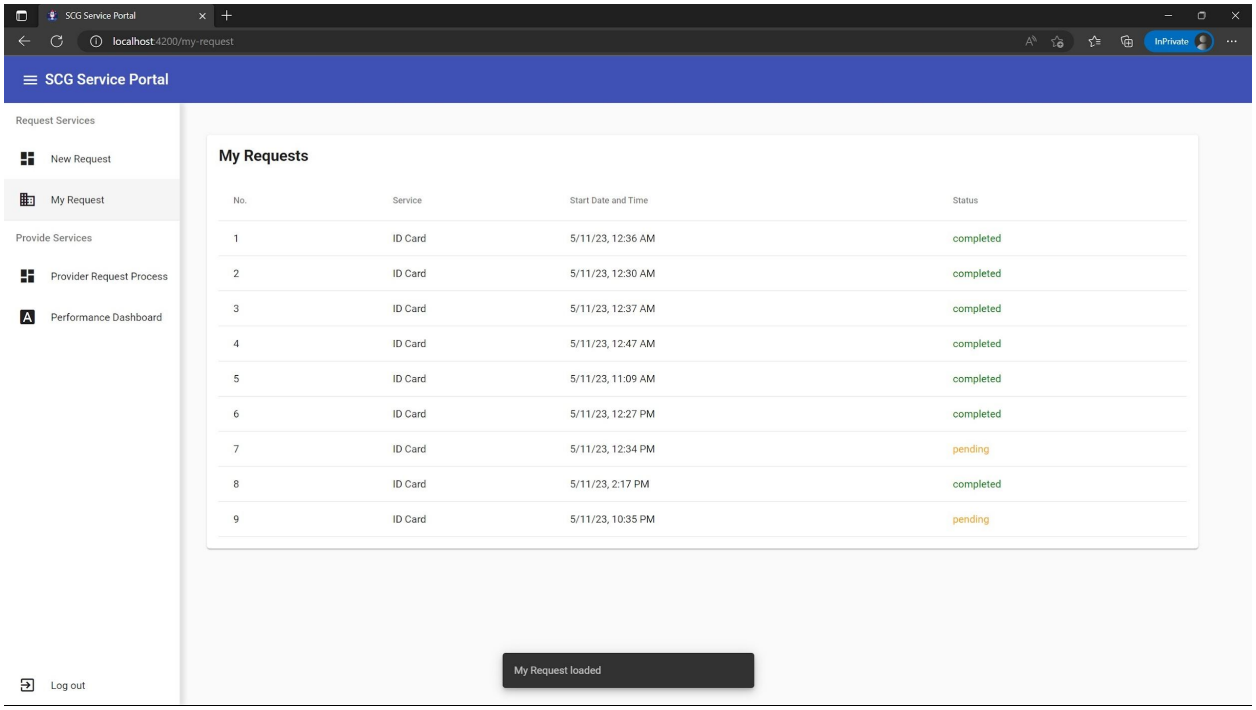
Once all necessary information is entered, then the requester must confirm that their information is correct.



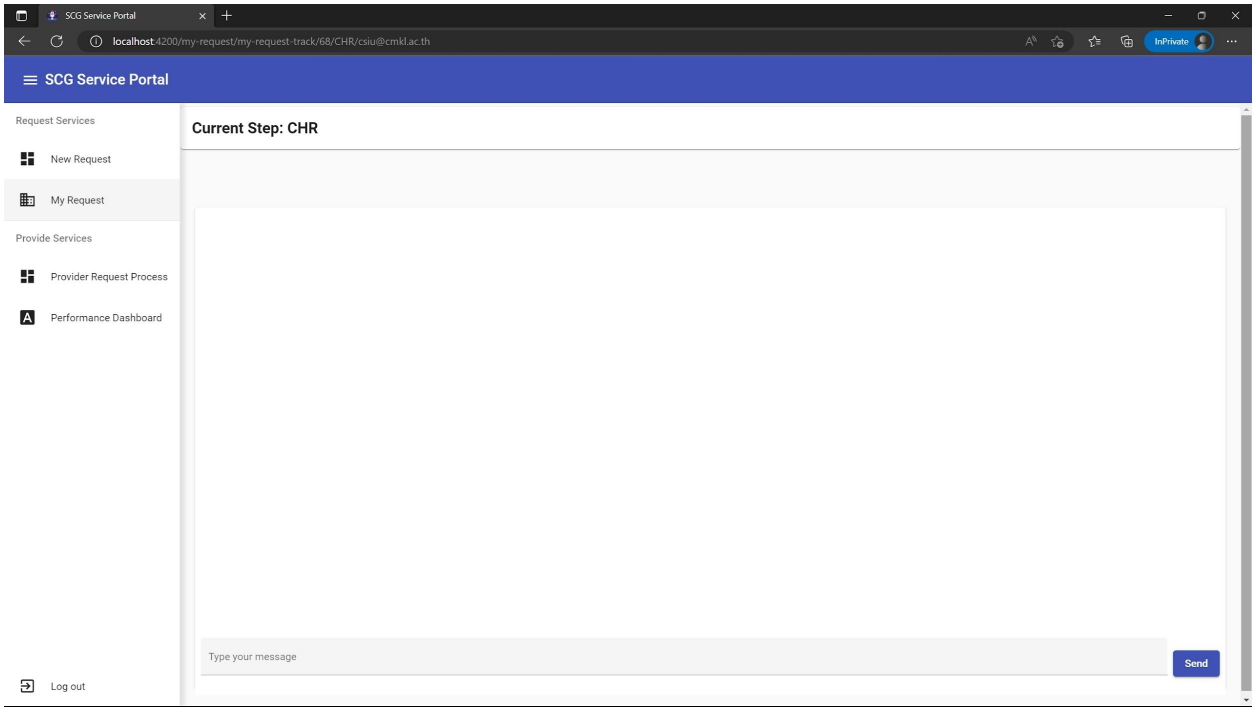
The portal will then provide a completion message once the request has been made.

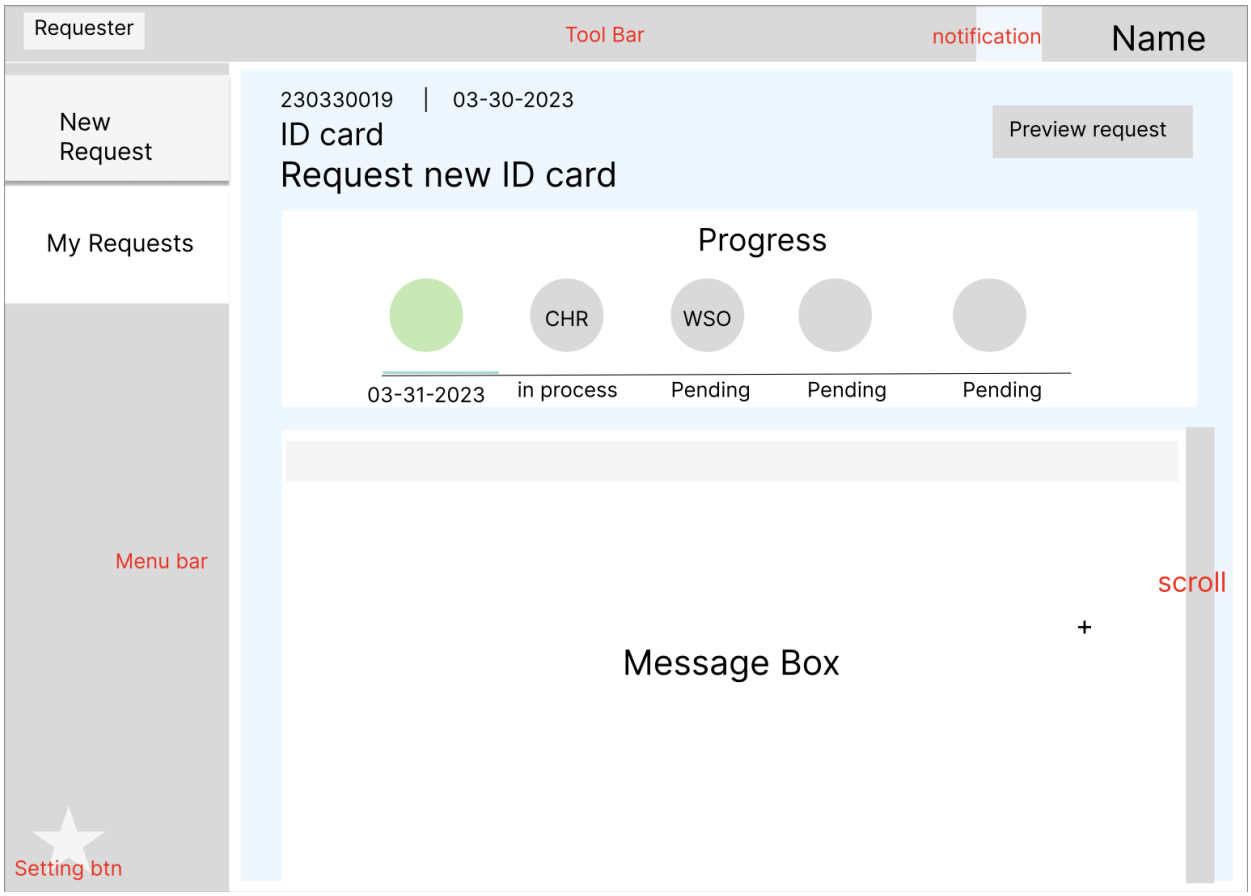


Now if the requester wants to follow up on their previous request, they can look at the my requests page. This page will contain a list of their previous requests along with some details such as the name of the service, date initiated and completed, and completion status.

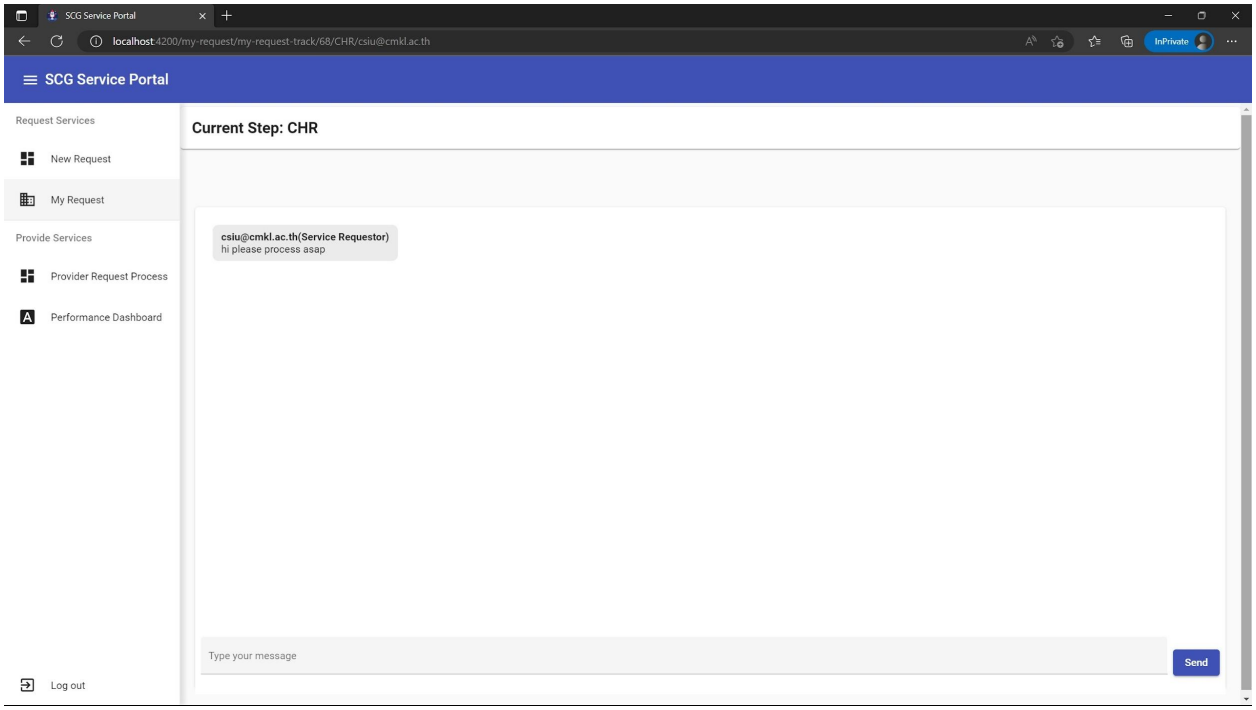


The requester can then select a request to view and they will be able to page. This is not yet complete in the prototype and the figma UI has been included for reference. They will be able to see a progress chart so that they know what tasks has been completed and what else is still pending.



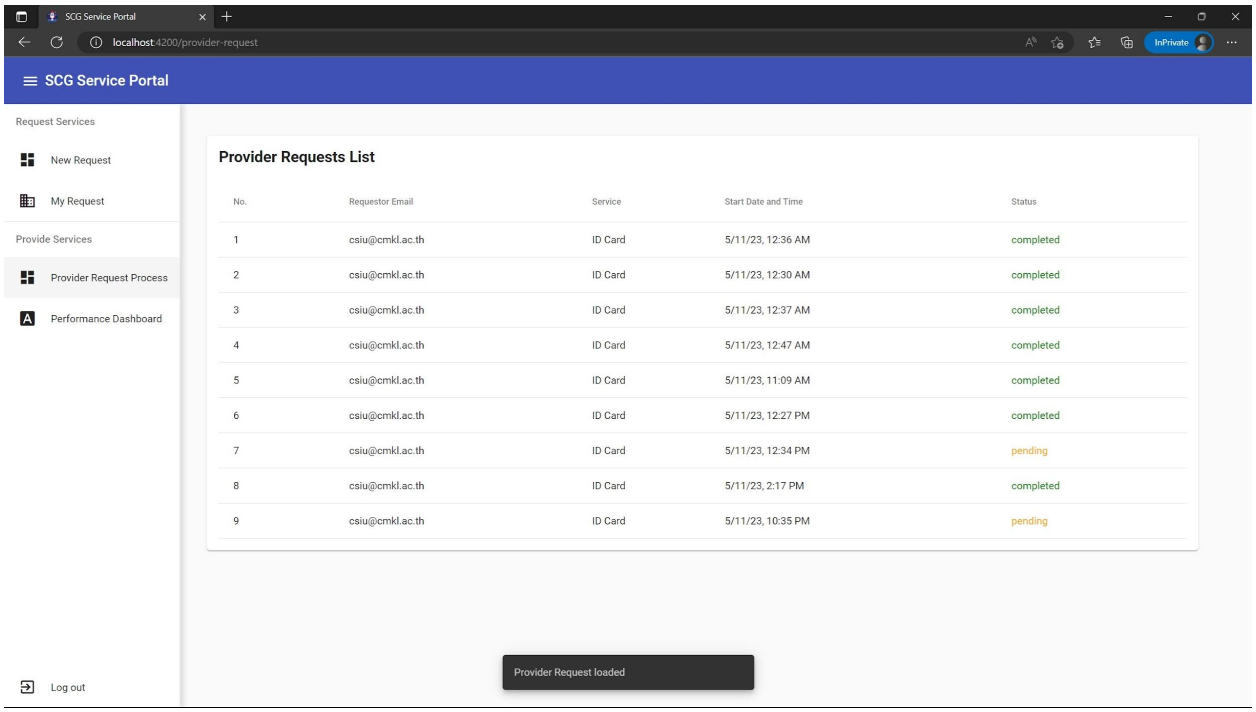


The requester can also utilize the chat function to talk directly with the provider. This helps eliminate use of unofficial means of communication such as LINE messenger which cannot be monitored by management and is not as secure. The messages will also be recorded to ensure quality service. For services with multiple providers, the requester will only be able to chat with the provider who is completing the task at the current step for privacy reasons.

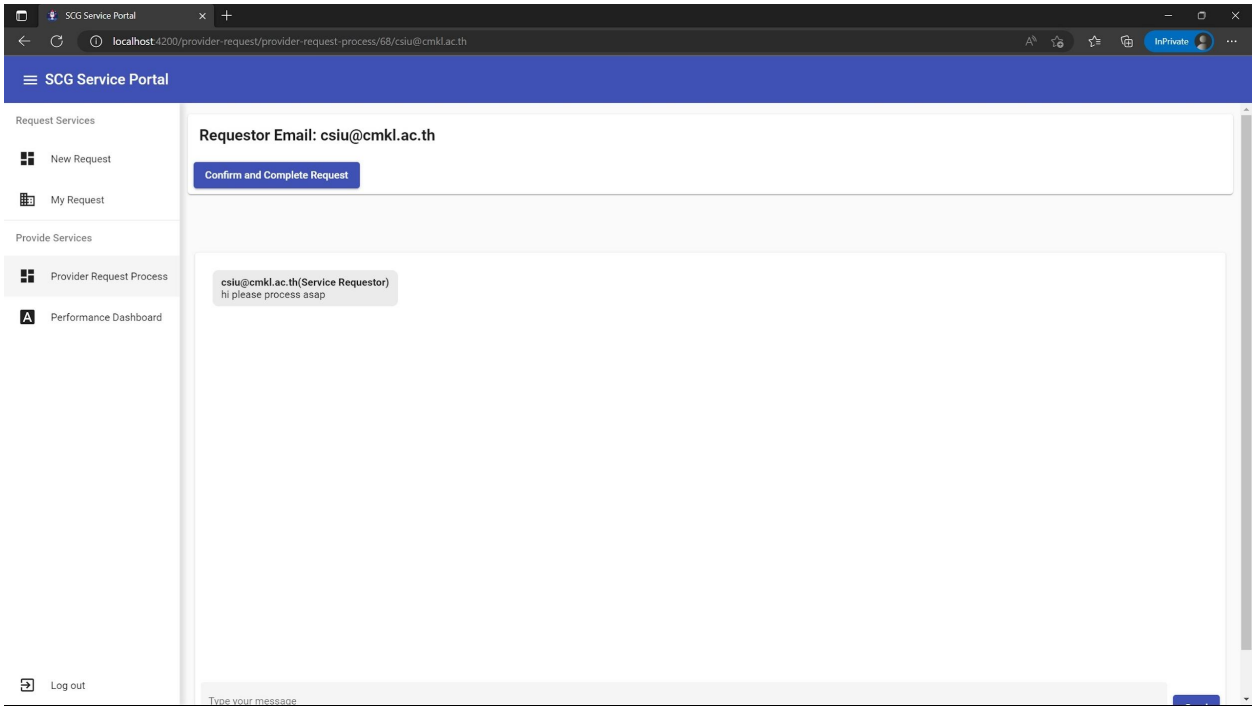


Provider

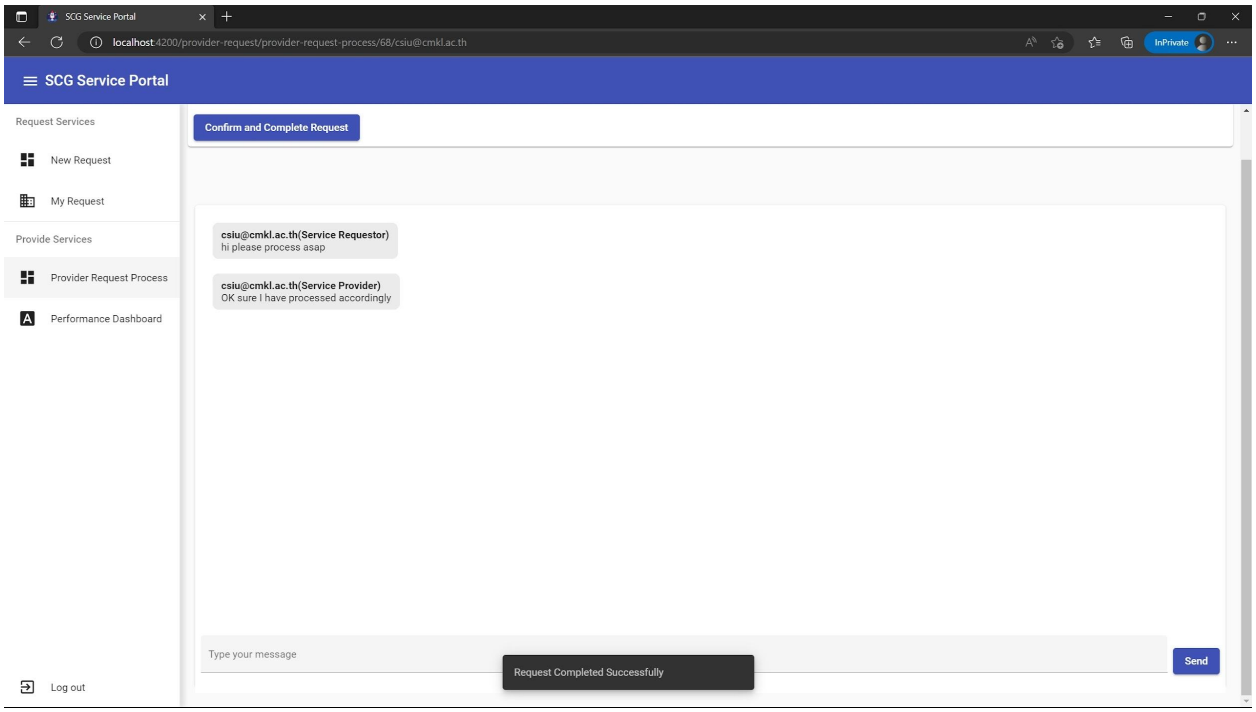
The provider will also be able to see their list of services they have pending and completed.



They can select the service that the earlier requester has just sent. The provider will be able to see the requester's message. In the implementation of the portal, the provider will also see the necessary information and instructions to complete their task.

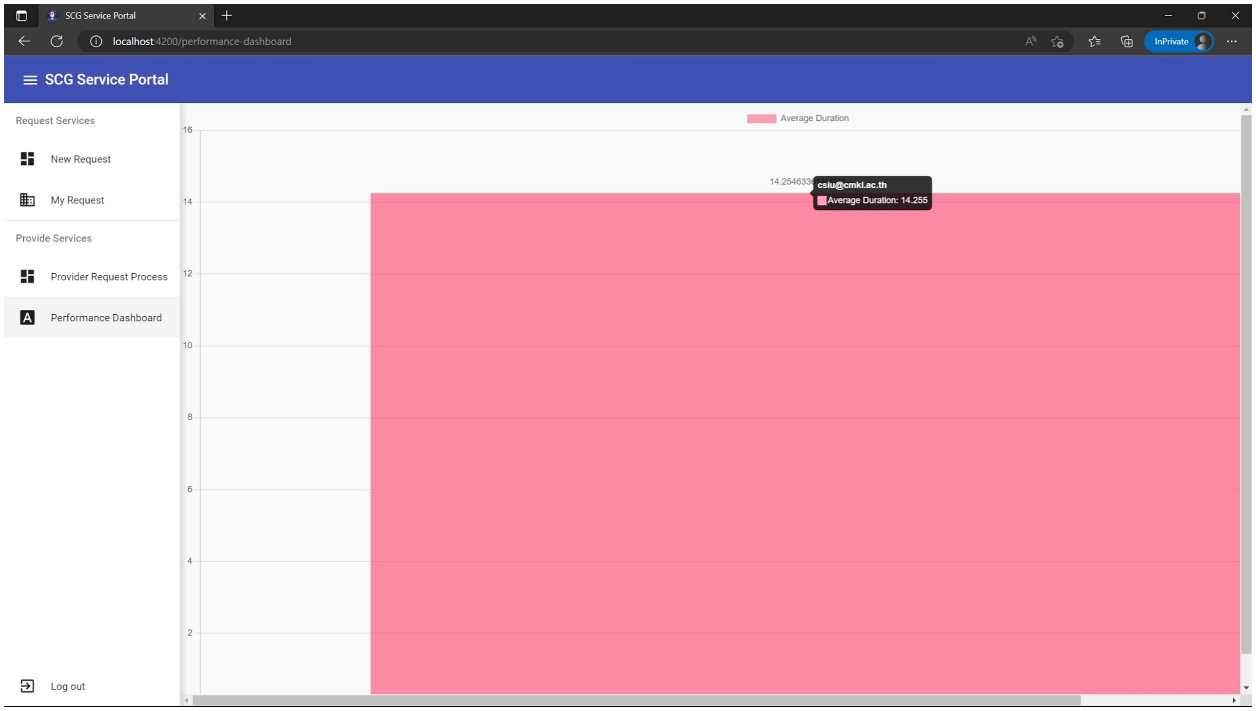


The provider will also be able to reply to the requester. Once the provider has completed their task, they will no longer be able to communicate with this requester through the portal until the requester has made another service request.

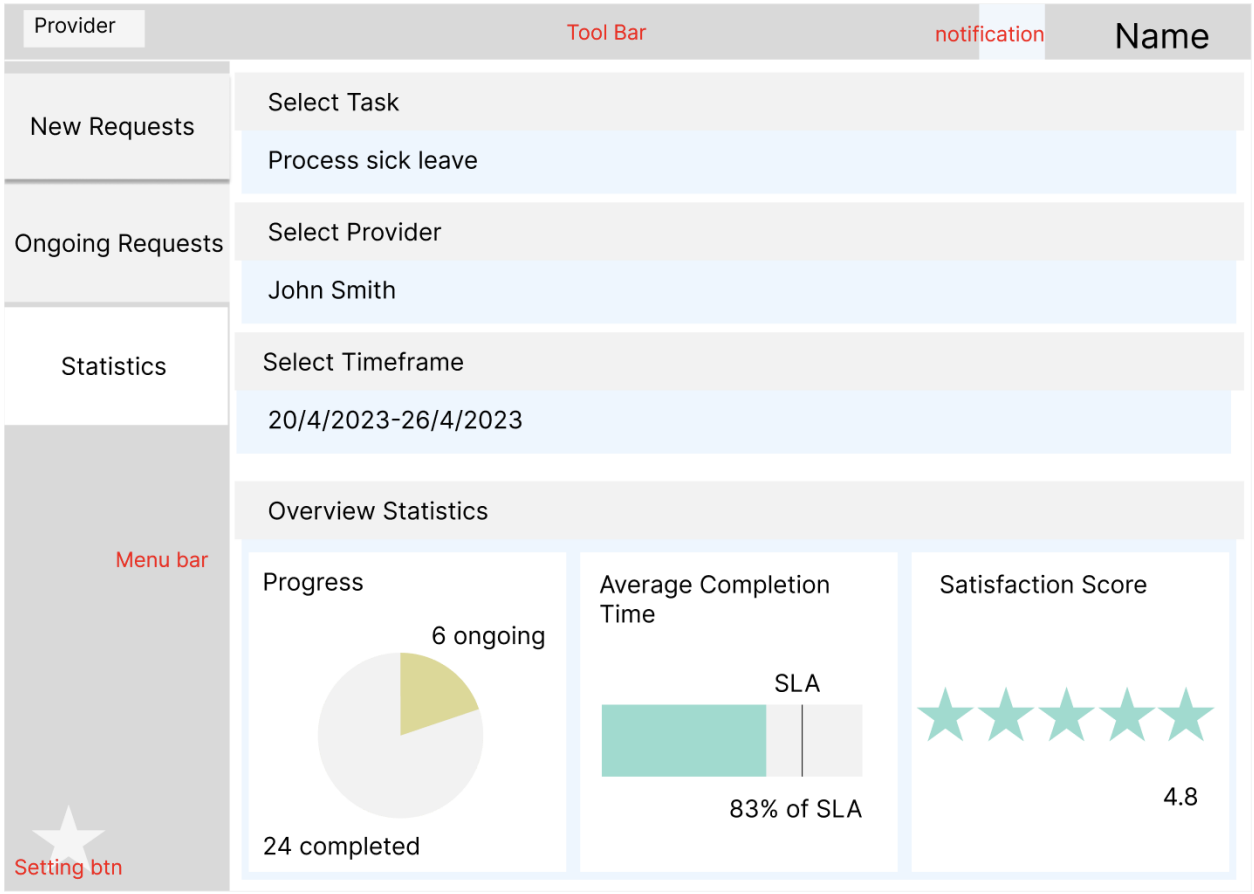


Statistics Dashboard

Providers, managers, and administrators will have access to the service delivery statistics dashboard. Our prototype currently uses ng2-charts to create a statistics dashboard. These tools are able to create various data visualizations such as pie charts, bar graphs, and histograms. Since we have no simulated backend data, our current dashboard is as seen below.



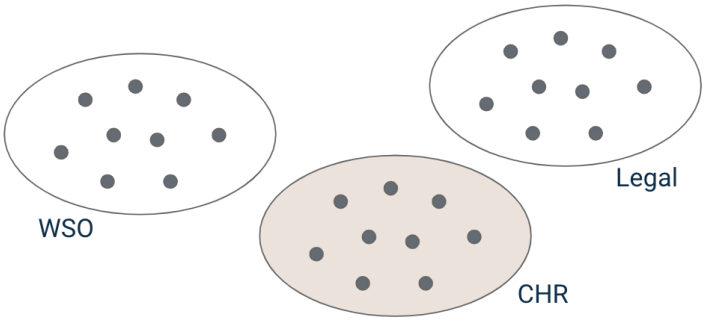
However, once we have backend data, we can make the dashboard look like the sample UI shown below for reference. This is an invaluable tool for providers, managers, and admins alike since they can use this to gain a better assessment of performance.



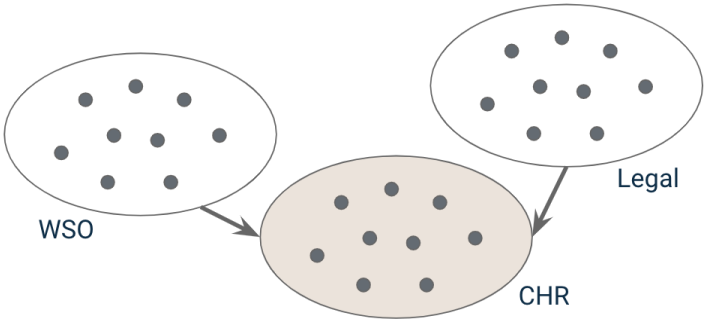
Transition Process Guideline

The transition process consists of the following 3 phases:

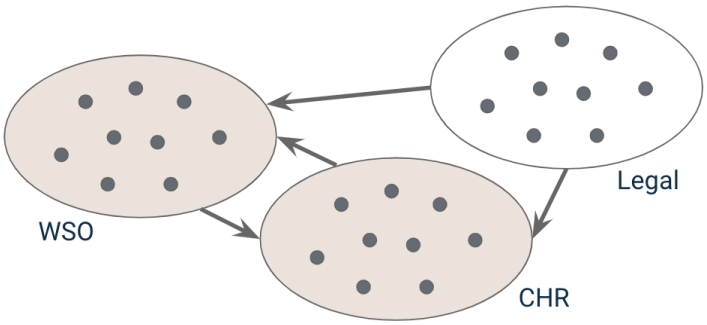
1. Internal HR Portal
   Goal:
   Construct our portal so that it has access to and the ability to edit the HR database. Used internally in the HR department. All HR employees can be requesters and providers.
   Tasks:
   - Use the current HR database as the basis of the eCA portal
   - Create services for use within the HR department
   - Create requester and provider roles for all HR employees



2. Portal for all HR Service Requests
   Goal:
   All HR services requested and completed through this portal. Employees in other departments are requesters.
   Tasks:
   - Create services that are completed by only the HR department
   - Create requester roles for all employees



3. Integration of Other Departments
   Goal:
   Services from other departments, as well as multi-departmental services, requested and completed through this portal. All relevant employees are now also providers.
   Tasks:
   - Select one department to integrate as provider at a time
   - Create services that are completed by only the integrating department or in conjunction with other departments already integrated
   - Create provider roles for relevant employees



Our approach to the transition process starts with building our portal around the HR department database since their database contains all the information of the employees so that accounts can be made. We also considered that a majority of corporate administrative tasks involve the HR department, therefore we believe that it is the best place to start.

Integrating the departments one at a time will make the transition easier as services do not have to all be created at once and requesters will have time to get used to using the portal. The second department we chose to integrate is WSO since a large portion of service requests involve that department. Furthermore, there are many services that the CHR and WSO departments work in conjunction on such as the ID card request. Therefore, this is a great department to integrate into the portal for maximum utility.

Issues and Obstacles

Technical

Despite the successful implementation of the prototype, there were several challenges encountered during the development and deployment stages. The primary issues included the steep learning curve associated with the AWS tech stack and the relatively high cost of AWS services.

Steep Learning Curve

One of the most significant challenges faced during the development process was the high learning curve associated with the AWS tech stack. AWS offers a comprehensive suite of services with complex functionalities and capabilities, which can be overwhelming for developers, especially those without prior experience with AWS.

The integration of various AWS services, such as AWS Cognito, AWS API Gateway, and AWS S3, required in-depth understanding and knowledge of how these services function, their dependencies, and their interplay with each other. This complexity led to substantial resistance from technicians, who had to spend considerable time and effort to familiarize themselves with these services.

Moreover, AWS's expansive and frequently updated documentation further added to the difficulty. While the documentation is thorough and provides a wealth of knowledge, navigating through it and finding relevant information can be a daunting task for newcomers.

To overcome these challenges, a significant investment in training and knowledge sharing was required. This included leveraging AWS's extensive online training resources, conducting internal workshops, and providing hands-on experience with the AWS services to the technicians.

High Cost of AWS Services

Another substantial obstacle was the relatively high cost of AWS services, especially when compared to competitors like Google Cloud, Tencent Cloud, and Alibaba Cloud. While AWS provides a robust and comprehensive suite of services, its pricing can be prohibitive.

The cost of services like AWS Cognito, AWS API Gateway, and AWS S3, which were integral to the web portal, significantly impacted the overall project budget. This was compounded by the charges incurred for data transfer, storage, and additional features like enhanced monitoring, which further escalated the overall cost.

To mitigate these cost challenges, a robust cost management strategy can be implemented. This involved continuous monitoring of AWS usage and costs, leveraging cost-optimization features and strategies provided by AWS, and shutting down unused resources or scaling down resources during off-peak periods.

Despite these challenges, the advantages offered by AWS in terms of scalability, reliability, and breadth of services made it a worthy choice for the development and deployment of the web portal. However, it's crucial to understand and plan for these challenges when embarking on a project using the AWS tech stack.

Project Management

Limited Timeframe

Our biggest challenge by far was our very limited time frame in contrast to how abstract the original problem presented to us was. With the abstract problem presented, we envisioned a grand blueprint of our project at the very beginning, such as a universal platform and database, a no code service expansion solution, and an automatic frontend code generation.

During the development, we abandoned some thoughts that are impossible to realize within our time frame and specified our goal. We changed the universal platform idea to separate portals for different users. We changed the all-in-one database idea to making use of existing databases and only recording the request history in our database. We found it infeasible to complete a no code solution in the given timeframe and thus used a low-code and structural design for replacement. We also gave up the idea of automatic UI generation and focused on other features. We tried our best to understand and get close to what the stakeholders are looking for, making it more reasonable to expect from our capability.

Internal Communication

At the beginning of our work, our main issue was communicating with each other. We each had different ideas and approaches to the problem while not communicating our ideas outside of our biweekly meetings. This led to a slower start to the project as we were still putting together our solution to SCG’s problem. We also did not effectively communicate how we were going to divide our tasks and ended up with some overlapping work, while some other tasks were incomplete.

After the proposal submission, we figured out our roles and responsibilities in the project as we had different strengths and weaknesses. We would tell each other what we will be working on each week in between meetings and use the meeting times to check on each other’s progress.

We had trouble figuring out a standard and comprehensive way to describe the activities of each

role, and to generalize the service workflow. We started by explaining in words, but it was neither clear nor concise. Then we tried to draw graphs, user cases, flowcharts, and statecharts to demonstrate it graphically. We modified the form of our graphs several times following Dr. Sally's advice and other tutorials, and finally reached a consensus within our team. Next time, from scratch we will focus more on how the graph would help our team members to integrate ideas and present to audiences, but not on how standard these graphs are.

Stakeholder Communication

As we were starting our project, the issue we understood as the most important that SCG wanted to address was consolidating the databases. However, we later realized that they were more interested in being able to keep track of their service delivery process to become more efficient after we had spent a long time trying to find a solution for the databases. Also, we could not get a reasonable sized population of different sample request scenarios that we could use for generalizing the request flow. This greatly hindered our progress in developing a generalized workflow as we were unsure if we were going in the right direction.

In communicating with SCG during our meetings, they seemed to like most of the ideas we came up with when we talked to them, and we had to see which ones they focused on more to determine what is more important to them. Starting from scratch we would be more direct in asking why they were interested in consolidating the databases as we found out that a consolidated database was for them to achieve their goal of being able to analyze service delivery performance between different departments.

Future Directions

Technical

Transition Process

We want to develop a more precise transition guideline document for SCG going forward. Our current transition process is not yet detailed as we require a better understanding of SCG's inner workings, especially within the HR department. The first phase of the transition process heavily relies on knowledge of the structure of the HR database, the information it contains, and services that the department performs.

Once we have a better understanding of the HR department, we will have a better idea of how phase 1 can be implemented. As we continue to the third phase, we will also need more information on the database and services of the department we are integrating so we can create the appropriate services that suit their needs. With our current plan, this can easily scale to integrate multiple departments as we receive more information about them.

No Code Solution

The current low code solution enables admin to create new services without the need for extensive coding. However, some coding knowledge is still required. In the future, we would ideally want to implement a full no code solution for this portal so that the creation of new services becomes even more accessible for SCG to continuously improve their service delivery.

Business

Our eCA portal is a highly marketable product as many companies are moving towards an online platform. This portal can increase productivity through more efficient workflow processes and improve the workplace environment by tracking and assessing employees' performance. Furthermore, it is adaptable to different industries due to generalized service workflow structure. The low code solution is great for startups looking to scale and increase services as well as mature organizations looking to improve current performance. It also enables a user-friendly transition process to an online platform as it can slowly integrate different departments.
The tracking of service delivery also helps companies overcome issues stemming from organizational silos.

Even with other competitor service portals, our main difference is the low code solution which encourages companies to continuously improve their service delivery. Since other portals will charge a fee for each additional service created, it will discourage companies from optimizing their current process. The eCA portal can be marketed towards companies that especially look to advance their service delivery.

Conclusion

We have achieved a prototype portal that is capable of scaling up to solve SCG's main issues with corporate administration services. The prototype is able to help requesters and providers improve their work efficiency by reducing redundant tasks and increase visibility by tracking their service requests and performance.

Although we were not able to consolidate the databases of the different departments, our portal still allows managers and administrators access to service delivery data through the use of the statistics dashboard. This data will help them analyze and assess current workflows and performance to better identify ways to improve service delivery. The most significant achievement we have made in this project was the low code solution as this greatly supports SCG's goal of continuous improvement.

A limitation of our prototype is the implementation of the low code solution as we did not have enough time to fully develop and implement a no code solution. Moreover, it is difficult to shift from low code to no code as that will require major adjustments to the portal. Another limitation of the prototype currently is our lack of specific knowledge of SCG's internal workings. This limits us in creating a more detailed transition process and building the prototype towards that transition.

References

[1] Viacheslav Klochan et al, "Digital Platforms as a Tool for the Transformation of Strategic Consulting in Public Administration," Nashrīyah-i Mudīrīyat-i Fannāvarī-i Iṭṭilā‘āt, vol. 13, (Special Issue: Role of ICT in Advancing Business and Management), pp. 42-61, 2021. Available: https://doaj.org/article/75988b978cf64b239890e127614e5b0b. DOI: 10.22059/jitm.2021.80736.

[2] D. Bauer et al, "Building and Operating a Large-Scale Enterprise Data Analytics Platform," Big Data Research, vol. 23, pp. 100181, 2021. Available: https://dx.doi.org/10.1016/j.bdr.2020.100181. DOI: 10.1016/j.bdr.2020.100181.

[3] L. Braubach, A. Pokahr and K. Jander, "JadexCloud - An Infrastructure for Enterprise Cloud Applications," Multiagent System Technologies, pp. 3, 2011. . DOI: 10.1007/978-3-642-24603-6_3.

[4] S. Bonomi et al, "Web Platform and Corporate Welfare: An Inclusive Organizational Solution," Lecture Notes in Information Systems and Organisation, pp. 75, 2020. . DOI: 10.1007/978-3-030-34269-2_6.

[5] I. B. Shubinsky and A. M. Zamyshlaev, "Unified corporate platform URRAN (UCP URRAN)," in Technical Asset Management for Railway TransportAnonymous 2022, Available: http://ebookcentral.proquest.com/lib/SITE_ID/reader.action?docID=6840177&ppg=179. DOI: 10.1007/978-3-030-90029-8_9.

[6] M. Considine, "THE CORPORATE MANAGEMENT FRAMEWORK AS ADMINISTRATIVE SCIENCE: A CRITIQUE," Australian Journal of Public Administration, vol. 47, (1), pp. 4-18, 1988. Available: https://api.istex.fr/ark:/67375/WNG-5LN4HZ42-2/fulltext.pdf. DOI: 10.1111/j.1467-8500.1988.tb01042.x.

[7] I. Pustylnick, "Enterprise collaboration platform based on social network architecture," SMC, Zug, 2015Available: http://www.econis.eu/PPNSET?PPN=899023150. DOI: 10.2139/ssrn.1789228.

[8] L. Ma, J. Chung and S. Thorson, "E-government in China: Bringing economic development through administrative reform," Government Information Quarterly, vol. 22, (1), pp. 20-37, 2005. Available: https://dx.doi.org/10.1016/j.giq.2004.10.001. DOI: 10.1016/j.giq.2004.10.001.

Appendix

A. Sample Scenarios

Scenario 2: Department database failure
Departments involved:

- Corporate Business Continuity Management Office (BCM)
- IT Department (CIT)
- Legal Office (LG)
- The affected department

Data:
BCM:

- Emergency Information: Type of emergency, location, time of emergency, affected systems and data
- Incident Response Plan: Activation of emergency response plan, assignment of roles and responsibilities
- Communication Plan: Internal communication to stakeholders, external communication to clients and partners
- Post-Incident Review: Documentation of the incident response, evaluation of the effectiveness of the response plan, identification of areas for improvement

CIT:

- Backup and Recovery Information: Identification of backup systems, assessment of backup data, recovery of the affected database

LG:

- Legal Compliance: Evaluation of legal obligations related to data protection and privacy, assessment of potential legal liabilities

Affected Department:

- Information about the affected database

Flow:

1. Emergency Declaration (input) - An emergency occurs, and the affected department declares the incident to the BCM department.
2. BCM - The BCM department activates the incident response plan, which includes coordinating with CIT, LG, and the affected department. BCM communicates with internal stakeholders and external clients and partners, as necessary.
3. CIT - CIT provides technical support to assess the backup data and recover the affected database. CIT works closely with the affected department to identify and assess the backup systems and data.
4. LG - LG provides legal support to ensure compliance with data protection and privacy regulations. LG evaluates the potential legal liabilities and provides advice to BCM and the affected department.

5. Affected Department - The affected department provides information and support to recover the database, including identifying the affected data and assisting CIT in the recovery process.
6. Recovery of the Database (output) - Once the database is recovered, BCM documents the incident response and conducts a post-incident review to evaluate the effectiveness of the response plan and identify areas for improvement.

Scenario 3: Employee Onboarding
Departments involved:

- Corporate Human Resources (CHR)
- Enterprise Brand Management Office (EBMO)

Data:
CHR:

- Personal Information: name, contact information, date of birth, social security number, etc.
- Status Information: job title, start date, salary, benefits information, etc.
- Background Check Information: results of background check, if applicable
- Training Information: training schedule, required courses, etc.

EBMO:

- Brand Information: introduction to SCG brand values, expectations for representing the brand, etc.

Flow:

1. CHR - Upon receiving the employee offer letter, CHR will gather and verify the personal and status information of the employee. They will also conduct background checks if required and schedule the training for the new employee.
2. EBMO - Once the employee's status is confirmed by CHR, EBMO will provide the onboarding package that includes the introduction to SCG brand values and expectations for representing the brand.
3. Onboarding Package Delivery (output) - After the completion of the onboarding package, the package is delivered to the new employee to ensure a smooth transition into their new role at the company.

Scenario 4: Office equipment maintenance
Departments involved:

- Workplace Solutions Office (WSO)
- Related Department (Department where the printer is located)

Data:
Related Department:

- Equipment Information: Printer model, location (e.g. third floor, room 304), serial number, etc.
- Maintenance Information: Hardware repair service, maintenance schedule, etc.
- Approval Information: Approval from authorized personnel, if necessary.

WSO:

- Cost Information: Cost estimate for the hardware repair service, budget information, etc.
- Completion Information: Date of completion, confirmation of completion, etc.

Flow:

1. Maintenance Request (input) - An employee notices that the office printer is not working and submits a maintenance request to the Related Department where the printer is located.
2. Related Department - The Related Department receives the maintenance request, checks the printer model, location, and serial number, and determines that the printer requires a hardware repair.
3. WSO - The Related Department sends a maintenance request to WSO for a printer hardware repair service. WSO receives the request, checks the printer maintenance schedule and confirms the cost estimate with the Related Department. If necessary, WSO seeks approval from authorized personnel for the repair cost.
4. Completion of Maintenance (output) - WSO schedules the printer hardware repair service and assigns a technician to fix the printer. After the completion of the repair, the technician confirms the date of completion and confirms that the printer is now in working condition. WSO sends the completion information to the Related Department, which then informs the employee who requested the repair

B. Admin Use Case Narratives

Case 1: Create new service
Main success scenario

1. Admin selects create new type of service
2. System asks for the title of new service
3. Admin puts in the title
4. System asks for what department is required to complete the task
5. Admin selects department
6. System asks for what module type is needed
7. Admin selects module type
8. System asks if task flow is complete
9. Admin selects complete
10. System asks to confirm
11. Admin confirms
12. New service is created

Ext (a) more than one module is required
8a. System asks if task flow is complete
9a. Admin selects add another module
10a. Return to step 4

Ext (b) admin cancels

10b. System asks to confirm
11b. Admin selects cancel instead of confirm
12b. New service is not created

Ext (c) invalid title
2c. System asks for the title of new service
3c. Admin puts in an invalid title (repeated title, offensive language, etc)
4c. System gives an error message
5c. Return to step 2

Case 2: View service delivery statistics
Main success scenario

1. Admin selects view service statistics
2. System presents list of services with average completion time
3. Admin selects service to view
4. System shows basic statistics such as service workflow, average time to complete service, expected time to complete service, how many times this service is used in the last 30 days, star rating for the service(?)

Ext (a) look at specific task
4a. System shows service workflow and average time to complete service
5a. Admin selects task in the workflow
6a. System shows task statistics such as average time to complete the task, expected time to complete task, what department is responsible for the task

Ext (b) search by department
2b. System presents list of services with average completion time
3b. Admin selects show by department
4b. System gives list of departments
5b. Admin selects department
6b. System presents list of services with average completion time for all services with tasks involving the selected department, other basic statistics for tasks by this department such as average time difference between expected vs actual task completion time, how many tasks per day
7b. Return to step 3

Ext (c) search by provider
2c. System presents list of services with average completion time
3c. Admin selects show by provider
4c. System provides search bar for admin to search for a provider by name/ID number
5c. Admin searches for provider
6c. System presents providers match name/ID
7c. Admin selects provider

8c. System presents list of services with average completion time for all services with tasks involving the provider, other basic statistics for tasks completed by this such as average time difference between expected vs actual task completion time, how many tasks per day
9c. Return to step 3

C. Sample UI Portals

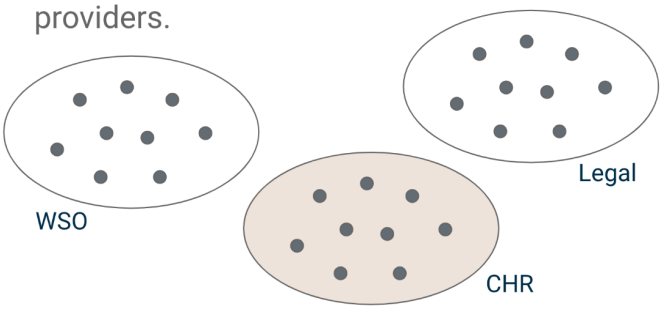
The link to the figma with the UI portals for requester, provider, and admin is given below:
https://www.figma.com/file/safp8hFo489nT7jU6dI4VC/UI-Portal?node-id=0-1&t=YgFVaROWXBZeBpiL-0

D. Transition Process Guideline

Phase 1: Internal HR Portal

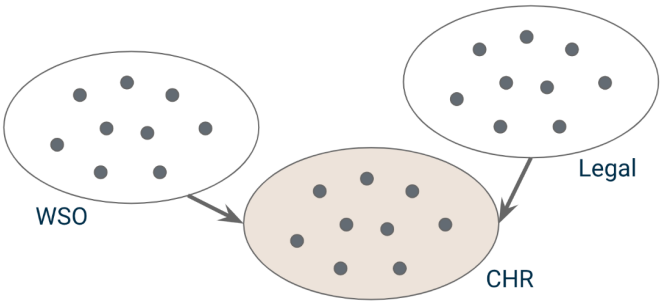
Portal with access to and the ability to edit the HR database. Used internally in the HR department. All HR employees can be requesters and providers.

- Use the current HR database as the basis of the eCA portal
- Create services for use within the HR department
- Create requester and provider roles for all HR employees



Phase 2: Portal for all HR Service Requests

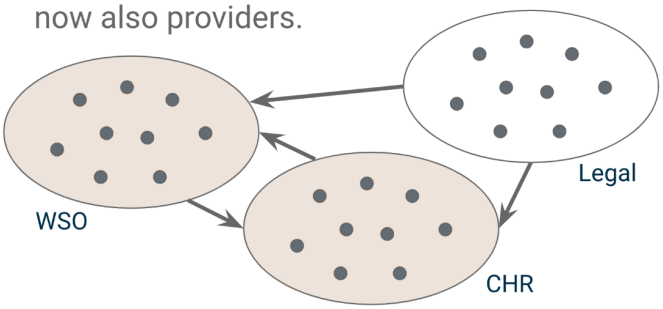
All HR services requested and completed through this portal. Employees in other departments are requesters.



- Create services that are completed by only the HR department
- Create requester roles for all employees

Phase 3: Integration of Other Departments

Services from other departments, as well as multi-departmental services, requested and completed through this portal. All relevant employees are now also providers.



- Select one department to integrate as provider at a time
- Create services that are completed by only the integrating department or in conjunction with other departments already integrated
- Create provider roles for relevant employees

E. Manager Use Case Narratives

Case: Manager approve of task
Main success scenario

1. Manager selects tasks pending
2. System presents tasks that are pending for the manager to approve
3. manager selects a task
4. System shows what inputs and outputs are expected
5. Manager selects approve task
6. System approves the task and sends to department to establish process

Ext (a) does not approve of task
4a. System shows what inputs and outputs are expected
5a. Manager selects not approve task instead of approve
6a. System returns task request to admin

Case: Manager approve of process
Main success scenario

1. Department admin selects processes pending
2. System presents processes that are pending for the department admin to approve
3. Department admin selects a task process
4. System shows departmental process for the task and SLA timeframe
5. Department admin selects approve process
6. System approves process for the task

Ext (a) does not approve of process
Ext (b) does not approve of timeframe

F. Prototype Code

The link to the Github folder with our prototype API and backend code is given below:
https://github.com/Zhou-owl/cmkl-rei/tree/backend