Women's Centres Connect

Technology Needs Assessment and Recommendation

August 2024

Contents

Introduction

This document introduces the reader to the reasoning behind Women's Centre Connect seeking for the design of a task -driven, easy to use, data collection system and the technology requirements for its implementation and evaluation. The system is designed to support their way of functioning by recording the unique services tailored to the issues in their respective catchment areas.

The Centres were using HIFIS (Homeless Individuals and Families Information System) as their data collection and reporting system which despite being high-end, was not accurate enough to portray the number of women, services and programs offered at each Centre. Thereby creating a gap in collecting and reporting. Representatives from the majority of centres were forced to employ a dual data collection system in order to more accurately track service and programming, thus making it difficult to meet the reporting requirements set by the Board, respond in a timely way to data requests and take action.

Problem Definition

The 9 Women's Centres within the provincial association of Women's Centres Connect lack a unified data collection method that facilitates efficient and accurate reporting, for the Centres' Board and Funders. This need was driven by the Centres currently employing various manual methodologies to collect data and fulfil their reporting requirements. The design of the proposed system includes developing a functioning prototype using suitable technology to facilitate at least one of the centres to meet the reporting requirements in an efficient manner.

In addition to this, there was a lack of availability and understanding of socio-demographic information of respective catchment areas to support the service, and program planning across the province. The proposed solution for this was to build socio-demographic dashboards that are powered by key characteristics that reflect the actual demographics of each area such as age groups, race, education, income, and health points. This analysis will justify the tailored service and program offerings made to address specific community needs effectively.

Scope

The scope of this project are detailed as below:

1. Analyse each Centre's workflows of recording and reporting collected data through the currently employed data collection system.
2. Conduct a series of interviews and surveys to collect data regarding the information flow in centres.
3. Generate requirements for the design of the new data collection system by analysing the results of interviews and survey responses.
4. Design overall architecture of the data collection system including database design, back-end API designs and the front-end user interfaces.
5. Develop a functional prototype that is to be released to a single centre to test its usability and functionality.
6. Implement role based access control for users in order to ensure personal data privacy between the centres and the coordinator.
7. Evaluate the prototype of the data collection system that allows for any improvements in the design.

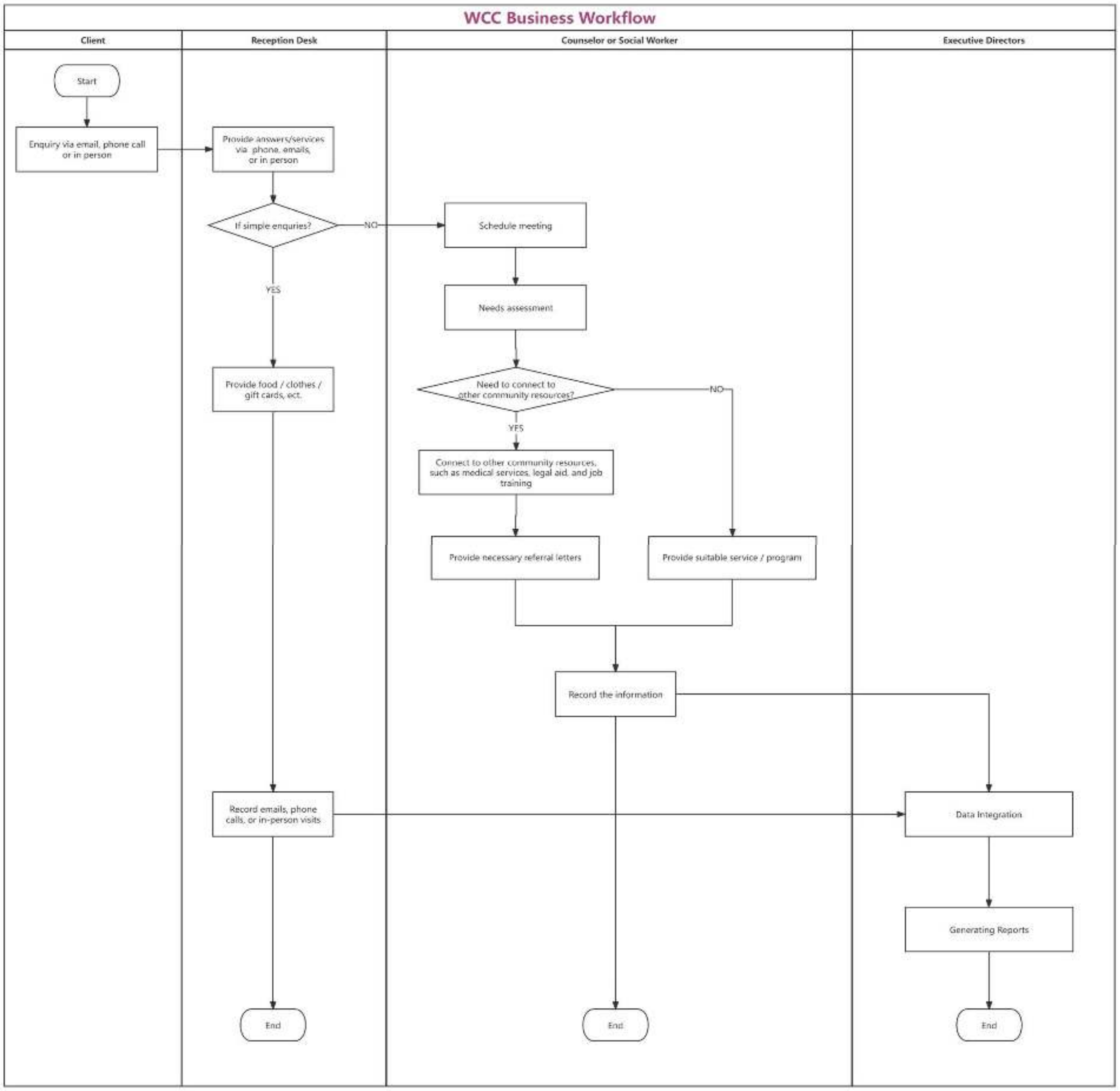
Restrictions

We anticipate certain restrictions, these will need to be agreed upon in terms of functionalities as well as the security and management of data. The following list describes them.

1. Limited Functionality Implementation: Certain features and functionalities may be restricted or limited in the initial version of the system to focus on core requirements and ensure a streamlined development process. This may involve prioritising essential functions while deferring more complex features to future iterations.
2. Restricted Access Control: Access control measures to the data layer and storage systems will be constrained. This means that while basic access controls will be in place, more advanced or granular control mechanisms may be limited or deferred.
3. Standardised Data Structure: The system's data structure will be designed to support common features agreed upon by all participating centres. This ensures compatibility and consistency but may limit customization options for individual centres.
4. Basic Authentication for Prototype: The prototype will incorporate a simple sign-in mechanism for user authentication. Advanced authentication features, such as multi-factor authentication, may be considered for future development.

5. Simplified Search Functionality: The search functionality within the system will be limited in scope. While basic search features will be included, more advanced search capabilities and filtering options may be reserved for future enhancements.

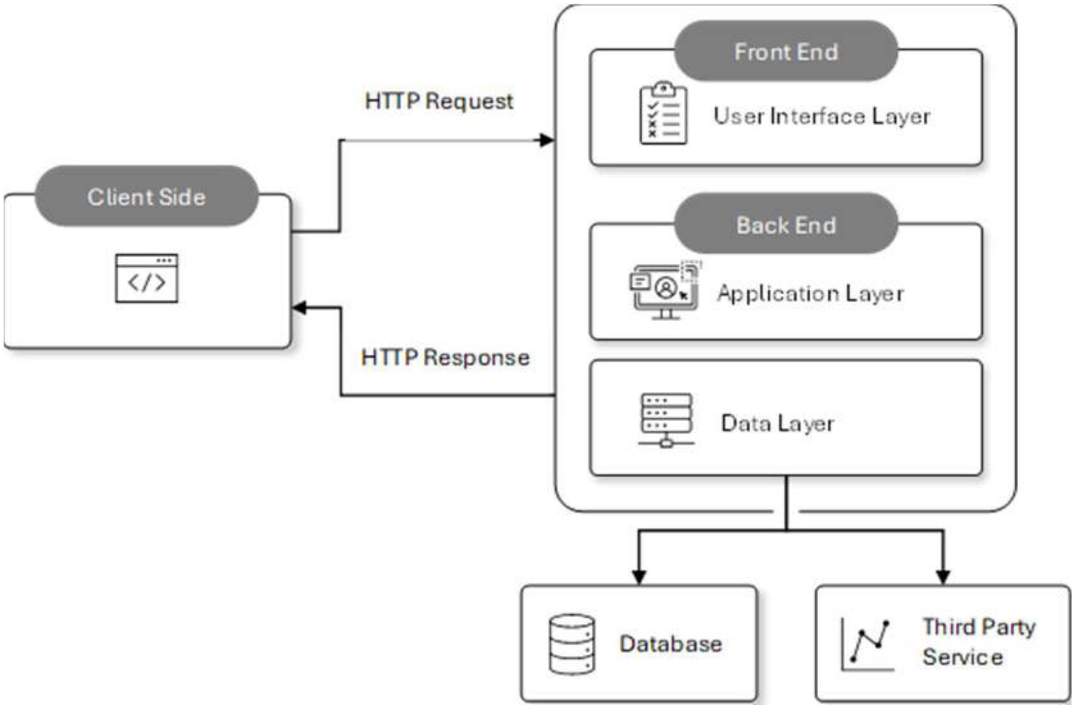
Business Workflow


WCC Business Workflow
Client
Reception Desk
Counselor or Social Worker
Executive Directors
Start
Enquiry via email, phone call or in person
Provide answers/services via phone, emails, or in person
If simple enquries?
NO
YES
Schedule meeting
Needs assessment
Need to connect to other community resources?
NO
YES
Provide food / clothes / gift cards, ect.
Connect to other community resources, such as medical services, legal aid, and job training
Provide necessary referral letters
Provide suitable service / program
Record the information
Record emails, phone calls, or in-person visits
Data Integration
Generating Reports
End
End
End

Proposed Solution

The proposed solution is a functioning prototype of a data collection tool that operates across three main layers, the user interface layer, the application layer and the data layer. These layers communicate with each other to ensure efficient data collection and accurate data retrieval for reporting.

1. Cloud Hosting option: This is the service provider that will allow the application to be hosted online, where anyone anywhere can access it.
2. User Interface Layer: This is the top most layer of the system responsible for displaying the information in the graphical user interface (GUI)
3. Application Layer: This acts as the logic layer responsible for communication between the GUI and the data layer.
4. Data Layer: This layer is responsible for data storage, data retrieval, data transformations, data management and other significant data processes.



Simple Architecture of proposed solution

Core Functionalities

The purpose of the data collection system to be designed for the Centres is to facilitate collection of data in its appropriate forms and generate meaningful reports that reflect the service offered by the centres over a period of time. They are as below:

1. **Client Registration:**

Functionality Description	Allows the users (staff at the Centres) to register an individual as a client by recording necessary personal information. This is proposed to be done in two ways - anonymous and non-anonymous. The Centres have to choose one of these two options that would suit their reporting requirements.
Interaction with Other Components	● **Client Profile Management:** New registrations are added to the Client Profile component. ● **Service Tracking:** Newly registered clients are included in the Service Update module for future service entries. ● **Reporting Module:** Client registration data is used for generating demographic and service usage reports.
Data Flow	● **Input:** User enters client details into the registration form. ● **Processing:** Data is validated and saved in the client database. ● **Output:** Profile is created, visible in Client Profile Management, and available for future interactions
Example Workflow	1. User selects "New Registration" from the dashboard. 2. Completes registration form with client details. 3. Submits the form; the system validates and creates a new profile. 4. New client profile appears in Client Profile Management.

2. **Client Profile:**

Functionality Description	Displays the details of each client along with the services they have used in the past, with the option to be able to add more in the future.
Interaction with Other Components	● **Service Update:** Profiles reflect all services provided. ● **Client Registration:** Newly registered clients are automatically added to this component. ● **Report Generation:** Client profiles contribute to reports on service usage and client demographics.
Data Flow	● **Input:** User requests to view or edit client profile. ● **Processing:** Profile data is retrieved and displayed or updated based on user input. ● **Output:** Updated profiles are reflected in the system and used in reports.
Example Workflow	1. User searches for a client profile. 2. Selects the profile to view or edit. 3. Updates or views client details and service history. 4. Changes are saved and reflected in both the profile and related reports.

3. **Service Update**:

Functionality Description	Enables users to update the services that clients have used each time they receive a service. This ensures that the client profiles are always up-to-date with the latest service usage.
Interaction with Other Components	● **Client Profile:** Updates are reflected in the client's profile. ● **Reporting Module:** Service data contributes to report generation.
Data Flow	● **Input:** User enters service details into the update form. ● **Processing:** Data is validated, processed, and integrated into the client profile. ● **Output:** Updated service information is visible in profiles and reports.
Example Workflow	1. Users access the "Service Update" feature. 2. Enter details about the new service provided. 3. Submits the update; system validates and saves the information. 4. Updated details are visible in the client's profile and in relevant reports.

4. **Report Generation:**

Functionality Description	Allows users to generate reports in a fixed format over a period of time chosen by the user, that contains statistical measures calculated using the data collected.
Interaction with Other Components	● **Client Profile:** Data from client profiles is used to generate reports. ● **Service Update:** Includes updated service data in the reports.
Data Flow	● **Input:** User selects report parameters and generates a report. ● **Processing:** System aggregates data from client profiles and service updates. ● **Output:** A formatted report is produced and made available for review.
Example Workflow	1. User selects "Generate Report" from the dashboard. 2. Chooses parameters (e.g., date range, service types). 3. System aggregates relevant data and creates the report. 4. User reviews and exports the report.

5. **Role Based Data Access:**

Functionality Description	Assign different roles to control access, allowing users to view, query data, or generate reports based on their assigned permissions.
Interaction with Other Components	● **Service Update**: Personnel from each centre can only update service data specific to their own centre in the reports. ● **Client Profile Management**: Personnel from each centre can only input new registrations for their own centre into the Client Profile component. ● **Service Tracking**: Each centre can view and query only their own client information. ● **Reporting Module**: Each centre can generate only their own reports. Additionally, coordinators have permission to generate reports that exclude any personal information.
Role-Specific Data Access and Permissions	**1. Personnel in Each Centre:** a. Can update service data only for their own centre. b. Can input new registrations into the Client Profile component for their own centre. c. Can view and query client information specific to their own centre. d. Can generate reports that are limited to their own centre's data. **2. Coordinator:** a. Can generate reports for any centre, but these reports exclude personal information.

Timeline

No.	Phase	Objective	Duration
1	Project Initiation & Requirement Analysis	Understand project scope and requirements	2 weeks
2	Design Phase	Design UI, system architecture, and database schema	2 weeks
3	Development: Client Registration & Profile	Implement Client Registration and Client Profile components	2 weeks
4	Development: Service Update	Implement Service Update functionalities	2 weeks
5	Development: Report Generation	Implement Report Generation module	1 week
6	Development: Role-Based Data Access	Implement role-based access control	2 weeks
7	Testing & Quality Assurance	Conduct comprehensive system testing for the prototype	2 weeks
8	Documentation & Prototype Presentation	Prepare documentation and present the prototype to stakeholders	1 week
9	Final Review & Prototype Refinement	Review feedback and make necessary adjustments	1 week

Technical Requirements

The technical requirements for the proposed prototype were identified by a series of interview sessions with the Executive Directors of the Centres and a requirement gathering survey circulated among the other staff. The overall responses were analysed to generate the following set of requirements, for each layer of the system.

User Interface Layer

The user interface layer for the proposed system is designed to provide an intuitive and user-friendly experience. It will display user and client information stored in the data layer, allowing users to access and modify the data with ease. This layer will present data in a visually appealing format to enhance user understanding. Users will be able to interact with the interface to create, read, update, and delete information in the data layer, ensuring that information is readily accessible and actionable. Additionally, the interface will facilitate data entry and submission through forms, ensuring a seamless user experience for both retrieving and contributing data.

The components are as follow:

1. Web Interface
    a. Developed using modern web technologies (e.g., HTML, CSS, JavaScript frameworks like React).
2. Dashboard
    a. A central hub providing access to different system functions (e.g., report generation, service tracking, client registration).
    b. Displays an overview of recent activities and notifications.
3. Forms
    a. Data entry forms for client registration, service updates, and other necessary inputs.
    b. Include validation to ensure accurate data entry.

Application Layer

The application layer serves as the core functional component of the system, managing the business logic and processing requests between the user interface and data layer. It handles API requests from the user interface, processes these requests, and performs necessary business operations. This layer is responsible for executing the logic required to interact with the data layer, including data retrieval, transformation, and manipulation. It ensures that data is properly handled and routed to the appropriate components for further processing or storage. The application layer also manages user authentication, authorization, and other critical operational functions, providing a robust framework to support the overall system's performance and security.

The components are as follow:

1. Business Logic Model
    a. Client Registration Module
        i. Handles the registration of clients, supporting both anonymous and non-anonymous modes.
    b. Service Update Module
        i. Manages the updates to client service records each time a service is provided.
    c. Report Generation Module
        i. Calculates and generates reports based on user-defined criteria, such as time periods and specific services. Includes logic to handle different reporting needs for various centres.
2. Security and Authentication
    a. Role-based access control (RBAC) system to ensure only authorised users can access certain features.

Data Layer

The data layer for the proposed system should be able to receive the collected dataset from the application layer via an API. The data is then stored in a staging layer for further transformations. A data pipeline is built that handles the data transformations to convert the raw data from the application layer to the statistical data points. These data points become the attributes on which the reports are to be generated. The generated reports will be available at the user interface layer for the users to download.

The data thus collected will also be cleaned to be of appropriate format to populate the dashboards for each Centres to showcase the various socio-demographic characteristics of their respective client base.

1. Data Storage
    a. Database Schema: The database will be designed to store client information and related consultation records, including but not limited to the following tables:

Table Name	Definition
Clients Table	Stores basic client information such as ID, name, contact details, etc.
Consultations Table	Stores records of each consultation, including date, duration, category, and related client ID.
Categories Table	Defines different consultation types such as Support Counselling and Crisis Intervention, Advocacy and Accompaniment, Information and Referrals.

    b. Data Integrity: Ensures consistency and integrity of data, using foreign key constraints to maintain relationships between clients and consultation records.
2. Data Aggregation and Analysis:
    a. Aggregation Logic: The system will implement aggregation logic to automatically calculate total consultations, total consultation duration for each client, and categorised summary data.
    b. Query Design: Optimised SQL queries will be used to generate the necessary statistical data for specific time periods or specific clients.
    c. Category-Wise Analysis: The system will perform category-wise analysis, summarising the number of consultations and total duration under different categories.

3. API Interface for Data and Reporting:
    a. API Design: The system will provide an API interface allowing the front-end or other systems to request statistical data or generate reports.
4. Report Generation:
    a. PDF Report Creation: The system will use tools to convert statistical data into PDF format reports.
    b. User Interface: Users will be able to generate and download reports through the system interface.

Tech Stack Selection

Cloud Hosting

To guarantee that the application is accessible to users from anywhere at any time, a reliable and scalable cloud service provider is crucial. Cloud hosting not only ensures the availability and performance of the application but also offers the flexibility to scale resources based on demand, thereby optimising costs and performance. The chosen cloud provider will enable us to host both the user interface (UI) and backend services on its infrastructure. This approach allows the UI to be accessible online, providing users with a seamless experience, while the backend application handles the business logic, processes requests, and manages data transactions efficiently. Additionally, the cloud provider will offer a variety of storage options, ensuring that the application's data is stored securely and can be retrieved and managed as needed.

Microsoft Azure

Microsoft Azure stands out as a comprehensive cloud service platform, particularly well-suited for hosting and managing web applications. Azure's core offering in this space is the Azure App Service, a powerful Platform-as-a-Service (PaaS) that enables developers to quickly build, deploy, and scale web apps and APIs. This service supports a wide array of programming languages and frameworks, such as .NET, Java, Node.js, and Python, making it an ideal choice for diverse development environments.

For the backend application, Azure's compute services will be employed to execute business logic, process API requests, and handle interactions between different layers of the application. This ensures that the application can handle a large number of concurrent users and transactions, maintaining high performance and reliability. In terms of data management, Azure Storage and Azure SQL Database provide secure, scalable, and high-performance storage solutions. These services enable the backend to store, retrieve, and manage data efficiently, with options for automatic backups, disaster recovery, and geo-replication to ensure data integrity and availability.

Additionally, Azure offers a comprehensive ecosystem of services, including Azure Blob Storage for managing unstructured data, Azure Data Lake Store for data manipulation, Azure Databricks for transforming data into actionable insights, and Azure Synapse Analytics for creating dashboards and generating reports.

By utilising Microsoft Azure, we can ensure that the application is not only robust and scalable but also secure and cost-effective, with the ability to evolve and adapt to future requirements. This makes Azure a highly suitable platform for deploying the web application and meeting the demands of users and the organisation

User Interface Layer

To meet the requirements for creating a graphical user interface (GUI) that users can interact with, a frontend web application is necessary. This application will allow users to submit client information, request data, and update client details as needed. To support these functionalities, a robust and scalable frontend application framework that provides the necessary organisational structure and GUI options is essential. The following technology stack is recommended to achieve these goals, with the overall cost discussed in subsequent sections.

React Framework

The React framework is an ideal choice for developing the graphical user interface (GUI) due to its flexibility and robust ecosystem. Its component-based architecture enables developers to create reusable UI elements, promoting consistency and efficiency across the application, which contributes to a highly responsive and dynamic user experience. Moreover, React's extensive library of third-party components offers powerful tools for managing tasks such as form submissions, data requests, and the seamless presentation of information. These qualities make React particularly well-suited for the user interface layer, allowing it to handle complex interactions while maintaining a smooth and intuitive experience for end-users.

Application Layer

To fulfil the requirements for creating a backend system that processes data and responds to API requests, a robust backend web application is essential. This system will enable users or other applications to request data from its database, store data in the database, and, if needed, perform statistical analysis for reporting purposes. To support these functionalities, a scalable backend application framework that offers the necessary REST or SOAP API capabilities is crucial. The following technology stack is recommended to achieve these objectives, with the overall cost discussed in subsequent sections.

Spring Framework

The Spring Boot framework is an outstanding choice for building the backend system, thanks to its versatile capabilities, comprehensive functionalities, and robust ecosystem. Its Model-View-Controller (MVC) architecture enables developers to create a clean, modular, and easily maintainable backend system that can efficiently manage and respond to API requests, utilising both REST and SOAP protocols.

Spring Boot's extensive library of components provides developers with powerful tools to handle a wide array of backend tasks, including HTTP request processing, data validation, and transaction management. Its built-in support for JPA (Java Persistence API) and seamless integration with various databases empower developers to efficiently perform CRUD operations, enabling smooth interaction with databases. The framework’s repository system abstracts data access, making it easier to connect with relational databases and manage complex data queries.

Additionally, Spring Boot’s scalability and performance optimization features make it particularly well-suited for handling high traffic loads and large volumes of data. The framework's ability to efficiently manage a large number of concurrent requests ensures that the backend can scale horizontally to meet growing demand without compromising on performance. Moreover, the integration of security modules like Spring Security ensures that sensitive data is well-protected, adhering to industry-standard security practices.

Its comprehensive toolset and flexibility make it the ideal choice for projects requiring a reliable backend that can handle everything from basic data processing to complex business logic, all while ensuring long-term maintainability and ease of scaling.

Data Layer

To suit the requirements for persisting data in our system and for generating reports and creating dashboards with the data collected, there is a need to maintain a database that allows seamless operations to carry out the same. It also requires transformation of collected data to infer the statistical data points, thus generating insights from the raw data. To support these functionalities, a robust and scalable cloud that can host the database and the data pipelines is required. For this purpose, the following technology stack is recommended. The overall cost is discussed in subsequent sections.

Microsoft Azure Data Products

A subscription to Microsoft Azure will allow the use of its various data services which are used for different stages in a data pipeline. Azure SQL Database can provide secure, scalable, and high-performance storage solutions. These services enable the backend to store, retrieve, and manage data efficiently, with options for automatic backups, disaster recovery, and geo-replication to ensure data integrity and availability.

Microsoft Azure also provides various products for data ingestion, data transformation, data storage and feeding data to other downstream processes like report generation and dashboarding.

Microsoft PowerBI

Microsoft PowerBI can be used to create dashboards from the data collected. The data pipeline built in Microsoft Azure can deliver the data appropriate for the dashboards, thus providing a descriptive analysis of the data collected through the data collection system prototype.

Development Cost

Option 1: Azure subscription (Pay as you go)

Service category	Service type	Region	Description	Estimated monthly cost (CAD)

Compute	App Service	Canada East	Basic Tier;
			1 B1 (1 Core(s), 1.75 GB RAM, 10 GB Storage) x 1 Month;
			Windows OS;
			0 SNI SSL Connections;
			0 IP SSL Connections;
			0 Custom Domains;
			0 Standard SLL Certificates;
			0 Wildcard SSL Certificates
				$83.94

Databases	Azure Database for MySQL	Canada East	Flexible Server Deployment, General Purpose Tier, 1 D2DS v4 (2 vCores) x 1 Month, 0 million Paid IO, 0 GB Additional Backup storage with LRS, without High availability	$192.89
Support (Optional)	Basic	-----	Included for all Azure customers	$0.00
	Developer	-----	Trial and non-production environments	$40.18
	Standard	-----	Production workload environments	$123.53

Option 1 Overall Cost

Option	Services	Total monthly cost (CAD)	Total yearly cost (CAD)
1	Compute + Database + Basic Support	$276.83	$ 3,321.94
2	Compute + Database + Develop Support	$317.01	$ 3,804.10

3	Compute + Database + Standard Support	$400.36	$ 4,804.30

Option 2: Azure subscription (1 year reserved)

Service category	Service type	Region	Description	Estimated monthly cost (CAD)

Compute	App Service	Canada East	Basic Tier;
			1 B1 (1 Core(s), 1.75 GB RAM, 10 GB Storage) x 1 Month;
			Windows OS;
			0 SNI SSL Connections;
			0 IP SSL Connections;
			0 Custom Domains;
			0 Standard SSL Certificates;
			0 Wildcard SSL Certificates
				$83.94

Databases	Azure Database for MySQL	Canada East	Flexible Server Deployment, General Purpose Tier, 1 D2DS v4 (2 vCores) x 1 Month, 0 million Paid IO, 0 GB Additional Backup storage with LRS, without High availability	$116.96
Support (Optional)	Basic	-----	Included for all Azure customers	$0.00
	Developer	-----	Trial and non-production environments	$40.18
	Standard	-----	Production workload environments	$138.53

Option2 Overall Cost

Services combination	Total monthly cost (CAD)	Total yearly cost (CAD)
Compute + Database + Basic Support	$200.90	$ 2,410.78
Compute + Database + Develop Support	$241.08	$ 2,892.94
Compute + Database + Standard Support	$339.43	$ 4,073.14

Option 3: Azure subscription (3 year reserved)

Service category	**Service type**	**Region**	**Description**	**Estimated monthly cost (CAD)**

Compute	App Service	Canada East	Basic Tier;
			1 B1 (1 Core(s), 1.75 GB RAM, 10 GB Storage) x 1 Month;
			Windows OS;
			0 SNI SSL Connections;
			0 IP SSL Connections;
			0 Custom Domains;
			0 Standard SLL Certificates;
			0 Wildcard SSL Certificates
				$83.94

Databases	Azure Database for MySQL	Canada East	Flexible Server Deployment, General Purpose Tier, 1 D2DS v4 (2 vCores) x 1 Month, 0 million Paid IO, 0 GB Additional Backup storage with LRS, without High availability	$78.57
Support (Optional)	Basic	-----	Included for all Azure customers	$0.00
	Developer	-----	Trial and non-production environments	$40.18
	Standard	-----	Production workload environments	$138.53

Option 3 Overall Cost

Services combination	**Total monthly cost (CAD)**	**Total yearly cost (CAD)**
Compute + Database + Basic Support	$162.51	$ 1,950.10
Compute + Database + Develop Support	$202.69	$ 2,432.26
Compute + Database + Standard Support	$301.04	$ 3,612.46

Dashboard Deployment

Service Category	Service Type	Estimated Cost
Microsoft PowerBI Pro License (Non-Profit)	PowerBI Pro	$4.11 CAD per user monthly for non-profit organisations
Domain	Google Site Domain	$28.56 CAD for 1 year, $183.67 CAD for 5 years, cheaper options might be available

Maintenance Cost

In order to maintain the data collection tool and the dashboards, a personnel with an expertise in software(front-end and back-end), data pipeline management, dashboard development and maintenance is required. Following are the three kinds of personnel who can handle these tasks and their hourly rate(CAD). It is possible to find one person with all of these skills but a higher salary will be expected.

Software Consultant	$24.52 to $34.13 per hour
Data Consultant	$23.08 to $33.26 per hour
IT Consultant	$21.15 to $29.81 per hour

Consultant fee (based on data from Glassdoor)

Recommendation

We recommend the following for the development and maintenance of the data collection system and the dashboards.

- **Recommendation 1:** *Option 2 Azure subscription (1 year reserved)*
  The 1 year reserved subscription to Azure combined with the Microsoft Power BI Pro license and a purchased domain suits the best as it provides the developers with ample time to develop the prototype in an agile manner and for pilot testing. It also makes it easier for the organisation to manage a 1 year subscription.

- **Recommendation 2:** *Option1: Azure subscription (Pay as you go)*
  The Pay-As-You-Go subscription to Azure, combined with the Microsoft Power BI Pro licence and a purchased domain, is also suitable as the organisation is billed only for the resources used on a monthly or yearly basis.

We recommend hiring a software consultant to maintain the front-end and back-end interfaces and a data consultant to manage data pipelines and dashboards.