



SOEN-6841 Software Project Management

Instructor: Dr. Joumana Dargham

Deliverable 2

By: Team 4

Fall 2024

**Group Members:**

Name	Student ID
Daivik	40279335
Smridhi Verma	40266036
Shashidhar Krovvidi	40110242
Harshit Saxena	40275098
Raghav Dalmia	40263888

## Feasibility Study

### Technical Feasibility:

As specified in Deliverable 1, a well-designed technology stack is mandatory for the system since it incorporates health data management, alerts for taking medication, a communicator and emergency response systems. Such technology encompasses the following:

- **Frontend:** the web frontend is built with React.js and the mobile with React Native for cross-platform seamless experience on both iOS and Android devices.
- **Backend:** Django (Python) for managing RESTful APIs, users and core business logic.
- **Database:** MongoDB (NoSQL) for storing reusable health record data.
- **APIs & Third-Party Integrations:** Syncing with Google Calendar for appointments, and WebRTC for safe video communication.
- **Security Considerations:** Data security is achieved with the use of SSL / TLS encryption for real-time communication. In addition, scalability and ability to integrate with the IoT and other future technologies are taken into account.

Our system's implementation is feasible within the available technologies and resources. Amongst the challenges are data privacy and security of data transmission.

- **Technical Risks:** There are risks including but not limited to the resistance of users to an upgrade of the system, the large data volume, and the need for seamless integration with several APIs. Backup systems are regarded as provided to prevent data loss.
- **Alternative Approaches:** The alternative solutions include, but are not limited to AWS as a cloud solution for system growth other than what is currently existing. Further improvements such as IoT incorporation are also considered to improve data gathering and supervision.
- **Risk vs. Reward Analysis:** MongoDB will provide an opportunity for better management of different health related records than the conventional SQL databases. React native will also support cross-platform development with hardly any code repeats.

### Operational Feasibility:

Our system directly impacts workflows in elderly care with:

- **Process Changes:** Integrating health monitoring, medication reminders, and emergency alerts in a single interface helps streamline workflows, reducing caregiver burden and minimizing risks due to missed notifications.
- **Operational Benefits:** The fact that all caregivers have a central control panel from which they can access health and medication records reduces the time for emergencies. Also, it is now easier for family members and other providers of healthcare services to offer regular updates and there is no longer a need for physical checkups again.

- **Training and adoption of users:** There is a specific interface for seniors which is very easy to use, several training sessions are held as well as continuous support factors which contribute to successful adoption across all groups of users.

Challenge in the application of the new system includes the following:

- **Resources:** Internal resources include the unwillingness of a more senior audience to manage an unfamiliar interface, the need for reorganization of the caregiver's workload, the need to remain within the confines of data relevant regulations which protects abuse of the data.
- **Improvements:** A centralized platform creates good linkage and communication within the platform enhancing better focus by caregivers on direct care activities.
- **Action Plan:** Training is scheduled for every user category before a stepwise introduction where feedback is expected at the end of all the sessions forming a continuous profile. Possesses initiation of technical assistance and engagement of users on the application of the system.

#### **Economic Feasibility:**

We prepare a comprehensive cost estimate which encompasses the figures of software development, deployment, third parties AP costs and maintenance. A cost estimate for the initial stage of the project would comprise the costs relating to the establishment of infrastructure and for the long-term, the cost estimates take into consideration the scaling requirements. Potential feature upgrade costs are also considered in the budget.

The resources are also arranged in such a manner that development testing and support will be done within the human and technical resources that are available. Anticipated ROI consists of a time savings to the caregiver, decrease in hospitalization for elderly users, and expansion of the market further one when more providers start using the app. Detailed cost-benefit analysis forecasts the payback period of the app and points on the economic benefit to families and healthcare providers in the long run.

## Project Plan Document

Project Name: Elder-Aid – Elderly Care Management System

Team Members: Daivik, Smridhi Verma, Shashidhar Krovvidi, Harshit Saxena, Raghav Dalmia

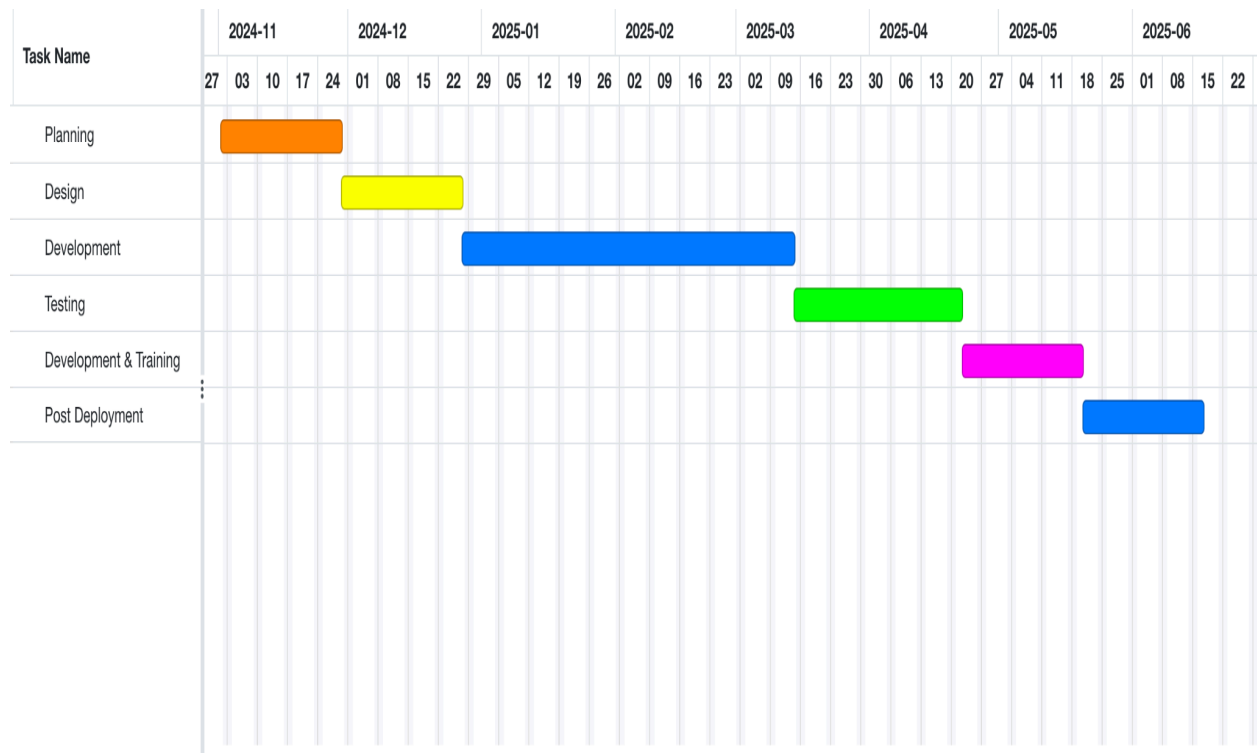
Goal: The goal is to develop a strong and easy-to-use platform designed specifically for elderly care. This platform will include features like health tracking, medication reminders, emergency response and smooth communication between caregivers and healthcare providers. The aim is to improve support for both elderly individuals and their caregivers.

### Project Phases and Timeline

The project is structured into a six-phase development cycle spread across 24 weeks and the time is allocated proportionally to the complexity and priority of each phase.

Phase	Duration	Weeks	Description
Planning	3 weeks	Week 1 – Week 3	Define all of the objectives, gather requirements, engage the stakeholders and lay a solid project foundation.
Design	3 weeks	Week 4 – Week 6	Complete architectural design and UI/UX prototyping for usability.
Development	8 weeks	Week 7 – Week 14	Develop key functionalities based on design specifications.
Testing	4 weeks	Week 15 – Week 18	Comprehensive testing to ensure performance, reliability and alignment with the requirements.
Deployment & Training	3 weeks	Week 19 – Week 21	Deploy system, train users and complete the system rollout.
Post Deployment	3 weeks	Week 22 – Week 24	Monitor performance, collect feedback and perform continuous improvements.

Gantt Chart covering all the six phases of development (Planning, Design, Development, Testing, Deployment & Training and Post Deployment).



Milestones and Deliverables

The Milestones are referenced as M<ID> and the deliverables are referenced as D<ID>

Phase 1: Planning (Weeks 1 - 3)

M1: Project Charter Completion

- Description: Finalize the project charter, which will establish a clear understanding of project goals, objectives and scope along with specific success metrics. This document will guide all team members and stakeholders on the purpose, vision and parameters of Elder-Aid.
- Resources: Project Manager
- Duration: 8 hours
- Dependencies: None

M2: Stakeholder Engagement

M2.1 Conduct Interviews and Surveys:

- Description: Engage with primary stakeholders including elderly individuals, caregivers, and healthcare providers, through interviews and surveys. The purpose is to gather insights on their needs/expectations to inform project design and ensure relevance to end-users.
- Resources: Team Members
- Duration: 12 hours
- Dependencies: Completion of Project Charter

#### M2.2 Compile Stakeholder Feedback Report:

- Description: Compile insights gathered from stakeholder interviews and surveys into a feedback report. This document will highlight critical observations and specific needs identified by the end-users.
- Resources: Project Manager
- Duration: 6 hours
- Dependencies: Completion of Stakeholder Engagement

### M3: Requirements Documentation

#### M3.1 Functional Requirements Gathering:

- Description: Define the core functional requirements of Elder-Aid based on stakeholder feedback which includes identifying the necessary features, technical capabilities and user interactions for each module.
- Resources: Technical Lead
- Duration: 10 hours
- Dependencies: Completion of Stakeholder Feedback Report

#### M3.2 Non-functional Requirements Gathering:

- Description: Gather and documenting non-functional requirements which will cover aspects such as quality standards, performance benchmarks, usability and compliance essential for ensuring system reliability and better user experience.
- Resources: QA Lead
- Duration: 6 hours
- Dependencies: Completion of Functional Requirements Gathering

#### Deliverables:

- Project Charter Document: D1
  - o A foundational document detailing the project's objectives, scope, success metrics and key stakeholders. This document serves as a roadmap which helps in aligning all team members and stakeholders on the project's purpose and vision.

- Stakeholder Feedback Report: D2
  - o A summary of findings from stakeholder interviews and surveys. This report provides a user-centered perspective that captures specific needs, desires and areas for potential improvement.
- Requirements Specification Document: D3
  - o A list of both functional and nonfunctional requirements. This document lays out all system expectations, performance criteria, and usability standards that the project must meet.

## **Phase 2: Design (Weeks 4 - 6)**

### **M4: System Architecture**

- Description: Create a detailed system architecture outlining module structures, data flow and integration points. This design will ensure technical feasibility, scalability and reliability by mapping out inter-module relationships and database interactions.
- Resources: System Architect
- Duration: 18 hours
- Dependencies: Project Charter completion

### **M5: UX/UI Design Completion**

#### **M5.1 Design Mobile Interfaces:**

- Description: Develop initial prototypes for mobile interfaces with a focus on accessibility, readability and ease of use. These interfaces are tailored to the needs of elderly users which emphasizes clear visuals and simple navigation.
- Resources: UX Designer
- Duration: 12 hours
- Dependencies: Requirements Specification completion

#### **M5.2 Design Web Interfaces:**

- Description: Create web interface prototypes for key features to ensure that the design aligns with mobile prototypes for consistency across platforms. User flows will focus on essential functionalities such as health monitoring, medication management and emergency response.

- Resources: UX Designer
- Duration: 8 hours
- Dependencies: Completion of Mobile Interfaces Design

#### M6: Design Review Meeting

- Description: Conduct a formal design review meeting with stakeholders for feedback, ensuring that architectural and UX/UI designs align with end-user expectations and project goals.
- Resources: Project Manager, Team Leads, Stakeholders
- Duration: 6 hours
- Dependencies: Completion of System Architecture and UX/UI Prototypes

#### Deliverables:

- System Architecture Document: D4
  - o A document with a visual representation of data flow, modular structure, database schema and the integration details for all core functionalities.
- UX/UI Prototypes: D5
  - o High-quality visual designs of the app and web interface showing important features like health tracking, medication reminders, emergency alerts, and messaging between caregivers and users. These designs will display how users interact with each part of the app.
- Design Approval Report: D6
  - o A signed off document that confirms stakeholder approval of the designs and provides any final feedback.

### **Phase 3: Development (Weeks 7 - 14)**

#### M7: Core Module Development

- Description: Implement development of primary modules - health monitoring, medication management, doctor appointment booking, emergency response and communication tools.
- Resources: Frontend and Backend Developers
- Duration: 80 hours

#### M8: Module Integration

- Description: Connect all modules ensuring modular communication, data consistency, and stability across the system.



- Resources: Full Stack Developers
- Duration: 40 hours

#### M9: Initial Functional Testing Completion

- Description: Conduct primary functional tests for each module to ensure that core functionalities align with user requirements and system design specifications.
- Resources: QA Team
- Duration: 16 hours

#### Deliverables:

- Core Modules Source Code: D7
  - o Well documented source code for each completed module that will cover all essential functions and meeting specified requirements.
- Integrated System Build: D8
  - o A functioning version of Elder-Aid that includes all core modules which are fully integrated and ready for testing.
- Initial Test Results Report: D9
  - o A document with test results for each module to confirm the functionality and identify any initial issues in subsequent testing phases.

### **Phase 4: Testing (Weeks 15 - 18)**

#### M10: Unit Testing

- Description: Conduct unit tests on each module to verify that individual components function correctly and independently to meet all the required requirements.
- Resources: QA Team
- Duration: 40 hours

#### M11: System Testing

- Description: Conduct system level tests to validate intermodular dependencies and overall functionality to ensure a stable and cohesive system which is ready for user testing.
- Resources: QA Team and Developers
- Duration: 20 hours

#### M12: User Acceptance Testing

- Description: Perform pilot testing with a select group of end-users to assess real world usability and user satisfaction. It will also help in identifying areas for improvement.
- Resources: QA Lead and Stakeholders
- Duration: 20 hours

#### Deliverables:

- Unit Test Results Report: D10
  - o Detailed report documenting the performance and functionality of each module which will include success rates and any identified issues.
- System Test Report: D11
  - o Comprehensive results from system testing to confirm stability, reliability and integration performance.
- User Feedback Report: D12
  - o Summary of feedback gathered from UAT sessions which will provide insights into user experience and also identifies potential areas for enhancement.

### **Phase 5: Deployment and Training (Weeks 19 - 21)**

#### M13: Cloud Deployment

- Description: Deploy the developed system to a secure cloud environment, ensuring reliable access for users and scalability for future updates.
- Resources: DevOps Engineer
- Duration: 24 hours

#### M14: User Training Materials

- Description: Create good training materials including user manuals, tutorial videos, and FAQs to support both elderly users and caregivers.
- Resources: Technical Writer
- Duration: 12 hours

#### M15: Gather Training Feedback

- Description: Collect feedback on training sessions to assess effectiveness and areas for improvement.
- Resources: Project Manager

- Duration: 4 hours
- Dependencies: Completion of User Training Material

#### Deliverables:

- Cloud-Hosted Application: D13
  - o The application will be deployed on a secure and scalable cloud infrastructure, making it accessible to users online. It will include configuring the server, deploying the application code, setting up a database and ensuring secure access.
- User Training Guides and Videos: D14
  - o The user training materials consist of written guides and instructional videos that provide clear instructions for using the Elder-Aid system. These materials aim to assist end-users and administrators in understanding the application's features and functionalities.
- Training Feedback Report: D15
  - o The Feedback Report will compile the responses from end-users and administrators after the training sessions. It will record their feedback on the clarity, completeness and effectiveness of the training materials and sessions.

### **Phase 6: Post-Deployment Monitoring and Maintenance (Weeks 22-24)**

#### M16: Performance Monitoring

- Description: Implement monitoring tools to track application performance, usage and any potential errors.
- Resources: QA Team
- Duration: 6 hours
- Dependencies: Successful Deployment

#### M17: User Feedback Collection

- Description: Gather qualitative feedback from users on their experience and system usability.
- Resources: Project Manager
- Duration: 4 hours
- Dependencies: Deployment

#### M18: Issue Resolution

- Description: Address minor bugs and issues reported during initial post-deployment by the end users and the testing team.

- Resources: Development
- Duration: 8 hours
- Dependencies: Completion of Stability Report

Deliverables:

- Stability and Performance Report: D16
  - o The Stability and Performance Report will provide a thorough assessment of the application's performance and reliability after its deployment. This deliverable evaluates key system metrics such as uptime, bugs and performance under various load conditions to ensure that the application can handle real-world applications effectively.
- User Feedback Summary: D17
  - o The User Feedback Summary will help in understanding how users are interacting with the Elder-Aid application and how well it meets their needs. It will consist of both quantitative and qualitative data gathered from user surveys, interviews and usage analytics.
- Maintenance and Improvement Plan: D18
  - o Outline the strategies and steps required to keep the Elder-Aid application running smoothly over time. This includes a structured schedule for regular maintenance activities such as software updates, bug fixing and performance optimizations.

## **Risk Assessment and Mitigation**

### **Risk Identification**

#### **1. Technical Risk**

- **System Integration and Interoperability:** Integrating several capabilities such as health monitoring, medication reminders, doctor booking, and communication tools may cause compatibility concerns between modules and platforms (e.g., web, mobile). This raises the likelihood of service interruptions and data discrepancies.
- **Data Privacy and Security:** Given the sensitivity of health data, securing data protection from cyber attacks is critical. Failure to comply with security rules (such as PIPEDA in Canada) may result in legal consequences and a loss of user confidence.
- **Scalability Challenge:** Server capacity, data storage, may be tested as the system expands to accommodate additional users and services. Future-proofing the infrastructure is crucial for preventing performance issues.
- **Reliability on third party APIs:** Dependence on external services (e.g., Google Calendar for appointment booking, Twilio for alerts) increases the risk of disruption, reducing system reliability.

#### **2. Operational Risk**

- **User Adoption and Usability:** Elderly consumers may have difficulty adjusting to digital platforms. Complex interfaces or a lack of user-friendly design may result in low engagement, reducing the system's efficacy.
- **Dependency on Caregivers and Family:** The reliance on caregivers and family members for data input and health monitoring might result in data discrepancies or missing records owing to variable degrees of engagement.
- **Resource Allocation and Maintenance:** Continuous technical support and maintenance are required to maintain smooth functioning. Insufficient resources or delays in fixing issues and user feedback may result in lower system effectiveness.

#### **3. Financial Risk**

- **High Development and Maintenance Cost:** Development and maintaining a complete system that includes cross-platform compatibility, data storage, and security features may be expensive. Underestimating these expenses may jeopardize the project's financial sustainability.
- **Revenue Model and Profitability:** The system targets a sensitive group with low income, therefore monetising the platform without sacrificing user accessibility or happiness may be difficult.
- **Market Competition:** Established rivals, such as Medisafe, Birdie, and Dynseo, already have a market presence. Capturing market share may require large

marketing efforts, as well as the provision of competitive pricing or new features, which may have an impact on profitability.

#### 4. Market and Adoption Risk

- **Changing Needs and Market Dynamics:** The elderly care business is rapidly growing, with an increased emphasis on IoT integration and wearable devices. The project's decision to exclude IoT integration may limit its appeal if competitors adopt similar technologies.
- **Stakeholder Satisfaction and Retention:** It is vital to maintain stakeholder satisfaction (including carers, healthcare professionals, and family members). Poor user experience may cause user churn, reducing the system's efficacy and reputation.

#### 5. Environmental and Social Risk

- **Data Security and Cybersecurity Threats:** The platform handles sensitive health information, which makes it a potential target for attackers. A data breach will affect user confidence and create significant reputational damage.
- **Pandemic Related Disruptions:** External events, such as COVID-19, have the potential to alter development timetables, personnel availability, and demand for senior care systems. Such circumstances may modify system priorities or cause delays in product delivery.

#### 6. Legal and Compliance Risk

- **Health Data Regulations:** Compliance with Canadian health-data regulations is critical, being noncompliant may lead to penalties and legal consequences.
- **Contractual Obligations with Service Providers:** Contracts with third-party services (such as Twilio and Firebase) may have binding conditions. Failure to satisfy service level agreements (SLAs) may result in fines or service interruptions.
- **Liability in case of Health Incidents:** The platform includes health monitoring and emergency response, accountability for any bad events caused by system flaws or data inaccuracies might lead to legal action.

### **Risk Impact Analysis**

#### 1. Technical Risk

- **System Integration and Interoperability:**  
Cost: Development cost can increase by 15-25% due to technical issues.  
Time: Increment in timeline by 1-2 months.
- **Data Privacy and Security:**  
Cost: Non-compliance with security standards may result in fines and higher insurance rates.  
Time: Development of security features may result in additional 3-4 weeks.

- Scalability Challenge:  
Cost: Project cost may go up by 20% due to upsizing resources.  
Time: Feature rollouts can be delayed by 2-3 weeks.
- Reliability on third party APIs:  
Cost: Inhouse development of APIs is an expensive solution.  
Time: Additional 1-2 months in development phase.

Risk	Probability (P)	Impact (I)	Risk Score (P*I)
System Integration and Interoperability	4	4	16
Data Privacy and Security	3	5	15
Scalability Challenges	4	3	12
Reliability of third party APIs	3	4	12

## 2. Operational Risk

- User Adoption and Usability:  
Cost: UI and UX development could increase the cost by 10-15%.  
Time: Additional 2-4 weeks in research and designing phase.
- Dependency on Caregivers and Family:  
Quality: Increased chances of data inaccuracies.
- Resource Allocation and Maintenance:  
Cost: Maintenance could increase the operational cost by 10%.  
Quality: Chances of reliability issues due to insufficient maintenance. Thus, affecting user satisfaction.

Risk	Probability (P)	Impact (I)	Risk Score (P*I)
User Adoption and Usability	4	4	16
Dependency on Caregivers and Family Members	4	3	12
Resource Allocation and	3	4	12

Maintenance			
-------------	--	--	--

### 3. Financial Risk

- High Development and Maintenance Cost:  
Cost: The initial development cost can increase by 50% of the proposed cost.  
Time: Addition of 3 additional months.
- Revenue Model and Profitability:  
Cost: Increment in customer acquisition cost could be experienced.
- Market Competition:  
Cost: Additional 25% could be spent on marketing.  
Quality: Rollout pressure could affect the quality of the feature.

Risk	Probability (P)	Impact (I)	Risk Score (P*I)
High Development and Maintenance Costs	4	4	16
Revenue Model and Profitability	3	4	12
Market Competition	4	3	12

### 4. Market and Adoption Risk

- Changing Needs and Market Dynamics:  
Cost: Development cost for integrating new technology may increase the cost by 30%.  
Time: An additional 2 months can be seen in the development phase.
- Stakeholder Satisfaction and Retention:  
Cost: Increment of 10% can be seen in operational costs.  
Quality: Negative reviews can be seen due to reduced user satisfaction.

Risk	Probability (P)	Impact (I)	Risk Score (P*I)
Changing Needs and Market Dynamics	4	4	16



Stakeholder Satisfaction and Retention	4	3	12
--	---	---	----

## 5. Environmental and Social Risk

- Data Security and Cybersecurity Threats:

Cost: Security breaches might result in immediate expenses of up to \$250,000 in containment and legal fees.

Time: An incident response might cause service to be disrupted for several days or weeks.

Quality: Following a data breach, trust and satisfaction would drop.

- Pandemic Related Disruptions:

Cost: Rental costs might decrease.

Time: Addition of 1-2 months due to workforce unavailability.

Risk	Probability (P)	Impact (I)	Risk Score (P*I)
Data Security and Cybersecurity Threats	3	5	15
Pandemic Related Disruptions	2	4	8

## 6. Legal and Compliance Risk

- Health Data Regulations:

Cost: Noncompliance penalty might exceed \$100,000, with extra legal fees.

Quality: If the system is proven to be non-compliant, its reputation may suffer significantly.

- Contractual Obligations with Service Providers:

Time: Transition could add another 1 month to the timeline.

- Liability in case of Health Incidents:

Cost: Lawsuits might result in compensation surpassing \$500,000, affecting the project's financial stability.

Risk	Probability (P)	Impact (I)	Risk Score (P*I)
Health Data Regulations	3	5	15

Contractual Obligations with Service Providers	3	4	12
Liability in case of Health Incidents	2	5	10

## Risk Mitigation Strategies

### 1. Technical Risk

- **System Integration and Interoperability:** Separating functionalities into discrete modules to facilitate integration.  
Contingency: If integration difficulties emerge, isolate the troublesome modules to keep the system operational.
- **Data Privacy and Security:** To ensure compliance, encrypt all user data and obtain insurance to cover the costs associated with possible data breaches.  
Contingency: Working with a third-party cybersecurity supplier to enhance monitoring and response capabilities.
- **Scalability Challenge:** Regular load testing is recommended to detect possible bottlenecks.  
Contingency: Implement automated scaling to deal with abrupt surges in consumption.
- **Reliability on third party APIs:** Choose API providers with strong SLAs to ensure dependability.  
Contingency: Set up notifications for API performance issues, so rapid switch to backup choices can be done.

### 2. Operational Risk

- **User Adoption and Usability:** Conduct usability testing with random elderly people to create a user-friendly design.  
Contingency: Release a pilot version to get input before the final launch, allowing for changes.
- **Dependency on Caregivers and Family:** Automated reminders could reduce the need for caregiver involvement.  
Contingency: Implement machine learning algorithms for detecting and verifying unusual entries.
- **Resource Allocation and Maintenance:** Schedule maintenance depending upon the importance and urgency for the task.  
Contingency: Prioritize critical maintenance above feature upgrades to guarantee core functionality.

### 3. Financial Risk

- High Development and Maintenance Cost: To stay on budget, keep tight track of spending and prioritize high-impact features.  
Contingency: Maintain a 10-15% contingency reserve to meet unexpected expenditures.
- Revenue Model and Profitability: To maximize user reach, provide basic features for free while charging for premium capabilities.  
Contingency: Consider partnering with healthcare providers for sponsorships or business agreements.
- Market Competition: Focus on unique and innovative features.  
Contingency: To stay in business, assess rivals on a regular basis and adjust features and pricing.

### 4. Market and Adoption Risk

- Changing Needs and Market Dynamics: Create a modular design that is easily adaptable to new technologies, providing adaptability in the future market conditions.  
Contingency: Gather user input on a regular basis to better understand changing demands and keep the system relevant.
- Stakeholder Satisfaction and Retention: Create a strategy for regular engagement with all stakeholders, such as surveys, focus groups, and beta testing of new features.  
Contingency: Encourage retention by offering loyalty programs or incentives to long-term users.

### 5. Environmental and Social Risk

- Data Security and Cybersecurity Threats: Enable multi-factor authentication and role-based access restriction.  
Contingency: Implement daily backups to safe places to ensure fast recovery.
- Pandemic Related Disruptions: Give team members secure access to work from home if regular working conditions are affected.  
Contingency: When availability changes, allow for quick work reallocation across team members.

### 6. Legal and Compliance Risk

- Health Data Regulations: Provide transparent privacy policies and specific user consent methods  
Contingency: Obtain liability insurance to reduce the financial risk of regulatory noncompliance.
- Contractual Obligations with Service Providers: Negotiate conditions that enable for a smooth transfer to alternative providers.  
Contingency: Include penalty provisions to incentivise provider compliance with SLAs.

- Liability in case of Health Incidents: Include disclaimers that emphasize that the system is a support tool, not a replacement for medical treatment.  
Contingency: Obtain insurance to protect any legal claims arising from health occurrences.

## **Solution Proposal**

### **Detailed Specifications of the Software Solution being offered:**

The ElderAid Elderly Care Management System has been built to be secure, scalable and easy-to-use that facilitates the unification of elderly care under different aspects. The architecture follows a strong technology architecture which includes a web based interface on React.js and a mobile application utilizing React Native to support range of devices. The backend is powered by Django that supports REST API, user profiles and business logic, while securing health data with management features built on MongoDB.

Each core component describes in detail the essential component of elderly care which includes approaches for health monitoring, taking medications, making appointments, making emergency calls, and even communication tools. To enhance scalability, ElderAid employs AWS or Google Cloud to enable storage management while allowing for expansion in the future of user base. Security measures during the transmission of data are provided by SSL/TLS encryption, and architecture's high degree of flexibility is complemented by provision for IoT in the future, increasing health records input channels through wearable devices. Careful privacy policies are designed to protect the data and suit Canadian PIPEDA compliance standards.

This vision on the other hand focuses upon the long term run of the solution in a manner that is reduces the burden on caregivers, increases the quality of health outcomes and enhances the quality of lives of the elderly by integrating all such workflows on a single platform which keeps pace with of the growing technologies.

### **How ElderAid Responds to the Problem or the Opportunity Presented:**

ElderAid focuses on integrating health tracking, medication management, urgent care response, and interaction in applications to confront the issues of elderly fragmentation. This addresses the credible threats posed by siloed approaches such as failure to take medication or reporting health data gradually that can lead to complication and expensive prices. ElderAid instantly increases the efficiency of families and caregivers by reminding them of important health information and providing essential information that allows them to respond quickly in case of emergencies.

Case studies demonstrate that ElderAid's uncluttered design minimizes the required caregiver efforts focused on health data input while emergency alerts are sent automatically when required. Most of the patients proceed to disregard hospital arrangements because Mr. Matthews has been unwell but ElderAid's medication compliance and appointment management features limit needless expenses due to such situations. All in all, ElderAid establishes a new benchmark in elderly management because it provides unified solutions that the competitors do not offer.

## **What ElderAid is all about – Basic Features and Functions of ElderAid System:**

1. **Health Monitoring:** Users provide their vital signs using a simple manual input that gets time stamped and tracked in a MongoDB database. Abnormal readings are flagged, and a care giver is alerted using Twilio via SMS.
2. **Medication Management:** Uses Firebase Cloud messaging service to send scheduled notifications for the user taking the medication, the report is available on the dashboard of the caregiver.
3. **Doctor Appointment Scheduling:** The application is integrated with the Google calendar API to enhance booking, rescheduling and reminder notifications through Twilio for the patients.
4. **Emergency Response System:** Caregiver and family members of patients have a panic button in the mobile app that relays customized situational alerts to the targeted people with exact location which helps to reduce time in responding to emergencies.
5. **Secure Communication:** Video calls and real time chats are made easy with WebRTC capability and Twilio, allowing the elderly, their caregivers, and healthcare providers to stay connected.
6. **Data Analysis and Reporting:** Reports on health conditions using power point advanced graphs provided by the backend analytics based on python to care givers and medical professionals assisting the care providers in modifying the care plan.
7. **User Profiles and Roles:** Unique design for different users (elderly people, caregivers, family, and their health care providers) each has a unique design that limits their access to other irrelevant features for their ease of operation.

Each feature is strategically prioritized to address the unique needs of each stakeholder while maintaining technical feasibility and market demand. For instance, the emergency response system addresses safety, while secure messaging supports mental well-being and connection.

Every feature is carefully selected to meet the needs of the different stakeholders as well as the realities of technology and the target market. As for example, safety is guaranteed by the emergency response system and mental health and connection are facilitated through secure messaging.

## **Use cases/scenarios that describe how end-users will interact and make use of the solution:**

1. **Medication Reminder:** An older adult user is notified via the mobile phone application that it is time for him or her to take medication. After taking medication,

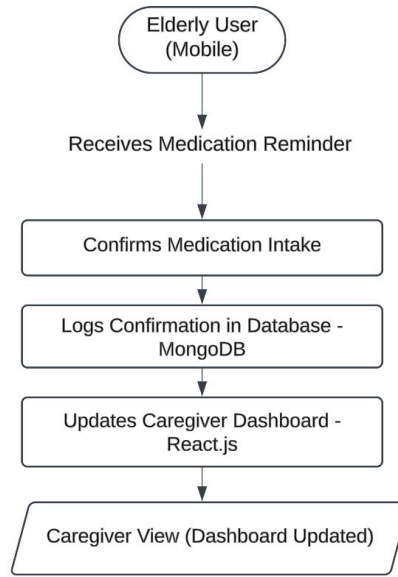
the user confirms that he or she has taken the medication, and the duties of patients are logged as an adherence level and presented on the care providers dashboard. This scenario depicts the role of reminders and adherence logging and how it helps to ease the burden on the caregivers as well as enhances medication compliance.

2. **Emergency Alert Activation:** An older adult user who has a panic button can use it in case of emergencies which can instantly alert the selected caregivers/family members along with the user's location when the panic type button is pushed. In such cases, this allows for faster response times, minimizing the impact of health risks potential in crisis situations.
3. **Health Monitoring:** A family caregiver registers the blood pressure or 'BP' of an older user into the system. A graph created on the dashboard traces such abnormalities and sustains their sustenance over a period, hence ensuring the caregivers can check on medical advice when needed timely.
4. **Doctor Appointment and Communication:** A doctor's appointment is booked by the elderly user or their caregiver using Google Calendar. The system allows the elderly user and their caregivers to remain updated on the appointments through automatic reminders and updates preventing the likelihood of missing appointments and enhancing healthcare.

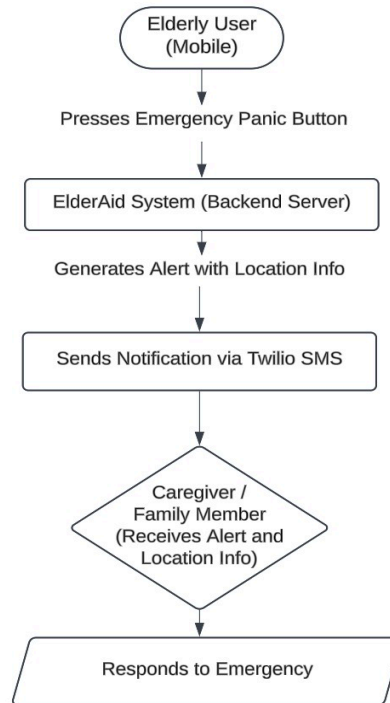
These varied use cases in this case demonstrate the core competencies of ElderAid solution in addressing the user's needs in the context of different roles, the provision of edge behavior such as responding to emergencies, and even the likely challenges of user engagement during the application of delivering such alerts and feedback timely and relevant.

**Some process flows for key use cases, to show interactions happening between the user and the system.**

1. Medication Reminder and Adherence Tracking Process Flow

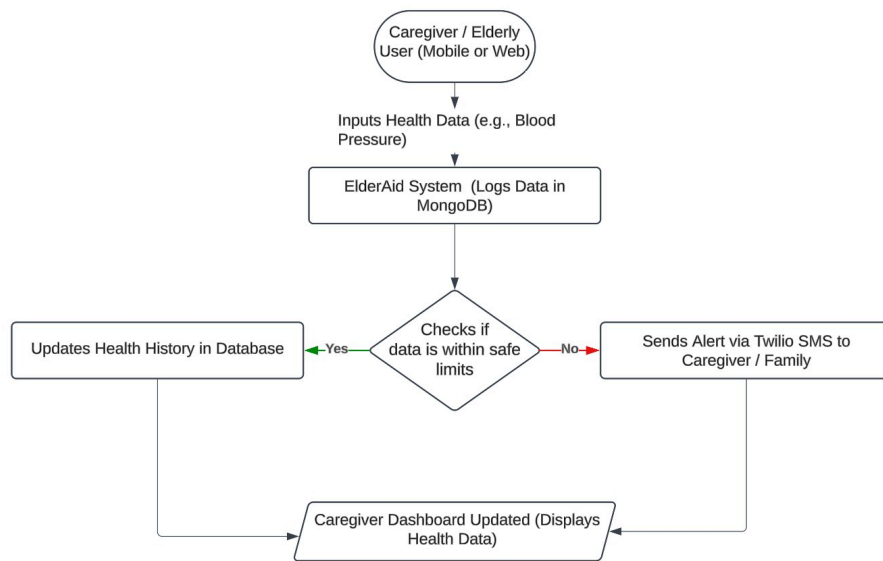


## 2. Emergency Alert Activation Process Flow

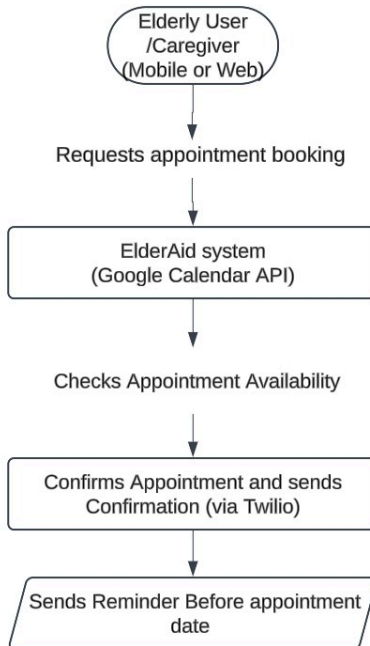


## 3. Health Monitoring and Alert Process Flow

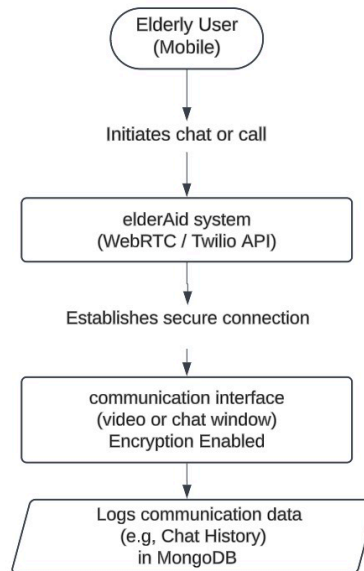




#### 4. Doctor Appointment Booking Process Flow



#### 5. Secure Communication and Video Conferencing Process Flow



### **Benefits that Users and Stakeholders Will Derive from the Solution:**

ElderAid offers wide benefits to every stakeholder involved in this case:

1. Elderly Users: Increased freedom and safety by getting medication reminders, getting health tracking, getting emergency response thereby decreasing reliance on the caregiver and improving self-care.
2. Caregivers: More organized elderly patients' health records and less supervision of the elderly's care enabling them to concentrate on nurturing quality interactions rather than routine engagements.
3. Family Members: Enhanced peace of mind from the knowledge that there will be constant health report updates and alerts in the case of an emergency despite physical distance.
4. Healthcare Providers: Having the right health information at the right time so that early steps can be taken and reliance on hospitalization is minimized resulting in efficiency of the health care system.
5. Assisted Living Facilities: Effective operational management through caregiver information access, ensuring better service delivery and marketing opportunities.

Among the immediate advantages are decrease of cognitive load for caregivers and better observance of the pharmacological regimen. Long-term associations regard operational expenditure reductions incurred by families in maintaining better health conditions for the elderly providers.

ElderAid as a solution impacts positively the elderly by enabling independence, securing the aged while encouraging proactive health management.

More importantly the improvement in the services provided by ElderAid, reduces unnecessary hospitalizations and the related costs for the family and providers.

On a larger front, ElderAid is in line with the sustainable development of digital health care solutions merging with the perfect example of centralized care that can impact the industry at large in future. As the elderly care will be more widening up with aiming at the digital solutions, the ElderAid will be setting the tone on scaling the elderly care and targeting on the combination of real time health check ups with the emergency calls.

## Budgeting

### Cost Categories:

We will use functional point analysis to decide how much funds to allocate for different cost categories.

<https://www.geeksforgeeks.org/software-engineering-functional-point-fp-analysis/>

There are several features which ElderAid will implement :

1. Health Monitoring
2. Medication Management
3. Doctor Appointment Scheduling
4. Emergency Response System
5. Secure Communication
6. Data Analysis and Reporting
7. User Profile and Role Management
8. Multi-Factor Authentication

We assign function point types and complexity based on the code base available and feedback from the developers regarding the complexity, functionality requirements , and time needed to implement these features

1. **Health Monitoring:** External Input : 2, medium complexity  
Internal Logical File : 4, medium complexity
2. **Medication Management:** External Input: 3 , medium complexity  
External Output: 3, medium complexity  
Internal Logical Function: 4, medium complexity
3. **Doctor Appointment Scheduling:**  
External Input: 3, medium complexity  
External Inquiry: 3, low complexity  
External Output: 3, medium complexity  
External Interface File: 4, medium complexity
4. **Emergency Response System:**  
External Output: 3, medium complexity  
External Input: 3, low complexity

Internal Logical Function: 4, medium complexity

**5. Secure Communication:**

External Inputs: 3, low complexity

External Outputs: 3, medium complexity

External Inquiries: 2, low complexity

Internal Logical Files: 2, low complexity

External Interface Files: 3, medium complexity

**6. Data Analysis and Reporting**

External Inputs: 3, medium complexity

External Outputs: 3, medium complexity

External Inquiries: 2, low complexity

Internal Logical Files: 4, medium complexity

External Interface Files: 3, medium complexity

**7. User Profile and Role Management**

External Inputs: 2, low complexity

External Outputs: 2, low complexity

External Inquiries: 2, low complexity

Internal Logical Files: 2, medium complexity

**8. Multi-Factor Authentication:**

External Inputs: 2, medium complexity

External Outputs: 2, medium complexity

External Inquiry: 2, medium complexity

Internal Logical Files: 3, medium complexity

External Interface File: 2, medium complexity

Feature	Function Point Type	Count	Complexity	Points for Complexity	Total Points = Count * Points for Complexity
Health Monitoring	External Input (EI)	2	Medium	4	8
	Internal Logical File (ILF)	4	Medium	10	40
Medication Management	External Input (EI)	3	Medium	4	12
	External Output (EO)	3	Medium	5	15
	Internal Logical File (ILF)	4	Medium	10	40
Doctor Appointment Scheduling	External Input (EI)	3	Medium	4	12
	External Inquiry (EQ)	3	Low	3	9
	External Output (EO)	3	Medium	5	15
	External Interface File (EIF)	4	Medium	7	28
Emergency Response System	External Output (EO)	3	Medium	5	15
	External Input (EI)	3	Low	3	9
	Internal Logical File (ILF)	4	Medium	10	40

Secure Communication	External Input (EI)	3	Low	3	9
	External Output (EO)	3	Medium	5	15
	External Inquiry (EQ)	2	Low	3	6
	Internal Logical File (ILF)	2	Low	7	14
	External Interface File (EIF)	3	Medium	7	21
Data Analysis and Reporting	External Input (EI)	3	Medium	4	12
	External Output (EO)	3	Medium	5	15
	External Inquiry (EQ)	2	Low	3	6
	Internal Logical File (ILF)	4	Medium	10	40
	External Interface File (EIF)	3	Medium	7	21
User Profile and Role Management	External Input (EI)	2	Low	3	6
	External Output (EO)	2	Low	4	8
	External Inquiry (EQ)	2	Low	3	6

	Internal Logical File (ILF)	2	Medium	10	20
Multi-Factor Authentication	External Input (EI)	2	Medium	4	8
	External Output (EO)	2	Medium	5	10
	External Inquiry (EQ)	2	Medium	4	8
	Internal Logical File (ILF)	3	Medium	10	30
	External Interface File (EIF)	2	Medium	7	14
				<b>Unadjusted Function Point Count</b>	512

Next, we calculate the Cost Adjustment Factor, we have assigned the scores based on the influence that these General System Characteristics have on ElderAid

<b>General System Characteristics (GSC)</b>	<b>Score</b>
Data Communications	3
Distributed Data Processing	4
Performance	5
Heavily Used Configuration	3
Transaction Rate	4
On-Line Data Entry	2
End-User Efficiency	4
On-Line Update	2



Complex Processing	4
Reusability	5
Installation Ease	1
Operational Ease	2
Multiple Sites	4
Facilitate Change	5

Next, calculate the Complexity Adjustment Factor by using the formula:  $CAF = 0.65 + (0.01 * \sum GSC)$

Total GSC value (from the table above) = 48

$CAF = 0.65 + (0.01 * 48)$

$CAF = 1.13$

So, Adjusted Function Point Count = Unadjusted Function Point Count \* CAF

$AFP = 512 * 1.13 = 578.56$

Based on our team's expertise level and the market forces, we have arrived at a conclusion that each function point would require 300 CAD to be implemented.

So, total budget for development phase =  $300 * 578.56$

= 173,568 CAD

Based on feedback from developers, we estimate that keeping development phase budget as a baseline, following phases will take up below percentages of the development cost

Planning : 15%

Design: 20%

Testing: 50%

Deployment and Training: 15%

Post Deployment Monitoring and Maintenance: 20%

So, we have:

Phase	%of Development Budget	Phase Budget (CAD)
Planning	15%	26,035.20
Design	20%	34,713.60
Testing	50%	86,784.00
Deployment and Training	15%	26,035.20
Post-Deployment Monitoring & Maintenance	20%	34,713.60
	Budget for Development =	173,568
	<b>Total Budget for all Phases</b>	381,849.60

## Resource Costing

Since ElderAid is an application that deals with people from different age groups and serves the critical social purpose of dealing with elderly care, we have decided to hire dedicated full-time employees who would work for the standard 40 hours per week. We decided not to go for outsourcing or for hiring workers on contract because the success of our application depends on our market image, so we want to stick with full-time employees at least for a couple of years so that a core group of employees is present who share our vision and have the passion to make ElderAid the best product in its category. After a couple of years, when we have a solid market share and enough knowledge base of our development activities, we can look to hire part-time workers, contract employees or outsource some maintenance activities depending on the market conditions.

For the first couple of years, we will operate on a work-from-home setup, saving us costs for office space, electricity, etc.

We will outsource marketing activities to a marketing agency. Based on available data, the annual budget to hire an agency= 9,000 CAD \* 12 = 108,000 CAD.

<https://www.bark.com/en/ca/marketing/marketing-agency-price-guide/>

We are going to use outsource customer service so that they can handle incoming customer complaints which could be forwarded to the developers.

As per available data, we plan to outsource level 1 customer service to a company in India which would cost around 83,200 USD annually which is 113,062 CAD.

per hour cost of customer service = 13.59 CAD

<https://www.customerly.io/blog/the-ultimate-guide-to-outsourcing-customer-service/>

We will have 2 HR managers to deal with human resource-related activities like participating in the hiring process, scheduling interviews, etc.

[https://www.glassdoor.ca/Salaries/human-resources-salary-SRCH\\_KO0,15.htm](https://www.glassdoor.ca/Salaries/human-resources-salary-SRCH_KO0,15.htm)

The annual cost for an HR manager = 64,000 CAD

Annual Budget = 128,000 CAD

Hourly payment for HR (40 hours per week) = 30.77 CAD per hour

We will use services for financial service providers like **Gusto** for services such as payroll processing and the annual cost would be 40 dollars per month plus 6 dollars per employee per month.

The total number of employees =  $2 + 6 + 5 + 2 + 1 + 1 = 17$

So per month cost =  $6 * 17 + 40 = 142$  dollars

Annual cost = 1,704 dollars which is approx. 2,376.31 CAD

<https://www.marketwatch.com/guides/business/adp-payroll-pricing/>

Currently, there are 6 full stack developers

There are 5 qa testers, 2 security testers, one project manager, and one product owner.

All the developers, testers, and security engineers along with the project manager and product owner would need to use laptops and we are going to stick to Apple MacBook Pro 16 GB RAM 512 GB storage for all team members, the cost estimation including the annual apple care for this device comes up to be 2,849.08 CAD. So, for 23 employees the cost would be 65,535.74 CAD.

Cost for experienced senior full stack developers:  $104,000 * 6 = 624,000$  CAD

Per-hour pay for senior full stack developers(40 hours per week) is **50.00 CAD per hour**.

[https://www.glassdoor.ca/Salaries/senior-stack-developer-salary-SRCH\\_KO0%2C22.htm](https://www.glassdoor.ca/Salaries/senior-stack-developer-salary-SRCH_KO0%2C22.htm)

Cost for senior security engineer:  $136,146 * 2 = 272,292$  CAD

Per hour pay for senior security engineer(40 hours per week) = 65.45 CAD per hour

<https://ca.talent.com/salary?job=senior+security+engineer>

Cost for senior QA tester:  $122697 * 2 = 245,394$  CAD

Per hour pay for QA Tester(40 hours per week) = 58.99 CAD per hour

<https://ca.talent.com/salary?job=senior+qa>

Cost of a Project Manager = 80,000 CAD

Per hour pay for Project Manager (40 hours per week) = 38.46 CAD per hour

[https://www.glassdoor.ca/Salaries/experienced-project-manager-salary-SRCH\\_KO0%2C27.htm](https://www.glassdoor.ca/Salaries/experienced-project-manager-salary-SRCH_KO0%2C27.htm)

Cost of a Product Owner = 89,000 CAD

Per hour pay for Product Owner (40 hours per week) = 42.79 CAD per hour  
[https://www.glassdoor.ca/Salaries/product-owner-salary-SRCH\\_KO0%2C13.htm](https://www.glassdoor.ca/Salaries/product-owner-salary-SRCH_KO0%2C13.htm)

We are planning a marketing cost budget of 40,000 dollars annually based on available statistics.  
<https://www.transcenddigital.com/blog/what-is-a-good-marketing-budget-for-startup>  
<https://www.tbdc.com/setting-a-marketing-budget-for-your-startup/>

We are not planning to use any software that involves licensing and we want to stick to open-source software including Apache Open Office, Google Docs, and the free version of Slack to cut costs over a couple of years.

Planning an annual legal cost of 10,000 CAD based on available data.  
<https://www.cimphony.ai/insights/startup-legal-fees-how-much-to-budget-in-2024>

We will use two AWS services: EC2, S3, and Load Balancer to host our application on their servers, static file hosting, use the backup services, and deal with high traffic.

Here's the breakdown:

Annual Cost of AWS S3: \$0.023 per GB per month for 500TB = \$141,312 annually.  
<https://aws.amazon.com/s3/pricing/>

Annual Cost of AWS EC2: <https://aws.amazon.com/ec2/pricing/>

We will be using the t2.xlarge instance type whose hourly rate is \$0.2266. So annually, the cost would come up to be \$1,985.02

Annual cost of AWS Load Balancer: **\$22.42 per month = \$269.04 annually**

This pricing is based on the pricing example 1 from <https://aws.amazon.com/elasticloadbalancing/pricing/?nc=sn&loc=3> assuming that the hosted application receives an average of one new connection per second, each lasting two minutes.

### Summary:

Resource Category	Annual Cost (CAD)	Hourly Rate (CAD)
Human Resources - Full Stack Developers	624,000	50
Human Resources - QA Testers	245,394	58.99
Human Resources - Security Engineers	272,292	65.45
Human Resources - Project Manager	80,000	38.46

Human Resources - Product Owner	89,000	42.79
Human Resources - HR Managers	128,000	30.77
Technology - Laptops (MacBook Pro)	65,535.74	-
External Services - Marketing	108,000	-
External Services - Level 1 Customer Support	113,062	13,59
External Services - Legal	10,000	-
External Services - Financial (Payroll Service)	2,376.31	-
Cloud Services - AWS S3	141,312	-
Cloud Services - AWS EC2 (t2.xlarge)	1,985.02	-
Cloud Services - AWS Load Balancer	269.04	
<b>Total Resource Category Budget</b>	<b>1,881,226</b>	

## Contingency Budget

To deal with the technical risk of data privacy and security, we would be partnering with a third-party cybersecurity supplier to enhance monitoring and response capabilities. Based on available data, we would be allocating a budget of 4,000 CAD per month which is slightly higher than \$3,500 per month figure as mentioned in:

<https://www.vc3.com/blog/managed-cyber-security-services-cost>

1. So, annual cost for hiring a third party cybersecurity supplier = 42,000 CAD  
Applies to: Testing, Maintenance, Implementation

To deal with the risk of scalability, we would perform regular load testing (free tools like Apache JMeter could be used). In case the need for scalability arises, we could use AWS AutoScaling along with AWS Loadbalancer and there is no additional cost for using this service.

<https://aws.amazon.com/autoscaling/pricing/>

2. To deal with the financial risk, we set a contingency budget of 15% of the total budget.

Total budget =

Contingency budget for financial risk =

[https://biz.libretexts.org/Courses/Northeast\\_Wisconsin\\_Technical\\_College/Project\\_Management\\_Fundamentals\\_\(NWTC\)/05%3A\\_Resource\\_Management\\_Budget\\_and\\_Procurement/5.10%3A\\_Contingencies](https://biz.libretexts.org/Courses/Northeast_Wisconsin_Technical_College/Project_Management_Fundamentals_(NWTC)/05%3A_Resource_Management_Budget_and_Procurement/5.10%3A_Contingencies)

Applies to: Throughout the entire project; all phases

To ensure stakeholder retention by offering loyalty programs/incentives to long-term users, we need some additional budget.

Let us say that we give 50 CAD refund to every customer who has been using our software solution for more than 12 months once a quarter,

We would end up spending = number of users \* 50 \* 4 CAD annually.

It takes time, at least an year for people to gain trust on the application and us as a company to get hold of all the organizational processes and accumulate the knowledge. We have set a target of gaining 10,000 customers in the first year and 20,000 by the end of the second year. Considering the very optimistic case that we gained more than 15% of what we expected at the end of the first year, i.e., 11,500. Assuming, all of them end up using the software solution for at least 12 months, we need to spend  $1150 * 50 * 4 = 230,000$  CAD annually.

Note: if the % of customer gain is more than 15%, we would abandon the loyalty program/incentive to save costs.

3. So, contingency budget for maintaining Stakeholder Satisfaction and Retention = 230,000 CAD annually

Applies to: Deployment and Maintenance

4. To protect our company against legal and compliance, we plan to take up Professional Liability Insurance and Cyber Liability Insurance. For Professional Liability insurance, we keep a contingency budget of 5000 CAD annually

<https://www.easycover.ca/blog/average-cost-of-professional-liability-insurance/>

Applies to: Deployment and Testing

5. And for cyber liability insurance, we set a contingency budget of 2000 CAD annually which is double than usual amount of 1000 CAD annually as per the available data:

<https://www.cmbinsurance.ca/lp/cyber-insurance/>

Applies to: Implementation, Deployment and Maintenance

Cybersecurity related risks are highly critical in nature and may cost a company huge financial losses if legal cases are filed against the company for any breach of user data, especially their health data.

6. To deal with the risk of delays in hiring a new employees or attrition, we have set aside 15% more salary budget, ie.

Employee Types	Current Hourly Rate (CAD)	15% of Hourly Rate (CAD)	15% Annual Equivalent (CAD)	Number of Employees	Total Annual Equivalent for All Employees (CAD)
Full Stack Developers	50	7.5	15,600	6	93,600
QA Testers	58.99	8.85	18,408	5	92,040
Security Engineers	65.45	9.82	20,435.20	2	40,870.40
Project Manager	38.46	5.77	11,993.60	1	11,993.60
Product Owner	42.79	6.42	13,353.60	1	13,353.60
HR Managers	30.77	4.62	9,609.60	2	19,219.20
				Total	271,077

So, in total, we have set aside 271,077 CAD. This makes sure that we have enough budget if most of the employees leave or we need to scale up the teams by even doubling the employees due to increased work.

Note: 15% annual equivalent amount is calculated by multiplying 15% hourly rate by 52 weeks and then by 40 hours.

Applies to: Planning, Maintenance, Implementation

Total contingency budget = annual cost for hiring a third party cybersecurity supplier + Contingency budget for financial risk + contingency budget for maintaining Stakeholder Satisfaction and Retention + contingency budget for Professional Liability insurance + contingency budget for Cyber Liability Insurance + contingency budget for hiring delays/attrition

= 42,000 + + 230,000 + 5,000 + 2,000 + 271,077

Total Contingency Budget = 550,077 + Contingency budget for financial risk

We are keeping aside 15% of the total budget to mitigate the financial risk.

Contingency budget for financial risk =  $0.15 * (\text{budget for all phases} + \text{budget for all resource categories}) = 0.15 * (381,849.60 + 1,881,226) = 339,461.34 \text{ CAD}$

**Total Contingency Budget = 339,461.34 CAD**