**Course Project - System Design and Implementation - Final Report**

**IS316, Spring Semester**

**Sahil lnu, Amani Mulaudzi, Angela Mast and Marina Perdiguero**

**May 4, 2025**

**Professor David Monchusie**

**Park University**

# Table of Contents

[I. Executive Summary 4](#_ia1lr5ku7fam)

[II. System Proposal 5](#_t2bpfjontow9)

[III. System Acquisition Strategy 14](#_8c0hus5dzcii)

[IV. System Architecture 17](#_9rsga6xm9bzs)

[V. Hardware and Software 20](#_5yqsuvz6btxg)

[VI. User Interface 27](#_r9clecrl2vbd)

[VII. Physical Process Model 30](#_xzpctt92sd01)

[VIII. Physical Data Model 31](#_m2terbunnrwe)

[IX. Test plan 32](#_1gnwtp9r0jkp)

[X. Migration Plan 44](#_1137fwjucuxc)

[XI. Training Plan 52](#_to7j6odw1via)

[XII. System Support and Maintenance Plan 58](#_3iei39s09s5)

[XIII. Conclusion 60](#_txcol2p6vv6i)

[References 61](#_biqzkctdguko)

[Appendix A - Project Gantt Chart 63](#_zgqk02oopug0)

[Appendix B - User Interface Design 65](#_q02df77q1f5u)

[Appendix C - Structure Charts 69](#_keoabyvuqncv)

[Appendix D - Program Specifications 72](#_6kcq13dxwpwj)

[Appendix E - Data Dictionary 79](#_kfmkh6g3lhh0)

# I. Executive Summary

The South Dakota Department of Labor’s Workers’ Compensation Division is struggling with a physical file system. Searching for claims in space constraining file cabinets is a long and cumbersome task for the clerks. Addition of new claims causes storage issues, and it is overall a long process that can affect customer and clerk satisfaction. The addition of a digital system to manage claims would provide an exponentially faster experience for the department of labor and the claimants. There will be profit due to the lesser need of clerks because of the efficient system. This could save the department a substantial amount of money and time.

# II. System Proposal

# Description of the System

The Workers' Compensation Digital Case Management System is a full-featured digital system created to help the South Dakota Department of Labor's Workers' Compensation division manage injury claims, reports, and medical records more efficiently. By switching to an online, safe, and user-friendly platform, the system seeks to eradicate the inefficiencies connected with paper-based procedures.

Employees, medical providers, and clerks will be able to conveniently access, upload, or manage case-related data thanks to the system. This system will centralize data, streamline processes, enhance security, and ensure regulatory compliance, addressing long-standing issues and improving service delivery for stakeholders. All stakeholders will benefit from improved accessibility and usability thanks to sophisticated search features and simple navigation.

# Key Aspects of System Feasibility

## **Technical Feasibility**

### The project team’s limited experience with the current paper-based system and its lack of digital integration presents a significant challenge, requiring the development of new processes and procedures. While the shift to a fully digital solution introduces a steep learning curve, leveraging prior experience with government or insurance-related applications can help mitigate these risks.

#### ***Technology Familiarity:*** The proposed solution incorporates advanced technologies, including digital document management, database systems, and web-based applications. If the internal team lacks experience in these areas—such as database administration, web development, or cybersecurity—the likelihood of encountering difficulties increases. However, this risk can be managed by engaging experienced vendors or hiring IT professionals with relevant expertise.

#### ***Project Scope and Scale:*** The project is ambitious, encompassing the creation of a secure online platform, integration of a comprehensive database system, and digitization of a vast repository of records. Such a large-scale undertaking naturally involves greater complexity and inherent risks. Effective project management will be essential, utilizing phased implementation, clear milestones, and regular progress reviews to maintain control and minimize risks.

### ***System Compatibility:***The department’s manual, paper-based system is not designed to integrate seamlessly with modern technology. The new system must ensure secure access for authorized users and maintain compatibility with other state systems. Challenges such as data migration and compliance with legal and regulatory standards, including HIPAA, represent moderate risks.

**Economic Feasibility  
*Development costs:*** They include labor for programmers, analysts, and project managers, along with expenses for data entry to transfer physical records into a database. Hardware will be required to replace the current physical file system, and software costs will vary depending on whether it is outsourced or created in-house.  
***Operating costs:***They include labor for a small team to maintain and update the software, annual licensing fees for outsourced software, and ongoing training costs for staff due to system updates and new hires (Singal et al., 2023).  
***Tangible benefits:*** They include faster service, allowing for more claims to be processed and fewer employees required. The online database enables increased client capacity and reduces paper and storage costs, resulting in significant savings.

***Intangible benefits:*** They include greater storage capacity, improved accessibility to client records from anywhere, better customer service, and enhanced reputation due to faster and more reliable service. Additionally, digital systems improve security and eliminate the risk of misplaced files.

|  | Devel | Year 1 | Year 2 | Year 3 | Year 4 | Year 5 | Total |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tangible Benefits: |  |  |  |  |  |  |  |
| Reduction of Staff |  | $250,000 | $250,000 | $250,000 | $250,000 | $250,000 | $1,250,000 |
| Reduced paper storage costs |  | $5,000 | $5,000 | $5,000 | $5,000 | $5,000 | $25,000 |
| Ability to hold more Clients |  | $15,000 | $15,000 | $15,000 | $15,000 | $15,000 | $75,000 |
| Improved accuracy (fewer errors) |  | $10,000 | $10,000 | $10,000 | $10,000 | $10,000 | $50,000 |
| Total Tangible Benefits |  | $280,000 | $280,000 | $280,000 | $280,000 | $280,000 | $1,400,000 |
| Development Costs: |  |  |  |  |  |  |  |
| Labour | $420,000 |  |  |  |  |  | $420,000 |
| Hardware Purchase | $150,000 |  |  |  |  |  | $150,000 |
| Data Entry | $20,000 |  |  |  |  |  | $20,000 |
| Software | $80,000 |  |  |  |  |  | $80,000 |
| Total Development Cost | $670,000 |  |  |  |  |  | $670,000 |
| Operational Costs |  |  |  |  |  |  |  |
| Software Maintenance |  | $5,000 | $5,000 | $5,000 | $5,000 | $5,000 | $25,000 |
| Software Licensing |  | $1,000 | $1,000 | $1,000 | $1,000 | $1,000 | $5,000 |
| Training and Development |  | $20,000 | $20,000 | $20,000 | $20,000 | $20,000 | $100,000 |
| Total Operational Costs |  | $26,000 | $26,000 | $26,000 | $26,000 | $26,000 | $130,000 |
| Total Costs | $670,000 | $26,000 | $26,000 | $26,000 | $26,000 | $26,000 | $800,000 |
| Total Benefits - Total Costs | -$670,000 | $254,000 | $254,000 | $254,000 | $254,000 | $254,000 | $600,000 |
| Cumulative Net Cash Flow | -$670,000 | -$416,000 | -$162,000 | $92,000 | $346,000 | $600,000 |  |
| Return on Investment (ROI) |  |  |  |  |  |  | 75% |
| Payback Period | 3.64 year |  |  |  |  |  |  |

## **Organizational Feasibility**

#### ***Strategic Alignment:*** The project aligns with the goals of the South Dakota Department of Labor, particularly the Workers’ Compensation Division, by addressing the inefficiencies of its outdated paper-based system. Transitioning to a digital system will streamline processes, enable faster responses to inquiries, and enhance customer service, ensuring fair and timely handling of workers' compensation claims.

#### ***Project Champions:*** Important leaders in the departments will push for the changeover and make sure that funds and personnel are provided. Their assistance is crucial to advancing the project, getting over opposition, and keeping the department's modernization objectives front and center.

#### ***Senior Management:*** In order to approve resources and highlight the significance of the project, senior management support is essential. Their support will guarantee proper funding and prioritization, bringing the project into line with organizational goals.

#### ***Users:*** Clerks and administrative staff, the primary users, will benefit from the new system’s efficiency. By enabling quick digital searches for claims, the system will reduce errors and save time. User involvement in the design process will ensure a user-friendly system and high adoption rates.

#### The project offers substantial long-term economic and operational benefits, including reduced costs, improved efficiency, and better service delivery. With support from champions, senior management, and end-users, the digital system will modernize the division’s operations, making it a sound investment that aligns with the department’s mission.

# Key Aspects of Business Requirements

## **User Requirements**

The user requirements would be a digital solution that enables efficient retrieval and management of claim records through a secure database searchable by name, address, or date of injury. The system should support online submission of injury reports and medical records in various formats, automatically organize and archive files, and provide real-time updates on claim status. It must include secure access with role-based permissions to protect sensitive data, comply with legal standards like HIPAA, and handle data growth of 5% annually. Additionally, features like notifications and prioritization of severe cases would improve communication and streamline operations.

## **Functional Requirements**

The system's functional requirements address both process-oriented and information-oriented needs to ensure efficient and organized operations.

***Process-Oriented Requirements:*** The system must provide the capability to upload records online in multiple formats, such as PDFs and faxes, and integrate medical records from healthcare professionals. Additionally, it must include a robust search functionality to access individuals’ records by name. To enhance tracking, the system should document the dates when records are uploaded, ensuring accurate record-keeping and audit trails.

***Information-Oriented Requirements:*** The system is designed to support a storage capacity of 20,000 files, with an additional 5% growth per year to accommodate increasing data needs. To manage archival efficiency, the system will automatically archive files that have not been accessed in six years. Furthermore, it must categorize documents within each individual's file by type and enable users to filter or sort documents based on type and date, optimizing retrieval and organization.

## **Nonfunctional Requirements**

### ***Operational Requirements:*** The system needs to be compatible with the current software and hardware. Medical professionals should be able to submit reports using an easy-to-use interface, and clerks without any prior knowledge should find it straightforward to use. It must also work with different operating systems and browsers, such as Windows, Chrome, Edge, and Mac.

***Performance Requirements:*** The system will be accessible 24/7, except during maintenance, and will support up to 600 concurrent users without lag. Records will be retrieved within five seconds when searched by name or claim number, ensuring high performance and efficiency.

### ***Security Requirements:*** The system must log all user activities, including login attempts, data access, and modifications, to ensure auditing and compliance. Database access should be restricted to authorized employees of the South Dakota Department of Labor, Worker’s Compensation Division. File uploads must be limited to verified medical professionals and claimants. Backups must be encrypted and performed daily, with a disaster recovery plan that ensures system recovery within two hours of a security breach.

### ***Cultural and Political Requirements:*** The system must comply with federal and state laws related to the handling of medical and personal data. It should include multilingual support to serve South Dakota residents who do not speak English. The system must also be designed for easy updates to accommodate future changes in labor and compensation laws.

## **System Requirements**

This system will provide a secure, scalable, and user-friendly solution for managing claims and medical records, meeting all operational and regulatory requirements while ensuring efficiency and accessibility (Katz et al., 2024).

***Database and Storage Requirements:*** The system will use a relational database like MySQL or PostgreSQL, supporting 20,000 files initially with 5% annual growth. It will include full-text search for efficient retrieval by name or date of injury. Files inactive for six years will be automatically archived, and documents categorized by type and date for better organization.

***User Interface Requirements:*** The interface will be intuitive, requiring no prior training for clerks or medical practitioners. It will support multiple browsers, including Mac, Windows, Chrome, and Edge, and adhere to accessibility standards. Multilingual support will ensure inclusivity, and features like hotkeys and logical navigation will enhance usability.

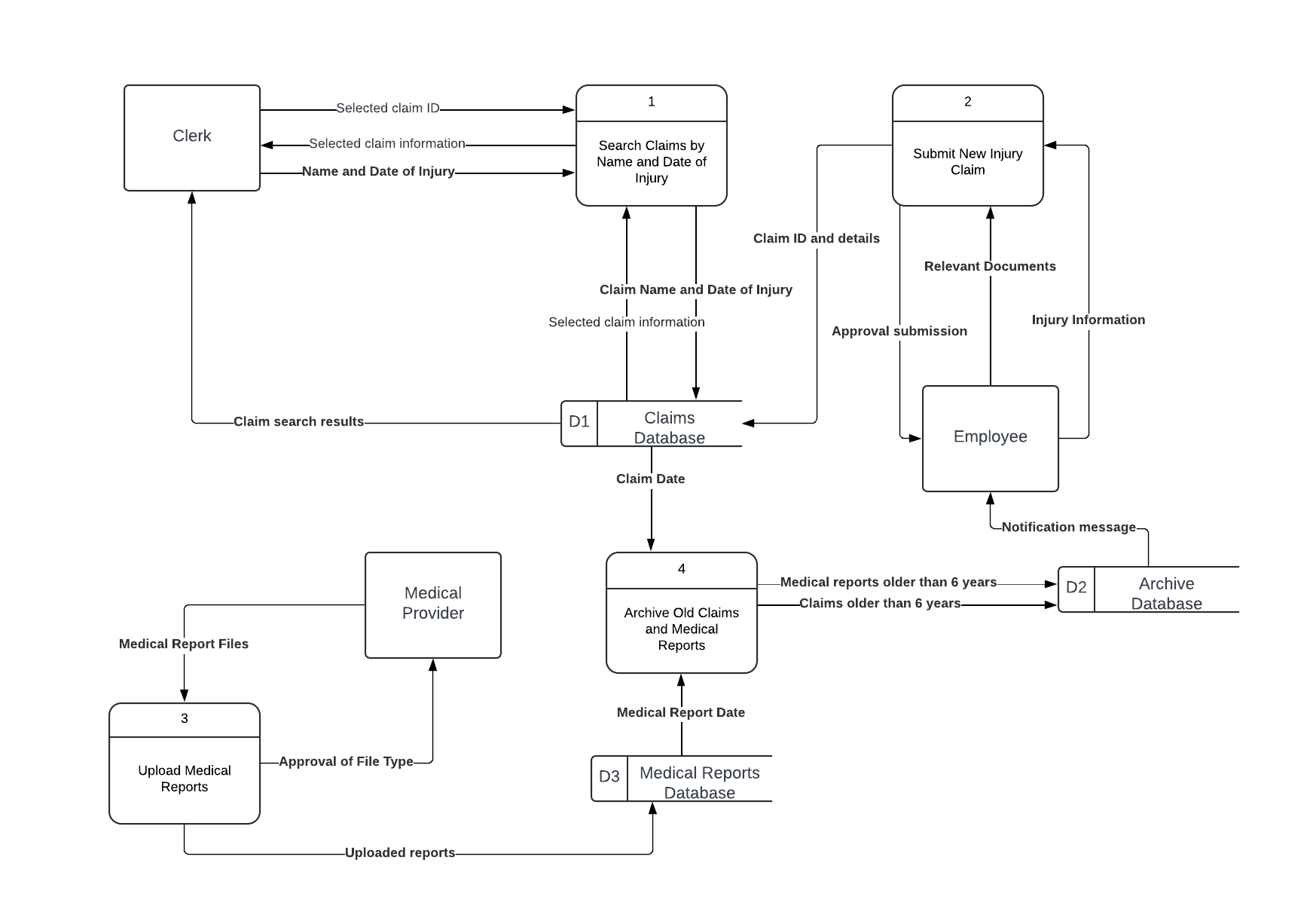
***Security Requirements:*** Role-based access control (RBAC) will ensure data security, restricting access based on user roles. User activities will be logged, and data transmissions encrypted with TLS. Files will use AES-256 encryption, with daily backups and a disaster recovery plan ensuring recovery within two hours. Multi-factor authentication will be mandatory for all users.

***Integration and Upload Requirements:*** The system will allow file uploads in various formats, like PDFs and images, and integrate directly with healthcare systems via secure APIs. Upload timestamps will provide an accurate audit trail, maintaining a transparent record of all activity.

***Compliance and Legal Standards:*** The system will comply with HIPAA and all state and federal data-handling laws. It will be designed for easy updates to adapt to future regulatory changes, ensuring ongoing compliance and operational continuity.

# General Concept of Required System Processes

## **Level 0 DFD**



## 

## **Process Descriptions**

**1. Search Claims by Name or Date of Injury**

This process allows the clerk to retrieve claim information from the Claims Database by searching for a specific employee's name or the date of injury. It provides quick access to relevant claim details, ensuring efficient case handling. The process returns the requested information, enabling the clerk to review or update claims as necessary.

**2. Submit New Injury Claim**

When an employee files a new injury claim, the clerk collects the required information, such as the employee’s details, injury specifics, and any accompanying documentation. This data is entered into the system and stored in the Claims Database. The process ensures that new claims are accurately recorded and accessible for further processing or follow-up.

**3. Upload Medical Reports**

The medical provider submits medical reports related to an employee’s injury, including treatment details and recommendations. The clerk uploads these reports to the Medical Reports Database, linking them to the corresponding claim. This process ensures all relevant medical documentation is securely stored and associated with the appropriate case.

**4. Archive Old Claims and Medical Reports**

To maintain an organized and efficient system, older claims and their associated medical reports that are no longer active are moved to the Archive Database. This process ensures the active databases remain manageable while still preserving older records for compliance and reference purposes.

# III. System Acquisition Strategy

**Alternative Matrix Analysis**

The Alternative Matrix provides a structured evaluation of three different approaches—Make, Buy, and Outsource—for developing a system. The evaluation is based on three key categories: Technical Issues, Economic Issues, and Organizational Issues, with each criterion assigned a weight (1-5) based on its importance. Each alternative is scored on a scale of 1-10, and the weighted score is calculated by multiplying the weight by the given score.

***Make (Total Score: 399)***

* This option receives the highest total score, indicating that developing the system internally is the most advantageous.
* Strengths include high scores in System Customization, Security and Compliance, and Long-Term Maintainability, emphasizing strong control over technical requirements and future flexibility.
* It also scores well in ROI and Internal Expertise, suggesting that while initial development costs are higher, the return over time and organizational capability to manage the system are significant assets.
* Weaknesses include relatively low scores in Development Costs and Implementation Time, reflecting the high initial investment and extended project timeline required for in-house development.

***Buy (Total Score: 394)***

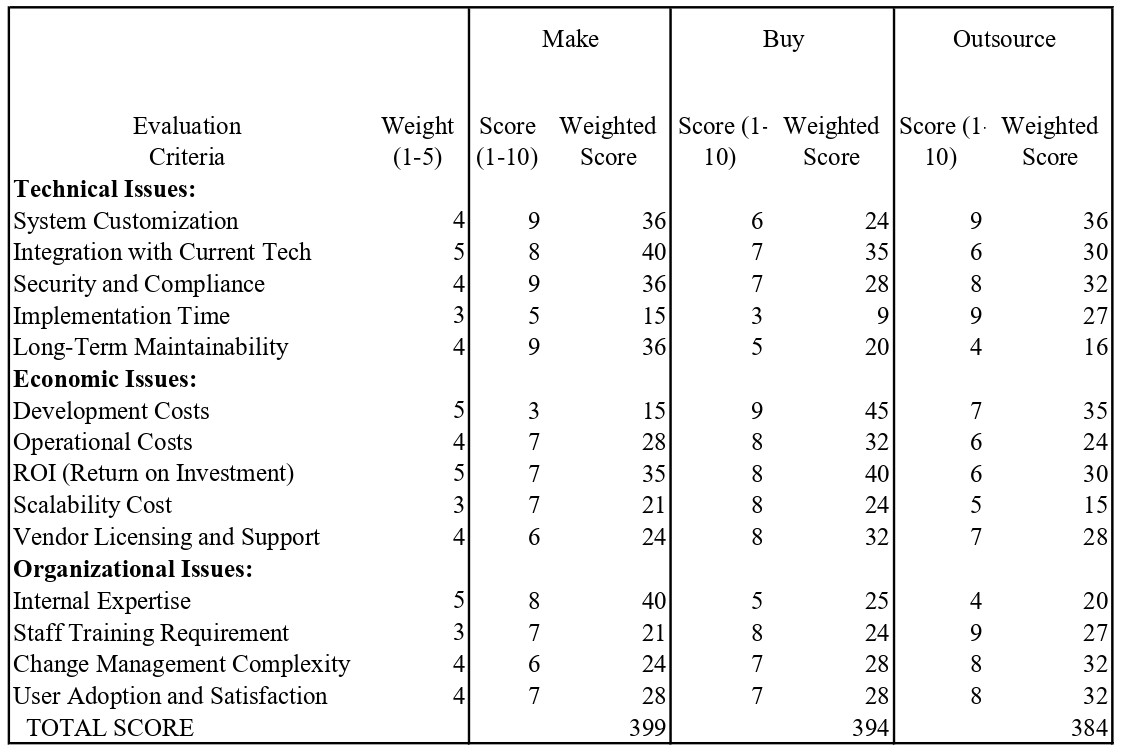
* The Buy option ranks closely behind Make, making it a strong contender for organizations seeking a balance between speed and functionality.
* It excels in Development Costs, Operational Costs, and Implementation Time, offering a cost-effective and rapid deployment option.
* Scores in ROI and Vendor Licensing and Support are also strong, making this option financially appealing.
* However, it lags behind in System Customization and Internal Expertise, suggesting potential limitations in tailoring the system and the need to rely on vendor knowledge and support.

***Outsource (Total Score: 384)***

* Outsourcing is a viable third option with competitive scores but slightly lower than Make and Buy.
* It performs well in Implementation Time and Vendor Licensing and Support, indicating quicker delivery and support availability from external vendors.
* Strong points also include Staff Training Requirement and User Adoption, meaning a smoother rollout for users.
* Lower scores in System Customization, Long-Term Maintainability, and Scalability Cost indicate potential challenges in adapting the system over time and maintaining long-term value.

***Conclusion***

Based on the evaluation matrix, Make is the most favorable option due to its strength in technical control, long-term maintainability, and strong organizational fit, despite higher upfront costs and longer development time. Buy is a very close alternative, especially suitable for organizations needing quicker implementation and lower initial costs. Outsource, while viable, presents more trade-offs in long-term flexibility and customization, making it the least preferred among the three in this context.



# IV. System Architecture

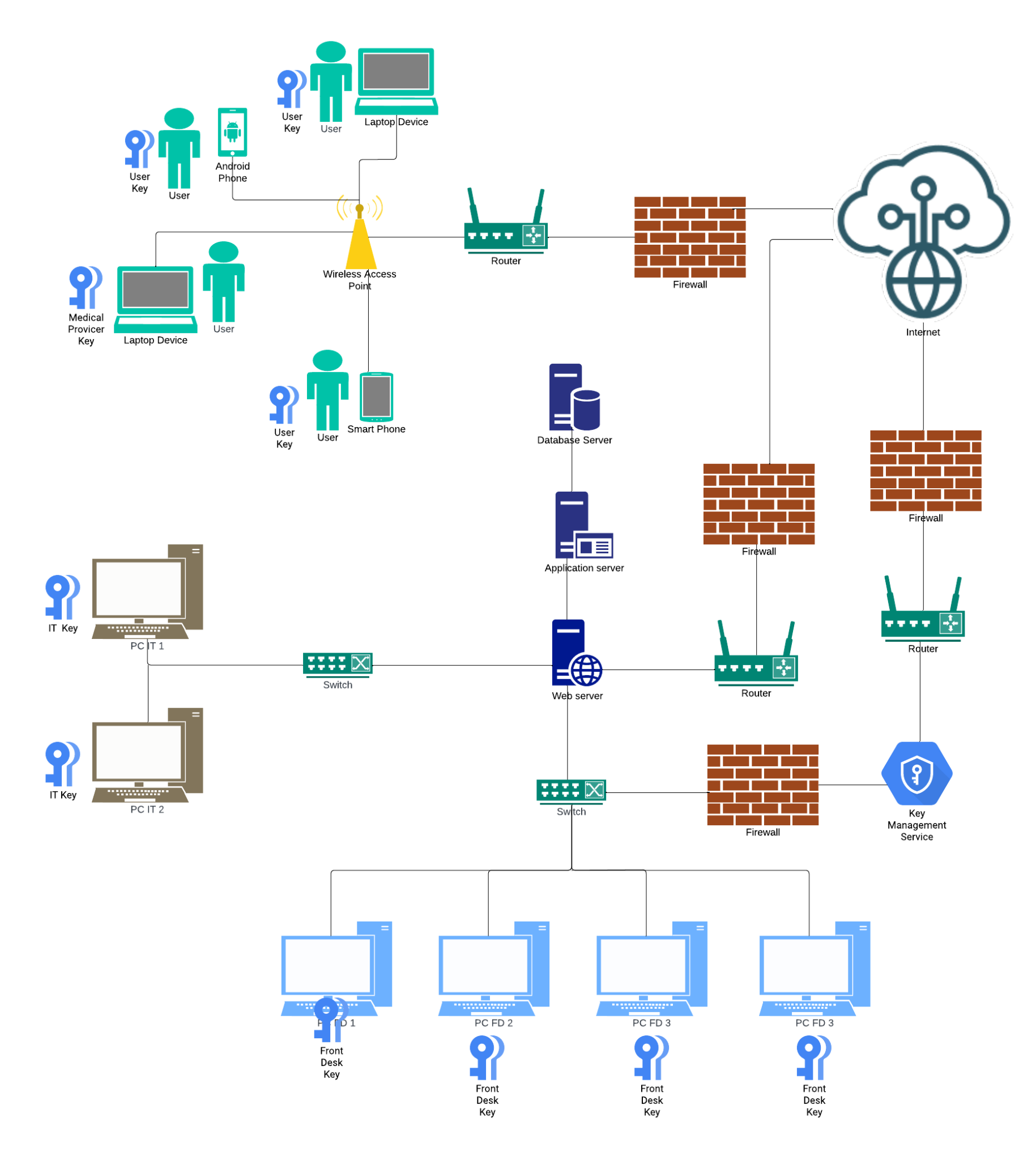
**Server-Side Description**

In the South Dakota Department of Labor Workers’ Compensation Division, there will be four computers provided for the front desk clerks (PC FD 1-4). Each of these computers will have a front desk key, providing them with the access to their needed privileges in the system. These four desktops will then be physically connected to a switch, which then connects to the Web Server. There are two other computers that will be used by the IT section of this department, labeled PC IT 1-2. These desktops will also be given IT keys, which provides the needed access to administrative permissions. The IT PCs will be connected to a separate switch, which then connects to the Web Server. Both the IT PCs and the front desk PCs will be providing the presentation logic for the system. This system architecture will use a N-tiered client-server approach. The web server will handle the web-related logic, such as the website for submitting claims or medical reports. The web server will connect to the application server, which provides management of business logic such as processing new claims to be sent to the database server. And it will also connect to the Internet through a router with a firewall for security purposes. Lastly, the application server is then connected to the database server, which is responsible for database tasks. These tasks include retrieving data, changing data, uploading data, and holding data (such as claims, medical reports, timestamps, etc.).

The switch connected to the front desk PCs (PC FD 1-4) will then connect to a firewall, which ensures enhanced security for the confidential information contained in the servers. This firewall then is connected to the key management service, which is in charge of managing and maintaining keys and permissions for each user. The key management service is then connected to another firewall. It is important to have a firewall on both sides of the key management service because it can lead to major security breaches if the key management service is tampered with. Following this firewall is the internet, which is how the users/clients can access the system through an internet connection.

**Client-Side Description**

Connected to the internet is a firewall, which can protect the wireless access point from being exploited. Following the firewall is the wireless access point that provides a wireless connection to the internet. Users can now access the system using the internet, and all of these client users are provided with a user key. The user key only provides the clients with access to submit claims and review their records. There is also a medical provider key, which allows the medical provider to search clients and submit medical records pertaining to the client. All of these users can access the system via website using mobile devices, desktop computers, or laptops.



# V. Hardware and Software

**Hardware Specifications**

The hardware used in the system must be high-performing, secure, scalable, and reliable to handle the daily workload of processing workers' compensation claims efficiently. Below is a detailed breakdown of the recommended hardware:

***Servers***

Servers form the backbone of the system, hosting the database, application, and web services. The choice of servers depends on the expected workload, security requirements, and scalability needs.

* Type: Hybrid deployment (on-premises and cloud-based)
* Processor:
  + Intel Xeon Gold 6330 (32-core, 2.0 GHz)
* Memory (RAM): 128GB DDR4 ECC RAM, expandable up to 256GB
* Storage:
  + Primary: 10TB NVMe SSD with RAID 10 for redundancy
  + Backup: 50TB NAS (Network-Attached Storage)
* Operating System (OS):
  + Windows Server 2022 (for compatibility with Microsoft systems)
* Security Features:
  + Hardware-based encryption (AES-256)
  + Intrusion detection and prevention systems (IDPS)
  + Firewalls and DDoS protection

***Workstations for Clerks & Administrators***

The workstations must be responsive and capable of handling document processing, case tracking, and communications.

* Processor: Intel Core i7-13700K or AMD Ryzen 7 7700X
* Memory (RAM): 16GB DDR4 (expandable to 32GB)
* Storage: 512GB SSD (NVMe) + 1TB HDD
* Display: Dell 24 Monitor - SE2425H
* Peripherals:
  + Ergonomic keyboard and mouse
  + Dual-monitor setup for improved efficiency

***Network Infrastructure***

A reliable, high-speed, and secure network infrastructure is crucial for the system to function efficiently.

* Firewall: Cisco Firepower 1000 Series (enterprise-grade security)
* Router: Cisco ISR 4000 Series with built-in VPN support
* Switches: Cisco Catalyst 9300 (Layer 3) for internal network communication
* Wireless Access Points: Aruba AP-515 for secure and fast Wi-Fi access
* Security Measures:
  + VLAN segmentation for better traffic management
  + Role-Based Access Control (RBAC)

***Backup and Disaster Recovery***

Ensuring data integrity and availability is a priority, so the system includes robust backup and recovery mechanisms.

* Cloud Backup Provider: AWS S3 or Microsoft Azure Backup
* On-Prem Backup Solution: Veeam Backup & Replication
* Backup Strategy:
  + Daily incremental backups
  + Weekly full backups
  + Retention policy: Data stored for six months

***Security Hardware***

To protect sensitive case data, the system includes physical and digital security measures.

* Multi-Factor Authentication (MFA) Devices: Google Titan Security Keys
* Access Control:
  + Biometric authentication (facial recognition)
  + Smart card readers for physical security

**Software Specifications**

The software used in the system must be efficient, secure, and scalable to support case processing, document management, and user access control. Below are the key software components:

***Operating System***

* Servers:
  + Windows Server 2022 (best for Microsoft ecosystem compatibility)
  + Ubuntu Server 22.04 LTS (preferred for security and cost-effectiveness)
* Workstations:
  + Windows 11 Professional
  + macOS Monterey (for Apple-based devices)

***Database Management System (DBMS)***

The database must be highly available, secure, and support structured data storage.

* Type: Relational Database Management System (RDBMS)
* Software:
  + Microsoft SQL Server 2022
* Storage Capacity: 20,000+ records - 15TB, scalable with 5% annual growth
* Backup Strategy: Automated daily backups with encryption

***Application Development***

The backend and frontend technologies must support fast performance and secure transactions.

* Backend:
  + Python (Django Framework) or Node.js
* Frontend:
  + React.js with Material UI for a responsive and modern UI
* API Management:
  + RESTful APIs using FastAPI or Express.js
* Authentication:
  + OAuth 2.0
  + JWT (JSON Web Token)
  + Role-Based Access Control (RBAC)

***Security Software***

The system must have advanced security to protect confidential workers' compensation data.

* Antivirus & Endpoint Protection:
  + CrowdStrike Falcon
  + Windows Defender ATP
* Data Encryption:
  + AES-256 encryption for stored data
  + TLS 1.3 for data in transit
* Access Control:
  + Okta Identity Management
  + Microsoft Active Directory (AD)

***Document Management System (DMS)***

Efficient document processing and storage are essential for managing case files.

* Software:
  + Alfresco Community Edition or OpenKM
* Features:
  + OCR (Optical Character Recognition) for scanned documents
  + Version control and role-based access

***Cloud & Virtualization***

To ensure scalability and remote access, the system utilizes cloud technologies.

* Cloud Hosting:
  + Microsoft Azure VM
* Virtualization:
  + VMware vSphere

***Compliance & Audit Software***

Since the system deals with sensitive employee compensation data, compliance is critical.

* Regulatory Compliance Tools:
  + HIPAA & GDPR Compliance Checker
* Audit Logging:
  + Elastic Stack (ELK) for monitoring and log management

***Collaboration & Communication***

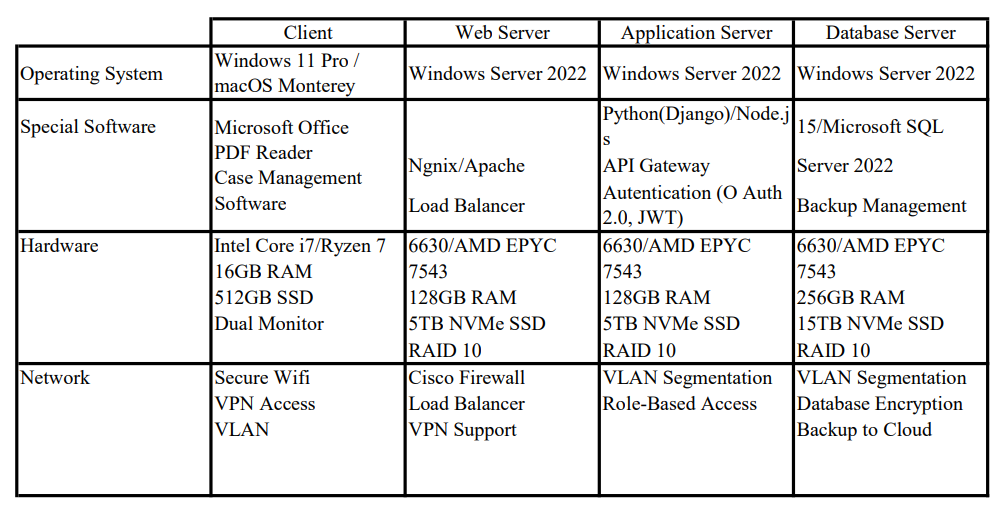
For seamless internal communication among case handlers, legal teams, and administrators, the system includes:

* Email System:
  + Google Workspace Drive
* Messaging & Alerts:
  + Discord

**Conclusion**

The selected hardware and software ensure the Workers' Compensation Digital Case Management System is fast, secure, scalable, and compliant. The high-performance infrastructure, enterprise-grade security, and cloud integration will help in:

* Reducing claim processing time
* Improving user experience
* Ensuring robust data security
* Maintaining cost-efficiency



# VI. User Interface

# User personas

| **Front Desk: wants efficient case handling and communication** | |
| --- | --- |
| "I need a system that helps me quickly log, track, and update workers' compensation cases." | |
| **Knowledge:** | Understands basic case intake, documentation, and appointment scheduling but not technical details of case management systems. |
| **Tasks**: | Receives and logs new cases, updates case statuses, schedules medical evaluations, and communicates with employees and medical providers. |
| **Interests:** | Wants a user-friendly interface to efficiently manage case workflows, reduce errors, and improve communication. |
| **Characteristics:** | Organized, detail-oriented, and focused on multitasking. Prefers clear instructions and easy-to-use tools. |

| **Employee: wants smooth claim processing and case updates** | |
| --- | --- |
| "I just want my claim processed quickly and know what’s happening at each step." | |
| **Knowledge:** | Limited understanding of workers' compensation processes but expects clear guidance and updates. |
| **Tasks**: | Submits claims, provides necessary documents, attends medical evaluations, and follows up on case status. |
| **Interests:** | I Want timely updates, clear instructions, and an easy way to submit and track claims. |
| **Characteristics:** | Practical, concerned about timelines, and values transparency in case processing. |

| **Medical Provider: wants clear case details and smooth reporting** | |
| --- | --- |
| "I need accurate case information to provide the right evaluations and submit reports efficiently." | |
| **Knowledge:** | Understands medical evaluation processes and required documentation for workers' compensation cases. |
| **Tasks**: | Conducts evaluations, submits medical reports, and communicates findings to the case management system. |
| **Interests:** | Wants streamlined access to case details and a simple way to submit evaluations and reports. |
| **Characteristics:** | Precise, time-conscious, and values accurate communication to avoid case delays. |

# 

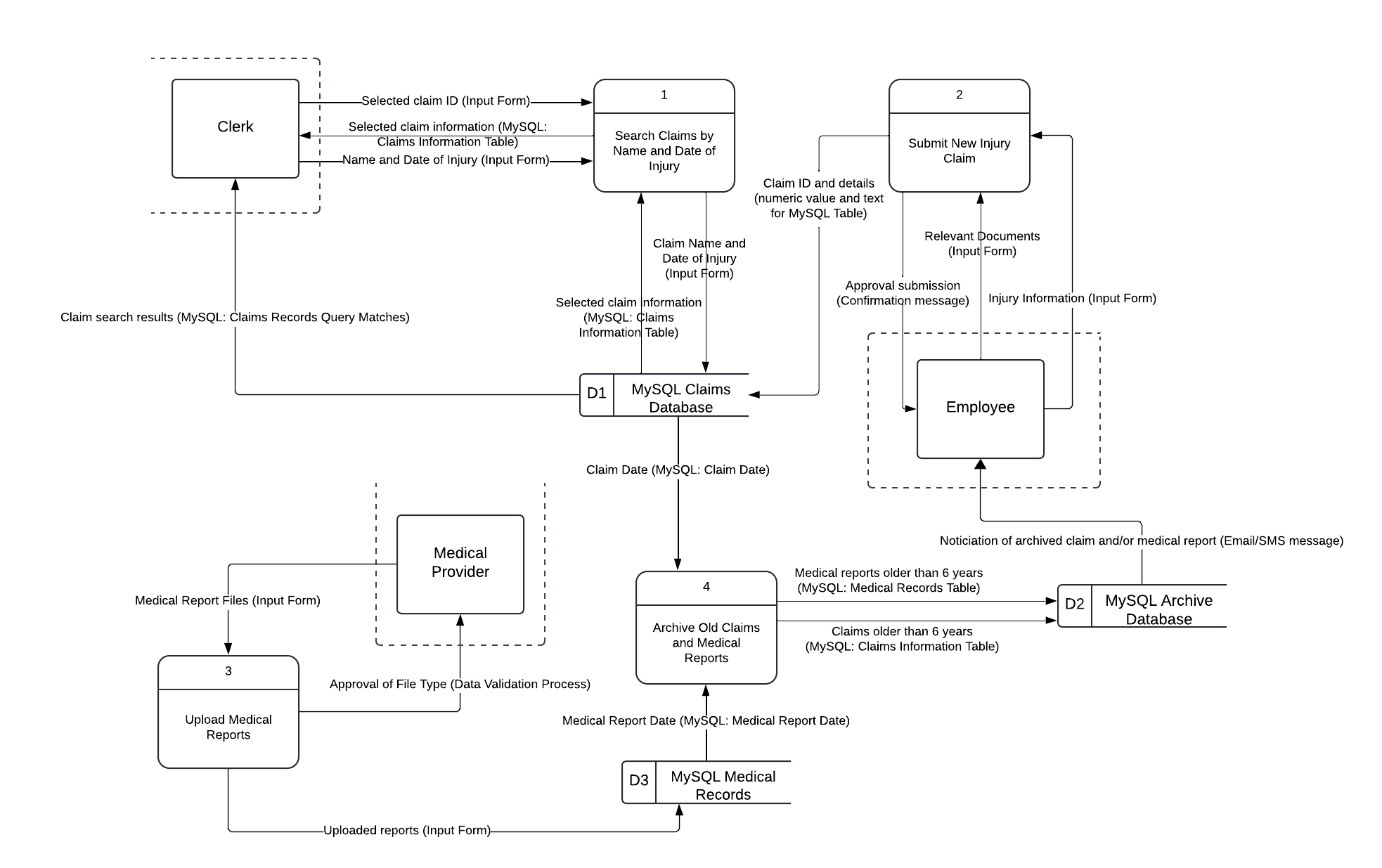
# Use scenarios

| **Use Scenario: Clerk Searching for a Claim** |
| --- |
| The claim status is retrieved in seconds, eliminating the need to manually search through paper files. |
| 1. A worker calls to check the status of their injury claim. 2. The clerk logs into the claims system. 3. The worker provides their name and date of injury. 4. The clerk enters the name and date of injury into the search function. 5. The system queries the Claims Database and retrieves the matching claim. 6. The clerk reviews the claim details and provides the caller with the status update in real time. 7. If necessary, the clerk downloads relevant documents and sends them to the caller. |

| **Use Scenario: Employee Submitting a New Injury Claim** |
| --- |
| The employee successfully submits a claim without needing to mail physical documents or visit the office. |
| 1. The employee logs into the secure claims website. 2. They fill out an online injury claim form, including personal details, employer information, and injury description. 3. The employee uploads relevant medical documents (e.g., doctor’s reports, hospital bills). 4. The system verifies the submission and saves the claim in the Claims Database. 5. The system generates a Claim ID and confirmation receipt. 6. The employee receives an email confirmation of the submission. |

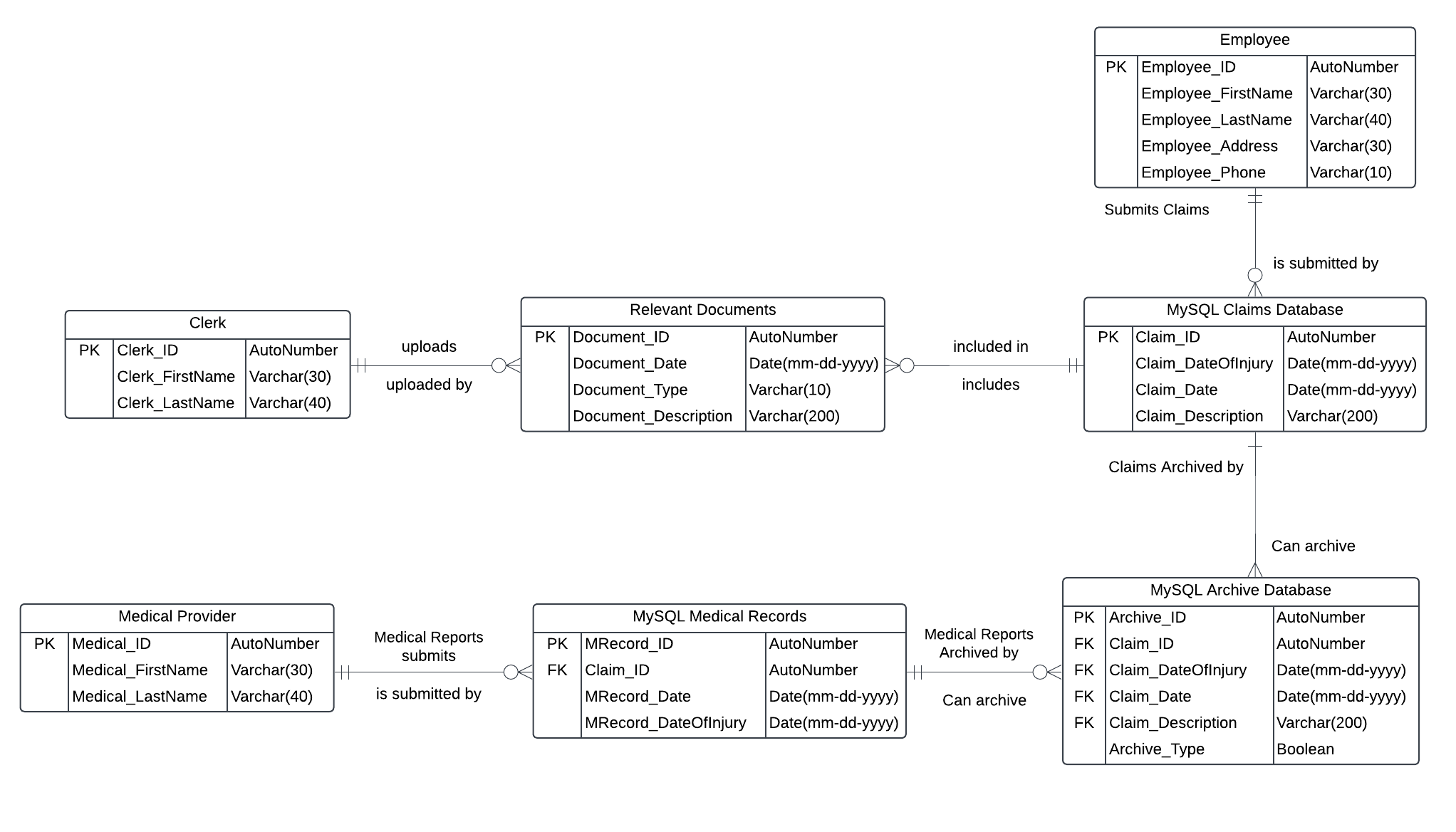
| **Use Scenario: Medical Provider Uploading a Medical Report** |
| --- |
| The medical report is instantly stored and accessible to clerks and claim reviewers. |
| 1. The medical provider logs into the system. 2. They select the option to Upload Medical Reports. 3. The provider enters the worker’s Claim ID or searches by name and injury date. 4. The provider uploads a digital report (PDF or faxed file). 5. The system checks for file type approval. 6. Once approved, the system stores the report in the Medical Reports Database. 7. The system notifies the clerk that a new medical report has been added. |

# VII. Physical Process Model



# 

# VIII. Physical Data Model



# IX. Test plan

## **Unit Tests**

Test Plan

Program ID: ClaimLookup Version Number: 1.0

Tester: Amani Date Designed: 4/12 Date Conducted:4/13

Results: Passed Open items: None

Test ID:U001 Requirement addressed: Search by Claim number or name

Objective: Ensure claim records can be retrieved correctly when a valid claim number or name is entered.

Test Cases Data Entered Result

1. Joe Highsmith Claim

2. 0424 Claim

3. Null Invalid entry

4. Susan C Hall Invalid entry

Script

1. Open the claim lookup module.

2. Enter claim number or name as specified.

3. Click "Search".

4. Verify result or error message.

Expected Results/Notes

Correct record is returned or appropriate message shown.

Actual Results/Notes

All test results matched expectations.

Test Plan

Program ID: MedicalRec Version Number: 1.0

Tester: Marina Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: UT-002 Requirement addressed: Save uploaded medical report

Objective: Verify that uploaded PDF files are saved and assigned to the correct claim.

Test Cases Data Entered Result

1. Report0867.pdf File saved to claim

2. Report0424.pdf File saved to claim

3. Injurydetails.txt Error: Invalid file

4. BenMedreport.pdf File saved to claim

Script

1. Login as admin.

2. Navigate to upload section.

3. Select file and choose claim number.

4. Upload and verify result.

Expected Results/Notes

Files should save correctly, invalid files rejected.

Actual Results/Notes  
Functionality passed for all valid inputs.

Test Plan

Program ID: UserEntry Version Number: 1.0

Tester: Angela Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: UT-003 Requirement addressed: Validate user authentication

Objective: Ensure login system correctly handles valid and invalid login attempts.

Test Cases Data Entered Result

1. Username: clerk01 Password:correct123 Login successful

2. Username: clerk01 Password: wrongpass Login Failed

3. Username: Brett23 Password: anything Login failed

4. Username: admin Password:admin@123 Login successful

Script

1. Open login page.

2. Enter credentials.

3. Click login and record the outcome.

Expected Results/Notes

Only correct credentials should succeed.

Actual Results/Notes

Login validation working as intended.

Test Plan

Program ID:ArchiveClaim Version Number: 1.0

Tester: Sahil Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: UT-004 Requirement addressed: Archive claim with uploaded document

Objective:

Ensure claims and relevant documents that are older than 6 years are archived

Test Cases Data Entered Result

1. Claim: 0867 Date of claim: 06-14-2015 Claim 0867 Archived

2. Claim: 0424 Date of claim: 09-09-2016 Claim 0424 Archived

3. Claim: 0914 Date of claim: 03-22-2022 Claim 0915 not Archived

4. Claim: 0895 Date of claim: 04-05-2019 Claim 0915 Archived

Script

1. Open the Archive database

2. Select claim to be assessed

3. Verify date of claim is 6 years and older

4. Claims 6 years and older placed in archive database

5. Claims not older than 6 years sent back to claims database

Expected Results/Notes

Claims should only be archived when confirmed to be 6 years and older.

Actual Results/Notes

Working as expected.

Integration Tests

## **Integration Tests**

Test Plan

Program ID: SearchandView Version Number: 1.0

Tester: Amani Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: IT-001 Requirement addressed: Search and view claim

Objective: Verify database and UI integration retrieves full claim details by name.

Test Cases Data Entered Result

1. Name: Joe Highsmith DOI: 06/14/2015 Claim 0867 Displayed

2. Name: Susan C Hall DOI: 09/09/2016 Claim 0242 Displayed

3. Name: Jeremy Doyle DOI: 03/22/2022 Claim 0914 displayed

4. Name: Bob DOI: 01/01/2023 Claim not found

Script

1. Log into the system as a clerk.

2. Go to the 'Search Claim' screen.

3. Enter a valid claimant's name and date of injury.

4. Click 'Search'.

5. Confirm that the system retrieves the correct record from the database.

Expected Results/Notes

The correct claim record associated with the input name and date should be retrieved and displayed.

Actual Results/Notes

Record retrieved as expected. No errors or mismatches found.

Test Plan

Program ID: MedSearchandView Version Number :1.0

Tester: Marina Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: IT-002 Requirement addressed: Upload and Search Medical Files

Objective: Ensure newly uploaded files are searchable and viewable by claim.

Test Cases Data Entered Result

1. armscan.pdf File listed under claim

2. blank.pdf Error: invalid file

3. finalReport.pdf File listed under claim

4. DriversLicence.pdf no file found

Script

1. Log into the system as a clerk.

2. Search for an existing claimant's case.

3. Upload a new PDF medical report.

4. Check the case file for the uploaded document.

Expected Results/Notes

The uploaded file should appear in the claimant's case file immediately.

Actual Results/Notes

File successfully linked to case. Visible on claimant record page.

Test Plan

Program ID: FaxIntegration Version Number :1.0

Tester: Angela Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: IT-003 Requirement addressed: Fax to Digital Integration

Objective: Verify faxed documents are converted and attached to claim.

Test Cases Data Entered Result

1. testfax1 PDF saved under claim

2. corruptedfax Error: unreadable file

3. blankfax Error: no data received

4. testfax2 File processed successfully

Script

1. Send a test fax document to the system.

2. Wait for processing.

3. Search for the related case by name.

4. Confirm the faxed document appears in the document list.

Expected Results/Notes

Document appears under the claimant's record, labeled as a faxed document.

Actual Results/Notes

Faxed file processed correctly. Accessible in the document section.

Test Plan

Program ID: UserPerm Version Number :1.0

Tester: Sahil Date Designed: 04/12 Date Conducted: 4/13

Results:Passed Open items: None

Test ID: IT-004 Requirement addressed: User permissions and file access

Objective: Test different user roles and their document access rights.

Test Cases Data Entered Result

1. Clerk: C002 Access granted

2. Intern: I023 Access denied

3. Admin: A01 Access granted

4. Admin: A02 Access granted

Script:

1. Log in with a Clerk or Admin account.

2. Attempt to open both assigned and unassigned case files.

3. Monitor access control behavior.

Expected Results/Notes:

Access granted only to authorized records. Unauthorized attempts are blocked.

Actual Results/Notes:

Access control functioning as expected.

## **System Tests**

Test Plan

Program ID: EndToEndSearch Version Number 1.0

Tester: Amani Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: ST-001 Requirement addressed: End-to-end claim search

Objective: Simulate full user interaction to search and view claim from UI.

Test Cases Data Entered Result

1. Search: Joe Highsmith, 6/14/2015 Claim 0867 displayed

2. Search: Bob, 5/5/2027 no result

3. click on result full claim shown

4. refresh and repeat search Same result appears

Script

1. Navigate to the main claim search page.

2. Enter appropriate details in the name and date fields.

3. Press "Search." and click on the claim result .

4. Review the displayed claim details.

5. Refresh the page.

6. Repeat the search with same inputs.

7. Enter an incorrect date and search again..

Expected Results/Notes

Clicking on the result should show full claim details.

Same result should appear on re-search.

No results should be returned if inputs are invalid.

Actual Results/Notes

System accurately retrieved claim.

Full details were shown on click.

Refresh and repeated search worked as expected.

Invalid date returned no results.

Test Plan

Program ID: FileUp Version Number 1.0

Tester: Marina Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: ST-002 Requirement addressed: Upload via portal

Objective: Verify web form to upload files, system stores and lists properly

Test Cases Data Entered Result

1 docA.pd Upload successful, visible

2. docB.jpg Upload rejected

3. docC.pdf Upload successful, visible

4. None Error: file required

Script

1. Log into portal as an authenticated user.

2. Navigate to “Upload Medical Reports.”

3. Select and upload a document

Expected Results/Notes

File is accepted and listed in the claim, File format is rejected with error.

Actual Results/Notes

All test results matched expectations.

Test Plan

Program ID: LoginAndSession Version Number: 1.0

Tester: Angela Date Designed: 04/12 Date Conducted: 4/13

Results:Passed Open items:

Test ID: ST-003 Requirement addressed: Secure login and session flow

Objective: Verify that login flow, session handling, and logout behave securely.

Test Cases Data Entered Result

1. Login clerk, upload File uploaded

2. Wait 30 min, resume Session timeout

3. Admin login, Delete File deleted

4. Guest attempts to view file Access denied

Script

1. Log in as clerk and upload a valid document.
2. Wait 30 minutes without activity, then try a new upload.
3. Log in as admin and delete a test file.
4. Attempt to access the same file as a guest

Expected Results/Notes

Clerk and Admin login and session flows as needed, guests shouldn’t have access

Actual Results/Notes

All test results matched expectations.

Test Plan

Program ID: MultiLogin Version Number: 1.0

Tester: Sahil Date Designed: 04/12 Date Conducted: 4/13

Results: Passed Open items: None

Test ID: ST-004 Requirement addressed: Performance under load

Objective: Check system response with multiple users simultaneously searching

Test Cases Data Entered Result

1. 5 Users All results returned fast

2. 25 Users Slight lag, all passed

3. 50 Users Moderate lag, no crash

4. 100 users System warning:slowdown

Script

1. Simulate 5 users logging in and searching simultaneously.

2. Simulate 25 users performing the same action.

3. Increase to 50 users and repeat.

4. Push to 100 concurrent users to observe system behavior.

Expected Results/Notes

System should return results for all user levels.

Performance may degrade slightly at 50+ users but must not crash.

At 100 users, system should warn of potential delays.

Actual Results/Notes

Up to 25 users: No issues.

At 50: Some noticeable lag, all results still returned.

At 100: System issued performance warning; no failure or crash.

# 

# X. Migration Plan

## **Preparing the Business**

***Conversion Strategy***

For the system migration, a Phased Conversion Strategy will be implemented. This approach will be used to gradually roll out the new system, beginning with one department or region at a time, allowing teams to adapt and identify any issues before full-scale implementation.

* Phase 1: Pilot program with a selected department to assess initial system performance.
* Phase 2: Gradual rollout across departments, incorporating feedback from Phase 1.
* Phase 3: Full deployment across the organization, with ongoing monitoring and optimization.

***Business Contingency Plan***

Even with careful analysis, detailed design, and structured project management, unexpected issues can occur during system implementation. As such, a robust business contingency plan is essential to ensure the organization can continue operations if the new system encounters problems.

The primary objective is to prevent small technical issues from escalating into major business disruptions during and immediately following the system migration.

**A. Identify Potential System Failures and Worst-Case Scenarios**

**Scenario 1:** New System Becomes Unusable During Rollout

Impact: Business processes halt; employees are unable to access key tools and data.  
Response Plan:

* Maintain the old system using parallel conversion during the initial phase of rollout.
* If parallel is not feasible for some modules, revert to manual processes temporarily using pre-prepared paper forms and offline data entry.

**Scenario 2:** Data Migration Fails or Corrupts Data  
Impact: Critical business records lost or altered; compliance and reporting issues arise.  
Response Plan:

* Perform data migration in staged batches with validation after each batch.
* Keep daily backups of both source data and migrated data for quick restoration.
* Involve data stewards to audit and verify migrated data on a rolling basis.

**Scenario 3:** Key Software Functions Are Buggy or Inoperable  
Impact: Users are unable to complete essential tasks like submitting claims, generating reports, or accessing client records.  
Response Plan:

* Prioritize a bug triage process with a technical response team to implement patches within 24–48 hours.
* Maintain a list of manual workarounds (e.g., alternative ways to process forms or generate reports) documented and distributed to users in advance.

**Scenario 4:** Hardware or Network Failure  
Impact: No access to the new system due to server crash or connectivity issues.

Response Plan:

* Use redundant servers and establish cloud-based failover systems.
* Ensure backup generators and uninterruptible power supplies (UPS) are available at key locations.
* Maintain a paper-based backup plan for essential operations like claim intake and employee scheduling.

**B. Stakeholder Involvement and Communication**

A Crisis Management Team has been established, consisting of:

* Sarah Kim (IT Director) – Leads technical recovery and system rollback efforts.
* Jason Patel (Operations Manager) – Coordinates business continuity and manual processes.
* Maria Gonzalez (Claims Supervisor) – Manages user communication and help desk support.
* Emily Chen (HR/Training) – Oversees user training and ensures policy compliance.

Pre-written communication templates will be used for system outage alerts, and a dedicated help desk extension will be active during migration. Department leads will conduct drills and maintain printed fallback instructions in Contingency Binders to ensure business users are prepared for system downtime.

**C. Parallel Systems and Rollback Plan**

Parallel Conversion: For at least the first two weeks post-deployment, the legacy system will run in parallel with the new one for all core functions, including claims intake, case updates or medical report tracking  
Rollback Strategy: If critical failures persist beyond a 72-hour period the system will be rolled back to the legacy platform using the most recent synchronized backup and all data entered into the new system post-failure will be documented and manually re-entered into the legacy system to avoid loss.

**D. Budget and Time Buffer for Emergencies**

Contingency Budget: Allocate $20,000–$25,000 specifically for emergency fixes, extended support hours, or additional consulting resources.  
Time Buffer: Add a two-week buffer to the final go-live timeline, specifically reserved for issue resolution and rollback if needed.

**E. Testing the Contingency Plan**

Run a simulation of a system failure one week prior to go-live to ensure all fallback processes, contact lists, and manual workflows are functioning as expected and collect feedback from the crisis management team and make final adjustments.

## 

## **Preparing the Technology**

***Install Hardware***

The installation process will begin with provisioning all necessary hardware components, including servers, workstations, network devices, and physical security systems. Servers will be set up in a hybrid configuration to support both on-premises and cloud functions. Workstations for clerks and administrators will be installed with dual monitors and ergonomic peripherals to support productivity. Network infrastructure components—such as routers, switches, firewalls, and wireless access points—will be deployed to ensure secure and reliable connectivity. All equipment will be tested for performance and compliance with system requirements.

***Install Software***

Once hardware is in place, system software will be installed and configured. This includes server and workstation operating systems, database management systems, backend and frontend applications, security software, and document management tools. Authentication services, encryption protocols, and role-based access controls will be implemented to protect sensitive data. Compliance tools and communication platforms will also be set up to support operational needs. The installation phase concludes with thorough testing to validate system functionality and ensure readiness for data migration.

***Convert Data***

Data Migration Plan:

* Data Audit: Assess existing data for relevance and cleanliness before migration.
* Data Preparation: Cleanse and standardize data (e.g., remove duplicates, correct inconsistencies).
* Pilot Migration: Migrate a small subset of data (e.g., a specific department’s data) to verify the migration process.
* Full Migration: Once the pilot is validated, migrate the full dataset in stages.
* Validation: Perform a post-migration audit to ensure data accuracy and completeness.

## 

## **Preparing the People**

***Revise Management Policies***

Policy Updates:

* System Access: Revise access policies to include user roles and permissions specific to the new system.
* Data Security: Update data protection and privacy policies to comply with new system standards.
* Employee Incentives: Link system adoption to employee performance metrics, providing rewards for active engagement with the new system.

***Assess Costs and Benefits***

**Cost Analysis:**

The cost analysis includes both initial and ongoing costs, as well as the expected benefits from system implementation.

Initial Costs:

* Software Purchase: $100,000
* Hardware Upgrade: $150,000
* Training: $30,000
* Contingency (Risk Management): $20,000
* Total Initial Costs: $300,000

Ongoing Costs:

* Maintenance and Support: $20,000 per year
* Licensing Fees: $15,000 per year
* Training & Development: $20,000 annually for staff training on system updates and new hires.
* Total Ongoing Costs: $55,000 annually

**Benefit Analysis**

Tangible Benefits:

* Efficiency Gains: 30% reduction in processing time.
* Staff Reduction: Fewer employees needed, resulting in labor savings.
* Reduced Paper Storage Costs: $5,000 saved annually.
* Scalability: Supports more clients without physical space limitations.
* Improved Accuracy: $10,000 saved annually from fewer errors.

Intangible Benefits:

* Improved Customer Service: Faster processing leads to higher satisfaction.
* Better Data Security: Enhanced features reduce data breach risks.
* Accessibility and Mobility: Allows remote access and faster response times.
* Enhanced Reputation: Demonstrates innovation and modern practices.

Economic Feasibility:

* Total Development Costs: $670,000.
* Total Operational Costs: $130,000 over 5 years.
* Total Benefits: $1,400,000 over 5 years.
* Net Cash Flow: $600,000 over 5 years, with ROI of 75% and a payback period of 3.64 years.

***Motivate Adoption***

Engagement Strategy:

* Ready Adopters: Identify key early adopters within departments and use them as champions to encourage their colleagues.
* Reluctant Adopters: Offer incentives like extra training opportunities and career development linked to the adoption of the system.
* Resistant Adopters: Use strong management communication, explaining the long-term benefits and the necessary transition for organizational growth. This may include linking job security or career progression to the system’s successful adoption.

Political Strategy: Engage influential leaders within the organization to advocate for the system’s adoption and emphasize organizational benefits.

# XI. Training Plan

## **A. Who Will Be Trained**

## To ensure the effective use of the new digital case management system, multiple categories of system users will receive role-specific training. These include clerks, medical providers, and internal IT or administrative personnel. Each group has unique responsibilities within the system and will receive customized training modules to address their needs.

### **1. Clerks (Administrative Support Staff)**

#### **a. What Will They Be Trained On?**

Clerks serve as the frontline users responsible for data entry, case file management, and system operation on a daily basis. Their training will focus on:

* System Navigation: Familiarity with the user interface, login procedures, dashboards, and menu structure.
* Data Entry & Uploads: Step-by-step training on uploading injury claims, forms, and documents in different formats (PDFs, images, scanned files).
* Search and Retrieval: Using full-text and filtered search functionalities to locate records by claimant name, address, or date of injury.
* Case Management: Tagging, updating, and organizing case files within an individual’s digital profile.
* System Notifications: Enabling and responding to system-generated alerts, such as document receipt confirmations or claim status updates.
* Security Awareness: Role-based access training to ensure that only authorized personnel access sensitive information.
* Compliance Guidelines: Understanding HIPAA and state-level confidentiality standards while handling medical records.

#### **b. How Will They Be Trained?**

Delivery Method: In-person sessions at the department office supplemented by instructional videos for later review.

Training Materials: Printed user handbooks, interactive slide decks, and live software demonstrations.

Schedule: 3 consecutive days with 2-hour training blocks per day.

Practice Exercises: Hands-on training through sandbox accounts to simulate real-world tasks.

Evaluation: Post-training quiz and supervisor sign-off on demonstrated proficiency.

Follow-Up Support: Weekly office hours and access to an internal helpdesk for 60 days post-training.

### **2. Medical Providers**

### **a. What Will They Be Trained On?**

Medical professionals interact with the system primarily to submit medical records, treatment updates, and diagnostic information. Their training will emphasize:

* System Access and Authentication: Logging in using multi-factor authentication (MFA) and verifying credentials.
* Uploading Medical Records: Step-by-step guide for uploading medical documentation in supported formats (e.g., PDF, .jpg, .png).
* Integration via Secure APIs: For clinics using compatible EHR systems, API-based data submission training will be provided.
* Submission Verification: Checking confirmation receipts and tracking the status of submitted records.
* Privacy and Security Protocols: Understanding data encryption, role-based access restrictions, and digital signatures.
* Troubleshooting Common Errors: What to do if a file fails to upload or if access is denied.

#### **b. How Will They Be Trained?**

Delivery Method: Live virtual webinars and self-paced e-learning modules hosted on the Department’s Learning Portal.

Training Materials: Role-specific digital guides and system walkthrough videos tailored for healthcare providers.

Schedule: 2 days, each with a 1-hour live session followed by optional Q&A.

Flexibility: All sessions recorded and available on-demand for those unable to attend live.

Certification: Providers will receive a “System Access Certification” upon successful completion of training.

### **3. Internal IT & System Administrators**

#### **a. What Will They Be Trained On?**

This group is responsible for maintaining system integrity, managing users, troubleshooting issues, and ensuring long-term sustainability. Their training will cover:

* User and Role Management: Creating user profiles, setting access levels, and managing permissions.
* Database Monitoring: Basic administration of the MySQL/PostgreSQL database, including backup and restoration protocols.
* Audit Trail & Logs: Monitoring user activity logs for security compliance and performance auditing.
* Incident Response Protocols: Procedures for data breaches, system failures, or access issues.
* Backup and Disaster Recovery: Implementing daily encrypted backups and conducting recovery drills.
* Legal Compliance: Understanding HIPAA requirements, system audit obligations, and data lifecycle policies.
* System Integration: Ensuring interoperability with external state systems and healthcare APIs.

#### **b. How Will They Be Trained?**

Delivery Method: On-site technical workshops and live vendor demonstrations.

Training Materials: Admin manuals, database schema documentation, and configuration templates.

Schedule: 3 full days of 3-hour technical sessions.

Hands-On Practice: Access to the staging environment for live configuration and scenario-based exercises.

Ongoing Support: Continuous access to vendor support and monthly check-in meetings with the IT project lead.

## **B. Training Schedule**

The training will be conducted over a three-week period to ensure all stakeholders are adequately prepared before the system goes live.

| User Group | Training Dates | Format | Duration | Location/Platform |
| --- | --- | --- | --- | --- |
| Clerks | May 5–7, 2025 | In-person workshop + videos | 3 days (2 hrs/day) | SD Dept. of Labor HQ |
| Medical Providers | May 8–9, 2025 | Live webinars + e-learning | 2 days (1 hr/day) | Department Learning Portal |
| IT/Admin Staff | May 12–14, 2025 | On-site technical workshop | 3 days (3 hrs/day) | IT Training Lab |
| Refresher for All | May 20, 2025 | Virtual open Q&A session | 1 day (1 hr) | Zoom (invite via email) |

## **C. Training Tools and Support Structure**

Learning Management System (LMS): A central portal will host training materials, videos, and assessments.

Helpdesk: A dedicated support team will be available for 60 days after training via email and phone.

Multilingual Access: All materials will be translated into Spanish and designed to meet accessibility standards.

Feedback and Improvement: Trainees will be asked to complete surveys, and suggestions will be used to enhance future training rounds.

## **D. Evaluation and Certification**

Performance Checklists: Each participant will complete a practical checklist to confirm understanding of core functions.

Knowledge Assessment: A short quiz will be administered post-training.

Certificates of Completion: Issued to all participants who complete training and assessments successfully.

Manager Sign-Off: Supervisors will confirm readiness of their staff through formal evaluation forms.

# XII. System Support and Maintenance Plan

***System Monitoring***  
To ensure optimal performance and reduce security risks, continuous monitoring of system usage is essential. This includes automated error logging to detect and document issues, helping support and maintenance teams analyze root causes and develop appropriate solutions. Security monitoring is also a vital part of this process, logging unauthorized access attempts and other suspicious activities. These logs serve as both a security tool and a record for identifying trends or vulnerabilities that may need to be addressed through system updates.

***User Support***  
After the system is installed, it is turned over to the operations group, which becomes responsible for its day-to-day functioning. A help desk will be available to provide on-demand support to users. Level 1 support staff will handle general user requests, such as login problems, navigation assistance, and simple troubleshooting. Their goal is to resolve at least 80% of user issues on the first contact. More complex issues that cannot be resolved at this level will be escalated to level 2 support staff, who are more specialized and familiar with the specific system. These staff members may have been involved in the testing and implementation phases, allowing them to provide expert guidance and resolve deeper technical issues.

***Maintenance and Patch Management***  
System maintenance is an ongoing process necessary for adapting the system to evolving business needs and ensuring continued stability. Most change requests originate from issue reports, user feedback, or changes in supporting technologies. These requests are evaluated for feasibility and prioritized accordingly. Maintenance includes correcting software bugs, enhancing functionality based on user suggestions, and updating the system to stay compatible with other technologies.  
Scheduled maintenance will occur from 6:00 PM to 12:00 AM on the first Tuesday of each month. This window allows the IT team to perform necessary updates, such as applying patches, testing system performance, or conducting security audits. Any updates deployed during these windows will be documented with release notes detailing the changes and resolutions. For urgent issues, unscheduled updates may also be conducted with advance user notification.

# XIII. Conclusion

In conclusion, our project successfully addressed the pressing need for a modernized, efficient, and reliable claims management system for the South Dakota Department of Labor’s Workers’ Compensation Division. By transitioning from an inefficient physical file system to a streamlined digital solution, the proposed system offers significant improvements in accessibility, processing speed, storage efficiency, and overall user satisfaction.

Throughout this project, we carefully planned and implemented each phase of the system development life cycle, from initial proposal and acquisition strategy to system architecture, user interface design, data modeling, and testing. We also outlined detailed plans for user training, data migration, system maintenance, and ongoing support to ensure a smooth transition and sustainable operation.

The anticipated benefits of this new system extend beyond operational efficiency. It has the potential to reduce overhead costs, minimize human error, and enhance the experience for both staff and claimants. Our comprehensive design ensures that the system can adapt to future demands while maintaining security and integrity of sensitive information.

Ultimately, this project demonstrates how thoughtful system design and implementation can solve real-world administrative challenges and deliver lasting organizational value. We are confident that, upon deployment, this digital claims management system will positively transform the way the Workers’ Compensation Division operates.

# References

BItTitan. (2025, April 8). *Migration Planning & Strategy Guide*. Bittitan help center. <https://help.bittitan.com/hc/en-us/articles/360044417394-Migration-Planning-Strategy-Guide>

Dennis, A., Wixom, B. H., & Roth, R. M. (2022). Systems analysis and design. John Wiley & Sons, Inc.

Dam, R. F. and Teo, Y. S. (2024, February 8). Personas – A Simple Introduction. Interaction Design Foundation - IxDF. <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>

John. (2020, November 24). *Network Firewall vs network switch VS network router: FS Community*. Knowledge. <https://community.fs.com/article/network-switch-router-firewall-why-need-all-three.html>

Katz, T., Wheatcraft, L., & Ryan, M. (2024, November 25). System requirements definition. Guide to the System Engineering Body of Knowledge. <https://sebokwiki.org/wiki/System_Requirements_Definition>

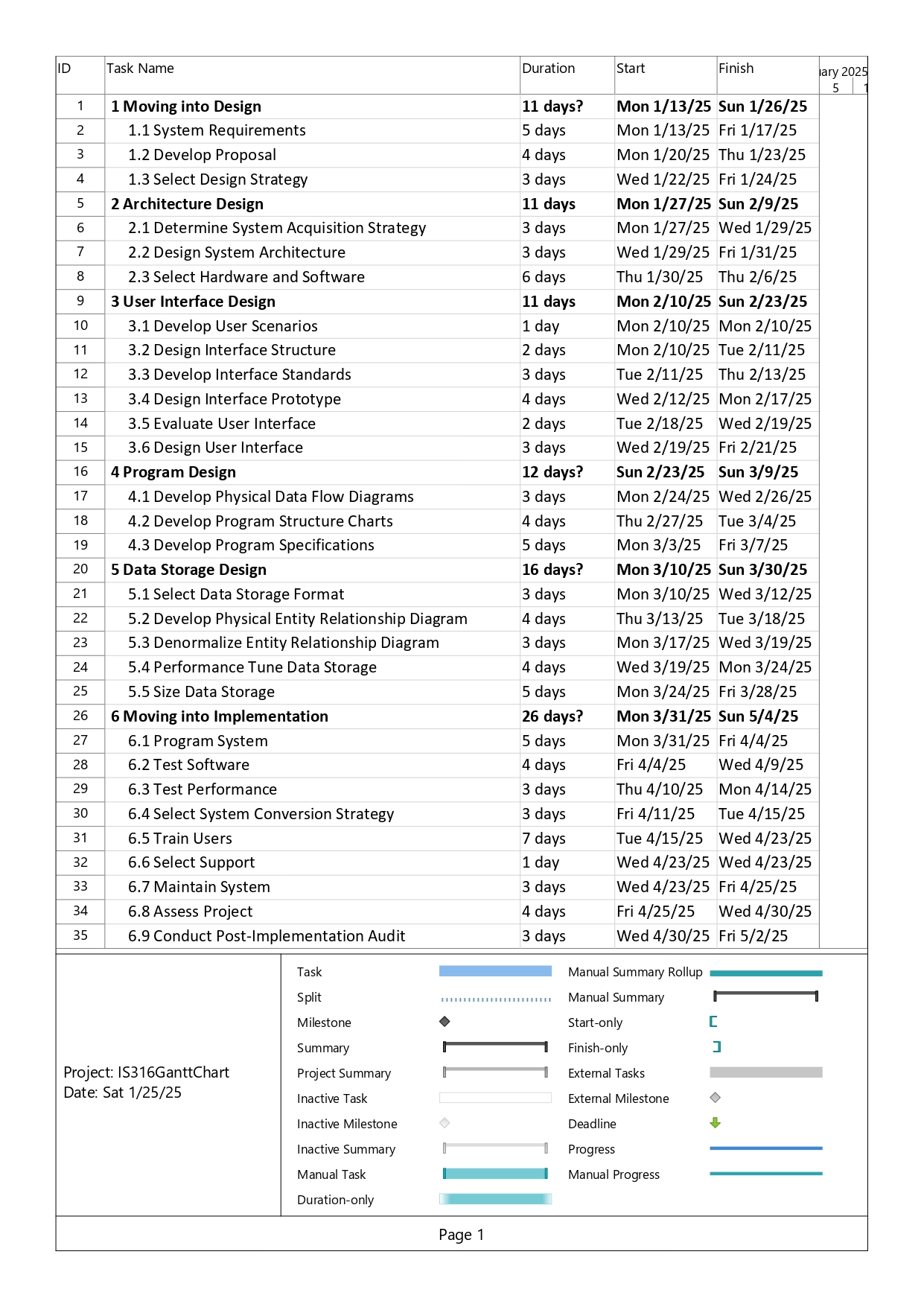
Lucidchart. (n.d.). *Logical vs. physical data flow diagram*. <https://www.lucidchart.com/pages/data-flow-diagram/logical-vs-physical-data-flow-diagram#:~:text=A%20current%20physical%20DFD%20depicts,code%20that%20fuels%20the%20requirements>.

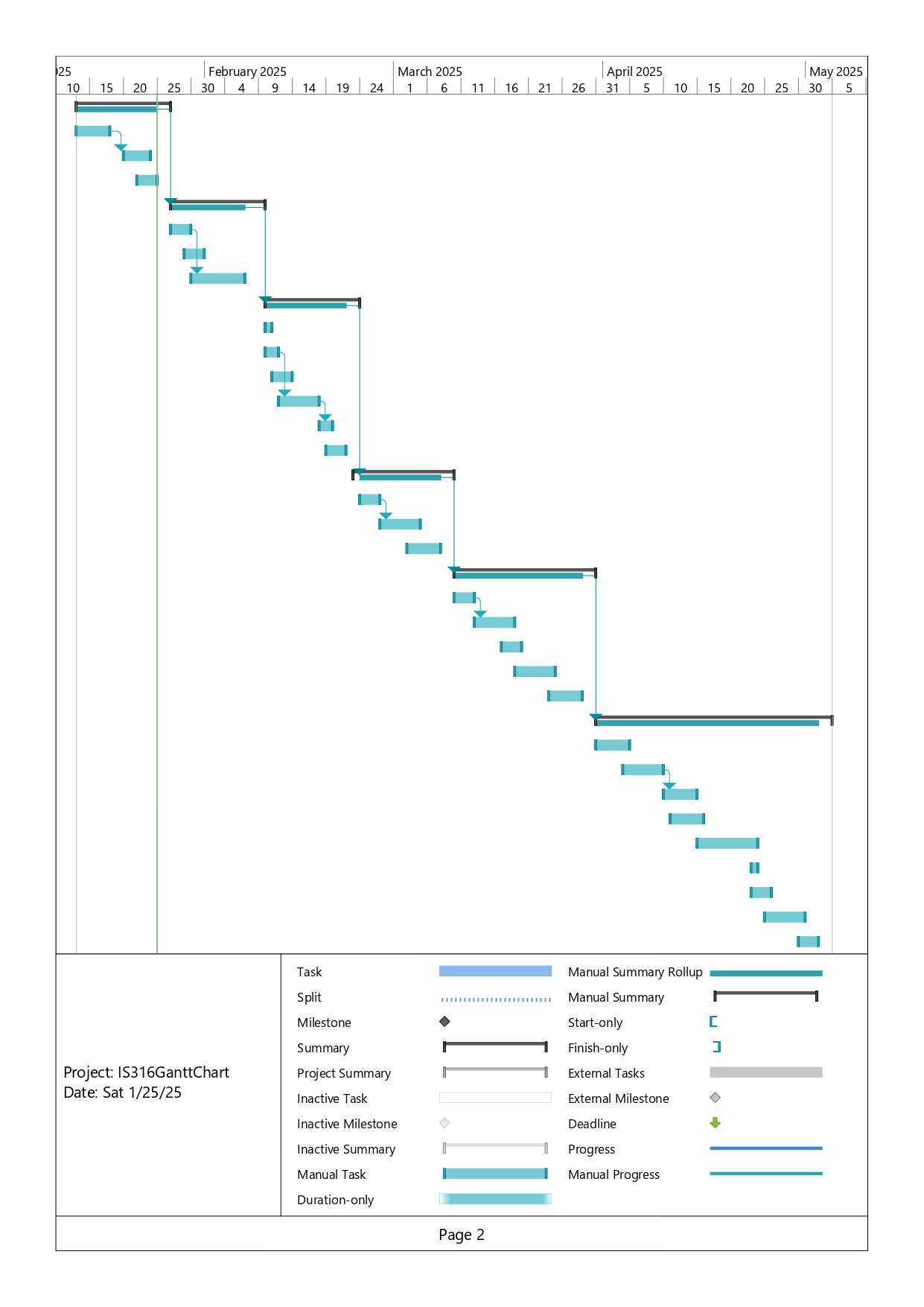
Singal, S. K., Goel, V., Nautiyal, H., & Papantonis, D. E. (2023). Economic feasibility. Economic Feasibility - an overview | ScienceDirect Topics. <https://www.sciencedirect.com/topics/engineering/economic-feasibility>

Suszterova, Sandra. (March 7, 2024). *Physical Data Model vs. Logical Data Model*. GoodData. <https://www.gooddata.com/blog/physical-vs-logical-data-model/>

Watson, R. (2013). Information Systems. Retrieved from <http://solr.bccampus.ca:8001/bcc/file/5f0cf1f4-2c31-4424-804e-e8d57075a5fe/1/Information%20Systems.pdf>

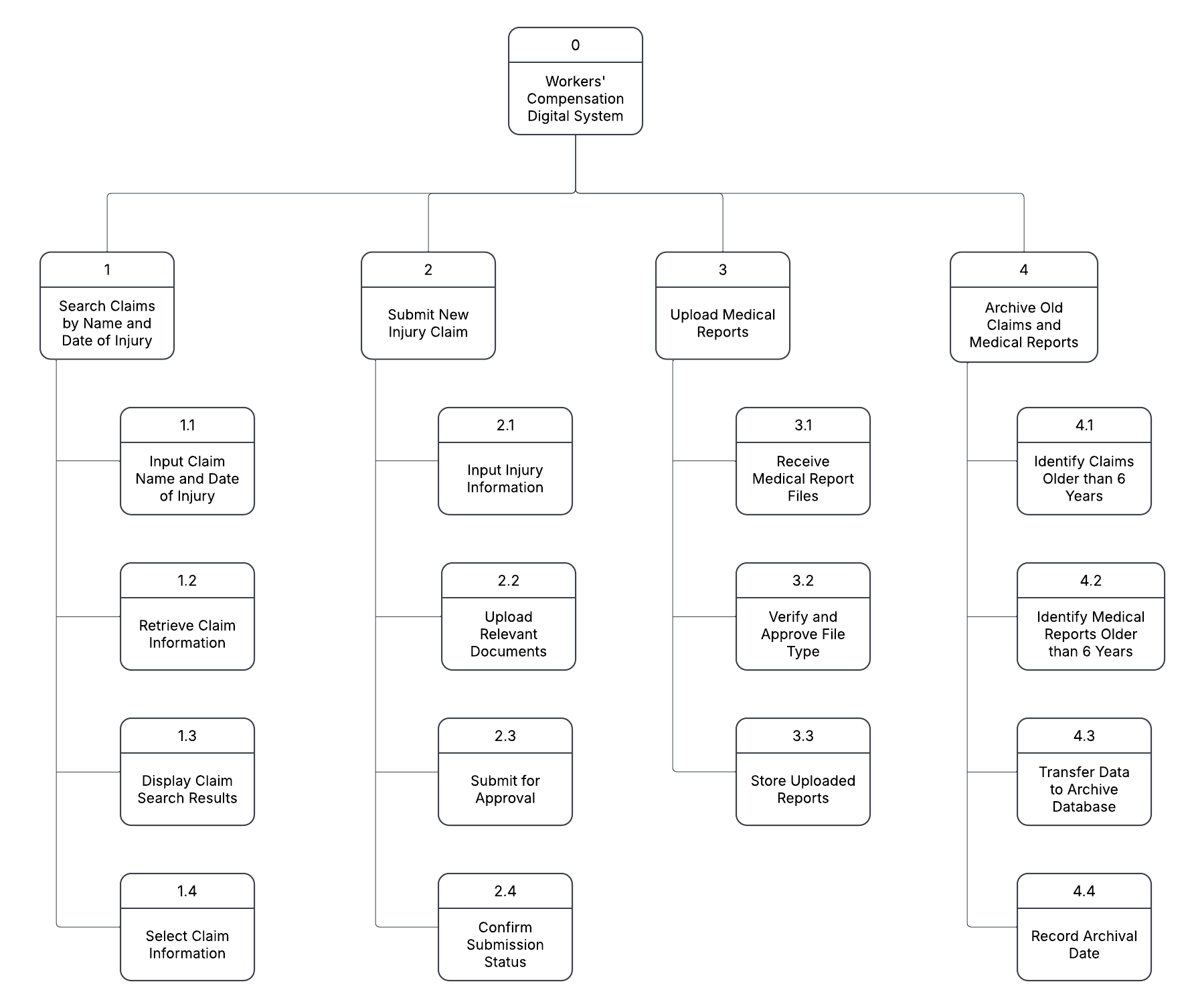
# Appendix A - Project Gantt Chart

******

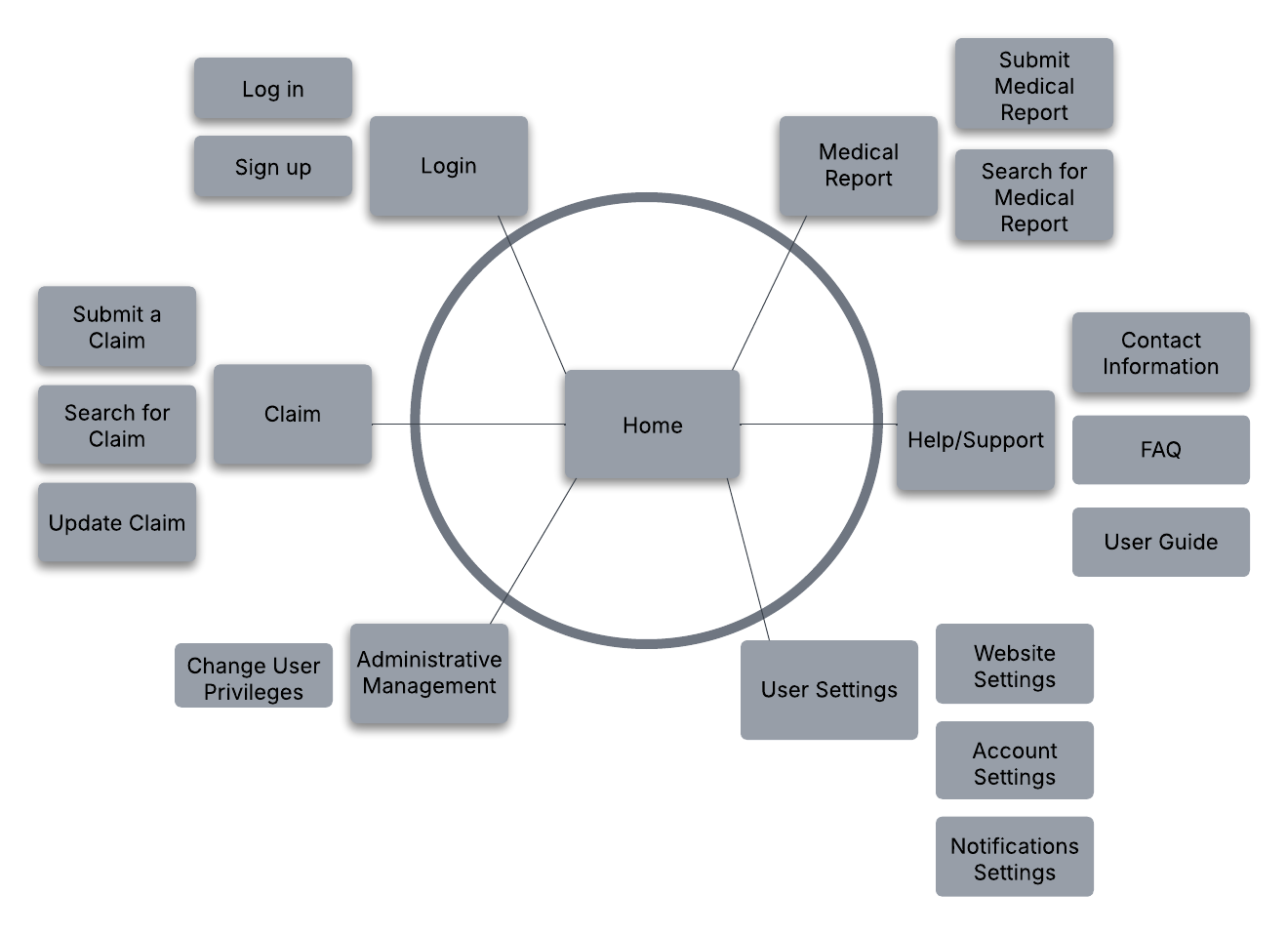
******

# Appendix B - User Interface Design

# Interface structure diagram



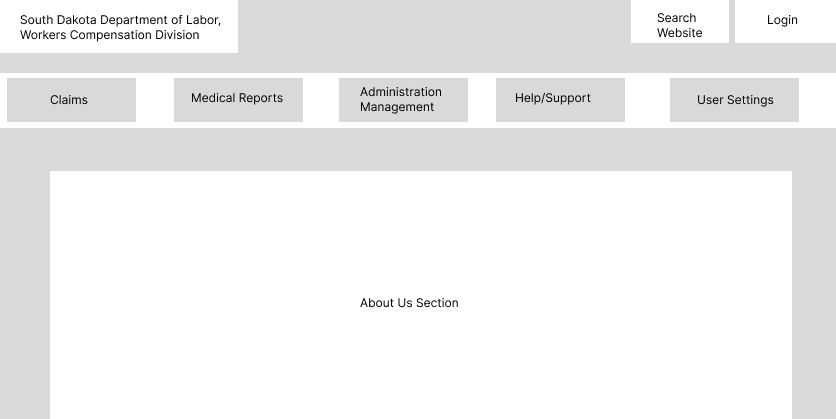
**Site Map**



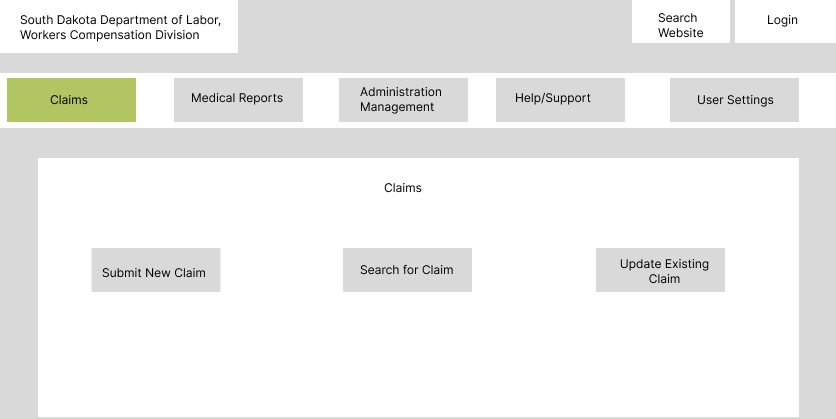
# 

# Monitor screen layouts

**Home Page**

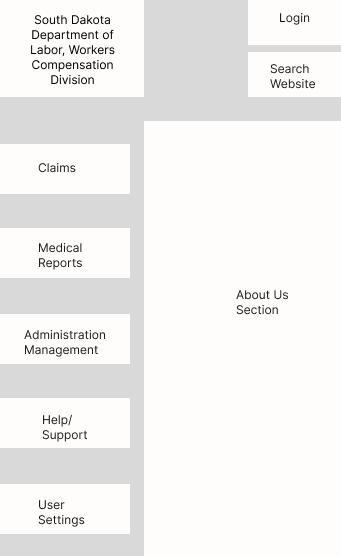
****

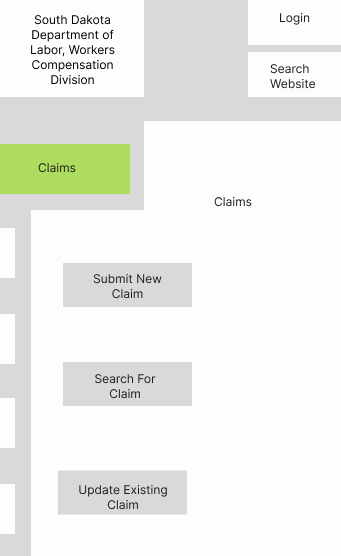
**Claims Page**



# Mobile device screen layouts

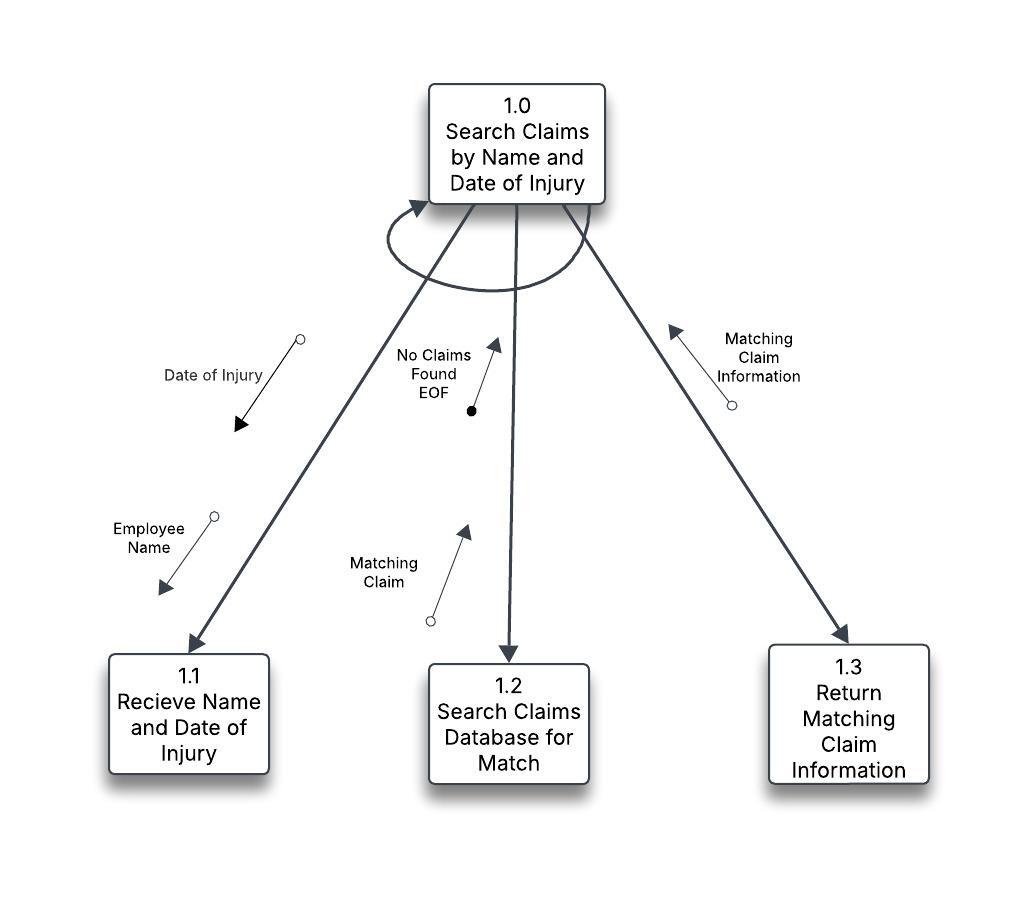
**Home Page**

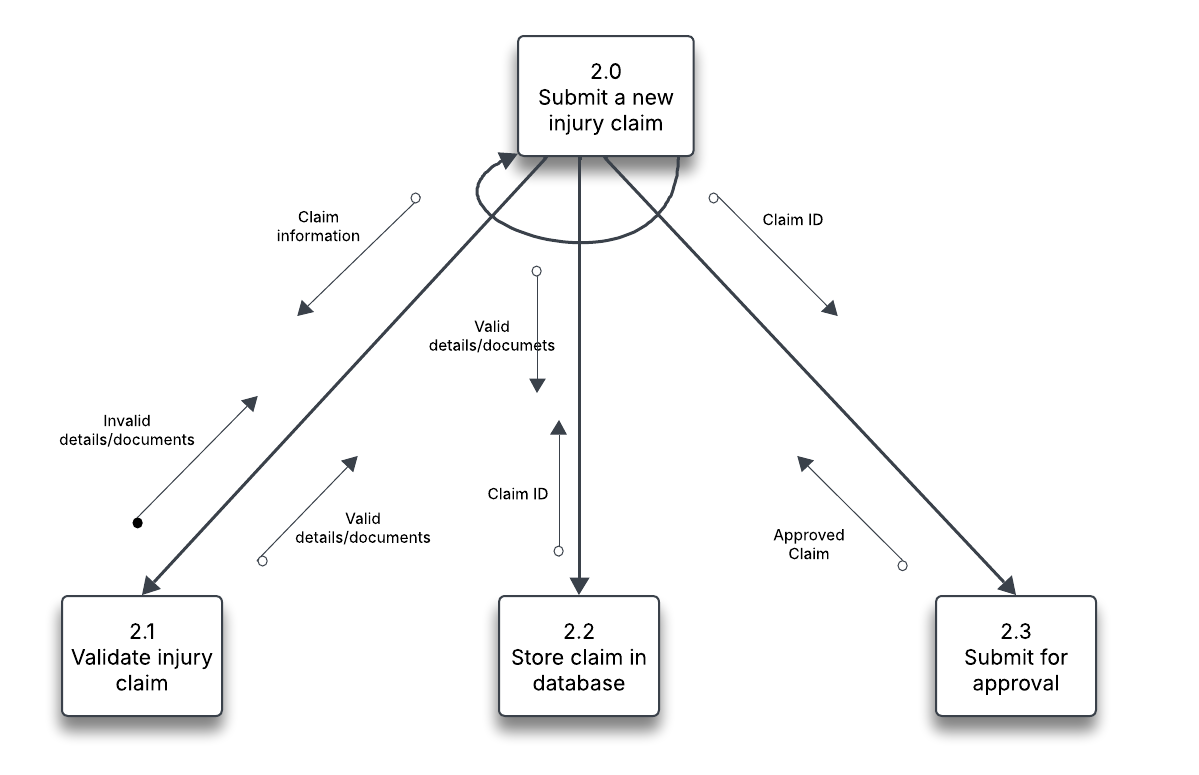
****

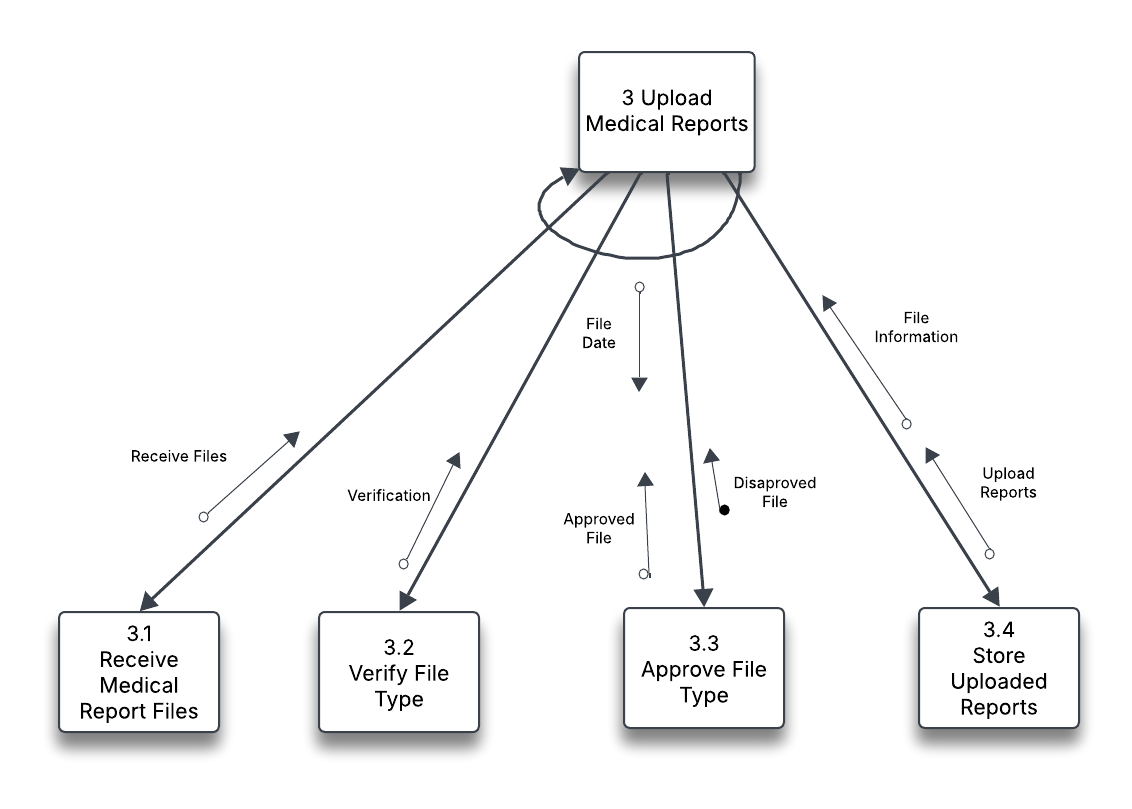
**Claims Page**

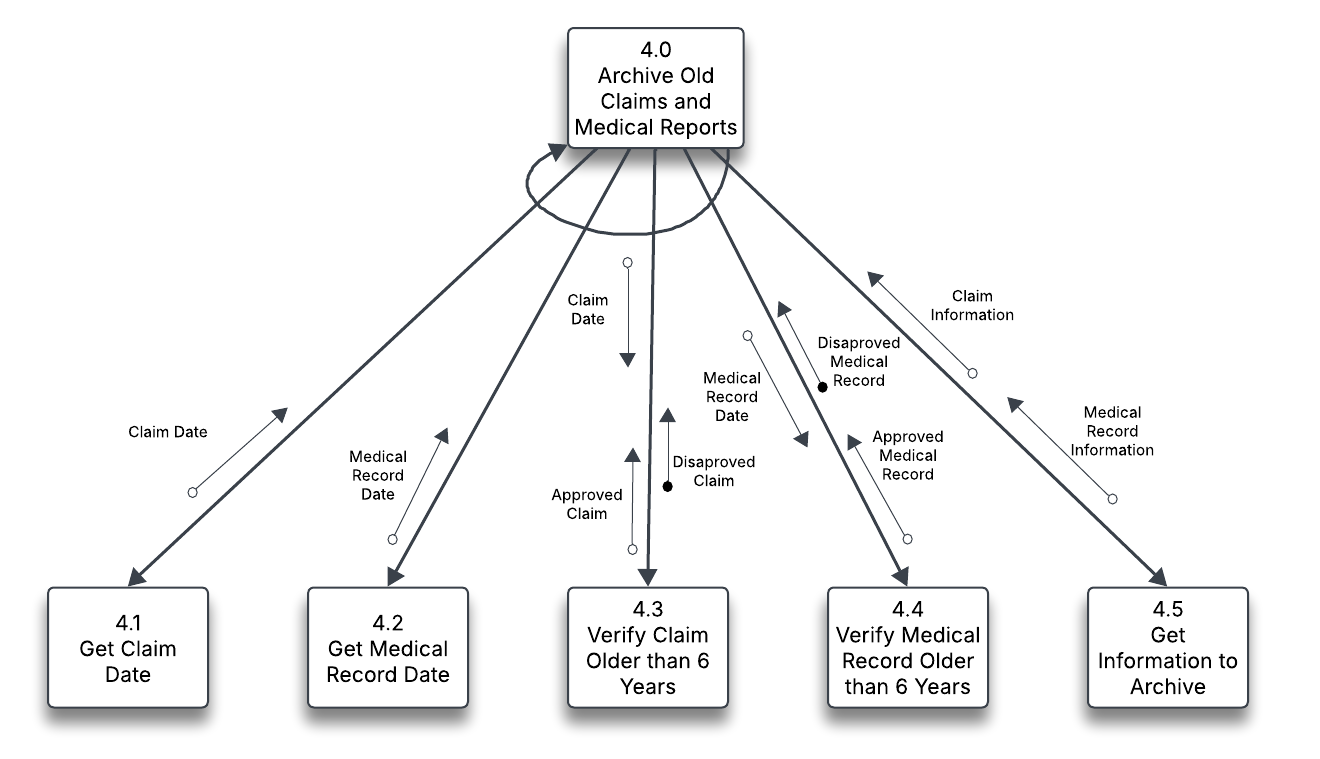
# 

# Appendix C - Structure Charts









# Appendix D - Program Specifications

**Program Specification Form**

**Program Specification No.:** 1  **System:** Claims System for South Dakota Department of Labor, Worker’s Compensation Division

**Module**

**Name:** Search Claims by Name and Date of Injury

**Purpose:** To search and locate claims by name and date of injury.

**Programmer:** Angela

**Date due:** 3/9/25

**Language:** Python

**Event(s)**

Occurs when a clerk fills out the search criteria in the “search claims” section of the system.

| **Input Name** | **Type** | **Provided by** | **Notes** |
| --- | --- | --- | --- |
| Date of Injury | Date (mm-dd-yyyy) | Clerk | Should be listed in the claim information. |
| Employee First Name | String up to 30 characters | Clerk | Retrieved to search for matching claim(s). |
| Employee Last Name | String up to 30 characters | Clerk | Retrieved to search for matching claim(s). |

| **Output Name** | **Type** | **Used by** | **Notes** |
| --- | --- | --- | --- |
| Matching Claim(s) | Record | D1: Claims Database | Returns claim(s) that match the given criteria. |
| Matching Claim(s) Information | Record | D1: Claims Database | Returns claim detail(s) that match the given criteria. |

**Pseudocode / Decision Tree / Decision Matrix**

Search\_Claims\_Name\_and\_DOI(First Name, Last Name, Date of Injury)

FOR (All Claims in Database, iterate by 1)

CurrentClaim = the current index

IF (First Name == CurrentClaim.fn AND Last Name == CurrentClaim.ln AND

DOI == CurrentClaim.DOI)

Return Matching Claim AND Matching Claim Information

ELSE

No Claims Found, End of File

**Other notes**

The first and last name inputs are not case sensitive, but the date must be in the correctly indicated format, otherwise the search will not produce reliable results.

**Program Specification Form**

**Program Specification No.:** 2  **System:** Claims System for South Dakota Department of Labor, Worker’s Compensation Division

**Module**

**Name:** Submit New Injury Claim

**Purpose:** To allow employees to submit a new injury claim by entering relevant details and uploading supporting documents.

**Programmer:** Amani

**Date due:** 3/9/25

**Language:** Python

**Event(s)**

Occurs when a new claim is being submitted and approved

| **Input Name** | **Type** | **Provided by** | **Notes** |
| --- | --- | --- | --- |
| Employee ID | String | Employee | Used to identify the submitter |
| Injury Details | Text | Employee | Description of the injury |
| Date of Injury | Date (mm-dd-yyyy) | Employee | Must be a valid past date |
| Supporting Documents | Record | Employee | Uploaded files for claim verification |
| Approval Submission | Boolean | D1: Claims Database | Triggers approval workflow |

| **Output Name** | **Type** | **Used by** | **Notes** |
| --- | --- | --- | --- |
| Claim ID | String | D1: Claims Database | Auto-generated unique identifier |
| Confirmation Message | Text | Clerk | Notifies user of successful submission |
| Approval Request | Boolean | D1: Claims Database | Triggers further processing |
| Stored Documents | Record | D1: Claims Database | Saved for verification |

**Pseudocode / Decision Tree / Decision Matrix**

#### **Pseudocode for Submit New Injury Claim**

START

Submit\_new\_injury\_claim(employee\_id, injury\_details, date\_of\_injury, documents):

If not all([employee\_id, injury\_details, date\_of\_injury]

return "Error: Missing required fields"

try:

current\_date

IF (current\_date <input\_date)

return: "Error: Date of injury cannot be in the future"

except ValueError:

return "Error: Invalid date format (use mm-dd-yyyy)"

FOR doc in documents:

uploaded\_files.append(doc)

DEF send\_for\_approval(claim\_id):

return f"Claim {claim\_id} sent for approval"

approval\_message = send\_for\_approval(claim\_id)

return f"Claim submitted successfully! Claim ID: {claim\_id}. {approval\_message}"

uploaded\_files = []

End

END IF

Save claim details in Claims Database

Upload supporting documents

Generate unique Claim ID

Send approval request for processing

Display confirmation message to the employee

END

### **Other Notes**

Error handling should be implemented for missing or invalid inputs. The system should enforce data validation rules (e.g., required fields, date format). Uploaded documents must meet file type and size restrictions. The approval process should be tracked for claim status updates.

**Program Specification No.: 3 System: Claims System for South Dakota Department of Labor, Worker’s Compensation Division**

**Module**Name: Upload Medical Reports  
Purpose: Upload medical report files and verify their validity before storage  
Programmer: Sahil  
Date due: 3/9/25  
Language: Python

**Event(s)**Occurs when a user uploads a medical report file for submission

| **Input Name** | **Type** | **Provided by** | **Notes** |
| --- | --- | --- | --- |
| File Date | Date (mm-dd-yyyy) | User | Date of file upload |
| File Information | File (PDF, DOCX, etc.) | User | Contains medical report details |
| Receive Files | Record | System | Claims details to archive |

| **Output Name** | **Type** | **Used by** | **Notes** |
| --- | --- | --- | --- |
| Approved File | Record | D3: Medical Records | File is verified and stored |
| Disapproved File | Record | User Notification | File type is invalid or corrupt |
| Uploaded Reports | Record | System | Successfully stored medical reports |

**Pseudocode / Decision Tree / Decision Matrix**Upload\_Medical\_Reports (FileDate, FileInfo)  
RECEIVE FileInfo  
VERIFY FileType  
IF (FileType is valid) THEN  
APPROVE File  
STORE in D3: Medical Records  
ELSE  
DISAPPROVE File  
Notify User  
End of File

**Other notes**The employee will receive an email or SMS message informing them about the status of their uploaded medical report file.

**Program Specification No.:** 4  **System:** Claims System for South Dakota Department of Labor, Worker’s Compensation Division

**Module**

**Name:** Archive Old Claims and Medical Reports

**Purpose:** Archive claims and medical reports that are 6 years and older

**Programmer:** Marina

**Date due:** 3/9/25

**Language:** Python

**Event(s)**

Occurs when claims or medical report’s date of submission reaches 6 years from current day.

| **Input Name** | **Type** | **Provided by** | **Notes** |
| --- | --- | --- | --- |
| Claim Date | Date (mm-dd-yyyy) | D1: Claims Database | Retrieved to calculate the age of the claim |
| Medical Report Date | Date (mm-dd-yyyy) | D3: Medical Records | Retrieved to calculate the age of the medical records |
| Claim Information | Record | D1: Claims Database | Claims details to archive |
| Medical Report Information | Record | D3: Medical Records | Medical Records details to archive |

| **Output Name** | **Type** | **Used by** | **Notes** |
| --- | --- | --- | --- |
| Approved Claims | Record | D2: Archive Database | Approved when claims are older than 6 years |
| Approved Medical Report | Record | D2: Archive Database | Approved when medical reports are older than 6 years |
| Disapproved Claims | Record | D1: Claims Database | Claim is not older than 6 years, so it is not transferred to the archive database. |
| Disapproved Medical Report | Record | D3: Medical Records | Medical report is not older than 6 years, so it is not transferred to the archive database. |

**Pseudocode / Decision Tree / Decision Matrix**

Archive\_Old\_Claims\_And\_MR (ClaimDate, MRDate)

IF (ClaimDate < CurrentDate - 6 years) THEN

Approve Claim

Store in D2: Archive Database

ELSE

Disapprove Claim

End of File

IF (MRDate < CurrentDate - 6 years) THEN

Approve MR

Store in D2: Archive Database

ELSE

Disapprove MR

End of File

**Other notes**

The employee will receive an email or SMS message informing them that their claim or medical report has been archived.

# Appendix E - Data Dictionary

# Entity Descriptions

| **Entity** |
| --- |
| Entity Name: Clerk |
| Entity Description: User who has authorized access to the claims system. Clerks are responsible for managing and searching claims, generating reports, and assisting with claim processing. |
| Attributes:  PK Clerk\_ID  Clerk\_FirstName  Clerk\_LastName |
| Notes: Clerks have varying permission levels depending on their role within the organization. |

| **Entity** |
| --- |
| Entity Name: Relevant Documents |
| Entity Description: The Relevant Documents entity stores all documents that are associated with the claims submitted by employees. These documents may include medical reports, accident reports, and any other documentation related to the employee’s claim. The documents are uploaded by clerks and are linked to specific claims in the MySQL Claims Database. |
| Attributes:  PK Document\_ID  Document\_Date  Document\_Type  Document\_Description |
| Notes: This entity helps keep track of all documents tied to a claim, ensuring that all required paperwork is properly linked, stored, and easily retrievable. |

| **Entity** |
| --- |
| Entity Name: MySQL Claims Database |
| Entity Description: It stores all injury claims submitted by employees. It includes details about each claim, such as the date of injury, the date the claim was filed, and a description of the claim. This database is essential for tracking and processing claims and is linked to other entities, such as MySQL Medical Records and MySQL Archive Database. |
| Attributes:  PK Claim\_ID  Claim\_DateOfInjury  Claim\_Date  Claim\_Description |
| Notes: Claims in this database are subject to archival after a specified period and are linked to related medical reports. |

| **Entity** |
| --- |
| Entity Name: Employee |
| Entity Description: The Employee entity represents individuals who submit claims within the claims management system. This entity captures key details about each employee involved in the claims submission process. |
| Attributes:  PK Employee\_ID  Employee\_FirstName  Employee\_LastName  Employee\_Address  Employee\_Phone |
| Notes: This entity is linked to MySQL Claims Database entity to show who submits each claim. |

| **Entity** |
| --- |
| Entity Name: MySQL Archive Database |
| Entity Description: It stores archived claims within the system. It is used for managing the archiving process and tracking the status and type of each archived claim. |
| Attributes:  PK Archive\_ID  FK Claim\_ID  FK Claim\_DateofInjury  FK Claim\_Date  FK Claim\_Description  Archive\_Type |
| Notes: This entity is related to both the MySQL Claims Database and MySQL Medical Records to manage archived claims and medical records effectively. |

| **Entity** |
| --- |
| Entity Name: MySQL Medical Records |
| Entity Description: This entity stores the medical reports filed. This is where the medical reports are stored and retrieved from. |
| Attributes:  PK MRecord\_ID  FK Claim\_ID  MRecord\_Date  MRecord\_DateofInjury |
| Notes: Claim\_ID is a foreign key from the claims database. |

| **Entity** |
| --- |
| Entity Name: Medical Provider |
| Entity Description: This is the health professional who is responsible for uploading medical reports to the medical records database. |
| Attributes:  PK Medical\_ID  Medical\_FirstName  Medical\_LastName |
| Notes: The primary key of this entity is Medical\_ID. |

# Attribute Descriptions

| **Attribute** |
| --- |
| Attribute Name: Clerk\_ID |
| Attribute Description: A unique identifier for each clerk in the system. |
| Values and Meanings: Numeric or alphanumeric, system-generated. |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: This attribute is used as the primary key for the Clerk entity. |

| **Attribute** |
| --- |
| Attribute Name: Clerk\_FirstName |
| Attribute Description: The first name of the clerk. |
| Values and Meanings: Text, includes first name. |
| Type and Length: Varchar (30 characters, variable storage size) |
| Notes: Used for identification and display in reports and system logs. |

| **Attribute** |
| --- |
| Attribute Name: Clerk\_LastName |
| Attribute Description: The last name of the clerk. |
| Values and Meanings: Text, includes last name. |
| Type and Length: Varchar (40 characters, variable storage size) |
| Notes: Used for identification and display in reports and system logs. |

| **Attribute** |
| --- |
| Attribute Name: Document\_ID |
| Attribute Description: Unique identifier for each uploaded document. |
| Values and Meanings: Numeric code that uniquely identifies each document in the Relevant Documents entity. |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: This attribute is the primary key of the Relevant Documents entity. |

| **Attribute** |
| --- |
| Attribute Name: Document\_Date |
| Attribute Description: The date on which the document was submitted. |
| Values and Meanings: Date format (mm-dd-yyyy). |
| Type and Length: Date (3 bytes) |
| Notes: The date of submission can be useful for documentation purposes. |

| **Attribute** |
| --- |
| Attribute Name: Document\_Type |
| Attribute Description: The file type of the document submitted. |
| Values and Meanings: To understand which file type was used. |
| Type and Length: Varchar (10 characters, variable storage size) |
| Notes: PDF, .dox, and .txt are all acceptable file types. |

| **Attribute** |
| --- |
| Attribute Name: Document\_Description |
| Attribute Description: A short description that gives key points about what the document entails. |
| Values and Meanings: Text, includes relevant details about the injury. |
| Type and Length: Varchar (200 characters, variable storage size) |
| Notes: Relevant documents could include police reports and witness statements, along with pictures of where the injury occurred. |

# 

| **Attribute** |
| --- |
| Attribute Name: Claim\_ID |
| Attribute Description: Unique identifier for each claim. |
| Values and Meanings: Numeric code that uniquely identifies each claim in the MySQL Claims Database. |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: This attribute is the primary key for the MySQL Claims Database. |

| **Attribute** |
| --- |
| Attribute Name: Claim\_DateOfInjury |
| Attribute Description: The date on which the injury occurred. |
| Values and Meanings: Date format (mm-dd-yyyy). |
| Type and Length: Date (3 bytes) |
| Notes: This attribute is the date of the injury, NOT the date of submission of claim. |

| **Attribute** |
| --- |
| Attribute Name: Claim\_Date |
| Attribute Description: The date on which the claim was filed. |
| Values and Meanings: Date format (mm-dd-yyyy). |
| Type and Length: Date (3 bytes) |
| Notes: This attribute is used to track the timeline of the claim process; important for legal and administrative purposes. |

| **Attribute** |
| --- |
| Attribute Name: Claim\_Description |
| Attribute Description: A brief description of the claim, detailing the circumstances and nature of the injury. |
| Values and Meanings: Text, includes relevant details about the injury. |
| Type and Length: Varchar (200 characters, variable storage size) |
| Notes: This attribute can be helpful for clerks and employees alike to get a quick overview of the circumstances pertaining to the injury claim. |

| **Attribute** |
| --- |
| Attribute Name: Employee\_ID |
| Attribute Description: Unique identifier assigned to each employee |
| Values and Meanings: Numeric code that distinguishes each employee within the system. |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: Primary key for the Employee entity, used to link employees with the claims they submit. |

| **Attribute** |
| --- |
| Attribute Name: Employee\_FirstName |
| Attribute Description: First name of the employee |
| Values and Meanings: Text field containing the employee’s first name. |
| Type and Length: Varchar (30 characters, variable storage size) |
| Notes: Used to identify employees by name within the system. |

| **Attribute** |
| --- |
| Attribute Name: Employee\_LastName |
| Attribute Description: Last name of the employee |
| Values and Meanings: Text field containing the employee’s last name. |
| Type and Length: Varchar (40 characters, variable storage size) |
| Notes: Used to identify employees by name within the system. |

| **Attribute** |
| --- |
| Attribute Name: Employee\_Address |
| Attribute Description: Residential address of the employee. |
| Values and Meanings: Text field formatted to include street address, city, state, and postal code. |
| Type and Length: Varchar (30 characters, variable storage size) |
| Notes: Optional field; relevant for record-keeping or correspondence. |

| **Attribute** |
| --- |
| Attribute Name: Employee\_Phone |
| Attribute Description: Personal phone number of the employee. |
| Values and Meanings: Text field formatted to include the phone number. ((XXX) XXX - XXXX) |
| Type and Length: Varchar (10 characters, variable storage size) |
| Notes: Optional field; relevant for record-keeping or correspondence. |

| **Attribute** |
| --- |
| Attribute Name: Archive\_ID |
| Attribute Description: Unique identifier for each archived record |
| Values and Meanings: Numeric code that uniquely identifies each entry in the Archive Database. |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: Primary key for the Archive Database entity. |

| **Attribute** |
| --- |
| Attribute Name: Claim\_ID |
| Attribute Description: Unique identifier that links each archived record to a specific claim. |
| Values and Meanings: Numeric code corresponding to a claim in the Claims Database. |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: Acts as a foreign key, allowing reference to claims from the Claims Database. |

| **Attribute** |
| --- |
| Attribute Name: Claim\_DateOfInjury |
| Attribute Description: The date on which the injury occurred. |
| Values and Meanings: Date format (mm-dd-yyyy). |
| Type and Length: Date (3 bytes) |
| Notes: Foreign key from the claims database, used for claim tracking and as a key piece of information for reports and archival processes. |

| **Attribute** |
| --- |
| Attribute Name: Claim\_Date |
| Attribute Description: The date on which the claim was filed. |
| Values and Meanings: Date format (mm-dd-yyyy). |
| Type and Length: Date (3 bytes) |
| Notes: Foreign key from the claims database, used to track the timeline of the claim process; important for legal and administrative purposes. |

| **Attribute** |
| --- |
| Attribute Name: Claim\_Description |
| Attribute Description: A brief description of the claim, detailing the circumstances and nature of the injury. |
| Values and Meanings: Text, includes relevant details about the injury. |
| Type and Length: Varchar (200 characters, variable storage size) |
| Notes: Foreign key from the claims database, used by clerks and medical providers to understand the context of the claim; this field may be referenced when generating claim reports. |

| **Attribute** |
| --- |
| Attribute Name: Archive\_Type |
| Attribute Description: Specifies the category or nature of the archived claim |
| Values and Meanings: Boolean where True=claims or False=medical record |
| Type and Length: Boolean (1 byte) |
| Notes: Helps in categorizing archived claims for easier retrieval and management. |

| **Attribute** |
| --- |
| Attribute Name: MRecord\_ID |
| Attribute Description: Identifier of a medical record that gives a specific ID to a record for storage and retrieval. |
| Values and Meanings: Numeric code corresponding to a medical record in the Medical Reports Database. |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: MRecord\_ID is a primary key of the Medical Reports Database entity. |

| **Attribute** |
| --- |
| Attribute Name: Claim\_ID |
| Attribute Description: Identifier that links a claim to a specific ID for ease of retrieval and documentation. It can be combined with medical records so certain records for a claim can be linked together. |
| Values and Meanings: Numeric code corresponding to a claim in the Claims Database |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: This is a foreign key for the Medical Reports Database from the claims database. |

| **Attribute** |
| --- |
| Attribute Name: MRecord\_Date |
| Attribute Description: This attribute records the date that the medical report was filed on. |
| Values and Meanings: A date type value (mm-dd-yyyy). |
| Type and Length: Date (8 bytes) |
| Note: This can make searching for medical reports easier if only the date is known. |

| **Attribute** |
| --- |
| Attribute Name: MRecord\_DateofInjury |
| Attribute Description: Stores the date of the actual injury that is in the medical report. |
| Values and Meanings: A date type value. (mm-dd-yyyy). Separate from MRecord\_Date, since that is the date of the filing, not the date of the actual injury. |
| Type and Length: Date (8 bytes) |
| Notes: This can help a clerk search for a claim or medical report if only the date of the injury is known. |

| **Attribute** |
| --- |
| Attribute Name: Medical\_ID |
| Attribute Description: Stores a unique identifier assigned to each medical provider to track who submitted the medical reports. |
| Values and Meanings: Numeric code that uniquely identifies each medical provider in the system. |
| Type and Length: AutoNumber (Long Integer, 4 bytes) |
| Notes: Primary key for the Medical Provider entity. |

| **Attribute** |
| --- |
| Attribute Name: Medical\_FirstName |
| Attribute Description: This is the first name of the medical provider who completed the filing of medical records. |
| Values and Meanings: Text, includes medical’s first name. |
| Type and Length: Varchar (30 characters, variable storage size) |
| Notes: Medical\_FirstName can be found from the Medical\_ID, since they should both be linked to the same medical provider. |

| **Attribute** |
| --- |
| Attribute Name: Medical\_LastName |
| Attribute Description: This is the last name of the medical provider who completed the filing of medical records. |
| Values and Meanings: Text, includes medical’s last name. |
| Type and Length: Varchar (40 characters, variable storage size) |
| Notes: Medical\_LastName can be found from the Medical\_ID, since they should both be linked to the same medical provider. |

# 

# Relationship Descriptions

| **Relationships** |
| --- |
| Relationship Name: Uploads |
| Relationship Description: A Clerk uploads zero or many Relevant Documents. Each Relevant Document is uploaded by one and only one Clerk. |
| Attached Entities:  Clerk  Uploaded by - Min: One, Max: One  Relevant Documents  Uploads - Min: 0, Max: Many |
| Notes: Clerks may upload multiple documents, but each document is associated with only one Clerk. |

| **Relationships** |
| --- |
| Relationship Name: Included in |
| Relationship Description: A Claim includes zero or many Relevant Documents. Each Relevant Document is included in one and only one MySQL Claim Database. |
| Attached Entities:  Relevant Documents  Included in - Min: 0, Max: Many  Claims Database  Included - Min: One, Max: One |
| Notes: A claim may have no attached documents or multiple documents related to it. |

| **Relationships** |
| --- |
| Relationship Name: Submits Claims |
| Relationship Description: An employee submits zero or many claims to MySQL Claims Database, Claims are submitted by one and only one employee |
| Attached Entities:  Employee  Submits - Min: 0, Max: Many  Claims Database  Is submitted by - Min: One, Max: One |
| Notes: An employee does not need to submit any claims to the claims database, but they can submit many if needed. |

| **Relationships** |
| --- |
| Relationship Name: Claims Archived by |
| Relationship Description: The claims database can archive zero or many claims, and the archive database’s claims are archived by one and only one claims database. |
| Attached Entities:  Claims Database  Can archive - Min: 0, Max: Many  Archive Database  Archived by - Min: One; Max: One |
| Notes: There are other processes required to retrieve claims from the archive database. |

| **Relationships** |
| --- |
| Relationship Name: Medical Reports Archived by |
| Relationship Description: The medical reports database can archive zero or many claims, and the archive database’s medical reports are archived by one and only one medical reports database. |
| Attached Entities:  Medical Reports Database  Can archive - Min: 0, Max: Many  Archive Database  Archived by - Min: One; Max: One |
| Notes: There are other processes required to retrieve reports from the archive database. |

| **Relationships** |
| --- |
| Relationship Name: Submits Medical Reports |
| Relationship Description: A medical provider submits zero or many medical reports to the MySQL Medical Records, Medical Reports from MySQL Medical Records are submitted by one and only one medical provider |
| Attached Entities:  Medical Reports Database  Submits - Min: 0, Max: Many  Medical Provider  Is submitted by - Min: One, Max: One |
| Notes: A medical provider does not need to submit any medical reports to MySQL Medical Records, but they can submit many if needed. |