REVENORA AI: AN AI-POWERED APPROACH TO MEDICAL BILLING AND CLAIM VALIDATION

Ms.M.Suguna
Assistant Professor
Department of Computer Science and Engineering
SNS College of Technology, Coimbatore Tamil Nadu India,
sugunasae@gmail.com

Ms.B.Prarthana
III year B.E-CSE
Department of Computer Science and Engineering
SNS College of Engineering, Coimbatore Tamil Nadu India,
prarthanabharathiraja@gmail.com

Ms.D.Kiruthika
III year B.E-CSE
Department of Computer Science and Engineering
SNS College of Engineering, Coimbatore Tamil Nadu India,
kirithidev8@gmail.com

Ms.K.C.Gopika Sree
III year B.E-CSE
Department of Computer Science and Engineering
SNS College of Engineering, Coimbatore Tamil Nadu India,
kcgopika0423@gmail.com

Ms.R.Sudharsshana
III year B.E-CSE
Department of Computer Science and Engineering
SNS College of Engineering, Coimbatore Tamil Nadu India,
sudharsh2201@gmail.com

***Abstract*: Medical billing is about converting patient visits, clinical records, treatment information, medical codes and insurance data into claims for reimbursement. It can be time-consuming to manually review these elements and if any information is missing or inconsistent, it can result in rework, processing delays, and claim rejection. In this paper, we introduce a web-based prototype of a medical billing and claim-validation system that leverages Revenora AI to assist with claim validation before submission and in the monitoring phase after. It has several key features, such as an AI Claim Reviewer, Billing Staff Dashboard, Active Claims Queue, Claim Tracking, Patient Care and Claim Portal, and Notifications. The reviewer prioritizes possible issues and provides supporting information, severity indicators and suggested actions for authorized billing professionals. The platform integrates the claim level results into the operational workflow visibility and enables employees to investigate issues and track the progress of claims within the same application. Revenora AI is a decision support tool, not a replacement for clinical, coding/billing judgment and is not a claims adjudicator or approver/rejecter. The current implementation is a demonstration of the main interfaces and the expected workflow. There is no evidence of production payer integration, full coverage of coding rules, or reduced claim denials and measured accuracy of AI. Secure healthcare-system connectivity, improved document processing, privacy controls and systematic evaluation with representative claims are future tasks.**

Keywords - Revenora AI, Medical Billing, Claim Validation, Healthcare Revenue Cycle, Artificial Intelligence, Coding Review, Denial Risk, Decision Support.

I. INTRODUCTION

Medical billing involves the administrative procedure that transforms the details of a patient visit into a claim for reimbursement. Patient identity, diagnosis, treatment information, clinical notes, procedure information, medical codes, supporting documents and insurance coverage information can be included in a claim. These elements are typically developed or managed by various departments and need to be verified as a group before filing a claim with an insurer or third party administrator (TPA). This process can be interrupted by a missing document, an incomplete treatment note, or inconsistent patient details or mismatch between clinical documentation and a billed code. Staff may have to find the original record, call another department, correct the claim and re-review.

The administrative time required for such rework can slow down the claim processing. The challenge is not complete after claim preparation. Billing teams need to keep track of claims that are awaiting verification, claims that are ready for submission, claims that are under payer review and claims that need correction. If claim status and review history are kept in different records or channels of communication, staff will need to invest more time in finding information and determining if it is a priority claim.

Revenora AI solves this workflow issue by packaging the pre-submission review, issue identification, staff monitoring, claim tracking and patient-facing status information into a single web-based prototype. It is meant to assist billing professionals in identifying potential issues at an earlier stage, understanding why a claim has been flagged, and the represented workflow of the claim.

A. Problem Statement

Medical billing is a complex process with a lot of documentation, coding requirement, insurance verification and reimbursement procedures. Manual workflows are labor-intensive and can leave out important data or inconsistencies until later in the process. A valuable support platform should be able to find out what information is missing, offer clear explanations for potential problems, recommend solutions, and keep track of claims and give professionals a clear view of the

status of the claims.

B. Objectives

The main goal is to create a viable solution that helps billing staff to audit claims prior to submission and track them throughout the claims processing. Specific objectives include identifying gaps in documentation, raising potential medical necessity concerns, outlining coding and compliance concerns, outlining potential risk of denial, suggesting corrective action steps, organizing claims by workflow stage, and notifying of important events. Another goal is to give the patient a more simplified perspective of claims progress without divulging any internal billing information.

C. Scope and Contribution

This project includes a browser-based prototype with a landing page, AI Claim Reviewer, Billing Staff Dashboard, Active Claims Queue, Claim Tracking interface, Patient Care and Claim Portal, and Notifications. The contribution is the relationship between findings from a claim-level review and visibility of an operational workflow. A warning is shown with context and a potential next step, and is linked to a claim that staff can maintain an eye on. Payer-specific rule execution, production of protected health information are not yet demonstrated, nor is direct integration with live EHR/EMR systems with payer portals.

II. RESEARCH THE LITERATURE AND BACKGROUND INFORMATION

The quality of clinical documentation, the use of diagnosis and procedure codes, and the needs of the payer to whom the claim is submitted are all important to medical claim processing. The background for defining validation checks is official coding and claims-processing guidance. The references used in this project are the International Classification of Diseases, Tenth Revision (ICD-10), the National Correct Coding Initiative (NCCI) and the Medicare Claims Processing Manual.

A. Medical Coding and Claims Processing

ICD-10 coding information for classification of diagnoses is published by the Centers for Medicare & Medicaid Services (CMS) [1]. Diagnosis codes are intended to convey the condition that is related to the service and should be substantiated by the clinical record. The presence of a code in a claim does not necessarily indicate that it is the correct code for a specific encounter. The supporting documentation and appropriate coding guidance shall also be taken into account.

Additionally, CMS has the National Correct Coding Initiative with correct-coding methodologies and edits that are designed to decrease improper coding combinations [2].

These are applicable when there are code combinations or relationships in a claim that need to be investigated. There is more information about claims processing procedures and requirements in the Medicare Claims Processing Manual [3]. These materials collectively demonstrate the need to review claims with clinical evidence, coding information and applicable payer rules - rather than just looking at isolated fields.

B. Technology for Workflow Applications

A claim-validation platform requires a web interface, a server-side application layer, an AI service and a data-management layer. Frontend captures user actions and displays the claim information and findings.

The backend manages requests, processes application rules and interacts with the AI service. Related entities like patients, claims, documents, findings, status events and notifications can be stored in structured storage. The relevance of documentation for deployment is that the user accesses the prototype via a browser.

This implementation is based on a React frontend, TypeScript, and Vite, along with a Node.js backend, for Revenora AI. The back end interacts with the Groq API for AI claim review. The prototype that is deployed is hosted via Vercel. These technologies are all technologies of the application environment and do not, in isolation, demonstrate the clinical validity and accuracy of the AI output.

C. Existing Workflow and Identified Gap

Typical manual claim process involves employees gathering claim data, reviewing documents, matching with coding and insurance criteria, editing missing data and tracking claim through submission and payer review.

These activities can be spread out among various tools or communication methods. A simple warning isn't very useful if the user doesn't have access to the evidence that led to the warning or doesn't understand why it is important or what should be done next.

The project is aimed at this gap in operation between the identification of a potential claim issue and the ability of staff to take action.

Revenora AI collates review insights within a claim-level workspace, and ties that workspace to staff monitoring and claim status. It also distinguishes detailed information on internal billing from the easier progression information that patients require.

D. Proposed System

The system proposed integrates structured pre-validation, AI support for interpretation and workflow monitoring. Simple checks will detect missing or inconsistent fields.

AI-driven analysis can be used to interpret unstructured documentation or to compare clinical evidence to the context of the claim. Findings are returned in a standardised format for staff to view the category, explanation, evidence, severity and suggested action. Authorized people make the decision to correct, clarify or proceed with the claim.

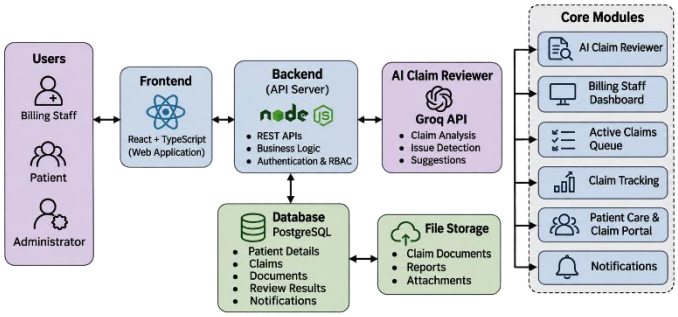
E. Design Principles

The design is based on three principles. Findings must be explainable: a user must be able to explain the reason for a flag. Second, the review should be human controlled: AI output is a call to investigate, not to make a decision.

Third, claim information should be linked throughout the workflow, so that a correction, status change or notification could be linked to the correct claim.

III. PROPOSED METHODOLOGY

Revenora AI has a multi-stage claim review process. Information and supporting documents related to the claim are provided to the application. The system validates the information provided and identifies missing data and possible inconsistencies. AI-driven analysis can then yield structured results for billing personnel. Staff review the evidence, make corrections as needed and review again. The claim passes through the submission and payer-review stages in the tracking interface after it has been internally reviewed.



A. Claim Input, Pre-validation

The workflow starts with patient, treatment, documentation, coding and insurance related information that goes with a claim. First checks determine if necessary information is present. These may include patient information that is not complete, treatment information missing, supporting paperwork missing, medical codes missing, insurance information missing or needs verification. This goal is to ensure that there is no information gap that can be seen before the claim advances into the billing process.

B. AI Claim Reviewer

The AI Claim Reviewer is the main workspace for identifying claims. A selected claim is discussed and the results are shown in discrete statements, not free-form answers. A finding can contain a category, a description, supporting evidence, a severity or confidence rating and a suggested action (if available). The prototype represents issues like lack of discharge documentation and lack of medical necessity support.

The structured presentation enables a billing professional to grasp the reasons behind the flagging of a claim and what information is needed to review. The back-end pushes the needed claim context to the AI service and handles the response from the AI service to present in the application. The output of AI is subject to review of the source record as incomplete or ambiguous input may impact the output.

C. Documentation and Medical-Necessity Review

Documentation Review is a process for determining if the documentation required for a service exists. This can be clinical notes, treatment information, discharge information or any other documents that are relevant to the claim. Medical-necessity review looks at the clinical record to determine if the billed service seems to be supported by the information in the medical record. The system can display a finding for the documentation to be reviewed if the documentation is incomplete or the relationship between the documented condition and service is not clear. These checks focus on potential deficiencies and are not meant to supplant clinical judgment.

D. Coding and Compliance Considerations

Coding review is a process that determines if the medical codes that are recorded for a claim are consistent with the information provided on the claim and if coding related concerns need to be addressed. Compliance related review may identify potential coding issues, documentation or claim completeness issues. The ICD-10 and NCCI references give relevant context [1] [2]. The current prototype shows the results of the coding related findings at a workflow level, and full rule coverage and continuously updated validation for payers are future needs.

E. Denial-risk Information and Corrective Action

Denial-risk information consolidates issues that could result in a claim needing to be corrected or verified. It can be anything from missing documentation, coding mismatch, coverage-policy concerns and possible duplicate billing. The concern is presented to the reviewer and a suggested action is given to staff for investigation. If a correction or clarification is made, the claim may be reviewed again. The final decision to proceed is up to the authorized billing professional.

F. Claim Tracking and Notifications

Claim tracking offers insights into the hospital verification process, pre-authorization preparation, submission to insurance, payer review and approval or settlement. Notifications inform about workflow events like approval updates, insurance updates, new review findings and claims ready to be submitted. Linking these functions to the claim record assists staff to follow work following initial review.

G. Validation Logic

The validation approach is a hybrid of deterministic validation and AI-based interpretation. Deterministic checks can be used to check for presence of a required field, or for checking if a claim identifier occurs multiple times in the claim. AI-assisted analysis is designed to be used with unstructured documentation for interpretation or with clinical evidence to compare with the context of the claim. Findings are normalized into a common structure to be displayed uniformly.

Findings = validation of claim data, documents, codes and insurance information

This is not an accuracy formula, measured or a statement of autonomous claim adjudication. Design is human in the loop: AI-generated findings are prompts to be checked by authorized staff.

IV. THE SYSTEM ARCHITECTURE AND IMPLEMENTATION

The system is structured in cooperating layers, which means that the user interface, workflow logic, AI processing and data management are assigned to different layers. The architecture

enables a claim to be reviewed, corrected and monitored to status. External healthcare and payer connectivity is viewed as an integration layer of the future.

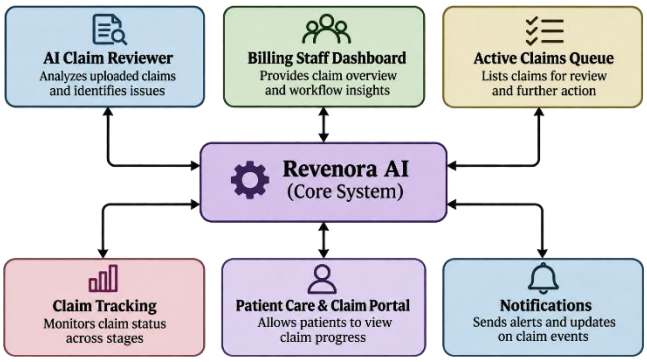


Fig. 5. Module Interaction Diagram of Revenora AI

A. Presentation Layer

The presentation layer features the landing page, Billing Staff Dashboard, AI Claim Reviewer, Active Claims Queue, Claim Tracking, Patient Care and Claim Portal, and Notifications. The dashboard provides an overview of claim activity. The reviewer presents results of the claim. The queue and tracking interfaces enable staff to determine which stage each claim is at. The patient facing portal shows a simplified progression view.

B. Backend and Application Layer

The Node.js back end is the application layer on the server. It receives requests from the frontend, coordinates the claim-review operation, communicates with the AI service and returns a result for the frontend to display. It also offers a spot to perform application-level validation and workflow activities like updating a claim status. This way, the service logic remains on the server, instead of being stored in the browser-side code.

C. AI Processing Layer

The AI processing layer will help with the interpretation of claim context and the generation of review findings using the Groq API. The application supplies relevant information to the service and fills the reviewer interface with the results. The results should be considered as recommendations only. For a production implementation, it would be necessary to implement strict input filtering, structured-output validation, error handling, prompt/version management and systematic testing against representative claims.

D. Data and Integration Layer

The platform requires related information for patients, claims, documents, findings, status events and notifications. There can be multiple claims associated with a patient, and multiple documents, findings and status events in a claim. These relationships aid in the continuity of the claim history between staff and patients. Structured data storage is part of the intended architecture as identified in the project report, but the final production database configuration should be documented based on the final deployed configuration.

The integration layer will be used for future connections with Electronic Health Record (EHR) and Electronic Medical Record (EMR) systems and payer platforms. Appropriate interfaces, identity verification, authorization, data mapping, and audit controls would be necessary for such connections.

E. User Roles and Interface Design

Billing staff require detailed claim information, validation findings and suggested corrective actions. Administrative users require a consolidated view of claim activity and approvals, claim denial reasons, and review workload. Patients need to have a more streamlined understanding of claim flow and related payment or coverage data. Revenora AI delivers these perspectives individually to allow each user to view information that is relevant to the task. Role-based access control is a required production requirement and should not be just a hiding of interface elements, but should be enforced by the backend.

F. Technology Stack

The official application environment is React, TypeScript and Vite for the front-end, Node.js for the back-end, the Groq API for AI-powered review and Vercel for web deployment. You can access the application via a regular browser at https://revenora-ai.vercel.app/. This is an environment that allows an academic demonstration of the proposed workflow. Using the system as a web application does not mean that it is ready for use in the healthcare environment with live data or production payer systems.

G. Security Considerations

The information in healthcare should be guarded carefully. Secured authentication, server-side authorization, encrypted transport and storage, access control to documents, audit logging, secure secret management, backups and defined data retention policies are all features for a production version. API credentials must be kept on the server. Access should be based on the principle of least privilege and sensitive patient data should not be displayed in logs or unnecessary notifications.

V. RESULTS AND DISCUSSION

The evaluation will take into account if the prototype shows the desired claim workflow, and if the key user-facing modules are working together. The project evidence available includes the web prototype deployed, views of the interface, and documented functional test scenarios. The results are therefore not in terms of clinical or financial performance claims but rather as prototype-level functional outcomes.

A. Demonstrated Prototype Modules

The landing page introduces Revenora AI and provides an entry point to the application. The Billing Staff Dashboard displays claim summary data, validation status, approvals and denial cause indicators. The Active Claims Queue is used to queue claims in various stages. The AI Claim Reviewer shows the selected-claim results along with documentation and medical-necessity issues. The Claim Tracking feature offers visibility of claim progress, and the Patient Care and Claim Portal delivers a streamlined view of claims' status. Notifications are used to send updates about workflow.

Module	Demonstrated prototype outcome
Landing page	Workflow overview and application entry
Billing Staff Dashboard	Claim summary, trends and denial-cause indicators
AI Claim Reviewer	Claim selection and review findings
Active Claims Queue	Claim organization and workflow visibility
Claim Tracking	Claim progression and status visibility
Patient Care and Claim Portal	Patient-facing claim progression
Notifications	Workflow-related updates and alerts

B. Functional Testing

Functional testing verifies if each function is taking the expected screen or state. Workflow testing determines if a claim can transition from one workflow step to another without losing the information associated with the claim. Usability testing takes into account whether a user can interpret the meaning of a finding or status without explanation. The following scenarios are documented: Open the landing page, Open the dashboard and view claim metrics, Select a claim and view claim details, Run a review and view a finding, Open the patient portal, Open notifications, View a missing document finding, View a coding related finding (when it exists), Move a claim to a new status.

Test scenario	Expected behaviour
Open landing page	Application introduction is displayed
Open dashboard	Claim metrics and summary are visible
Select a claim	Selected claim details are displayed
Run AI review	Review findings are returned for display
Missing-document case	Relevant documentation gap is shown
Coding-related case	Available coding concern is displayed
Open patient portal	Simplified claim status is visible
Change claim status	Updated workflow state is reflected

C. Observed Outcomes

The project report documents the successful implementation of the Landing page, Billing Staff Dashboard, Active Claims Queue, Claim tracking, Patient care and Claim portal, Notifications and claim status visibility. Additional prototype-demonstrated features include the AI Claim Reviewer and claim-validation results and denial-risk information. These outcomes demonstrate that review of claims, monitoring of staff, tracking of claims and visibility of patients can be integrated into a single web-based interface.

D. Practical Value

The benefit of the platform is that it makes it easier to grasp the state of a claim. A billing professional can start at the dashboard, open a claim and review the findings and the suggested correction. This can then be tracked throughout its workflow stages. The patient view makes it easier to see how the patient's condition is progressing, and notifications inform you of events that might need to be addressed. These are proposed workflow gains, but have not yet been quantified in terms of processing time or reimbursement.

E. Limitations

The current implementation is an academic working prototype and is NOT a production ready healthcare billing system. It does not directly integrate with live hospital EHR/EMR systems or with all payer-specific needs and every medical billing scenario. The completeness and accuracy of source records and the rules and AI service used to validate the data are both factors that affect validation quality.

AI generated findings are not final coding, clinical or reimbursement decisions and should be reviewed by authorized professionals. The project has not set an accuracy percentage in the AI model, nor has it proven a reduction in claim denials, or a production comparison against a hospital billing baseline. In real-world deployment, the security controls, privacy protections, rule coverage, and formal evaluation and extensive operational testing should be enhanced.

VI. CONCLUSION AND FUTURE WORK

Revenora AI offers a streamlined medical billing and claim validation process, offering a unified web-based prototype that incorporates pre-submission review, claim issue identification, staff monitoring, claim tracking, patient visibility, and notifications. The main benefit it offers is organizing the information that billing experts need to examine the claim and make a decision on what to do.

The AI Claim Reviewer offers a list of potential problems and recommended steps with the supporting documents, and the dashboard and tracking interfaces give an overview of the entire claim process. The major interfaces and the intended claim-review journey are shown in the current prototype. It provides a baseline for future development, but can't guarantee production-level payer connectivity, extensive coding validation, AI accuracy or financial gains. AI results are still decision support data and will need to be reviewed by a qualified billing, coding or clinical staff member.

A. Future Enhancements

One of the key improvements is the secure integration with EHR/EMR systems. This connectivity may enable patient,

treatment and clinical documentation to be made available in the validation workflow and limit the need for duplicate manual entry. Payer and insurance integration may offer access to documentation requirements, coverage rules, and claim-processing procedures that are specific to the payer and insurance. These integrations would need to be authenticated, authorized, validated, logged, and managed with control of the changing rules of the payers.

Additional coding validation may offer more robust validation of the coding and including ICD and procedure code information, as well as inconsistencies between the clinical documentation and submitted codes. Document processing could be enhanced to accept other file formats and scanned documents via optical character recognition (OCR) and then have the extracted document fields verified by humans.

The correctness, consistency and evidence quality of findings – including false alerts and missed issues – should be measured by AI evaluation, and representative, properly governed data should be used. The final coding decision shall continue to be made by an appropriately authorized professional. Other features that could be added are analytics on common denial categories, documentation gaps, coding-related problems, claim-processing stages, and review workloads.

B. Deployment and Maintenance

For a production deployment, you would need secure hosting, identity management, protection of your databases, monitoring and backup. Data exchanges and access control for interfaces with EHR/EMR systems would have to be well-defined. There would have to be controls on prompt and context creation, output validation, logging, and model changes for the AI service. Maintenance should include software updates, security patches, monitoring of AI findings, review of false positives and regular user-feedback sessions. Changes to rules and configurations should be documented in a change log.

C. Proposed Success Measures

The rate at which claims are validated internally before submission, the time taken to resolve claims (time to issue) and the proportion of claims that are reworked and need to be corrected after submission could be measured as a result of future operational evaluation. Other measures involve the distribution of denial categories, frequency and consistency of staff use and the proportion of claims with a current patient facing status. These are suggested evaluation measures and not measures of the current prototype.

D. Business Model Direction

Hospitals, clinics and medical billing service providers are some of the potential customer segments. One option is a subscription-based approach, where fees might be tied to claim volume and/or advanced integrations. Enterprise customers might have a need for integration services, role management, analytics and dedicated support. The price and willingness to pay would need to be validated in market research and pilot deployments.

REFERENCES

[1] Centers for Medicare & Medicaid Services, ICD-10 Codes. Available: https://www.cms.gov/medicare/coding-billing/icd-10-codes

[2] Centers for Medicare & Medicaid Services, “National Correct Coding Initiative (NCCI).” Available: https://www.cms.gov/medicare/coding-billing/national-correct-coding-initiative

[3] Centers for Medicare & Medicaid Services, Medicare Claims Processing Manual. Available: https://www.cms.gov/medicare/regulations-guidance/manuals/internet-only-manuals-ioms

[4] Node.js, “Node.js Documentation.” Available: https://nodejs.org/docs/latest/api/

[5] Groq, “Groq API Documentation.” Available: https://console.groq.com/docs

[6] Vercel, “Vercel Documentation.” Available: https://vercel.com/docs

[7] React, “React Documentation.” Available: https://react.dev/learn

[8] Vite, “Vite Documentation.” Available: https://vite.dev/guide/