POLICY HUB: Knowledge Base

1. Comprehensive Overview

Policy Hub is a scalable, modular, and secure digital platform designed to streamline end-to-end operations for insurance intermediaries, brokers, and their clients. The platform provides a unified environment where users can manage and oversee critical processes across multiple domains. The key difference in this new logic is that new policies are not created manually within Policy Hub; they are generated exclusively by insurance companies. Policy Hub users instead review, update, and finalize these imported policies. In addition, the sales process now includes a robust quote management workflow that drives policy creation by insurers.

Key Domains:

- **Policy Administration:**
  Imported policies from insurance companies are reviewed, updated, and finalized within the system. The platform manages policy addendums (amendments), provides detailed overviews of related data (claims, sales processes, payment/commission information, documents), and enforces a rigorous audit trail.

- **Sales & Lead Management:**
  The sales cycle begins with lead capture and progresses through a new Quote Management stage. Advanced tools such as Kanban boards and funnel diagrams help visualize progress and optimize conversions. Interested leads receive quotes for a given insurance type; these quotes are sent to insurance companies for responses. The broker then forwards the insurer quotes to the potential client, who selects the best option. The selected quote is sent back to the insurer, resulting in a policy that is imported into Policy Hub.

- **Claims Processing:**
  A centralized module where claims are registered, processed, and resolved. Claims automatically pull in policy details from imported policies, ensuring consistency and reducing manual entry.

- **Financial Operations:**
  Integrated modules handle commission calculations, invoicing, unlinked payment reconciliation, and bank statement processing with both automated and manual functionalities. All financial operations now reference policies imported from insurers.

- **Agent Management & Payouts:**
  Supports flexible commission structures (fixed, client-specific, manual overrides) and

provides detailed payout calculations and historical reporting.

- **Reporting & Analytics:**
  Customizable reports across production, clients, agents, and claims deliver actionable insights. Users can filter, sort, and export data for informed decision-making.

- **Localization & Internationalization:**
  The application supports multiple languages (English, Serbian, Macedonian, Spanish) to serve a global user base.

- **Role-Based Access Control:**
  A robust RBAC system ensures that Super Admins, Broker Admins, and Employees see only the features and data relevant to their roles.

The platform's modular architecture and clearly structured sidebar/menu enable intuitive navigation. Robust integrations between modules and external systems ensure data consistency, efficient workflows, and scalability.

---

2. Sidebar/Menu Structure

- **Dashboard**

- **Policies (Policy Management)**

  - All Policies (Policy Directory)

  - Policies Workflow (Policy Processing) – *Review & Finalization of Imported Policies*

  - Policy Addendums (Policy Amendments)

  - Unlinked Payments (Unmatched Payments)

  - Documents (Policy Documents)

- **Sales (Sales & Leads)**

  - Pipeline Overview (Sales Pipeline)

  - Leads (Prospects)

- ○ Sales Processes (Sales Workflow & Quote Management)

- ○ Responsible Persons (Sales Assignments)

- **Claims (Claims Management)**

- **Finances (Financial Operations)**

  - ○ Commissions (Commission Management)

  - ○ Invoicing (Invoice Management)

  - ○ Statement Processing (Bank Statement Reconciliation)

- **Codebook (Master Data)**

  - ○ Clients (Client Directory)

  - ○ Insurance Companies (Insurer Directory)

  - ○ Insurance Codes (Product Codes)

- **Agent (Agent Portal)**

  - ○ Fixed Commissions (Standard Commissions)

  - ○ Client Commissions (Client-specific Commissions)

  - ○ Manual Commissions (Ad-hoc Commissions)

  - ○ Calculate Payouts (Payout Calculation)

  - ○ Payout Reports (Payout History)

- **Reports (Analytics & Reporting)**

  - ○ Production Report (Policy Production Report)

  - ○ Clients Report (Client Activity Report)

  - ○ Agents Report (Agent Performance Report)

  - ○ Claims Report (Claim Summary Report)

- **Settings (Administration)**

Feature	User Story	Acceptance Criteria	BDD Scenario
Navigation Structure	As a user, I want a consistent sidebar structure, so that I can easily access different modules.	- Sidebar must contain all top-level modules as defined. - Nested structure matches described hierarchy. - Each menu item routes to correct view.	Given I view the sidebarWhen I click on a menu itemThen I am routed to the correct screen.
Role-Based Visibility	As an admin, I want sidebar items to show/hide based on user roles.	- Items are dynamically rendered per user role. - Super Admin sees all; Employee sees only authorized modules.	Given I log in as an employeeWhen I view the sidebarThen only authorized items appear.

3. Detailed Feature Descriptions

3.1 Dashboard

Purpose:
The landing page presents key performance indicators (KPIs) and summary information for quick decision-making.

Key Widgets & Functionality:

- **Upcoming Policies:**

  - **Data:** Display policies (imported from insurers) expiring within the next 40 days with direct links to review them.
  - **Display:** Count or list with a link to the Policy Directory.

- ○ **Action:** Clicking filters policies for renewal review.

- ● **Incomplete Policies:**

  - ○ Show count/preview of policies missing mandatory fields (e.g., insurer data, documentation, commission details), with links to the Policy Processing module for review and finalization.

- ● **Ready-to-Input Policies:**

  - ○ List policies with complete imported data pending final confirmation.
  - ○ **Display:** With a "Finalize" action.

- ● **Open Claims:**

  - ○ Summary of active and unresolved claims with links to detailed views.

- ● **Sales Pipeline Overview:**

  - ○ Visual display (e.g., Kanban or funnel) summarizing lead conversion and quote management stages, enabling drill-down into individual sales processes details.

Development Considerations:

- ● Create aggregated API endpoints for multi-module data.

- ● Implement caching for frequently accessed dashboard data.

- ● Dynamically render widgets based on the user's role.

Feature	User Story	Acceptance Criteria	BDD Scenario
Widget Visibility	As a user, I want dashboard widgets to show only what's relevant to my role.	- Dashboard adjusts widgets based on user role (e.g., claims not visible to sales rep).	**Given** I am a logged-in user**When** I load the dashboard**Then** I only see widgets matching my permissions.

Performance & Data Sync	As a product owner, I want dashboards to load quickly and show real-time summaries.	- Data caching used for summary values.- API endpoints provide aggregated data.- Widget refresh in <2 seconds.	**Given** the dashboard loads**When** widgets fetch their data**Then** they load within 2 seconds and show accurate counts.

3.2 Policies

Manages the entire lifecycle of policy records—now exclusively imported from insurance companies. Policy Hub users review, update, and finalize these policies.

3.2.1 All Policies (Policy Directory)

Overview:
 A comprehensive directory of all imported policies—active, expired, or incomplete—with robust search, detailed policy management, document handling, and audit capabilities. For any selected policy, the system displays a detailed dashboard of related data (claims, sales processes, policy addendums, payment information).

Key Data Fields:

- **Policy Number**

- **Policy Type** (Internal/External, as provided by the insurer)

- **Start Date and Expiry Date**

- **Policyholder & Insured Details** (sourced from the Codebook)

- **Insurer & Product** (sourced from the Codebook)

- **Financial Fields:**

    - Premium Amount (in foreign currency and in RSD)

    - Currency (RSD, EUR, USD)

- ○ Payment Frequency

- **Commission Settings:**

  - ○ Manual vs. Automatic

  - ○ Commission Percentage

- **Internal Notes** for additional remarks

- **Additional Attributes:**

  - ○ Lien (Vinkulacija) status

  - ○ Agent Assignment (with Agent Type: Internal, Employee, or Partner)

  - ○ Responsible Person

Features:

- **Search/Filter:**

  - ○ Search by policy number, status (In Progress, Expired, Incomplete), policyholder, insured, insurer, responsible person, insurance type, and product.

  - ○ Filter by date range (start and expiry), premium range, and commission status.

- **Sorting & Pagination:**

  - ○ Options to sort policies by start date, expiry, policy number, premium, etc.

  - ○ Pagination options (10, 25, 50, 100 items per page).

- **Policy Management (Review & Editing):**

  - ○ **Import & Review:**

    - ■ New policies are imported from insurance companies (manual creation of policy is disabled).

    - ■ Users review imported data with **real-time validation** on the review form.

  - ○ **Document Upload:**

- ■ Attach supporting files (e.g., policy documents, related financial documents).

- ○ **Review & Finalization:**

  - ■ After reviewing, users validate all mandatory fields and **mark the policy as complete** (finalize the record).

- ○ **Editing/Deletion:**

  - ■ Update or delete existing policy records if permitted by RBAC.

- ● **Detailed Overview of Related Data:**

  - ○ Upon selecting a policy, display a dashboard-like view consolidating:

    - ■ Claims overview

    - ■ Sales process details (including quote management stages)

    - ■ Policy addendums

    - ■ Payment and commission information

    - ■ Document attachments

- ● **Audit Trail:**

  - ○ Maintain a detailed change log recording every modification (user ID, timestamp, IP address, and change description).

Feature	User Story	Acceptance Criteria	BDD Scenario

Search & Filter Policies	As an employee, I want to search and filter policies by multiple criteria (e.g., policy number, status, insurer), so that I can quickly find the relevant records.	- System shall allow filtering by policy number, status, policyholder, insured, insurer, insurance type, and product.- Filters include date, premium, commission.- Results returned within 2s.- Clear all filters with one click.	**Given** I am on the Policy Directory page**When** I apply filters**Then** I see relevant policies and can reset view with 'Clear Filters'.
Policy Review & Finalization	As a broker employee, I want to review and finalize an imported policy after completing all required fields.	- System validates required fields.- Finalize button only appears when fields are valid.- Status updated and audit log entry created.	**Given** I review a policy**When** I finalize it**Then** it is marked 'Completed' and the action is logged.
Audit Logging	As an admin, I want all policy modifications to be logged, so I can audit changes.	- Each change logs user ID, IP, timestamp, and action.- Logs are accessible and exportable per policy.	**Given** a policy is updated**When** I view the log**Then** I see all modifications listed with metadata.

3.2.2 Policy Addendums (Policy Amendments)

Purpose:
To manage modifications to policies (coverage extensions, premium adjustments, commission recalculations) by linking amendments to their parent policies and updating related financial and operational details.

Key Data Fields:

- **Addendum Number:** Unique identifier.

- **Policy Type:** Inherited from the parent policy.

- **Parent Policy Number:** The associated policy (imported from insurer).

- **New Effective Dates:** Updated start and expiry dates.

- **Party Details:** Policyholder, Insured, Insurer (from parent policy; overridable if needed).

- **Insurance Product Details:** Category (Non-life/Life), Insurance Type, Product (from Codebook).

- **Agent Assignment:** (Employee/Partner) handling the amendment.

- **Payment Dynamics:** Options for full, monthly, quarterly, semi-annual, or annual payment schedules for the adjusted premium.

- **Premium Adjustments:** New premium amount (in foreign currency and RSD).

- **Commission Adjustments:**

    - Commission Calculation Method (Automatic/Manual)

    - Updated Commission Percentage and recalculated Commission Amount.

- **Lien Status:** (Yes/No)

- **Internal Notes:** Remarks regarding the amendment.

Workflow & Features:

- **Linking:**

    - Associate the addendum with its parent policy via form-based entry (parent policy imported from insurer).

    - Provide comprehensive search and filtering (by policy number, responsible person, policyholder, insurer, category, insurance type, product).

- **Data Entry & Validation:**

    - Form-based entry with **real-time validation** to ensure all mandatory fields are completed.

    - Allow updates/modifications to key fields.

- **Document Attachment:**

  - Upload supporting documents (e.g., scanned addendum files) with options for document type (Aneks, Policy, Invoice, Lien, Notification, Miscellaneous) and document description.

- **Review & Finalization:**

  - Validate all mandatory fields and mark the addendum as complete (finalized), fully integrating it with the parent policy.

- **Audit Logging:**

  - Record every modification with a change log (user ID, timestamp, IP address, change description).

Actions:

- "Save Changes", "Add Addendum", "Delete Addendum"

- Option to view the related Sales Process of the parent policy

- Export addendum data (if required)

Feature	User Story	Acceptance Criteria	BDD Scenario
Addendum Linking	As an employee, I want to link an addendum to an existing policy.	- Form allows policy selection and pre-fills inherited data.- Valid parent policy required.- Can override specific fields.	**Given** I create an addendum**When** I link it to a policy**Then** inherited fields are auto-filled and editable.
Addendum Finalization	As a user, I want to finalize an addendum after	- Required fields validated before save.-	**Given** I finalize an addendum**When** all fields are valid**Then** the status

	filling all mandatory fields.	Finalize triggers status change and logging.	becomes complete and audit log is updated.
Audit Trail	As an admin, I want to audit addendum modifications.	- Each change recorded with user, time, IP.- Viewable within addendum and exportable.	**Given** I access an addendum**When** I open audit log**Then** all modifications are shown in sequence.

3.2.3 Unlinked Payments (Unmatched Payments)

Purpose:
 To manage payments that have not been automatically matched to an imported policy, allowing users to search, review, and manually link these payments.

Key Data Fields:

- **Insurance Company:** Selected from a dropdown or input field.

- **Policyholder (Ugovarač):** Input field.

- **Payment Reference/Number:** Input field.

- **Base Amount (Osnovica)**

- **Commission Percentage (Provizija %)**

- **Payment Amount (Iznos)**

- **Payment Date (Datum)**

- **Invoice Reference (Faktura)**

Features:

- **Search & Filtering:**

- ○ Filter and search payments by insurance company, policyholder, and payment reference/number.

- **Display:**

  - ○ List unlinked payments with detailed fields: base amount, commission percentage, payment amount, payment date, invoice reference.

- **Manual Linking:**

  - ○ Allow users to manually assign a payment to a policy.

  - ○ Provide a separate "Policy Selection" step where users choose the appropriate policy (via input or dropdown).

  - ○ After linking, display a consolidated view ("Pregled uplata za izabranu polisu") of all payments linked to that policy.

- **Actions:**

  - ○ "Dodaj uplatu" button to link the selected payment.

  - ○ "Preuzmi XLS" option to export the list of unlinked payments for reconciliation.

- **Audit Logging:**

  - ○ Record changes related to payment linking (date, IP address, change description).

Feature	User Story	Acceptance Criteria	BDD Scenario
Manual Linking	As a finance specialist, I want to manually link unmatched payments to policies.	- Policy selection validated.- Linked payment appears in policy's payment view.- Action logged.	**Given** I assign a payment**When** I link it to a policy**Then** it becomes visible in the payment overview.

Export Payments	As a user, I want to export unmatched payments for reconciliation.	- Export includes filtered payments.- Outputs XLS file with all displayed fields.	**Given** I filter unlinked payments**When** I click export**Then** I receive an XLS with correct fields.
Audit Logging	As an admin, I want to audit payment-linking activity.	- Linking actions are logged (who, what, when, IP).- Logs viewable and exportable.	**Given** I review payment audit log**When** a manual link was made**Then** I see the user ID and time of the action.

3.2.4 Documents (Policy Documents)

Purpose:
 To serve as a centralized repository for all files related to policies, enabling easy upload, retrieval, and version tracking.

Key Functionalities:

- **File Upload:**

    - Allow single or bulk uploads of policy-related documents.

    - Support multiple file formats (e.g., PDF, DOCX, JPG, PNG).

- **Search & Filtering:**

    - Search by file name.

    - Filter by document type (Policy, Invoice, Lien, Notification, Miscellaneous).

    - Filter by associated policy (using the policy number or identifier).

- **Version Control (Optional):**

    - Track document revisions, allowing users to view or revert to previous versions.

Actions:

- "Upload Document(s)"

- "Delete Document"

- "View Document Versions" (if enabled)

Feature	User Story	Acceptance Criteria	BDD Scenario
Upload Documents	As a user, I want to upload and categorize policy-related files.	- Supports multiple formats.- Document type and description required.- Stored with metadata (user, time).	**Given** I upload a document**When** I assign its type**Then** it is stored and visible with metadata.
Document Search	As a user, I want to search documents by name, type, or policy.	- Filter/search options must return results within 2s.- Includes document type and associated policy filter.	**Given** I search by file name**When** I apply filters**Then** matching files are listed.
Version Control	As a user, I want to manage document versions.	- System keeps version history.- Older versions viewable/downloadable.- Allow reversion if enabled.	**Given** I upload a new version**When** I view document history**Then** I can access previous versions or revert.

3.3 Sales (Sales & Leads)

Handles the entire sales cycle from lead capture, through quote management, to policy finalization.

3.3.1 Pipeline Overview (Sales Pipeline)

Purpose:
To provide an overview of current sales activities.

Features:

- **Visual Layout:** Options for Kanban boards or funnel diagrams.

- **Data Display:** Total leads, stage-wise breakdown, estimated revenue.

- **Drill-Down:** Clickable elements for detailed views.

Feature	User Story	Acceptance Criteria	BDD Scenario
Sales Pipeline Visualization	As a sales manager, I want to view leads in a visual format (Kanban/funnel), so I can track progress.	- Displays lead stages.- Lead cards show company/contact, value, and current stage.- Click to open lead detail.	**Given** I view the pipeline**When** I click a stage**Then** I see all leads in that stage and can access their details.

3.3.2 Leads (Prospects)

Key Data Fields:

- Lead Name/Company, Contact Person, Phone, Email, Source.

- Status: New, Qualified, Converted, Lost.

- Assignment: Linked to a Sales Representative.

Features:

- **Creation & Editing:** Simple forms to add new leads.

- **Conversion Process:** Workflow to convert leads into sales processes.

- **Follow-Up:** Set reminders and next steps.

Feature	User Story	Acceptance Criteria	BDD Scenario
Lead Creation	As a sales rep, I want to capture new leads.	- Required: company, contact person, phone/email.- Select lead source.- Assigned to sales rep.	**Given** I create a new lead**When** I enter contact and source**Then** the lead is saved and visible in pipeline.
Assignment	As a sales manager, I want to assign leads to team members.	- Assignment dropdown populated from active users.- Logged and visible in lead history.	**Given** I reassign a lead**When** I save**Then** the new owner is notified and audit updated.

3.3.3 Sales Processes (Sales Workflow & Quote Management)

Purpose:
To structure the multi-step process from initial client interest to finalized policy.
Quote Management:

- **Quote Generation:** When a lead expresses interest, the broker generates a quote for a specific insurance type.

- **Insurer Quote Response:** The generated quote is sent to multiple insurance companies.

- **Client Selection:** The broker reviews insurer responses, forwards the quotes to the client, and records the client's choice.

- **Policy Import:** Once the client selects a quote, the broker sends the chosen quote back to the insurance company, which then creates the policy. The policy is imported into

Policy Hub for review and finalization.

Workflow Steps:

- **Pre-Process:** Capture initial details; assign a sales number.

- **Quote Management:**

    - Generate and send quotes to insurance companies.

    - Collect insurer responses and forward to the client.

    - Record the client's selection and notify the insurer.

- **Authorization:** Upload client authorization forms.

- **Client Requests:** Record specific insurance requirements.

- **Insurance Offers:** Upload and compare insurer proposals.

- **Policy Proposal:** Finalize and present the proposal.

- **Policy Receipt:** Confirm issuance from the insurer.

- **Signed Policies:** Upload the finalized signed policy (imported from the insurer).

- **Conclusion:** Mark the process as successful or unsuccessful.

Features:

- **Document Management:** Attach files at each step.

- **Notifications:** Automated reminders for pending steps.

- **Audit Logging:** Track each transition and user action.

Feature	User Story	Acceptance Criteria	BDD Scenario

Quote Generation	As a broker, I want to send a quote request to insurers.	- Quotes contain insurance type and client preferences.- Send to one or more insurers.- Logged.	**Given** I create a quote**When** I choose insurers**Then** requests are sent and listed in the quote history.
Client Selection	As a broker, I want to record which insurer offer the client accepts.	- List insurer responses.- Select client choice.- Trigger policy import.	**Given** I receive offers**When** client selects one**Then** I notify the insurer and import begins.
Sales Audit	As an admin, I want to audit all sales process steps.	- Every stage (quote, client selection, policy) logged.- Viewable by admins.	**Given** I view a sales record**When** I open the audit log**Then** all sales actions are displayed chronologically.

3.3.4 Responsible Persons (Sales Assignments)

Purpose:
 To assign and manage accountability for leads and sales processes.

Features:

- **User Selection:** Dropdown or auto-suggest for available employees.

- **Reassignment:** Easily change responsibility.

- **Filtering:** View assignments by team, status, or role.

Feature	User Story	Acceptance Criteria	BDD Scenario

Sales Pipeline Visualization	As a sales manager, I want to view leads in a visual format (Kanban/funnel) to assess progress.	- Display stages visually.- Click to drill into lead.- Shows lead count, value, and stage.	**Given** I open Sales Pipeline**When** I click a funnel stage**Then** I see associated leads with details.
Lead Capture & Assignment	As a sales rep, I want to create a lead and assign it, so it's tracked.	- Must input lead name, contact, source.- Assignment dropdown required.- Notification sent to assignee.	**Given** I add a new lead**When** I assign it to a rep**Then** it appears in their pipeline and they are notified.
Quote Management Workflow	As a broker, I want to manage the quote process end-to-end.	- Broker sends quote to insurers.- Insurer replies logged.- Client choice recorded.- Policy imported and linked.	**Given** I handle a quote**When** the client picks an offer**Then** a new policy is created and imported.
Document Management	As a broker, I want to upload files at each sales step.	- Each step supports document upload.- Type and description fields required.- Files viewable in sales process.	**Given** I reach a new sales stage**When** I upload documents**Then** they are saved and shown in the process timeline.
Audit Logging	As an admin, I want to track every change in a sales process.	- All updates/actions are logged with timestamp, user, IP.- Logs viewable per sales process and exportable.	**Given** I review a closed sale**When** I open audit trail**Then** I see all actions with timestamps and details.

Sales Assignment	As a sales manager, I want to assign a responsible person to a lead or sale.	- Dropdown lists available employees.- Change recorded in audit log.- Reassignment available.	**Given** I edit a lead**When** I change responsible person**Then** the update is saved and logged.
Filter by Responsible	As a user, I want to filter sales by assigned person.	- Filter shows leads and sales grouped by assignee.- Default shows "mine".	**Given** I apply a filter**When** I choose a colleague**Then** I only see their assigned sales.

3.4 Claims (Claims Management)

Overview:
 Manages the complete lifecycle of claims, with automatic population of policy details when filing a claim. (Note: Policies are imported from insurers.)

Key Data Fields:

- **Claim ID**

- **Associated Policy:** Automatically populated (insured, insurer, policy type, dates, premium info) upon selection.

- **Damage Description**

- **Location**

- **Date of Damage & Date of Claim Submission**

- **Claimed Amount vs. Approved Amount**

- (Optional: internal claim number, insurer claim number, deductible details, rejection reason, notes)

Status:

- In Processing, Reported, Accepted, Rejected, Partially Accepted, Appealed, Withdrawn

Features:

- **Registration:** Form-based entry for new claims; automatically populate policy details upon policy selection.

- **Status Management:** Buttons or dropdowns to update claim status.

- **Document Upload:** Attach photos, reports, and other supporting files.

- **Audit Trail:** Log every update (timestamp, user details, IP address).

- **Search/Filter:** By policy number, claim ID, status, and date range.

Feature	User Story	Acceptance Criteria	BDD Scenario
Claim Registration	As a user, I want to register claims and prefill policy data.	- Policy data auto-filled.- Required: claim ID, date, description, amount.- Date validation.- Unique claim ID generated.	**Given** I start a new claim**When** I select a policy and complete the form**Then** the claim is saved and shown in system.
Status Management	As a claims adjuster, I want to update status.	- Status list includes: In Processing, Accepted, Rejected, etc.- Status change requires a comment.- Change is logged.	**Given** I update a claim**When** I select a new status**Then** the claim reflects that and audit log is updated.

Document Upload	As a user, I want to upload supporting files for claims.	- File types: image, PDF, doc.- Must associate with claim.- Metadata: uploader, date, type.	**Given** I open a claim**When** I upload documents**Then** they appear under that claim.
Search & Filtering	As a user, I want to search/filter claims.	- Filter by policy, ID, date, status.- Response time <2s.- Results paginated.	**Given** I filter by claim status**When** I click search**Then** I see matching results.
Claim Audit Trail	As an admin, I want to audit changes to claim records.	- All field changes and status updates logged.- Exportable in .csv format.	**Given** I review a claim**When** I view its audit trail**Then** I see user actions with timestamp and values.

3.5 Finances (Financial Operations)

Handles commission calculations, invoicing, and bank statement reconciliation.

3.5.1 Commissions (Commission Management)

Key Data Fields:

- Commission Rate

- Premium Amount

- Calculated Commission

- Status (Due, Paid, etc.)

Features:

- **Automated Calculations:** Triggered by policy import or updates to automatically compute commissions.

- **Manual Adjustments:** Allow overrides for special cases via manual inputs.

- **Filtering & Reporting:**

  - Filter by agent, policy, or period.

  - Date Filtering: "Datum od" to "Datum do".

  - Status Filtering: Options such as In Progress, Processed, Confirmed, Invoiced.

  - Insurance Company Filter: Filter commissions by the associated insurance company
  - Calculation Document Filter: Option to filter based on calculation document details.

- **Document Management:**

  - Template Download for a calculation template.

  - File Upload:

    - Original Calculation Document (PDF).

    - Processed Calculation Document (.csv).

- **Notes Field:** Input area for notes regarding commission calculations.

- **Processing Workflow:**

  - Actions for "Process Calculation" and "Confirm Calculation", so that calculations are moved through their lifecycle

- **Audit Trail:** Detailed change log (timestamp, IP, description).

Feature	User Story	Acceptance Criteria	BDD Scenario

Commission Calculation	As a finance specialist, I want commissions calculated automatically when a policy is imported or updated.	- Uses premium and rate fields.- Calculates both percentage and fixed commissions.- Only active policies are included.	**Given** a policy is imported**When** it includes commission fields**Then** a commission record is calculated and stored.
Manual Override	As a finance user, I want to override calculated commissions for special cases.	- Requires justification comment.- Only allowed for users with override permission.- Change logged in audit trail.	**Given** I override a commission**When** I enter a reason and save**Then** the new amount is recorded and logged.
Status Tracking	As a user, I want to track commission status (due, paid, confirmed).	- Status options selectable.- Filterable by status in commission table.	**Given** I update commission status**When** I mark it as 'Confirmed'**Then** it reflects in list view and filters.

3.5.2 Invoicing (Invoice Management)

Key Data Fields:

- Invoice Number

- Issue Date

- Due Date

- Currency (RSD, EUR, USD)

- Invoice Type: Domestic/Foreign

- Invoice Category: Automatic (Obračunska)/Manual (Manuelna)

- Calculation Reference Number (Broj obračuna)

- Calculation Method (e.g., by policy number or by insurance category)

- Billing Period (Obračunski period)

- Itemized Line Items (including additional items)

- Status: In Progress, Confirmed, Partially Paid, Fully Paid

- Notes: Total amount in words, supplementary notes, payment instructions

Features:

- **Invoice Generation:** Manual or auto-generated from commission settlements.

- **Search/Filter:** By date range, invoice status, invoice type, and insurance company.

- **Document Management:**

  - Template Download, Document Upload (scanned invoice copies, PDF).

- **Invoice Actions:**

  - Buttons: "Save Changes", "Create Invoice", "Delete Invoice", "Print Invoice", "Confirm Invoice".

- **Payment Tracking:** Update status based on received payments.

- **Itemized Invoice Lines:** Add extra line items with description and amount.

- **Audit Trail:** Detailed change log (timestamp, IP, description)

- **Notes & Instructions:** Fields for total amount in words, supplementary notes and payment instructions.

Feature	User Story	Acceptance Criteria	BDD Scenario

Invoice Generation	As a finance user, I want to generate invoices based on commission settlements.	- Must allow both manual and automatic generation.- Invoice includes reference numbers, date, billing period.- PDF template available.	**Given** I confirm commission data**When** I click 'Create Invoice'**Then** the invoice is generated and visible with status 'In Progress'.
Edit & Confirm Invoice	As a finance user, I want to finalize invoice details before confirming.	- All fields editable before confirmation.- Status changes from 'In Progress' to 'Confirmed' upon save.	**Given** I edit invoice line items**When** I click Confirm**Then** the invoice is locked for changes and marked confirmed.
Export & Print	As a finance user, I want to export and print invoices.	- Buttons to download PDF.- Export log entry added.	**Given** I view a confirmed invoice**When** I click 'Print'**Then** a PDF version is downloaded.

3.5.3 Statement Processing (Bank Statement Reconciliation)

Purpose:
To reconcile bank statements with internal financial records by importing statement files, matching transactions, and allowing manual adjustments.

Key Data Fields:

- Statement Number

- Status (In Progress, Processed, Confirmed)

- Bank (e.g., UNICREDIT, KOMBANK)

- Date

- Document – PDF version

- Document – TXT version

- Note

Features:

- **Import & Parsing:** Accept PDF and TXT statement files.

- **Filtering & Search:** By date range, statement status, and bank.

- **Document Management:** Actions to "Add Statement," "Delete Statement," "Save Changes."

- **Processing Workflow:** Buttons for "Process Statement" and "Confirm Statement."

- **Matching Algorithm:** Auto-match transactions with invoice or policy references.

- **Manual Linking:** Option to manually assign unmatched transactions.

- **Audit Logging:** Detailed change log (date, IP, description).

Feature	User Story	Acceptance Criteria	BDD Scenario
Import Bank Statement	As a finance user, I want to upload and parse bank statements.	- File types: PDF and TXT.- Each line item parsed into transactions.- Validations applied.	**Given** I upload a statement file**When** parsing completes**Then** the transaction list appears.
Auto & Manual Matching	As a user, I want system to match payments to invoices automatically or manually.	- Auto-match uses invoice or policy reference.- Manual match allows user selection.- Matches logged.	**Given** transactions are imported**When** matches are found**Then** linked invoices are updated.

Statement Audit Trail	As an admin, I want to see all statement-related actions in logs.	- Changes to status (Processed, Confirmed) logged.- User, time, and action recorded.	**Given** I confirm a statement**When** I view logs**Then** I see a list of all related operations.
Commission Calculation	As a finance specialist, I want commissions calculated automatically.	- Based on premium and rate.- Triggered on policy import/update.- Override possible with justification.	**Given** a policy is imported**When** it includes commission settings**Then** the system calculates it and stores the record.
Manual Commission Override	As a finance user, I want to override commissions in exceptions.	- Requires justification.- Permission check before override.- Action is logged.	**Given** I override a commission**When** I save the update**Then** it is applied and audit trail is updated.
Invoice Generation	As a finance user, I want to create invoices from commissions.	- Pulls confirmed commission data.- Supports manual or auto modes.- Must allow edits before confirming.	**Given** I select commission items**When** I click 'Create Invoice'**Then** an editable invoice is generated.
Bank Reconciliation	As a finance user, I want to reconcile bank statements.	- Import PDF/TXT formats.- Auto-matches known invoices/policies.- Manual matching option required.	**Given** I import a statement**When** transactions are found**Then** the system matches or flags them.
Financial Audit Logging	As an auditor, I want to view all	- All commissions, invoices, and transactions	**Given** I access a financial record**When** I open audit

| financial operation history. | logged.- Logs filterable and exportable. | logs**Then** all changes and actions are shown. |

3.6 Codebook (Master Data)

Provides centralized reference data.

3.6.1 Clients (Client Directory)

Key Data Fields:

- Client Type (Legal Entity/Individual; indicates Partner status)

- Business Details: Short Name, Full Business Name, Tax ID (PIB), Registration Number, Current Account, Responsible Person

- Address Information: Primary Address (Address, City, Postal Code); Mailing Address (Address 2, City 2, Postal Code 2)

- Agent Type: Options (Internal, Employee, Partner)

- Authorization Document: Option to generate and attach a signed authorization

- Contact Persons: Name, Position, Phone, Mobile, Email, Username, Password

Features:

- **Search & Filtering:** Search bar, Active/Inactive filters, Client Type filter.

- **Client Management:** Form-based entry to create, edit, and deactivate client profiles including all key data fields.

- **Contact Management:** Add, edit, delete multiple contact persons.

- **Audit Trail:** Detailed change log (timestamp, IP address and description)

Feature	User Story	Acceptance Criteria	BDD Scenario
Client Directory Management	As an admin, I want to manage client records.	- Required fields: business name, tax ID, registration number, address, contacts.- Status toggle (active/inactive).- Type: individual or legal entity.	**Given** I create a client**When** I complete the form**Then** the client appears in the list and is available in policy creation.
Contact Management	As a user, I want to add and manage multiple contact persons per client.	- Name, role, phone, email are required.- Editable and deletable.	**Given** I add a contact to a client**When** I save it**Then** the contact appears under the client and is editable.
Client Audit Logging	As an auditor, I want to track all changes to client data.	- Each change logs timestamp, user ID, and IP address.- Log export available.	**Given** a client is updated**When** I view logs**Then** I see all field-level changes listed chronologically.

3.6.2 Insurance Companies (Insurer Directory)

Key Data Fields:

- Business Details: Short Business Name, Full Business Name, Parent Company, Tax ID (PIB), Registration Number, Current Account, Broker Code, Active Status

- Address Details: Headquarters Address, City, Postal Code; Mailing Address (Address 2, City 2, Postal Code 2); Responsible Person

- Contact Persons: Name, Function, Phone, Mobile, Email

- Client Request Emails: Multiple emails separated by semicolon

● Audit Trail: Change log record (timestamp, IP address, description)

Features:

● **Search & Filtering:** By insurer name, Active status, insurer type (Parent vs. Branch).

● **Record Management:** Create, edit, delete insurer records.
  ○ Update registration details, addresses, and responsible person information

● **Contact Management:** Add, edit, delete contact persons associated with each insurer

● **Audit Logging:** Detailed change log (timestamp, ID address, description_

● **Actions:** "Save Changes", "Add Insurer", "Delete Insurer."

Feature	User Story	Acceptance Criteria	BDD Scenario
Insurer Directory	As an admin, I want to manage insurer records.	- Required fields: name, parent company, tax ID, broker code, contact.- Editable.- Must include at least one email.	**Given** I add a new insurer**When** I submit the form**Then** the insurer is available in policy forms and sales.
Contact Persons	As a user, I want to manage insurer contacts separately.	- Multiple contacts per insurer.- Editable independently.- Must include phone and email.	**Given** I edit an insurer**When** I update contacts**Then** changes are saved and visible under the record.
Audit Trail	As an admin, I want to track edits to insurers.	- Logs user, IP, timestamp, and fields changed.- Exportable.	**Given** I review insurer logs**When** I export**Then** I receive a CSV with edit history.

3.6.3 Insurance Codes (Product Codes)

Key Data Fields:

- Code (Šifra): Unique identifier for the insurance product

- Category: Dropdown selection with following options: Life, Non-life

- Active Status: Boolean toggle to mark the code as active or inactive

- Insurance Name (Naziv osiguranja): The product name in the primary language

- English Insurance Name (Engleski naziv osiguranja): The product name in English

Features:

- **Search & Filtering:** Search bar to query by insurance code or name, filter by active status and category.

- **Record Management:** Create new, edti and update existing, and delete insurance codes.

- **Multilingual Support:** Manage primary and English names for each insurance product.

- **Audit Trail:** Change log (timestamp, IP address, description)

Feature	User Story	Acceptance Criteria	BDD Scenario
Product Code Management	As a manager, I want to define codes for each insurance product.	- Code must be unique.- Each entry has category (Life/Non-Life), local and English names.- Multilingual support required.	**Given** I add a new product code**When** I save**Then** it is listed with the proper language fields.

Search & Filter	As a user, I want to filter product codes.	- Search by name or code.- Filter by category and status (active/inactive).	**Given** I enter a keyword**When** I filter by Non-Life**Then** only matching active codes are shown.

Contact Person Management	As a user, I want to manage contact persons for both clients and insurers.	- Name, function, phone, mobile, email required.- Linked to parent entity (client or insurer).	**Given** I add a new contact**When** I associate it with an insurer**Then** it appears in the insurer's contact list.

Codebook Change Logs	As an admin, I want to view and export audit logs for all Codebook changes.	- Every change tracked (add, edit, delete).- Log includes field changed, user, and time.- Export as CSV.	**Given** a record is edited**When** I open audit history**Then** I see detailed logs for that entity.

Client Directory Management	As an admin, I want to manage client records.	- Required fields: business name, tax ID, address, contact person.- Status toggle (active/inactive).	**Given** I create a client**When** I submit the form**Then** the client appears in the directory.
Insurer Directory	As an admin, I want to manage insurer records for accurate quoting.	- Must support contact persons and multiple email addresses.- Editable and searchable.	**Given** I view an insurer record**When** I update contact

			info**Then** changes are saved and logged.
Product Codes Management	As a manager, I want to manage product codes.	- Each product has a unique code and multilingual name.- Active/inactive toggle.- Categorized as life/non-life.	**Given** I add a new insurance code**When** I submit the form**Then** it appears in the active code list.
Contact Person Management	As a user, I want to manage contact persons for clients/insurers.	- Add, edit, remove multiple contacts.- Fields include name, position, email, phone.	**Given** I access a client**When** I add a contact**Then** they are shown under the client and saved.
Codebook Audit Logging	As an admin, I want to track changes to master data.	- Every update/delete logs user ID, timestamp, IP.- Logs are exportable per entity.	**Given** I update an insurer**When** I view audit log**Then** the change appears with metadata.

3.7 Agent (Agent Portal)

Focuses on agent-specific commission management and payouts.

3.7.1 Fixed Commissions (Standard Commissions)

Features:

- Set fixed commission percentages per agent with effective dates.

- Filter, update, and manage commission settings.

Feature	User Story	Acceptance Criteria	BDD Scenario
Fixed Commission Setup	As an admin, I want to define fixed commission rates per agent.	- Set by agent, product, and effective date.- Only one active fixed rate per agent/product.- Historical rates archived and viewable.	**Given** I set a new fixed rate**When** I submit the form**Then** the rate is saved and replaces any prior active rate.

3.7.2 Client Commissions (Client-specific Commissions)

Features:

- Define custom commission rates for specific agent-client pairs.

- Override fixed commissions where applicable.

Feature	User Story	Acceptance Criteria	BDD Scenario
Client-Based Overrides	As an admin, I want to override fixed rates with client-specific ones.	- Overrides applied when both agent and client match.- Takes priority over fixed rates.- Editable and logged.	**Given** I define a client override**When** the client and agent match in a policy**Then** the override commission is applied.

3.7.3 Manual Commissions (Ad-hoc Commissions)

Features:

- Allow manual overrides on a per-policy basis with justification.

- Record adjustments and track historical changes.

Feature	User Story	Acceptance Criteria	BDD Scenario
Manual Override	As a finance user, I want to override commission values manually for specific cases.	- Justification required.- Field editable only with permission.- All changes logged.	**Given** I open a commission record**When** I manually change the amount**Then** it is saved with my reason and logged.

3.7.4 Calculate Payouts (Payout Calculation)

Features:

- Aggregate commissions due to each agent over a defined period.

- Finalize payouts and mark as paid.

Feature	User Story	Acceptance Criteria	BDD Scenario
Calculate Agent Payouts	As a finance user, I want to calculate total payouts due to each agent.	- Calculates by agent and period.- Uses confirmed commission records only.- Exportable summary view.	**Given** I choose a date range**When** I run payout calculation**Then** I see totals grouped by agent.

3.7.5 Payout Reports (Payout History)

Features:

- Provide historical payout records, filterable by agent, date range, and status.

- Export Option: XLS download.

Feature	User Story	Acceptance Criteria	BDD Scenario
Payout Report History	As a user, I want to view and export historical agent payouts.	- Filterable by date, status, agent name.- Downloadable as XLS.- Includes status (Paid, Pending).	**Given** I filter for Paid payouts**When** I export the report**Then** I receive a file with all matching rows.

Fixed Commission Setup	As an admin, I want to define fixed commission rates.	- Must assign rate, agent, effective date.- Only one active rate per agent.- Past rates are archived.	**Given** I define a new fixed rate**When** I assign it to an agent**Then** it's active and older ones archived.
Client-Specific Commission	As an admin, I want to override commission by client-agent pair.	- Overrides take precedence over fixed.- Applies only to associated client.	**Given** I define a client-specific rate**When** a policy is processed**Then** override is applied.
Manual Commission Override	As a finance user, I want to override commissions manually.	- Requires justification.- Only users with permissions allowed.- Changes logged.	**Given** I override a commission**When** I save with reason**Then** value changes and log updates.
Calculate Agent Payouts	As a finance user, I want to calculate	- Aggregates confirmed commissions.- Payout summary includes agent	**Given** I select a date range**When** I

	agent payouts over a period.	name, amount.- Filterable by period.	calculate**Then** I see payouts per agent.
Payout Reports	As a user, I want to view/export agent payout reports.	- Filter by date, status, agent.- Supports XLS export with complete data.	**Given** I apply filters**When** I export**Then** the XLS file includes the matching payout records.

3.8 Reports (Analytics & Reporting)

Generates and exports reports based on data from all modules.

3.8.1 Production Report (Policy Production Report)

Features:

- Filters: Client, Insurer, Insurance Type, Agent, Date Range, Commission Status.

- Data: Policy number, premium, commission details, collected amounts.

- Export: XLS.

Feature	User Story	Acceptance Criteria	BDD Scenario
Production Reporting	As a manager, I want to see production metrics including policies and commissions.	- Filters: client, insurer, insurance type, agent, date range.- Shows: policy number, premium, commission, collected amounts.- Exportable to XLS.	**Given** I apply filters to the report**When** I click generate**Then** matching production data is shown and can be exported.

3.8.2 Clients Report (Client Activity Report)

Features:

- Aggregates metrics (e.g., number of policies per client, claim counts).

- Export: XLS

Feature	User Story	Acceptance Criteria	BDD Scenario
Client Activity Report	As an admin, I want to track client activity and history.	- Shows number of policies, claims, and overall engagement per client.- Filterable by active/inactive status.- Exportable.	**Given** I select active clients**When** I run the report**Then** I see aggregated metrics for each client.

3.8.3 Agents Report (Agent Performance Report)

Features:

- Summarizes agent performance (sales volume, commissions, payouts).

- Export: XLS.

Feature	User Story	Acceptance Criteria	BDD Scenario

| Agent Performance Report | As a supervisor, I want to assess agent performance across commissions and sales. | - Filters: agent name, insurer, date.- Metrics include total commissions, number of policies, total payouts.- Exportable to XLS. | **Given** I filter by agent and month**When** I generate the report**Then** I see their totals for that period. |

3.8.4 Claims Report (Claim Summary Report)

Features:

- Detailed claim data with filters (policy, client, insurer, date range, status).

- Export: XLS (with "No data found" messages if applicable).

Feature	User Story	Acceptance Criteria	BDD Scenario
Claims Summary Report	As a claims manager, I want to view summaries of claims by status and client.	- Filters: policy number, client, insurer, status, date range.- Exportable to XLS.- Shows message if no results.	**Given** I apply filters**When** no claims match**Then** I see a message 'No data found' and no file is downloaded.

| Production Report | As a manager, I want to see production data by policy and commission. | - Filters: client, insurer, type, agent, date.- Shows policy number, premium, commissions. | **Given** I run the report**When** filters are applied**Then** I see matching results and can export to XLS. |

Clients Report	As an admin, I want to track client activity.	- Aggregates policy count, claims per client.- Filterable and exportable.	**Given** I open client report**When** I filter by active clients**Then** I see policy and claim metrics.
Agents Report	As a supervisor, I want to assess agent performance.	- Shows sales, commissions, payouts.- Filters: agent, insurer, period.- Exportable.	**Given** I view agent report**When** I select a date range**Then** I see the filtered agent performance.
Claims Report	As a claims manager, I want to summarize claims by status/client/date.	- Displays claims by type, amount, decision.- Filterable and exportable with 'No data' fallback.	**Given** I run a claims report**When** no matches exist**Then** I see a 'No data found' message.

3.9 Settings (Administration)

Centralized area for global settings and user management.

3.9.1 Employees (Zaposleni)

Features:

- Manage employee profiles (Name, Email, Password, Function, Phone, Bank Details, Group, JMBG, Address, Skype, Active Status).

- Email Settings: Configure "From" name, email, SMTP server, port, SSL, BCC, signature.

- Actions: Add, edit, send credentials, activate/deactivate.

- Logging: Detailed change logs (timestamp, IP address, description)

Feature	User Story	Acceptance Criteria	BDD Scenario
Employee Management	As an admin, I want to manage employee profiles and system access.	- Add/edit: name, email, password, role, function, phone.- Activate/deactivate employees.- Trigger credential email on creation.	**Given** I add a new employee**When** I submit their info**Then** the account is created and credentials are emailed.

3.9.2 Privileges (Privilegije)

Features:

- Configure page view and action-based privileges.

- Actions: Grant/restrict access to modules.

- Audit Logging: Track changes (timestamp, IP address, description)

Feature	User Story	Acceptance Criteria	BDD Scenario
Permission Configuration	As an admin, I want to configure module and action-based permissions.	- Can restrict access to entire modules and specific actions (view, edit, delete).- Changes take effect immediately.- Permission changes logged.	**Given** I modify a user's permission**When** I save**Then** their access updates instantly and is recorded in the log.

3.9.3 Company Data (Preduzeće)

Features:

- Manage company information (Company Name, Activity, Address, City, Postal Code, Country, Phone, Account, PIB, Registration Numbers).

- Invoice Notes & Payment Instructions: Default texts for invoices

- Actions: Update and save with logging.

Feature	User Story	Acceptance Criteria	BDD Scenario
Manage Company Info	As an admin, I want to manage my organization's data used in the app and on invoices.	- Editable fields: name, address, PIB, account, phone.- Data appears in invoice header.- Updates tracked in logs.	**Given** I update company address**When** I save it**Then** new invoices reflect the update and it is logged.

3.9.4 Instructions (Uputstva)

Features:

- Create and manage internal guidelines (Title, detailed text, associated module).

- Actions: Add, edit, delete, and search instructions.

- Usage: Display contextually within the application.

Feature	User Story	Acceptance Criteria	BDD Scenario
Internal Guidelines	As an admin, I want to publish internal	- Fields: title, rich-text content, module context.- Searchable by	**Given** I create an instruction**When** users open the related module**Then** they

	instructions by module.	keyword.- Editable and deletable.	see the contextual help content.

Employee Management	As an admin, I want to manage employee profiles and credentials.	- Add/edit name, role, contact, credentials.- Send welcome email.- Toggle active/inactive status.	**Given** I add an employee**When** I fill required info**Then** a login email is sent and user appears in the list.
Permission Configuration	As an admin, I want to manage what modules users can access.	- Page/action-level permission controls.- Immediate effect after saving.- Changes logged.	**Given** I update a user's permissions**When** I save**Then** changes take effect and are recorded.
Company Information	As an admin, I want to configure company info for branding and invoicing.	- Editable fields: name, address, bank info, invoice notes.- Format validation.- Updates logged.	**Given** I change invoice details**When** I save**Then** new invoices reflect the update.
Instructions Module	As a product admin, I want to publish how-to guides per module.	- Includes title, rich-text, and module link.- Editable and searchable.- Shown contextually.	**Given** I create an instruction**When** users enter the module**Then** it appears as help content.

4. User Roles & Access Control

4.1 Overview

Policy Hub uses RBAC with three primary roles:

- **Super Admin:** Full control by Policy Hub owners.

- **Admin:** Broker company representatives managing their internal users.

- **Employee:** Regular users with restricted access.

4.2 Detailed Role Definitions

4.2.1 Super Admin

- **Description:** Reserved for Policy Hub owners.

- **Responsibilities:**

    - Full system control (global settings, user management, company administration).

    - Manage broker companies and Admin accounts.

- **Privileges:**

    - Unrestricted access; override broker-specific restrictions.
- **Example Actions:**
    - Creating new broker companies.
    - Configuring system-wide integrations and security policies.
    - Managing Admin accounts.

4.2.2 Admin

- **Description:** Assigned to broker companies.

- **Responsibilities:**

    - Manage all employees within their company.

    - Configure feature access and adjust privileges.

    - Monitor usage and generate company-specific reports.

- **Privileges:**

- ○ Access to operational modules (Dashboard, Policies, Sales, Claims, Finances, Codebook, Agent, Reports).

- ○ Manage Employee accounts.
- **Example Actions:**
  - ○ Onboarding a new Sales Representative.
  - ○ Granting or revoking access to specific modules (e.g., Claims, Invoicing).
  - ○ Modifying privilege settings for employees.

4.2.3 Employee

- **Description:** Regular users with restricted access.

- **Responsibilities:**

  - ○ Execute day-to-day tasks (processing policies, handling leads, submitting claims).

- **Privileges:**

  - ○ Limited access as defined by Admin.
- **Example Actions:**
  - ○ Creating/updating policy records.
  - ○ Entering lead data.
  - ○ Submitting claim forms.

4.3 Role Assignment & Management Process

- **Initial Setup:** Super Admins create Admin accounts; Admins onboard Employees.

- **Modifications:** Super Admins update roles globally; Admins adjust Employee privileges.

- **Enforcement:**

  - ○ Frontend: Dynamic UI renders menus based on roles.

  - ○ Backend: JWT-based API role validation.

- **Audit Logging:** All changes to roles and privileges are logged.

4.4 Integration with System Modules

- **Dashboard:** Personalized widgets based on user role.

- **Module Access:** Role restrictions ensure Employees see only permitted features.

- **Custom Adjustments:** Super Admins and Admins can fine-tune privileges.

Feature	User Story	Acceptance Criteria	BDD Scenario
Role Definitions	As a system architect, I want roles defined with clear access.	- Roles: Super Admin, Admin, Employee.- Each role has clear permission set.- UI renders accordingly.	**Given** a user logs in**When** their role is checked**Then** only allowed modules/actions are shown.
Role Assignment	As a Super Admin, I want to assign roles to users.	- Admins assign Employees.- Super Admins assign Admins.- Audit log updated.	**Given** I assign a role**When** I save**Then** the user has new access and it is logged.
Permission Enforcement	As a developer, I want both frontend and backend to enforce role-based permissions.	- UI restricts access.- API validates JWT roles.- Unauthorized actions denied and logged.	**Given** I access a restricted endpoint**When** I'm unauthorized**Then** I get an error and it is logged.
Role Audit Logging	As an auditor, I want role/permission changes tracked.	- Logs user ID, change, timestamp, IP.- Exportable and filterable.	**Given** a role change occurs**When** I view audit log**Then** all changes are listed with metadata.

5. User Flows

5.1 New Employee Onboarding Flow

- **Admin Login & Navigation:** Admin logs in and navigates to Settings > Employees.

- **Initiate Onboarding:** Click "Add Employee" to open the form.

- **Form Entry:** Enter employee details and assign role.

- **Submit & Confirmation:** Validate, create record, log change, send welcome email.

- **Employee Onboarding:** New Employee logs in and sees a role-tailored UI.

5.2 Policy Review & Finalization Flow

- **Employee Navigation:** Employee navigates to Policies > Policies Workflow.

- **Policy Import:** New policies are imported from insurance companies (manual creation is disabled).

- **Review:** Employee reviews imported policy details (policy number, type, dates, parties, insurer/product, financials, commission settings, notes, and additional attributes).

- **Document Upload:** Attach supporting files as needed.

- **Finalization:** Validate all mandatory fields and mark the policy as complete.

- **Audit:** Record all modifications.

5.3 Claim Submission & Processing Flow

- **User Navigation:** Employee or Claims Adjuster navigates to Claims.

- **Initiate Claim Submission:** Click "New Claim" to open the form.

- **Form Entry:** Input claim details (claim ID, policy number—auto-populated details, damage description, location, date, amounts, etc.) and upload documents.

- **Submit Claim:** System sets status to "In Processing" and logs the submission.

- **Review & Update:** Claims Adjuster reviews and updates status.

- **Final Resolution:** Claim is marked as resolved; details appear in Claims Report.

5.4 Sales Process & Quote Management Flow

- **Lead Capture:** Sales Rep adds a new lead in Sales > Leads.

- **Quote Generation:** If a lead is interested, broker generates a quote for a specific insurance type.

- **Quote Distribution:** Broker sends the quote to multiple insurance companies.

- **Insurer Response:** Insurance companies review the quote and respond with their offers.

- **Client Selection:** Broker forwards insurer quotes to the client; client selects the best quote.

- **Policy Import:** Broker sends the chosen quote back to the insurer; insurer creates the policy, which is imported into Policy Hub.

- **Review & Finalization:** Imported policy is reviewed and finalized as described in 5.2.

5.5 Financial Operations Flow

- **Commission Calculation:** Finance Specialist navigates to Finances > Commissions; commissions are computed automatically upon policy import/updates.

- **Invoice Creation:** Finance creates invoices based on commission data under Finances > Invoicing.

- **Statement Processing:** Finance imports bank statements under Statement Processing; system auto-matches or allows manual linking.

- **Logging & Reporting:** All financial operations are logged and reflected in reports.

Flow	User Story	Acceptance Criteria	BDD Scenario
New Employee Onboarding	As an admin, I want to onboard a new user.	- Form collects required info.- Sends welcome email.- Logs onboarding.	**Given** I fill the onboarding form**When** I submit**Then** user is created and receives credentials.
Policy Review & Finalization	As a broker, I want to finalize imported policies after validation.	- Required fields validated.- Document uploads supported.- Finalize triggers audit log.	**Given** I review a policy**When** I finalize it**Then** it is marked completed and logged.
Claim Submission & Processing	As a user, I want to submit and manage claims.	- Prefilled policy data.- Files and dates required.- Status updated by authorized users.	**Given** I submit a claim**When** I attach documents**Then** claim enters 'In Processing'.
Sales & Quote Flow	As a broker, I want to manage lead to policy conversion.	- Track all stages (lead → quote → selection → policy import).- Insurer and client decisions recorded.	**Given** I select a quote**When** it's accepted**Then** policy is created by insurer and imported.
Financial Operations	As a finance user, I want to manage commissions, invoices, and payments.	- Commissions auto-calculated.- Invoices generated from settlements.- Reconciliation includes matching.	**Given** I upload a statement**When** I process it**Then** payments are matched to invoices and logged.

6. Localization & Internationalization

Purpose:
To support multiple languages (English, Serbian, Macedonian, Spanish) so users can interact in their preferred language.

6.1 Supported Languages

- English (EN)

- Serbian (SR)

- Macedonian (MK)

- Spanish (ES)

6.2 Implementation Strategy

- **Frontend Localization:**

  - Use language JSON files (e.g., /locales/en.json).

  - Use a translation library (e.g., react-i18next).

  - Include a global language switcher in the header/settings.

  - Adapt date, number, and currency formats based on locale.

- **Backend Localization:**

  - Localize API responses based on Accept-Language header or user profile.

  - Middleware sets locale per request.

- **Content Management:**

  - Interfaces in Settings for managing translations.

  - Dynamic content remains in the original language.

- **Testing & Future Enhancements:**

- ○ Snapshot tests and manual UI checks.

- ○ Architecture supports easy addition of new languages.

- ○ Integrate with translation tools (e.g., Crowdin, Lokalise).

Feature	User Story	Acceptance Criteria	BDD Scenario
Language Selection	As a user, I want to select a language for the interface.	- Languages: EN, SR, MK, ES.- Saved across sessions.- Toggle available in UI.	**Given** I choose Spanish**When** I navigate**Then** the UI shows in Spanish.
Localized UI Content	As a user, I want translated UI content.	- All labels/buttons/menu texts must translate.- Translations stored in JSON files.	**Given** I change language**When** I open any module**Then** labels are translated.
Locale-Specific Formats	As a user, I want local formatting for dates, numbers, currency.	- Date and currency formats adjust to locale.- Formatting adapts dynamically.	**Given** I use the Serbian locale**When** I open invoices**Then** they use Serbian date and RSD currency formatting.
Backend Localization	As a developer, I want API responses localized.	- Responses adapt to Accept-Language header.- Locale middleware sets language per request.	**Given** a request includes Accept-Language**When** I call API**Then** responses follow the requested language.

Translation Management	As a content manager, I want to manage translation keys.	- Strings editable via UI or Crowdin.- Changes reflected in real time.	**Given** I update a translation key**When** I save**Then** the UI shows the new text.

Localized Interface	As a global user, I want all static interface elements translated based on my language setting.	- All labels, menu items, and button text translated.- Stored in JSON files using translation keys.- Dynamic data (e.g., user notes) not translated.	**Given** I navigate to the Sales module in Serbian**When** I view the UI**Then** all standard UI text appears in Serbian.

Date & Currency Formatting	As a user, I want dates, numbers, and currencies formatted according to my locale.	- Date, number, and currency format match locale conventions.- Based on selected language or browser default.	**Given** my locale is set to Serbian**When** I view policy premiums**Then** they appear with RSD currency and Serbian date format.

Localized API Responses	As a developer, I want the backend to respect user language preferences in error and system messages.	- API reads Accept-Language header or user profile.- Middleware adjusts response messages accordingly.	**Given** I send a request with Accept-Language: es**When** the server responds**Then** the messages are in Spanish.

Manage Translation Keys	As a content manager, I want to manage UI translations without deploying code.	- Translations editable via UI or third-party integration (e.g., Crowdin).- Changes reflected immediately on frontend.	**Given** I update a translation in Crowdin**When** I publish the changes**Then** the UI reflects the updated text automatically.

---

7. Technical & Implementation Guidance

7.1 Backend (Node.js)

- **Framework:** Express or Nest.js for RESTful API.

- **Database:** MSSQL or PostgreSQL with an ORM (Sequelize, TypeORM, Prisma).

- **Authentication:** JWT-based with middleware for role verification.

- **API Documentation:** Utilize Swagger/OpenAPI.

7.2 Frontend (React/Next.js)

- **Reusable Components:** Develop common components (forms, tables, modals).

- **State Management:** Use Context API or Redux.

- **Role-Based UI:** Dynamically render content based on role and locale.

7.3 Security & Compliance

- **Encryption:** Use HTTPS; secure data at rest.

- **RBAC Enforcement:** Validate user roles on every API call.

- **Audit Logging:** Maintain detailed logs for sensitive actions.

7.4 Testing & Scalability

- **Testing:** Use Jest/Mocha for backend, Cypress for E2E tests.

- **Caching & Indexing:** Implement caching (e.g., Redis) and optimize indexes.

- **Horizontal Scaling:** Design Node.js services for load balancing and scalability.

7.5 Future Enhancements

- **Mobile App:** Consider React Native for mobile.

- **Advanced Analytics:** Integrate AI-based risk and claims prediction models.

- **Expanded API Integrations:** Connect with external insurers, OCR services, email processors.

System Requirements Table

Category	Requirement	Details
Backend Framework	Node.js (Express/Nest.js)	REST API with middleware, controllers
Database	MSSQL / PostgreSQL	ORM: Sequelize, Prisma, or TypeORM
Authentication	JWT	Token-based auth with role middleware
API Documentation	Swagger / OpenAPI	All endpoints documented
Frontend Framework	React / Next.js	Reusable UI components

State Management	Context API / Redux	For global app state
UI Access Control	Role-based rendering	Menus and views show/hide based on role
Localization Support	react-i18next	JSON-based translation with fallback
Logging	Audit trail	Logs with user, time, IP, and description
Security	HTTPS, encryption	TLS endpoints; encrypt data at rest
Caching	Redis	Speeds up response for heavy endpoints
Testing	Jest/Mocha, Cypress	Unit and end-to-end test coverage
Scalability	Horizontal scaling	Load balancing support
Future Enhancements	React Native, AI analytics, OCR	Mobile app, predictive modeling, and OCR for document parsing