NABU AI

FROM CHAOS TO CLARITY

SPRINT 1

presented by

PUSHTI, RUTHVIK, SRESHTA, SANJAY & VISHAL

overview

what

NabuAI is an AI-powered tool for academia, turning their scattered resources into a single, intelligent knowledge base.

why

Manual organization is messy and inefficient, creating a "saved content graveyard" that costs academia hours of productive work.

how

With three simple steps: capture, curate, converse. We save any content with a single click, our AI organizes it instantly, and you can then take action on your knowledge base.

functional requirements

content capture

Users can capture and save content (URL, text, PDF, image) via the browser extension or web interface.

semantic retrieval

The system must execute a semantic search that retrieves content chunks based on the meaning of the user's query

asynchronous ingestion

Upon content capture, the system must immediately return a 'Success' status, initiating asynchronous processing.

conversational ai

Allow users to input natural language queries into a chat interface and receive answers or generate content

content organization

The system must allow users to create, rename, and assign tags to the collections (Scribes)

collaboration & sharing

The system must allow a user to invite others to view and/or edit their Scribes, with defined access permissions (multi-user support).

non-functional requirements

performance (latency)

Capture Latency < 3 Seconds. The time from the user clicking "Save" to receiving the success status must be less than 3s

performance (retrieval)

Semantic search query results must be returned to the conversational AI within 500 milliseconds.

reliability (availability)

platform must maintain 99.9% uptime during standard academic work hours (7 AM – 11 PM local time).

scalability

The system must be designed to scale horizontally to support 100K active users without requiring major overhaul.

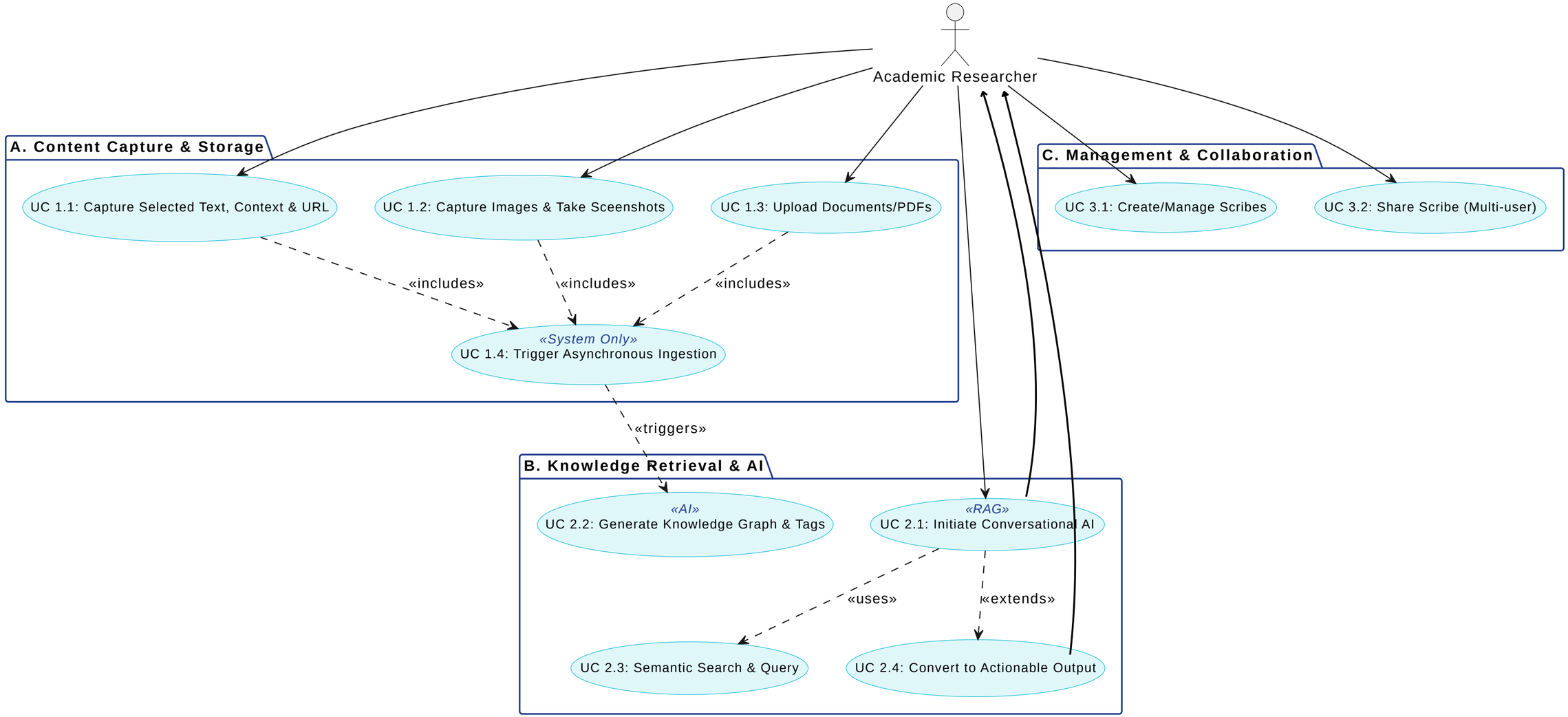
security

All data (content files, database records, vector embeddings) must be encrypted at rest and in transit (HTTPS/TLS), adhering to multi-tenant privacy isolation standards.

usability (ux)

The browser extension UI must be clean, minimalist, and use a maximum of three primary interaction steps to save content.

use case diagram

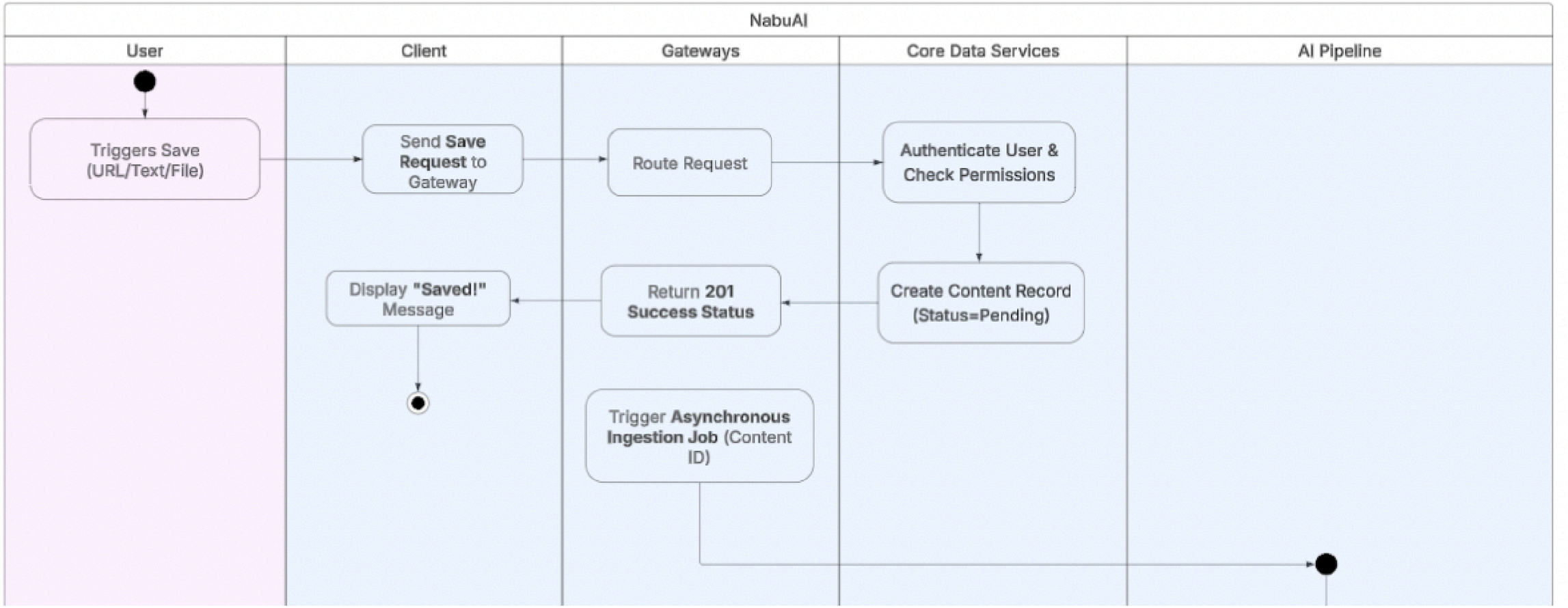


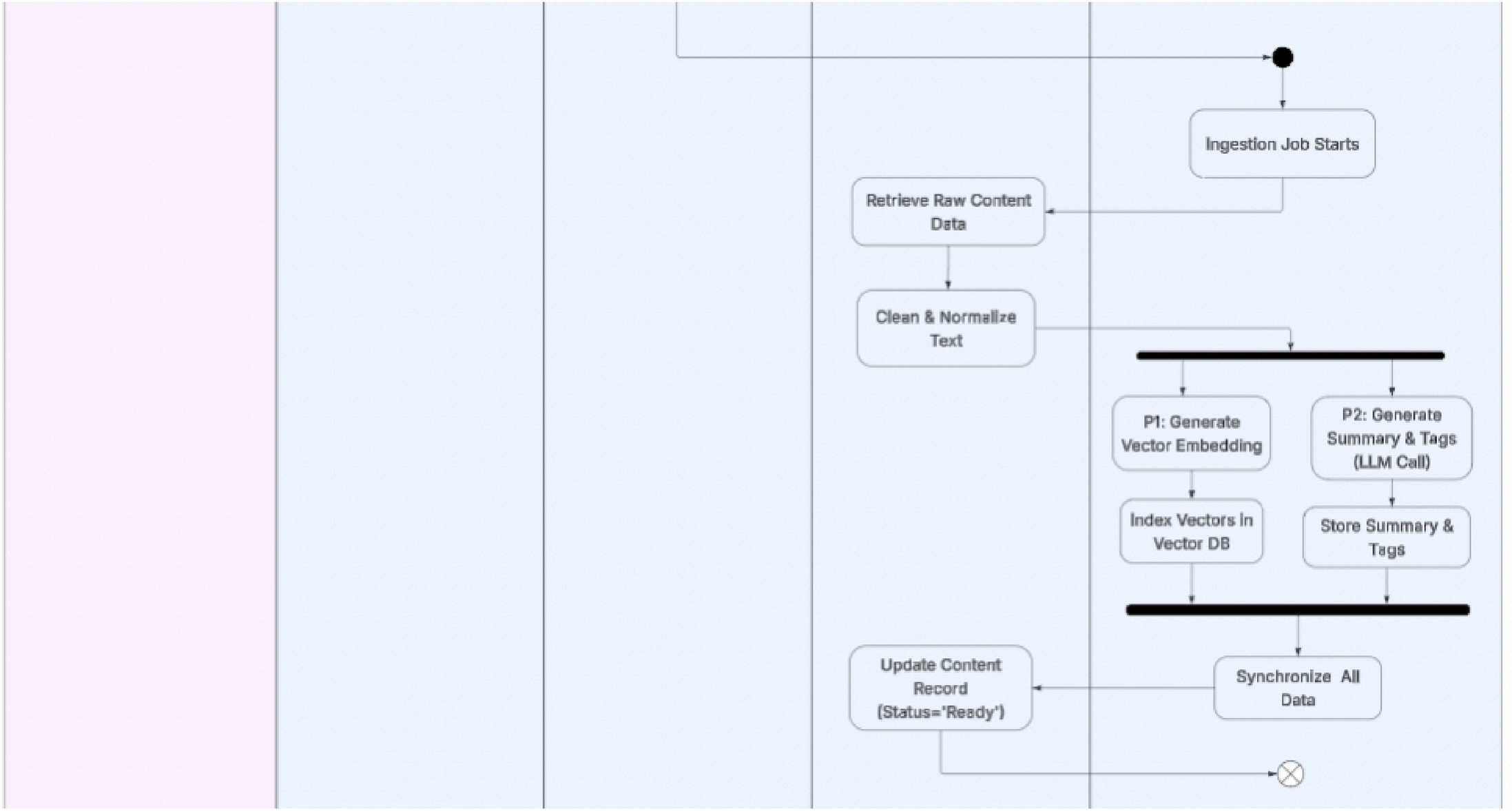
activity diagram

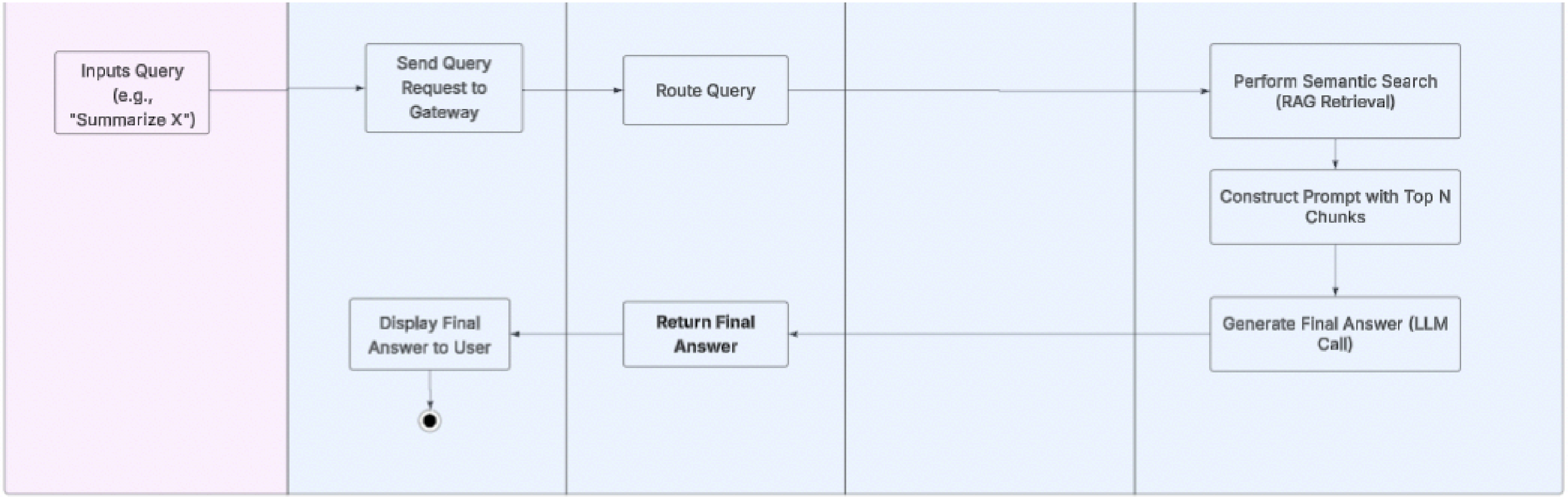
This Activity Diagram serves as the definitive process map for the NabuAI system, detailing how every piece of content travels from the user's browser to our intelligent archive.

The diagram is structured using distinct swimlanes to clearly define the responsibilities of the **User, Client**, **Gateway**, **Core Data Services**, and **AI Pipeline**. It visually proves two essential activities:

1. **Content Capture:** We show how the system ensures you get an instant "Saved!" confirmation in the browser, even while the heavy, complex AI analysis/tagging is triggered asynchronously in the background.
2. **Knowledge Processing, Retrieval & Generation:** This process outlines the Retrieval-Augmented Generation (RAG) flow of the NabuAI system. This process begins when a user submits a question, which is converted to a vector to semantically retrieve the most relevant content chunks from their respective scribe. This retrieved context is then used by the AI to either return an accurate answer grounded directly in the source material or to generate new content (like outlines and reports) based solely on their research.


NabuAI
User
Client
Gateways
Core Data Services
AI Pipeline
Triggers Save (URL/Text/File)
Send Save Request to Gateway
Route Request
Authenticate User & Check Permissions
Create Content Record (Status=Pending)
Return 201 Success Status
Display "Saved!" Message
Trigger Asynchronous Ingestion Job (Content ID)


Ingestion Job Starts
Retrieve Raw Content Data
Clean & Normalize Text
P1: Generate Vector Embedding
P2: Generate Summary & Tags (LLM Call)
Index Vectors in Vector DB
Store Summary & Tags
Synchronize All Data
Update Content Record (Status='Ready')


Inputs Query (e.g., "Summarize X")
Send Query Request to Gateway
Route Query
Perform Semantic Search (RAG Retrieval)
Construct Prompt with Top N Chunks
Generate Final Answer (LLM Call)
Return Final Answer
Display Final Answer to User

user stories & product backlog

The Product Backlog for NabuAI, serves as the **single source of truth** for all future development. It translates our **user research, business goals,** and **system architecture** into prioritized, actionable **User Stories**, each with **clear** and **testable acceptance criteria.**

We will manage this entire backlog using the **Scrum** framework. All stories are tracked in **Jira** to provide full **transparency** on the workload, **monitor team velocity** across short sprints. This process allows us to rapidly test features, gather feedback from our target users, and quickly adapt the product vision.

The backlog is organized into three distinct **EPICS**:

- **Content Capture**
- **Content Ingestion, Storage**
- **Knowledge Graph, Information Retrieval & Conversational AI**

The effort estimates for the user stories are decided by playing the **Planning Poker**, and work items are prioritized based on need to build out the functional MVP

content capture

As a Researcher, I **want** a Chrome extension icon that shows a dedicated NabuAI save button, **so** I can initiate capture quickly.

WHEN I click the NabuAI icon, **THEN** the core capture modal must appear instantly (NFR 1.0) and display a "Save" button.

As a Researcher, I **want** to select text on any webpage and save it via the context menu, **so** I can capture key quotes instantly.

GIVEN I have selected text, **WHEN** I right-click, **THEN** a "Save to NabuAI" option must appear and open the modal with the text pre-populated.

As a Researcher, I **want** to open a PDF/PPTX/DOCX file in an embedded viewer in the dashboard **so** I can save it directly without downloading.

GIVEN I upload a file under 10MB, **THEN** the raw file is stored in S3, and a content record is created with a fileType specified.

As a Researcher, I want the dashboard to support drag-and-drop functionality so I can easily organize content into Scribes.

. . . and more

content ingestion, storage & processing

As a Researcher, I want to generate an influence map for a specific Scribe, showing which content items are most semantically related to the central topic

As a System, I need a process to chunk and vectorize all new raw text content, so it can be indexed for semantic search (RAG foundation)

As a System, I need to ensure large content files (PDFs/Images) are stored securely in S3 and linked via URI to the main database record

As a System, I must store the raw content text, file link, and metadata in the database upon successful ingestion, so it is ready for AI processing

. . . and more

information retrieval & conversational AI

As a Researcher, I want a chat box where I can ask natural language questions ("What are the key themes in X?"), so I can get instant, grounded answers.

As a Researcher, I want the system to automatically suggest a relevant existing Scribe for a newly saved item, so I can keep my library instantly organized

As a Researcher, I want the AI to rewrite retrieved quotes in my own voice/tone, so I can integrate sourced material into my drafts

As a Researcher, I want the AI to identify and list key entities (names, concepts, dates) from a document, so I can quickly extract structured data without re-reading.

. . . and more

sprint 1 backlog & check-in

- **Sprint Name:** NabuAI Sprint 1
- **Duration:** Sep 29 – Oct 11, 2025
- **Total Work Items:** 8 items
- **Completed Work Items:** 3 items
- **Sprint Goal:** To complete the content capture work items and the dashboard to view the saved content
- **Completion Rate:** 31.2%
- **Total Story Points Committed:** 16 points
- **Total Story Points Completed:** 5 points
- **Highlights:** Substantiated the use of a chrome plugin to capture the content and various content types an academic researcher might interact

As a Researcher, I **want** a Chrome extension icon that shows a dedicated NabuAI save button **so** I can initiate capture quickly.

Story Points: 2 **Assignee:** Ruthvik Vijayakumar **Status:** Completed

As a Researcher, I **want** to select text on any webpage and save it via the context menu **so** I can capture text with context instantly.

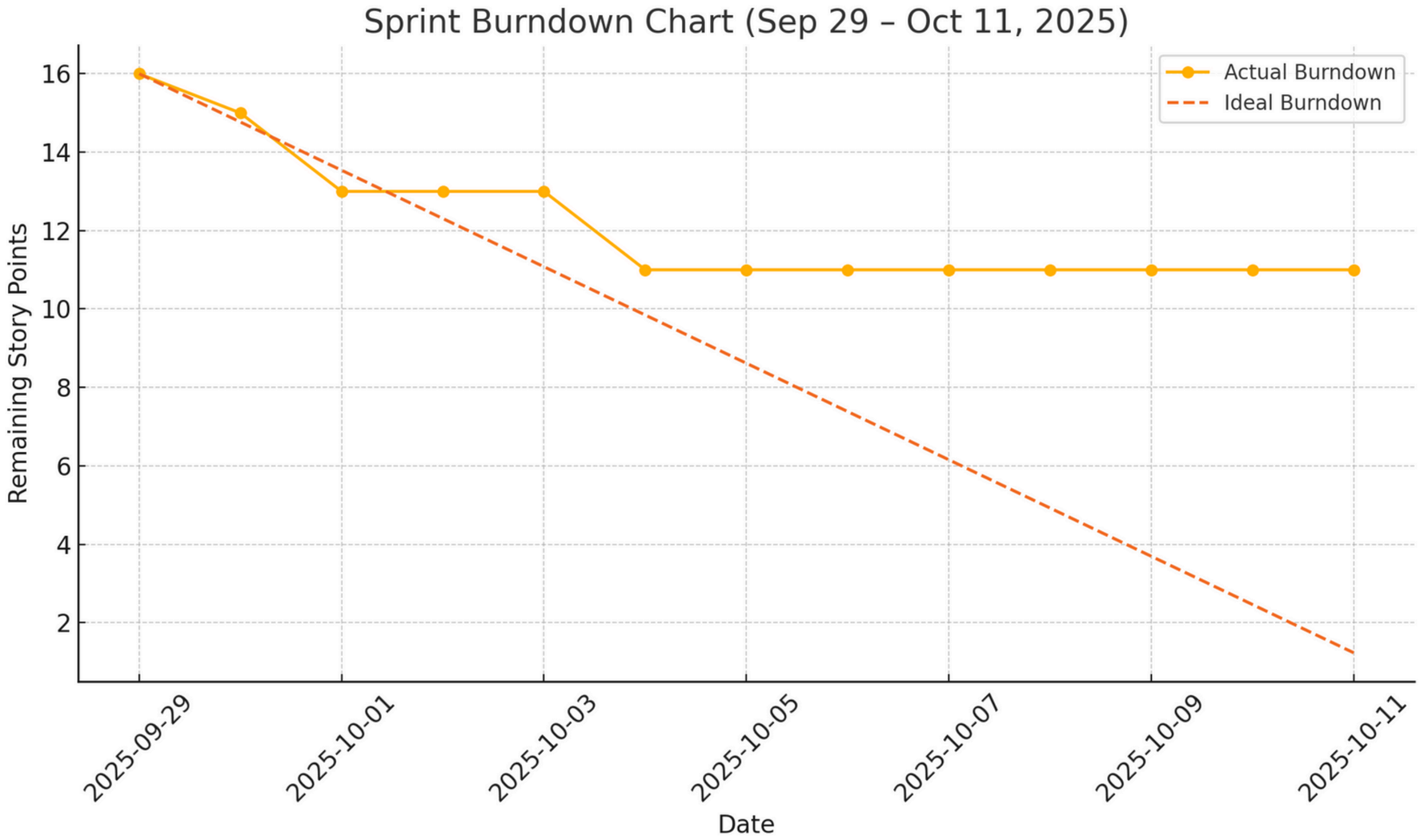
Story Points: 2 **Assignee:** Sanjay Devang **Status**: Completed

As a Researcher, I **want** to open a PDF/PPTX/DOCX file in an embedded viewer in the dashboard **so** I can save it directly without downloading.

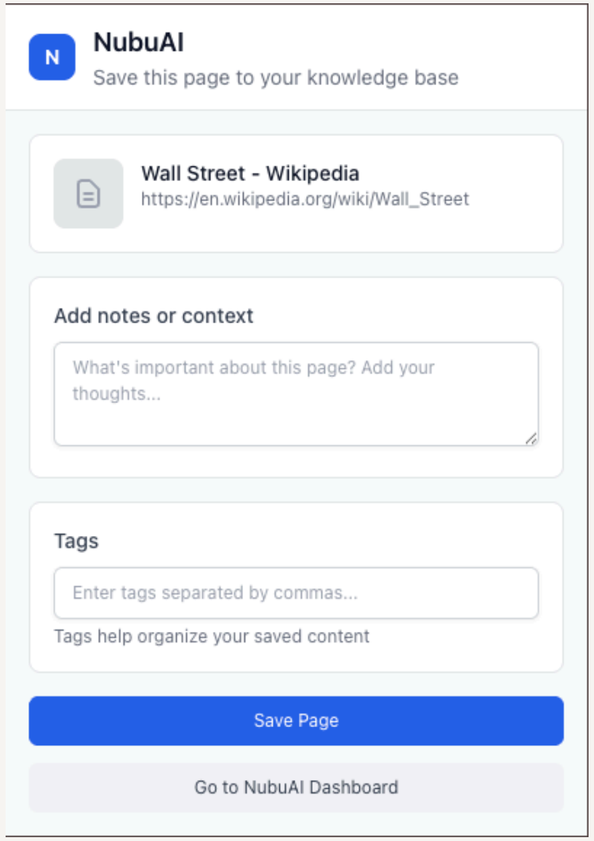
Story Points: 4 **Assignee:** Ruthvik Vijayakumar **Status:** In Progress

. . . and more

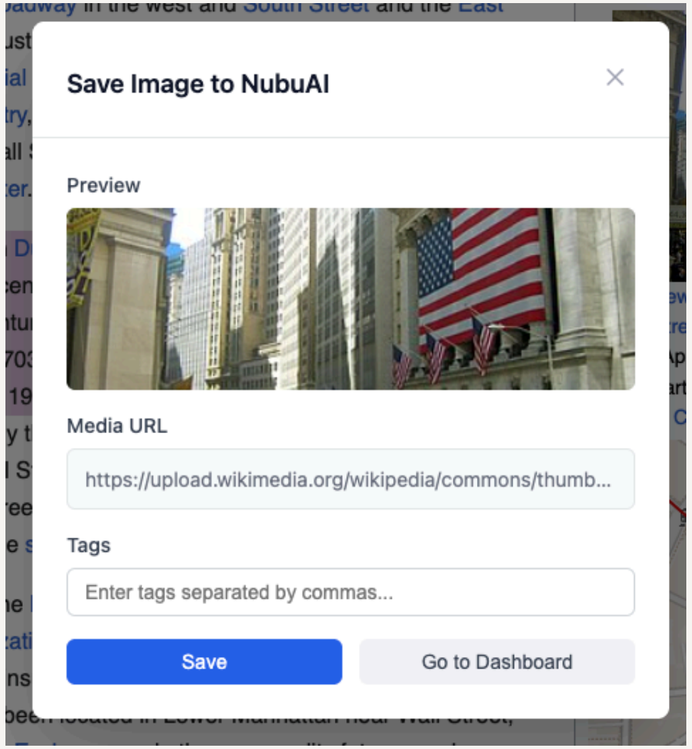
sprint 1 burndown chart



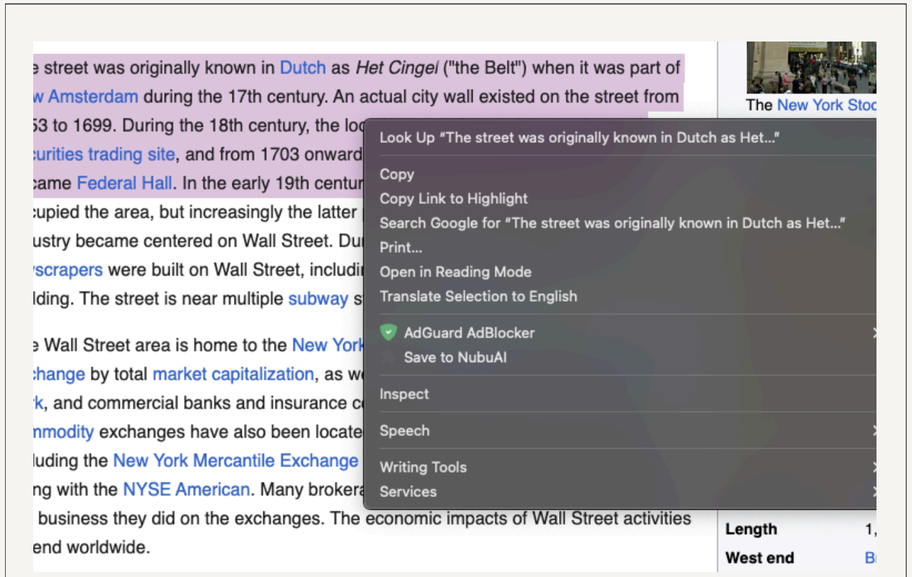
some updates so far . . .



chrome extension popup



saving images with context



saving text content with context

what’s next & demo

- Develop a **comprehensive design language** and brand system in **Figma**, including UI components for each user interaction and touchpoint.
- Begin backend development, setting up robust APIs and a **scalable data pipeline** to handle content ingestion, with data stored in **MongoDB** and/or **PostgreSQL** depending on the use case.
- Implement **authentication and authorization services** to ensure secure user access and service-level protection.
- Build an interactive d**ashboard with drag-and-drop functionality**, enabling users to customize their workflows and visualizations seamlessly.

THANK YOU

NABUAI - FROM CHAOS TO CLARITY

presented by

PUSHTI, RUTHVIK, SRESHTA, SANJAY & VISHAL

feedback from class presentation

- Include feedback mechanism for the usecase diagram from the system or its components back to the main actor to represent a more complete interaction flow.
- Divide the activity diagrams into separate stages for better readability and clarity, as the font size in the current version was too small to view on screen.
- Consider adding a sentiment analysis feature for text selection to display the number of positive and negative sentiments identified.