



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

194.207 Generative AI (VU 4,0) 2025W

Group : 181

Group members : Al- Mamun Abdullah | Hasan Razib

Table of Contents

Section	Subsections
1. Introduction	1.1 Target Group 1.2 User Goals 1.3 User Workflow
2. Data	—
3. The Problem	—
4. The Solution	4.1 Concept 4.2 User Interface 4.3 Technical Approach
5.1 Project Plan	5.1.1 Users 5.1.2 Goals 5.1.3 User Workflow 5.1.4 Evaluation 5.1.5 Evaluation Procedure
6. Project Timeline	—
7. Group Member Contribution	—
8. Risks and Limitations	8.1 Risks 8.2 Limitations
9. Scope Constraints and Exclusions	9.1 Out-of-Scope Features 9.2 Technical Limitations 9.3 Summary of Scope

Personal Study Notes Searcher (PSNS)

1. Introduction

The Personal Study Notes Searcher (PSNS) is a generative AI tool designed to help TU Wien students efficiently search and summarize their diverse study materials. Students often manage many PDFs, screenshots, handwritten notes, and digital files across multiple courses, making retrieval slow and fragmented. PSNS offers semantic search, summarization, and question answering over personal academic content, supporting faster exam preparation and more effective learning.

1.1 Target Group

PSNS is intended for TU Wien Bachelor's and Master's students who handle large amounts of educational material, including lecture PDFs, exercise sheets, screenshots, handwritten notes, and various digital documents. These resources are essential for studying, completing assignments, and connecting concepts across courses.

1.2 User Goals

Students want to:

1. Quickly access relevant information
2. Find specific concepts within fragmented notes
3. Summarize long texts (e.g., lecture PDFs)
4. Improve exam preparation with targeted questions
5. Reduce manual file searching
6. Manage heterogeneous materials in one interface

1.3 User Workflow

A typical workflow involves:

1. Attending lectures and gathering PDFs or screenshots
2. Taking handwritten or digital notes
3. Organizing files (often inconsistently)
4. Manually searching for information before exams
5. Re-reading long documents to locate definitions
6. Asking classmates when something cannot be found

During exam periods, the difficulty of retrieving information across many files increases. PSNS directly supports this workflow by enabling fast semantic search and summarization.

2. Data

The system works with five types of user-provided data:

1. The system works with five types of user-provided data:
2. Lecture PDFs
3. Slide screenshots (PNG/JPG)
4. Handwritten note photos (OCR)

5. Plain-text notes (Markdown, Notepad)
6. Homework PDFs (problems and solutions)



Image 01: Data Processing

Data Characteristics Along Required Dimensions

- **Granularity:** The dataset includes short, atomic notes (e.g., screenshots of single concepts) and long-form documents (50–200-slide PDFs). This variation makes retrieval challenging without specialized tools.
- **Connections:** Relationships between documents are mostly implicit, as students rarely use tags or metadata. The system must infer connections using embeddings.
- **Completeness:** Materials range from complete lecture PDFs to incomplete handwritten notes and context-poor screenshots. This inconsistency increases retrieval difficulty.
- **Context:** Metadata is minimal. Most files lack time, course, or structural information, requiring content-based rather than metadata-based search.
- **Heterogeneity:** Data spans multiple formats (PDF, images, text) and languages (English/German). Preprocessing is necessary to unify inputs.

Dataset Size Justification

A dataset of 50–200 documents is targeted. This size is large enough to evaluate retrieval quality while remaining computationally manageable. Students will contribute sample materials and additional synthetic or public documents as needed.

3. The Problem

TU Wien students commonly face:

- **Retrieval Difficulty:** Important formulas or definitions are hard to locate across many course documents.

- **Fragmented Knowledge:** Notes exist in mixed formats (PDF, JPG, TXT), preventing unified search.
- **Time Pressure:** Students often re-read long PDFs for small pieces of information.
- **No Easy Summarization:** Slides and handwritten notes rarely provide condensed overviews.

Chosen Core Problem

Fast, accurate retrieval of concepts across heterogeneous study materials (aligned with the “Retrieval” project category). Summarization of long academic documents is a secondary goal.

4. The Solution

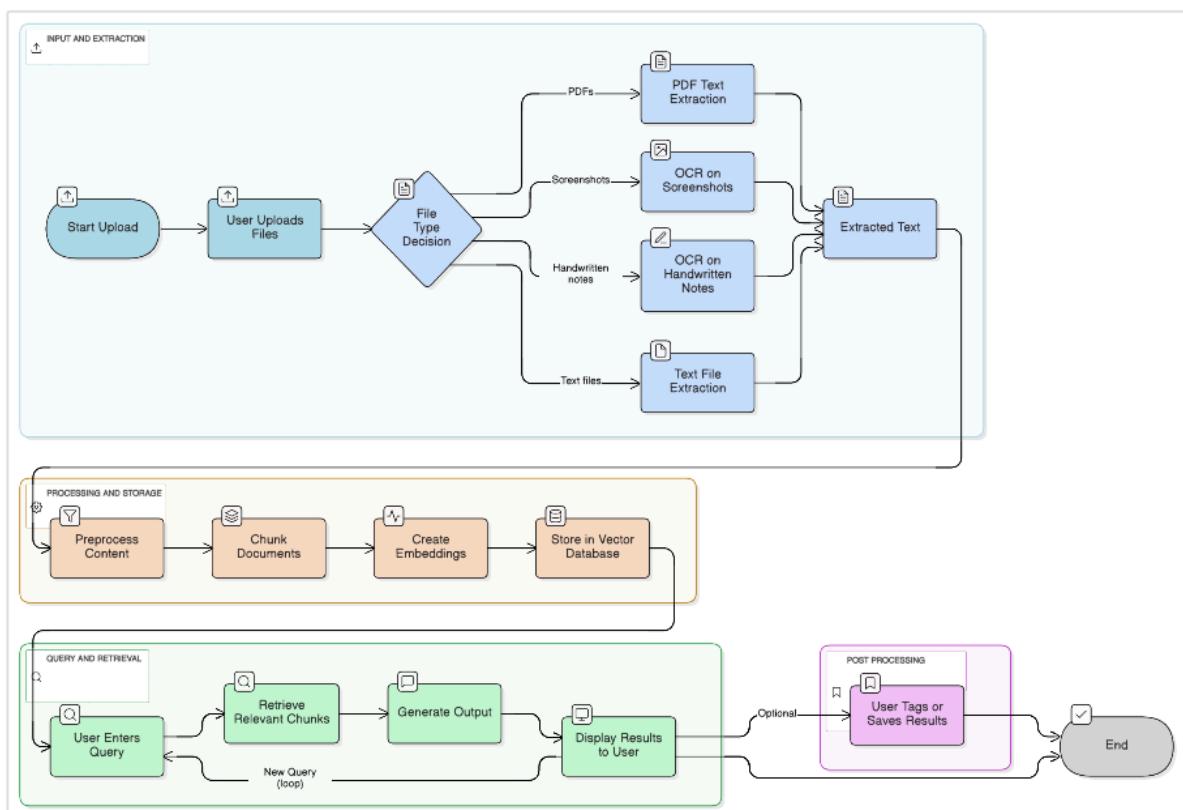


Image 02 : Work Flow Diagram

4.1. Concept: The PSNS System

The Personal Study Notes Searcher (PSNS) performs four main tasks:

1. **Understand:** Extract text from PDFs, apply OCR to images, chunk documents, and compute embeddings.
2. **Retrieve:** Accept user queries, perform vector search (FAISS), and return the most relevant chunks with source references.
3. **Generate:** Produce short summaries or context-based answers using RAG.
4. **Augment:** Allow optional tagging of retrieved items for later reference.

Workflow Summary:

Upload → Extract → Chunk → Embed → Query → Retrieve → Summarize/Answer → (Optional) Tag.

4.2. User Interface

A **Streamlit** web app will serve as the prototype interface due to fast development and minimal frontend effort.

- **Core UI Elements:**
 - File upload area
 - Status/indexing log
 - Search bar
 - Results display with chunk source references and summaries

4.3 Technical Approach

The architecture relies on **Retrieval-Augmented Generation (RAG)** with a vector database.

Key Technical Challenges:

1. **Optimal Chunking Strategy:** Documents vary widely in structure. The project will compare fixed, semantic, and structure-aware chunking using Retrieval Precision@k.
2. **Embedding Model Selection:** Technical TU Wien terminology may challenge generic embeddings. Multiple models will be tested using a small question set evaluated via MRR.
3. **OCR and Metadata Integration:** OCR inaccuracies affect retrieval quality. The pipeline will incorporate reliable OCR engines and attach minimal metadata (filename, timestamps) for hybrid filtering.

5.1 Project Plan

5.1.1 Users

Primary users are TU Wien Bachelor's and Master's students working with large sets of study documents.

5.1.2 Goals

Students want to retrieve concepts quickly, summarize large materials, ask questions over their own notes, reduce search time, and centralize resources.

5.1.3 User Workflow

Students typically collect PDFs/screenshots, take notes, store documents inconsistently, and manually search for information—especially during exams.

PSNS directly supports this workflow through unified semantic search and summarization.

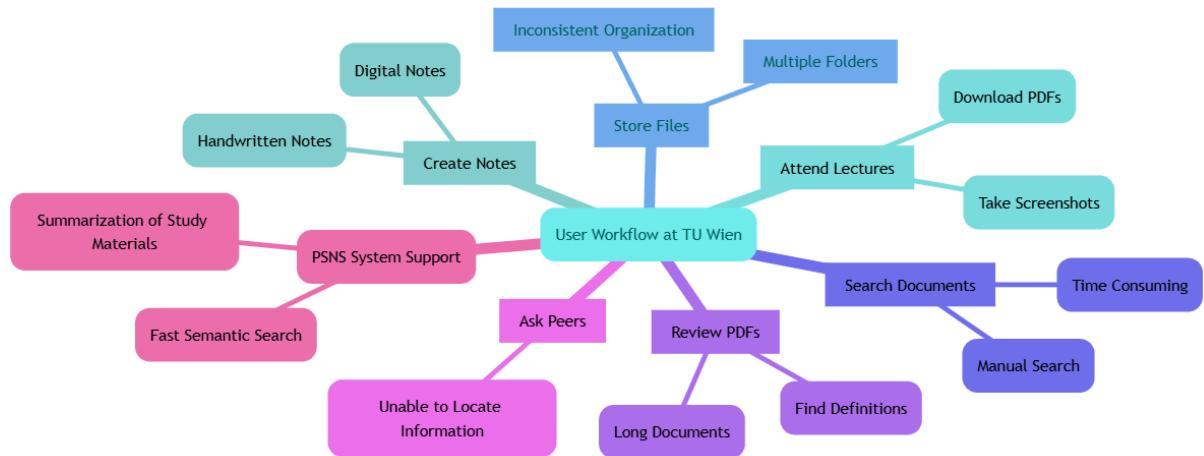


Image 03 :User Workflow

5.1.4 Evaluation

Only quantitative measures are used: retrieval and generation metrics (not usability).

Table 01: Success Metrics (Concrete and Measurable)

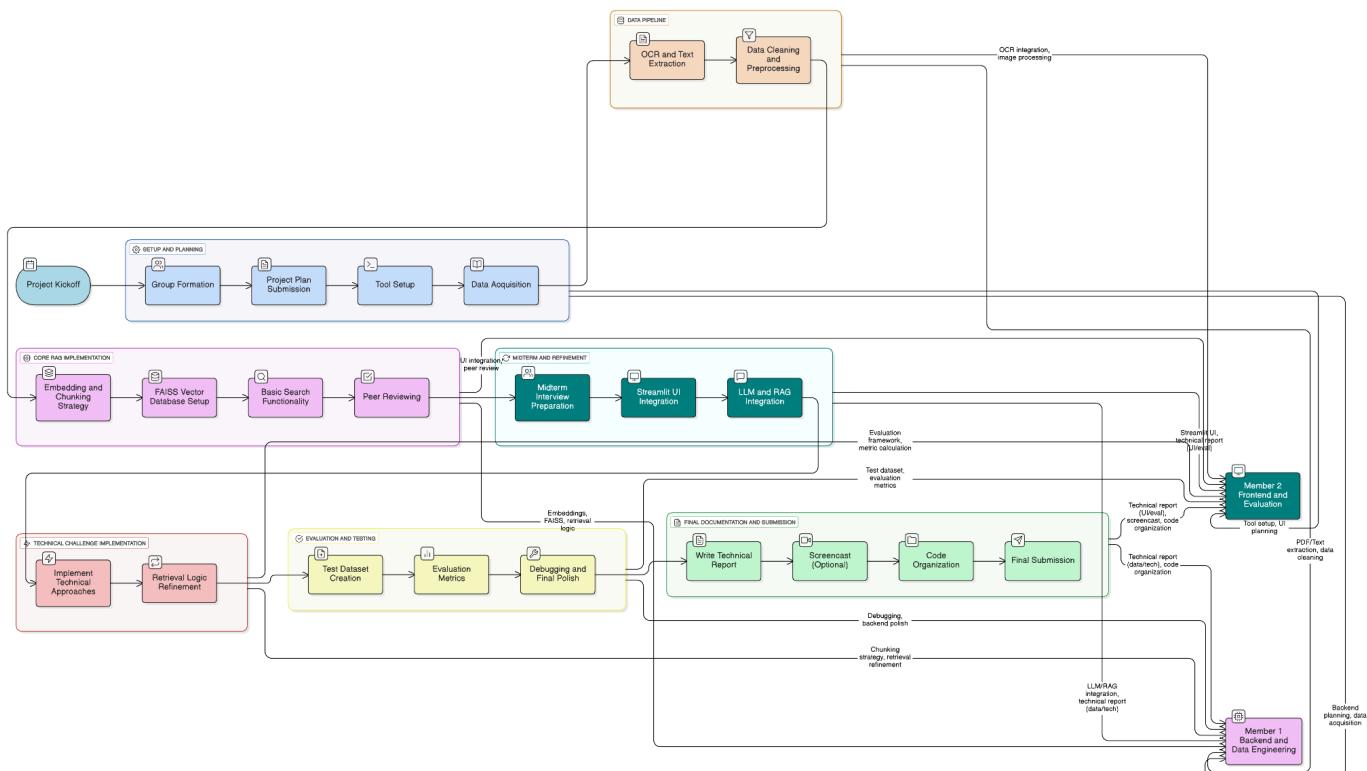
Metric	Goal	Method/Justification
Retrieval Precision@k	\$P@3 > 0.8\$	For a set of 20 test questions, calculate the percentage of times at least one of the top 3 retrieved chunks is relevant to the question. Higher value = better retrieval relevance.
MRR (Mean Reciprocal Rank)	\$MRR > 0.7\$	For the same set of 20 questions, measure the inverse of the rank of the first relevant document. Higher value = first result is more relevant.
Summarization/Answer Usefulness	80% Useful Score	A manual scoring of 20 generated answers (scale 1-3: Useless, Partially Useful, Fully Useful) to verify that the RAG pipeline provides correct, non-hallucinated answers.

Indexing Latency	\$< 5\$ seconds per 10 pages	Measure the time taken to process, chunk, embed, and index a batch of lecture PDFs. Ensures the system is practically usable.
------------------	------------------------------	---

5.1.5 Evaluation Procedure

- Test Dataset Creation:** Curate a small, dedicated test set of 20 complex questions based on the input documents. Manually identify the "ground truth" relevant chunks for each question.
- Benchmark Testing:** Run the test questions against the final system implementation.
- Result Analysis:** Calculate the defined metrics (P@k, MRR, Latency) and record the manual scoring for Answer Usefulness. The report will present these values and use them to justify the system's success in solving the core retrieval problem.

6. Project Timeline



This plan assumes a group size of **two members** and a total estimated effort of 2 hours \ times 80 = 160 working hours per member, totaling **320 group hours**, as per TUWEL expectations.

Table 02: Timeline & Durations

Week	Date Range (Approx.)	Phase/Milestone	Deliverables/Focus	Group Effort (Person-Hours)
1	Oct 22 – Nov 2	Setup & Planning	Group formation, Project Plan Submission (Nov 19) , Tool setup (Streamlit, FAISS), Data acquisition (collecting own notes).	40 (20/member)
2	Nov 3 – Nov 16	Data Pipeline	OCR & Text Extraction pipeline (PDF, Image, Text). Basic data cleaning and preprocessing.	70 (35/member)
3	Nov 17 – Nov 30	Core RAG Implementation	Embedding & Chunking Strategy implementation. FAISS Vector DB setup. Basic search functionality. Peer Reviewing (ends Nov 30) .	80 (40/member)
4	Dec 1 – Dec 13	Mid-Term & Refinement	Mid-Term Interview Preparation . Streamlit UI integration. LLM/RAG Integration (initial summarization/Q&A).	50 (25/member)
5	Dec 14 – Jan 4	Technical Challenge Implementation	Implementing the chosen technical approaches (e.g., semantic vs. structure-aware chunking). Refinement of the retrieval logic. (Includes holidays)	40 (20/member)

Week	Date Range (Approx.)	Phase/Milestone	Deliverables/Focus	Group Effort (Person-Hours)
6	Jan 5 – Jan 18	Evaluation & Testing	Test Dataset creation. Running evaluation metrics (P@k, MRR, Latency). Debugging and final polish.	30 (15/member)
7	Jan 19 – Jan 21	Final Documentation	Writing the Technical Report, Screencast (optional), Code organization. Final Submission (Jan 21).	10 (5/member)
Total				320 Person-Hours

7. Group Member Contribution

(Note: As the group consists of 2 members, roles are dual-hatted to cover both key areas.)

Table 03: Team Works Contribution

Member	Focus Area	Detailed Contributions
Member 1	RAG Backend & Data Engineering	Core RAG pipeline implementation (embeddings, FAISS, retrieval logic). PDF/Text extraction, Chunking strategy comparison, Technical Report - Data and Technical Approach sections.
Member 2	Front-end & Deployment/Evaluation	Streamlit UI development, OCR integration and image processing pipeline. Evaluation framework implementation (P@k, MRR calculation), Technical Report - UI and Evaluation sections.

8. Risks and Limitations

The Personal Study Notes Searcher (PSNS) effectively addresses the core retrieval problem but involves several risks and limitations that may affect performance and user experience.

8.1 Risks

8.1.1 OCR Errors

Handwritten or low-quality images may produce inaccurate OCR results, leading to poor embeddings and irrelevant retrievals.

Mitigation: Apply preprocessing (contrast enhancement, thresholding), test multiple OCR engines, and allow users to edit extracted text.

8.1.2. Embedding Model Limitations

General-purpose embeddings may not fully capture technical TU Wien content such as formulas or algorithms.

Mitigation: Evaluate multiple models using metrics like Precision@k and MRR.

8.1.3. Retrieval Inaccuracy

Ambiguous queries or overlapping course content may lead to irrelevant results.

Mitigation: Use query rewriting, course/semester metadata filters, and hybrid search.

8.1.4. Computational Constraints

Embedding large or image-heavy documents can be slow on limited devices.

Mitigation: Use batching, caching, and incremental embedding; inform users about expected processing time.

8.1.5. LLM Hallucinations

Summaries may include unsupported or incorrect statements.

Mitigation: Enforce strict RAG context and include chunk metadata in final outputs.

8.2 Limitations

- **Single-User System:** Supports only local, individual use—no collaboration or cloud syncing.
- **No Automatic Note Syncing:** Files must be uploaded manually; TUWEL, OneDrive, Google Drive, or Notion integration is not included.
- **Limited Diagram/Formulas Understanding:** Complex diagrams and formulas may not be fully recognized or searchable.

- **No Advanced Knowledge Modeling:** Features such as topic modeling or knowledge graph generation are outside scope.
- **Dependent on Input Quality:** System performance relies heavily on clear, well-structured study materials.

9. Scope Constraints and Exclusions

To keep the project feasible within the limited team size and time (2 members, ~320 hours), several features are explicitly excluded. The focus remains solely on semantic retrieval and summarization of personal study materials.

9.1 Out-of-Scope Features

- **Multi-User or Collaborative Use:** The system is strictly single-user; no shared workspaces, syncing, or collaborative dashboards.
- **Cloud Sync or Automatic Import:** No integrations with TUWEL, Google Drive, OneDrive, Dropbox, Notion, or Obsidian. Users must upload files manually.
- **Real-Time Indexing:** PSNS will not watch folders or auto-update indexes. All indexing is manual.
- **Advanced Formula/Diagram Interpretation:** The system will not extract LaTeX from handwriting, interpret diagrams, perform symbolic math, or convert complex expressions.
- **Knowledge Graph Features:** No dynamic knowledge graph, entity linking, clustering, or topic modeling.
- **Mobile App Development:** No Android/iOS app; only a desktop Streamlit interface.
- **Long-Term Personalization:** The system will not learn user preferences or adapt retrieval strategies.
- **High-Performance or Enterprise Scaling:** Not designed for large institutional datasets, huge repositories, or multi-user concurrency.

9.2 Technical Limitations

- **GPU Acceleration:** Only CPU-based OCR and embeddings are supported for portability.
- **Proprietary APIs (Except LLM Inference):** No paid OCR or vision APIs beyond the required LLM.
- **Production-Grade Security or Full Error Handling:** Only basic error handling; no advanced authentication.

9.3 Scope Summary

The project is limited to the core pipeline:

Upload → Extract → Chunk → Embed → Retrieve → Summarize

All other advanced or peripheral functionality remains outside the scope.