

Notation tool application – Notely

Software engineering Project

Project development process

Metropolia University of Applied Sciences

Information and communication technologies

Software engineering

Software project

27.08.25

# **1. Project overview**

1.1 Problem summary

1.2 Audience

1.3 Main Features

1. **Objectives**
2. **Scope and deliverables** 
   1. Included
   2. Excluded
   3. Deliverables
3. **Timeline** 
   1. Phase 1 Planning and analysis
   2. Phase 2 Design and Architecture
   3. Phase 3 Core Development
   4. Phase 4 Advanced Features
   5. Phase 5 Testing and quality assurance
   6. Phase 6 Documentation and deployment
   7. Milestones
4. **Resource allocation** 
   1. Team members and roles
   2. Software and tools
   3. Technology stack
   4. External resources
5. **Risk management**
   1. **High priority risks**
   2. Medium priority risks
6. **Testing and quality assurance**
   1. Testing strategy
   2. Success criteria
   3. Testing tools and frameworks
7. **Documentation and reporting**
   1. Documentation plan
   2. Progress reporting
8. **Conclusion**

1. Project overview

This document covers the purpose, chosen methodologies, development process, and final outcomes of the “Notely” our note taking application for software engineering students and people who work in the IT- industry.

* 1. Problem Summary

ICT and Computer Science students often find it challenging to take notes in clear and efficient way. The existing features of general note-taking applications are not well suited for creating programming notes. Code snippets attached to the notes are often poorly formatted, which makes them look messy and reduces writing efficiency.

* 1. Target audience

Primary users of the app are ICT and Computer Science students in universities, and other educational centers. Also, educational staff and lecturers support the students’ learning. Also skilled technology users who prefer effective and performance efficient applications, with clear and intuitive user interface design.

* 1. Main Features
* Core note management features for adding, editing and deleting notes, with included text formatting options.
* Organization tools for custom categories of notes.
* Search functionality for filtering by keywords.
* Multi-tab interface for having multiple pages of notes open at the same time.
* Collaborative features for real time sharing with an invitation.
* Code integration for formatting pasted code into notes, with proper syntax highlighting.
* Multimedia support for Images and diagrams.

1. Objectives

Our objectives are for at least 40% usage among the test group within 2 weeks of deployment. And to maintain user satisfaction scores of 70% or higher through the testing phase. We want to deliver a responsive application with low load times down to under 2 seconds. For online and collaborative features, we want to provide editing with multiple users at the same time.

While the application is developed, we want to keep the application as open source for broader community adoption. We also want to see a note taking increase for programming students by 50% compared to traditional methods.

1. Project scope and deliverables

3.1. Included

* Web based note taking
* Collaborative features
* Code pasting integration
* Image attachment
* User authentication
* Search
* Responsiveness
* Basic documentation

3.2. Deliverables

1. Functional prototype
2. Complete application
3. User documentation
4. Technical documentation
5. Testing reports
6. Timeline

4.1. Phase 1

Planning and analysis will be mostly done in sprint 1 between 18.08.25 – 09.09.25.

* Requirements gathering and analysis.
* Technology stack finalization
* Project setup
* Team role assignments

4.2. Phase 2

Implementing frontend and unit tests will be implemented in sprint 2 between 09.09.25 – 23.09.25

* Ui and Ux implementations for frontend
* Database schema design
* System architecture planning
* API endpoint planning
* Unit tests

4.3. Phase 3

Implementation of backend services and Jenkins implementation in sprint 3 23.09.25 – 07.10.25

* Backend API implementation
* Basic authentication
* Jenkins implementation
* Basic connection with frontend

4.4. Phase 4

Implementation of database and extending product functionality, docker implementation and finishing of documentation in sprint 4 between 07.10.25 – 21.10.25

* Database implementation
* Docker implementation
* Documentation and repository finalization
* Final fixing and testing
* Analysis and reporting

4.7. Key milestones

1. Project plan submission

2. Design specifications

3. Functional prototype

4. Testing phase completion

5. Final product delivery

1. Resource Allocation

5.1. Team members and roles

5.2. Software and tools

5.3. Technology stack

5.4. External resources

1. Risk management

6.1. High priority risks

6.2. Medium priority risks

1. Testing and quality assurance

7.1. Testing strategy

7.2. Success criteria

7.3. Testing tools and frameworks

1. Documentation and reporting

8.1 Documentation plan

8.2 Progress reporting

1. Conclusion