Tizkit

AI Powered Visual LaTeX Editor Platform

Software Requirements Specification

Project Name: Tizkit - Visual LaTeX Editor Platform
Version: 1.0
Iteration: 1
Date: November 10, 2025
Authors: H.M. Mehedi Hasan (Roll-13), Abu Bakar Siddique (Roll-47), Abantika Paul (Roll-21)
Document Type: Software Requirements Specification

Course: Software Engineering

Contents

1 Introduction **2**
1.1 Purpose 2
1.2 Scope 2

2 Overall Description **3**
2.1 Use-Case Model Survey 3
2.1.1 Use Case 1: Generate LaTeX Table from Visual Editor 3
2.1.2 Use Case 2: Create TikZ Diagrams Visually 4
2.1.3 Use Case 3: Convert Image to LaTeX using OCR 5

1 Introduction

This document specifies the **Software Requirements Specification (SRS)** for the **Tizkit** platform, developed in accordance with the **Rational Unified Process (RUP)** and **IEEE 830-1998** standards. It defines the systems purpose, scope, and primary functions using a use-casedriven approach, ensuring a shared understanding among all stakeholders.

1.1 Purpose

The purpose of this SRS is to describe the functional and non-functional requirements of the Tizkit system. It outlines how users interact with the platform and defines constraints necessary for design, development, and validation. The document serves as a reference for developers, testers, and project evaluators to ensure that implementation aligns with user needs and project goals.

1.2 Scope

Tizkit is a web-based visual LaTeX editor that simplifies document creation through interactive tools. It enables userssuch as students, researchers, and educatorsto visually build LaTeX tables, TikZ diagrams, and extract text from images using OCR and AI.

The scope of this document includes:

- Functional requirements derived from three primary use cases:
    1. Generate LaTeX Table from Visual Editor
    2. Create TikZ Diagrams Visually
    3. Convert Image to LaTeX using OCR and AI

2 Overall Description

2.1 Use-Case Model Survey

2.1.1 Use Case 1: Generate LaTeX Table from Visual Editor

Use Case ID	UC-01
Use Case Name	Generate LaTeX Table from Visual Editor
Priority	High
Status	Implemented
Primary Actor	Researcher / Student / Academic Writer
Preconditions	User is logged in and has access to the workspace.
Main Flow	1. User opens the Table Editor interface. 2. User adds, removes, or modifies rows and columns. 3. User enters cell data and customizes formatting. 4. System generates LaTeX code dynamically. 5. User previews and exports the LaTeX code.
Alternative Flow	If invalid data is entered, an error tooltip suggests a fix.
Exception Flow	If export fails, system retries or asks the user to check the network.
Postconditions	A valid ‘.tex‘ snippet for the table is generated and ready to copy/export.
Extension Points	Integrates with Workspace Save and AI Code Refiner modules.

Table 1: Use Case UC-01: Generate LaTeX Table from Visual Editor

2.1.2 Use Case 2: Create TikZ Diagrams Visually

Use Case ID	UC-02
Use Case Name	Create TikZ Flowchart from Visual Canvas
Priority	High
Status	Implemented
Primary Actor	Researcher / Engineer / Student
Preconditions	User is logged in and the canvas editor is accessible.
Main Flow	1. User opens the TikZ Diagram Editor. 2. User drags and drops shapes such as process, decision, input/output blocks. 3. User connects the blocks using arrows. 4. System generates corresponding TikZ code in real time. 5. User previews or copies the TikZ code for LaTeX integration.
Alternative Flow	If user adds overlapping elements, a warning message appears for correction.
Exception Flow	If rendering fails, system displays TikZ Compilation Error with details.
Postconditions	A valid TikZ diagram code is generated and can be exported.
Extension Points	Integrates with Diagram Templates and AI Auto-Layout features.

Table 2: Use Case UC-02: Create TikZ Flowchart from Visual Canvas

2.1.3 Use Case 3: Convert Image to LaTeX using OCR

Use Case ID	UC-03
Use Case Name	Extract and Convert Text from PNG Image
Priority	Medium
Status	Implemented
Primary Actor	Student / Academic Writer / Document Creator
Preconditions	User has uploaded a PNG image containing readable text.
Main Flow	1. User uploads a PNG image to the converter. 2. System performs OCR using integrated OCR API. 3. Extracted text is shown in the text editor. 4. System optionally converts the raw text into LaTeX syntax. 5. User can copy, edit, or download the LaTeX version.
Alternative Flow	If OCR confidence is low, the system highlights uncertain text for user review.
Exception Flow	If image upload fails or API times out, user receives an alert message.
Postconditions	Converted LaTeX text or plain extracted text is available for further editing.
Extension Points	Integrates with AI Text Formatter and Latex Snippet Generator.

Table 3: Use Case UC-03: Extract and Convert Text from PNG Image