Task 1.1

Parsing/analysing UDA Document

HEADERS:

- The Gazette of the Democratic Socialist Republic of Sri Lanka

- ශී ලංකා පජාතාන්තික සමාජවාදී ජනරජෙය් ගැසට් පතය

- අති විෙශෂ

-

-

NUMBERED\_CLAUSES:

- 05.07.2021

- 05.07.2021

- 1. This regulation may be cited as the Urban Development Authority Planning & Development Regulations 2021.

- 2. By virtue of this regulation, the Extraordinary Gazette notification No. 392/9 dated 10.03.1986, No. 821/19 dated

- 01.06.1994, No. 935/6 dated 06.08.1996, No. 1068/13 dated 24.02.1999, No. 1459/20 dated 23.08.2006, No.

DEFINED\_TERMS:

- This Gazette Extraordinary can be downloaded from

-

-

- a.

- b.

MEASUREMENTS:

- 15.0 m

- 7.5 cm

- 5.5 cm

- 6.0 m

- 12.0 m

BULLET\_POINTS:

- (General) are revoked and is replaced by the New Planning & Development Regulations / Orders attached hereto.

- (1) A Preliminary Planning Clearance (PPC) for any Development Activity, specified under Section 29 of the

- (2) The PPC shall be in writing and shall refer to the survey plan, the relevant development plan and planning

- (3) A PPC shall be obtained from the Authority for any development activity as prescribed hereunder:

- (4) Any Developer or his Agent who is desirous of obtaining a PPC shall make an application to the Authority

TABLES:

Found 83 tables

Task 1.2

Extraction Technologies and Libraries

The 2021 General Regulations PDF consists of various data in different formats, which should be extracted first before being added to the database. These data types include:  
1. Raw data (in the form of text)

2. Structured data (from tables, headings and numbered lists)

3. Structured text (includes data with structured numbering or legal terms)

A. Extracting Raw Data /Text extraction from PDFs

1. PyMuPDF:

1. Can extract and identify headers,font styles, numbered clauses and bullet points.
2. Retains font size, boldness, italics, and colors, which helps in identifying headers, key terms, and section numbering.
3. Scanned images in the PDF can be converted to image which may be passed through Tesseract OCR or OpenCV for text recognition.
4. Works well with Regex to extract numbered regulations.

B. Extracting Structured Data from PDFs

1. PDF Plumber:

1. Useful when working with PDFs that have scattered tables where boundaries are unclear.
2. Can extract tabulated data from the PDF without losing the structure of the PDF.
3. Position based text extraction to identify columns in the table.
4. Can extract data from footnotes or annotations in the document.
5. Also works well with Regex to extract structured regulation information.

C. Extracting Structured Text from PDFs using NLP

1. spaCy:

1. Recognizes structured text elements such as measurements or defined legal terms.
2. Detects headings, numbered sections and legal terms.
3. Identifies entities such as dates, locations, etc using NER (Name Entity Recognition)
4. Keyword extraction is possible.
5. Processes extracted text from PyMuPDF and PDF Plumber and works well with Regex.

2. Regex:

1. Identifies structured patterns in text and bullet points.
2. Can extract specific clauses, articles, section numbers, numbered regulations, dates/monetary values and cross-references.
3. Can be integrated into a spaCy NLP pipeline to enhance the extraction, segmentation, and structuring of the PDF.

D. Computer Vision Technologies for Regulation Extraction

1. OpenCV:

1. Can detect table boundaries and text positioning in tables of the PDF.
2. Can extract handwritten and printed text.

In summary:

1. PyMuPDF - Extract raw text from the Regulation PDF

2. Regex and SpaCy - Identifies and extracts headings clauses and measurements (categorizes regulations)

3. PDF Plumber - Extracts tabulated data

4. OpenCV - Extracts text from scanned images in the Regulation PDF.

Task 2.0

Structuring Database

### \*\*Analysis and Structuring of the 'Regulations Database' for the Web Application\*\*

Based on the provided \*\*Project Proposal\*\* and \*\*Sri Lanka's General Regulations Document\*\*, I have analyzed the structure of the regulations and the best way to develop the \*\*Regulations Database\*\* to support the "Regulations Filtering" feature.

---

## \*\*1. Analysis of the Regulations Document Structure\*\*

The \*\*2021 General Regulations\*\* document follows a structured hierarchy:

1. \*\*Legal Basis and Authority\*\*

- Cited under the \*\*Urban Development Authority Law No. 41 of 1978\*\*.

- Mentions prior amendments and revocations of previous regulations.

2. \*\*Categorization of Regulations\*\*

- \*\*Preliminary Planning Clearance (PPC)\*\*: Requirements for obtaining approval before any land development.

- \*\*Land Subdivision, Amalgamation, and Development\*\*: Guidelines on minimum plot sizes, access roads, and open spaces.

- \*\*Construction, Alteration, and Renovation of Buildings\*\*: Building permits, structural requirements, and safety measures.

- \*\*Green Building Standards\*\*: Conditions for obtaining a \*\*Green Building Certificate\*\*.

- \*\*Planning, Computation, and Designing of Buildings\*\*: Floor area ratios, zoning guidelines, and construction limitations.

- \*\*Special Conditions & Exemptions\*\*: Cases where exceptions can be granted.

3. \*\*Regulations Format\*\*

- Each regulation has:

- \*\*Clause Number\*\* (e.g., "Regulation 1", "Regulation 4(1)")

- \*\*Section Title\*\* (e.g., "Preliminary Planning Clearance")

- \*\*Detailed Description\*\* (word-for-word legal text)

- \*\*Applicable Conditions\*\* (e.g., location-based, zoning requirements)

- \*\*Referenced Documents or Authorities\*\* (e.g., Gazette Numbers, prior amendments)

---

## \*\*2. Suggested Database Structure\*\*

To efficiently filter regulations based on \*\*building type, region, scale, and additional conditions\*\*, the database should follow a \*\*relational structure\*\* with multiple linked tables.

### \*\*Entity-Relationship Model (ER Model)\*\*

The database should have at least \*\*four core tables\*\*:

### \*\*1. Regulations Table\*\*

| Column Name | Data Type | Description |

|-----------------|---------------|-------------|

| \*\*regulation\_id\*\* | INT (Primary Key) | Unique ID for each regulation |

| \*\*clause\_number\*\* | VARCHAR | Regulation clause reference (e.g., "Regulation 4(1)") |

| \*\*title\*\* | TEXT | Section title of the regulation |

| \*\*full\_text\*\* | TEXT | Full legal text (word-for-word) |

| \*\*gazette\_number\*\* | VARCHAR | Gazette number for reference |

| \*\*last\_updated\*\* | DATE | Date of last amendment |

### \*\*2. Regulation Categories Table\*\*

| Column Name | Data Type | Description |

|-----------------|---------------|-------------|

| \*\*category\_id\*\* | INT (Primary Key) | Unique ID for regulation category |

| \*\*category\_name\*\* | VARCHAR | Type of regulation (e.g., "Building Regulations", "Zoning Laws") |

### \*\*3. Regulation Filters Table\*\* (Links regulations to filtering criteria)

| Column Name | Data Type | Description |

|-----------------|---------------|-------------|

| \*\*filter\_id\*\* | INT (Primary Key) | Unique ID for filtering category |

| \*\*regulation\_id\*\* | INT (Foreign Key) | Links to Regulations Table |

| \*\*region\*\* | VARCHAR | Specific region/district (e.g., "Colombo", "Kandy") |

| \*\*building\_type\*\* | VARCHAR | E.g., "Residential", "Commercial", "Industrial" |

| \*\*building\_scale\*\* | VARCHAR | E.g., "Less than 1000m²", "More than 5 floors" |

### \*\*4. Index & References Table\*\*

| Column Name | Data Type | Description |

|-----------------|---------------|-------------|

| \*\*index\_id\*\* | INT (Primary Key) | Unique ID for each index entry |

| \*\*regulation\_id\*\* | INT (Foreign Key) | Links to Regulations Table |

| \*\*clause\_number\*\* | VARCHAR | Clause number for easy lookup |

| \*\*cross\_references\*\* | TEXT | Any other clauses referenced |

---

## \*\*3. Querying the Database to Filter Regulations\*\*

To retrieve relevant regulations \*\*word-for-word\*\* based on user input, queries should \*\*filter\*\* based on building type, location, and scale.

### \*\*Example Queries\*\*

1. \*\*Filter by region, building type, and scale\*\*

```sql

SELECT r.clause\_number, r.full\_text, r.gazette\_number

FROM regulations r

JOIN regulation\_filters rf ON r.regulation\_id = rf.regulation\_id

WHERE rf.region = 'Colombo'

AND rf.building\_type = 'Residential'

AND rf.building\_scale = 'Less than 1000m²';

```

2. \*\*Get all regulations that mention a specific keyword (e.g., "parking requirements")\*\*

```sql

SELECT clause\_number, full\_text

FROM regulations

WHERE full\_text LIKE '%parking requirements%';

```

3. \*\*Retrieve regulations applicable to industrial buildings in Kandy\*\*

```sql

SELECT r.clause\_number, r.full\_text, r.gazette\_number

FROM regulations r

JOIN regulation\_filters rf ON r.regulation\_id = rf.regulation\_id

WHERE rf.region = 'Kandy' AND rf.building\_type = 'Industrial';

```

---

## \*\*4. Best Technologies for Database Implementation\*\*

Since the project must use \*\*free and open-source technologies\*\*, I recommend:

1. \*\*Database Management System\*\*

- \*\*PostgreSQL\*\* (Best for complex queries, indexing)

- \*\*MySQL\*\* (More beginner-friendly)

- \*\*SQLite\*\* (Lightweight option for simple applications)

2. \*\*Web Application Integration\*\*

- \*\*Backend Framework:\*\* \*\*Django (Python) or Node.js (JavaScript)\*\*

- \*\*Frontend:\*\* \*\*React.js or Vue.js\*\* for UI

- \*\*REST API:\*\* Create API endpoints to query and return results in \*\*JSON format\*\*.

3. \*\*Hosting & Deployment\*\*

- \*\*Cloud-based:\*\* \*\*Supabase (for PostgreSQL)\*\*, \*\*Railway.app\*\*, or \*\*Google Firebase\*\*

- \*\*Self-hosted:\*\* Host on \*\*AWS Free Tier\*\*, \*\*Heroku (free plan)\*\*, or \*\*DigitalOcean (low-cost VPS)\*\*

4. \*\*Text Search Optimization\*\*

- Use \*\*PostgreSQL Full-Text Search (FTS)\*\* for fast filtering.

- Create an \*\*Elasticsearch\*\* index if advanced search capabilities are needed.

---

### \*\*Summary of Implementation Plan\*\*

\*\*Database Structure:\*\* \*\*Regulations, Filters, Categories, and Index Tables\*\*

\*\*Filtering Queries:\*\* Based on \*\*region, building type, and scale\*\*

\*\*Word-for-Word Retrieval:\*\* Ensures compliance with \*\*government documentation\*\*

\*\*Technology Stack:\*\* \*\*PostgreSQL + Django/Node.js + React.js\*\*