**1. High-Level Architecture (Block Diagram)**

This block diagram shows the core components of the Resume Customization Tool’s MVP. It focuses on three primary modules: Frontend, Backend, and External Services.

**Diagram Components:**

* **Frontend**
  + User Interface (Web App)
    - Resume Upload (DOCX/PDF)
    - Job Description Input (Text/Upload)
    - Display Customized Resume
    - Download Customized Resume (DOCX/PDF)
* **Backend**
  + API for Resume Parsing (Basic parsing of DOCX and PDF formats)
  + Job Description Parser (Basic text extraction and keyword matching)
  + Resume Customization Engine (Simple keyword matching algorithm for MVP)
  + File Handling (Ensuring format consistency across uploads/downloads)
* **External Services**
  + Hosting Platform (e.g., Google Cloud, AWS, or Azure for deployment)
  + Authentication (Basic user session management if necessary)

**2. Workflow Design (Flowchart)**

A flowchart will visually represent the user's interaction with the system, highlighting the flow from uploading a resume to downloading the customized version.

**Flowchart Process:**

1. **Start**
2. **User Uploads Resume**
   * Format check (DOCX/PDF)
   * Parse resume
   * If invalid format → show error.
3. **User Inputs Job Description**
   * Text or Upload (DOCX/PDF)
   * Parse job description
   * If invalid → show error.
4. **Basic Customization Engine**
   * Keyword matching between resume and job description.
   * Prioritize relevant sections.
5. **Generate Customized Resume**
   * If successful → proceed to download.
   * If failure → retry process.
6. **User Downloads Customized Resume**
7. **End**

**3. Message Sequence Chart (MSC)**

The MSC will illustrate how different components of the system communicate during the resume customization process.

**Message Sequence:**

1. **User → Frontend**
   * User uploads base resume and provides job description.
2. **Frontend → Backend**
   * Resume and job description are sent to the backend for processing.
3. **Backend → Resume Parser**
   * The backend sends the resume file to the Resume Parsing component for text extraction.
4. **Resume Parser → Backend**
   * Resume data is returned to the backend.
5. **Backend → Job Description Parser**
   * The backend sends the job description to the Job Description Parser for extracting key requirements.
6. **Job Description Parser → Backend**
   * Parsed job description is returned to the backend.
7. **Backend → Resume Customization Engine**
   * The backend passes both the parsed resume and job description to the Customization Engine for keyword matching and reordering of relevant sections.
8. **Resume Customization Engine → Backend**
   * Customized resume is returned.
9. **Backend → Frontend**
   * The backend sends the customized resume to the frontend for display.
10. **User → Frontend**
    * User downloads the customized resume.

**4. UI Wireframing**

You can create simple wireframes that visualize the user interface of the tool. For a PowerPoint slide, you can either sketch the wireframes or use tools like Figma, Balsamiq, or PowerPoint’s own drawing features.

**Key UI Screens:**

* **Home/Start Screen**
  + Title: “Resume Customization Tool”
  + Buttons:
    - Upload Resume (DOCX/PDF)
    - Input Job Description (Text/Upload)
* **Customization in Progress Screen**
  + Loading indicator with a message: “Customizing your resume based on the job description...”
* **Result Screen**
  + Display Customized Resume
  + Button to Download Resume (DOCX/PDF)
* **Error Messages (if applicable)**
  + Simple modal or alert box for unsupported formats or invalid files.

**5. Slide Structure for PowerPoint**

Here’s how you can organize these elements into a PowerPoint presentation:

**Slide 1: Title Slide**

* **Title**: Resume Customization Tool MVP
* **Subtitle**: Automating Resume Customization Based on Job Descriptions

**Slide 2: Introduction**

* **Objective**: Provide an overview of the MVP functionality.
* **Key Features**:
  + Upload Resume
  + Input Job Description
  + Customization Engine
  + Download Customized Resume

**Slide 3: High-Level Architecture**

* Include the **block diagram** described above.

**Slide 4: Workflow Design**

* Present the **flowchart** and describe each step.

**Slide 5: Message Sequence Chart**

* Insert the **message sequence chart** and briefly describe how the system components interact.

**Slide 6: UI Wireframes**

* Display the key wireframes for the tool (upload screen, result screen, etc.)

**Slide 7: Conclusion**

* Summarize how the MVP focuses on core functionality.
* Mention future possibilities for scaling and enhancements.