

AtlascanApp

The project involves developing a robust cloud-native application that integrates OCR technology to automate the extraction of crucial information from identification documents. The application must be able to handle uploaded documents and camera feeds. The extracted data, especially the ID number, must be accurately transferred to a web form for a simpler and error-free user experience.

Technologies

1) Backend Technologies

The backend of the application will be developed using two frameworks: Spring and Flask. Both of these technologies are well-suited for handling the backend logic of the application. Spring is a robust, enterprise-level framework that provides high scalability and security. Flask, on the other hand, is a lightweight Python framework ideal for rapid development and prototyping, making it easier to implement OCR functionality. The choice of these two technologies provides flexibility and allows for scaling the application based on the complexity of the OCR processing and other backend tasks.



Flask



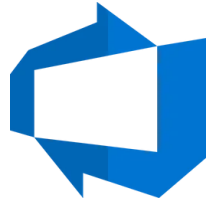
2) Frontend Technology

For the frontend, Android Studio will be used to build the mobile application. Android Studio is the official IDE for Android development, and it provides the necessary tools and frameworks for developing mobile apps efficiently. Given that the OCR app is targeted for use on mobile devices, Android Studio will allow for seamless integration with the camera feed and the user interface components required to scan and display data extracted from documents.



3) Project Management Tool

Azure DevOps will be used for project management and tracking. This tool provides an integrated suite for planning, tracking, and collaboration, making it easier to manage tasks, sprints, and resources. Its integration with version control and CI/CD pipelines ensures a smooth and efficient workflow, allowing for continuous development and delivery of the app.



4) CI/CD Tools

Jenkins will be used for continuous integration and continuous delivery (CI/CD). Jenkins is a widely adopted tool that automates building, testing, and deploying code, which helps to ensure that code changes are smoothly integrated and deployed without errors. This will allow for faster development cycles and more frequent releases of updates for the OCR app.



5) Documentation Tool

For documentation purposes, Microsoft 365 will be used. The Microsoft 365 suite provides a comprehensive set of tools, including Word, Excel, and PowerPoint, that allow for efficient documentation, collaboration, and project management. It's ideal for storing technical documentation, user guides, task descriptions, and other project materials that can be accessed and edited by the project team in real-time.



6) AI Model

For the AI model powering the OCR functionality, we will explore using models such as Tesseract or custom models built with TensorFlow or PyTorch. Tesseract is a popular open-source OCR engine that works well with scanned documents, while TensorFlow and PyTorch can be used for creating custom AI models that are more specific to the project's requirements, such as accuracy, language support, and document types. The final choice will depend on the project's scale and the accuracy required for OCR.



7) Deployment Technologies

For deployment, AWS and Kubernetes will be used. AWS provides scalable cloud infrastructure for hosting and managing the OCR application, while Kubernetes ensures that the application is containerized and can be deployed across multiple environments with ease. Kubernetes also ensures the application can scale according to demand, improving reliability and availability.



8) Meeting Tool

Google Meet will be used for virtual meetings and team collaboration. Google Meet is easy to use, secure, and integrates well with other Google tools like Gmail and Google Calendar, which facilitates scheduling and communication among team members.



9) Containerization Tool

Docker will be used for containerizing the application. Docker allows for creating lightweight, portable containers that can run the OCR app in any environment. This ensures consistency across development, testing, and production environments, making deployment more predictable and efficient.



10) Testing Tools

JMeter and JUnit will be used for testing the OCR application. JMeter is an open-source tool for performance and load testing, which will be critical in ensuring the scalability and efficiency of the application, especially when dealing with large amounts of document data. JUnit will be used for unit testing of individual components to ensure that the core functionality, such as OCR and data processing, is working correctly.



11) Quality Assurance

SonarQube will be used to ensure the code quality of the OCR application. SonarQube provides static code analysis to detect bugs, vulnerabilities, and code smells. By using SonarQube, the development team can maintain high code quality and ensure the application is reliable, secure, and easy to maintain.



12) Version Control

Git and GitHub will be used for version control. Git is a distributed version control system that tracks code changes and allows for collaboration on the project. GitHub will serve as the central repository for the project, making it easy for the development team to access the latest version of the code and collaborate effectively.

