Process Deliverable I

Given the feedback from our requirements elicitation, the following requirements were prioritized:

1. Centralized Access and Permissions Management
   Priority: High
   Obtaining access permission for different resources is constantly a major delay in the onboarding process. Many of those whom we interviewed stated that days have been wasted simply waiting for their access requests to be approved. Automating this process will immediately save our users time and improve productivity. In this way, new hires can have their accesses streamlined in a single, efficient step. This also ensures faster access to tools and reduces the need for back-and-forth communication between project managers and IT workers.
2. Automated Development Environment Setup
   Priority: High
   Feedback indicated that one of the more time-consuming tasks in the onboarding process, especially for software engineers, is setting up a development environment. This process often requires help from senior members, making it a task that affects productivity across a team.
3. Automated Codebase Documentation Overview
   Priority: Medium
   Feedback revealed that understanding the structure and functionality of large code bases was a challenge for many new hires. This is often due to either incomplete or outdated documentation. This becomes an even bigger problem when dealing with high-level architecture. An automated documentation overview would provide users with an immediate understanding of the project's main components and their relationships, without presenting too much information as to overwhelm the user. This feature will allow our users to reduce the time spent trying to understand how modules interact with one another, allowing developers to dice into code with context, reducing the overall onboarding time.
4. Task Progress Tracking System
   Priority: Medium
   New hires expressed a need for a clearer way of tracking the onboarding progress, as oftentimes they're left uncertain about which steps have been completed and which remain. A task tracking system would give our users a real-time overview of their progress, making it easier to prioritize tasks and measure completion. This feature also reduces the reliance on mentors or managers for guidance during the onboarding process.

Requirements Analysis

5 Non-Functional Requirements

- Usability
    - The system shall guide new users through onboarding tasks, enabling them to complete initial setup within 1 hour, with an average of 5-7 clicks per main task. Additionally, the interface shall score at least 75% in usability satisfaction tests conducted with the new users.
- Reliability
    - The system shall maintain 99% uptime during business hours (8am - 5pm, Monday - Friday), with scheduled maintenance limited to one day per month. In addition, any critical failures must be reported and recoverable within 24 hours, with automatic error logging and a restart option available to users for any non critical failure.
- Performance
    - The system shall have a response time of no more than 2 seconds for user interactions, support up to 50 concurrent onboarding sessions with a maximum 10% increase in setup times under load, and limit CPU usage to 70% and memory usage to 60% under typical operating conditions.
- Supportability
    - The system shall allow for updates and modifications with an average turnaround time or no more than 24 hours for critical issues reported by users, and it shall achieve a 95% first attempt resolution rate.
- Implementation / Constraints
    - The system shall be developed using Java and adhere to industry best practices for code quality, ensuring at least 80% code coverage from unit tests. It must be compatible with existing tools and technologies used within the organization, supporting a minimum of two versions of any required third-party libraries or frameworks

5 Functional Requirements

- Centralized Access and Permissions Management
    - The system must automate the process for requesting and granting access permissions for various resources to new hires, enabling them to access necessary tools and documentation in a single step.
- Automated Development Environment Setup

    - The system must automatically configure the development environment for new software engineers by setting up required repositories and permissions without requiring assistance from senior team members.
- Automated Codebase Documentation Overview
    - The system must generate an automated overview of the project’s codebase documentation, providing new hires with a clear understanding of the project's structure and key components.
- Integration with Existing Tools and Systems
    - The system must integrate with existing project management and communication tools to streamline notifications and updates regarding onboarding tasks and relevant project information.
- Task Progress Tracking System
    - The system must provide a task tracking feature that allows new hires to view their onboarding progress in real-time, including completed and outstanding tasks.

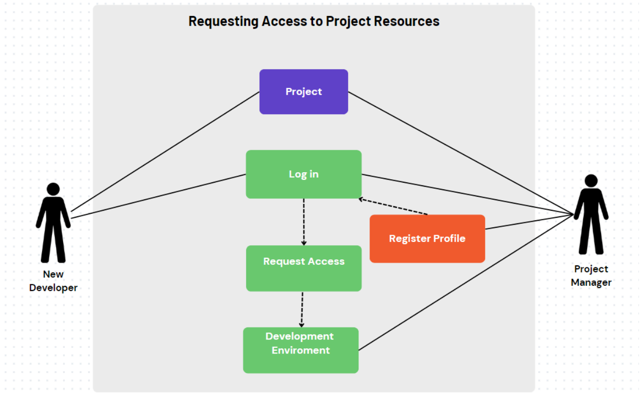
Use Cases

Use Case 1: Requesting Access to Project Resources

Actors: New Developer, Project Manager

Description: The new developer requests access to a group repository along with certain dev tools. Our system recognizes that the new hire is supposed to have access to those resources, since the project manager registered a profile for the new hire on our platform prior to their starting day. Thus, access to the specified resources are granted automatically.

Diagram:

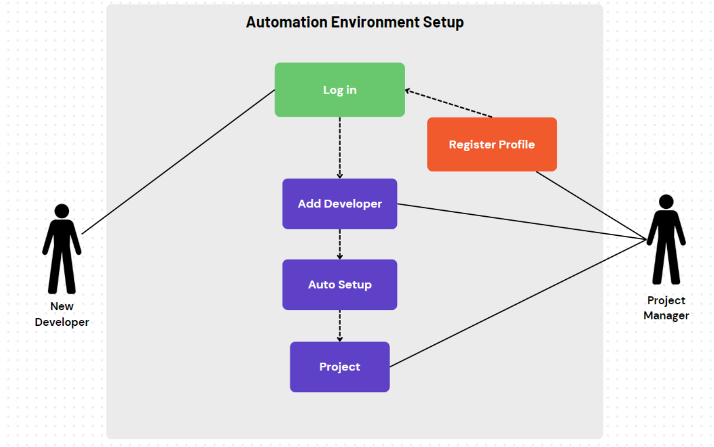


Use Case 2: Automation Environment Setup

Actors: New Developer, Project Manager

Description: The new developer triggers the environment setup when the project manager adds them to a new project. The system automatically installs and configures necessary softwares. This is because their profile was already registered in our platform by the project manager, so the system is aware of the resources needed to set up the dev environment.

Diagram:

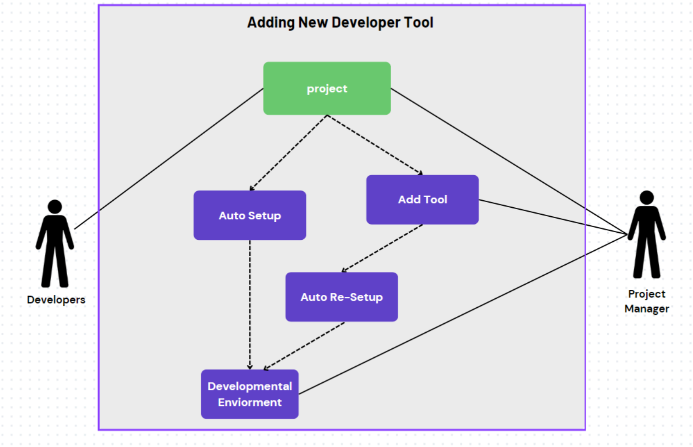


Use Case 3: Adding New Developer Tool

Actors: Project Manager, Developers

Description: The project manager wants to add a new developmental tool to the project environment. The manager can select a new tool and they agree too add it to the environment, the setups of all developers working on the project will automatically be updated to accommodate the new tool.

Diagram:

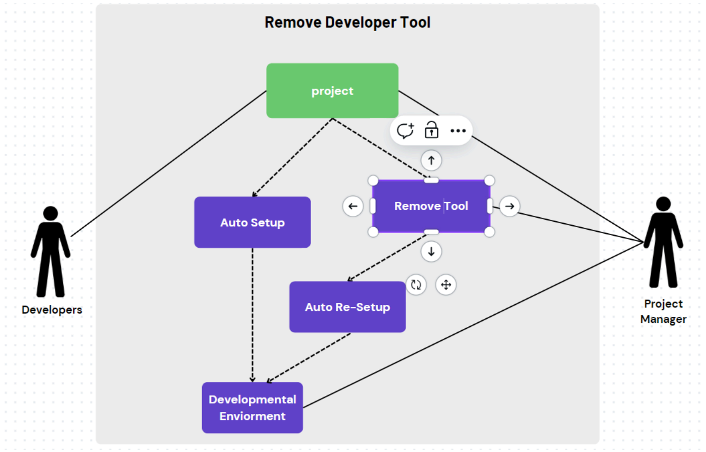


Use Case 4: Remove Developer Tool

Actors: Project Manager

Description: The project manager wants to remove an existing development tool from the project environment because it is unneeded. The manager can select the tool or tools they want to remove, and upon agreeing to the changes, the setups of all developers working on the project will automatically be such that they no longer have access to the tool. .

Diagram:

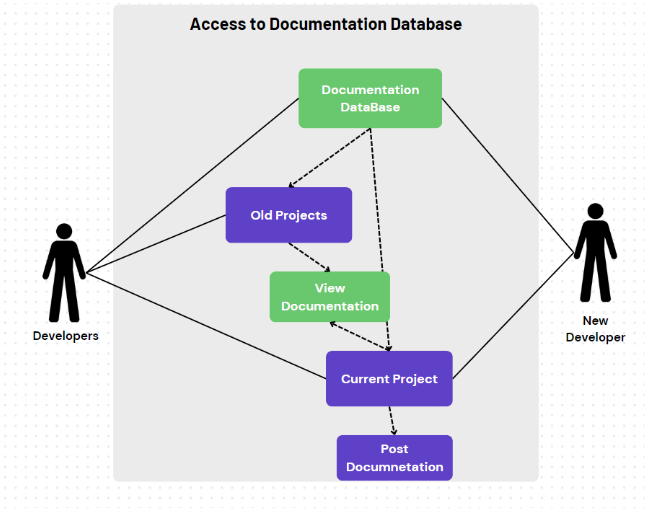


Use Case 5: Access to Documentation Database

Actors: New Developer, Developers

Description: New developers will be given access to a database that contains documentation to the project they are working on and all documentation of past projects so that they can understand what the project is about and study old documentation from their coworkers. They will be able to log into the database and search bases on project names and project contributors.

Diagram:



Requirements Specification

Write four user stories from the perspective of at least two different actors. Provide the acceptance criteria for these stories. For each user story mentioned above, estimate the amount of effort needed to complete relevant subtasks using function points. Explain your answer.

User Story 1: Requesting Access to Project Resources

As a new developer,
I want the system to automatically grant me access to required project resources,
so that I can begin working without delays.

Acceptance Criteria:

The new developer must be able to request access through the onboarding interface.
The system should check the project manager’s pre-registered profile for resource permissions.
Access should be granted automatically if the resources are pre-approved.
Notification of granted access should be sent to both the new developer and project manager.

Effort Estimation using Function Points:

Interface to request access: 3 FP
Resource access automation logic: 4 FP
Notifications system: 2 FP
Total: 9 FP
Explanation: The effort includes designing the user interface, automating access checks, and implementing a notification system.

User Story 2: Automated Development Environment Setup

As a project manager,
I want to ensure that the new developer's development environment is set up automatically,
so that they can start coding without manual configuration.

Acceptance Criteria:

The system must recognize the new developer’s profile and pre-configured environment requirements.
The automated setup must install the necessary software, clone repositories, and configure environment variables.
The process should be completed within an hour.
A status report must be available to the project manager.

Effort Estimation using Function Points:

Profile recognition and configuration logic: 4 FP
Automation script for environment setup: 6 FP

Status report generation: 3 FP
Total: 13 FP
Explanation: Setting up involves scripting installations, environment configuration, and a reporting tool to show setup status.

User Story 3: Access to Documentation Database

As a new developer,
I want to access the project's documentation database,
so that I can understand the project and refer to past project documentation.

Acceptance Criteria:

The system must allow developers to log into the documentation database.
Search functionality must be available to look up projects by name or contributor.
Access to relevant project documents must be seamless and secure.
The database must include documentation for both current and past projects.

Effort Estimation using Function Points:

Secure login system: 3 FP
Search functionality: 4 FP
Document access and viewing interface: 4 FP
Total: 11 FP
Explanation: Developing a secure login system, search capabilities, and a document viewing interface are needed for this functionality.

User Story 4: Task Progress Tracking System

As a new developer,
I want a task progress tracking system to guide me through my onboarding steps,
so that I can know what tasks are completed and which ones are still pending.

Acceptance Criteria:

The task tracking system must show a real-time status of completed, in-progress, and pending tasks.
Each task should have a description and expected completion time.
The system should provide notifications for upcoming deadlines or tasks that are overdue.
The progress tracking must be integrated with the onboarding dashboard for easy access.

Effort Estimation using Function Points:

Real-time task tracking interface: 4 FP

Task update and notification system: 3 FP

Task details display and completion status logic: 3 FP

Dashboard integration: 2 FP

Total: 12 FP

Explanation: This involves creating a user-friendly tracking interface, logic for task status updates, and notifications for a seamless onboarding experience.