Tooling Selection and Justification

Programming Languages and Frameworks

1. **JavaScript (React Native)**
    - Purpose: React Native enables the development of a cross-platform mobile application using a single codebase, supporting Android and iOS.
    - Justification: This framework is ideal for the project's focus on internal communication as it accelerates development and ensures feature parity across platforms. Its vast library ecosystem and community support make it a reliable choice.
2. **Firebase**
    - Purpose: A comprehensive backend platform offering real-time databases, authentication, file storage, and hosting.
    - Justification: Firebase simplifies the development of key functionalities such as user authentication, data synchronization, and document sharing. Its scalability supports the growing needs of small organizations and groups.

Integrated Development Environment (IDE)

3. **Visual Studio Code**
    - Purpose: A lightweight yet powerful code editor with extensive support for languages, debugging, and extensions.
    - Justification: VS Code's plugins for React Native and Firebase, along with its Git integration, streamline the development workflow, ensuring a smooth coding and debugging experience.

Version Control and CI/CD

4. **GitHub (Version Control System)**
    - Purpose: A platform for managing code repositories and enabling collaboration among team members.
    - Justification: GitHub ensures version control, code review, and seamless team collaboration, which are essential for a project involving multiple contributors.
5. **GitHub Actions (CI/CD)**
    - Purpose: Automates build, test, and deployment workflows directly within GitHub.
    - Justification: GitHub Actions provides a streamlined way to maintain code quality and quickly deploy updates, essential for iterative development and a consistent user experience.

Project Management Tools

6. **GitHub Projects**

- Purpose: A task management system integrated with GitHub repositories.
- Justification: This tool aligns perfectly with version control, allowing the team to manage tasks and track progress directly within their development environment.

Design Tools

7. **Figma**

    - Purpose: A collaborative design tool for creating UI/UX mockups and prototypes.
    - Justification: Figma supports real-time collaboration among team members, ensuring that the user interface aligns with project objectives and is intuitive for the target audience.

Collaboration and Documentation Tools

8. **Discord**

    - Purpose: A communication platform for real-time discussions, updates, and collaboration.
    - Justification: Discord facilitates seamless communication between team members, enabling quick issue resolution and brainstorming during development.

9. **Google Docs**

    - Purpose: A cloud-based platform for drafting, editing, and sharing documentation.
    - Justification: It allows the team to collaboratively work on project vision, scope, and specifications, ensuring alignment and clarity across all members.

Integration and Complementation

- ✓ React Native, Firebase, and VS Code form the core development stack, streamlining app functionality and UI implementation.
- ✓ GitHub integrates version control, project management, and CI/CD workflows, ensuring efficient collaboration and delivery processes.
- ✓ Figma complements React Native by providing a blueprint for UI design, directly influencing implementation.
- ✓ Discord and Google Docs support team communication and documentation, bridging gaps between development and planning phases.

Michał Wirkowski

Piotr Gruchała

Mateusz Klemann