from: manager@company.org
to: managersquared@company.org
subject: programming language for super-hot project.

Google's Golang fits quite well for the specifications provided and it stands as the best choice for the upcoming project. Golang's package system is the simplest to integrate. Once the language and toolchain is installed, it's easy to download and include external packages and libraries with a single command and import line, though the standard library is rich enough to provide most use-cases. Golang's built-in standard library includes: I/O packages ("bufio", "os") for reading from stdin ("os.Stdin"), writing to stdout ("os.Stdout"), communicating over socket-based connections ("net"), JSON and XML processing libraries ("encoding/json", "encoding/xml"), automated unit testing ("testing") alongside coverage evaluation and visualization packages ("cmd/cover"), and a plugin system for dynamic program loading ("plugin", added recently). The only project requirement left unfulfilled from the standard libraries is a GUI, but this is remedied through an external package allowing for an HTML UI for your app ("murlokswarm/app").

As for an IDE, the best choice is JetBrains' GoLand, as it comes with syntax highlighting, automatic test running for files changed, and quick documentation lookup for any method and its signature, whether from the standard library or user-defined within the project. Given that Go similarly requires less than a week of onboarding for competency, the two paired together lets developers move through onboarding quickly, minimizing overhead.

Golang allows for easy cross-compilation between systems, allowing for developers to work on whichever machine they prefer yet deploy on the specified Linux system with minimal conflict (provided the developers test any OS-specific components on the Linux system before proper deployment).

If you've got further inquiry or questions, feel free to schedule a time to talk further about my decision, and I'd be happy to explain in further detail.