

Type: Primary Feature	Title: Command Logic	Status: X	RI: E1-T2
Context: With all backend systems operational comes the essence of this undertaking: a command that initiates a scheduler [of 24h as preset] for sending poetry prompts acquired through a single query from a database that is read and/or manipulated within local caching that would otherwise update its cache through another query when integrity is violated.			

Core Decision: Create a one-off Discord command that timely posts poetry prompts in a specified channel using a database

Prospects	Decision Parameters	OTD Concerns
<ul style="list-style-type: none"> – A one-off command that schedules the sending of poetry prompts <ul style="list-style-type: none"> - Schedule is 24h default – Users must be authorized for the command – Query limit is 1 but effectively expands on caught exceptions – Implements Separation of concerns design principle <ul style="list-style-type: none"> - Displaying and specifying a channel are two separate concerns - Import display logic to the specifying logic 	<ul style="list-style-type: none"> – Ternary operation that gives access to the imported command <ul style="list-style-type: none"> - Parse interaction for command name, (e.g., is caller /setpromptchannel or /prompt?) - Vital if user would like to manually call the current day’s prompt – Require the user to have <i>Administrator</i> privileges to execute the command – Create a global state timer that dynamically updates within recursive timeouts – Prevent unnecessary DB operations <ul style="list-style-type: none"> - Safeguard duplicate prompt channel requests - Use upsert to modify existing records 	<ul style="list-style-type: none"> × Administrators are unable to modify required privileges/roles to execute the command × Administrators are unable to set multiple prompt channels × Administrators cannot adjust the time of day of which the prompts are sent <ul style="list-style-type: none"> - 07:00 PST is the agreed-upon target; consistency ensured