CSC 440 (001) Database Management Systems - Swimlanes
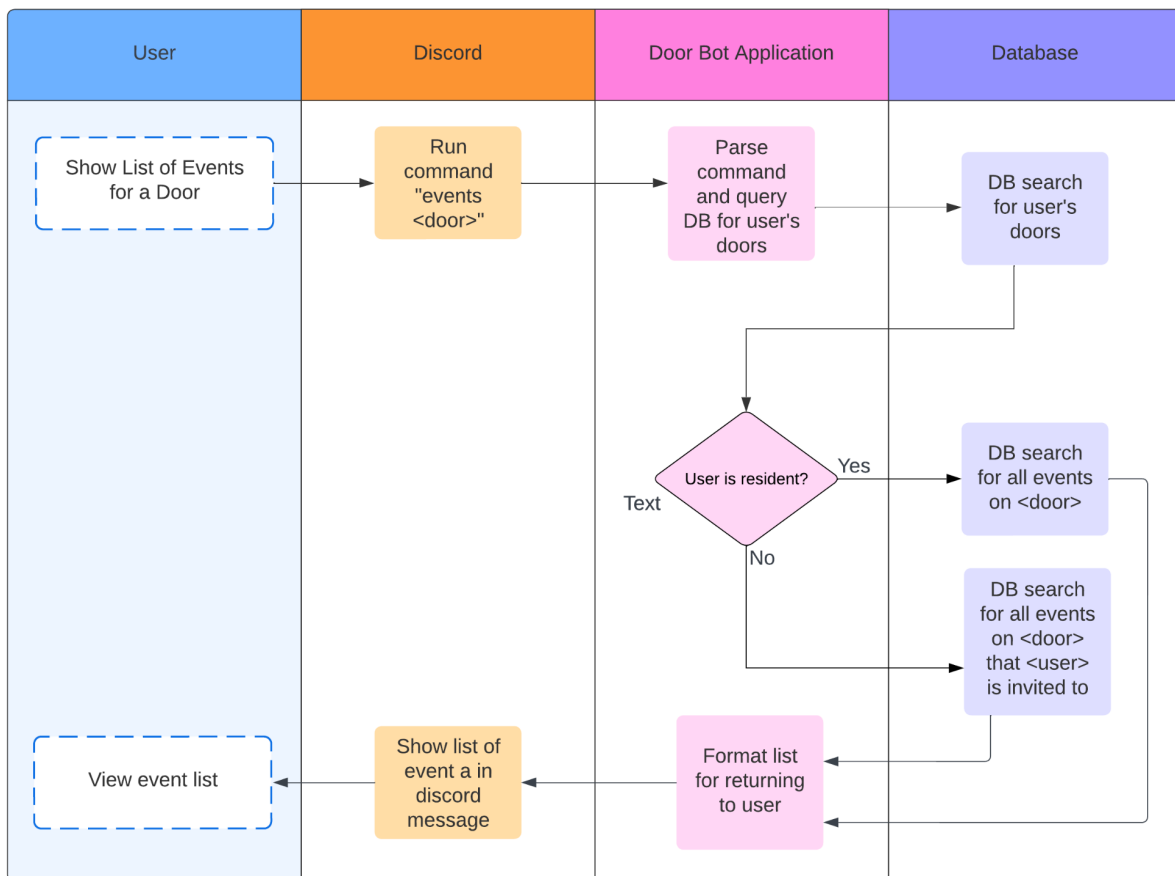
Team Chupacabra
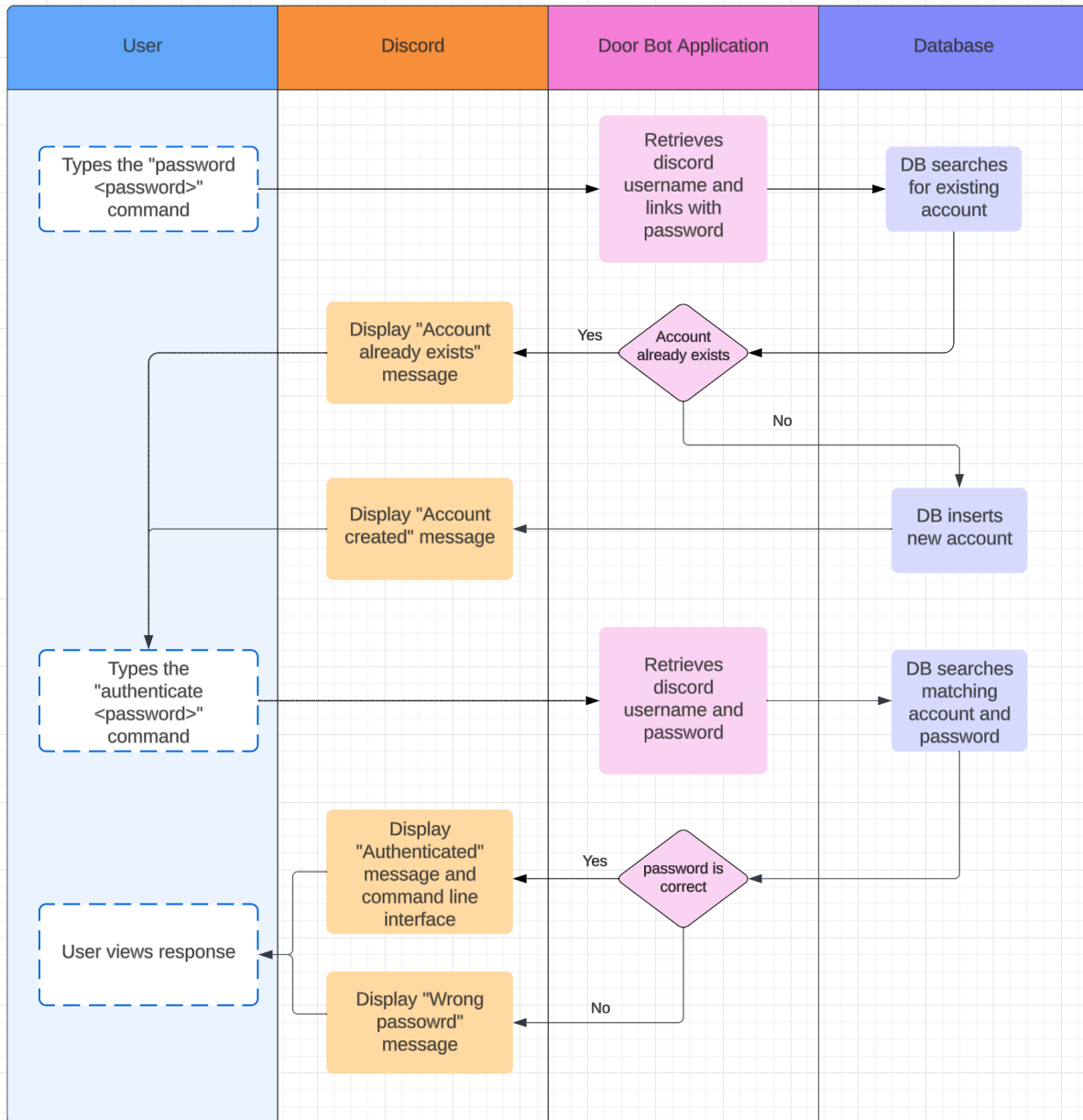
Caleb Rollins , Matthew Welker , Finn Bacheldor , Alex Hill , Jakari Robinson

List Events for a Door

User	Discord	Door Bot Application	Database

Show List of Events for a Door

Run command "events <door>"

Parse command and query DB for user's doors

DB search for user's doors

User is resident?

Text

Yes

No

DB search for all events on <door>

DB search for all events on <door> that <user> is invited to

View event list

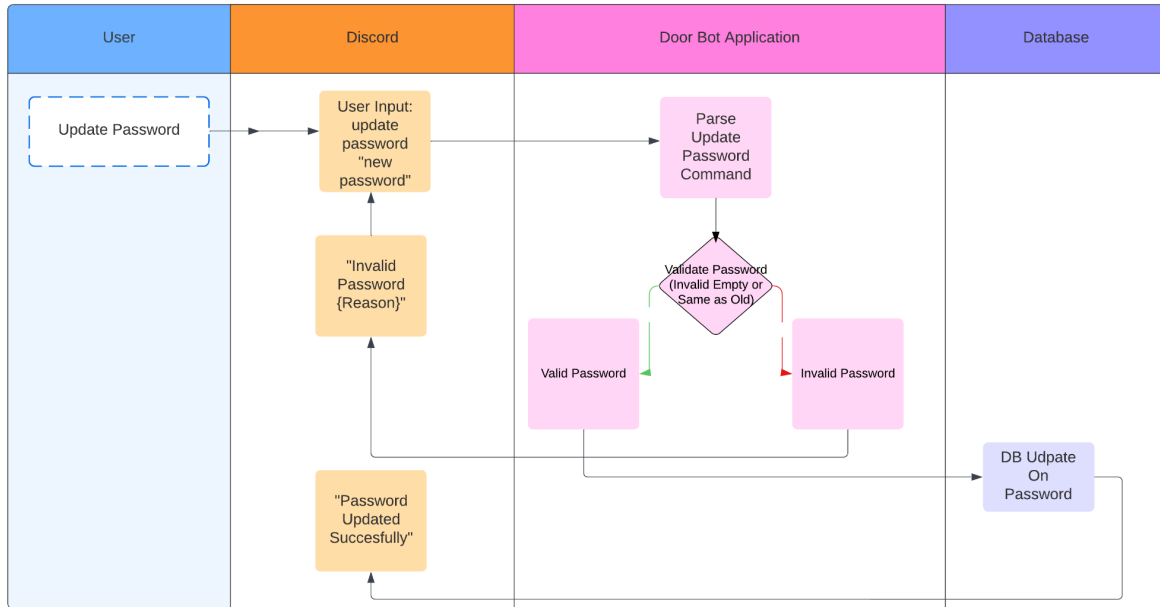Show list of event a in discord message

Format list for returning to user

Users can run a command to see what events are available for a door. If they are a resident, they can see all events for that door. Otherwise, they can only see events they are invited to.

Account creation and authentication

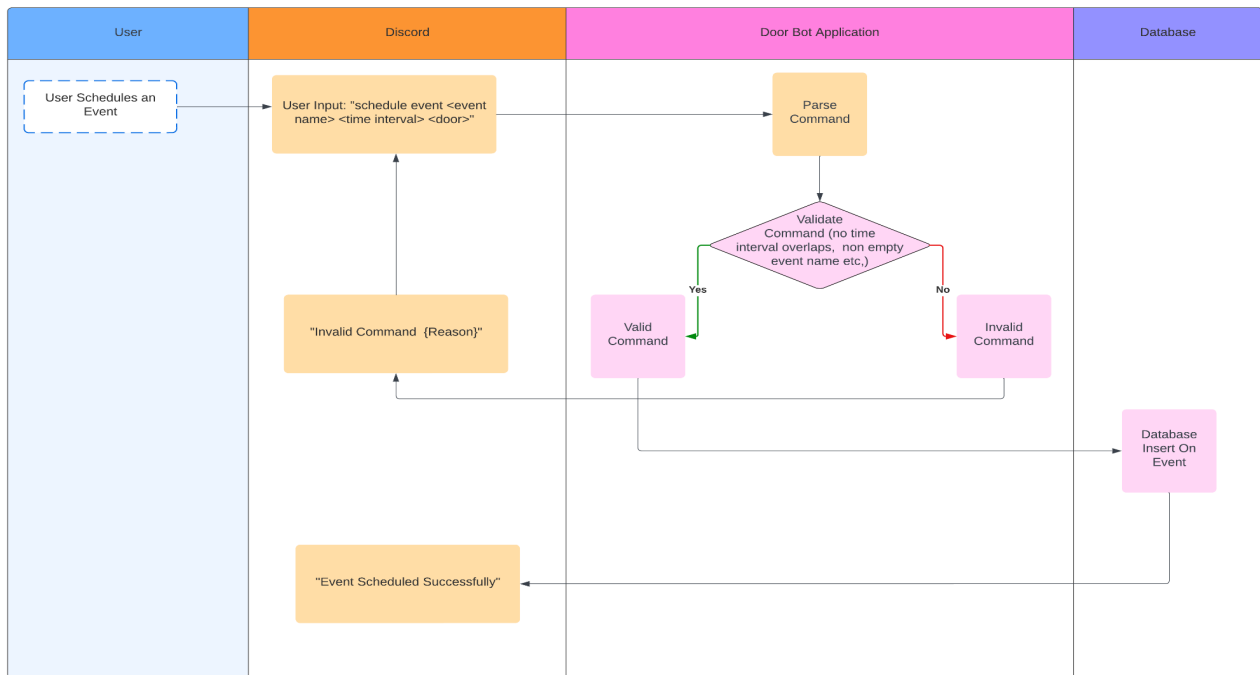| User	Discord	Door Bot Application	Database
Types the "password <password>" command		Retrieves discord username and links with password	DB searches for existing account
	Display "Account already exists" message	Account already exists — Yes	
	Display "Account created" message	No	DB inserts new account
Types the "authenticate <password>" command		Retrieves discord username and password	DB searches matching account and password
	Display "Authenticated" message and command line interface	password is correct — Yes	
User views response	Display "Wrong passowrd" message	No	

The user first tries to use the password command. If the username is already registered with an account the user is notified appropriately, otherwise a new account is created. To authenticate the user types in the "authenticate <password" command which checks if the password is correct then displays the command line interface if correct or the "Wrong password" message is incorrect.

Update Password Swinlane

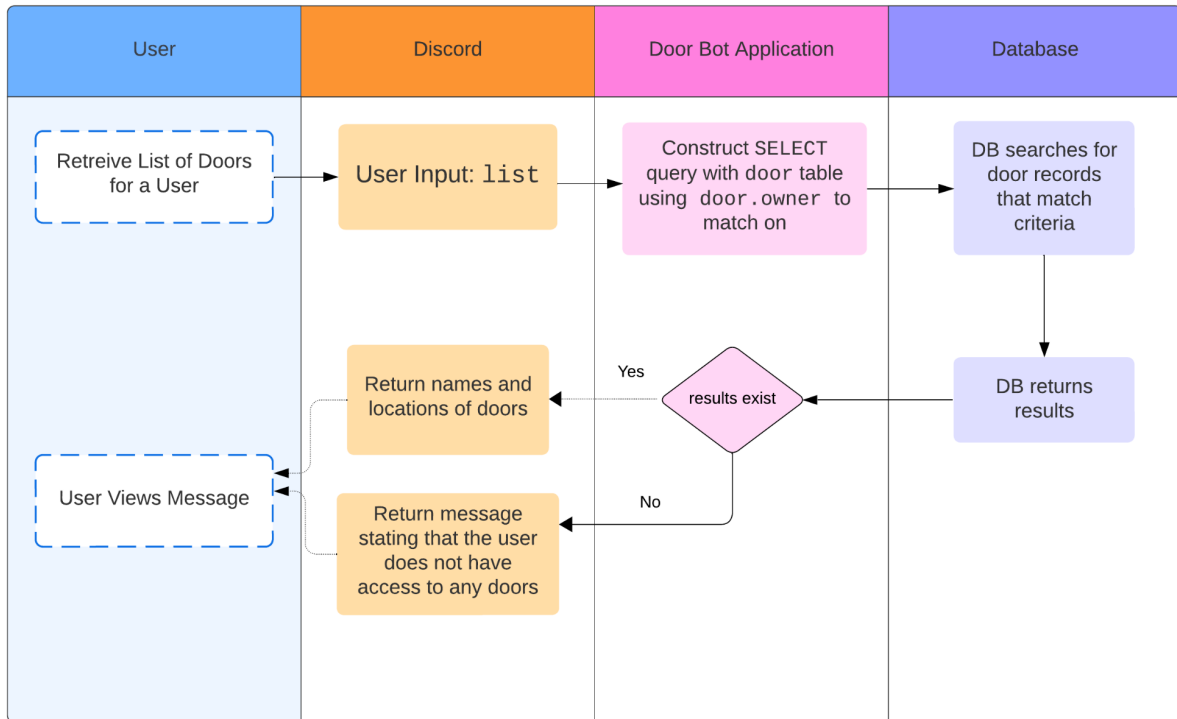| User	Discord	Door Bot Application	Database
Update Password	User Input: update password "new password"	Parse Update Password Command	
	"Invalid Password {Reason}"	Validate Password (Invalid Empty or Same as Old)	
		Valid Password Invalid Password	
	"Password Updated Succesfully"		DB Udpate On Password

Updates the user's associated password for their account and checks that the password is valid.

Schedule Event Swimlane

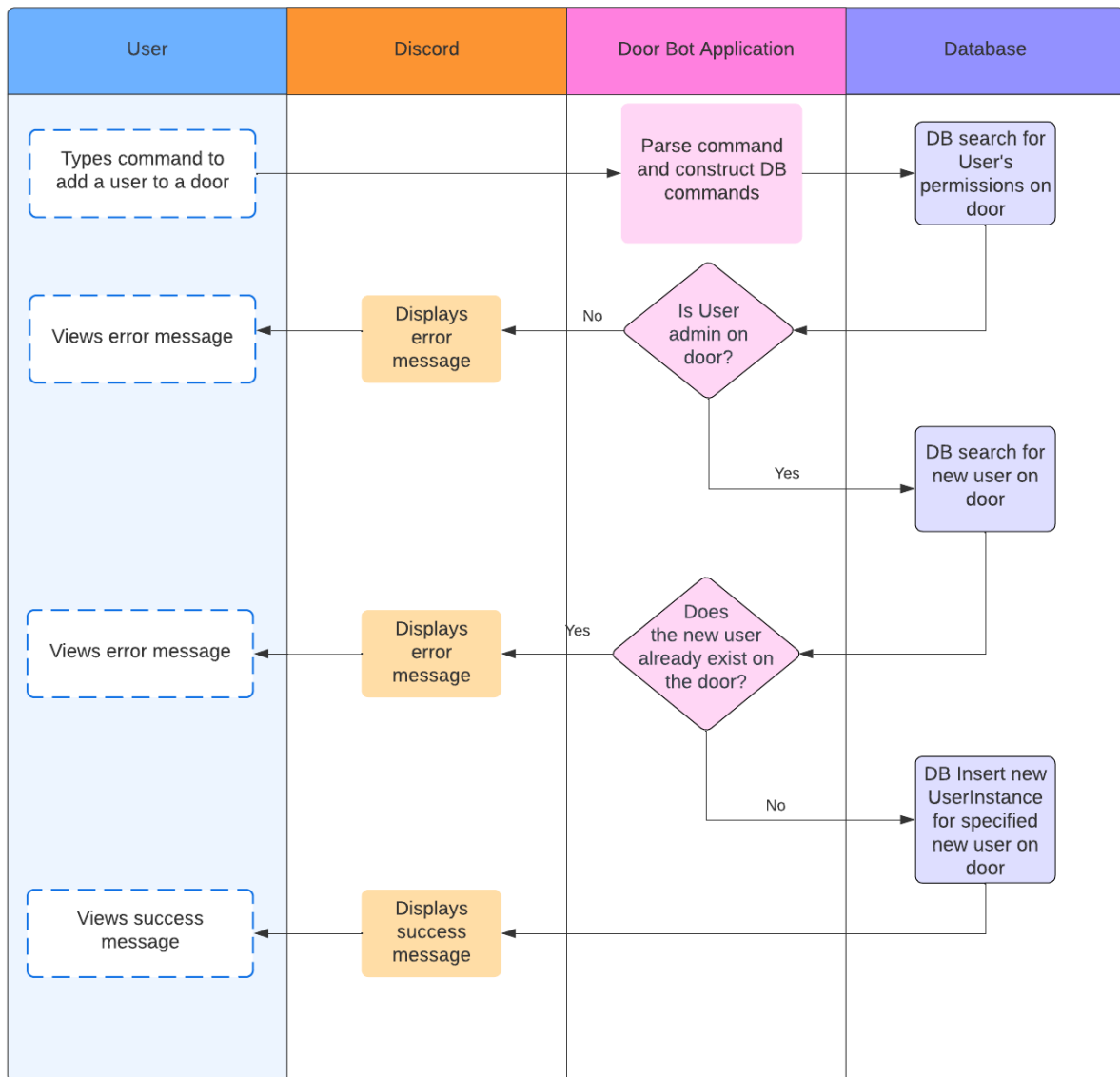| User	Discord	Door Bot Application	Database
User Schedules an Event	User Input: "schedule event <event name> <time interval> <door>"	Parse Command	
		Validate Command (no time interval overlaps, non empty event name etc,)	
	"Invalid Command {Reason}"	Valid Command Invalid Command	
			Database Insert On Event
	"Event Scheduled Successfully"		

Yes / **No**

Schedules a door event (when a door should be opened and why) for an associated user validating the event parameters before insertion. Only owners of the specified door can schedule an event.

List Doors that a User can access

User	Discord	Door Bot Application	Database

Retreive List of Doors for a User → User Input: `list` → Construct SELECT query with door table using `door.owner` to match on → DB searches for door records that match criteria

DB returns results → results exist

Yes → Return names and locations of doors

No → Return message stating that the user does not have access to any doors

User Views Message

Swimlane for when a user wants to see which doors they have access to open. For this operation, they will have to be authenticated and ideally given access by an administrator to at least one door. This operation searches for door records that have an owner equal to the user.

Add User To Door

User	Discord	Door Bot Application	Database

Types command to add a user to a door

Parse command and construct DB commands

DB search for User's permissions on door

Is User admin on door?

No → Displays error message → Views error message

Yes → DB search for new user on door

Does the new user already exist on the door?

Yes → Displays error message → Views error message

No → DB Insert new UserInstance for specified new user on door

Displays success message → Views success message

Swimlane for the process of adding a user to a door. The command requires admin privileges on the specified door and requires that the user to add does not already exist as a UserInstance tied to the specified door.