

## PM3 Deliverables from Team Seven

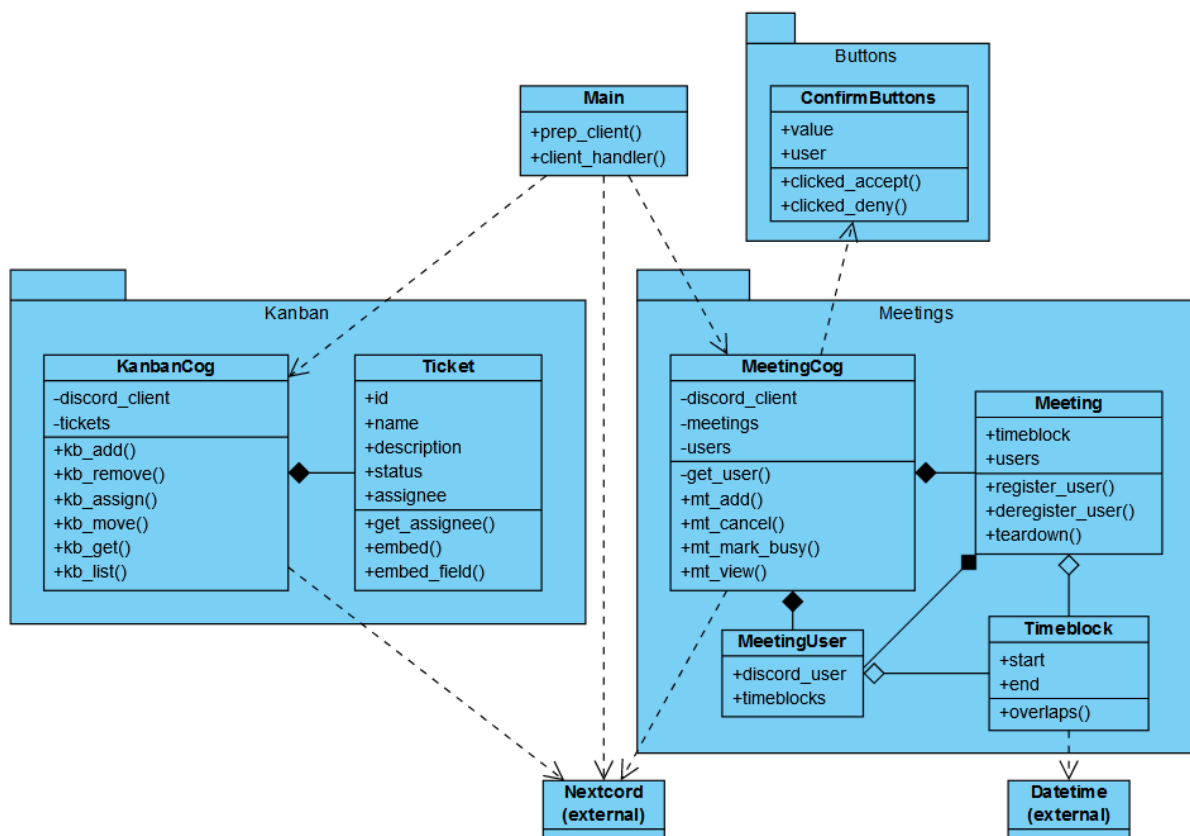
### High-Level Design:

From a higher level perspective, Discord DevOps will implement an event-based architecture. Most discord bots run on an event handling architecture, as they rely on asynchronously responding to an arbitrary number of users requesting tasks be performed at will. Therefore, it makes the most sense for us to choose an event-based architecture to best perform actions on demand.

### Low-Level Design:

From a lower level perspective, Discord DevOps will be working in the behavioral design pattern family, specifically using the "state" design pattern. Each portion of the bot contains stately objects that are modified during event handling, fitting the description of behavioral design. The kanban section tracks the tickets registered in the bot, and changes the state of those tickets in correspondence with user actions (such as reassigning or changing status). The meeting section tracks meetings registered in the bot, changing the state of the users and meetings in correspondence with user actions (such as making new meetings or RSVPing).

Simple UML diagram:



## Wireframe User Interface Mockup:

lorem ipsum dolor sit amet

consectetur

DcDevOps

Meeting 1: X:XXam-X:XXam

yes

no

/meeting busy

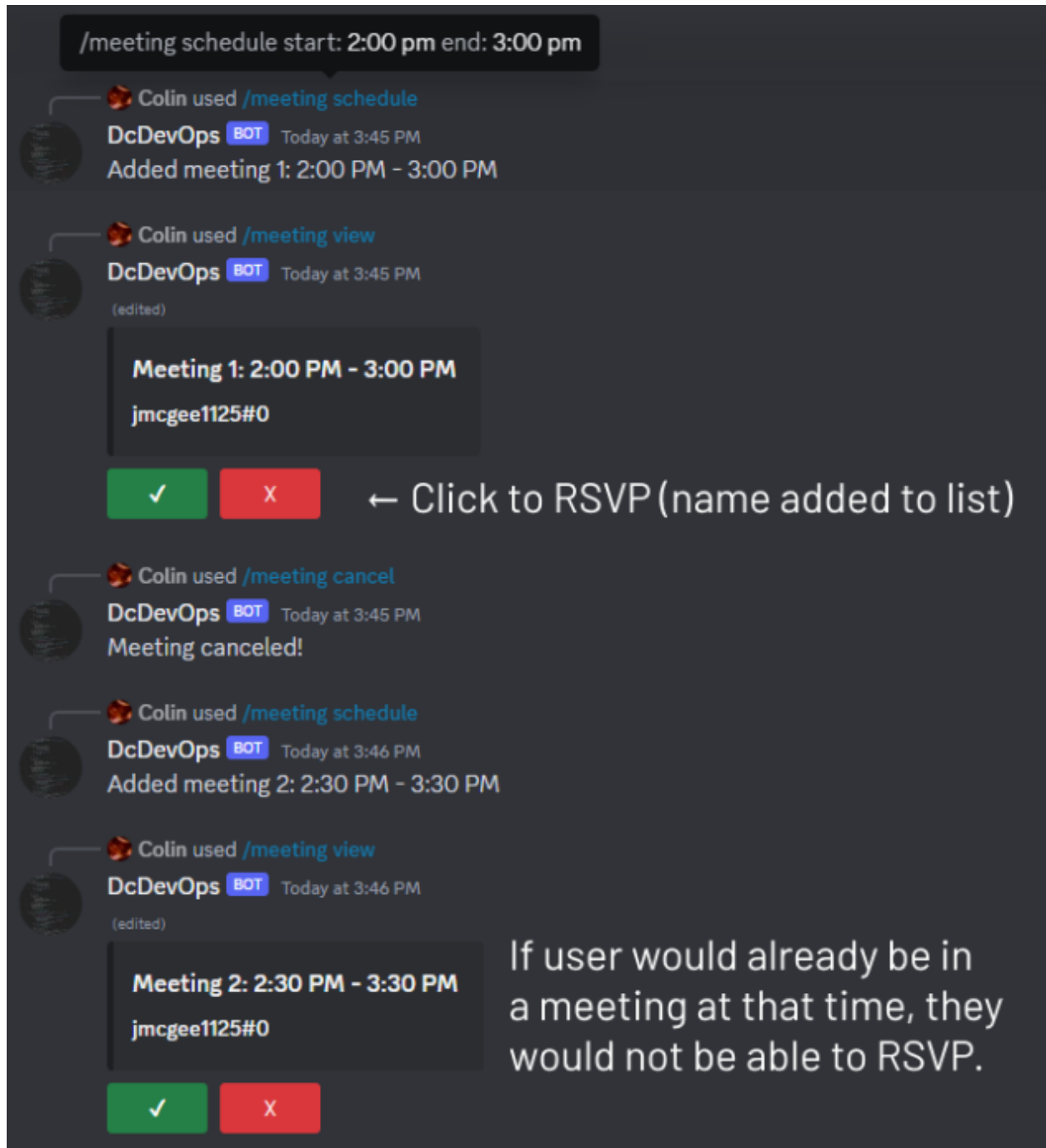
/meeting cancel

/meeting schedule

/meeting view

/meeting...

Process II Deliverable, Prototype:



Github repo:

<https://github.com/colinm25-vt/discord-devops>