Benjamin Dewey

The application is a virtual mailbox service network. Clients will be able to request their own personal mailbox from the service, search the list of existing mailboxes, and send and receive mail. The CRUD operations are as follows. Create: a client can ask the service to create a mailbox with a unique mailbox name supplied by the client, at which point they will receive their mailbox password; Read: a client can read the contents of their mailbox provided they have the mailbox name and password, and they can view/search the set of mailboxes that currently exist; Update: a client can send mail to another mailbox provided they have the recipient mailbox name and they have their own mailbox in which to stage the mail before delivery, and they can also empty their mailbox so it contains nothing; Delete: a client can destroy their mailbox so that it no longer exists.

When a client wishes to send mail, they must empty their mailbox of any contents and place the mail into their own mailbox. The service will periodically run a job that visits all mailboxes and transfers any outgoing mail into the correct recipient mailbox. If the mail cannot be delivered because the recipient mailbox does not exist, then the mail will be marked so that the next time the sender checks their mailbox they can see that their mail could not be delivered. A unit of mail consists of user and auto generated values. They are the date the mail was created, the sender’s mailbox name, the name of the mailbox to send the mail to, and a message written by the sender.

The client and service will communicate with each other via RPC, and the RPC system itself will be defined using the gRPC framework which will generate Python client and server stubs from service definitions defined by Protocol Buffers. Client interactions with the service will be handled by a command line utility.