

TeXtTop

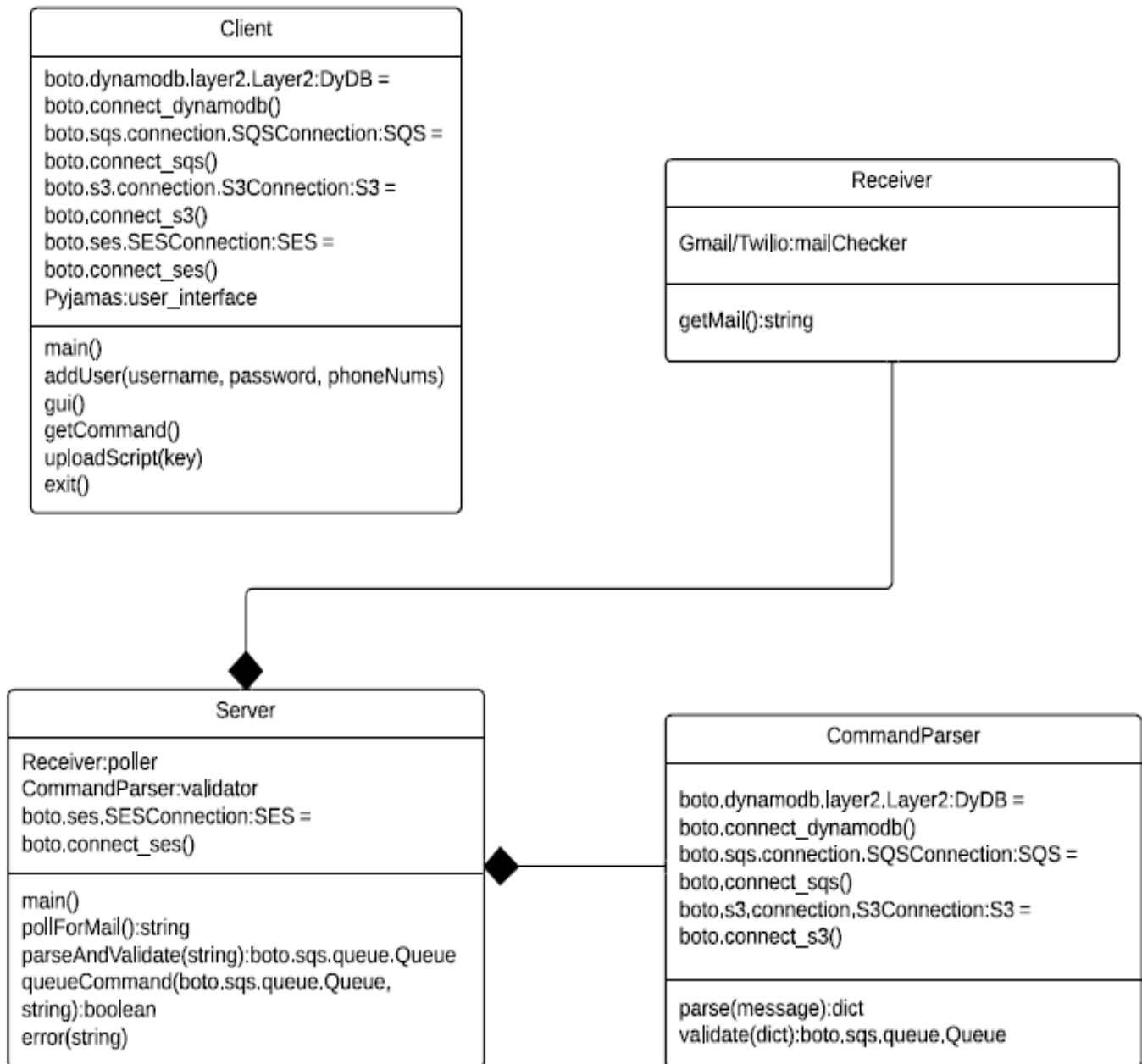
Stage III - Design

CSC 470
Dr. Depasquale
March 4 2013

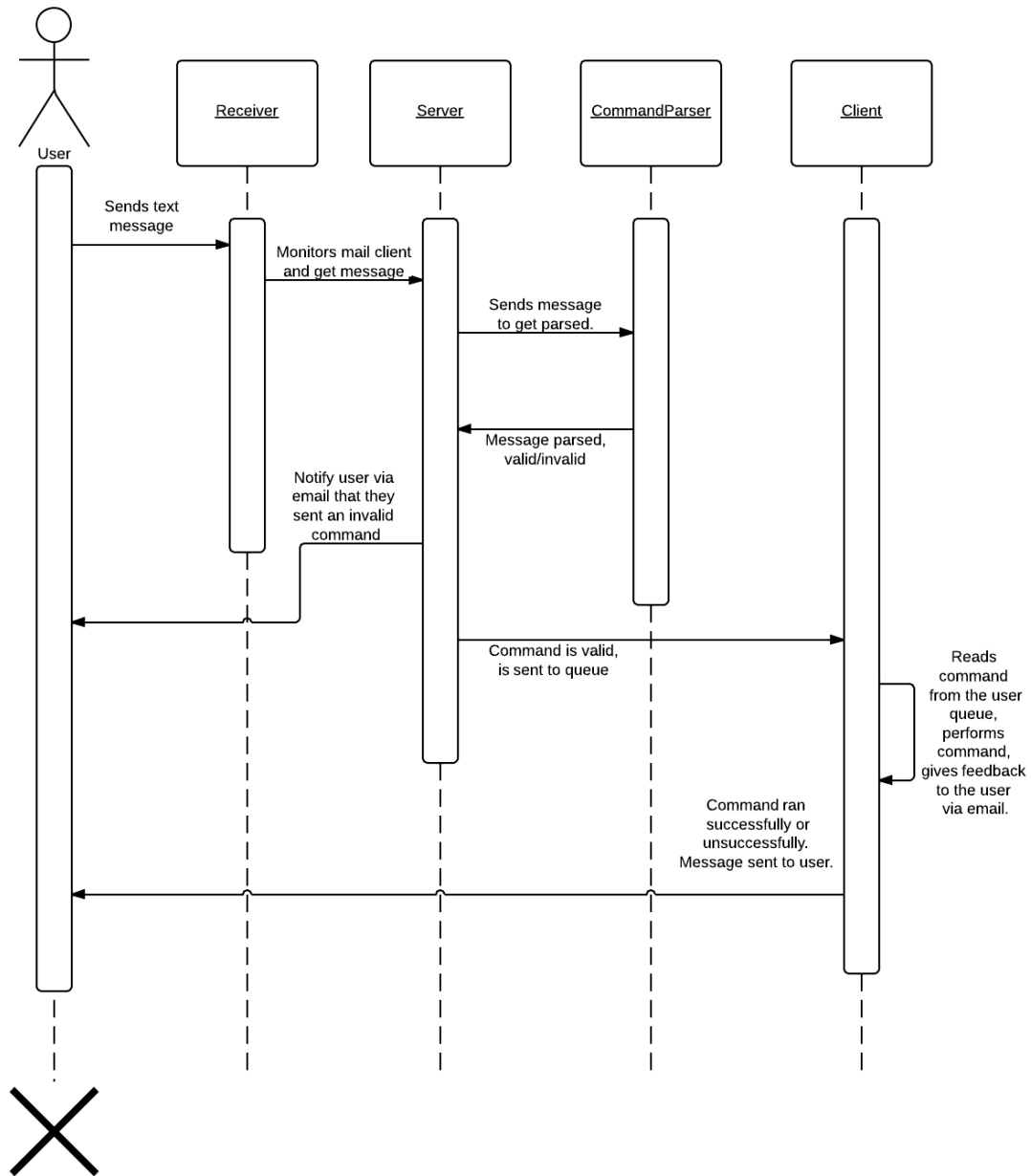
Dan Daly
Derek Duchesne
Steven Kolln
Dan Seminara

Github: <https://github.com/stee11/TeXt-Top>
WikiPage: <https://tcnj.instructure.com/groups/81511/wiki/front-page>

Design Class Diagram:

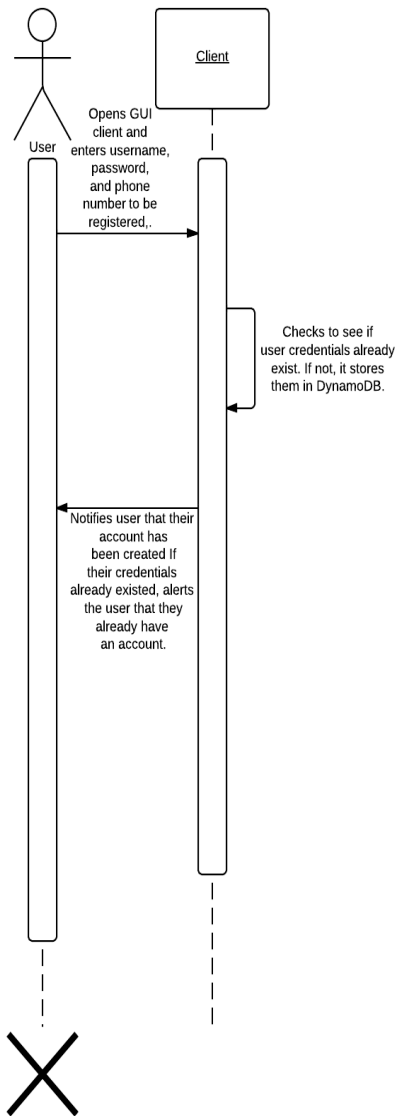


System Sequence Diagram:

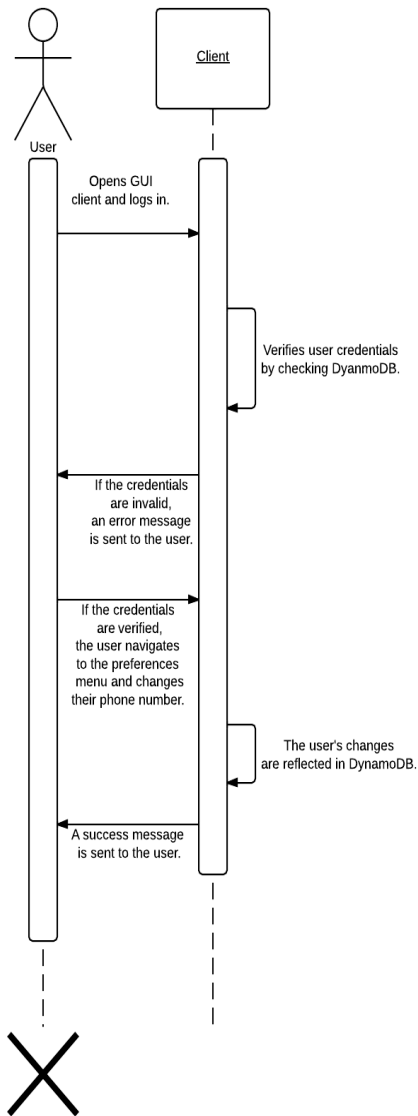


Client-side Sequence Diagrams

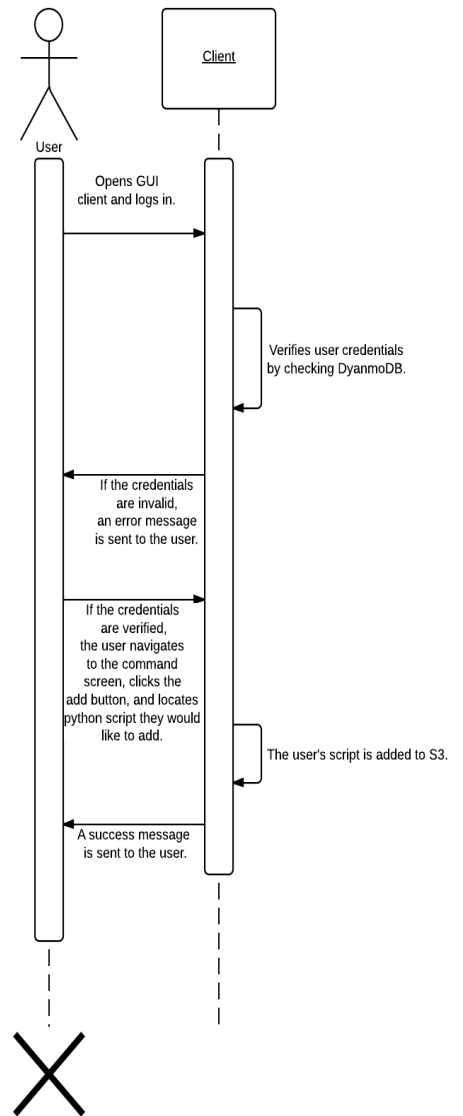
Account Creation



Changing Phone Number



Adding a New Command



UI 8 Golden Rules Descriptions:

1. Strive for consistency

The UI we designed is very simplistic and consistent throughout each of the menus. The UI allows for simple button clicking. It assists users in achieving actions easily, adding files and scripts to the commands page through finding them through their computer directories, and has simple to use text fields. The GUI does not overwhelm you with annoying buttons; each button and action exists for a reason.

2. Enable frequent users to use shortcuts

Within our UI, there is a preferences section which allows the program to be run on the startup of the computer. Furthermore, on the login screen there are options for users to minimize the window when the program starts and to allow automatic login.

3. Offer informative feedback

When users send our server a command, they are informed via a text message of whether or not the command they entered exists or not and whether or not they gave correct parameters to their command. The user will also be informed of whether or not the command they sent to the server was successfully ran or not on their client computer.

4. Design dialogs to yield closure

Each menu page within our UI has a specific function. There is a login page which specifically exists to handle user account creation and login authentication. If the login fails, the user will be notified via an error message. If not, the user is brought to the next screen. The main screen has four options: hide window, preferences, add commands, and exit. Upon clicking these buttons, the user will know the action is complete because the page of the UI changes to a new page. On the adding commands and preferences menu, a dialog box will tell the user of the result (success or fail) of adding a new command or editing their phone number or password.

5. Offer simple error handling

Users are given simple dialog boxes within our interface and through phone messages to be alerted of any errors that may occur. Errors include a command not running, a command not being able to be added or deleted from the list, and entering a phone number that already exists in the database. If a user enters a command that does not work, they can fix their command, delete their old one from the list, and add the new one. If a user wants to change their phone number it must be authorized through text message so that if the user enters the wrong phone number for their account, they can easily change it by simply entering a new phone number. Since their account has a password, other users cannot “hack” the user’s account even if they do manage to enter the correct phone number.

6. Permit easy reversal of actions

The actions completed by users will be very easy to undo. If the user adds the wrong script to their command list, they can easily add another while deleting the wrong one. Users can also add scripts back into the command library that they accidentally deleted by simply adding the script back. If the user at one point decides they like the idea of auto login, minimizing the window at startup, or any of the other checkmark-able features, they can simply be turned on or off with another click of the button.

7. Support internal locus of control

We give the users the power to change how the program loads such as having it be minimized when running and to load on computer startup. They also have the ability to add their own scripts and delete existing ones, and can easily change their phone number and password with the minimum number of steps required.

8. Reduce short-term memory load

One way our memory load is reduced is through having only the menus and functions needed for our project. For example, the only functions truly needed should be those for account creation, account editing, basic preferences, and script editing, all of which we support. The application is very light weight. There are only four menu screens with each only having a couple basic functions. These include logging in, changing preferences such as username and password, and adding and removing scripts.