Abdurrahman Mohammad

Ying Chang Cui

Eduardo Fragoso

Emmanuel Cestoni

**Assignment 3**

Summary:

This document contains both the agreed upon analysis phase and the design phase as well as the associated diagrams. The analysis phase is mentioned first, followed, by the design phase, followed by the diagrams. This document is also separated as well. Pictures are also in the zip.

**Analysis Phase**

Voicemail System

**---------- Functional Specification ----------**

Summary:

This is a voicemail system that does:

* Lets a person dial an extension number. An input line consisting of a single character 1 . . . 9 or # denotes a pressed button on the telephone touchpad
* Connects the call to the extension
* Provided the other party does not pick up the telephone, the caller leaves a message
* The other party can later retrieve the messages
* The other party can later keep the messages
* The other party can later delete the messages

Point of view of the caller:

The caller has the options to:

* When the caller calls the main number, he will be asked to dial an extension or hang-up
* The caller dials an extension and is connected to that extensions
* A greeting is played
* If the owner of the extension does not pickup, the caller will be directed to voicemail
* The caller can then hang-up or leave a 30 second message. The caller can leave a shorter message by hanging up after leaving the message.

Point of view of the owner:

* The owner calls the main number and dials his extension
* He will be played a greeting
* He will be asked to enter a password or wait to leave a message
* He will enter the correct password. If it is incorrect, he repeats the above steps
* The owner now can view unread messages or read messages
* Then the messages in the queue of the unread or read messages are played. He will have the option to delete the message or keep them immediately after each one is played
* After reading an unread message, it will be stored in the read messages queue
* Then the owner may hang-up after any message is played by pressing ‘H’ or he may listen to all the messages in the selected queue before the system hangs up.

**---------- User Manual ----------**

Dialing the main number of the system on the telephone (main) will initialize the voice mail system. You first will hear a welcome and some options. Then you can enter the extension you wish to reach or hang up (enter 3 numbers from 0 to 9 and end with a #). If you dial a valid extension, you will be connected to that extension if it is enabled. If the owner of the extension is available, he will pick up and you can converse with him. If he is not available, you will be led to voicemail. At voicemail, you will hear some options and can leave a message after the “beep.” The message can be at max 30 seconds long or you may shorten it by hanging-up. If you enter a password after the voicemail options message (enter 3 numbers from 0 to 9 and end with a #), you will have access to your voicemail mailbox and will be given the options to delete a message by pressing 0, hang-up by pressing H, or listen to the messages in the queue. You may hang-up after a message is played or delete it by pressing 0. The system will automatically hang-up after all the messages in the mailbox are played. If you are the admin, enter the extension ‘000’ and the password ‘987. You will be given the option to enable or disable mailboxes and change the password of mailboxes. You first enter the mailbox number and then its new password.

**---------- Use Cases ----------**

The caller calls the number of the system from the “main” worldwide telephone system (method). The system would start playing the caller’s options and wait for an input.

**Use Case: Invalid input**

If the input is invalid (letters, symbols, length larger than 3), the system would ask for another input or ask to enter H to hang-up. If the input is a symbol that is not a number, ‘H’, or ‘#’, you will be asked to reenter.

**Use Case: Valid input**

After a valid input (3 numbers followed by ‘#), the system would then search and connect the caller to the correct extension and call it. The user may pick up the phone and the system hangs-up after the call. You can press ‘H’ to hangup.

**Use Case: Admin Menu**

If the user dials ‘000’ and enters the correct password ‘987’, he will have the option to enable or disable mailboxes and change their passwords.

**Use Case: The extension owner is not available to pickup**

If the extension owner is not available to pick up the call, the caller would be led to voicemail. A greeting will be played recorded by the user. Then the user will hear more options, such as to enter a password or wait for the “bleep” to leave a message. These two cases will be discussed in the following paragraphs.

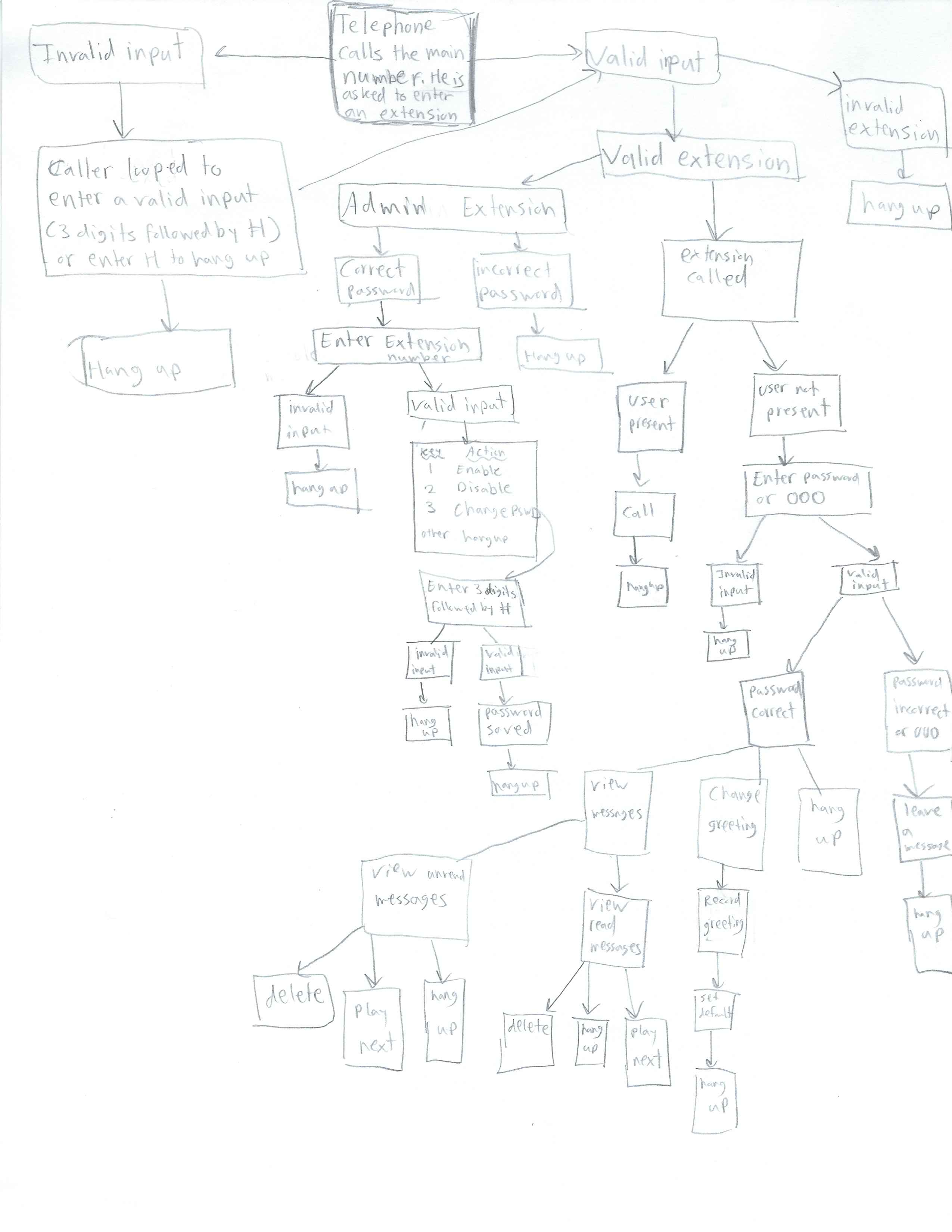
**Use Case: The user enters the correct password**

If the user enters a password, he will be played a message stating his options: press 0 to listen to unread messages and press 1 to listen to saved/read messages. After that, you will hear: press 0 to delete a message or do nothing to keep it, press H to hang-up, press 9 to record a greeting. After this intro message is played, the user can enter 0, H, or nothing after each message in the mailbox is played to perform an action. A message will be played, and the user will press 0 to delete the message or wait or press a random key for the next message. An unread message heard will be placed in the read messages queue. The system automatically hangs-up after playing all the messages in the mailbox.

**Use Case: Greeting management**

If the user presses 9, he will be asked which greeting he wants to change. He can press 1, 2, or 3 and then he can replace that stored greeting by speaking into the phone. After that, the user will be asked which greeting to set as default to which the user enters 1, 2, or 3. Pressing any other keys during this phase will hang up the phone.

**Use Case: The user enters a wrong password or waits**

If the user enters a wrong password or does not enter anything, he will be allowed to leave a message in the mailbox. A “beep” will be played. The caller has 30 seconds to leave a message. The caller can press H during his message to end his message if he wishes not to speak for 30 seconds. If he does so, the message is ended and saved before hanging up. After leaving a message, the call is hung-up, and the system is exited.

**Design Phase**

Message

Responsibilities:

* Stores a message
* Returns the message

Fields:

* Message: A String variable that stores a message

Methods:

* Default Constructor: The default constructor constructs a Message object with no actual message
* Overloaded Constructor: The overloaded constructor of the Message class creates a new Message object and stores a message passed in
* Get Method: returns the saved message as a String
* Set Method: Modifies/Replaces the saved message

Greeting

Responsibilities:

* Stores 3 greetings
* Sets a default greeting (one of the greetings in the array)
* Returns the default greeting

Fields:

* An array of Strings to store 3 greetings
* The index of the default greeting as an integer

Methods:

* Constructor
* Add greeting: adds a greeting to the array at index i
* Remove Greeting: Removes the recorded greeting at index i
* Set Default Greeting: Sets the index of the default greeting
* Get Greeting: Returns the default greeting as a String

MessageQueue

Responsibilities:

* Stores messages in a queue

Fields:

* An array of messages (max 10)
* A variable to record size

Methods:

* Play Message: Plays the message at index i
* Enqueue: Adds a message to queue
* Dequeue: Removes a message and fixes up queue to fill in any null spaces
* Get Size: Returns the size of queue
* Is Full: Returns true if the queue is full

Mailbox

Responsibilities:

* Plays a default greeting for the caller
* Checks the password with an inputted password
* Connects the caller to the owner if the owner is present
* If the owner is not present, the caller can add a message to the unread message queue
* The owner can record a greeting and replace a greeting stored in the greeting object at index i
* The owner can set the default greeting
* Adds messages to the queues saved messages and unread messages
* Removes messages from the queues saved messages and unread messages
* Lets the user listen to the message queues
* Lets the Admin change the password

Fields:

* A queue to store unread messages
* A queue to store read messages
* A Boolean variable to show the status of the mailbox’s activation
* A Boolean variable to show the user is present
* A Boolean variable showing the status of the mailbox being full (optional)

Methods:

* Constructor
* Play the default greeting: Lets the caller hear the default greeting
* Record a greeting: Adds a greeting to the list of 3 greetings
* Remove a greeting: Removes a greeting
* Set a default greeting: Sets a default greeting among the 3
* Leave a Message: Lets the caller leave a message and adds it to the unread messages queue
* Play Unread Messages: Lets the user listen to the unread messages
* Play Read Messages: Lets the owner listen to the queue of read messages
* Delete Selected Unread: Deletes the selected unread message
* Delete Selected Read: Deletes the selected read message
* ~~Delete All Unread: Deletes all the messages in the unread messages queue~~
* ~~Delete All Read: Deletes all the messages in the unread messages queue~~
* Password Check: Checks the password inputted with the password of the mailbox
* Set Password: Lets the admin the password
* Is Activated: Lets the admin activate or deactivate the mailbox
* Set Activated: Sets the activation of the mailbox
* Mailbox is full: Returns true if the mailbox is full (optional)

IO

Responsibilities:

* Prints out all the Strings passed in by VoiceMailSystem
* Inputs all the Strings requested by VoiceMailSystem
* The voice mail system uses this class for input and output

Fields:

* Scanner

Methods:

* Get Integer Input: Gets input from the caller and passes it to the voicemail system
* Get String Input: Gets input message of the user to simulate speaking. The information is sent back to the voicemail system.
* Print Output: Prints information sent by the voicemail system

VoiceMailSystem

Responsibilities:

* Simulates pushing a button
* Calls an extension
* If the user does not pick up, the caller goes to voicemail
* The caller can leave a message
* The owner can enter the voicemail password and retrieve the messages
* The owner can delete messages
* The admin can activate an extension and set its password

Fiels:

* An array of mailboxes which serve as the extensions

Methods:

* Constructor
* Initiate: Initiates the protocol of connecting the caller by combining the other methods
* Play Greeting: Welcomes the caller and asks for an extension
* Push Button: Simulates pushing a button
* Search Extension: Verifies that the extension is activated and is in range
* Invalid Extension: Plays an error message
* Call: Calls the extension and if the user is present, they talk and hang up
* Speak: Simulates talking. Used when the user picks up or leaves a message.
* Hang up: this method will exit the system
* Voicemail: If the owner is not available (after call), the caller can leave a message or check voicemail. The caller is asked to either enter a password or press ‘#’ to leave a message. Leaving a message utilizes Mailbox methods.
* Voicemail Management: If the password is correct, the caller can listen to messages, delete messages, and manage the greetings. This utilizes Mailbox methods.
* Admin Menu: If the caller dials ‘000’ and enters the password in the package which is ‘987, then he can activate or deactivate mailboxes and set the passwords. This utilizes mailbox methods.

Telephone (Main)

Responsibilities:

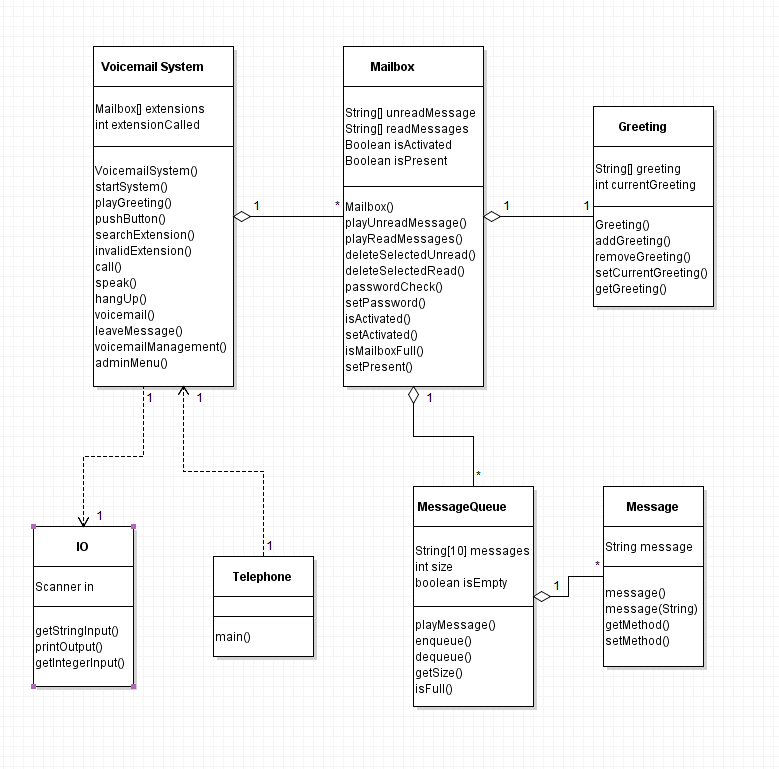
* Creates a VoiceMailSystem and tests the program

Methods:

* Main: This method would test the voice mail system. It would call a number and if the person does not pick up, you can leave a message. You can also access the messages and set up mailboxes. In this method, you simulate the caller, owner, or the admin calling and using the voice mail system.

**Relationships:**

* Telephone **uses** VoicemailSystem
* VoicemailSystem contains 0 to n (**aggregation**) mailboxes Mailbox and **uses** IO
* IO **uses** Scanner and **uses** System.out objects
* Mailbox contains 1 (**aggregation**) Greeting and contains 0 to n(**aggregation**) MessageQueue
* MessageQueue contains 0 to n (**aggregation**) Message type messages

**Class Diagram**

**Sequence Diagrams**

**Setup Greeting**

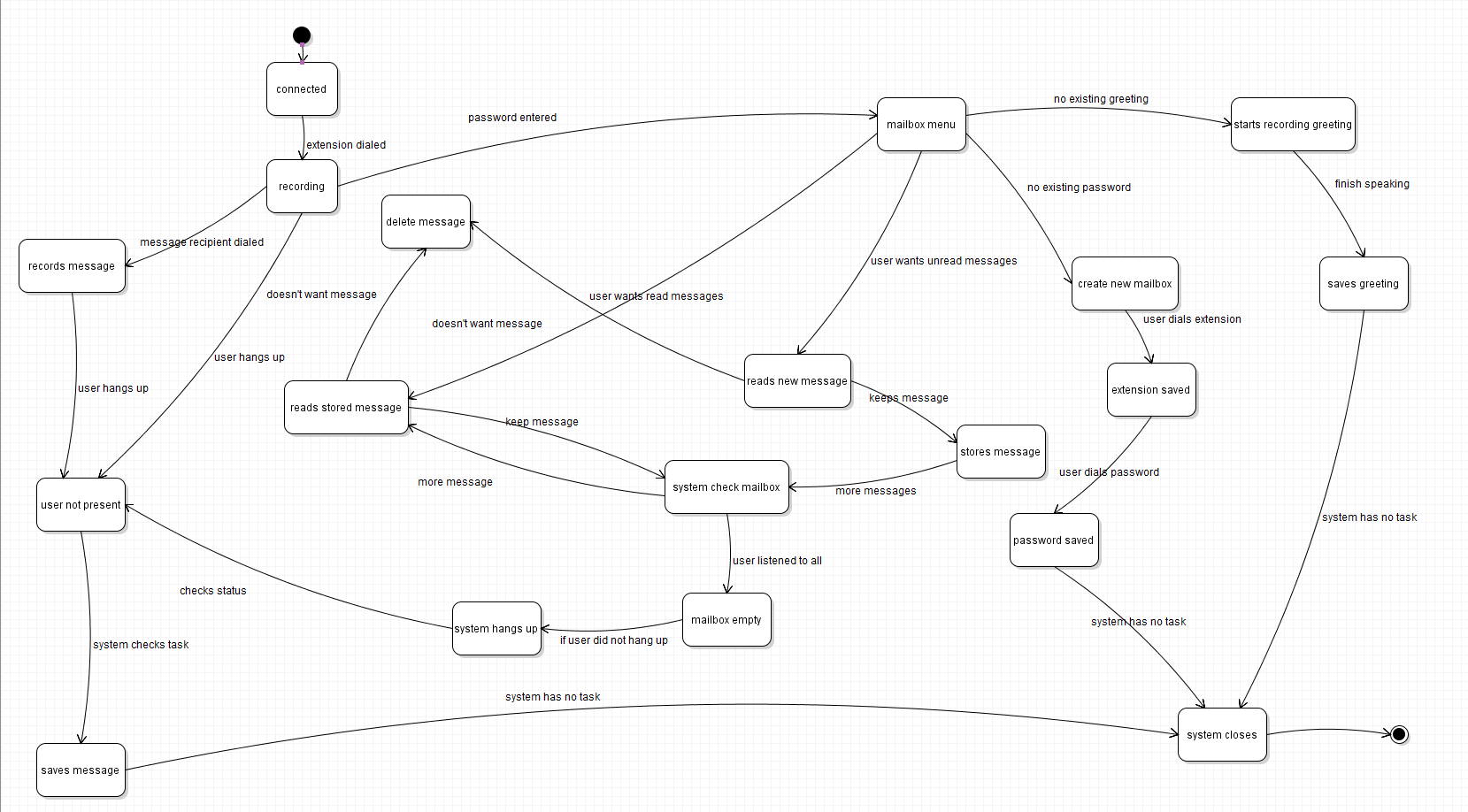
**Playing a Message**

**Playing a Message**

**Leaving a Message**

**Deleting a Message**

**State Diagram**

****