```
// Import statements
Import org.springframework.boot.SpringApplication;
Import org.springframework.boot.autoconfigure.SpringBootApplication;
Import java.util.set;
Import javax.ws.rs.core.Application;

// Deploys/executes the application
Class Application {
	// Main method for execution, utilizes SpringApplication
	public static void main(String[] args) {
		SpringApplication.run(Application.class, AccountService.class, args)
	}
}
```

```
// Inbox class
class Inbox {
	# private fields
	private String accountID;
	private List<Message> messages;
	private List<Notifications> notifications;

	// constructor
	Inbox() {
		super();
	}
	// getter methods
	public List<Message> getMessages() {
		return messages;
	}
	public List<Notifications> getNotifications() {
		return notification;
	}

	// Allow the user to modify the inbox, i.e. archive and delete messages
	public void editInbox(String condition) {

		if (condition.equals(“Archive”) {
			// have the user choose a message to save
			Message m = mouseClicked()
```

```
                messages.add(m) // append the message to archive
            }
            Else if (condition.equals(“Delete”) {
                String deleted= mouseSelected ()
                // have the user select the one to delete
                supplies.remove(newSudeletedpply)
                // remove the existing message from the list
            }

        }


}
```

```
# Message class
class Message {
        # private fields
        private String accountID;
        private Account sender;
        private List<String> recipients;
        private List<String> attachments;

        // constructor
        Message() {
                super();
}
        // two getters
        public List<Message> getMessage() {
                return message;
        }

        public List<Notifications> getAttachments() {
                return attachments;
        }

        // method to send a message.
        // will use java api POST calls
        public void sendMessage() {

                // encode the message,
```

```
                URLencoder.encode(String, UTF-8)
                Open.connection();              // open connection to recievers
                connect();
                setRequestMethod(POST)    // api POST call
                responseCode = getResponseCode()

                if responseCode == HTTPURLConnection { // if connection successful, read message
                        // read line from keyboard
                        While (readline()) {
                                Message.append(character)
                        }
                }
                close();                // close connection
                }
        }

}
```

```
class House {
        // Private fields
        private String Financier;
        private String ContractingCompany;
        private String Supplier;
        private List<String> supplies

        // constructor
        House() {
              super()
        }
        // getters
        public String getFinancier() {
              Return financier
        }
        public String getContractor() {
              Return ContractingCompany
        }
        public String getSupplier() {
              Return financier
        }
```

```
        // Ability to edit existing supply list of a house
        public void editSupplies(String condition) {

                If (condition.equals("Edit") {
                        String supply = mouseSelected // allow user to select the supply to modify
                        If (supplies.contains(supply)) {
                                supplies.edit(realLine())     // edit existing element
                        }
                }
                Else if (condition.equals("Add") {
                        String newSupply = readLine() // have the user enter in a new supply
                        supplies.add(newSupply) // append the new supply to the back of the list
                }
                Else if (condition.equals("Delete") {
                        String newSupply = mouseSelected ()
                        // have the user select the one to delete
                        supplies.remove(newSupply)
                        // remove the existing supply to the back of the list
                }



        }


}
```

```
// import statements
Import javax.model.Person;
Import javax.model.Response;
Import javax.ws.rs.GET;

Class Account {

        // private fields
        private String accountID;
        private String password;
        private List<House> houses;
        private Inbox inbox;

        public void createAccount(String ID, String pass) {
```

```
        Response r = new Response();
        Person p = new Person(ID, pass)
        // if
        If (p.getID() != null || invalid & p.getPassword() != invalid) {
                add(Person)    // add new account to database
        }
}

// Will return account information for a user, uses GET api call
public String  getAccountInformation() {

        HTTPURLConnection conn = openConnection(URL);
        // api GET call
        conn.setRequestMethod("GET")
         // have user enter in proper identification
        conn.setRequestProperty(readline(), readline())
        Int response = conn.getResponseCode()

        If  (response == HTTP-OK) {
                StringBuilder message;

                // get inputstream from the response of the connection
                InputStream get = new InputStream(conn.getInputResonse())
                BufferedReader input  = new BufferedReader(get)

                // read and append the message
                While (input.readline != null) {
                        // get the returning message from the input stream
                        message.append(readLine)
                }

                // if message properly received by client, return the message
                If (builder != null) {
                        Return builder
                }
                Else {
                    print(error.trace())
                }

        }

}
```

```
}
```