**Register**

1. **Description:**User will go to register section  
   Input user name + password  
   submit input to server  
   Wait for response from users  
   If successful then go to main program  
   Else then go back to previous stage
2. **Tasks:**
   1. **Client:  
      Input:** username + password.  
      **Output:** MessageClient -> JSON
      1. Create forms for user to register:  
         Form must have username and password
      2. Save input data as structure  
         { method = 2;  
         user = “username”  
         password = “password”}  
         Note: *key not in the field above is filled by default values*
      3. Wrap to JSON
      4. Send to server
      5. Receive Message from server
      6. Extract and read message
   2. **Server:  
      Input:** JSON String as message from client  
      **Output:** MessageServer -> JSON
      1. Receive the message
      2. Check for the type of method
      3. Examine data
         1. Fail  
            - Add error = 1 if user exists in database  
            - Add error = 2 if user is too long (> 15)  
            - Add error = 3 if pass is too short (< 5)  
            - Wrap MessageSever with *error* above and code = 7  
            {method = 1  
            code = 7  
            error = [errors]}
         2. Success:  
            Wrap MessageServer with code = 3  
            {method = 0  
            code = 1  
            error = [errors]}
      4. Wrap Message to JSON
      5. Send response to Client
3. **Tests:**
   1. **Available user name and password:**{user = “abcde”  
      password = “123456”}  
        
      -> Message Response: { error = [], code = 3}
   2. **Too long user name:**{user = “abcdefgh…xyz”  
      password = “123456”}  
        
      -> Message Response: { error = [2], code = 7}
   3. **Too short password:**  
      {user = “abcdefg”  
      password = “123”}  
        
      -> Message response { error = [1], code = 7}
   4. **Password too short and existed user name:**  
      {user = “abcde”  
      password = “123”}  
        
      -> Message response {error = [1, 3], code = 7}