

- **Commented Source code (Update Member Details):**

<https://github.com/edprince/ski-project/tree/996d30fa7a49fea3ab0eab3b87b2244d1db54a19/public>

JavaScript

```
function get(id) {
    return document.getElementById(id).value;
}

//Class for database connection
//Initialize Firebase
class DatabaseConnection {
    constructor() {
        var config = {
            apiKey: "AIzaSyDl__SyPP9HQ4-ViwJxH6s8EACpCY9-k34",
            authDomain: "skiproject-8236e.firebaseio.com",
            databaseURL: "https://skiproject-8236e.firebaseio.com",
            projectId: "skiproject-8236e",
            storageBucket: "",
            messagingSenderId: "633876681860"
        }
        firebase.initializeApp(config);
        var database = firebase.database();
    }
}

//Class Member
class Member {
    constructor(firstname, surname, email, address, dob, password, membership){
        this.firstname = firstname;
        this.surname = surname;
        this.email = email;
        this.address = address;
        this.dob = dob;
        this.membership = membership;
    }
}

//Class SlopeUI which controls class Controller
//based on GRASP use case control pattern
class SlopeUI {
    constructor() {
        var searchButton = document.getElementById("SearchButton");
        var deleteMember = document.getElementById("Delete");
        var updateMember = document.getElementById("Update");
        var clearButton = document.getElementById("Clear");
        //Event Listener which Searches for Member when button Search is pressed
        searchButton.addEventListener('click', function() {
            let controller = new Controller() // reference to GoF Singleton
            controller.SearchMember(Member); // call SearchMember function
        });
        //Event Listener which updates Member details when button Update is pressed
        updateMember.addEventListener('click', function() {
            var newUpdate = new Member( //object which gets all values from the HTML
            page where the SlopeUp has entered the Updated details
                get('firstname_update'),
                get('surname_update'),
                get('email_update'),
                get('membership_update'),
                get('dob_update'),
                get('address_update')
            );
            //After the button Update is pressed - Validate class is being
            triggered
        });
    }
}
```

```

        //and if all details are correct - triggers Update Member function
        //otherwise re-displays details again
        //and sends an alert to the screen - Invalid Input
        var validate = new Validate();
        if (validate.name(newUpdate.firstname) && validate.name(newUpdate.surname)
        && validate.email(newUpdate.email)) {
        //(validate.addressvalid(newUpdate.address)){
            let controller = new Controller(); //reference to GoF Singleton
instance (from Controller Class)
            controller.UpdateMember(Member);
        } else {
            console.log("Invalid input.");
            let controller = new Controller(); //reference to GoF Singleton
instance (from Controller Class)
            controller.SearchMember(Member);
            alert('Invalid input. Please re-enter details.');
```

```

        details.dob =
document.getElementById("dob_update");
        details.firstname.value =
snapshot.val().firstname;
        details.surname.value = snapshot.val().surname;
        details.email.value = snapshot.val().email;
        details.membership.value =
snapshot.val().membership_tier;
        details.dob.value = snapshot.val().dob;
        details.address.value = snapshot.val().address;
    });
}
UpdateMember(member) {
    //Update function which sends the updated details to the database
    //and displays them in a table on the page
    firebase.database().ref('user/'+userReference).set({
        'firstname': details.firstname.value,
        'surname': details.surname.value,
        'email': details.email.value,
        'address': details.address.value,
        'membership_tier': details.membership.value,
        'dob': details.dob.value
    });
    $("#table_body").append("<tr><td>" +
details.firstname.value + "</td><td>" + details.surname.value + "</td><td>" +
details.address.value + "</td><td>" + details.email.value + "</td><td>" +
details.membership.value + "</td><td>" + details.dob.value + "</tr></td>");
    $("#table_body").append("<tr>");
}
DeleteMember(Member) {
    //Delete function which deletes a member after the
    //Delete button is pressed
    firebase.database().ref('user/'+ userReference).remove();
    window.alert("Member Deleted!");
    window.location.reload();
};
}

//Validation class
class Validate {
    constructor() {};
    //Validation Functions
    email(email) {
        //Function to validate email
        var re =
/^((([^<>()\\[\]\\\\.,;:\s@"])+(\.([^<>()\\[\]\\\\.,;:\s@"])+)*)|(".+"))@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;
        return re.test(email);
    }
    name(name) {
        //Function to validate names (first name & surname)
        var re = /^[a-zA-Z]+$//;
        return re.test(name);
    }
    addressvalid(address) {
        return ",#- / !@%$^*( ) { } | [ ] \\\\".indexOf(address) >= 0;
    }
}

```

HTML code:

```
<!DOCTYPE html>
<html>
  <head>
    <script src="https://www.gstatic.com/firebasejs/3.7.0/firebase.js"></script>
    <link
href="https://fonts.googleapis.com/css?family=ABeeZee|Alegreya|Alegreya+Sans|Ques
trial" rel="stylesheet">
    <link
href="https://cdnjs.cloudflare.com/ajax/libs/foundation/6.3.0/css/foundation.css"
type='text/css' rel='stylesheet'> <link href='css/main.css' type='text/css'
rel='stylesheet'>
    <meta name='viewport' content='device-width, initial-scale=1'>
    <meta charset="utf-8">
    <title>Sphere Ski Centre</title>
    <style media="screen">
      body {
        background: #ECEFF1;
        color: rgba(0,0,0,0.87);
      }
    </style>
  </head>
  <body>
    <div class="form">
      <h1>Sphere - Update Details</h1>
      <input type = "text" placeholder='Search by First Name' id = "Search">
<button id = "SearchButton">Search</button>
      <br>
      <br>
      <table id='update_table'>
        <tbody id='table_body'></tbody>
        <thead>
          <tr>
            <td>First Name</td>
            <td>Surname</td>
            <td>Address</td>
            <td>Email</td>
            <td>Membership Status</td>
            <td>Date of birth</td>
            <td>Make Changes</td>
          </tr><tr>
            <td><input type='text' id='firstname_update'></td>
            <td><input type='text' id='surname_update'></td>
            <td><input type='text' id='address_update'></td>
            <td><input type='text' id='email_update'></td>
            <td><input type='text' id='membership_update'
readOnly></td>
            <td><input type='date' id='dob_update'></td>
            <td> <button id ="Update">Update</button>
            <button id = "Delete">Delete</button> </td>
          </tr>
        </thead>
      </table>
      <button id ="Clear">Clear</button>
    </div>
    <script>
      function date() {
        var dateVal = document.getElementById("date").value;
      }
    </script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
    <script src="https://www.gstatic.com/firebasejs/3.6.8/firebase.js"></script>
```

```
<script src='js/update.js'></script>
</body>
</html>
```

CSS:

```
@media (max-width: 800px){
  button {
    width: 100%;
  }
}

* {
  font-family: 'Questrial', sans-serif;
}

button {
  background-color: #039be5;
  color: white;
  text-transform: uppercase;
  padding: 1.2em;
  margin: 0 auto;
  margin-top: 1em;
  display: block;
  border-radius: 5px;
}

.form input {
  background-color: #e7e7e7 !important;
  padding: 1em;
  border: none;
  border-radius: 5px;
  margin-bottom: 1em;
}

.form {
  width: 90%;
  margin: 0 auto;
  display: block;
}

h1 {
  color: #039be5;
  margin: 1em;
  font-family: 'Questrial', sans-serif !important;
}

.left-align {
  text-align: left;
}

.center {
  text-align: center;
  display: block;
  margin: 0 auto;
}
```

Screen shots of my two prototypes (integrated with Ed's functionality – register user)

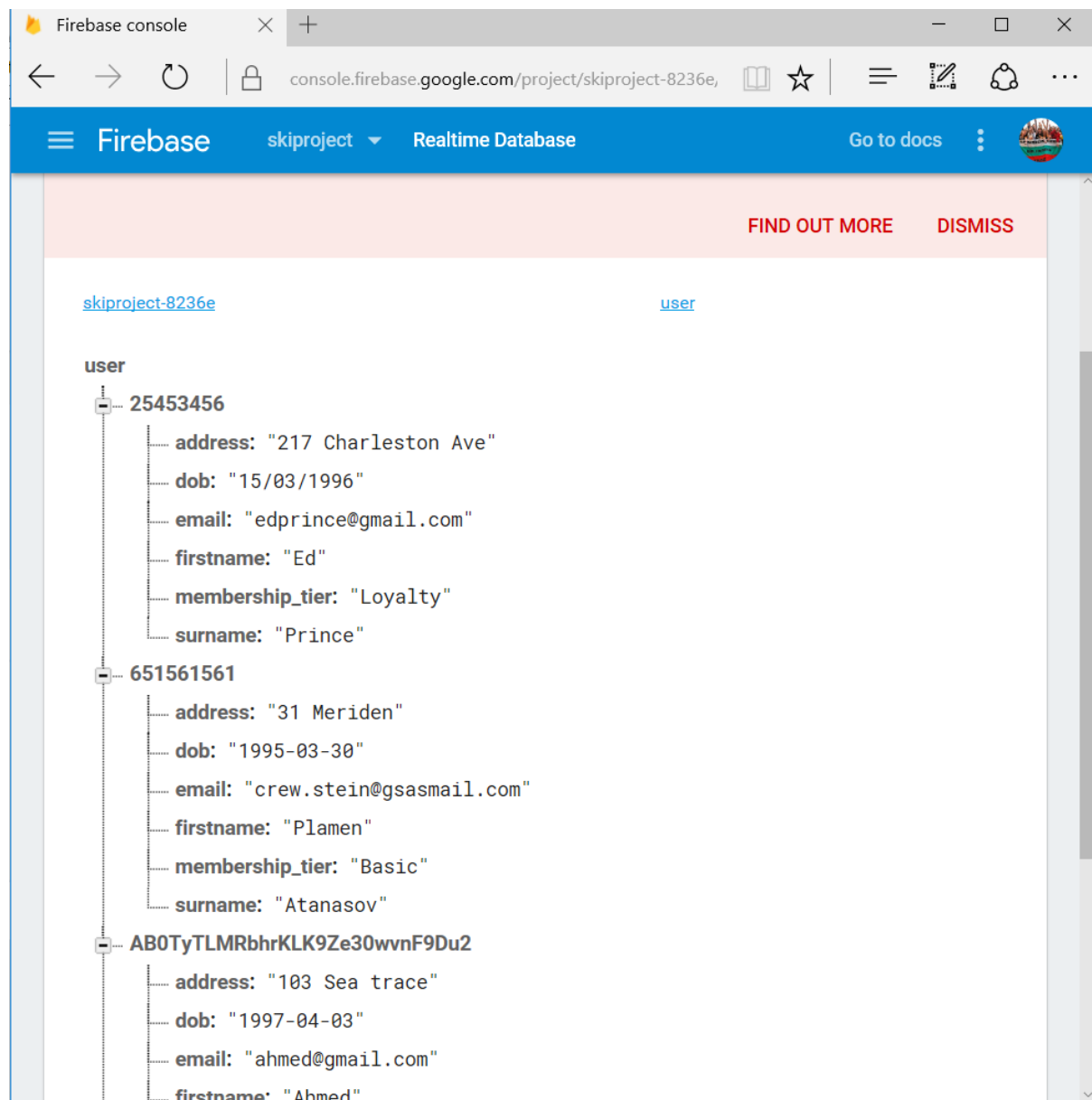
We used Firebase Database and Hosting for our project (by Google).

<https://skipproject-8236e.firebaseio.com/> - project

Main page: from here the Operator can choose to Add new user, Update Details or change membership:

Sphere - Welcome

[Add new user](#)
[Update Details](#)
[Become Free Member](#)
[Become Loyalty Member](#)



The screenshot above shows our database.

Integrated with Ed's functionality – Register user:

Sphere - Register

Would you like to become a member for free

☒ Yes ☐ No

SUBMIT

- FIRST PROTOTYPE**

This is what my page looked like before producing the second prototype.

Sphere - Update Details

SEARCH

First Name	Surname	Address	Email	Membership Status	Date of birth	Make Changes
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<div>UPDATE</div> <div>DELETE</div>

CLEAR

The operator was able to Search for a Member by First Name, Update Details, Delete Member and Clear the page.

This is how it worked:

The Operator enters a **Name** (First Name), clicks on the **Search** button and the details were being **displayed** underneath as shown below.

Sphere - Update Details

Plamen

SEARCH

First Name	Surname	Address	Email	Membership Status	Date of birth	Make Changes
Plamen	Atanasov	31 Meriden st	crew.stein@gmail.com	Basic	30/03/1995	<div>UPDATE</div> <div>DELETE</div>

CLEAR

To update something, let's say Surname – the operator had to click on that field (these are editable fields – apart from Membership Status, which is **read** only field) and change it then press **Update**.

Sphere - Update Details

Plamen

SEARCH

First Name	Surname	Address	Email	Membership Status	Date of birth	Make Changes
Plamen	Atanasov	31 Meriden st	crew.stein@gmail.com	Basic	30/03/1995	<div>UPDATE</div> <div>DELETE</div>

CLEAR

In my case I changed my **Surname** from **Atanasov** to **Ivanov**.

After clicking on the **Update** button, the updated member is being displayed underneath:

Sphere - Update Details

SEARCH

First Name	Surname	Address	Email	Membership Status	Date of birth	Make Changes
<input type="text" value="Plamen"/>	<input type="text" value="Ivanov"/>	<input type="text" value="31 Meriden st"/>	<input type="text" value="crew.stein@gmail.com"/>	<input type="text" value="Basic"/>	<input type="text" value="30/03/1995"/>	<div>UPDATE</div> <div>DELETE</div>
Plamen	Ivanov	31 Meriden st	crew.stein@gmail.com	Basic	30/03/1995	

CLEAR

As you can see from the screenshot above, the updated details are displayed underneath and **Surname** was **changed** to Ivanov.

After providing it to the client (Dr. Yih Ling), she tested it and wasn't happy that I had missed to implement any validation testing for wrong input. For example: she tried to enter invalid name (containing special character), invalid date of birth (30th of February) and the system successfully sent the data back to the database. Also, she didn't like the fact that **nowhere** was labelled that Search is possible only by First Name.

So, I implemented Validation which alerts the Operator with a **pop out message**.

- **SECOND PROTOTYPE**

Search menu labelled and calendar implemented (suggested by the client)

Sphere - Update Details

Search by First Name

SEARCH

First Name	Surname	Address	Email	Membership Status	Date of birth	Make Changes																																																	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="mm/dd/yyyy"/>	<div>UPDATE</div> <div>DELETE</div>																																																	
					<div>April 2017</div> <table><tr><td>Sun</td><td>Mon</td><td>Tue</td><td>Wed</td><td>Thu</td><td>Fri</td><td>Sat</td></tr><tr><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td><td>1</td></tr><tr><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr><tr><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr><tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td></tr><tr><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td></tr><tr><td>30</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>	Sun	Mon	Tue	Wed	Thu	Fri	Sat	26	27	28	29	30	31	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																																	
26	27	28	29	30	31	1																																																	
2	3	4	5	6	7	8																																																	
9	10	11	12	13	14	15																																																	
16	17	18	19	20	21	22																																																	
23	24	25	26	27	28	29																																																	
30	1	2	3	4	5	6																																																	

CLEAR

Validation for name and email: tried to change my name to Plamen! and a message popped out **not letting me update.**

Sphere - Update Details

Plamen

SEARCH

First Name	Surname	Address	Email	Membership Status	Date of birth	Make Changes
Plamen!	Atanasov	31 Meriden		Basic	03/30/1995	<div>UPDATE</div> <div>DELETE</div>

CLEAR

skipproject-8236e.firebaseio.com says:
Invalid input. Please re-enter details.

OK

Email: **empty field**

Sphere - Update Details

Plamen

SEARCH

First Name	Surname	Address	Email	Membership Status	Date of birth	Make Changes
Plamen	Atanasov	31 Meriden		Basic	03/30/1995	<div>UPDATE</div> <div>DELETE</div>

CLEAR

skipproject-8236e.firebaseio.com says:
Invalid input. Please re-enter details.

OK