**Project EasyTravel**

**User journey:**

The EasyTravel application allows the user:

* Manage Account
  + Authentification : this step allows the user to open the account using an email/username and password.
  + Account management (add-delete-reset/change password- recommend):
    - Add Account: create a new account, the user should enter the following information: First name, last name, Unique UserName, Unique Email, Password, Birthdate, phone number
  + Nice to have: create an account via FaceBook API
* Booking a travel product
* Publish/Share the booking

|  |  |  |
| --- | --- | --- |
| Must have | In between | Nice to have |
|  |  |  |

**Backend Technical flow**:

**Authentification :**

1. Retrieve from the UI the username/email as string.
2. Check if the username exists : Suggestion : using AJAX tech
   1. If exists: go to step3
   2. Trigger an error
3. Retrieve the password from UI
4. Hash the password
5. Compare the password hash with the one in db: “ select hashPW from User where Username =? Or Email=? “
   1. Case 1: null , trigger error 1 : wrong password/username
   2. Case2: not equal, trigger error 1

**Benchmark example** : sign in Datacamp.com flow

DB Schema draft:

|  |  |  |  |
| --- | --- | --- | --- |
| Username | Email | BirthDate | PWHash |
|  |  |  |  |

**Account management**

**Forget Password:**

1. Retrieve the email/user name from UI
2. Send an email with reset link
   1. Reset link: valid just for a period of time (24H) via a security token
3. Compare the password with the repetition : JS
   1. If they are equal and not empty activate the confirm button
4. Set the PWhash in the DB : “Update User set PWhash=? where username=?”

DB Table TOKEN draft:

|  |  |  |  |
| --- | --- | --- | --- |
| Token | UserName | Valid | AddedDate |
|  |  |  |  |

**Add new Account:**

1. Retrieve from the UI the following fields:
   1. Username: Via AJAX, check if the username exists in the DB and > 6 characters:
      1. If exists: trigger an error 2: the username already exists
      2. If not: validate the username
   2. Email : Via AJAX, check if the email exists in the DB:
      1. If exists: trigger an error 2: the email already exists
      2. If not: validate the email
   3. First Name
   4. Last Name
   5. BirthDate
   6. Password : JS , should contain one upper character , a number, a special character, more than 8 characters.
   7. If all fields respects the set constraints activate the button
2. Update the DB with the new user information: insert into User values(Username, ….); and valid=0 (as the account is not valid before the user confirmation)
3. Send an email to validate/confirm the account (link valid for 48 hours):
   1. If the user validates within 48 hours delete the account (call Delete account UC)
   2. Validate the User account. Set the valid attribute to 1 as valid