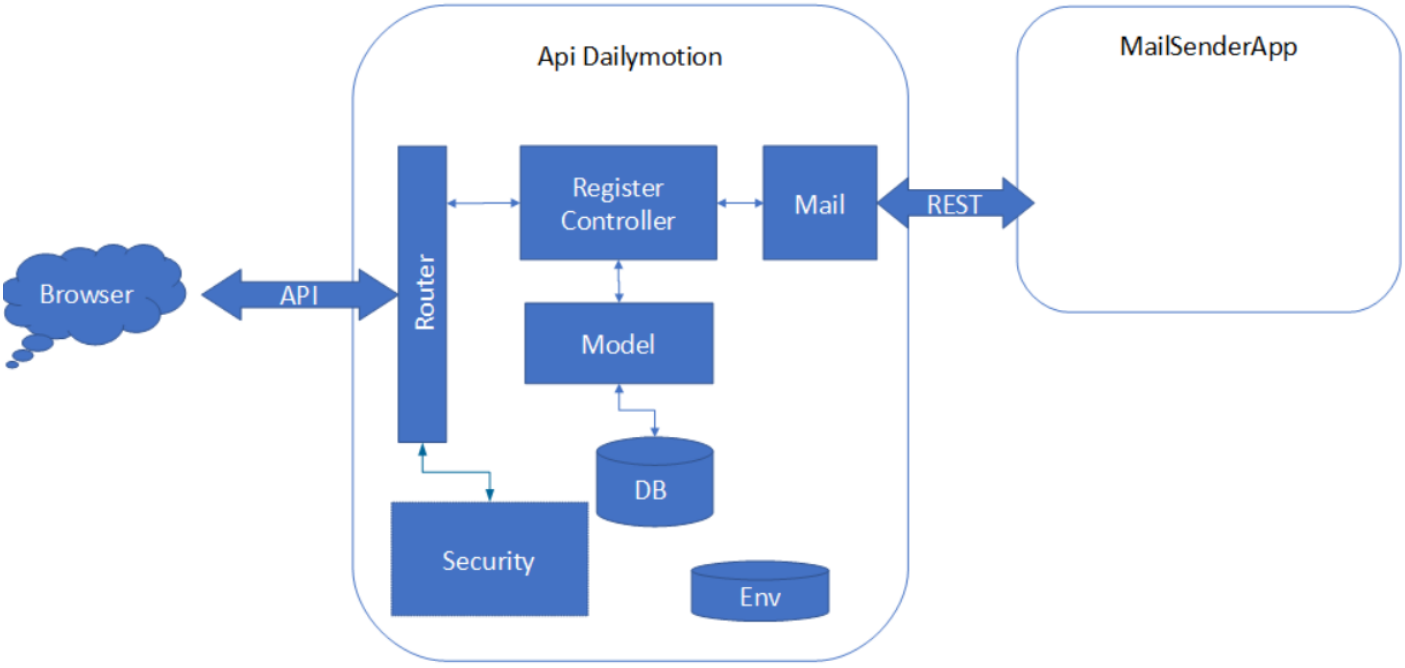
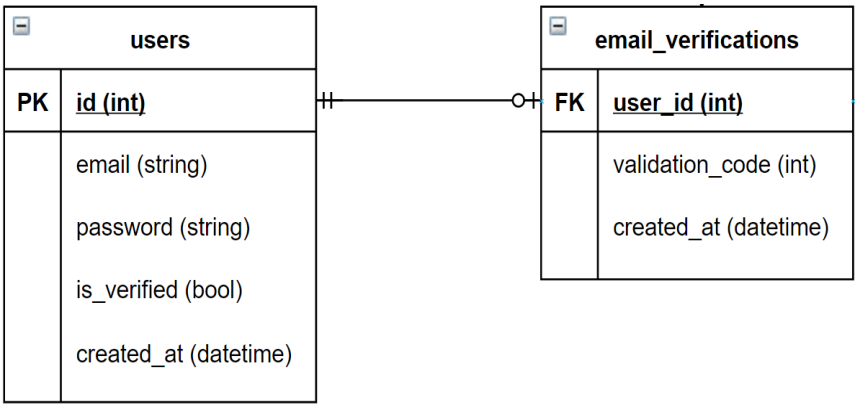
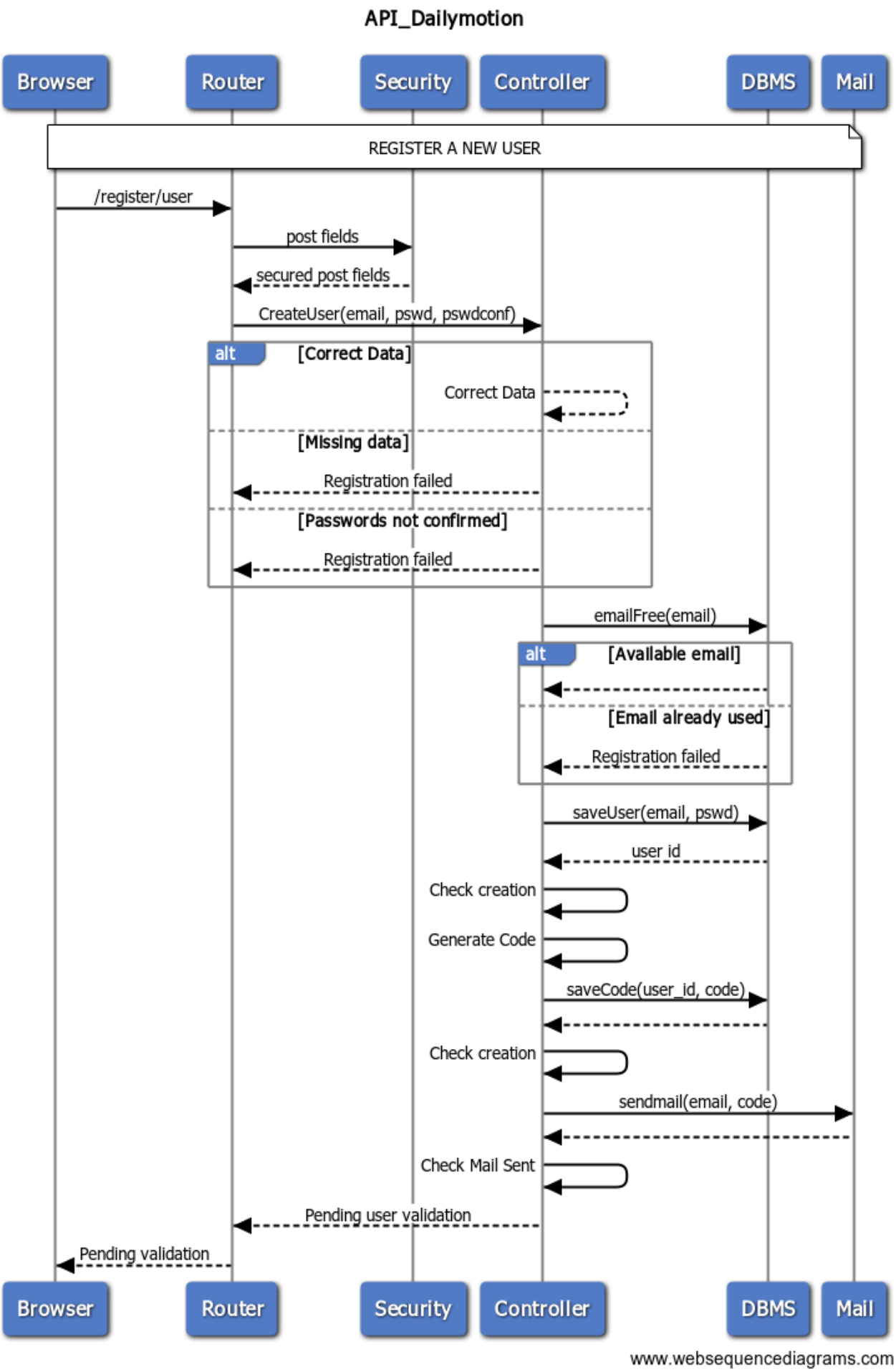
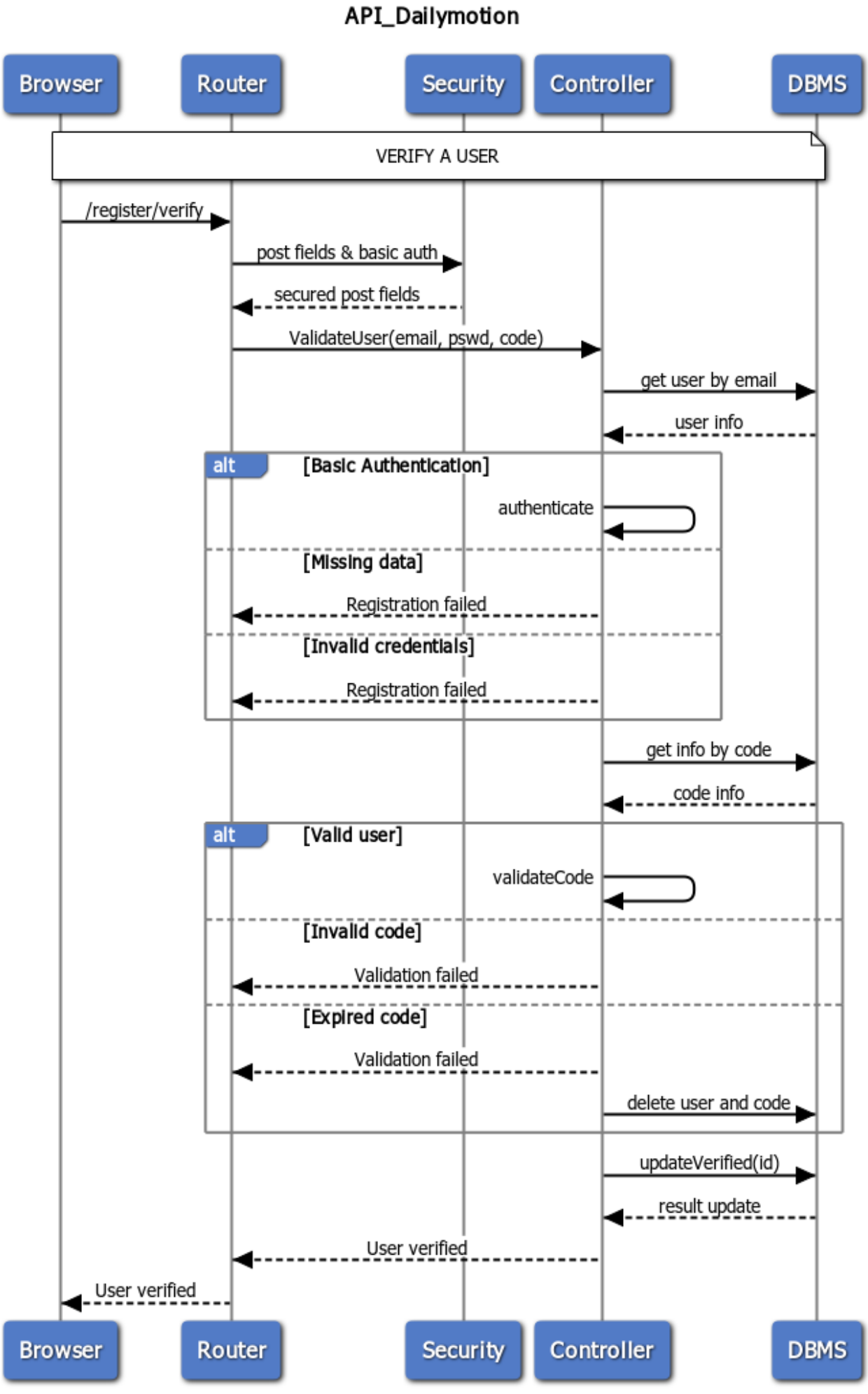
Project's architecture



Database tables




API_Dailymotion
Browser
Router
Security
Controller
DBMS
Mail
REGISTER A NEW USER
/register/user
post fields
secured post fields
CreateUser(email, pswd, pswdconf)
alt
[Correct Data]
Correct Data
[Missing data]
Registration failed
[Passwords not confirmed]
Registration failed
emailFree(email)
alt
[Available email]
[Email already used]
Registration failed
saveUser(email, pswd)
user id
Check creation
Generate Code
saveCode(user_id, code)
Check creation
sendmail(email, code)
Check Mail Sent
Pending user validation
Pending validation
Browser
Router
Security
Controller
DBMS
Mail
www.websequencediagrams.com


API_Dailymotion
Browser
Router
Security
Controller
DBMS
VERIFY A USER
/register/verify
post fields & basic auth
secured post fields
ValidateUser(email, pswd, code)
get user by email
user info
alt
[Basic Authentication]
authenticate
[Missing data]
Registration failed
[Invalid credentials]
Registration failed
get info by code
code info
alt
[Valid user]
validateCode
[Invalid code]
Validation failed
[Expired code]
Validation failed
delete user and code
updateVerified(id)
result update
User verified
User verified
Browser
Router
Security
Controller
DBMS
www.websequencediagrams.com

Ideas for improvement:

- create a route to resend an email with a new validation_code in case the user didn't receive the email or lost the validaction_code
- develop a purge for the table users and email_verifications, for example a daily job checking for non verified users and expired codes.
- install a package (like dotenv) to store environment variables into a .env file
- define security criteria for password (mandatory use of special characters, …)
- instead of the field 'is_validated', use a 'status' field with constants ('pending', 'active', 'inactive', 'blocked'...) according to the needs