}

return $this->auth\_key;

/\* \*

\* @inheritdoc \*/

public function validateAuthKey($authKey)

{

return $this->getAuthKey() === $authKey;

}

/\*\*

\* Validates password

\*

\* @param string $password password to validate

\* @return boolean if password provided is valid for current user \*/

public function validatePassword($password)

{

return Yii::$app->getSecurity()->validatePassword($password, $this- >password\_hash);

}

/\*\*

\* Generates password hash from password and sets it to the model

\*

\* @param string $password \*/

public function setPassword($password)

{

$this->password\_hash = Yii::$app->getSecurity()- >generatePasswordHash($password);

}

/\*\*

\* Generates "remember me" authentication key \*/

public function generateAuthKey()

{

$this->auth\_key = Yii::$app->getSecurity()->generateRandomString();

}

/\*\*

\* Generates new password reset token \*/

public function generatePasswordResetToken()

{

$this->password\_reset\_token = Yii::$app->getSecurity()- >generateRandomString() . '\_' . time();

}

/\*\*

\* Finds user by password reset token

\*

\* @param string $token password reset token

\* @return static|null \*/

public static function findByPasswordResetToken($token)

{

$expire = Yii::$app->params['user.passwordResetTokenExpire'];

$parts = explode('\_', $token);

$timestamp = (int) end($parts); if ($timestamp + $expire < time()) { return null;