Time to Pick Up the PACE

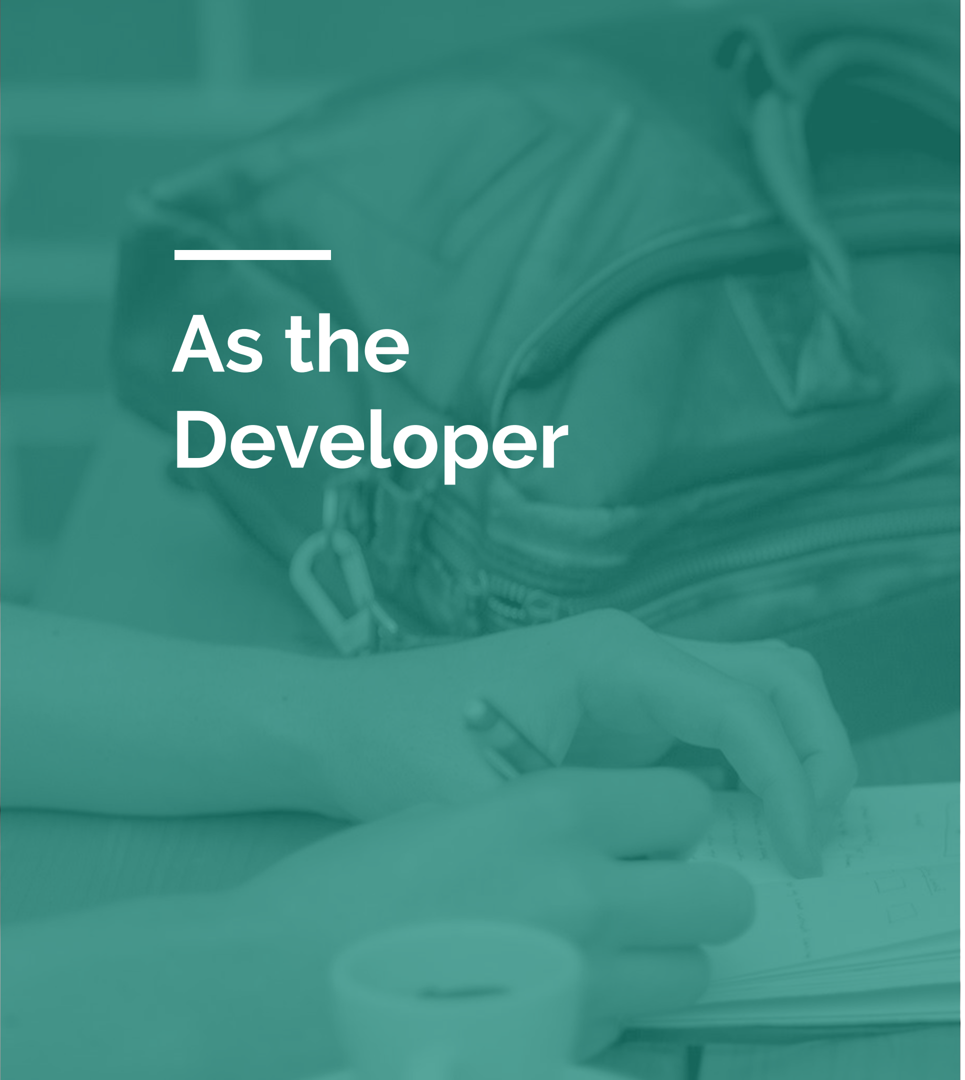
A Bespoke Proposal

vestibulum nibh, accumsan sagittis turpis nisl eu mauris. Mauris ut laoreet sapien. Nam suscipit, dui at condimentum accumsan, dui sapien tincidunt.

Disclaimer: PACE Bank and its affiliates offer investment products,

Roles Played:

- Developer
- Business Analyst
- Software Engineer



As the Developer

Built the PACE Bank web application using:

Front-End:

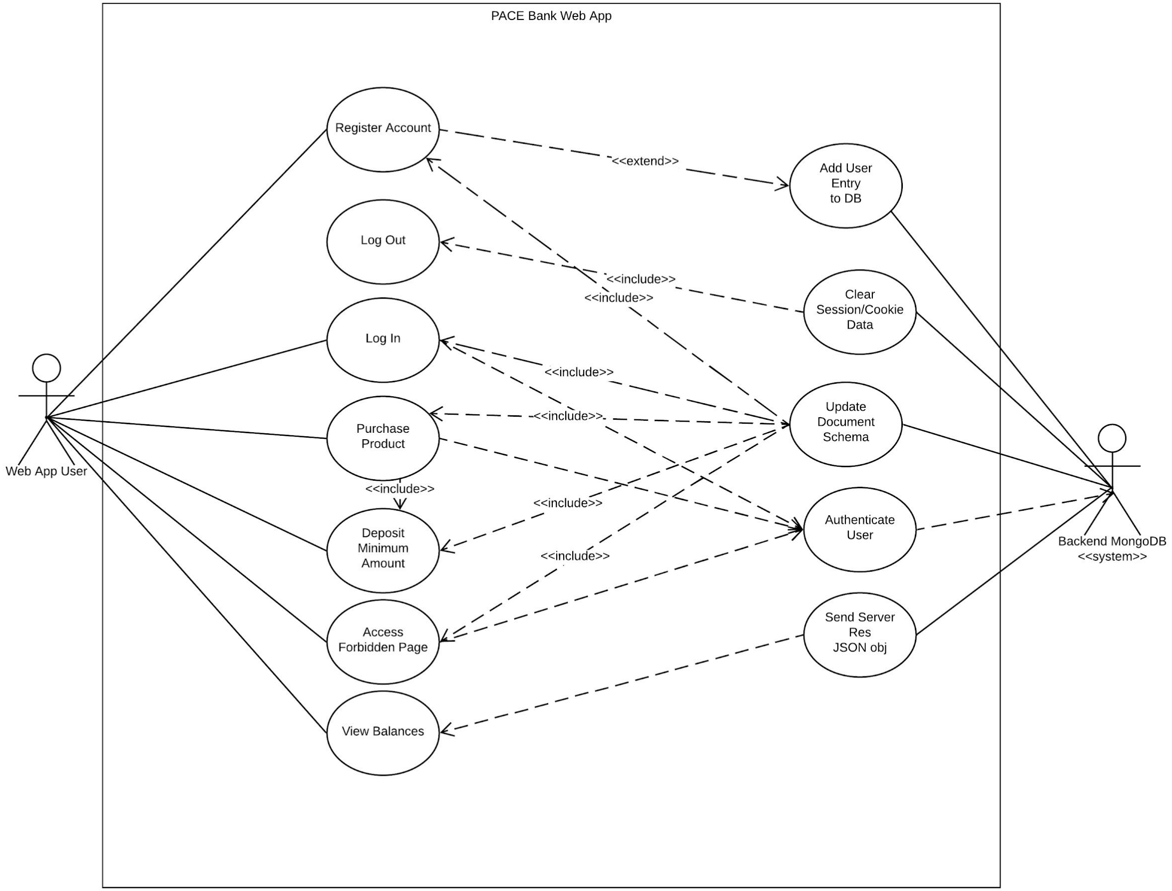
- JavaScript
- Node.js (w/ Express)
- CSS (w/ Boot Strap)
- HTML (w/ Pug engine & jQuery)

Back-End:

- MongoDB
- Using Mongoose as the ODM Library

Along with a few other npm modules including nodemon and bcrypt.

Use Case Model

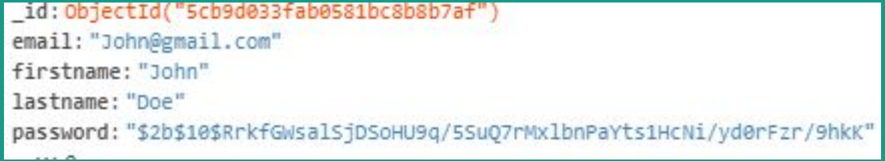


Use cases

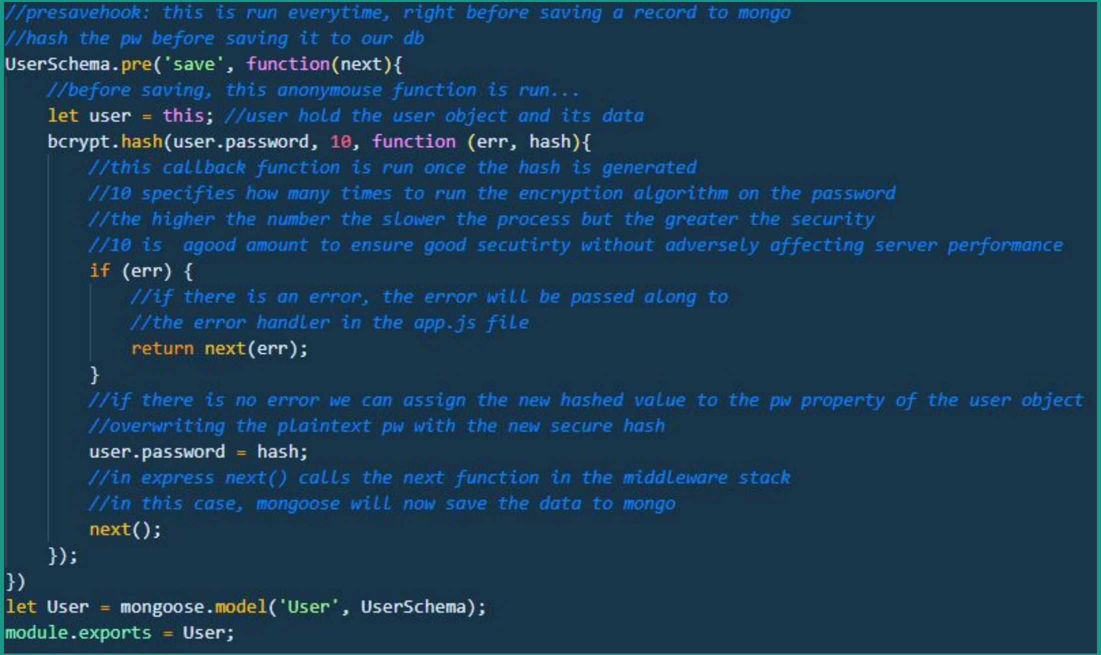
- ➔ As the User you can register, apply for accounts, and easily view your balances.
- ➔ As the Administrator you can authenticate roles will little effort.
- ➔ As the DBA you can store cookies and session data.
- ➔ You can even easily specify the strength of the security.

Security (with extra salt)

```
let bcrypt = require('bcrypt');
```



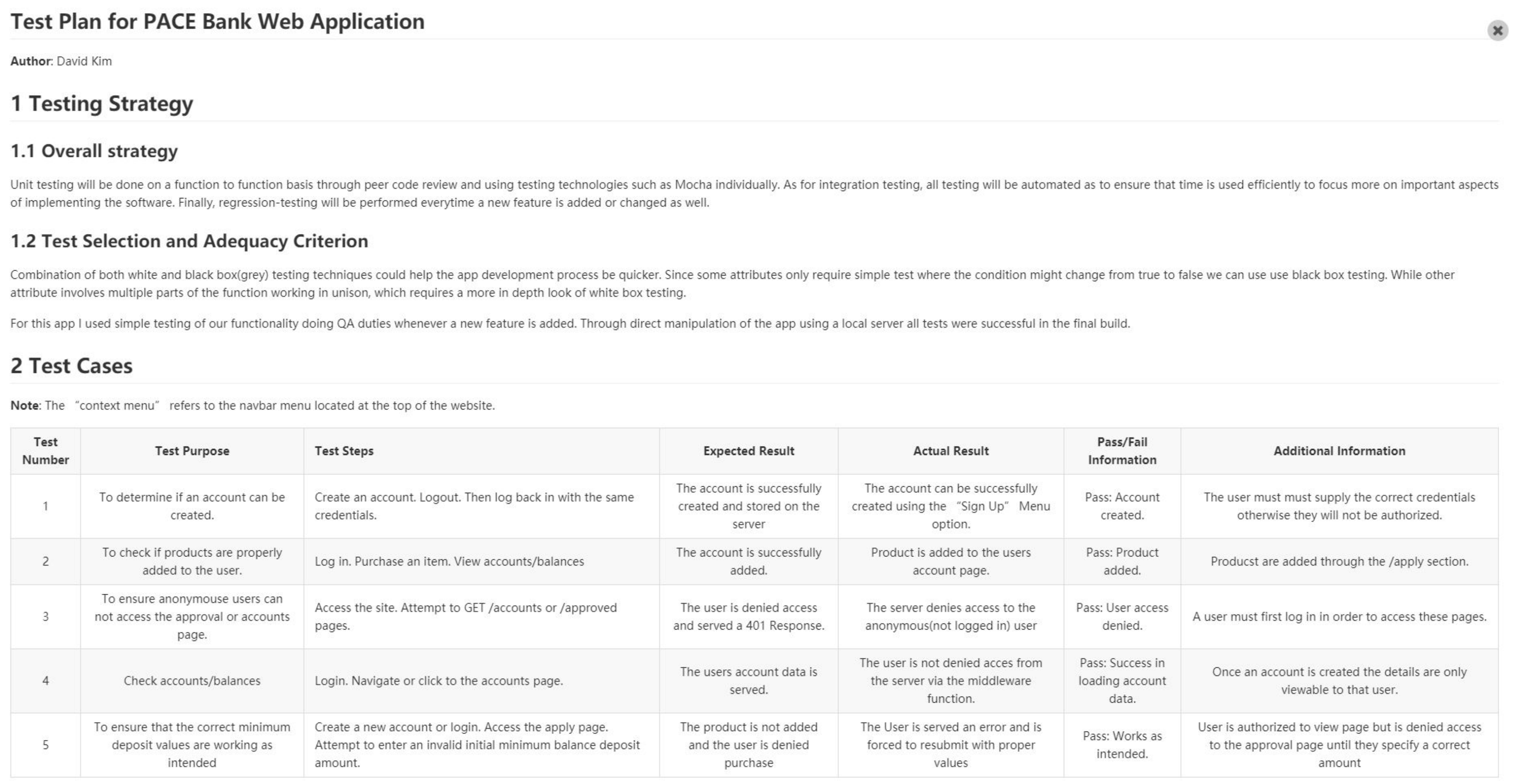
```
_id: ObjectId("5cb9d033fab0581bc8b8b7af")
email: "John@gmail.com"
firstname: "John"
lastname: "Doe"
password: "$2b$10$RrkfGWsalSjDSoHU9q/5SuQ7rMxlbnPaYts1HcNi/yd0rFzr/9hkK"
```



```
//presavehook: this is run everytime, right before saving a record to mongo
//hash the pw before saving it to our db
UserSchema.pre('save', function(next){
    //before saving, this anonymouse function is run...
    let user = this; //user hold the user object and its data
    bcrypt.hash(user.password, 10, function (err, hash){
        //this callback function is run once the hash is generated
        //10 specifies how many times to run the encryption algorithm on the password
        //the higher the number the slower the process but the greater the security
        //10 is  agood amount to ensure good secutirty without adversely affecting server performance
        if (err) {
            //if there is an error, the error will be passed along to
            //the error handler in the app.js file
            return next(err);
        }
        //if there is no error we can assign the new hashed value to the pw property of the user object
        //overwriting the plaintext pw with the new secure hash
        user.password = hash;
        //in express next() calls the next function in the middleware stack
        //in this case, mongoose will now save the data to mongo
        next();
    });
})
let User = mongoose.model('User', UserSchema);
module.exports = User;
```

- ➔ A salt can then be generated for each hash value and then concatenated to it.

Non-functional testing demonstrating security, reliability, scalability, usability



Test Plan for PACE Bank Web Application

Author: David Kim

1 Testing Strategy

1.1 Overall strategy

Unit testing will be done on a function to function basis through peer code review and using testing technologies such as Mocha individually. As for integration testing, all testing will be automated as to ensure that time is used efficiently to focus more on important aspects of implementing the software. Finally, regression-testing will be performed everytime a new feature is added or changed as well.

1.2 Test Selection and Adequacy Criterion

Combination of both white and black box(grey) testing techniques could help the app development process be quicker. Since some attributes only require simple test where the condition might change from true to false we can use use black box testing. While other attribute involves multiple parts of the function working in unison, which requires a more in depth look of white box testing.

For this app I used simple testing of our functionality doing QA duties whenever a new feature is added. Through direct manipulation of the app using a local server all tests were successful in the final build.

2 Test Cases

Note: The "context menu" refers to the navbar menu located at the top of the website.

Test Number	Test Purpose	Test Steps	Expected Result	Actual Result	Pass/Fail Information	Additional Information
1	To determine if an account can be created.	Create an account. Logout. Then log back in with the same credentials.	The account is successfully created and stored on the server	The account can be successfully created using the "Sign Up" Menu option.	Pass: Account created.	The user must must supply the correct credentials otherwise they will not be authorized.
2	To check if products are properly added to the user.	Log in. Purchase an item. View accounts/balances	The account is successfully added.	Product is added to the users account page.	Pass: Product added.	Producst are added through the /apply section.
3	To ensure anonymouse users can not access the approval or accounts page.	Access the site. Attempt to GET /accounts or /approved pages.	The user is denied access and served a 401 Response.	The server denies access to the anonymous(not logged in) user	Pass: User access denied.	A user must first log in in order to access these pages.
4	Check accounts/balances	Login. Navigate or click to the accounts page.	The users account data is served.	The user is not denied acces from the server via the middleware function.	Pass: Success in loading account data.	Once an account is created the details are only viewable to that user.
5	To ensure that the correct minimum deposit values are working as intended	Create a new account or login. Access the apply page. Attempt to enter an invalid initial minimum balance deposit amount.	The product is not added and the user is denied purchase	The User is served an error and is forced to resubmit with proper values	Pass: Works as intended.	User is authorized to view page but is denied access to the approval page until they specify a correct amount

As the Business Analyst

Bespoke VS. Commercial Off the Shelf

- Customer reach through rapid deployment and maneuverability.
- Tailored just for you and can be easily developed to match your companies look and feel.
- Easily evolves with customer trends.
- Full control over data
- Even though COTS is cheaper and proven, the benefits of bespoke development outweigh it.

What next?

- ➔ Google Maps API
- ➔ Listen carefully to customer feedback
- ➔ Data analysis