SW Engineering CSC648-848-05 Fall2023

OrderOwl

Team 06

Team Lead: Rita Belú Velazco, rvelazco@mail.sfsu.edu

Front End Lead: Jimmy Pan

Back End Lead: Tin Nguyen

GitHub Master: Luis Ramirez

Document Editor: Mankit Yeung

M4 Editor: Komaldeep Kaur

Database Master: David Lien

Milestone 4

December 2, 2023

History Table:

Revision	Date
M4V2	
M4V1	12/02/23
M2V2	12/02/23
M2V1	11/02/23

M2V2	11/02/23
M2V1	10/13/23
M1V2	10/12/23
M1V1	09/21/23

Table of Contents

I. Product summary:

Name of product: Order Owl
URL: https://orderowl.jpkit.us/

Nowadays, online shopping has become a part of people's everyday lives, and, to be frank, so has the frustration of keeping track of multiple orders from various retailers too. This is where OrderOwl steps in to become every online shopper's little assistant. Our product offers a centralized solution for tracking all your packages, regardless of where they're coming from. From international orders, to orders from your local stores, our simple motto says it all: "Track Your Order Anywhere, Anytime". With OrderOwl, the days of scrolling through countless emails or visiting numerous websites to check the status of your deliveries are over. Our platform is user-friendly, efficient, and requires no commitment, allowing anyone to quickly check the delivery status of any package using just a tracking number. Our application not only attends to personal use, but also for businesses of any size who could use a helping hand with their shipments or product orders.

For those who choose to register with OrderOwl, the benefits are even greater. Registered users have the convenience of saving all their order information in one place. Simply enter your tracking number once, and OrderOwl will automatically update you on the progress of your delivery through notifications and visual map updates. This registration not only saves time but also helps in organizing your orders effectively. Furthermore, the platform allows you to personalize your experience by organizing your orders into custom folders sorted by retailers, title your orders to a label that best helps you, or try out one of OrderOwl's unique features of listing an order as "hidden" so it does not appear on your main dashboard. OrderOwl offers a level of customization like no other so that your dashboard is tailored to your specific needs and preferences, making online shopping more organized and less stressful. This application also comes with a calendar that will give you a nice overview of when orders are set to be delivered. For businesses, one gets everything that a personal registered account would have, with the addition of reminders for frequent orders, and a separate page dedicated to shipments from your business; whether it be a small hand-crafts business to a larger clothing store. Let OrderOwl help support and make your business thrive! The platform is designed with the user in mind through its intuitive interface which makes navigation simple and straightforward; even for those who are not as tech-savvy. The website's design is sleek and modern, ensuring that users have a pleasant experience each time they log in.

In conclusion, OrderOwl is more than just a tracking tool; it's the overall solution for managing online orders. It simplifies the process of tracking packages from multiple retailers, offers a range of customization options such as business/personal accounts, and provides a user-friendly interface that makes online shopping a breeze. Whether you're a frequent online shopper or occasionally purchase items from the internet, OrderOwl is here to make your life easier and your online shopping experience more enjoyable.

a) Priority 1 Functions:

Registered Users:

1.1 Registered Users shall be able to track all their packages with tracking information.

1.2 Registered Users shall be able to add tracking information.

1.3 Registered Users shall be able to login.

1.4 Registered Users shall be able to create an account.

1.5 Registered Users shall be able to update their profile information.

1.7 Registered Users shall be able to see their history of completed tracked orders.

1.9 Registered Users shall be able to sort their tracking information.

1.20 Registered Users shall be able to view the overview calendar.

1.21 Registered Users shall be able to view any month within a year in advance or within the past year.

1.22 Registered Users shall optionally be able to provide their feedback for the application through email.

1.23 Registered Users shall be able to choose whether they would like a personal or business account when they register.

1.25 Registered Users shall be able to choose to register for an account or continue as a guest

1.26 Registered Users shall be able to label each tracked package with a custom title.

1.27 Registered Users shall be able to view their tracked orders alphabetically.

1.28 Registered Users shall be able to search for any tracked order by their carrier name.

1.29 Registered Users shall be able to search for any tracked order by their order title.

1.30 Registered Users shall be able to search for any tracked order by their retailer.

1.31 Registered Users shall be able to view a counter of currently tracked orders.

1.32 Registered Users shall be able to view their tracked orders in reverse alphabetical order.

1.33 Registered Users shall be able to get a list of most to least relevant results when doing a search.

1.34 Registered Users can send an email for help

1.35 Registered Users can send an email for questions

1.36 Registered Users can access the support page

1.37 Registered Users can access the About Me page to learn about the team.

1.38 Registered Users shall be able to sort their tracked orders into hidden or visible lists.

1.39 Registered Users shall be able to delete tracking information.

1.40 Registered Users shall be able to create their own titles for tracked orders.

1.41 Registered Users shall be able to delete their account.

1.42 Registered Users shall be able to create a custom pin to access their hidden list

1.43 Registered Users shall be able to sort their tracked orders into hidden

1.44 Registered Users shall be able to access their hidden lists.

1.45 Registered Users shall be able to save all their tracked orders to their accounts.

1.46 Registered Users shall be able to edit their account information after initial registration.

1.47 Registered Users shall be able to edit their account's first and last name after registration

1.47 Registered Users shall be able to edit their account's first and last name after registration

1.48 Registered Users shall be able to receive email notifications of tracking updates.

Unregistered Users:

9.1 UnregisteredUsers shall be able to track all their packages with tracking information.

9.2 UnregisteredUsers shall be able to add tracking information.

9.3 UnregisteredUsers shall be able to create an account.

9.4 UnregisteredUsers shall be able to sort their tracking information.

9.5 UnregisteredUsers shall be able to view the overview calendar.

9.6 UnregisteredUsers shall be able to view any month within a year in advance or within the past year.

9.7 UnregisteredUsers shall optionally be able to provide their feedback for the application through email.

9.8 UnregisteredUsers shall be able to choose to register for an account or continue as a guest

9.9 UnregisteredUsers shall be able to label each tracked package with a custom title.

9.10 UnregisteredUsers shall be able to view their tracked orders alphabetically.

9.11 UnregisteredUsers shall be able to search for any tracked order by their carrier name.

9.12 UnregisteredUsers shall be able to search for any tracked order by their order title.

9.13 UnregisteredUsers shall be able to search for any tracked order by their retailer.

9.14 UnregisteredUsers shall be able to view a counter of currently tracked orders.

9.15 UnregisteredUsers shall be able to view their tracked orders in reverse alphabetical order.

9.16 UnregisteredUsers shall be able to get a list of most to least relevant results when doing a search.

9.17 UnregisteredUsers can send an email for help

9.18 UnregisteredUsers can send an email for questions

9.19 UnregisteredUsers can access the support page

9.20 UnregisteredUsers can access the About Me page to learn about the team.

9.21 UnregisteredUsers shall be able to sort their tracked orders into hidden or visible lists.

9.22 UnregisteredUsers shall be able to delete tracking information.

9.23 UnregisteredUsers shall be able to create their own titles for tracked orders.

Business Accounts:

8.1 Business Accounts shall be able to have access to all personal account features.

8.2 Business Accounts shall be able to upload tracking numbers for their shipments.

8.3 Business Accounts shall be able to upload their business license.

9.4 Business Accounts shall have their shipments and deliveries tracked on the calendar.

Admin:

2.1 Admin shall be able to access all data.

2.2 Admin shall be able to modify any data.

2.4 Admin shall be able to create any type of account.

2.6 Admin shall be able to view another account.

2.10 Admin shall have the ability to conduct system maintenance and updates without disrupting user access.

2.11 Admin shall be able to receive and review user feedback and reports regarding system performance and functionality.

System:

3.1 System shall be able to collect traffic data.

3.2 System shall be able to store traffic data to improve user experience.

Notification:

4.9 Notifications shall be sent to users via email regarding tracking updates.

Tracking Information:

5.1 Tracking information shall be able to be stored.

5.3 Tracking information shall maintain an up to date dispatch alerts.

5.2 Tracking information shall be able retrieved in real time according to shipping carrier updates

5.4 Tracking information shall maintain an up to date database.

5.5 Tracking information shall be accessible through a mobile application for on-the-go access.

5.6 Tracking information shall update the overview calendar with ETA dates for tracked orders.

Overview Calendar:

6.1 Overview Calendar shall be able to mark the dates when orders are estimated to arrive.

6.2 Overview Calendar shall begin on the month a registered user signs up on.

6.3 Overview Calendar shall show the estimated deliveries of the month.

6.4 Overview Calendar shall contain embedded links for tracked orders.

6.5 Overview Calendar shall keep track of estimated deliveries.

6.6 Overview Calendar shall be updated according to any estimated delivery changes.

II. Usability test plan:

Objective:

Our objective for our usability test plan aims to introduce to our users a set of innovative features that you wouldn't commonly see in other applications or software. **The features include the ability to add titles to orders, seamlessly check/monitor delivery dates on the calendar, hide orders that have been uploaded to their account, create a separate business account with the name of the business and a personal account, and be able to sort your orders at your liking**. The inclusion of these functions are important for our software because we don't see a lot of other applications that allow users to use these types of features. It is also important for us to have these functions in order to give reason to the user to use our software rather than just using the site they ordered from. This is something we focused on as well because we can offer a more personalized and efficient platform such as being able to sort and track your orders at your own timing. Additionally, we also want to emphasize user safety and privacy when it comes to the items they order particularly through the option to hide an order. Testing the calendar feature we offer is vital to ensure users can effectively track their delivery dates.

Descriptions per Function:

i. **System setup:**

In order for the system to be set up correctly, we need to make sure that when you can see our homepage the buttons and images load just fine. They should start off by registering in order to use all of our features that we have to offer. This is important in order to have hidden orders on your account, being a guest would not be possible to upload an order as hidden. If there are no orders set up on our site, you can try uploading these tracking numbers, "ABC123456" or "XYZ101010". Once you use one of these tracking numbers you should see the information of the order on your screen.

ii. **Starting point:**

The user should be starting at our homepage of orderowl. In this location they should see the option of registering with us. All of our functions should be starting here because this is the main part where you can get to everything you need regarding uploading and tracking orders.

iii. **Intended Users:**

Being able to sort functions, we aim this feature to those that make a lot of orders and want to be able to find specific ones easily. Changing the name of your order makes this even better because you can have similar names of orders and find them easily when sorting. This also helps those that need to keep the name of the order private.

iv. **URL:**

The user should be going to our site, “orderowl.jpkit.us”.

Tests/Use Cases	% Completed	Errors	Comments	Efficiency
Add Title to Order	100%	none	User was freely able to add a title to their order	5 seconds
ETA Orders on Calendar	100%	none	User can see their orders on calendar if delivered in the month.	5 seconds
Hide Order	50%	Hidden list won’t show up	User was able to upload their order into hidden but when they access hidden list, their order does not show up	10 seconds
Business or Personal Account	50%	No errors	User feels like the difference between both accounts aren’t large.	10 seconds
Sort Orders	100%	none	User was able to sort their orders depending on the column.	5 seconds

Questionnaire:

Adding a Title to Your Order	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It was easy to add a title to your order					check

The title was uploaded after you inserted it					check
Adding title to order was helpful				check	

Check Delivery Dates on Calendar	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Your orders appeared up on the calendar			check		
You see the order on the date it will be delivered					check
The order you uploaded appeared on the calendar			check		

Hiding an Order	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
You were able to upload the hidden order to the hidden list					check

You see all the hidden orders that were uploaded	check				
Your hidden order pin let you access your hidden order list		check			

Create a Business or Personal Account	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
When registering you had the option of choosing between user personal and business					check
User sees the difference between the personal pages and the business pages			check		
You were able to access both type of accounts, personal and business			check		

Sorting Orders	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
You were able to find what to click on to change the sorting of your orders					check
All the sorting options worked properly (Retailer, carrier, tracking number, location, and address)					check
Sorting options were helpful to find specific orders			check		

III. QA test plan:

1. **Updates to tracking information should happen in real-time, with no more than a 5-second delay from the actual event.**

Test objectives:

- Objectives
    1. Ensure that the system update the tracking information in real time
    2. Make sure that the delay in updating tracking information does not exceed 5 seconds from the actual event.
- Purpose of Objectives
    1. To ensure the system's ability to have timely and accurate updates.
    2. To ensure the system meets the ideal performance criteria set for real-time updates.

HW and SW setup:

- URL (orderowl.jpkit.us)
- Any browser

Feature to be tested:

- The real-time tracking of the upload

QA Test Plan:

Number #	Test Title	Description	Test Input	Expected Output	Pass/Fail
1	Check system stability	Test Amazon order # upload	"A2B4C6"	Information shows up no more than five second	pass
2	Delay time check	Test eBay order # in upload	"123456789012"	Information shows up no more than five second	pass
3	Check if multiple order upload in short period but create time delay	Test if uploading multiple order # would affect the time	"A1B2C3D4" "A2B4C6"	Information shows up no more than five second	pass

2. **OrderOwl shall be tested on multiple browsers and devices to confirm the compatibility of the application.**

Test Objectives:

- Objectives
  1. Ensure that the website suitable across all browsers
  2. Ensure that the user has the same experience no matter what browser they are using.
- Purpose of the objectives
  1. To ensure cross-browser functionality, focusing on accessibility, usability, and user interface consistency.
  2. To identify any compatibility problem that could affect the user experience.

HW and SW setup:

- URL (orderowl.jpkit.us)
- Microsoft edge
- Firefox
- Chrome

Feature to be tested:

- Compatibility of the OrderOwl application on different web browsers

QA Test Plan:

Number #	Test Title	Description	Test Input	Expected Output	Pass/Fail
1	Check website compatibility through different browser	Test the website in Microsoft edge	URL provided	UI doesn't change due to different browser	Pass
2	Check website compatibility	Test the website in Firefox	URL provided	UI doesn't change due to different browser	Pass
3	Assess if application be able to perform the same experience no matter what browser are in	Test the website in Chrome	URL provided	UI doesn't change due to different browser	Pass

3. **Users shall be able to access the system at least 90% of the time.**

Test Objectives:

- Objectives
    1. To ensure that the system is accessible to users at least 90% of the time, have a high reliability.
    2. To measure and verify the system's uptime and accessibility.
- Purpose of the Objectives
    1. To measure the system's operational reliability and its ability to remain accessible to users in regular condition.
    2. To identify any potential issues that could lead to accessibility problems.

HW and SW setup:

- URL (orderowl.jpkit.us)
- Any browser

Feature to be tested:

- System availability and uptime.

QA Test Plan:

Number #	Test Title	Description	Test Input	Expected Output	Pass/Fail
1	Check website functionality in different time case 1	Test the application functionality at morning	try login	Login works perfectly	pass
2	Check website functionality in different time case 2	Test the application functionality at afternoon	try register an account	able to register an account	pass
3	Check website functionality in different time case 3	Test the application functionality at night time	try upload an order (Try "A1B2C3D4")	able to upload an order and can see it through dashboard or tracking page	pass

4. **Only authorized personnel should access sensitive data.**

Test Objectives:

- **Objectives**
  1. To ensure that the system's security effectively safeguards individual account data by ensuring that personal data is only accessible to the actual owner.
  2. To ensure that the system privacy controls prevent the visibility of personal information across different accounts.
- **Purpose of the Objectives**
  1. To ensure the system's security and privacy measures in place for protecting personal information.
  2. To ensure that the system has data protection which maintains user trust and privacy.

HW and SW setup:

- URL (orderowl.jpkit.us)
- Any browser

Feature to be tested:

- Access Control System for sensitive data.

QA Test Plan:

Number #	Test Title	Description	Test Input	Expected Output	Pass/Fail
1	Prevent unauthorized user to sensitive data	Test Unauthorized User Access	Continue as guest in the website and check homepage where it said recent, check dashboard and trackinfo page	User won't be able to see any order information in the dashboard, homepage and trackInfo page	Fail
2	Test security control at authorized user	Test Authorized User Access (login user)	Have a registered account and already uploaded a tracking number	After login user will be able to see their order information in homepage, dashboard and track info page	Fail
3	Assess system's ability at identifying role change	Test the Role Change Impact	Use that registered account that already create and have order uploaded to login check for orders and logout then continue as guest and check orders again	User will be able to see the order info when they login but when they logout then continue as guest they won't be able to see the order info they saw when they logged in	Fail

5. **System's user interface that's intuitive and easy to use for both personal and business accounts**.

Test Objectives:

- Objective
    1. To ensure that the system's user interface which is UI is intuitive and user-friendly for both personal and business account users.
    2. To ensure the UI design is easy to use, minimizing the learning curve and enhancing user experience.
- Purpose
    1. To measure the usability of the UI for different types of user, ensuring it meets the needs and expectations of both personal and business account users.
    2. Identifying any potential issue that causes confusion and inefficiency and to make sure the interface meets the satisfying standards of usability and accessibility.

HW and SW setup:

- URL (orderowl.jpkit.us)
- Any browser

Feature to be tested:

- User Interface Design and Usability for personal and business accounts.

QA Test Plan:

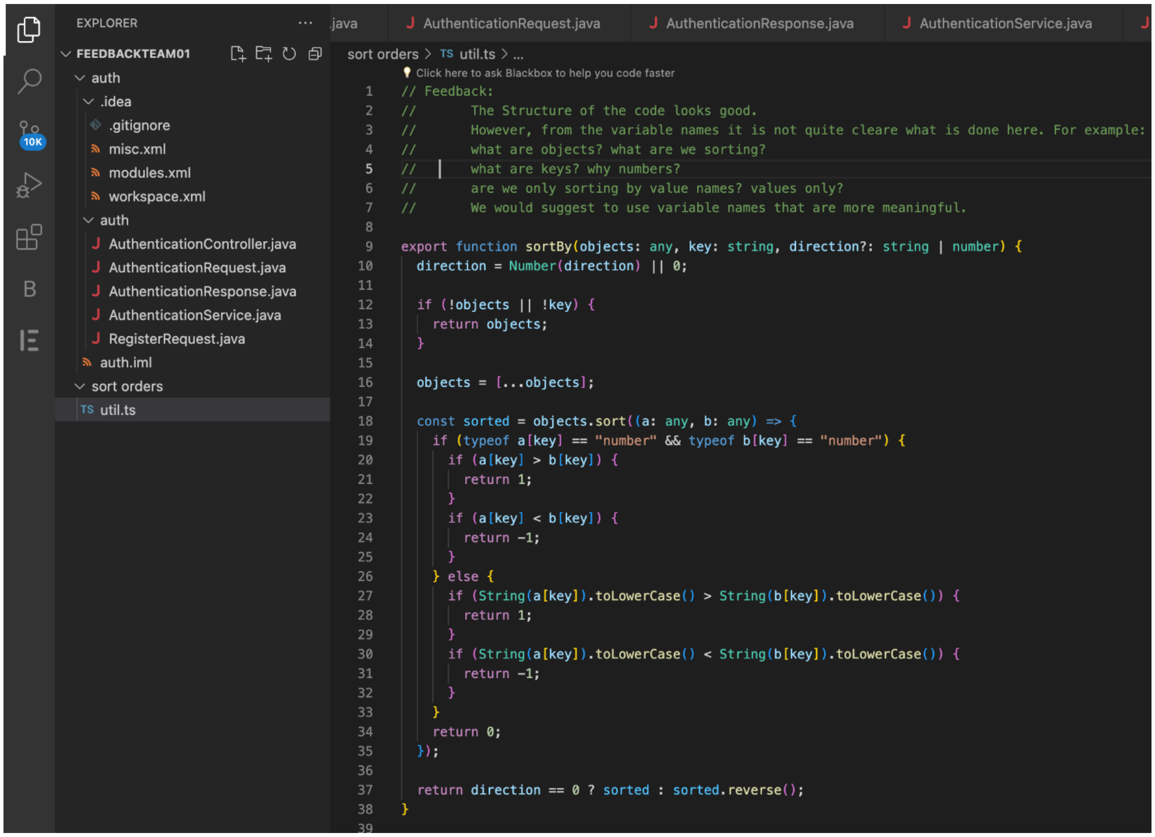
Number #	Test Title	Description	Test Input	Expected Output	Pass/Fail
1	Easy interface	Test Account Navigation and Basic Operations	User logs in to their account , navigate to upload or profile page and have the all basic functionality work	The user should be able to perform all these task smoothie without any guide	pass
2	Check application's in complex situation	Test the usability in completing a complex task	Complex test: register an new business account and upload the tracking number(A1B2C3D4) and then check it at tracking page	The user should be able to complete all these task without confusion	pass

3	Check website functionality in different account	Test the basic website functionality	Test the upload function and see the tracking information in tracking page but do this in guest account, registered personal account and business account	The tester should have a smoothie user experience in all those account	pass

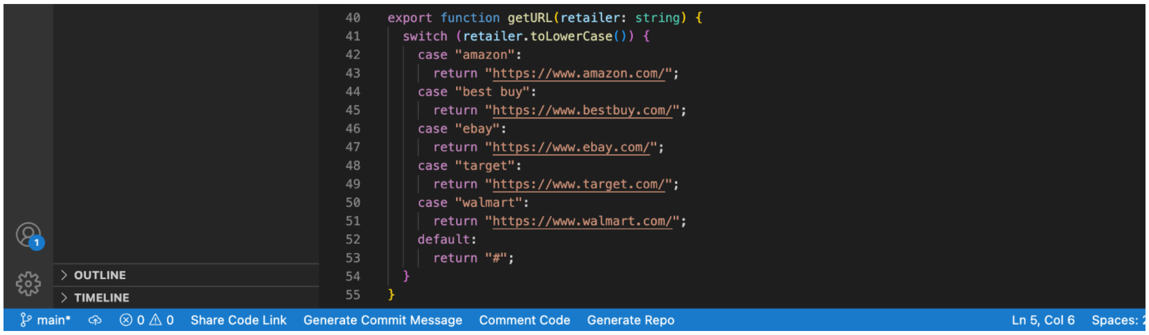
IV. Code Review:

a) The coding style the team shall use is 2 spaces in indentation instead of tabs, each line of code should approximately have 150 characters, and each line of code should be having an empty line between different logical code blocks. Overall, the team should leave else statements and curly brackets without any line breaks as well. This helps ensure the team isn't leaving any newline empty and headers/bodys of functions will be more orgaized to read.

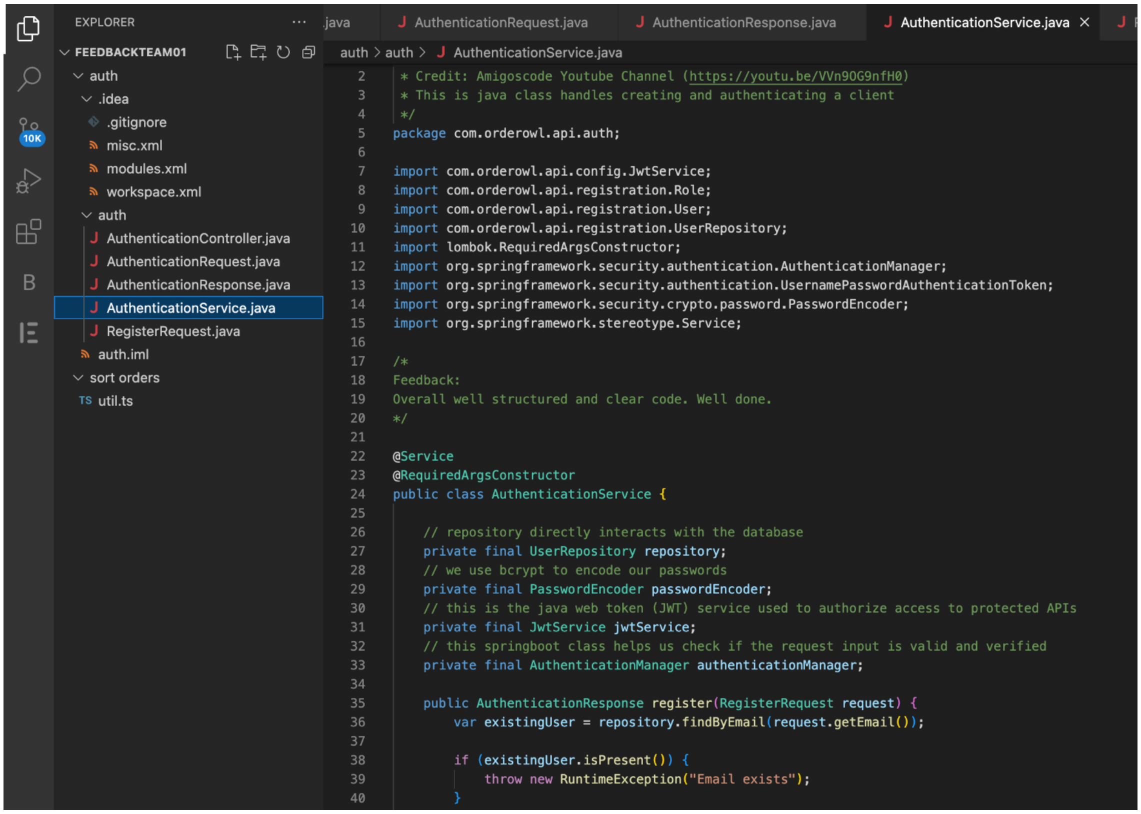
b)


EXPLORER
FEEDBACKTEAM01
auth
.idea
.gitignore
misc.xml
modules.xml
workspace.xml
auth
AuthenticationController.java
AuthenticationRequest.java
AuthenticationResponse.java
AuthenticationService.java
RegisterRequest.java
auth.iml
sort orders
util.ts
10K
B
.java
AuthenticationRequest.java
AuthenticationResponse.java
AuthenticationService.java
sort orders > util.ts > ...
Click here to ask Blackbox to help you code faster

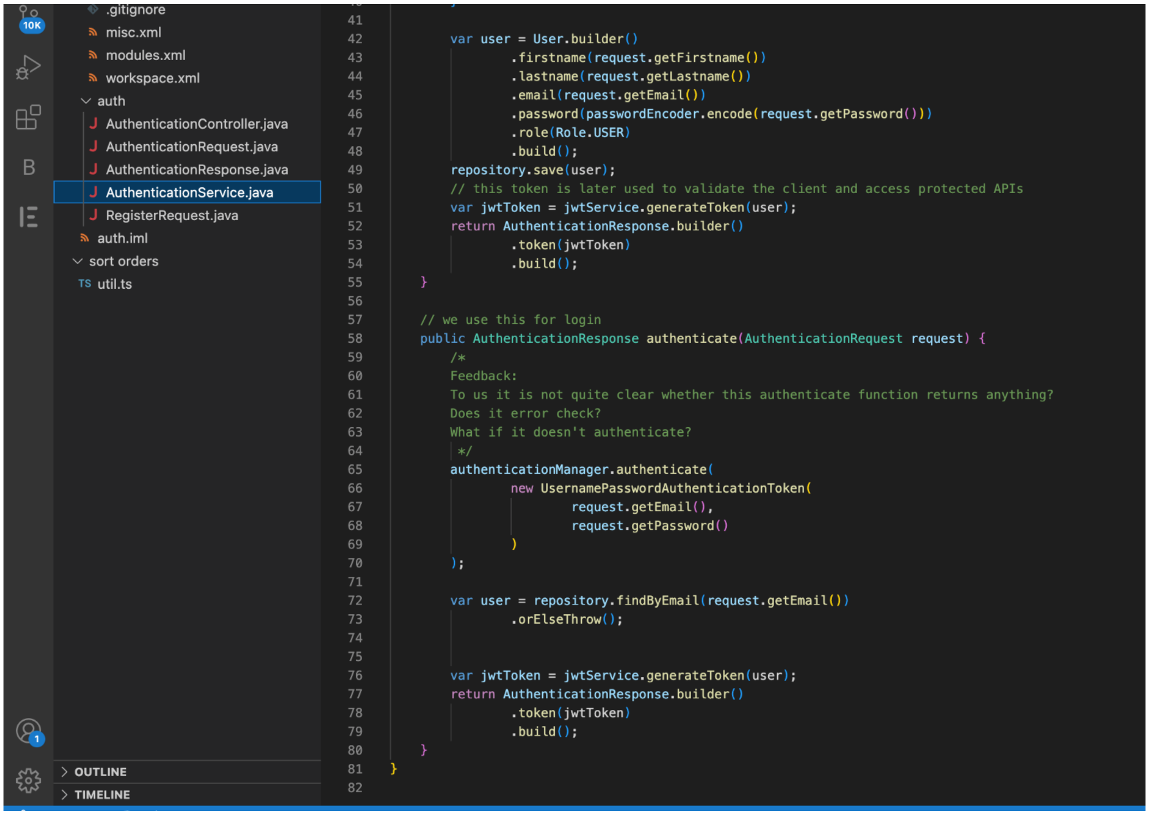
```
// Feedback:
//       The Structure of the code looks good.
//       However, from the variable names it is not quite cleare what is done here. For example:
//       what are objects? what are we sorting?
//       what are keys? why numbers?
//       are we only sorting by value names? values only?
//       We would suggest to use variable names that are more meaningful.

export function sortBy(objects: any, key: string, direction?: string | number) {
  direction = Number(direction) || 0;

  if (!objects || !key) {
    return objects;
  }

  objects = [...objects];

  const sorted = objects.sort((a: any, b: any) => {
    if (typeof a[key] == "number" && typeof b[key] == "number") {
      if (a[key] > b[key]) {
        return 1;
      }
      if (a[key] < b[key]) {
        return -1;
      }
    } else {
      if (String(a[key]).toLowerCase() > String(b[key]).toLowerCase()) {
        return 1;
      }
      if (String(a[key]).toLowerCase() < String(b[key]).toLowerCase()) {
        return -1;
      }
    }
    return 0;
  });

  return direction == 0 ? sorted : sorted.reverse();
}
```



```
export function getURL(retailer: string) {
  switch (retailer.toLowerCase()) {
    case "amazon":
      return "https://www.amazon.com/";
    case "best buy":
      return "https://www.bestbuy.com/";
    case "ebay":
      return "https://www.ebay.com/";
    case "target":
      return "https://www.target.com/";
    case "walmart":
      return "https://www.walmart.com/";
    default:
      return "#";
  }
}
```


EXPLORER
FEEDBACKTEAM01
auth
.idea
.gitignore
misc.xml
modules.xml
workspace.xml
auth
AuthenticationController.java
AuthenticationRequest.java
AuthenticationResponse.java
AuthenticationService.java
RegisterRequest.java
auth.iml
sort orders
util.ts
.java
AuthenticationRequest.java
AuthenticationResponse.java
AuthenticationService.java
auth > auth > AuthenticationService.java

```java
 * Credit: Amigoscode Youtube Channel (https://youtu.be/VVn9OG9nfH0)
 * This is java class handles creating and authenticating a client
 */
package com.orderowl.api.auth;

import com.orderowl.api.config.JwtService;
import com.orderowl.api.registration.Role;
import com.orderowl.api.registration.User;
import com.orderowl.api.registration.UserRepository;
import lombok.RequiredArgsConstructor;
import org.springframework.security.authentication.AuthenticationManager;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.stereotype.Service;

/*
Feedback:
Overall well structured and clear code. Well done.
*/

@Service
@RequiredArgsConstructor
public class AuthenticationService {

    // repository directly interacts with the database
    private final UserRepository repository;
    // we use bcrypt to encode our passwords
    private final PasswordEncoder passwordEncoder;
    // this is the java web token (JWT) service used to authorize access to protected APIs
    private final JwtService jwtService;
    // this springboot class helps us check if the request input is valid and verified
    private final AuthenticationManager authenticationManager;

    public AuthenticationResponse register(RegisterRequest request) {
        var existingUser = repository.findByEmail(request.getEmail());

        if (existingUser.isPresent()) {
            throw new RuntimeException("Email exists");
        }
```

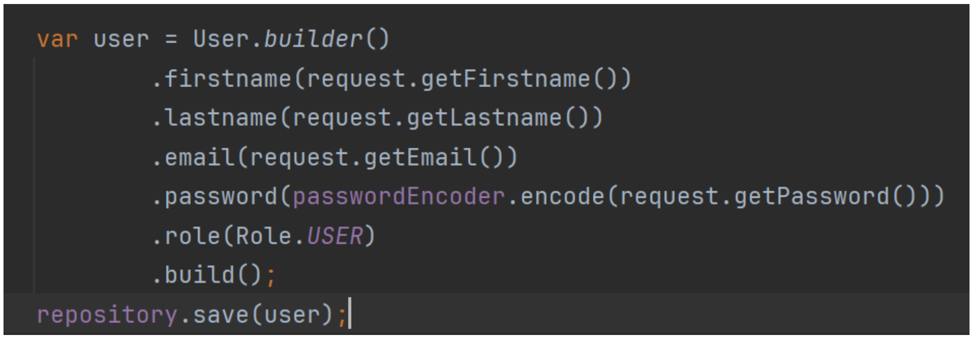

.gitignore
misc.xml
modules.xml
workspace.xml
auth
AuthenticationController.java
AuthenticationRequest.java
AuthenticationResponse.java
AuthenticationService.java
RegisterRequest.java
auth.iml
sort orders
util.ts
OUTLINE
TIMELINE

```java

        var user = User.builder()
                .firstname(request.getFirstname())
                .lastname(request.getLastname())
                .email(request.getEmail())
                .password(passwordEncoder.encode(request.getPassword()))
                .role(Role.USER)
                .build();
        repository.save(user);
        // this token is later used to validate the client and access protected APIs
        var jwtToken = jwtService.generateToken(user);
        return AuthenticationResponse.builder()
                .token(jwtToken)
                .build();
    }

    // we use this for login
    public AuthenticationResponse authenticate(AuthenticationRequest request) {
        /*
        Feedback:
        To us it is not quite clear whether this authenticate function returns anything?
        Does it error check?
        What if it doesn't authenticate?
         */
        authenticationManager.authenticate(
                new UsernamePasswordAuthenticationToken(
                        request.getEmail(),
                        request.getPassword()
                )
        );

        var user = repository.findByEmail(request.getEmail())
                .orElseThrow();


        var jwtToken = jwtService.generateToken(user);
        return AuthenticationResponse.builder()
                .token(jwtToken)
                .build();
    }
}

```

The files we shared with Group 1 for review were AuthenticationService from our backend and Sort Orders from our frontend. We chose these specific files for feedback because of the critical roles they play in our applications functionality and user experience are highly effective. We’d like the feedback to strengthen the security and usability aspects of our application and ensure an efficient user friendly experience for our customers.

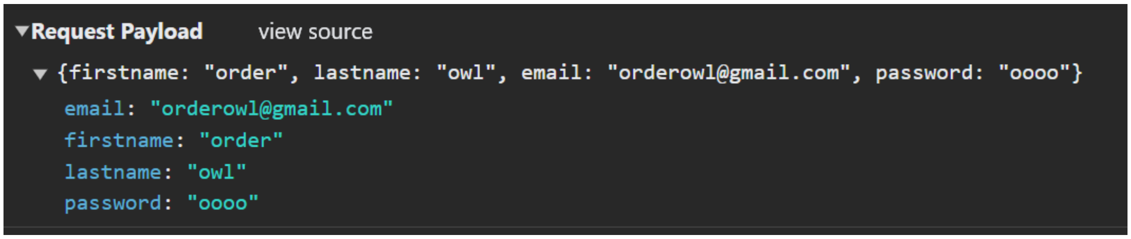
V. Self-check on best practices for security:

1. **List major assets you are protecting**
   a. User Information
      i. Name (first and last)
      ii. Address
      iii. Email
   b. Business Information
      i. Email
      ii. Business License
2. **Confirm that you encrypt PW in the DB (describe the process and show real examples (i.e screenshots……)**
   a. **When we accept the login form from the frontend we will run it through our User builder which will build the user with the password encoder. The encoder is set to encrypt with BCrypt. We then save the built user with the hashed password into the repository.**
   b.



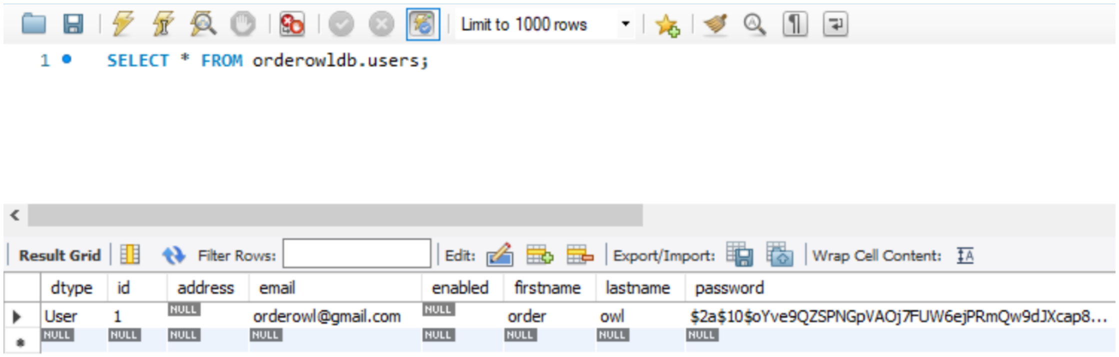
```
var user = User.builder()
        .firstname(request.getFirstname())
        .lastname(request.getLastname())
        .email(request.getEmail())
        .password(passwordEncoder.encode(request.getPassword()))
        .role(Role.USER)
        .build();
repository.save(user);
```

   c.



```
▼Request Payload      view source
  ▼ {firstname: "order", lastname: "owl", email: "orderowl@gmail.com", password: "oooo"}
      email: "orderowl@gmail.com"
      firstname: "order"
      lastname: "owl"
      password: "oooo"
```

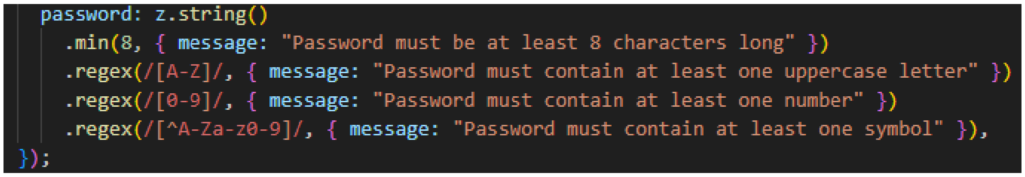
   d.



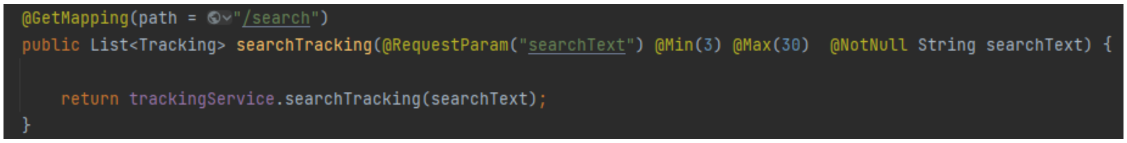
```
1 ● SELECT * FROM orderowldb.users;
```

	dtype	id	address	email	enabled	firstname	lastname	password
▶	User	1	NULL	orderowl@gmail.com	NULL	order	owl	$2a$10$oYve9QZSPNGpVAOj7FUW6ejPRmQw9dJXcap8...
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

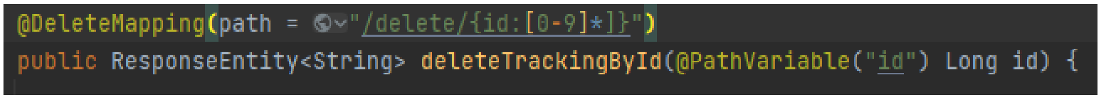
3. **Confirm Input data validation (list what is being validated and what code you used) – we request you validate search bar input;**
    a. **We are validating password, search bar queries and uploading tracking numbers.**

    b. **Password**
        i. 

```
  password: z.string()
    .min(8, { message: "Password must be at least 8 characters long" })
    .regex(/[A-Z]/, { message: "Password must contain at least one uppercase letter" })
    .regex(/[0-9]/, { message: "Password must contain at least one number" })
    .regex(/[^A-Za-z0-9]/, { message: "Password must contain at least one symbol" }),
});
```

    c. **Search Bar query**
        i. 

```
@GetMapping(path = "/search")
public List<Tracking> searchTracking(@RequestParam("searchText") @Min(3) @Max(30)  @NotNull String searchText) {

    return trackingService.searchTracking(searchText);
}
```

        ii. **With the annotations, we require the search to be not null and a minimum of 3 chars and a maximum of 30.**
    d. **Delete Tracking Number**
        i. 

```
@DeleteMapping(path = "/delete/{id:[0-9]*]}")
public ResponseEntity<String> deleteTrackingById(@PathVariable("id") Long id) {
```

        ii. **We require the ID to be a digit.**

VI. Self-check: Adherence to original Non-functional Specs:

Category	Non-Functional Requirement	Status
Performance	1.1 The system needs to smoothly handle a minimum of 1000 concurrent users without slowing down.	DONE
Performance	1.2 Updates to tracking information should happen in real-time, with no more than a 5-second delay from the actual event.	DONE
Expected Load	2.1 The system should be prepared to manage a 50% increase in traffic during peak shopping seasons (like Black Friday and Cyber Monday).	DONE
Expected Load	2.2 Even during high traffic periods, users shouldn't experience response times exceeding 2 seconds	DONE
Security	3.1 Users' personal data shall be safeguarded through stand encryption protocols.	DONE
Security	3.2 Only authorized personnel should access sensitive data.	DONE
Storage	4.1 The system should be capable of storing tracking information for a minimum of 2 years, which is essential for historical reference and auditing.	DONE
Availability	5.1 Users shall be able to access the system at least 90% of the time.	DONE
Usability	7.1 System's user interface	DONE

	that's intuitive and easy to use for both customers and shipping personnel.	
Compliance	8.1 System shall follow all shipping regulations and laws of any place of operation.	DONE
Compliance	8.2 Regular checks and updates will ensure ongoing compliance.	DONE
Logging and Monitoring	9.1 System shall keep comprehensive logs of system activities for auditing and debugging purposes.	ON TRACK
Platform/Browser Compatibility	10.1 Develop the application using web design that shall ensure usability on different platforms including devices	ON TRACK
Platform/Browser Compatibility	10.2 OrderOwl shall be tested on multiple browsers and devices to confirm the compatibility of the application	ON TRACK
Coding Standards	11.1 The team shall use 2 spaces for indentation instead of tabs.	DONE
Coding Standards	11.2 The team shall have 150 max characters per line of code.	ON TRACK
Coding Standards	11.3 The team shall have an empty line between different logical code blocks.	DONE
Coding Standards	11.4 The team shall leave the else statement and curly brace without a line break and its body will be on the next line.	DONE
Coding Standards	11.5 The team shall not leave an empty new line between	DONE

	the function header and body.	

VII. Detailed List of Contributions:

Team Lead:

Team Members	Contribution	Score
Belu Velazco	● Worked on the front-end side of the login and registration page by adding ● Led all team meetings ● Set deadlines and did every other day check-ins halfway through the milestone ● Organized threads within discussion channel ● Wrote up meeting summaries/reports ● Hosted some of the workshop calls ● Came to every in-class meeting ● Worked on product summary; section 1 ● Worked on section 6 and 7 for documentation ● Gave feedback and did code review for Team 1's code. ● Consistent participation and communication ● Redesigned Upload page and added a choose carrier drop down menu that fit Shippo's needs (later removed due to switching to EasyPost) ● Reached out to pair up with a team for section 4: code review ○ Sent out email with the code we wanted to get feedback on	8

	○ Sent out email with the feedback from team 6 to team 1’s code. ● Created complaint/feedback template for team ● Sent email for extension ● Created TOS Agreements page ● Created buttons for hidden and visible list ● Created Hidden order list page ●	
David Lien	● Worked on getting USPS API permission ● Set up EasyPost ● Tested tracking using real tracking number ● Was ready to lead a team meeting when team lead was ill ● Worked on authentication and authorization ● Started some workshop calls ● Joined and participated in every team meeting ● Came to every in-class meeting ● Worked on section 5 for documentation ● Consistent participation and communication ● Worked on code for users to edit pre-existing profile information. ○ Work in progress ● Added order titles to database ● Made it so users could choose personal or	10

	business; back-end ● Deployed server ● Implemented EasyPost API to track real orders. ● Did testing for section 2 and 3 ● Helped teammates	
Komaldeep Kaur	● Worked on product summary; section 1 ● Joined and participated in every team meeting ● Came to every in-class meeting ● Came up with ideas for business side of our application ● Worked on changing the header for our homepage regarding feedback from our M3 prototype → added 3 separate buttons for guest login, user login, and registration ● Participated and communicated during team meetings ● Worked on section 6 of M4V1 documentation ● Added password requirements checker to registration page ● Volunteered to take up tasks ● Formatted M4V1 document ● Fixed wording for section 2 summary ● Checked coding style for code files. ● Took charge of section 6 of M4V1.	8
Jimmy Pan	● Worked on sorting orders; front-end	8

	● Joined and participated in every team meeting ● Came to every in-class meeting ● Created page for personal user profile ● Worked on calendar feature to update with estimated time arrivals ● Worked on section 2 for documentation ● Gave feedback and did code review for Team 1’s code. ● Participated during team meetings ● Fixed google maps so they appear ● Fixed Login page ● Helped other front-end team members with questions ● Fixed TOS checkbox for validation ● Fixed alignment on images for login and registration page.	
Mankit Yeung	● Added images to homepage; front-end ● Joined and participated in every team meeting ● Came to every in-class meeting ● Worked on section 3 for documentation ● Worked on adding order titles into our tracking page and upload page so users can upload a title for their tracked orders. ● Gave feedback and did code review for Team 1’s code. ● Gave feedback and did	9

	code review for our code that we sent to Team 1. ● Consistent participation and communication ● Fixed redirection of the registration button on the homepage ● Updated section 3 of Milestone 3 v2 ● Worked on pop-ups for guests and entering a pin to access hidden orders.	
Luis Ramirez	● Worked on hidden orders; back-end ● Created pin system for users to access their hidden orders list ● Discussed hidden ● Started some workshop calls ● Joined and participated in every team meeting ● Came to every in-class meeting ● Worked on section 2 ● Checked how our application looked on Microsoft Edge, Google Chrome, Opera, Firefox, and Safari. ● Worked on making some pages responsive to other devices ● Consistent participation and communication	10

Tin Nguyen	● Joined and participated in every team meeting ● Came to every in-class meeting ● Worked on back-end for the business and personal accounts unique feature ● Worked on implementing Google Maps ● Fixed search query to make sure most relevant popped up ● Worked on section 3 and 6 for documentation ● Participated and communicated during team meetings ● Added Google maps API	7