

# Test Plan and Report

Datum

12/03/24

## System Test Scenarios

- User Story 1: As an employee, I need to be able to read and download files from a database in order to access materials added by other users and teams.
- Scenario 1: Download authorized files (Pass/Fail)
  1. Enter department page
  2. Wait for files to load from firebase
  3. Click on the file you want to access in the list. The PDF should open in a new tab
- User Story 2: As a merchandiser at a supply chain intermediary, I want to access client and supplier information readily, and easily map buyers to suppliers.
- Scenario 2: View information about buyers/suppliers (Pass/Fail)
  1. As an employee with access to merchandising enter the department page
  2. Wait for the page to load fully
  3. On the bottom of the page you should see all buyers and suppliers associated with the company
  4. Click on the relevant company. You should see their contact information & uploaded files
- User Story 3: As an employee, I want to be able to access the knowledge base from the web. This will allow me to view the company's files from home or work or anywhere in the world.
- Scenario 3: Access the webpage & be redirected to your department (Pass/Fail)
  1. Enter the webpage site in internet browser
  2. If logged in the user will be redirected to their department. If not, the user will be sent to the signin page.
  3. After signing in the user should be redirected to their department
  4. The user can access the knowledge base.
- User Story 4: As an employee, I want to access relevant information and clarify my understanding about certain documents in the knowledge base.
- Scenario 4: Interaction with the AI features (Pass/Fail)
  1. Have access to your department page
  2. Once the page is loaded the user can click on the AI button in the bottom indicated by the Ollama icon.
  3. The user should be able to use the summarize or chat feature as long as the company has set up their open source LLM.
  4. In summary, the user should be able to search the files in the department & summarize with AI one at a time.
  5. In chat, the user should be able to have a conversation freely with the LLM about the knowledge base.
- User Story 5: As an employee, I want to be able to quickly share documents to other departments to view
- Scenario 5: Be able to send over and share documents through a folder, which all departments can access
  1. Login to department account
  2. For files you own, press "share"
  3. Share with specific department of choice through drop down menu
- User Story 6: As an administrator, I want to have management access to all files in the database, in order to manage files created by multiple employees in my team.
- Scenario 6: Have free access to the entire website (Pass/Fail):
  1. Be logged in to admin account

2. Enter the home page
3. Access department pages through home page (Merchandising, Logistics, QA, HR)
4. Upload files through the “upload” button to see if they appear in file list
5. Delete files through “delete” toggle regardless of ownership

## **Unit Tests**

### **app/workplaces/page.test.tsx**

#### **Test Cases:**

1. Rendering a loading indicator while fetching companies.
2. Displaying fetched companies in a dropdown menu.
3. Redirecting to the sign-in page with the selected company ID when a company is selected.
4. Handling errors during company fetching without crashing.
5. Automatically redirecting authenticated users to their department URL.

### **app/Utilities/Upload/uploadFiles.test.ts**

#### **Test Cases:**

1. Uploading a file and updating Firestore with the correct download URL.
2. Throwing an error if no file is provided.
3. Handling file replacement confirmation and cancellation.
4. Updating Firestore for different collection types (e.g., Departments, Manufacturers).
5. Throwing an error if an invalid collection type is provided.

### **app/Utilities/Upload/deleteFiles.test.ts**

#### **Test Cases:**

1. Deletes files from Firestore and Firebase storage.
2. Deletes files from different types of collections.
3. Error handling for invalid firestore paths.
4. Handles errors deleting files from Firebase storage or Firestore.

### **app/api/resetPass/route.test.tsx**

#### **Test Cases:**

1. Returning a 405 status code for non-POST requests.
2. Returning a 200 status code and sending a reset email on valid input.
3. Returning a 500 status code if there is an error generating the reset link.

## **app/api/createUser/route.test.tsx**

### **Test Cases:**

1. Denying access when the Authorization header is missing.
2. Denying access when the user is not an admin.
3. Successfully creating a user and returning a 201 status code.
4. Returning a 500 status code if an error occurs during user creation.

## **app/authentication.test.js**

### **Test Cases:**

1. Getting departments and companies.
2. Signing in a user with email and password.
3. Sending a verification code to a phone number.
4. Logging out a user.
5. Retrieving an employee's profile data.

## **app/profile/page.test.tsx**

### **Test Cases:**

1. Rendering the loading state initially.
2. Redirecting unauthenticated users to the login page.
3. Displaying an error message if fetching profile data fails.
4. Displaying employee data and authorized departments.
5. Handling password reset functionality.
6. Sending and verifying a phone number update via verification code.
7. Handling sign-out functionality.

## **app/createUser/page.test.tsx**

### **Test Cases:**

1. Rendering form fields for email, name, phone number, role, and departments.
2. Loading departments and populating the department dropdown.
3. Displaying an error message if no departments are available.
4. Handling successful account creation.
5. Handling errors during account creation.

## **app/signin/page.test.tsx**

### **Test Cases:**

1. Rendering the sign-in form.
2. Handling user sign-in with email and password.
3. Handling errors during sign-in (e.g., invalid credentials).
4. Redirecting authenticated users appropriately.
5. Resetting a user's password.

## **app/page.test.tsx**

### **Test Cases:**

1. Redirecting authenticated users to their department URL.
2. Rendering the home page correctly for unauthenticated users.

## **app/faq/page.test.tsx**

### **Test Cases:**

1. Rendering the FAQ page with dropdown menus and a search button.
2. Displaying an error message when no search criteria are selected.
3. Handling the FAQ search functionality.

## **app/departments/logistics/page.test.tsx**

### **Test Cases:**

1. Rendering the Logistics department page with all its components.
2. Handling file selection in the FileList component.
3. Handling changes in the selected collection.
4. Handling successful and failed file uploads.
5. Refreshing the file list upon successful upload.
6. Interacting with the AI Button functionality.

## **app/departments/qa/fileList.test.tsx**

### **Test Cases:**

1. Rendering the loading state.
2. Displaying an error message when loading fails.
3. Showing a message when no files are available.
4. Rendering a list of files when data is fetched successfully.

## **app/home/page.test.tsx**

## Test Cases:

1. Rendering department links (e.g., Quality Assurance, Human Resources).
2. Navigating to different department pages.
3. Handling the display of the "Create Employee" option.
4. Ensuring proper authentication state management.

Anyone who wants to test Datum themselves specifically creating users and editing user phone and email you will need two more files that are intentionally not on our GitHub.

The first is our .env file:

```
EMAIL_USER=ucscdatum@gmail.com

EMAIL_PASS=enqf eyho epkv xtld

SMTP_HOST=smtp.gmail.com

SMTP_PORT=587

FIREBASE_ADMIN_JSON=./datum-115a-firebase-adminsdk-8jq7s-5464beb7be.json
```

The second is our datum-115a-firebase-adminsdk-8jq7s-5464beb7be.json file:

```
{
  "type": "service_account",
  "project_id": "datum-115a",
  "private_key_id": "5464beb7bec474287c3a4f601937dd7a4150c1d2",
  "private_key": "-----BEGIN PRIVATE
KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBCwggSjAgEAAoIBAQCzam2vmJWeSPA5
\ns8ah3G5ihmAdggEOx17AMTI9FCJqEc1mxjWXNYk5ku02sgNqlmWaB5HpJk3sd2sn\nnm+2KvQ
VIyxmUs7n94zwZBSaJmbne5xUtV70QqruimsaMWCM1lMFvLoRWzKlgNlPg\nnsYn7I9WOMWq7ym
A3HfQKZQ2fmKFoxhTpNpfIW13yyEbJ35kiLxHlutJuhetRgEjK\nYuvfCjF+0298rTGIFrnrDE
OpmbdbgWbeuz5r2B15fw8VnPkQRKmoNazAEFKJ31FZ\nn08wSQlHaeoXvEeHvrgmOMTuZt0IjOM
iMXfqXWDosHXC7P1zhkrIqafU1b815rph\nnw3YmnNP/AgMBAAECggEARAetA5I12VjqzdZfEO
zWZkFxnH65C/Mn5lRBKmvCCNor\nOjcaBvXjzJLro3uoU/gnGlqYdwFxnqKpXOGBFsRJhqqMTp
```

```
g/hAdvuET9sds6YUZZ\n+NF1/SXf88YTSc/5s7xOBGxs/cSozLSM5XwsR6/0v+/4+q/n/7451N
8wZT2opmMi\n6Y02vjfzoKBGZJlmzLKbqupnB4VYZdK00FyoTJWiyv7v16lG811O8783AaurqQ
/b\n1RtMpR0KLdtq6IXEW97/0xD3GD9E6iqeB680D8L7IqjJUhcZG5V9PcdtTm+aqt8\n9ybn
t3mH+Ct6LNLwP2EAYLKdRUTOVABknP0RrOcIwQKBgQDaBao0aoUlpWZoUNEz\njraGInZu9h68
mIvE5AHY/S7ddIKShW0VhxiWh/252nZNQtiFg32TRdl1OSnnr3Ek\n1Hz9L4f0E+QiB1BMQbk1
IeP71rzRpXuEMJGwMq/cfkIb5Bz0n6FS18M6nc4XaMjA\nwvphkAjaJckilzU95fCY/bFSzwKB
gQDSqyzy+smsWE2gMyExh8Dm8E5R7/uMzqOu\nP/li+fFr+aZ8WWrwO9+Zvm6gN307Yp9vQ6NZ
mFPjAyTfKdY7hfBjvWy8Y6orLfDF\nYnhhQa4lKskjs+rwaEer5H3/mx+Qzd9HDRtHnFeVqAdq
6XDEu2IROeqVef6AYcF9\nK4OYTYZ30QKBgEGn4xHV4je9j5FqhRWZrK3IQxI3XBo/dhfc8hF+
bVaTd57YP3l0\nLZtnaKlHeMjBkIjkiDSJnGY3SL9UviXxN5KndbGzK8hzm2LbiALQKe/lQvd/
dirg\n1XccsQIdo5rynyWrzNEqkshcLWwvSgWC1QSwc3Ip7uBnHsl9LW9lNrlRAoGAXXBn\na2
6QvMx9waxr8JgRylySoG0c2Ztdpg/jx16s4gNdAJJ4iA1GY96/GnkCJrDbnnIV\nnpcuWiX3/gK
wBxWzm7pY6AGUEORqrTCEc26ZSI1IS3K8UaYUXJPq3mAOpwIRPoCgE\nnDYMepdZdU+Iujlo+b3
2MurfdVfwF4grG12qrVcECgYEA0zHw8VZ4YF0/OLKOq9MP\nngPYENSiNKSQRdRRDuHEdC5Rckc
RY0kd0hhMMRyN2e8YbWsibreZ1S0hmkpIodDzG\nvRnFyYjYO4dEIEf+YVhvbsiidH3lo3ELlh+
hIJaiprbyXrNvLRdGyCuk1YYNg2ptF\nv1qBos38Fp6I7Oye4hG0I/c=\n-----END PRIVATE
KEY-----\n",
```

```

    "client_email":
"firebase-adminsdk-8jq7s@datum-115a.iam.gserviceaccount.com",

    "client_id": "103784606396383201348",

    "auth_uri": "https://accounts.google.com/o/oauth2/auth",

    "token_uri": "https://oauth2.googleapis.com/token",

    "auth_provider_x509_cert_url":
"https://www.googleapis.com/oauth2/v1/certs",

    "client_x509_cert_url":
"https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-8jq7s
%40datum-115a.iam.gserviceaccount.com",

    "universe_domain": "googleapis.com"
}
```

You would have to put this code in root directory under Datum, ( NOT INSIDE APP )

**app/home/page.test.tsx**

### Test Cases:

Genkit CLI: Used Genkit CLI to test different flows (functions) for indexing, retrieving, and generating. Used to track if flows were working correctly with traces (history). Once verified, the flows were implemented for the chatbot.

