

Contents

1	Overview	2
1.1	Purpose	2
1.2	Business Value	2
1.3	Key Features	2
2	User Guide	2
2.1	Viewing and Searching Users	2
2.2	Creating a New User	2
2.3	Updating Existing Users	3
2.4	Error Handling	3
2.5	Common Use Cases	3
2.5.1	Onboarding a New Employee	3
2.5.2	Updating User Department During Reorganization	3
2.5.3	Deactivating Users Who Left the Company	3
2.6	Tips and Best Practices	3
3	Technical Details & Known Limitations	3
3.1	Technical Overview	3
3.2	Known Limitations	4
3.3	Performance Considerations	4
3.4	System Architecture Overview	4
3.5	Technical References / Screenshots	4

1 Overview

1.1 Purpose

The primary goal of this system is to simplify user management operations to the utmost degree. In this way, user management can be handled by anyone without requiring any technical knowledge.

1.2 Business Value

Thanks to the system's simple and clear design, even staff without technical knowledge can manage users easily. Operations such as adding, deleting, and updating users can be performed more efficiently. Moreover, by keeping records of user creation and updates, the system prevents issues related to dates. Importantly, all of this can be done without requiring additional resources.

1.3 Key Features

1. **User List View with Search & Filter:** Users can be searched by name or email and filtered by role or department. Search and filters can be combined. (*See Appendix A, Figure 1*)
2. **User Creation with Validation:** Only Admin users can create new users. Required fields: Name, Email, Password, Department, Role, Status. Successful creation shows a confirmation message. (*See Appendix A, Figure 2*)
3. **Real-time User Updates:** Admin and Manager users can update user details. All changes are immediately reflected in the Firestore database with timestamps. (*See Appendix A, Figure 3*)
4. **Audit Logging:** All changes are logged with creation and update timestamps. (*See Appendix A, Figure 4*)
5. **CSV Export:** Only Admin users can export the user list to CSV format. (*See Appendix A, Figure 5*)
6. **Error Handling & User Feedback:** Duplicate emails are not allowed. Only hot-mail.com and gmail.com domains are permitted. Clear error messages are displayed to the user in case of invalid input or failed operations. (*See Appendix A, Figure 6*)

2 User Guide

2.1 Viewing and Searching Users

1. After login, the main user table is displayed.
2. Use the search bar to find users by name or email.
3. Use the dropdown menus to filter by role or department.
4. Filters can be combined with search. (*See Appendix A, Figure 1*)

2.2 Creating a New User

1. Click the "Create New User" button; a modal form appears.
2. Fill in required fields: Name, Email, Password, Department, Role, Status.
3. Click "Save" or "Create User" to submit. Confirmation is displayed. (*See Appendix A, Figure 2*)

2.3 Updating Existing Users

1. Find the user in the list and click the “Edit” button.
2. Update necessary fields (role, department, status, phone number).
3. Click “Save Changes”; updates appear in Firestore immediately. (*See Appendix A, Figure 3*)

2.4 Error Handling

- Duplicate emails are not allowed.
- Only hotmail.com and gmail.com domains are permitted.
- Clear error messages are displayed when an operation fails. (*See Appendix A, Figure 6*)

2.5 Common Use Cases

2.5.1 Onboarding a New Employee

1. Click “Create New User.”
2. Fill in the employee’s name, corporate email, assigned role, and department. Set status to “Active.”
3. Click “Create User.”

2.5.2 Updating User Department During Reorganization

1. Search or filter to find the employee(s) whose department needs updating.
2. Click “Edit” next to their name.
3. Change the “Department” field to the new department.
4. Click “Save Changes.”

2.5.3 Deactivating Users Who Left the Company

1. Search for the employee who has left.
2. Click “Edit” next to their name.
3. Change the “Status” field from “Active” to “Inactive.”
4. Click “Save Changes.”

2.6 Tips and Best Practices

- **Regularly Check Audit Logs:** Review audit logs to track changes.
- **Use Filters Effectively:** Combine role and department filters for fast searching.

3 Technical Details & Known Limitations

3.1 Technical Overview

- **Python Django Backend:** Provides REST APIs, handles business logic, authentication, and validation.
- **Firestore Database:** Stores users and audit logs in NoSQL format. All changes are reflected in real-time with timestamps.
- **Retool:** Low-code platform used to create the user interface and connect to Firestore.
- **Audit Logging:** Captures created and updated timestamps for every action, providing full transparency.

3.2 Known Limitations

- The backend structure and Firestore database synchronize profile pictures and thumbnails; free Google Cloud Storage (GCS) version has restrictions.
- Paid GCS version caused errors due to payment issues.
- Only hotmail.com and gmail.com domains are permitted for new users.

3.3 Performance Considerations

- Large user lists may occasionally cause slower performance due to Firestore query limitations and network latency.
- Efficient filtering, search, and pagination help mitigate these performance issues.

3.4 System Architecture Overview

The system follows a simple client-server model:

- **Frontend:** Retool dashboard provides a user-friendly interface.
- **Backend:** Firestore handles all data storage and real-time updates.
- **Profile Management:** Images (profile pictures and thumbnails) are stored and synchronized, but GCS limitations apply.

3.5 Technical References / Screenshots

- Screenshots of Firestore structure, audit logs, and other technical references. (*See Appendix A, Figures 1-6*)

Appendix A: Screenshots / Figures

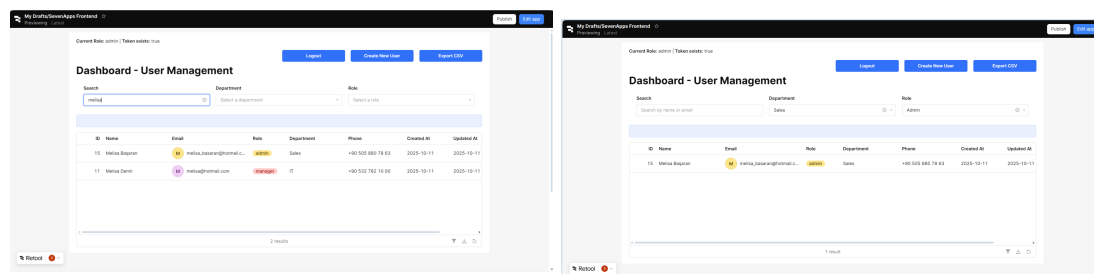


Figure 1: User List View with Search & Filter

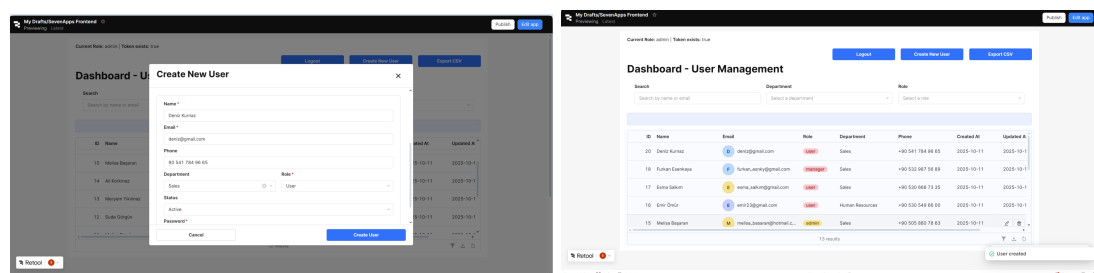


Figure 2: User Creation Form with Validation

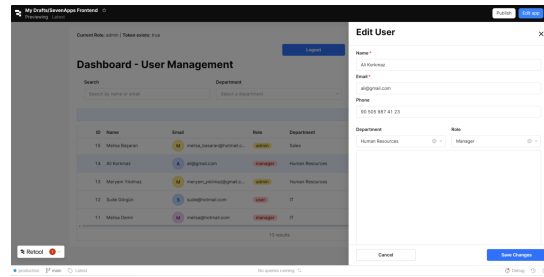


Figure 3: User Edit Modal/Form

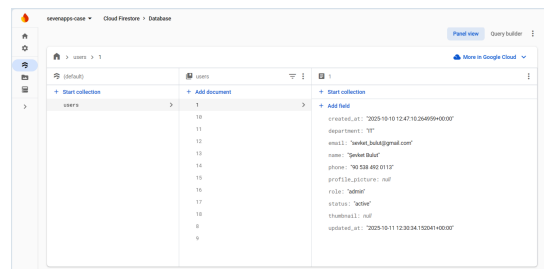


Figure 4: Audit Logging Example

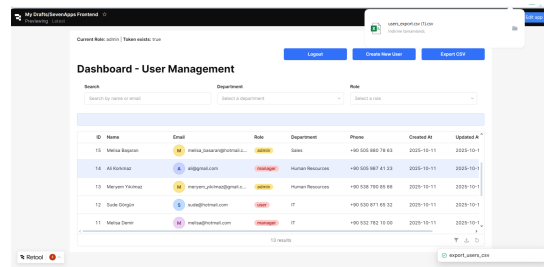


Figure 5: CSV Export Demonstration

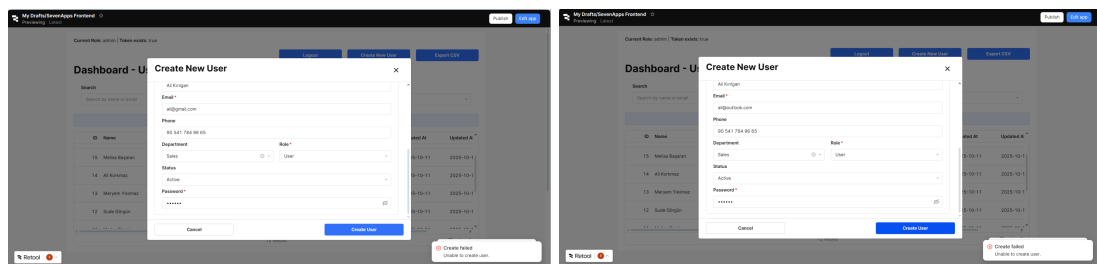


Figure 6: Error Handling Example