

# Employee Onboarding and IT Access Management Portal

## Project Documentation

Horacio Andrade, Amber Garcia, Sebastien Guillen, Ana Gutierrez, Jerry Symbleme

Knight Foundation School of Computing and Information Sciences (KFSCIS) — Florida International University

Instructor: Masoud Sadjadi | Spring 2026 | License: MIT

### 1. Executive Summary

The Employee Onboarding and IT Access Management Portal is a desktop application designed to streamline and track the employee onboarding process across HR, Management, and IT departments. Organizations often struggle with the timely passage of information between the different departments when onboarding a new employee. There is no defined onboarding progress tracker, leaving certain onboarding tasks to be delayed or forgotten. This becomes an issue when the new employee must begin their role and does not have proper access. Our solution facilitates the transfer of onboarding information between all departments involved via onboarding forms, manager review forms, IT provisioning forms, and an onboarding status tracker. By hosting the information in one application, it is easier for the responsible departments to review onboarding progress and complete the necessary onboarding tasks. The application improves efficiency of employee onboarding, ensuring that new hires get started in their role faster.

### 2. Problem & Requirements (O1)

#### A. Context

During the onboarding process, organizations often experience delays and miscommunications between HR, Managers, and IT. Tasks like laptop provisioning, account set-up, and approvals are not tracked in one system, leading to incomplete onboarding tasks, delayed system access, and an overall poor new-hire experience.

#### B. Stakeholders

- HR staff create onboarding requests for new employees and review the onboarding status tracker.
- Managers approve or deny onboarding requests from HR.
- IT staff provision tasks for approved onboarding requests from managers.
- New-hire employees experience the benefits of this portal.

#### C. Personas

- HR staff are the primary users and need to create onboarding requests and track their progress.
- The manager needs to approve or deny the onboarding request.
- The IT technician needs a clear list of provisioning tasks to complete them efficiently.

#### D. Functional Requirements

- HR can create onboarding requests.
- Managers can approve and reject onboarding requests.

- IT can view approved requests.
- IT can submit provisioning updates.
- System tracks onboarding status.
- Role-based access control (HR, Manager, IT).

#### **E. Non-functional Requirements**

- Desktop-based application in a Windows environment
- PostgreSQL relational integrity
- Response time less than 2 seconds for standard queries
- Data persistence and backup
- Version-controlled via GitHub
- Maintainable layered architecture

#### **F. Constraints**

- Built using Microsoft Windows Forms
- PostgreSQL database
- Version control through GitHub
- Desktop-only deployment (no web/cloud hosting)
- Academic project timeline

#### **G. Success Criteria**

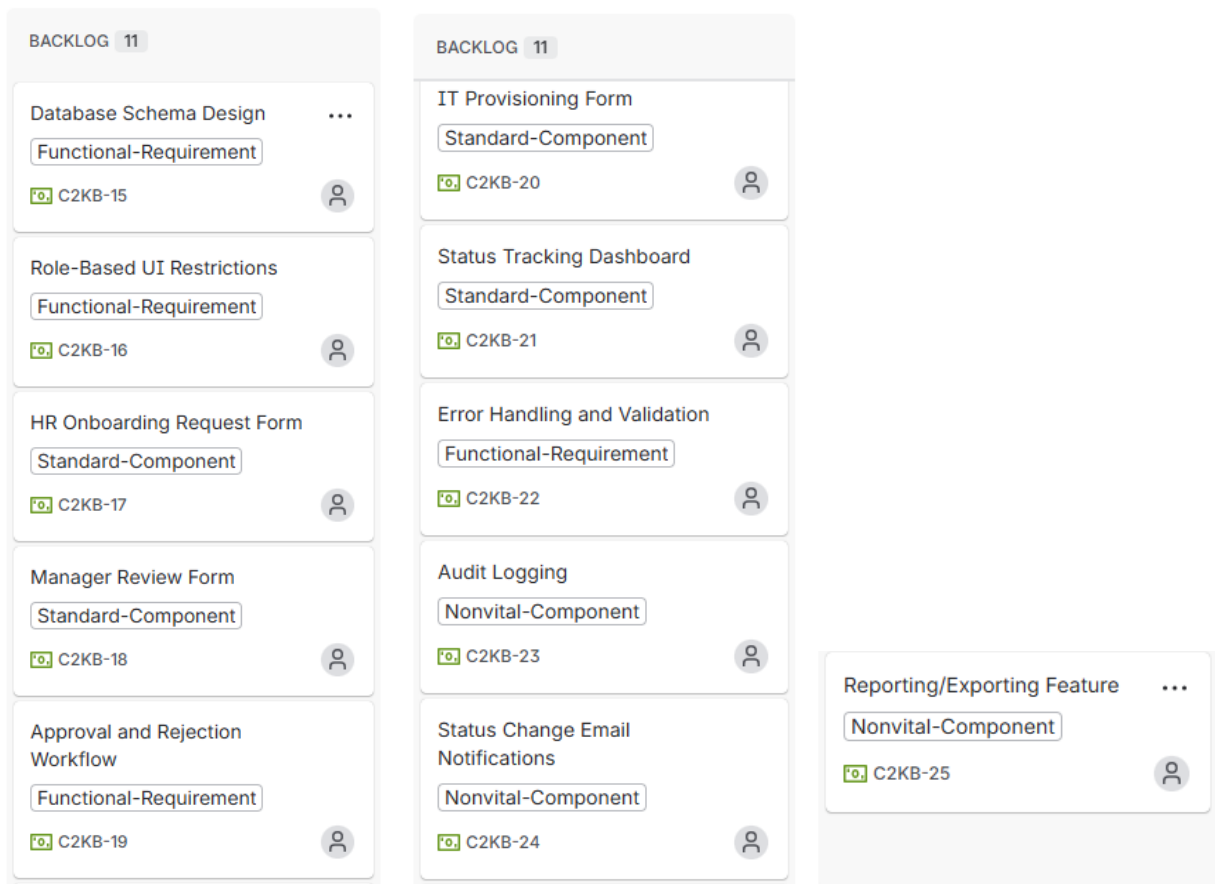
- 100% of onboarding actions are tracked in the system
- No missing provisioning tasks
- Managers approve within system
- IT can complete provisioning without an external email

#### **H. Risks**

- Database connection issues
- Role-based security issues
- Merge conflicts in GitHub
- Informal scope changes

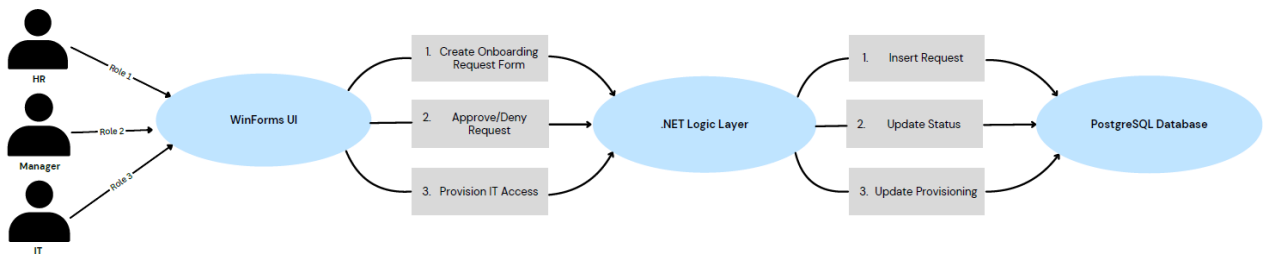
#### **I. Backlog Items**

- Database Schema design
- Role-based UI restrictions
- HR Onboarding Request form
- Manager Review form
- Approval and rejection workflow
- IT Provisioning form
- Status tracking dashboard
- Error handling and validation
- Audit logging
- Status change email notification
- Reporting/exporting feature



### 3. System Overview & Architecture (O2)

#### A. System Architecture Diagram



#### B. High-Level Component Overview

- The presentation layer is made with WinForms, allowing role-based access and user interfaces and makes forms for HR, Managers, and IT.
- The business logic layer is responsible for the authentication, request, approval, and provisioning services.
- The data access layer connects to the PostgreSQL database, which holds SQL queries and parameterized commands.

#### C. Technology Stack

- Microsoft Windows Forms (WinForms)
- C# (.NET)

- PostgreSQL

#### **D. API Overview**

This is a desktop application, meaning there is no REST API because the application directly interacts with the database.

#### **E. Database Overview**

The database supports the employee onboarding workflow by allowing HR to submit onboarding requests for new hires, managers to oversee the process, and IT staff to complete required setup tasks. All database objects are organized under the onboarding schema to keep the project structure clean and adjustable. The database is composed of core tables, lookup/status tables, and seed data scripts used to populate the system with initial testing data.

##### **Core Tables**

- Roles: Stores the different roles that exist in the system.
- Users: Stores system user accounts who interact with the onboarding system.
- Employees: Stores information about employees who are being onboarded.
- Onboarding Requests: Represents a request to onboard a new employee.
- IT Tasks: Stores tasks that IT must complete to prepare the employee's systems and access.

##### **Status Tables**

- Onboarding Request Statuses: Defines the possible status states for onboarding requests.
- IT Task Statuses: Defines the possible status states for IT tasks.

##### **Database Performance Indices**

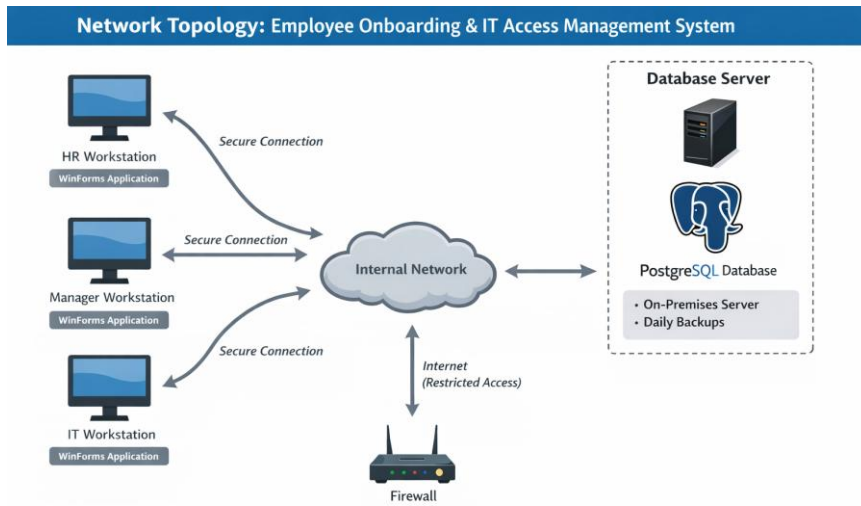
- Indexes were added to improve the speed of database queries for frequently used relationships in the system.
- These indexes help the database quickly locate information, making joins and filtering operations faster during onboarding workflow queries.

##### **Seed Data Scripts**

- The database includes several seed scripts used to populate test data for demonstration and development.
- Users.sql: Creates sample system users and roles including HR user, Manager user, IT user.
- Sample\_Employee.sql: Creates an initial employee record used to test onboarding workflows.
- More\_Sample\_Employees.sql: Adds additional employee records to simulate real onboarding scenarios across multiple departments.
- Onboarding\_Request.sql: Creates an onboarding request linked to the most recently inserted employee.
- IT\_Tasks.sql: Creates example IT tasks linked to the latest onboarding request.

## **4. Discipline-Specific Depth (O6)**

### **A. Architecture & Integration**



- N-Tier / Layered Architecture is utilized to structure code in logical layers—presentation, business, and data-access.
- The portal is built as a Windows Forms desktop application that allows users to interact with the onboarding system through a graphical interface.
- The business logic handles the processing of requests such as updating statuses, assigning tasks, and retrieving information from the database.
- The program connects to a database where all the data is stored, including employees, onboarding requests, and IT tasks.
- All parts of the system work together so that the user interface, business logic, and database support the onboarding workflow.

## B. DevOps & SRE

### Version Control Strategy

Project code is stored on GitHub allowing team members to work collaboratively and keep track of changes made to the project.

- Main branch remains stable
- Branches for foundational setup and app features
- Pull requests required for merging.
- Code reviews and team updates required before merge.

### CI/CD

Continuous testing and logging helps observe how the system behaves and identify and eliminate bugs.

- Integrated GitHub CI/CD status badge to provide real-time indicator if GitHub actions workflows are succeeding or failing.

### IaC Snippets

SQL codes are used to manually create the database structure and insert sample data so the system can be set up properly.

### Observability

- Basic error handling
- Platform logging to monitor application health

## **SLOs/SLIs**

- 99% uptime during business hours over a 30-day period.
- Form requests update next department in the workflow in less than 30 seconds.
- Less than 2 second response time for database queries.

## **Error Budget Policy**

- 1% uptime error budget
- With 50% budget consumed, developers will receive an alert that must be assessed before completion of other non-essential tasks.
- With 100% budget consumed, developers must halt all additional feature development tasks to enhance application reliability.

## **C. Operations & Security**

### **Access Management**

- Uses role based access control so different users only access features related to their role (HR, manager, or IT).

### **Backup/Restore**

Database backups help protect important data in case of unexpected errors within the system.

- Daily PostgreSQL dump
- Weekly full database backup
- Physical backup storage options

### **Change Management**

In the event of a requested change to the scope of the project, this process will be carried out between team members.

1. Change request is proposed to the team.
2. The team assesses the change's impact on the project timeline and quality of the project itself.
3. Team votes for implementation or rejection of the change.
4. If approved, change is added to the backlog and integrated in the sprint breakdown/user story planning based on its discussed priority.

### **Configuration Hardening**

Security measures help protect the system's functionality and employee information.

- Restricted database access
- Parameterized SQL
- User input validation

## **5. Implementation Notes (O2)**

### **A. Repository Structure**

```
capstone-II-spring/  
|—— .github/  
| |—— workflows/  
| |—— ci.yml  
|—— .vs/
```

```

├── capstone-II-spring/
│   ├── v17/
│   │   ├── .wsuo
│   │   └── DocumentLayout.json
│   └── VSWorkspaceState.json
├── database/
│   ├── IT_Tasks.sql
│   ├── More_sample_Employees.sql
│   ├── onboarding request.sql
│   ├── README.md
│   ├── Sample_Employee.sql
│   ├── Schema.sql
│   ├── Statuses.sql
│   └── Users.sql
├── src/
│   ├── EmployeeOnboarding/
│   │   ├── Business/
│   │   │   ├── HRDashboardService.vb
│   │   │   ├── RequestDetailsService.vb
│   │   │   └── RoleSelectionService.vb
│   │   ├── Data/
│   │   │   ├── InMemoryRequestStore.vb
│   │   │   └── SessionStore.vb
│   │   ├── Models/
│   │   │   ├── Employee.vb
│   │   │   └── OnboardingRequest.vb
│   │   ├── My Project/
│   │   │   ├── Application.Designer.vb
│   │   │   ├── Application.myapp
│   │   │   ├── AssemblyInfo.vb
│   │   │   ├── Resources.Designer.vb
│   │   │   ├── Resources.resx
│   │   │   ├── Settings.Designer.vb
│   │   │   └── Settings.settings
│   │   ├── App.config
│   │   ├── EmployeeOnboarding.vbproj
│   │   ├── frmHRdashboard.Designer.vb
│   │   ├── frmHRdashboard.resx
│   │   ├── frmHRdashboard.vb
│   │   ├── frmITdashboard.Designer.vb
│   │   ├── frmITdashboard.resx
│   │   ├── frmITdashboard.vb
│   │   ├── frmITProvision.Designer.vb
│   │   └── frmITProvision.resx

```

```

|   |—— frmITProvision.vb
|   |—— frmManagerDashboard.Designer.vb
|   |—— frmManagerDashboard.resx
|   |—— frmManagerDashboard.vb
|   |—— frmOnboardingRequest.Designer.vb
|   |—— frmOnboardingRequest.resx
|   |—— frmOnboardingRequest.vb
|   |—— frmRequestDetails.Designer.vb
|   |—— frmRequestDetails.resx
|   |—— frmRequestDetails.vb
|   |—— frmRoleSelection.Designer.vb
|   |—— frmRoleSelection.resx
|   |—— frmRoleSelection.vb
|—— .gitignore
|—— EmployeeOnboarding.sln
|—— README.md

```

## B. Coding Conventions

- PascalCase for all files
- UI files denoted with frm prefix
- Comments should give context for non-obvious or complex logic.

## C. Notable Patterns

- Model-View-Presenter (MVP): Creates different layers of functionality, making it easier to test and maintain the WinForms application.
- View represents the UI code that does not hold business logic.
- Presenter handles the UI logic, interacts with the model, and updates the view.
- Model represents the data and domain logic.
- Separation of Concerns (SoC): Separates sections of UI, business logic, and database to improve code readability, maintainability, and reusability.

## D. Tradeoffs

- Direct database access versus API layer: Chose direct database connection to align better with project scope.
- Desktop application versus web application: Chose desktop application to align with the team's past software development experience and technical strengths.

# 6. Testing & Evaluation (O3)

## A. Testing Strategy

- Unit testing during the feature developmental phases to ensure functionality of individual parts before building upon them.
- Integration testing conducted to support the connection of the layers (presentation, business logic, and data access).
- Manual testing for UI components.

## **B. Coverage**

- Testing coverage of the core workflow: high
- Testing coverage of edge cases: moderate

## **C. Performance and Reliability Testing**

- Test Cases Covered: User onboarding workflow, role-based access control, IT provisioning requests, and database interactions.
- Evaluation Approach: Verified expected outputs, ensured data consistency across layers, and validated successful completion of onboarding tasks.

## **D. KPIs/Metrics**

- Decrease new hire onboarding processing time by ~10%.
- Increase IT provisioning completion rate by ~5%.
- Maintain an error rate below 1% per onboarding request.

# **7. Security, Privacy, Accessibility & Compliance (O5)**

The Employee Onboarding and IT Access Management Portal addresses ethical considerations by ensuring fair and unbiased processing of all employee information throughout the onboarding process. It properly handles sensitive employee information, following labor laws and data protection to remain in compliance with legal requirements. Furthermore, security measures such as role-based access control and data encryption are implemented to protect personal and organizational information. Privacy is prioritized through the collection of only necessary data and secure storage of employee records. Additionally, the application supports accessibility by providing a user-friendly interface that accommodates individuals with diverse needs and abilities.

### **Features that Support these Considerations:**

- Onboarding Request status tracking creates transparency and ensures no employee is overlooked in the process.
- Maintains structured employee records in database tables to ensure consistency, accuracy, and proper organization of sensitive information.
- Logs updates and changes within the system to support auditability and accountability for HR, Manager, and IT actions.
- Applies role-based permissions, limiting unnecessary exposure of information.
- Integrates validation checks in forms to prevent incorrect or incomplete data entry during onboarding.
- Supports secure document handling by storing files in controlled and protected system environments.
- Ensures accessibility through a clear workflow interface, allowing new hires and staff to easily track progress and complete required steps.

# **8. Deployment & Operations**

The Employee Onboarding and IT Access Management Portal is deployed as a desktop application and requires manual setup of the application and database. Clone the Github repository and restore any

necessary dependencies in Microsoft Visual Studio. The PostgreSQL database must be locally configured with the necessary tables and schemas, then add the updated connection string in the application itself before execution. This application is run through Visual Studio by building the solution and launching the WinForms startup project. Ongoing operation depends on maintaining database availability, correct configuration settings, and manual management of updates and backups.

### [Appendices \(Evidence & How-To\)](#)

## **9. Limitations & Future Work**

### **A. Current Limitations**

- No web-based or mobile access to the application.
- No automatic notifications for status changes.
- Manual database backup
- Limited data reporting

### **B. Shortcuts and Technical Debt**

- Direct Database Access: This application connects WinForms directly to the PostgreSQL database for faster development and simpler architecture for the desktop application.
- Manual Database Setup: Database initialization is handled manually via SQL scripts, making it prone to setup inconsistencies and difficult to make schema changes in the future.
- Simplified Role-Based Access: True authentication methods (multi-factor authentication, Active Directory integration, etc.) were not utilized to eliminate the need for external identity providers, reducing overall complexity of the project scope.
- Basic UI Design: User interface is limited by WinForms design capabilities, leaving us unable to integrate responsive and modern UI elements.

### **C. Future Considerations**

1. Automatic email notifications for status changes.
2. In-depth dashboard analytics for each role.
3. Migration to a web-based and mobile application.
4. Integration with Active Directory.

## **10. References**

[1] "PostgreSQL 18.3 Documentation," *PostgreSQL Documentation*, Feb. 26, 2026.

<https://www.postgresql.org/docs/current/>

[2] "Windows Forms for .NET documentation," *Microsoft.com*, 2022. <https://learn.microsoft.com/en-us/dotnet/desktop/winforms/>

[3] G. Lindemulder and M. Kosinski, "What is role-based access control (RBAC)?," *IBM*, Aug. 20, 2024. <https://www.ibm.com/think/topics/rbac>

- [4] Claire Maynard, "Learn kanban with Jira," *Comprehensive kanban tutorial with Jira*.  
<https://www.atlassian.com/agile/tutorials/how-to-do-kanban-with-jira>
- [5] "How to Draw Architecture Diagrams," *GeeksforGeeks*, Mar. 04, 2025.  
<https://www.geeksforgeeks.org/system-design/how-to-draw-architecture-diagrams/>
- [6] "Layered (N-Tier) Architecture in .NET Core," *DEV Community*, Oct. 08, 2024.  
<https://dev.to/dotnetfullstackdev/layered-n-tier-architecture-in-net-core-51ic>
- [7] "Adding a workflow status badge - GitHub Docs," *GitHub Docs*, 2026.  
<https://docs.github.com/en/actions/how-tos/monitor-workflows/add-a-status-badge>
- [8] Jean-Paul Boodhoo, "Design Patterns: Model View Presenter," *learn.microsoft.com*, 2019.  
<https://learn.microsoft.com/en-us/archive/msdn-magazine/2006/august/design-patterns-model-view-presenter>
- [9] "Separation of Concerns (SoC)," *GeeksforGeeks*, Feb. 13, 2024.  
<https://www.geeksforgeeks.org/software-engineering/separation-of-concerns-soc/>
- [10] "Basic writing and formatting syntax," *GitHub Docs*. <https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>

## Appendices (Evidence & How-To)

### A. Installation Guide

1. Clone the repo.

```
git clone https://github.com/sebgui30/capstone-II-spring.git
cd capstone-II-spring
```

2. Set up the database (See **Database Setup** below).
3. Configure the connection string.
4. Restore dependencies (NuGet packages).

#### Database Setup

1. Open pgAdmin and create a database named: onboarding\_portal.
2. Create the following tables:

- employees
- onboarding\_requests
- users
- roles
- onboarding\_request\_statuses
- it\_tasks
- it\_task\_statuses

3. Ensure all tables exist under the onboarding schema.

*Note:* SQL scripts are not included in the repository. You may need to manually create the schema or request scripts from a team member.

4. Configure the database connection string:

```
Host=localhost;Port=5432;Username=postgres;Password=YOUR_PASSWORD;Database=onboarding_portal
```

*Notes:* Replace YOUR\_PASSWORD.

Make sure PostgreSQL is running on localhost:5432.

The connection string is hardcoded in multiple files; In Visual Studio: Ctrl + Shift + F → search connStr or connectionString → replace all.

#### Run with Visual Studio

1. Open the solution file: OnboardingAPI.sln
2. Restore or install NuGet packages if prompted (Npgsql version 6.0.8).
3. Set the main WinForms project as Startup Project.
4. Build the solution: Ctrl + Shift + B (Pre-run error check).
5. Click Start to run the application.

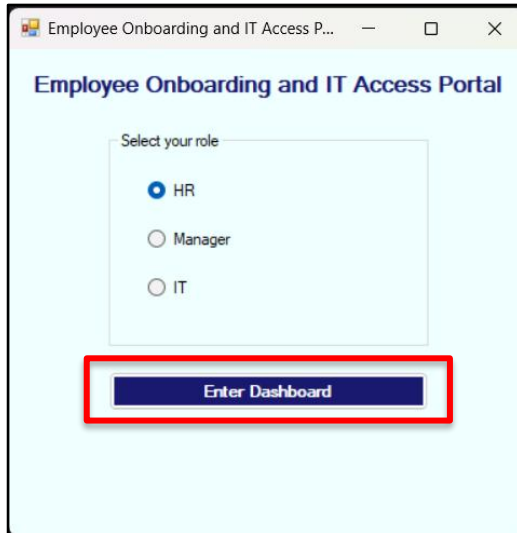
[View the complete README file here.](#)

[View the Install & Maintenance Guide Video here.](#)

## B. User Manual

### HR Walkthrough

1. Select the **HR** role on the login page and click *“Enter Dashboard”*.



Employee Onboarding and IT Access Portal

Select your role

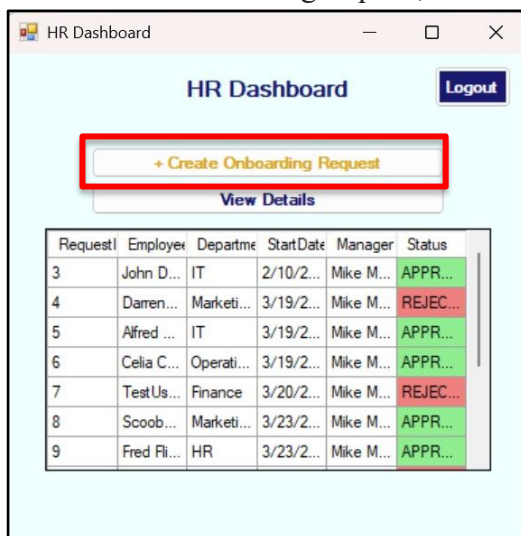
☒ HR

☐ Manager

☐ IT

Enter Dashboard

2. To create an onboarding request, click *“Create Onboarding Request”*.



HR Dashboard

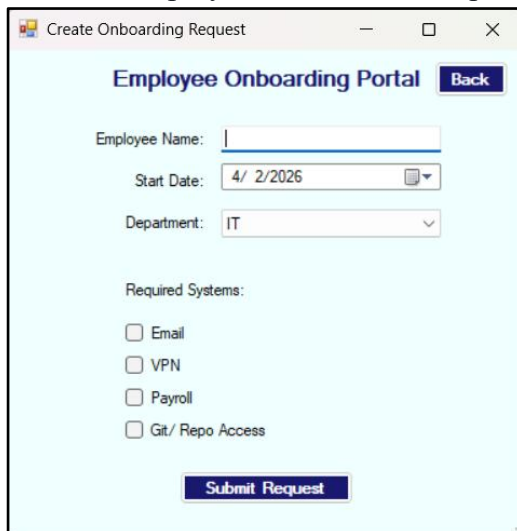
Logout

+ Create Onboarding Request

View Details

| Request# | Employee    | Departme   | StartDate | Manager   | Status   |
|----------|-------------|------------|-----------|-----------|----------|
| 3        | John D...   | IT         | 2/10/2... | Mike M... | APPR...  |
| 4        | Damen...    | Marketi... | 3/19/2... | Mike M... | REJEC... |
| 5        | Alfred ...  | IT         | 3/19/2... | Mike M... | APPR...  |
| 6        | Celia C...  | Operati... | 3/19/2... | Mike M... | APPR...  |
| 7        | TestUs...   | Finance    | 3/20/2... | Mike M... | REJEC... |
| 8        | Scoob...    | Marketi... | 3/23/2... | Mike M... | APPR...  |
| 9        | Fred Fli... | HR         | 3/23/2... | Mike M... | APPR...  |

3. Enter the employee’s name, date, department, and required systems.



Create Onboarding Request

Employee Onboarding Portal

Back

Employee Name:

Start Date:

Department:

Required Systems:

☐ Email

☐ VPN

☐ Payroll

☐ Git/ Repo Access

Submit Request

4. Click “*Submit Request*” to submit the onboarding request for manager review.

Create Onboarding Request

Employee Onboarding Portal [Back](#)

Employee Name:

Start Date: 4/2/2026

Department: IT

Required Systems:

☐ Email

☐ VPN

☐ Payroll

☐ Git/ Repo Access

[Submit Request](#)

5. Click “*Logout*” to return to the original portal login page.

HR Dashboard [Logout](#)

[+ Create Onboarding Request](#)

[View Details](#)

| Request | Employee    | Departme   | StartDate | Manager   | Status   |
|---------|-------------|------------|-----------|-----------|----------|
| 3       | John D...   | IT         | 2/10/2... | Mike M... | APPR...  |
| 4       | Darren...   | Marketi... | 3/19/2... | Mike M... | REJEC... |
| 5       | Alfred ...  | IT         | 3/19/2... | Mike M... | APPR...  |
| 6       | Celia C...  | Operati... | 3/19/2... | Mike M... | APPR...  |
| 7       | TestUs...   | Finance    | 3/20/2... | Mike M... | REJEC... |
| 8       | Scoob...    | Marketi... | 3/23/2... | Mike M... | APPR...  |
| 9       | Fred Fli... | HR         | 3/23/2... | Mike M... | APPR...  |

## Manager Walkthrough

1. Select the **Manager** role on the login page and click “*Enter Dashboard*”.

Employee Onboarding and IT Access Portal

Select your role

☐ HR

☒ Manager

☐ IT

[Enter Dashboard](#)

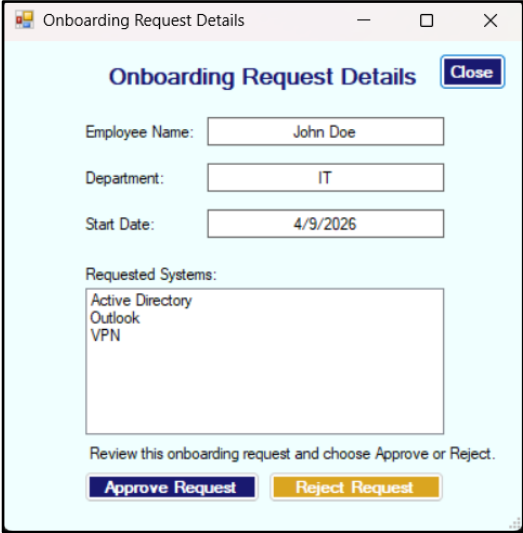
2. Select an employee in the dashboard view.



The screenshot shows a web application window titled "Manager Dashboard". It features a "Logout" button in the top right corner. Below the title, there is a section labeled "Pending Approvals:" containing a table with the following data:

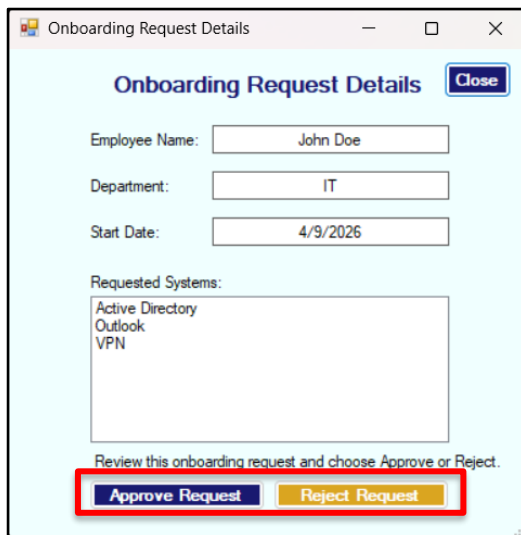
| Request# | Employee   | Department | StartDate | Manager   | Status   |
|----------|------------|------------|-----------|-----------|----------|
| 3        | John D...  | IT         | 2/10/2... | Mike M... | COMP...  |
| 4        | Darren...  | Marketi... | 3/19/2... | Mike M... | APPR...  |
| 5        | Alfred ... | IT         | 3/19/2... | Mike M... | APPR...  |
| 6        | Celia C... | Operati... | 3/19/2... | Mike M... | APPR...  |
| 7        | Test Us... | Finance    | 3/20/2... | Mike M... | REJEC... |
| 8        | Scoob...   | Marketi... | 3/23/2... | Mike M... | APPR...  |
| 9        | Fred Fl... | HR         | 3/23/2... | Mike M... | APPR...  |
| 10       | Tony ...   | Finance    | 3/23/2... | Mike M... | REJEC... |
| 11       | Kylan ...  | Operati... | 3/25/2... | Mike M... | APPR...  |
| 12       | Timmy ...  | Marketi... | 3/25/2... | Mike M... | APPR...  |
| 13       | Frankli... | Operati... | 3/26/2... | Mike M... | REJEC... |

3. On the Onboarding Request Details page, review the employee information and requested systems.



The screenshot shows a web application window titled "Onboarding Request Details". It features a "Close" button in the top right corner. Below the title, there are three input fields for "Employee Name" (John Doe), "Department" (IT), and "Start Date" (4/9/2026). Below these fields is a section labeled "Requested Systems:" containing a list box with the following items: Active Directory, Outlook, and VPN. At the bottom of the window, there is a text prompt: "Review this onboarding request and choose Approve or Reject." followed by two buttons: "Approve Request" and "Reject Request".

4. Click "Approve Request" to approve the onboarding request or click "Reject Request" to deny the onboarding request.



**Onboarding Request Details** [Close]

Employee Name:

Department:

Start Date:

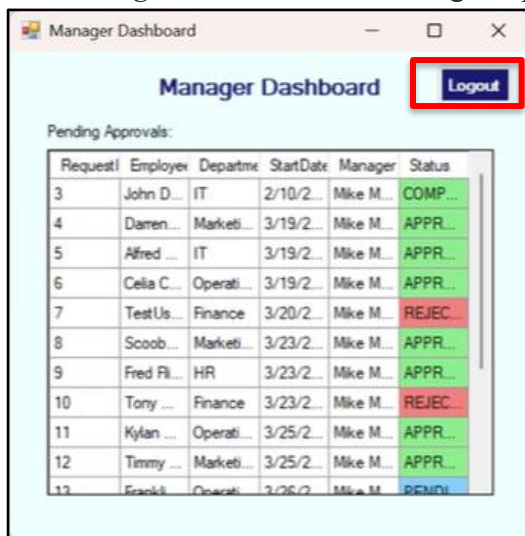
Requested Systems:

Active Directory  
 Outlook  
 VPN

Review this onboarding request and choose Approve or Reject.

Approve Request
Reject Request

6. Click “Logout” to return to the original portal login page.



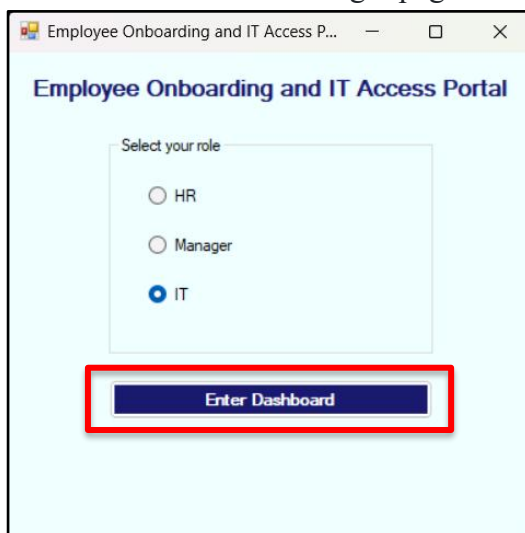
**Manager Dashboard** [Logout]

Pending Approvals:

| Request# | Employee   | Department | StartDate | Manager   | Status   |
|----------|------------|------------|-----------|-----------|----------|
| 3        | John D...  | IT         | 2/10/2... | Mike M... | COMP...  |
| 4        | Darren...  | Marketi... | 3/19/2... | Mike M... | APPR...  |
| 5        | Alfred ... | IT         | 3/19/2... | Mike M... | APPR...  |
| 6        | Celia C... | Operati... | 3/19/2... | Mike M... | APPR...  |
| 7        | Test Us... | Finance    | 3/20/2... | Mike M... | REJEC... |
| 8        | Scoob...   | Marketi... | 3/23/2... | Mike M... | APPR...  |
| 9        | Fred Fl... | HR         | 3/23/2... | Mike M... | APPR...  |
| 10       | Tony ...   | Finance    | 3/23/2... | Mike M... | REJEC... |
| 11       | Kylan ...  | Operati... | 3/25/2... | Mike M... | APPR...  |
| 12       | Timmy ...  | Marketi... | 3/25/2... | Mike M... | APPR...  |
| 13       | Franki...  | Operati... | 3/26/2... | Mike M... | PENDI... |

## IT Walkthrough

1. Select the **IT** role on the login page and click “Enter Dashboard”.



**Employee Onboarding and IT Access Portal**

Select your role

☐ HR

☐ Manager

☒ IT

Enter Dashboard

2. Select an employee in the dashboard view.

IT Dashboard

Logout

| RequestId | Employee     | Department | StartDate  | Manager    | Status   |
|-----------|--------------|------------|------------|------------|----------|
| 3         | John Doe     | IT         | 2/10/20... | Mike Ma... | COMPL... |
| 4         | Darren ...   | Marketing  | 3/19/20... | Mike Ma... | APPRO... |
| 5         | Alfred pe... | IT         | 3/19/20... | Mike Ma... | APPRO... |
| 6         | Celia Cruz   | Operations | 3/19/20... | Mike Ma... | APPRO... |
| 8         | Scoober...   | Marketing  | 3/23/20... | Mike Ma... | APPRO... |
| 9         | Fred Fin...  | HR         | 3/23/20... | Mike Ma... | APPRO... |
| 11        | Kylan G...   | Operations | 3/25/20... | Mike Ma... | APPRO... |
| 12        | Timmy T...   | Marketing  | 3/25/20... | Mike Ma... | APPRO... |
| 15        | Eddie M...   | Marketing  | 4/10/20... | Mike Ma... | APPRO... |
| 18        | Jerry Sy...  | Operations | 4/11/20... | Mike Ma... | APPRO... |

View Details

- On the IT Provision Details page, review the approved employee information and requested systems.

frmITProvision

IT Provision Details

Close

Employee Name: Jerry Symbleme

Department: Operations

Start Date: 4/11/2026

Requested Systems:

- Email
- VPN
- Git / Repo Access

Review the requested systems and mark provisioning as complete when setup is finished.

Provision Complete

- Click "Provision Complete" when all provisioning tasks are completed.

frmITProvision

IT Provision Details

Close

Employee Name: Jerry Symbleme

Department: Operations

Start Date: 4/11/2026

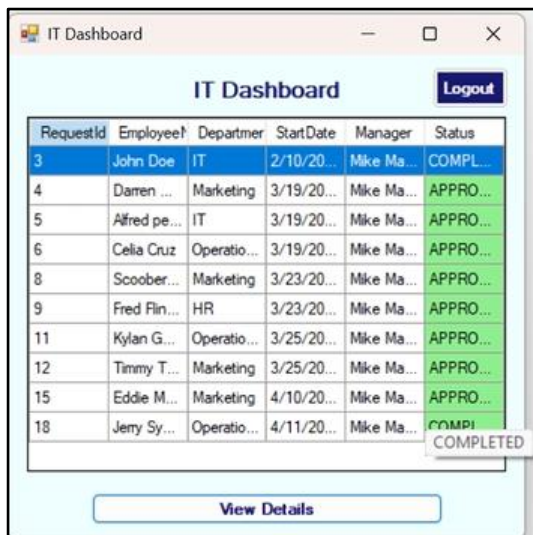
Requested Systems:

- Email
- VPN
- Git / Repo Access

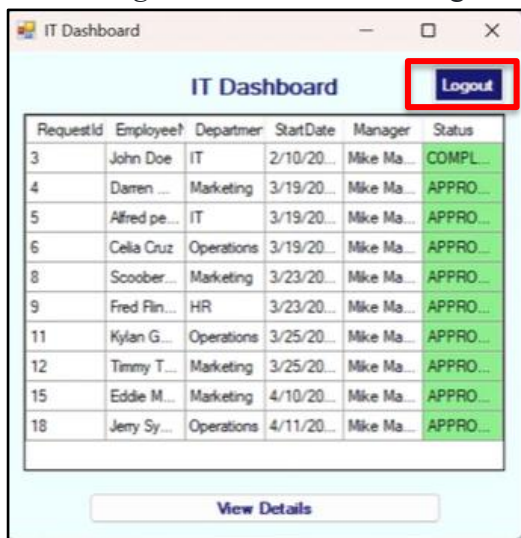
Review the requested systems and mark provisioning as complete when setup is finished.

Provision Complete

- The onboarding workflow is now completed for the selected employee.



- Click “Logout” to return to the original portal login page.



[View the User Guide Video here.](#)

### C. Admin/Operations Guide

The system is a desktop-based onboarding management application built with Microsoft Windows Forms and backed by a PostgreSQL database. It is used internally by HR, Managers, and IT to manage onboarding workflows.

#### Deployment & Startup (Runbook)

To start the system:

1. Ensure PostgreSQL database service is running.
2. Verify the database connection string is correctly configured in the application settings.
3. Launch the application via Visual Studio.
4. Confirm successful database connection on application startup.

#### Routine Operations

- Monitor onboarding requests through the application interface.

- Verify completion of IT provisioning tasks.
- Ensure database connectivity remains active.
- Review application behavior for any unexpected errors or failed processes.

### **Issue Handling**

Common issues and resolutions:

- Database connection failure: Verify database service is running and the connection string is correct.
- Failed onboarding request: Review input data and retry the request.
- Application crash or unresponsiveness: Restart the application and verify system resources.

If issues persist, escalate by reviewing system configuration and database status.

### **Backup & Restore**

- Backup: Perform periodic backups of the PostgreSQL database.
- Restore: In case of data loss, restore the database using a backup file..

After restoration, verify data integrity and system functionality.

### **Security & Access**

- Role-based access is implemented to restrict system functionality.
- Database credentials should be securely stored and not exposed in source code.
- Access to the system should be limited to authorized users.

### **Maintenance**

- Apply updates to the application as needed.
- Update database schema carefully to avoid data inconsistencies.
- Perform periodic testing to ensure system functionality after changes.

## **D. API Reference**

N/A

## **E. Project Poster, Presentation Slides, and Video Demo Links**

- Project Poster: [IT - EmployeeOnboardingAndITAccessManagementPortal.pdf](#)
- Presentation Slides: [IT - EmployeeOnboardingAndITAccessManagementPortal.ppt](#)
- Video Demos Link: [index.html](#)

## **F. Scrum Evidence Index**

- Sprint Plannings: [plannings](#)
- Sprint Backlogs: [backlogs](#)
- Sprint Dailies: [dailies](#)
- Sprint Retrospectives: [retros](#)
- Sprint Reviews: [reviews](#)