Design Specification

eTimely

v1.3

3/6/2022

Matt Korte

Caleb Obi

Chris Gumieny

Samia Chowdhury

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 2/26/2022 | Version 1 | Matthew Korte  Caleb Obi  Chris Gumieny  Samia Chowdhury | First draft sent to GTA for review |
| 3/1/2022 | Version 1.1 | Matthew Korte  Caleb Obi  Chris Gumieny  Samia Chowdhury | Second draft based on GTA feedback |
| 3/5/2022 | Version 1.2 | Matthew Korte  Caleb Obi  Chris Gumieny  Samia Chowdhury | Edited based off feedback from presentations |
| 3/6/2022 | Version 1.3 | Matthew Korte  Caleb Obi  Chris Gumieny  Samia Chowdhury | Edited based off feedback from Darla |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Matthew Korte | Team Lead | 3/5/22 |
|  | Caleb Obi | Backend Lead / QA Lead | 3/5/22 |
|  | Chris Gumieny | Frontend Lead | 3/5/22 |
|  | Samia Chowdhury | Presentation Lead / Documentation Lead | 3/5/22 |

TABLE OF CONTENTS

[Revision History II](#_Toc97478246)

[Document Approval II](#_Toc97478247)

[1. Introduction 1](#_Toc97478248)

[1.1 Purpose 1](#_Toc97478249)

[2. General Overview and Design Guidelines/Approach 1](#_Toc97478250)

[2.1 Assumptions & Constraints 1](#_Toc97478251)

[2.1.1 Assumptions 1](#_Toc97478252)

[2.1.2 Technical Constraints 1](#_Toc97478253)

[3. Architecture Design 2](#_Toc97478254)

[3.1 Hardware Architecture 2](#_Toc97478255)

[3.2 Software Architecture 3](#_Toc97478256)

[3.3 Security Architecture Caleb 6](#_Toc97478257)

[3.4 Communication Architecture 7](#_Toc97478258)

[4. System Design 8](#_Toc97478259)

[4.1 Use-Cases 8](#_Toc97478260)

[4.1.1 Business Use Cases 8](#_Toc97478261)

[4.1.2 Staff Use Cases 16](#_Toc97478262)

[4.1.3 Both User Type Use Cases 19](#_Toc97478263)

[4.1.4 Default Use Cases 21](#_Toc97478264)

[4.3 Sequence Diagram 25](#_Toc97478265)

[4.3.1 User Sign Up 26](#_Toc97478266)

[4.3.2 Entering Availability 27](#_Toc97478267)

[4.3.3 Confirmation Email 28](#_Toc97478268)

[4.4 Data Flow Diagrams 31](#_Toc97478269)

[4.4.1 Business User Data Flow Diagram 31](#_Toc97478270)

[4.4.2 Staff User Data Flow Diagram 32](#_Toc97478271)

[4.5 Database Design 32](#_Toc97478272)

[4.6 Application Program Interfaces 33](#_Toc97478273)

[4.7 User Interface Design 33](#_Toc97478274)

[4.7.1 Default View 34](#_Toc97478275)

[4.7.2 Staff View 38](#_Toc97478276)

[4.7.3 Business View 47](#_Toc97478277)

[Appendix A: References 55](#_Toc97478278)

[Appendix B: Key Terms 56](#_Toc97478279)

# 

# 1. Introduction

## 1.1 Purpose

This document's objective is to explain and track the architecture and design of eTimely application in technical terms. This comprises the architecture of hardware, software, security, and communications. The primary target for this document is the development team who will use it to help guide them through the process of designing and implementing the system. However, certain details such as the user interface, should be shared with the client, users, or any other stakeholder (if applicable) to obtain input or permission.

# 2. General Overview and Design Guidelines/Approach

## 2.1 Assumptions & Constraints

### 2.1.1 Assumptions

1. The application has a stable internet connection. Without it the application would not be able to connect to the server, thus it would not be able to fetch any information from the database. If we were to back up any information to a local machine, changes to the software architecture would need to be made.
2. The project will be using Firestore throughout its life cycle. If the database were to change at some point, much of the software architecture and database design would likely need to be changed as well.

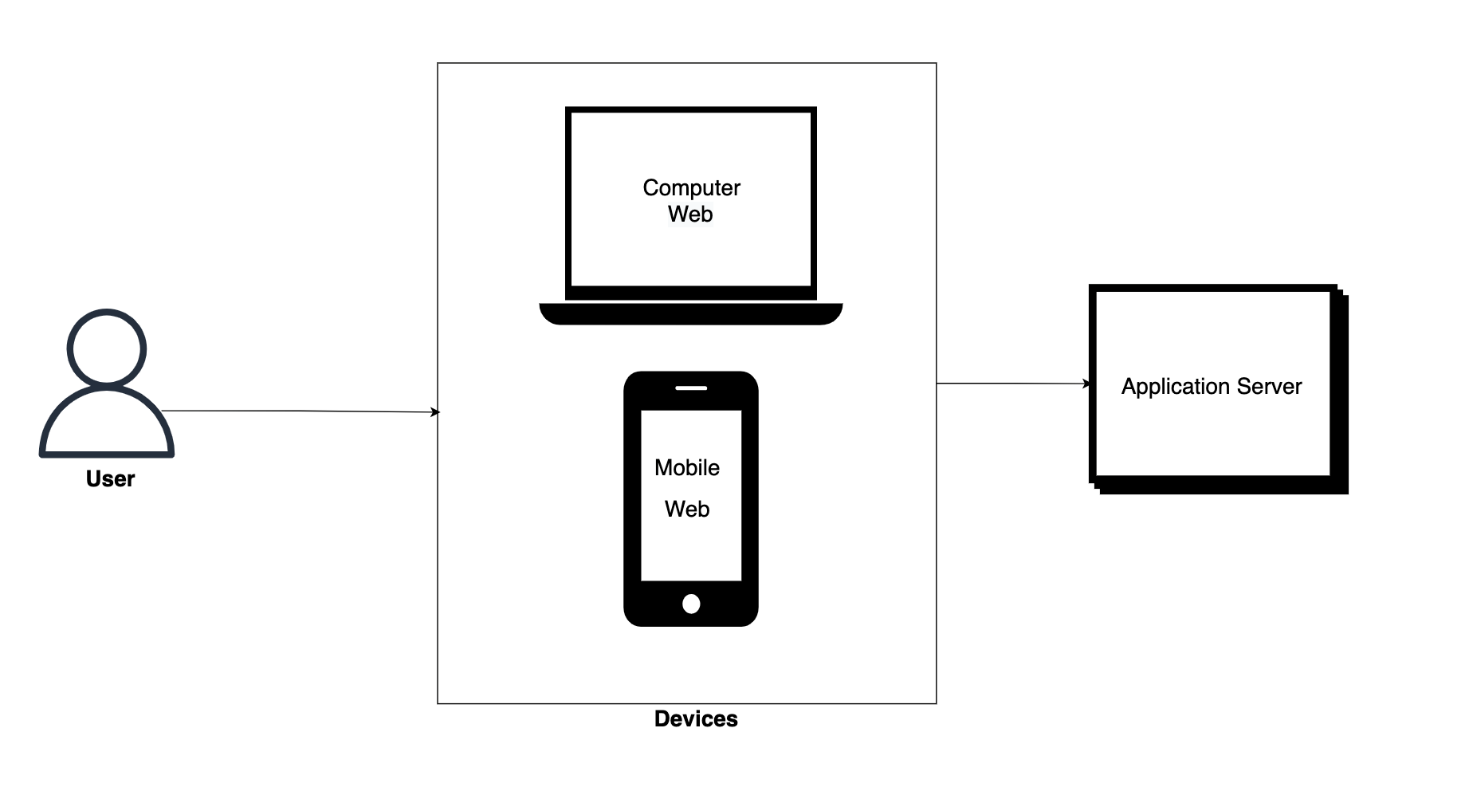
### 2.1.2 Technical Constraints

1. React v17.0.2 is being used for the frontend along with CSS for styling, limiting the choices for tools and libraries to be used to develop the frontend and user interface.
2. React is not fully compatible with older web browsers like Internet Explorer 10 [1]. This means we cannot guarantee functionality on such web browsers and will instead focus on “modern” browsers such as Google Chrome v90+, Microsoft Edge v91+, Mozilla Firefox v78+, and Apple Safari 14+.
3. The backend is being made with NodeJS v16.13.2 and ExpressJS v4.17.2.
4. This app is using the free version of Firebase, meaning the following constraints will be placed on the database [2].
   1. The amount of data stored will be limited to 1 GB
   2. Limit of 50,000 reads per day
   3. Limit of 20,000 writes per day
   4. Limit of 20,000 deletes per day

# 3. Architecture Design

## 3.1 Hardware Architecture

The eTimely application consists of three main components. As shown in the visualization below, there is the user, the devices section that shows the list of compatible devices that the users can use to view the application, and the application server.

**Figure 3.1**

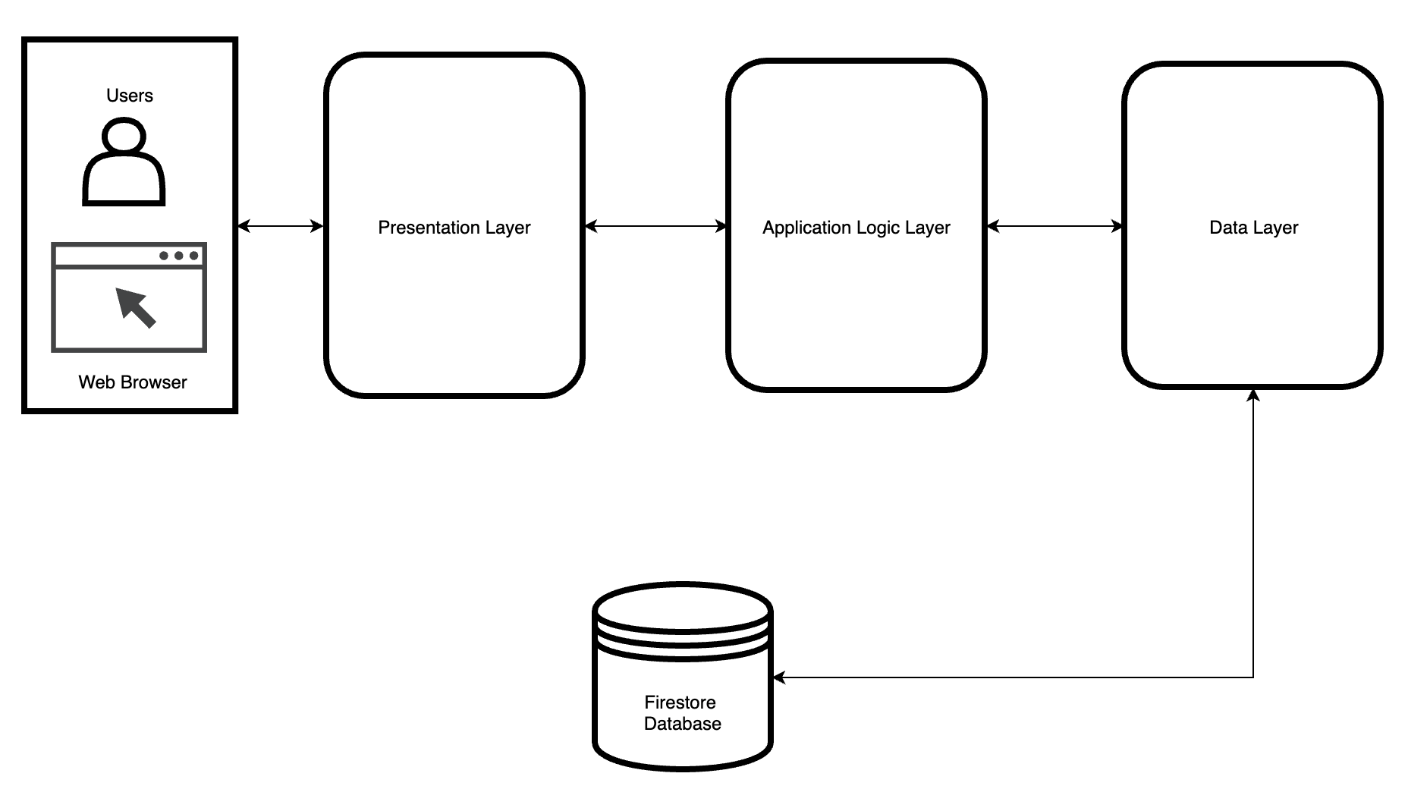
Devices

* Mobile device web browser
* Computer web browser

The application user interface will feature a fully responsive design that will allow users to view the application on a mobile device and have the same experience when accessing the application on a laptop web browser. Users can access the application using a web browser on any of the devices listed above. The application server will be responsible for getting and processing requests received from the client devices.

## 3.2 Software Architecture

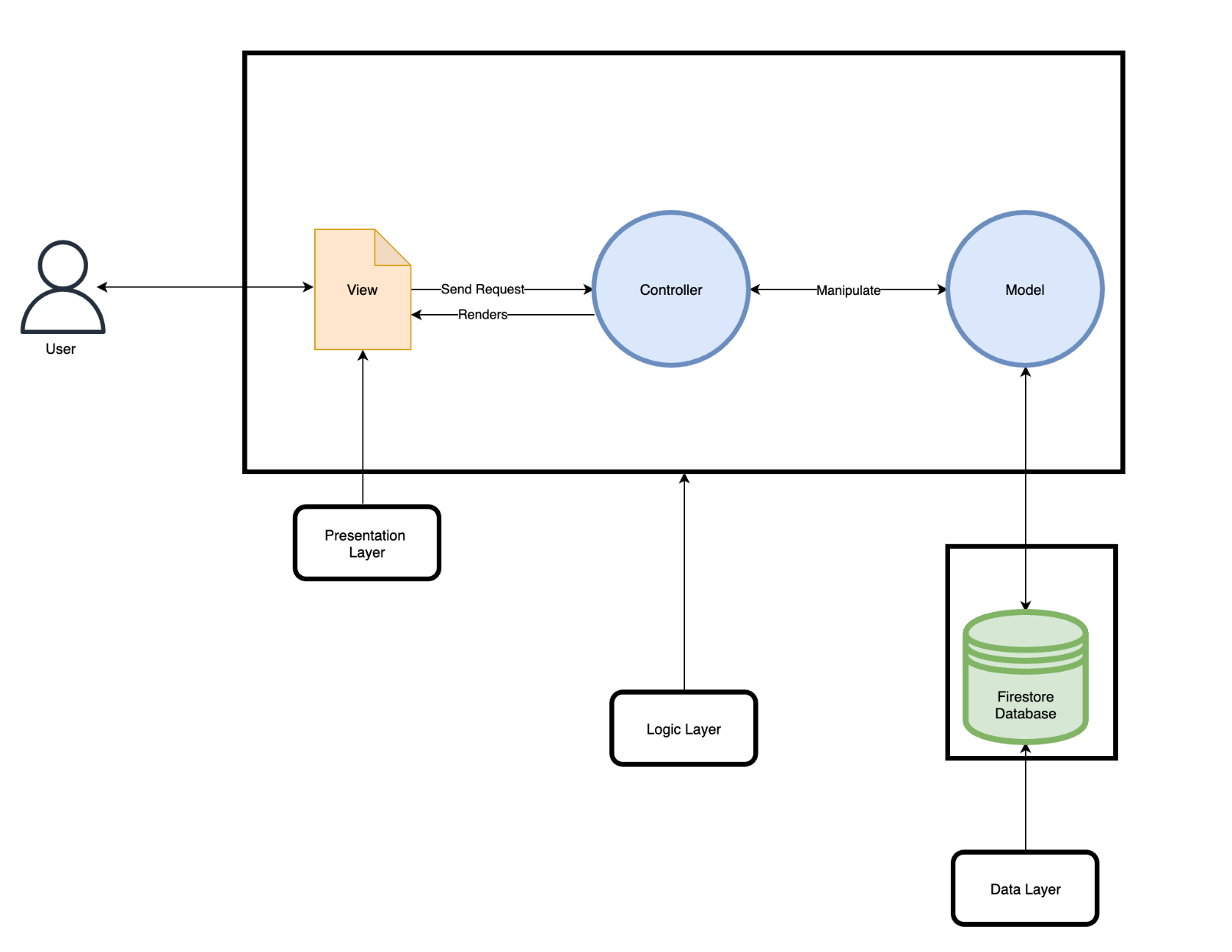
The eTimely application’s software architecture will utilize the three-tiered architecture paradigm as shown in the image below.



**Figure 3.2.1**

The **presentation layer**, often known as the client tier, oversees creating a user interface that can be viewed in a browser and with which users may interact. The presentation tier is the user interface and communication layer of the application where the end-user interacts with it. Its primary function is to display information to the user and collect information from them. The eTimely application will utilize ReactJS and Material UI to provide a user interface with which users can interact with.

The **logic layer** is often referred to as the application layer and business layer or middle tier. This layer processes information obtained in the presentation tier, sometimes in conjunction with the data gathered in the data tier. All the Rest APIs (Application Programming Interfaces) that will be used in this application will be housed in this layer. This layer’s APIs will oversee connecting with both the presentation layer and the data tier via HTTP requests. To construct all application’s logic, the eTimely application will use NodeJS and ExpressJS. Furthermore, the logic layer will arrange all the logic that will be used using the MVC (Model View Controller) design pattern as shown below.

**Figure 3.2.2**

The model contains the data as well as the code's business logic or service layer logic. It contains all the content's data structures, properties, and attributes that we need to transfer and manipulate entities, domain models, and view models. This might be either the data being transmitted between the view and controller components, or any other data related to business logic.

The view will represent the presentation layer, the user interface handles displaying the logic of code and containing the layout. It facilitates user interaction with the system by receiving input and providing suitable outputs to the user

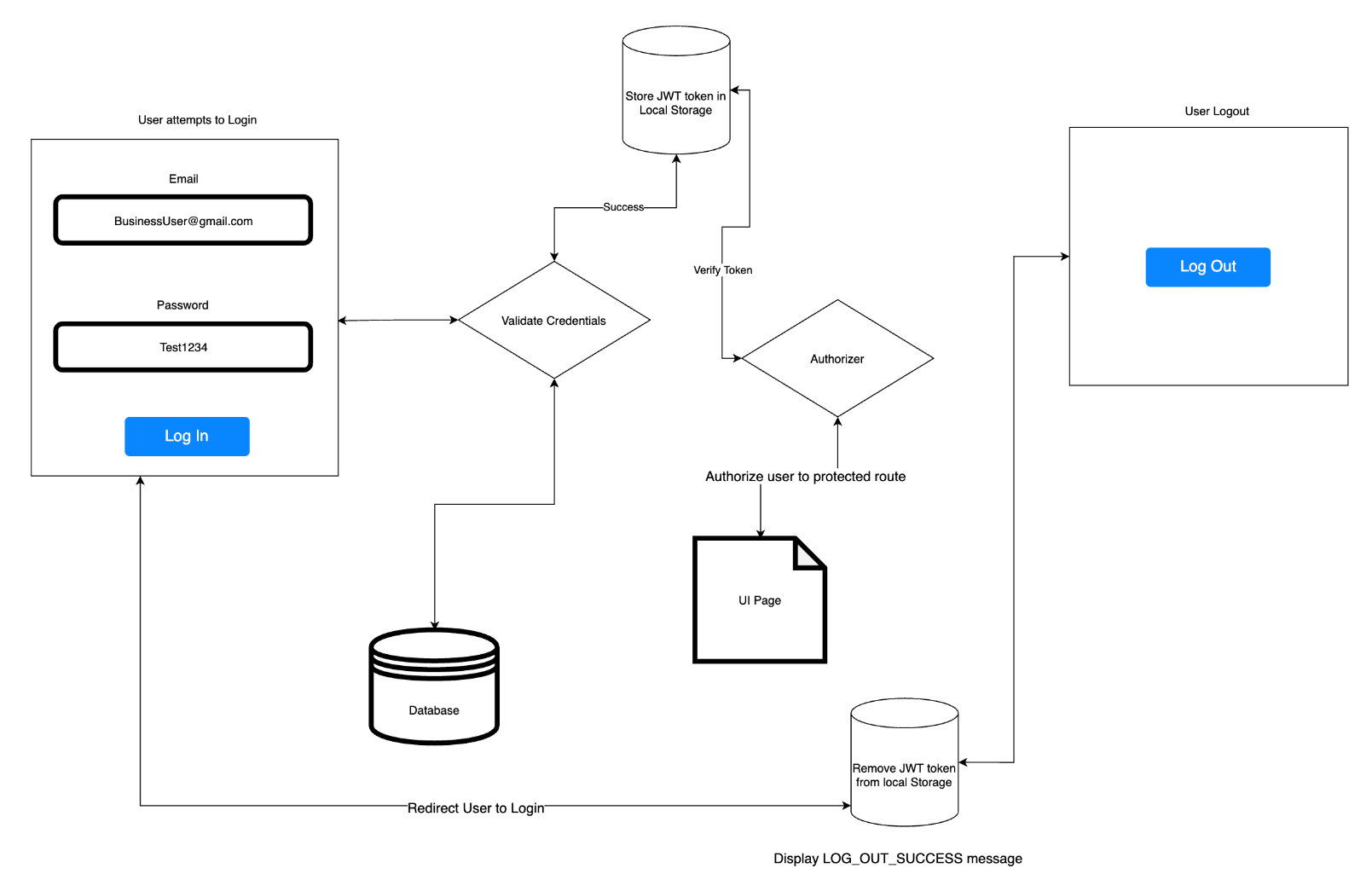
The controllers serve as a link between the model and view components, processing all business logic and incoming requests, manipulating data using the model, and interacting with views to produce the final output.

The **data tier,** also known as the database tier, data access tier, or back-end, is where the application's data is kept and maintained. The eTimely application will utilize Firebase Firestore to save and maintain all data used in the application.

## 3.3 Security Architecture Caleb

**3.3.1 Web Tokens**

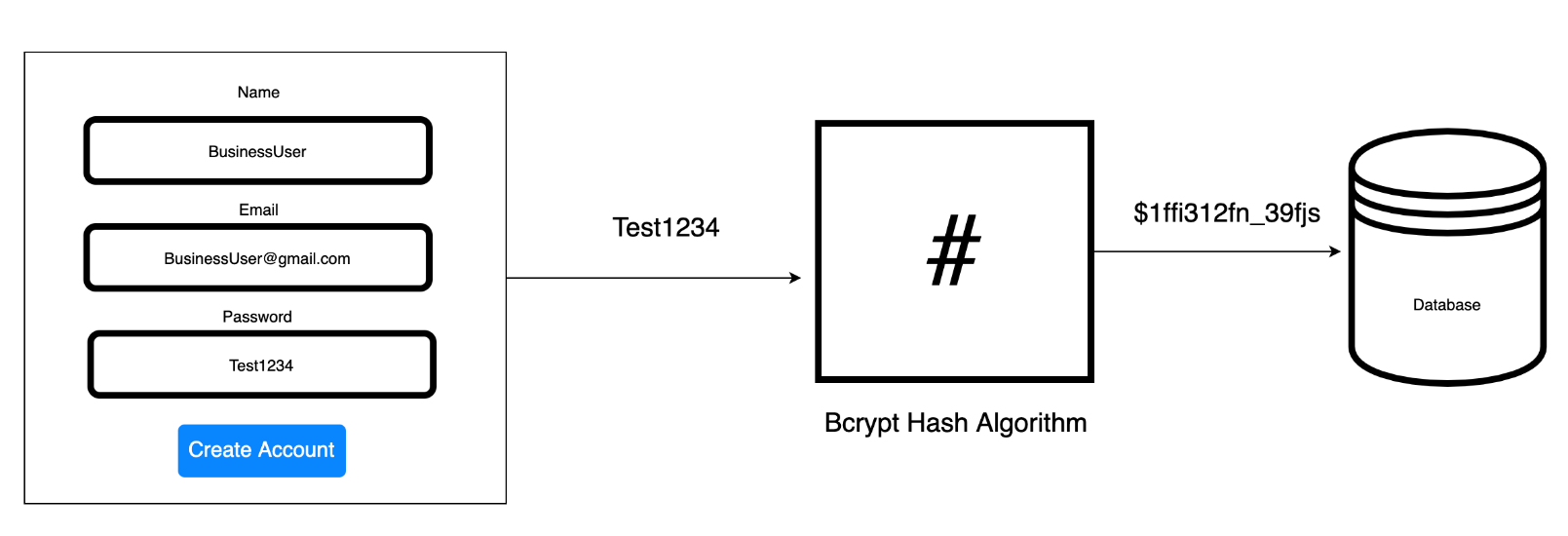
The eTimely application will utilize Json Web Tokens to authenticate and authorize users in the application. Once the user is logged in, each subsequent request will include the JWT, allowing the user to access protected pages and services that are permitted with that token.

**Figure 3.3.1**

The diagram above shows a high-level flow diagram illustrating how the eTimely application will use Jason Web Token (JWT) to authenticate and authorize users.

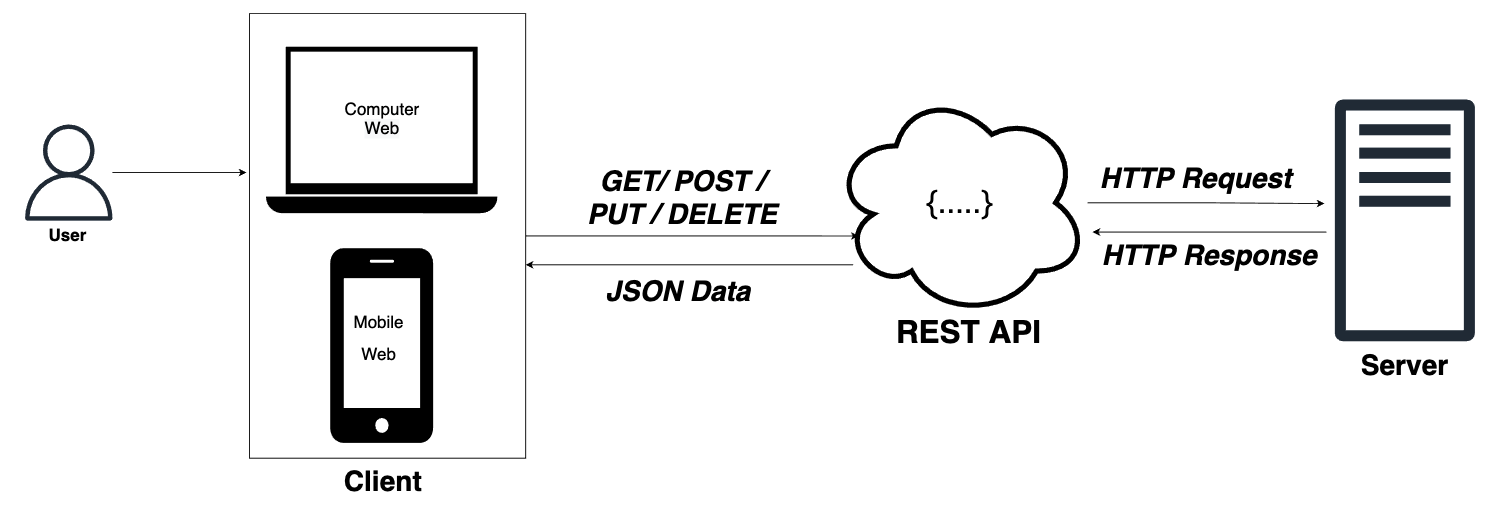
**3.3.2 Password Encryption**

The eTimely app will never save a user’s password in plain text. Password will always be hashed before being saved on the server and database using the bycrpt NodeJS package. In this manner, we can assure that a user’s information is kept secure.

**Figure 3.3.2**

## 3.4 Communication Architecture

The etimely application communication will use an HTTP request to communicate with both the client and the server using the REST (Representational State Transfer) API (Application Programming Interfaces). The figure below depicts how the client will interact with the server. The client will send queries to the API to retrieve information or make changes to the application. The server will receive the request from the client, process it, and then return a JSON response to the client.



**Figure 3.4**

# 4. System Design

## 4.1 Use-Cases

### 4.1.1 Business Use Cases

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-1 | | **Use Case Name:** Business Sign-Up |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Samia Chowdhury | | **Last Revision Date:** 3/3/2022 |
| **Actors** | New Business Users | |
| **Description** | New business user creating a business account | |
| **Trigger** | User clicking on the “Create Account” button | |
| **Preconditions** | User is on the sign-up page | |
| **Postconditions** | User account information added to database  Account verification email is sent to the business user email address | |
| **Normal Flow** | 1. User fills the business name, email, password, and confirm password fields 2. User checks the box to agree to terms and conditions 3. User clicks the “Sign-Up” button 4. Account conformation email is sent to the email the user entered 5. User is notified of the confirmation email being sent | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If the any of the fields are not filled in when the user clicks the “sign-up” button, they are shown an error message below that field telling them it is required. | |
| **Assumptions** | User enters an email that is not already in the database. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-2 | | **Use Case Name:** Business Login |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Samia Chowdhury | | **Last Revision Date:** 3/3/2022 |
| **Actors** | Business Users | |
| **Description** | Business user logins into their already existing account | |
| **Trigger** | The “Login” button on the sign in page | |
| **Preconditions** | User has created a business account | |
| **Postconditions** | User directed to dashboard | |
| **Normal Flow** | 1. User navigates to the sign in page 2. User enters in login information 3. User clicks “Sign In” 4. User is brought to the business dashboard | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If the user enters an email and password which do not match a created account, an error message is shown, and they are not redirected to the dashboard. | |
| **Assumptions** | User has created and activated an account. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-3 | | **Use Case Name:** Business Account Verification |
| **Created By:** Matt Korte | | **Date Created:** 2/25/2022 |
| **Last Updated By:** Samia Chowdhury | | **Last Revision Date:** 3/3/2022 |
| **Actors** | New Business Users | |
| **Description** | New business user creating a business account | |
| **Trigger** | Creating a new business account | |
| **Preconditions** | UC-1 has occurred | |
| **Postconditions** | A user account has been confirmed, and they can now access their account. | |
| **Normal Flow** | 1. Business users create an account using a business name, email, and password. 2. Verification email will be sent to the business user email address. 3. Business user clicks the verification link that was sent to their email address 4. They are redirected to a page that lets them know that their account is verified successfully. | |
| **Alternative Flow** | N/A | |
| **Exceptions** | In the case that the user does not open the verification link, it will expire in 24 hours which will require the user to repeat the normal flow. | |
| **Assumptions** | Email service used to send verification email is operational | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-4 | | **Use Case Name:** Business User Approval |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/2/2022 |
| **Actors** | Business Users | |
| **Description** | Business user approves a scheduling request for a staff member | |
| **Trigger** | UC-1 | |
| **Preconditions** | 1. User is signed into a business account and is on the business dashboard 2. UC-15 or UC-16 has occurred | |
| **Postconditions** | Notification is sent back to the staff user that sent the request alerting them of their request is approved. Additionally, the approved availability change/day off is saved in the database. | |
| **Normal Flow** | 1. User clicks the “View” button next to the name of the staff member with a pending request 2. A dialog box opens showing the details of the staff member’s scheduling request 3. User elects to accept the request 4. Notification is sent to staff user that request was expected 5. Availability information is updated for UC-6 | |
| **Alternative Flow** | 1. Repeat steps 1 and 2 of the normal flow 2. Business user rejects the request 3. Notification is sent to the staff user that the request was denied | |
| **Exceptions** | N/A | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-5 | | **Use Case Name:** Business User Announcements |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Samia Chowdhury | | **Last Revision Date:** 3/2/2022 |
| **Actors** | Business Users | |
| **Description** | Actor makes an announcement post | |
| **Trigger** | User navigates to announcement (or dashboard for alternative flow) page | |
| **Preconditions** | User is signed into a business account | |
| **Postconditions** | Announcement will be visible to all user in the organization on the “announcements” page | |
| **Normal Flow** | 1. User navigates to announcements page from navigation menu 2. User types an announcement into the input text field 3. User then hits the “Publish” button | |
| **Alternative Flow** | 1. Navigates to dashboard 2. Types announcements into input text box 3. Presses “Publish” button | |
| **Exceptions** | If text field for the announcement is empty, clicking the “publish” button will have no effect. | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-6 | | **Use Case Name:** Schedule Creation |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Samia Chowdhury | | **Last Revision Date:** 3/2/2022 |
| **Actors** | Business Users | |
| **Description** | Business users create the schedule for their staff | |
| **Trigger** | The business user navigates to the scheduling screen | |
| **Preconditions** | The user is signed into a business account | |
| **Postconditions** | The schedule information is saved in the database, to later be called by the staff users to view later. | |
| **Normal Flow** | 1. User navigates to the scheduling screen 2. User enters the specific date they want to create a schedule on 3. Select the specific role type they are scheduling to narrow down list of staff members 4. For a specific staff member, click the “Add” button 5. Enter shift start and end time 6. Click “Accept” button 7. Click the “Publish” button | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If the user tries to schedule a staff member at a time that they are unavailable, they will be prompted with a warning message which will ask them if they still want to schedule them. | |
| **Assumptions** | There are staff members in the organization to schedule. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-7 | | **Use Case Name:** Staff User Invite Generation |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/2/2022 |
| **Actors** | Business Users | |
| **Description** | Business user creating an invite link to add a new staff user | |
| **Trigger** | Clicking the “Invite Staff” button on the roster management page | |
| **Preconditions** | The user is signed into a business account | |
| **Postconditions** | An email is sent to a new staff user to create an account linked to the business account, along with a confirmation message showing that the invite was sent. | |
| **Normal Flow** | 1. Business user navigates to the roster management page 2. Clicks the “Invite Staff” button 3. Enters new staff member’s email, first name, last name address in the resulting dialog box 4. Clicks the “Invite” button 5. Invitation email is sent to the entered email address | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If any of the fields are left empty when pressing the “invite” button, an error message is shown prompting the user to fill them and try again. | |
| **Assumptions** | The entered email address is not already associated with an account | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-8 | | **Use Case Name:** Business User Profile Update |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/2/2022 |
| **Actors** | Business Users | |
| **Description** | Business user updating account information | |
| **Trigger** | Clicking on Settings tab on the left navigation | |
| **Preconditions** | Business account exists and business user is logged into the application | |
| **Postconditions** | All changes are saved to database | |
| **Normal Flow** | 1. Actor navigates to Settings page and clicks on profile. 2. Actor updates the business name and/or email address. 3. Actor clicks on “Update” button to save new changes | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If email address is already in use, give user an error message and prompt them to try a different one | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-9 | | **Use Case Name:** Remove staff member from roster |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 2/28/2022 |
| **Actors** | Business Users | |
| **Description** | Business user removing staff user form organization | |
| **Trigger** | User navigating to the roster management page | |
| **Preconditions** | 1. There is at least one staff user in the organization 2. The user is signed into a business account | |
| **Postconditions** | The staff user’s account is removed from the organization and is deleted | |
| **Normal Flow** | 1. Business user navigates to roster management page 2. Clicks the “Edit” button 3. Clicks the “Remove” button next to a user 4. Clicks “Confirm” in the resulting dialog box 5. Staff user account is removed from database 6. Return to roster management page | |
| **Alternative Flow** | 1. Follow steps 1-3 of the normal flow 2. User clicks the “Cancel” button 3. Return to roster management page with staff user still listed | |
| **Exceptions** | If there are no staff users in the organization, there will be no prompt to edit a staff member’s information | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-10 | | **Use Case Name:** Create New Role Type |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/2/2022 |
| **Actors** | Business Users | |
| **Description** | Business user creating a new role type to later be applied to staff users | |
| **Trigger** | Pressing the “Add Role” button on the settings page | |
| **Preconditions** | Business user is signed in | |
| **Postconditions** | New role type is saved in database, to be accessed in UC-11 | |
| **Normal Flow** | 1. After pressing the “Add Role” button 2. User types in a role name into the text box 3. Pressing the “Submit” button | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If the entered role is already in the database, give the user an error message and the role is not added to the database | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-11 | | **Use Case Name:** Assigning Role Types |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Samia Chowdhury | | **Last Revision Date:** 3/3/2022 |
| **Actors** | Business Users | |
| **Description** | Business user assigning roles to staff users within the organization | |
| **Trigger** | The “Edit” button on the roster page | |
| **Preconditions** | Business user is sign in | |
| **Postconditions** | The staff member now has a role assigned to them | |
| **Normal Flow** | 1. Navigate to the roster management page 2. Hitting the “Edit” button next to a staff members name 3. Click the “Edit Role” button 4. User assigns the staff member a role from the drop-down list and clicks the “Save” button 5. Returns to the default view of the roster page | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If no roles have been created, the drop-down list will be empty, and the user will not be able to assign a role to the user | |
| **Assumptions** | * There are staff members in the organization * The business user has created at least one role | |

### 4.1.2 Staff Use Cases

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-12 | | **Use Case Name:** Staff Sign-Up |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/2/2022 |
| **Actors** | Staff Users | |
| **Description** | Staff user making an eTimely account | |
| **Trigger** | Opening link sent in UC-7 | |
| **Preconditions** | UC-7 has occurred previously | |
| **Postconditions** | User account credentials (name, email, and password) are saved in the database, then they are rerouted to the dashboard | |
| **Normal Flow** | 1. User is directed to the staff sign up page 2. User fills in the first name, last name, password, and confirm password fields 3. Press the “Sign-Up” button 4. User is redirected to the staff user dashboard | |
| **Alternative Flow** | N/A | |
| **Exceptions** | In the case that the password and confirm passwords fields do not match, the user is shown an error message alerting them of this and they must try again. | |
| **Assumptions** | User has opened the link within 24 hours of it being sent. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-13 | | **Use Case Name:** Staff User Login |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Staff Users | |
| **Description** | Staff user logging into their account | |
| **Trigger** | Clicking the “Login” button on the login page | |
| **Preconditions** | 1. User has previously made a staff account 2. User is currently on the login page 3. User’s account has been activated | |
| **Postconditions** | The user is logged in and redirected to the dashboard | |
| **Normal Flow** | 1. User enters their email and password into their respective fields 2. User clicks the “login button” 3. User is shown a message to alert them that they have successfully signed 4. User is redirected to the dashboard page | |
| **Alternative Flow** | N/A | |
| **Exceptions** | * If the email or password fields are left empty when the user presses the sign in button, they will be given an error message and prompted to enter the missing field. * If the email and password do not match, an error message will be shown, and the user will be prompted to try again. | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-15 | | **Use Case Name:** Staff Availability Change |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/3/2022 |
| **Actors** | Staff Users | |
| **Description** | Staff user updating the days and times they are available to work | |
| **Trigger** | Clicking the “Availability” button on the left navigation | |
| **Preconditions** | The user is logged into a staff account | |
| **Postconditions** | Staff Availability is updated to reflect the times inputted by the staff user | |
| **Normal Flow** | 1. User navigates to the Availability page 2. User clicks add button 3. User enters new availability 4. User clicks the “Confirm” button | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If user enters an end time that is after the start time (for example, start time is 4pm and end time is 11am), an error will be shown prompting the user to reenter the times | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-16 | | **Use Case Name:** Staff Request Off |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Staff Users | |
| **Description** | Staff user sending in a request for a specific day off | |
| **Trigger** | Clicking the “Schedule” button on the left navigation | |
| **Preconditions** | The user is logged into a staff account | |
| **Postconditions** | Request is sent to manager to approve or deny | |
| **Normal Flow** | 1. User navigates to the availability page 2. Clicks the “Request Day Off” button 3. Dialog box that asks for the date they would like open 4. User inputs date and hits “Request” button 5. Request is sent to the business user 6. UC-4 | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If the user enters a date that has already passed, an error message will show. | |
| **Assumptions** | N/A | |

### 4.1.3 Both User Type Use Cases

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-17 | | **Use Case Name:** Announcement Viewing |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Staff Users, and Business Users | |
| **Description** | User navigating any previously made announcements | |
| **Trigger** | User clicks on Announcements tab on left navigation menu | |
| **Preconditions** | 1. At least one announcement was made as described in UC-5 2. User is signed into an account | |
| **Postconditions** | The user can view announcements | |
| **Normal Flow** | 1. User navigates to the announcements page from the side navigation 2. Sees the most recent announcement posted at the top of the screen 3. User scrolls down to view past announcements | |
| **Alternative Flow** | N/A | |
| **Exceptions** | N/A | |
| **Assumptions** | The business user has made at least one announcement post to view | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-18 | | **Use Case Name:** Message Board |
| **Created By:** Matt Korte | | **Date Created:** 2/25/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Staff Users or Business Users | |
| **Description** | Making and viewing posts on the message board | |
| **Trigger** | User clicks on Messages tab on left navigation | |
| **Preconditions** | User is signed into an account | |
| **Postconditions** | Message is sent to the message board and viewable for all other members of the organization. | |
| **Normal Flow** | 1. User navigates to the messages page from the side navigation menu 2. User enters a message into the text input box 3. User presses the “Send” button 4. The message is shown in the message feed, along with the name and profile picture of the user as well as the day and time it was sent | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If the text input box is empty, the use will not be able to enter send an empty message. | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-19 | | **Use Case Name:** Dashboard Navigation |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Business Users or Staff Users | |
| **Description** | User navigates to the dashboard page | |
| **Trigger** | User clicks on the dashboard tab on left navigation | |
| **Preconditions** | User is signed into an account | |
| **Postconditions** | User directed to dashboard | |
| **Normal Flow** | 1. User signs in 2. User is navigated to dashboard | |
| **Alternative Flow** | User logs in and lands on dashboard.  If a user cannot access the dashboard page, an error message will be displayed, and the user will be prompted to log in with their credentials. | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-20 | | **Use Case Name:** User Logout |
| **Created By:** Matt Korte | | **Date Created:** 2/24/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Business Users or Staff Users | |
| **Description** | Users logging out of eTimely | |
| **Trigger** | Logout button | |
| **Preconditions** | User is signed into an account | |
| **Postconditions** | User is signed out and redirected to home page | |
| **Normal Flow** | 1. User navigates to the settings page 2. User clicks the “Logout” button 3. User is no longer signed into eTimely and redirected to the home page | |
| **Alternative Flow** | N/A | |
| **Exceptions** | N/A | |
| **Assumptions** | N/A | |

### 4.1.4 Default Use Cases

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-21 | | **Use Case Name:** About Us Navigation |
| **Created By:** Matt Korte | | **Date Created:** 2/25/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Any user | |
| **Description** | User navigates to the about us page | |
| **Trigger** | The “About Us” link in the navigation bar | |
| **Preconditions** | 1. User is not signed into an account. 2. The navigation bar is visible on the page the user is currently on | |
| **Postconditions** | User is on the About Us page | |
| **Normal Flow** | 1. User clicks the “About Us” link in the navigation bar 2. User is redirected to the “About Us” page | |
| **Alternative Flow** | N/A | |
| **Exceptions** | If no user signed in and they navigate to the dashboard from the address bar, then they will be shown a screen telling them to sign in as well as a button to redirect them to the sign-up page | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-22 | | **Use Case Name:** Terms & Conditions Navigation |
| **Created By:** Matt Korte | | **Date Created:** 2/25/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Any user | |
| **Description** | User navigates to the terms and conditions page | |
| **Trigger** | The “Terms and Conditions” link in the footer | |
| **Preconditions** | 1. User is not signed into an account 2. The footer is visible on the page the user is currently on | |
| **Postconditions** | User is on the Terms & Conditions page | |
| **Normal Flow** | 1. User clicks the “Terms & Conditions” link in the footer 2. User is redirected to the Terms & Conditions page | |
| **Alternative Flow** | N/A | |
| **Exceptions** | N/A | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-23 | | **Use Case Name:** Home Navigation |
| **Created By:** Matt Korte | | **Date Created:** 2/25/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Any user | |
| **Description** | User navigates to the Home page | |
| **Trigger** | The “Home” link in the navigation bar | |
| **Preconditions** | 1. User is not signed into an account 2. The navigation bar is visible on the page the user is currently on | |
| **Postconditions** | User is on the Home page | |
| **Normal Flow** | 1. User clicks the “Home” link in the navigation bar 2. User is redirected to the Home page | |
| **Alternative Flow** | 1. User clicks the “Home” link in the footer 2. User is redirected to the Home page | |
| **Exceptions** | N/A | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-24 | | **Use Case Name:** Sign In Navigation |
| **Created By:** Matt Korte | | **Date Created:** 2/25/2022 |
| **Last Updated By:** Matt Korte | | **Last Revision Date:** 3/4/2022 |
| **Actors** | Any user | |
| **Description** | User navigates to the sign in page | |
| **Trigger** | The “Sign In” button in the navigation bar | |
| **Preconditions** | 1. User is not signed into an account 2. The navigation bar is visible on the page the user is currently on | |
| **Postconditions** | User is on the Sign In page | |
| **Normal Flow** | 1. User clicks the “Sign In” button in the navigation bar 2. User is redirected to the Sign In page | |
| **Alternative Flow** | If the user manual enters the link to any of the pages that require an account to access, they will be shown a screen that has a link to the Sign In page | |
| **Exceptions** | N/A | |
| **Assumptions** | N/A | |

|  |  |  |
| --- | --- | --- |
| **Use Case ID:** UC-25 | | **Use Case Name:** Sign Up Navigation |
| **Created By:** Matt Korte | | **Date Created:** 2/25/2022 |
| **Last Updated By:** Samia Chowdhury | | **Last Revision Date:** 3/1/2022 |
| **Actors** | Any user | |
| **Description** | User navigates to the sign in page | |
| **Trigger** | The “Get Started” button in the navigation bar | |
| **Preconditions** | The user is on the home page | |
| **Postconditions** | User is on the Sign-Up page | |
| **Normal Flow** | 1. User clicks the “Get Stared” link on the home page 2. User is redirected to the Sign-Up page 3. UC-1 | |
| **Alternative Flow** | N/A | |

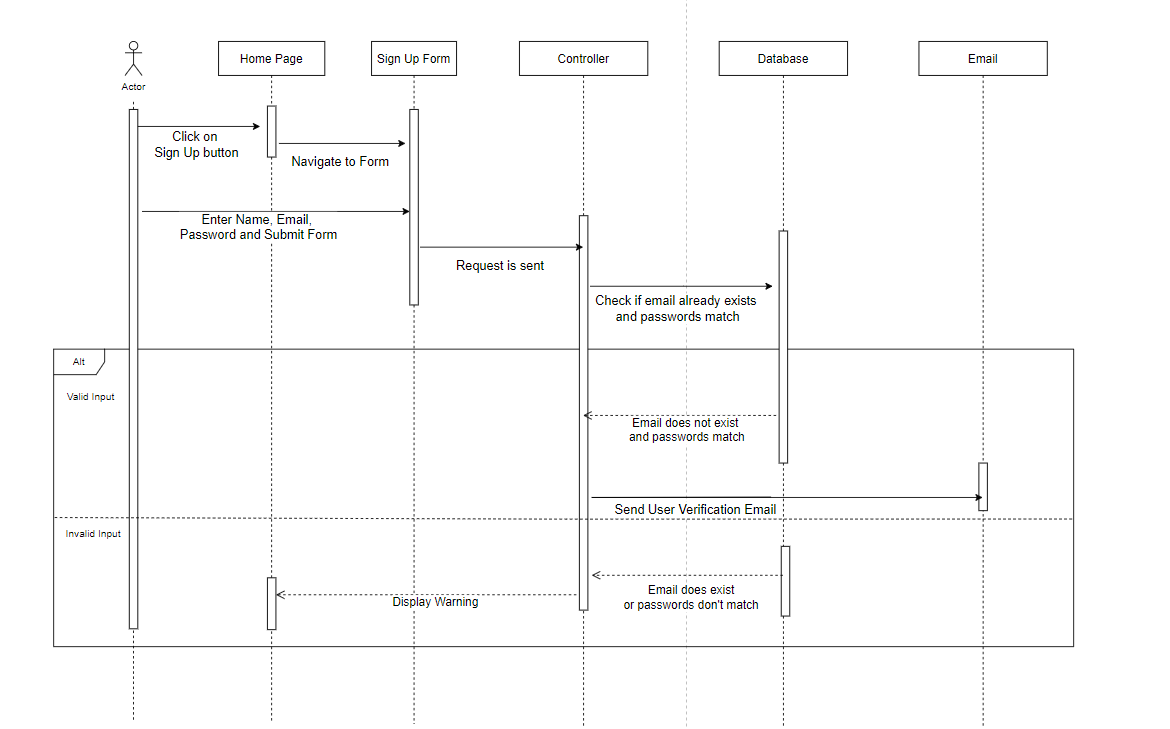
**4.2 Use Case Diagram:**



## 4.3 Sequence Diagram

The following subsections show five different sequence diagrams that will outline the steps taken by the user as well as the system to complete specific use case situations.

### 4.3.1 User Sign Up



### 4.3.2 Entering Availability

Diagram

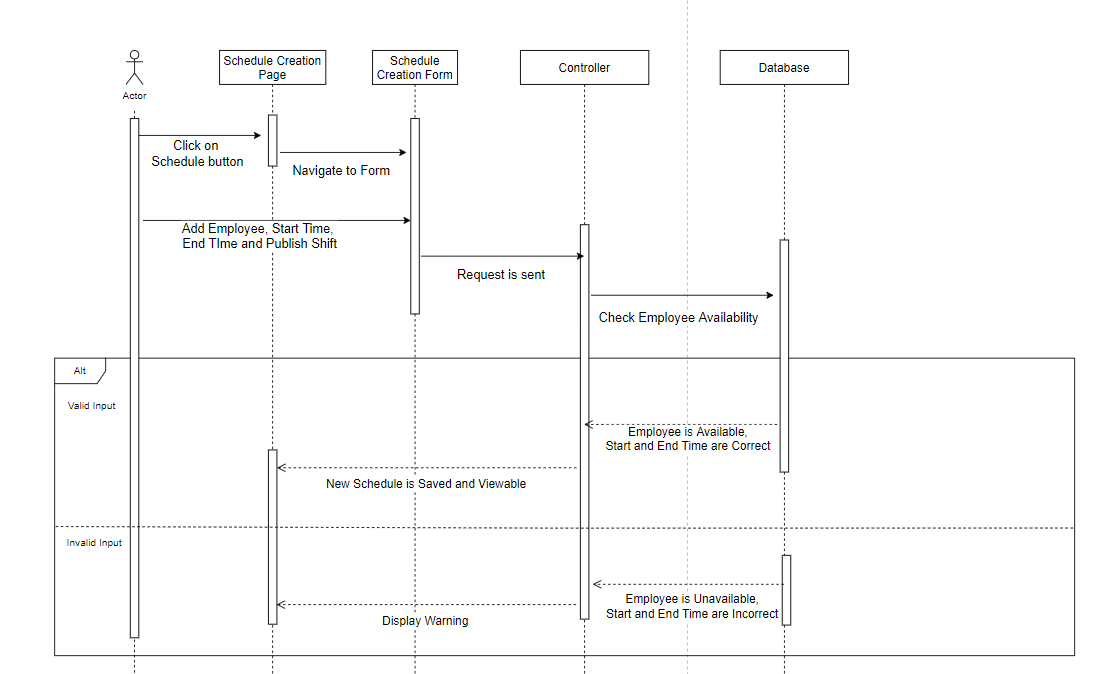
Description automatically generated

### 4.3.3 Confirmation Email

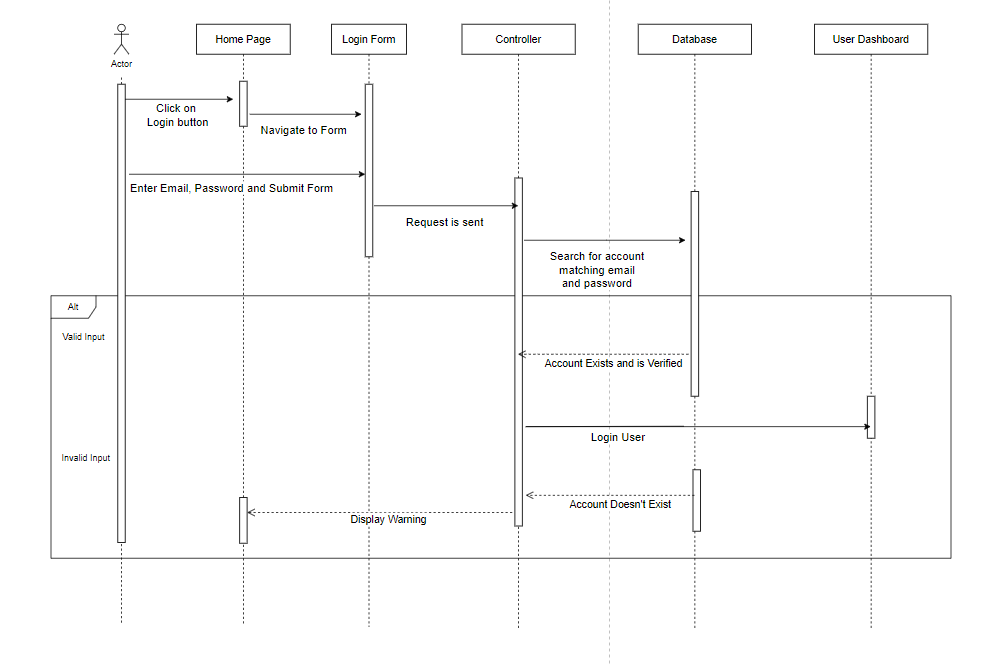
Diagram

Description automatically generated

**4.3.4 Schedule Creation**



**4.3.5 User Login**



## 4.4 Data Flow Diagrams

### 4.4.1 Business User Data Flow Diagram

Diagram, schematic

Description automatically generated**Figure 4.3.1**

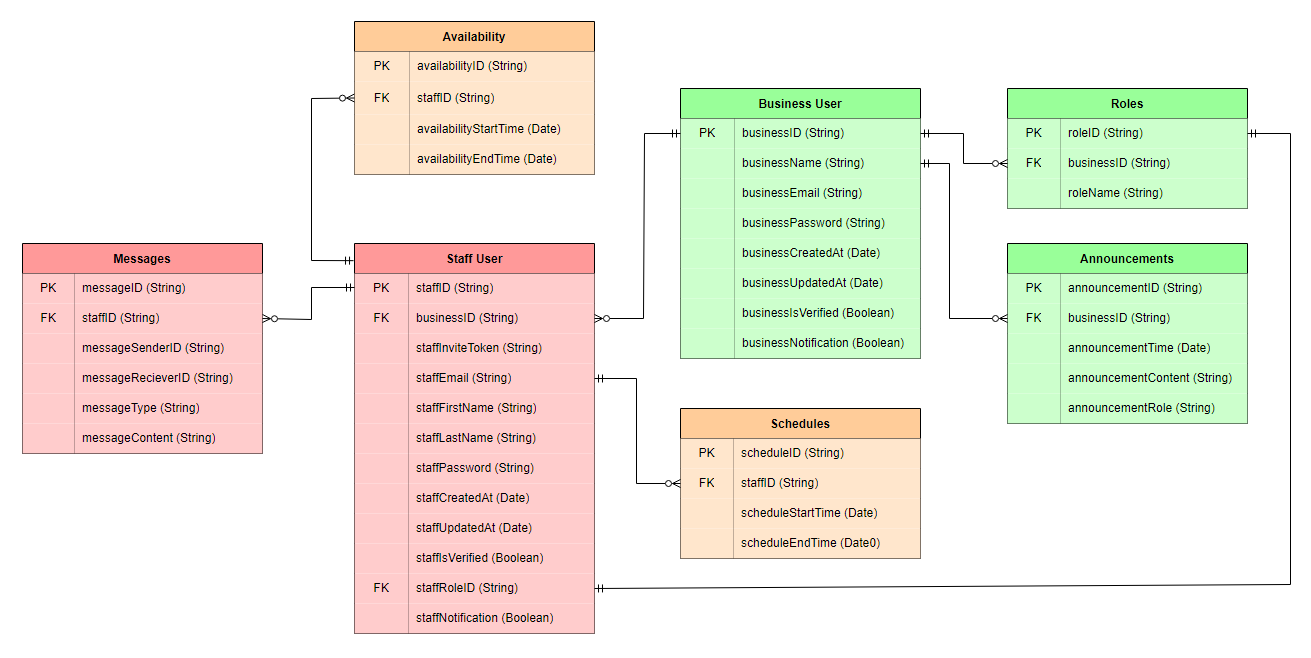
### 4.4.2 Staff User Data Flow Diagram

Diagram, schematic

Description automatically generated**Figure 4.3.2**

## 4.5 Database Design

This application is going to make use of the Firebase Firestore database in order to store all data that is used throughout the system. The diagram below shows the design behind the database and how data collections are going to be related. The Staff User, Business User, Availability and Schedules collections are going to be root collections in the database. The Roles and Announcements collections will be subcollections of the Business User root collection, while the Messages collection will be a subcollection of the Staff User root collection.

**Figure 4.4**

## 4.6 Application Program Interfaces

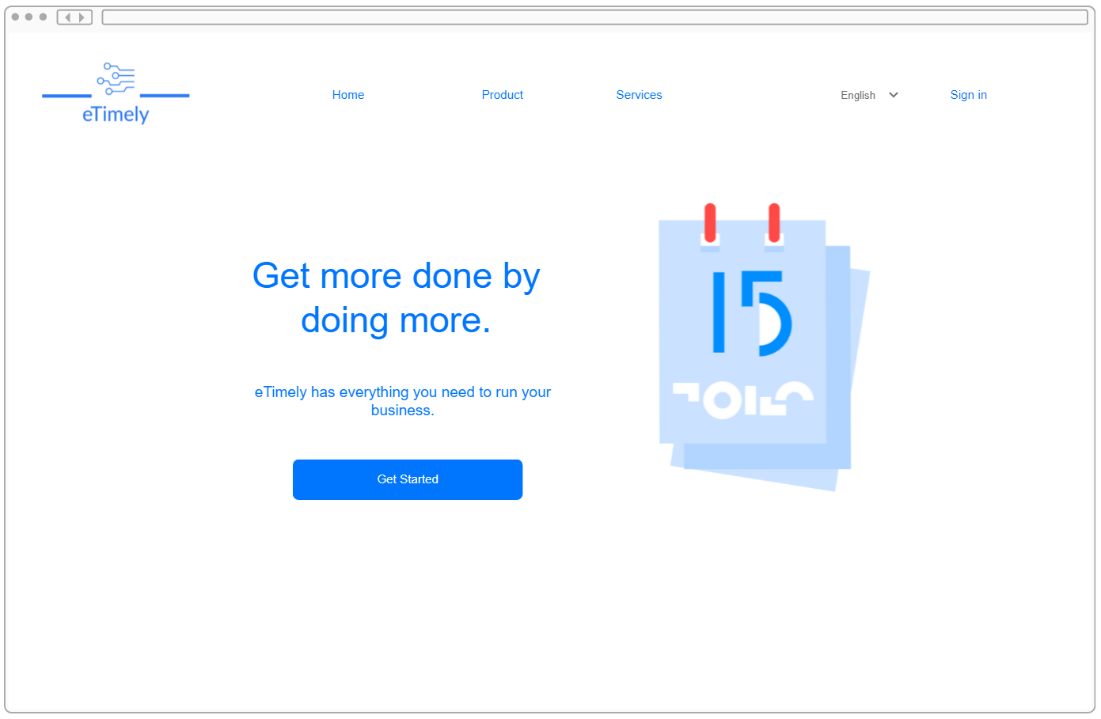
This application does not rely on any external API calls to gather or fetch information.

## 4.7 User Interface Design

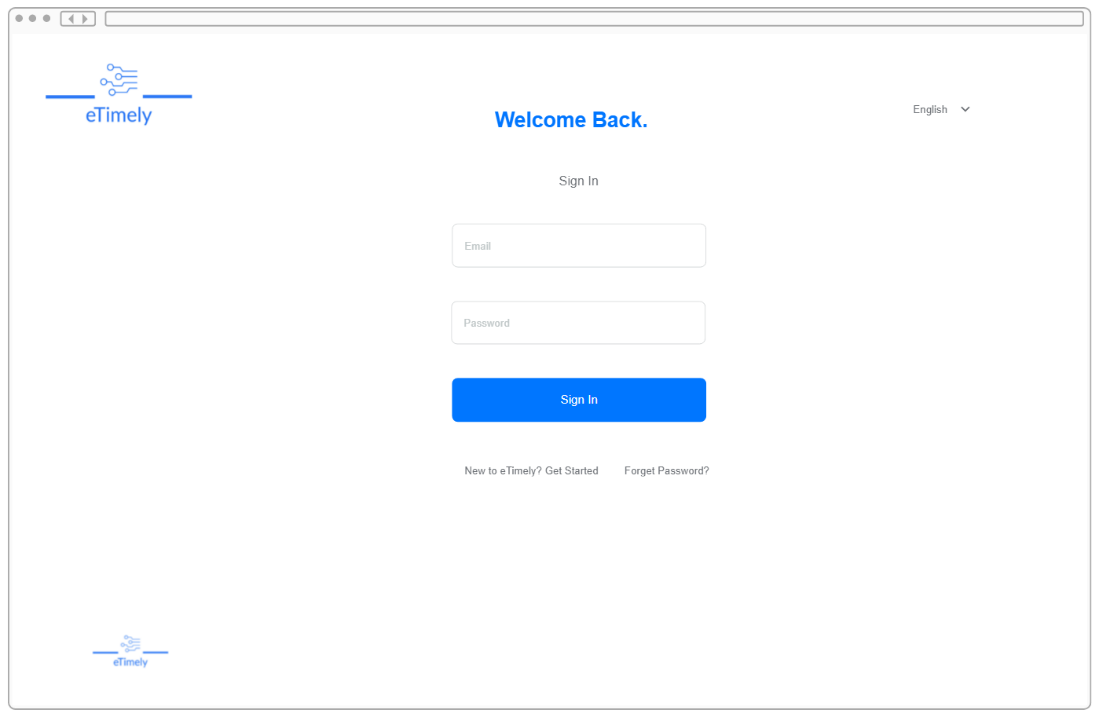
This application is going to have three distinct groups of user interfaces. The group of user interfaces that will be shown depends on whether the user is logged in. A user that is not logged in will only be able to navigate through the application’s Default View. If a user is logged in, the group of user interfaces presented to the user will depend on which type of account the user is logged in to. This is where the user will be able to take advantage of the functionalities of the application. These functionalities are highlighted by employees inputting availability and requesting schedule changes as well as the business user creating and publishing schedules and making decisions on schedule change requests.

### 4.7.1 Default View

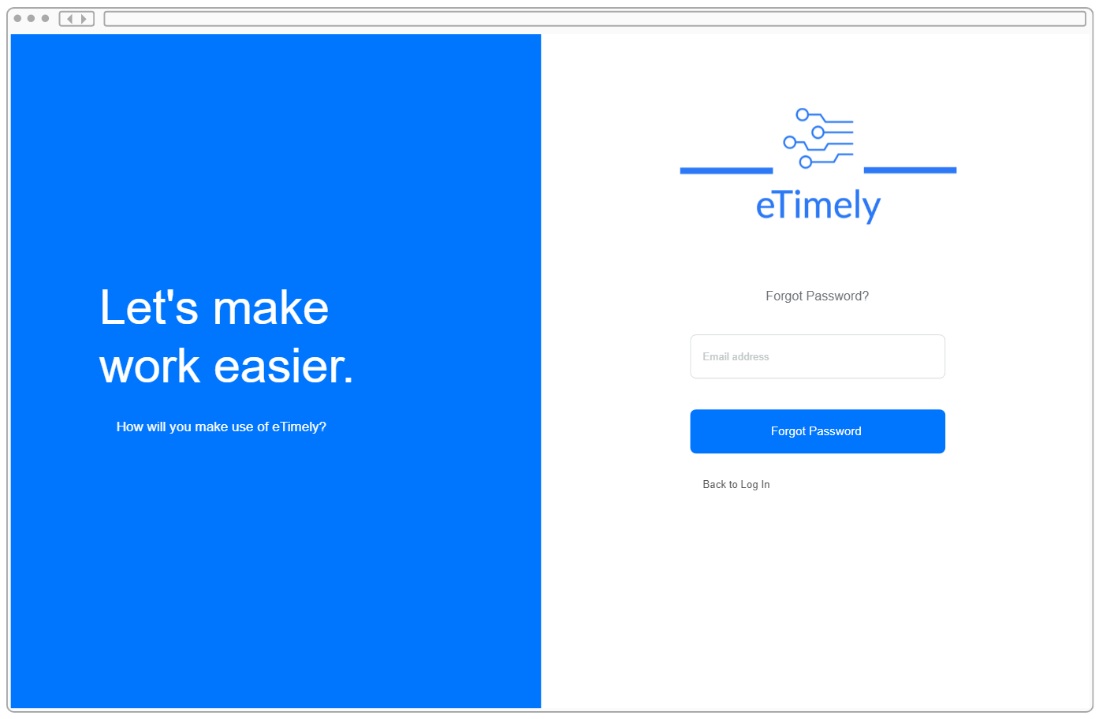
#### 4.7.1.1 Home Page



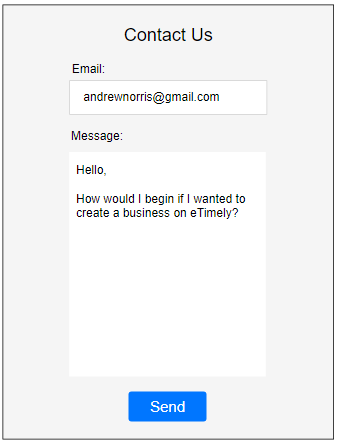
**4.7.1.2 Login Page**



**4.7.1.3 Password Recovery Page**

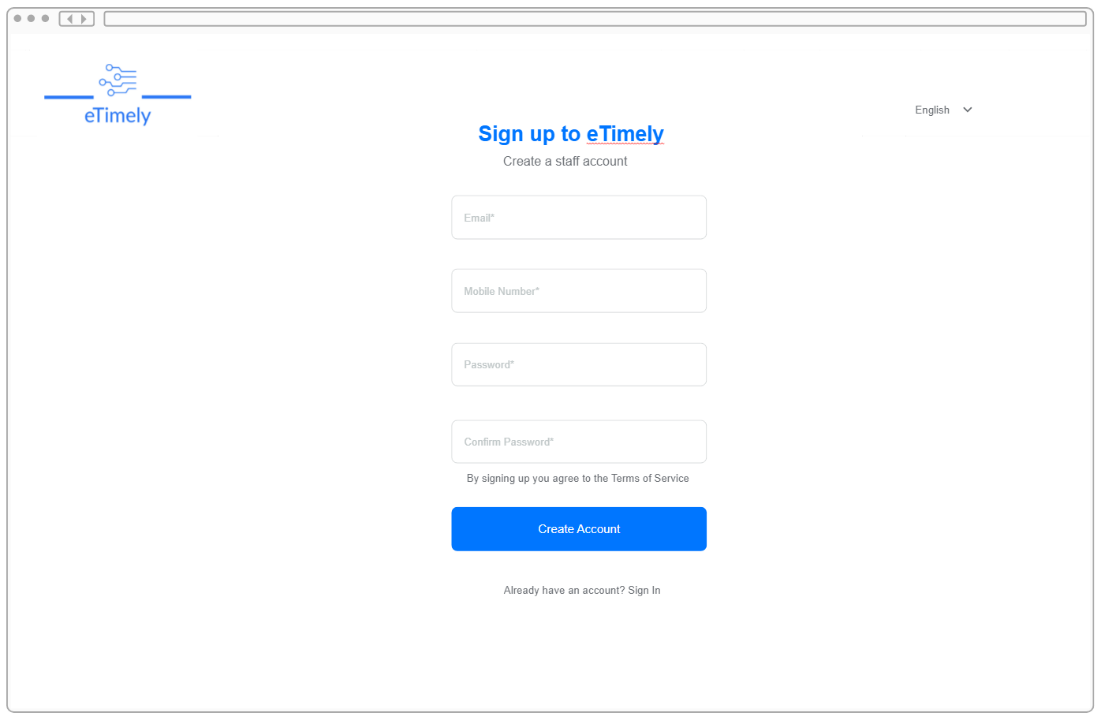


**4.7.1.4 Contact Us Window**

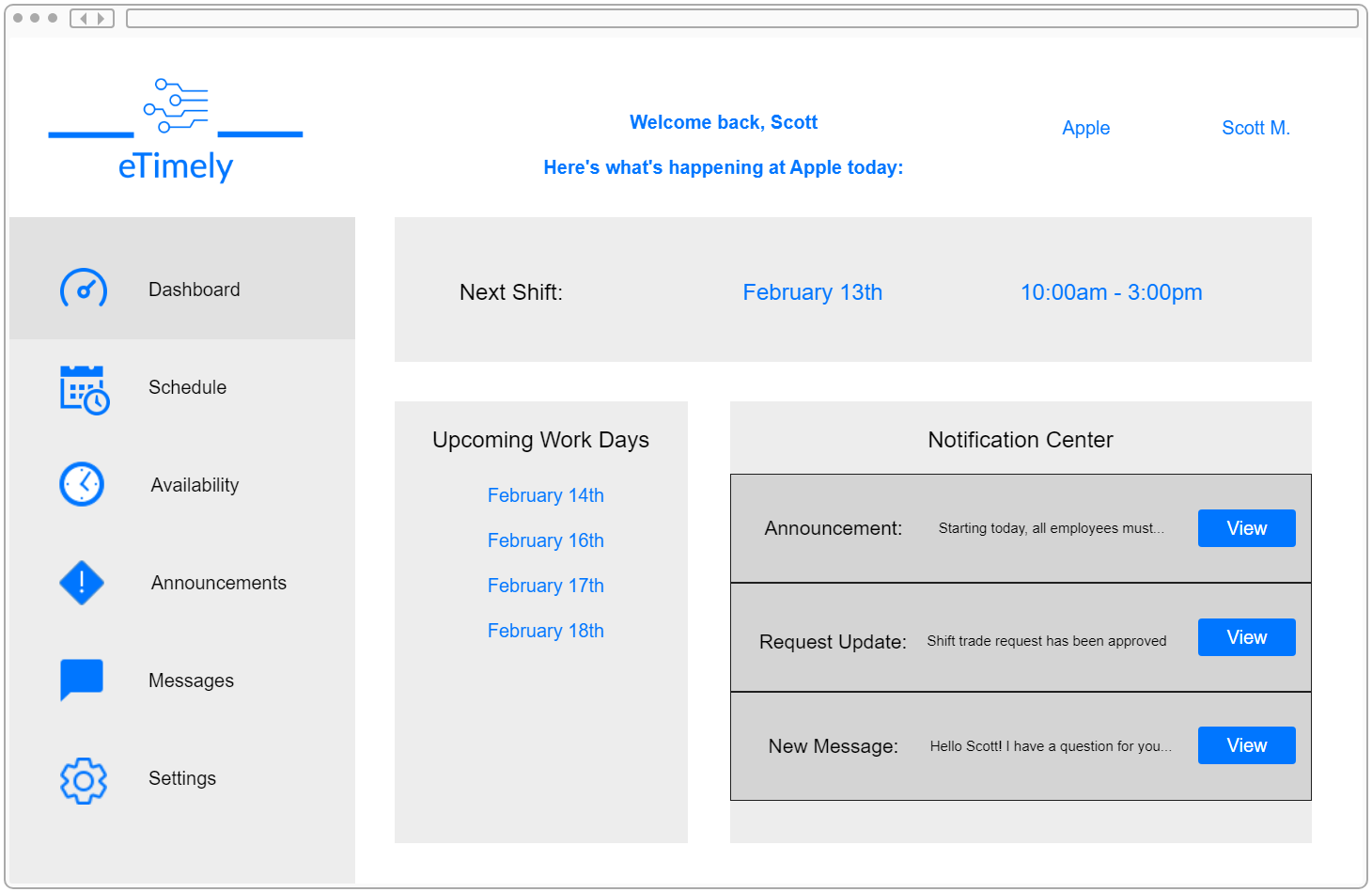


### 4.7.2 Staff View

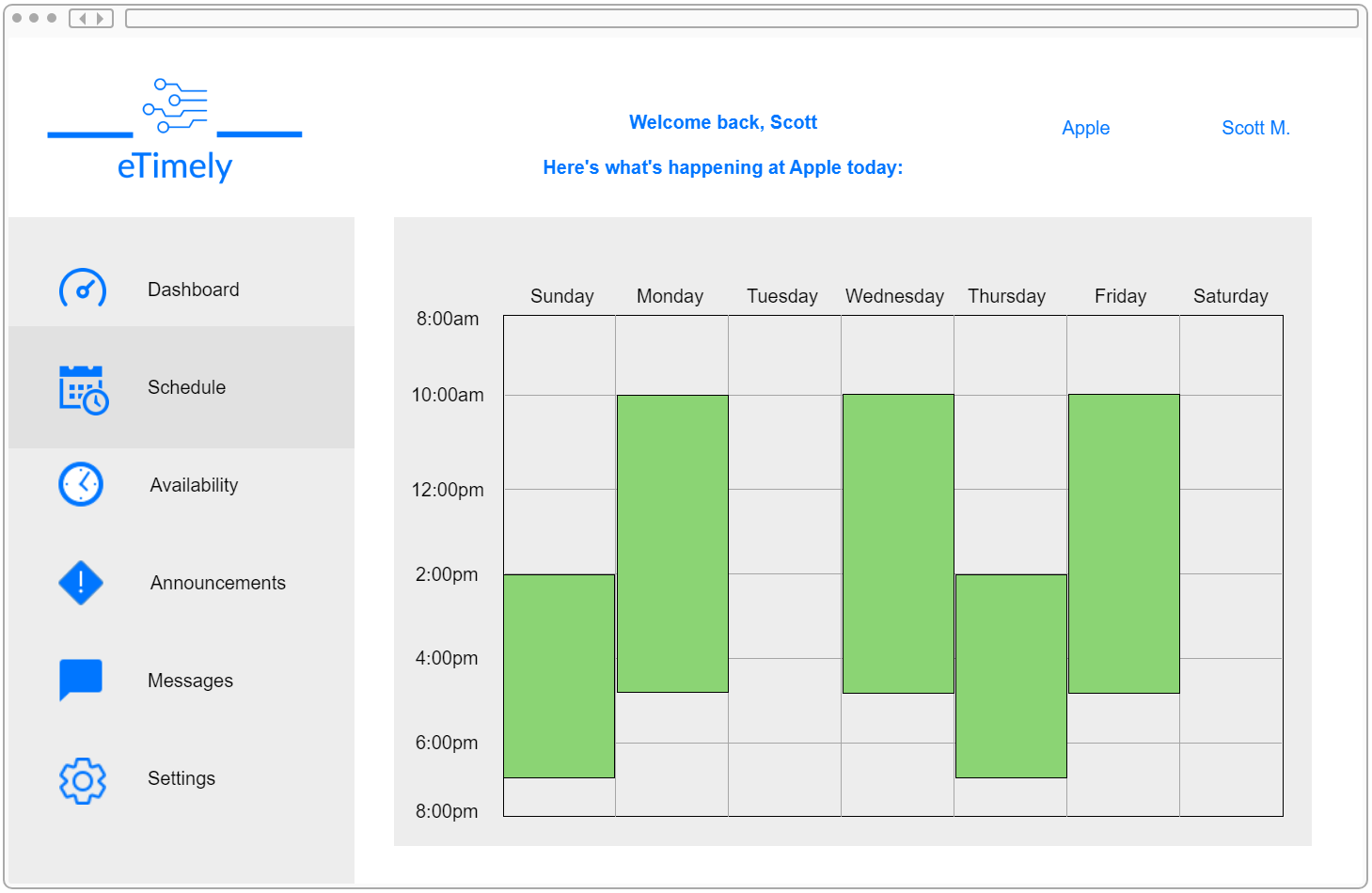
**4.7.2.1 Create Staff Account Page**



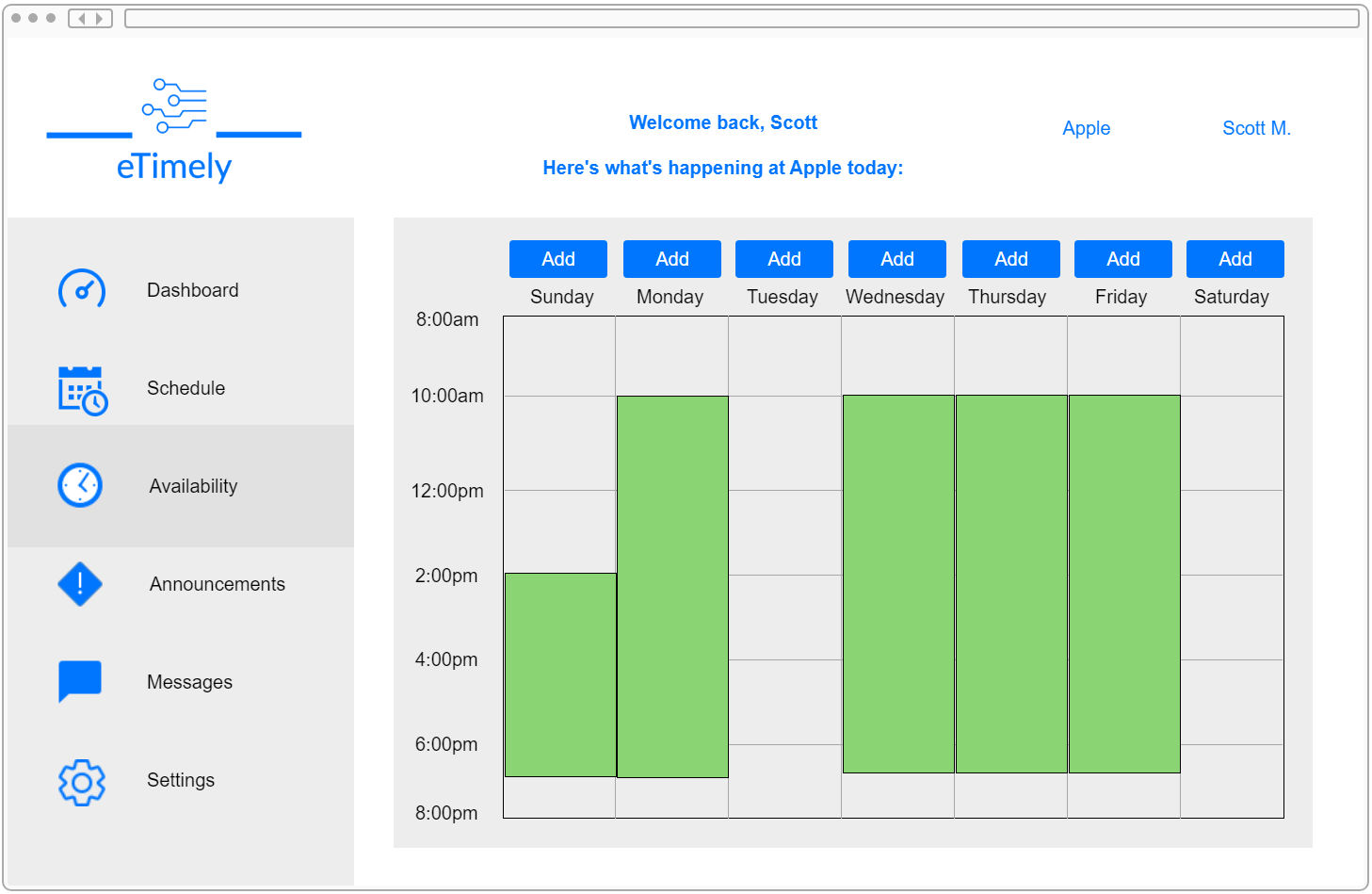
**4.7.2.2 Staff Dashboard View**



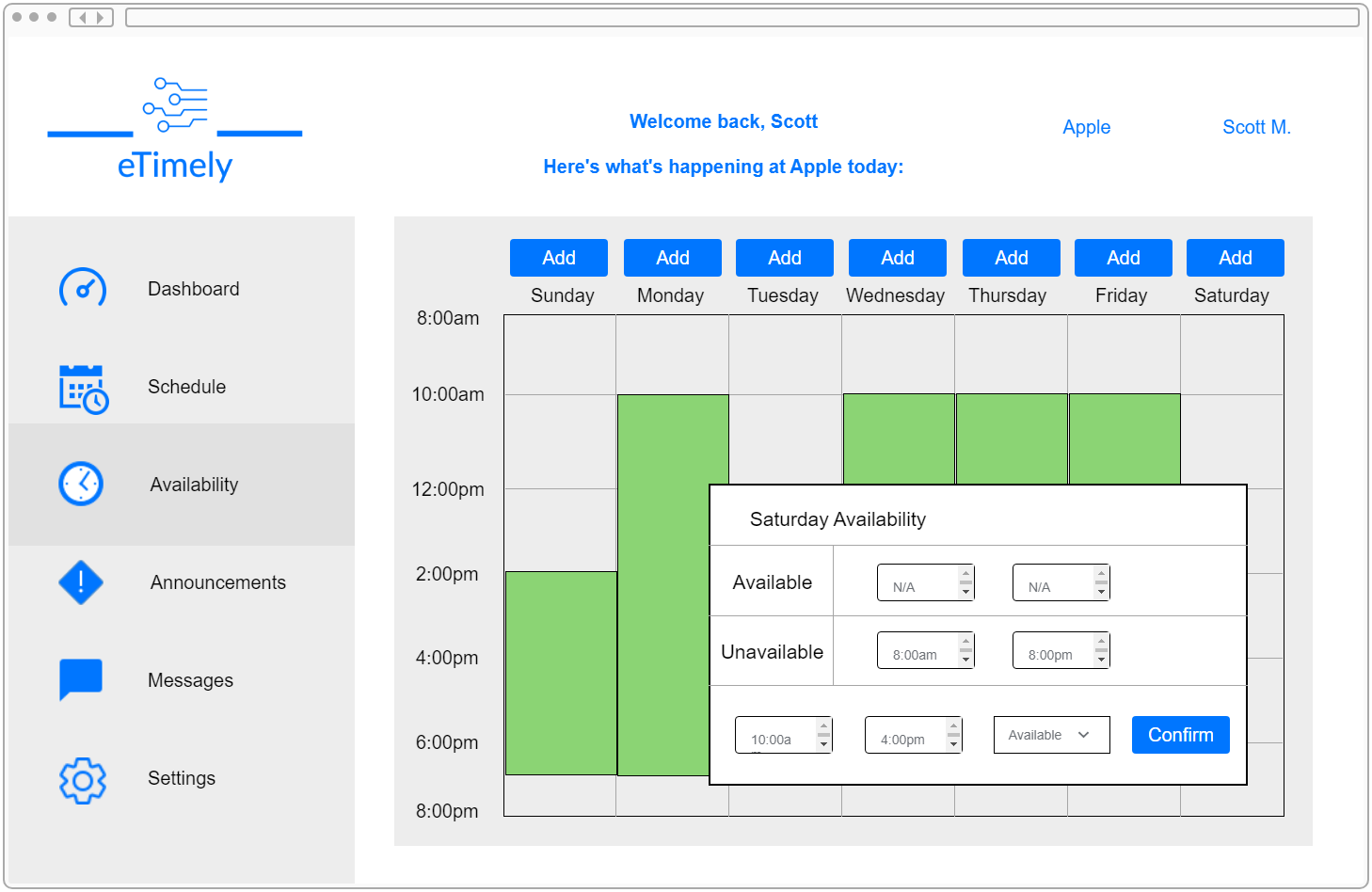
**4.7.2.3 Staff Schedule Tab**



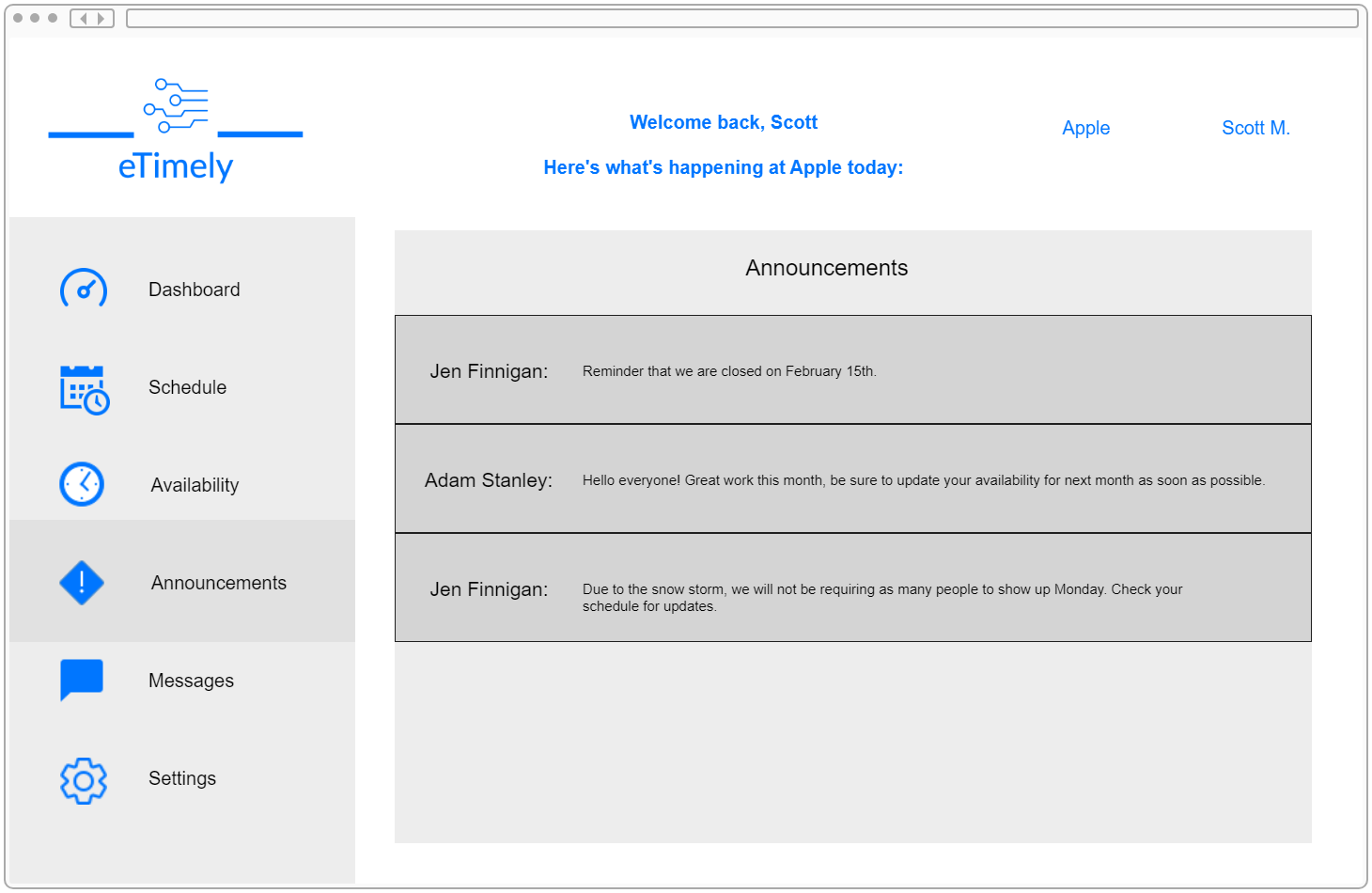
**4.7.2.4 Staff Availability Tab**



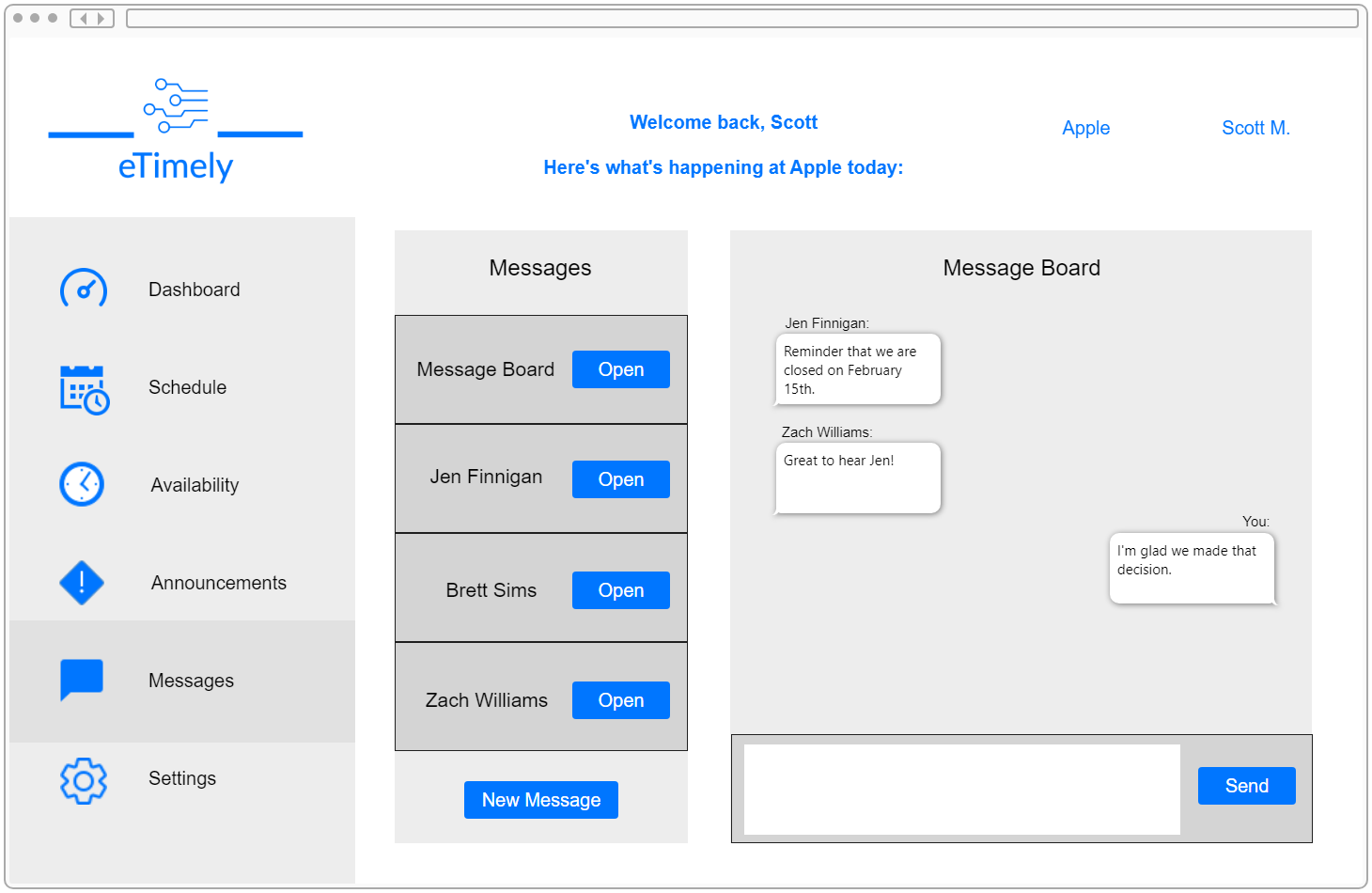
**4.7.2.5 Staff Availability Window**



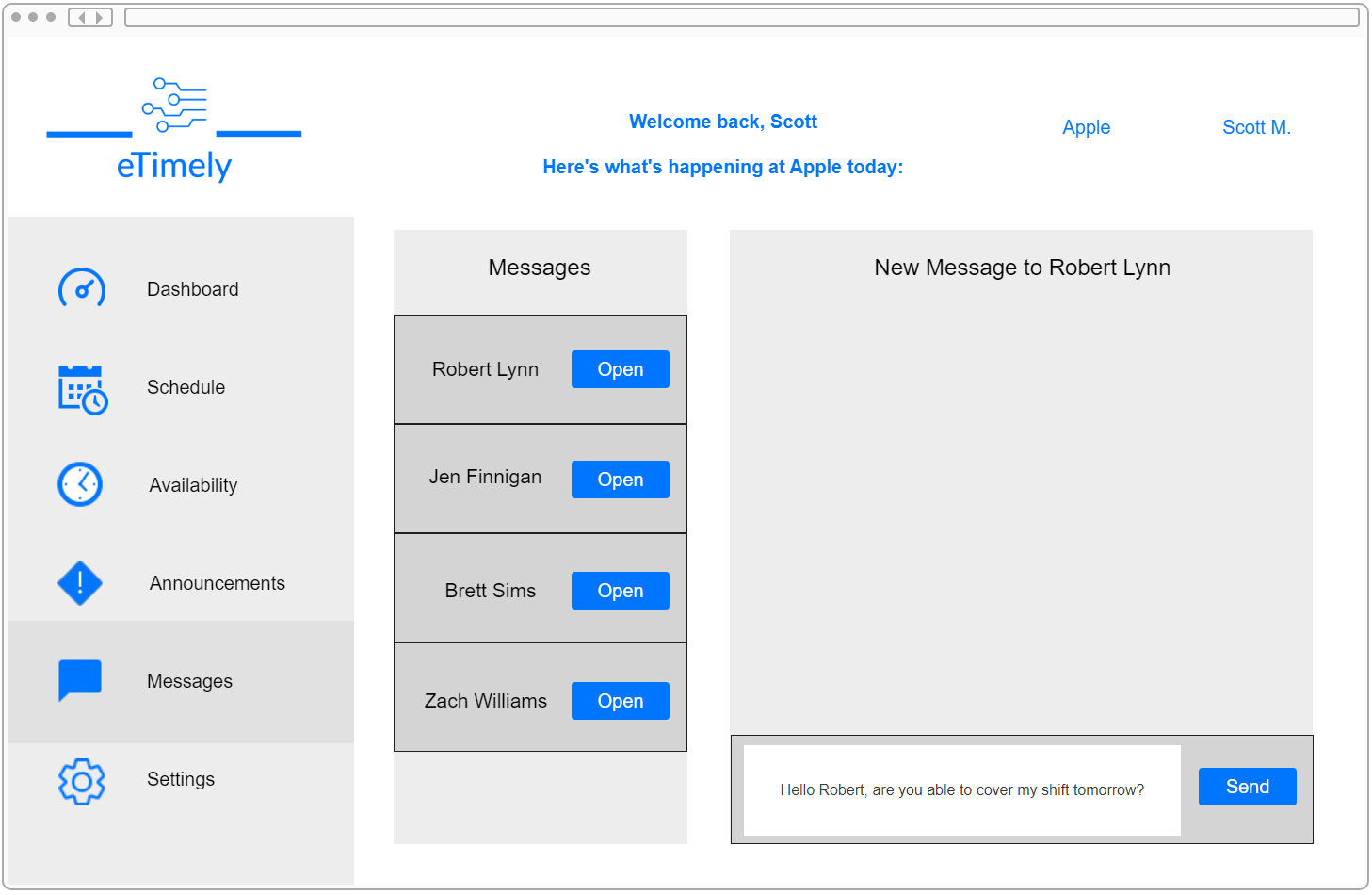
**4.7.2.6 Staff Announcements Tab**



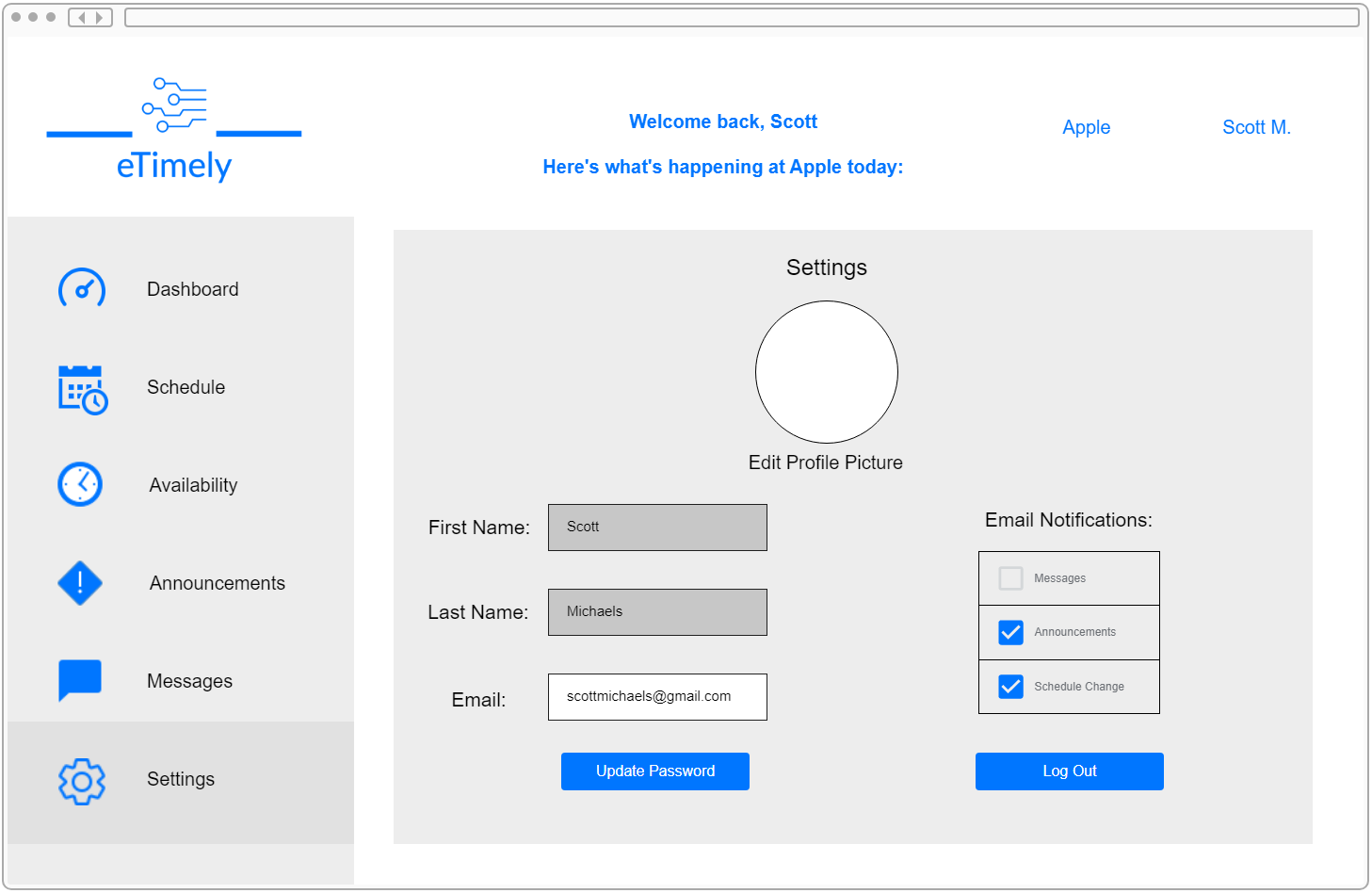
**4.7.2.7 Staff Messages Tab**



**4.7.2.8 New Message Window**

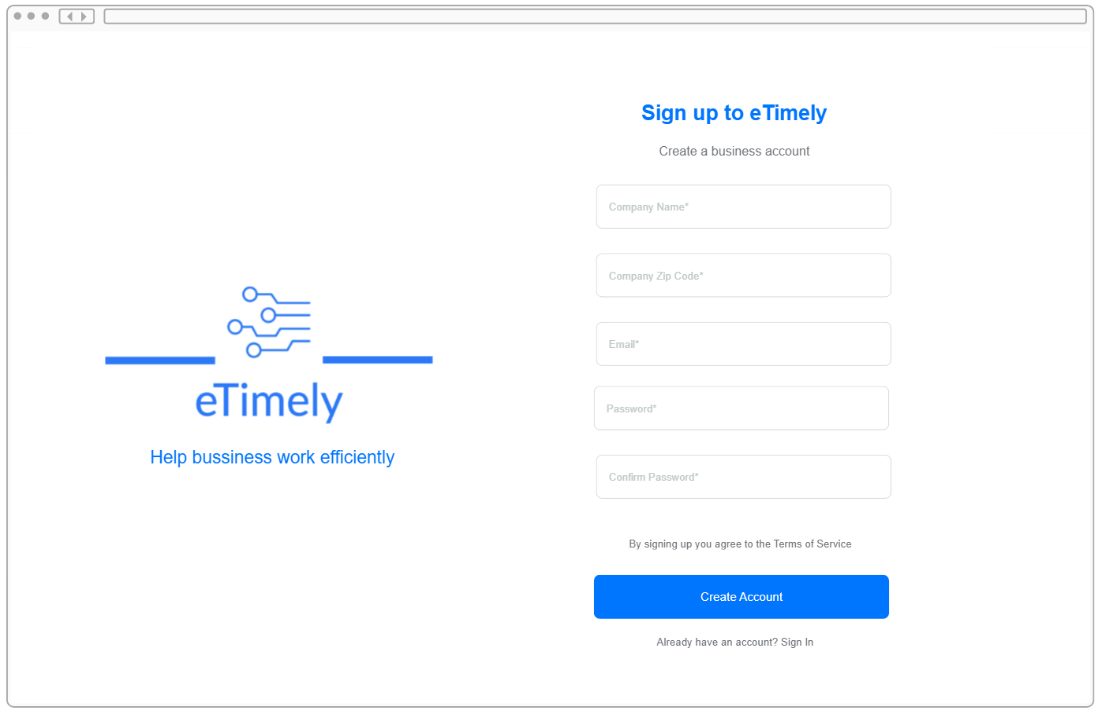


**4.7.2.9 Staff Settings Tab**

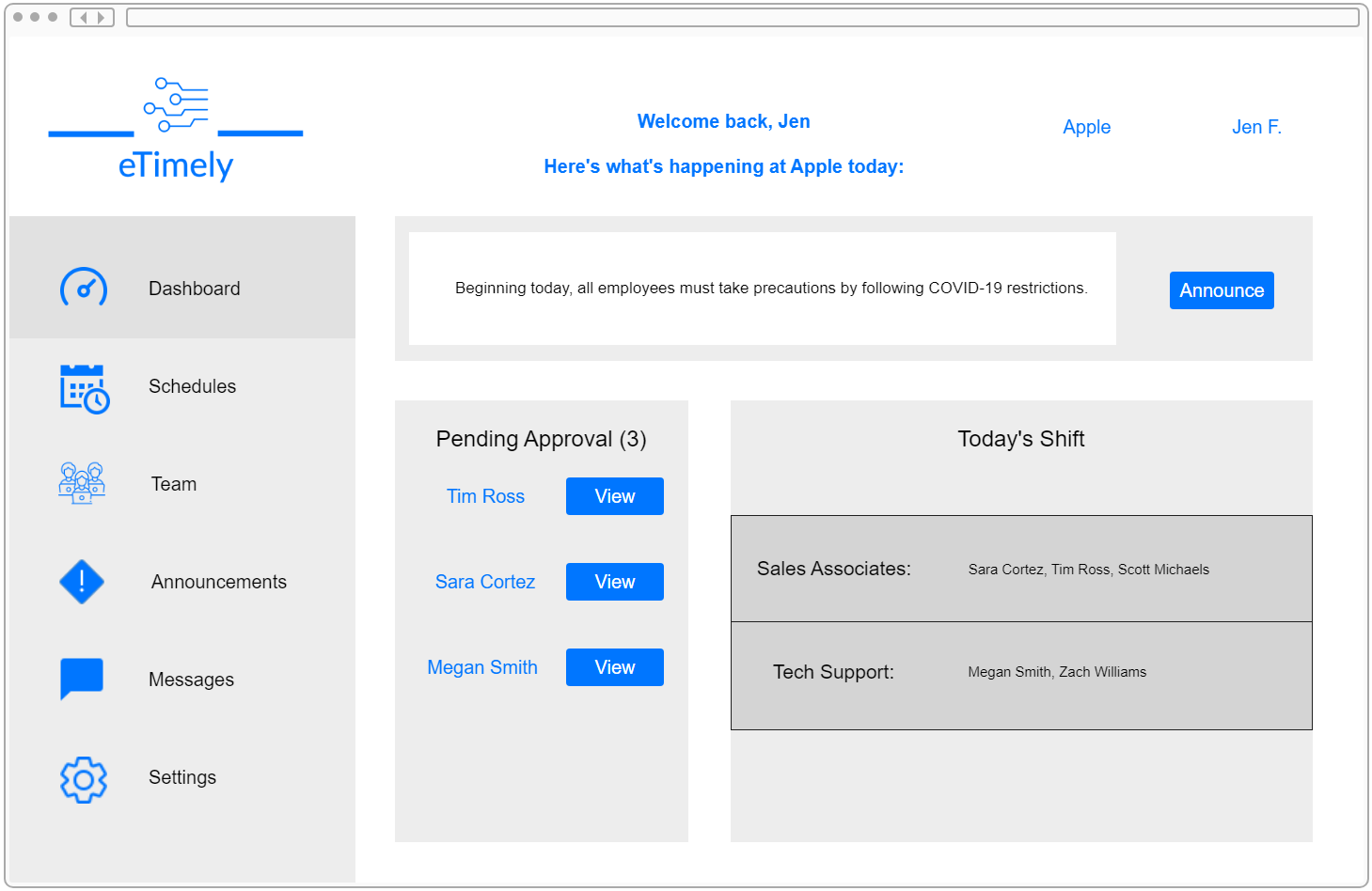


### 4.7.3 Business View

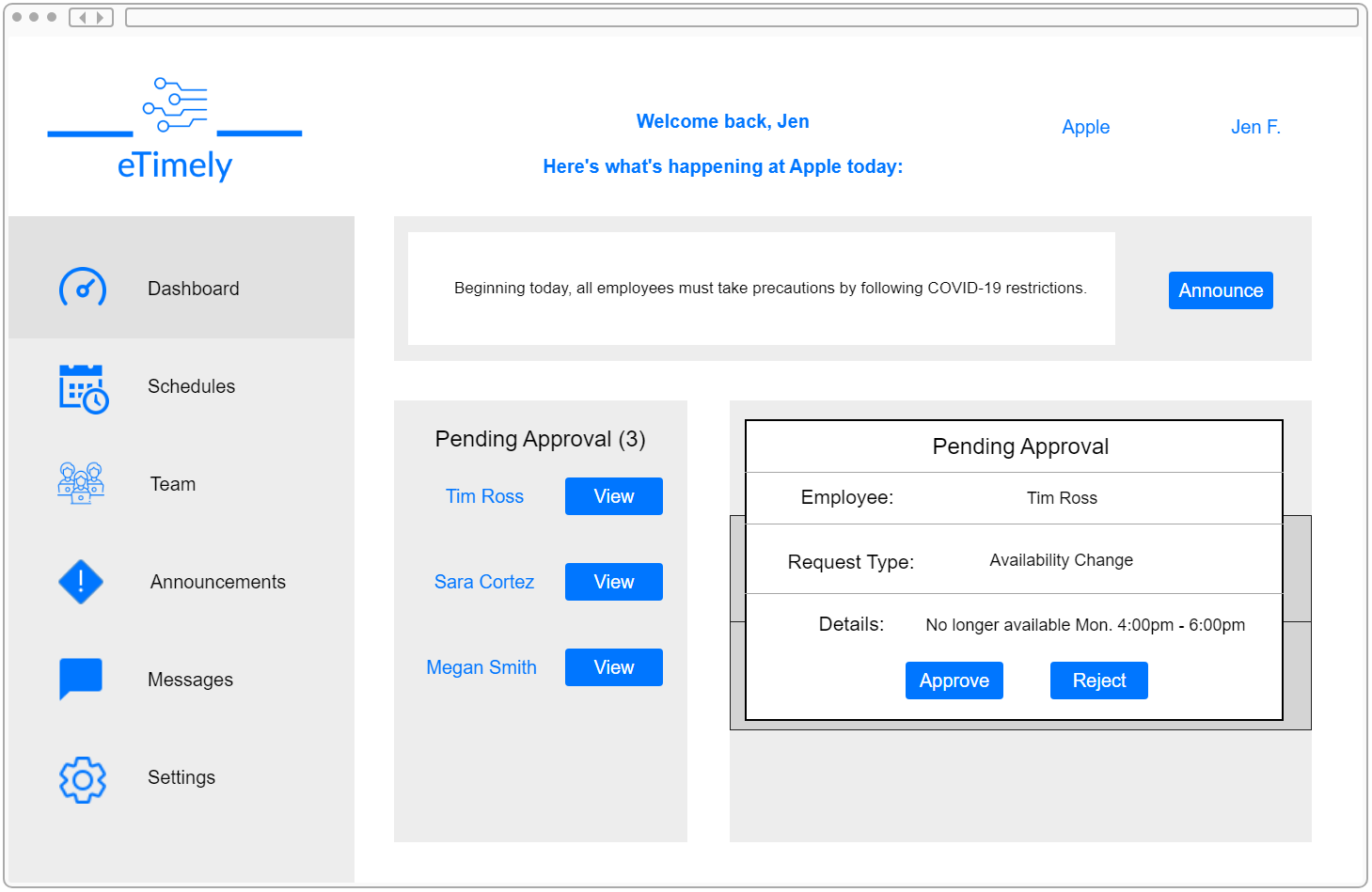
**4.7.3.1 Create Business Account Page**



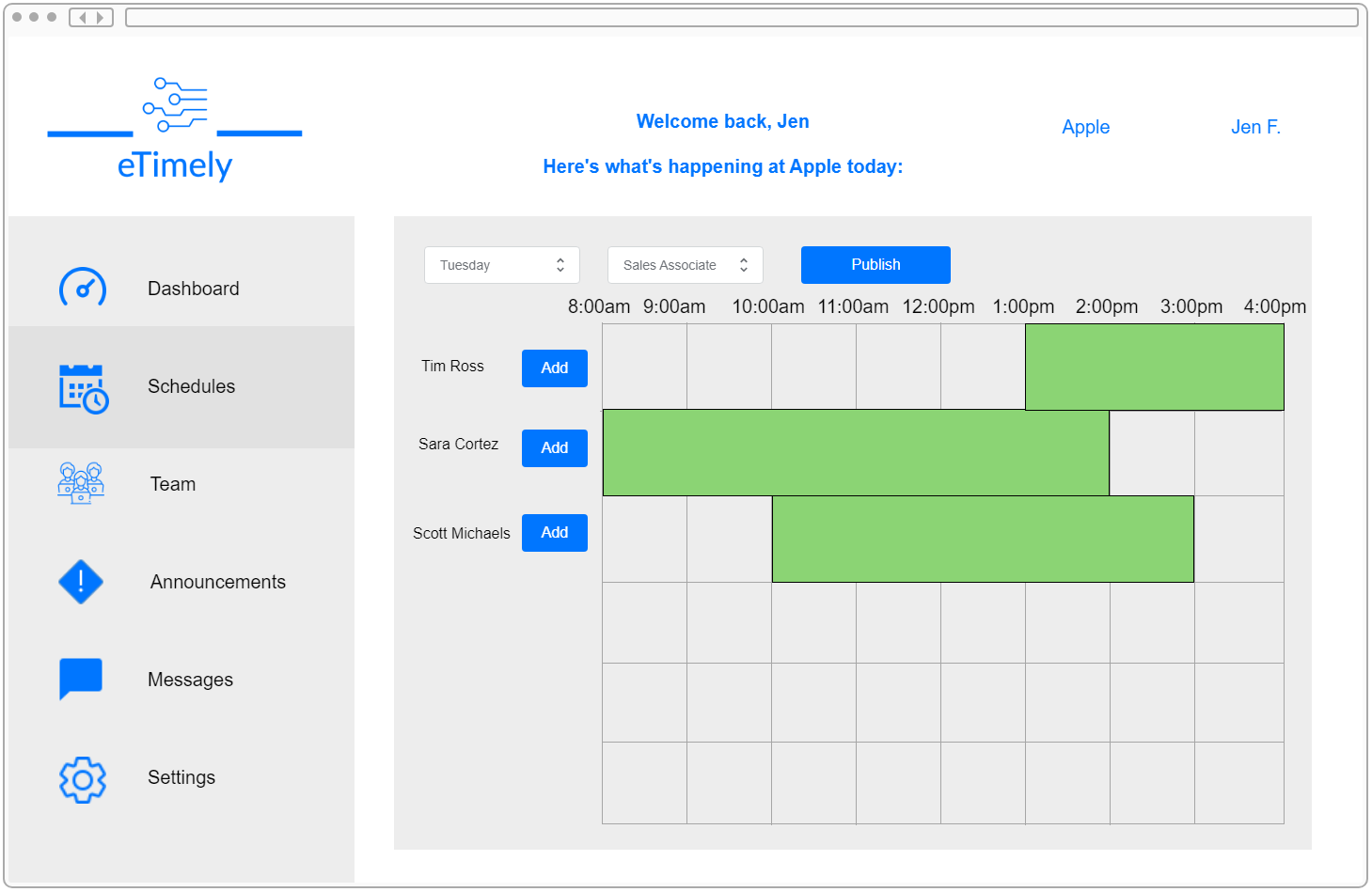
**4.7.3.2 Business Dashboard View**



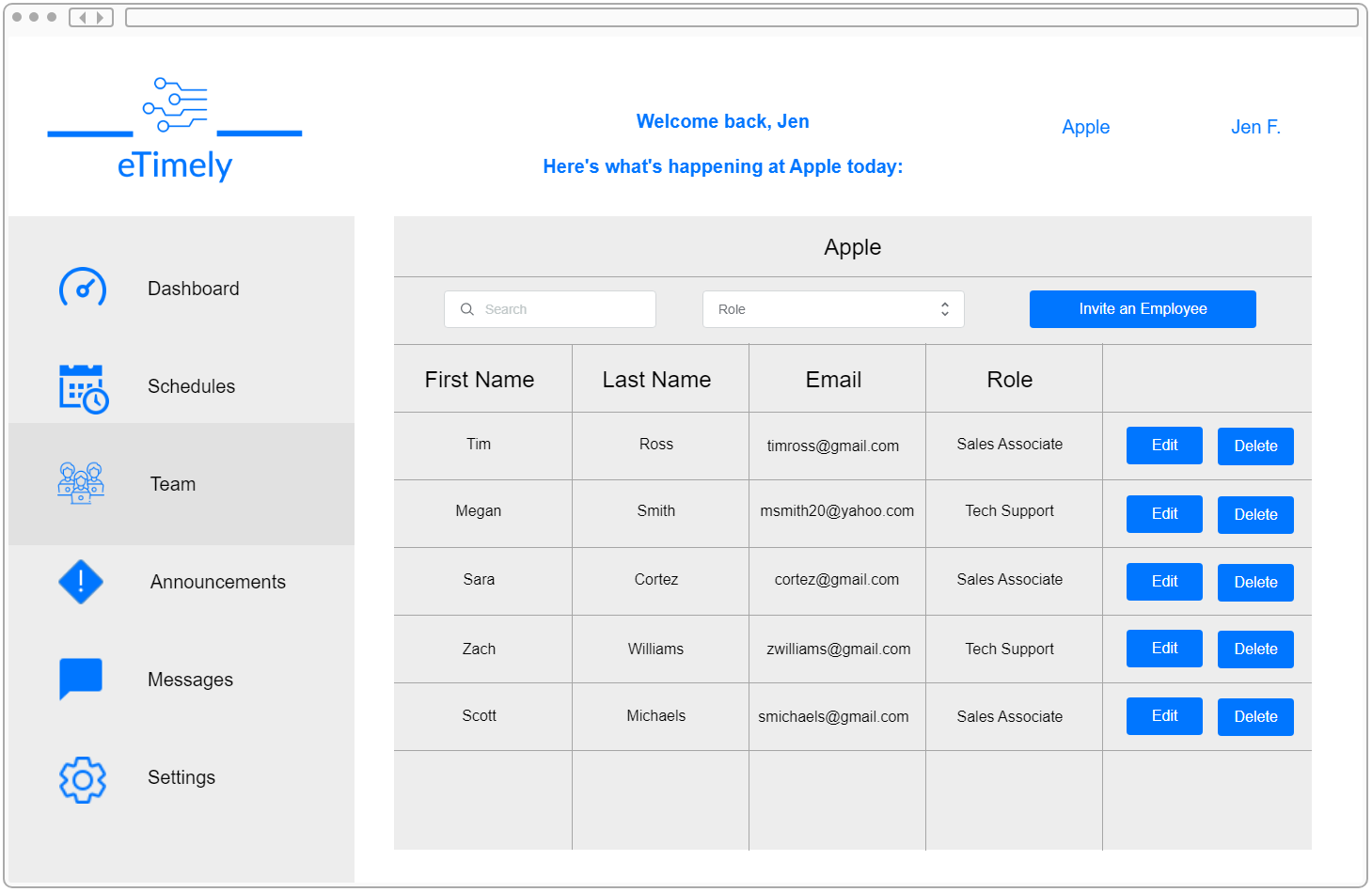
**4.7.3.3 Business Approval Window**



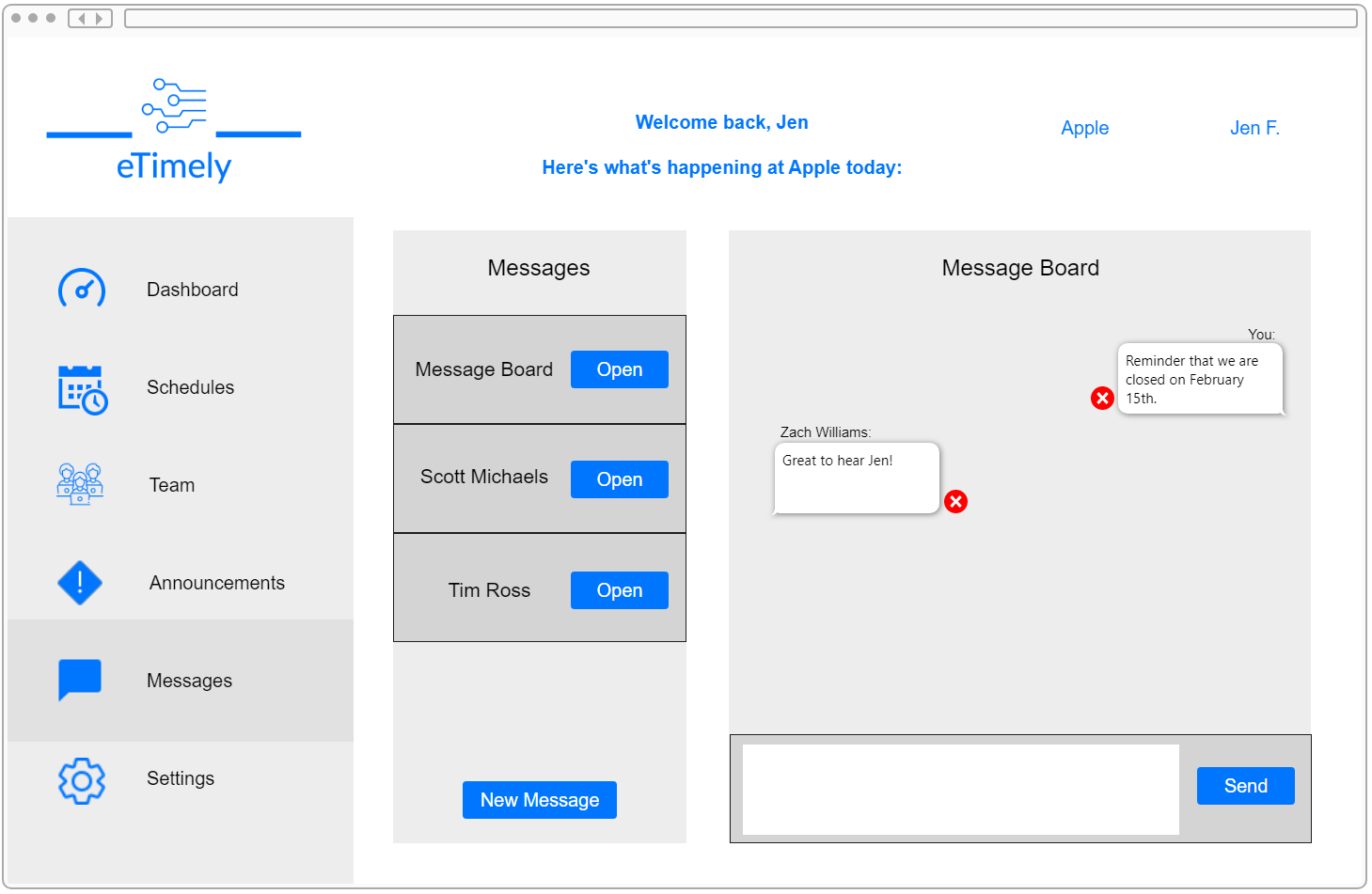
**4.7.3.4 Business Schedules Tab**



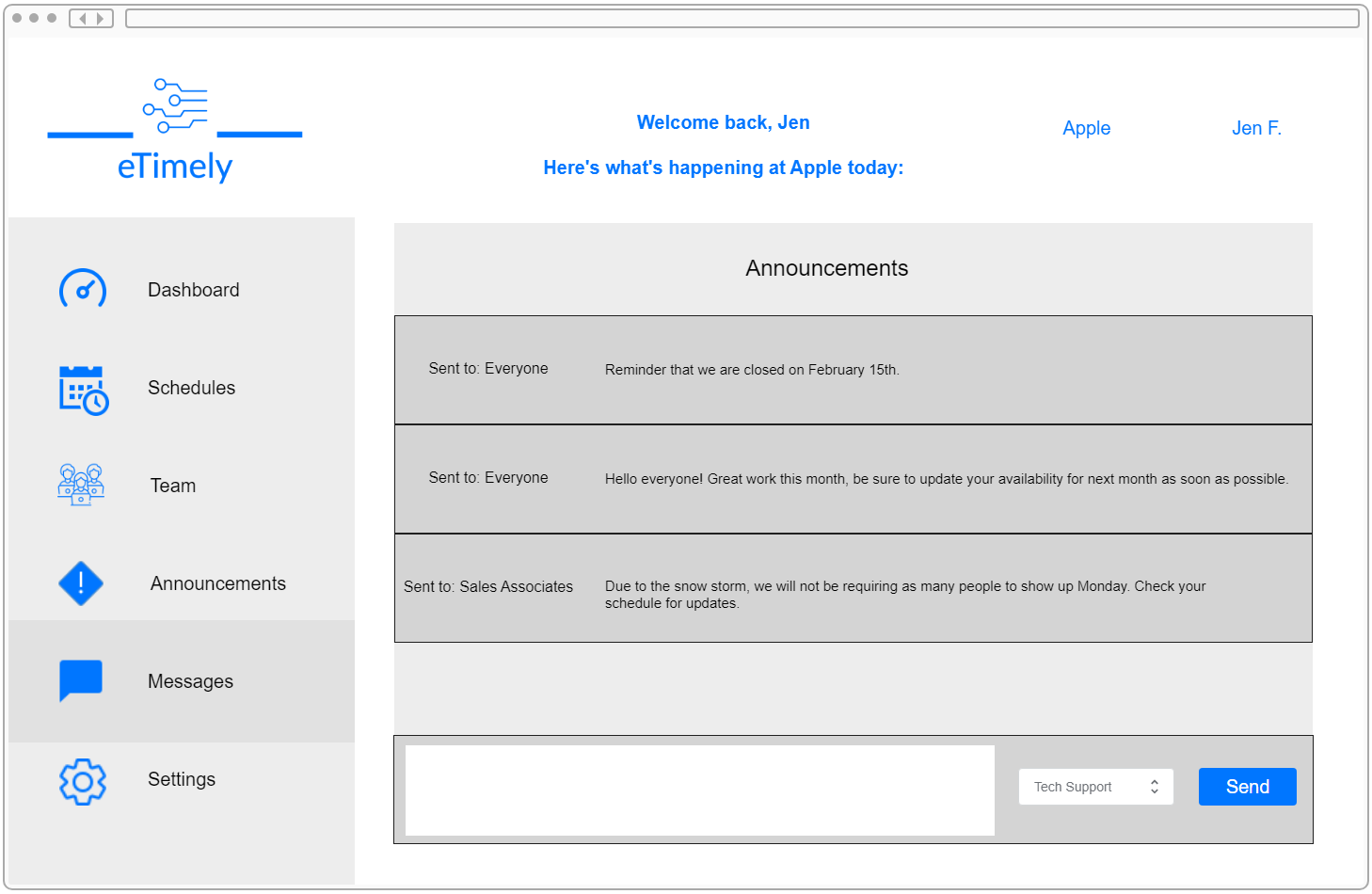
**4.7.3.5 Business Team Tab**



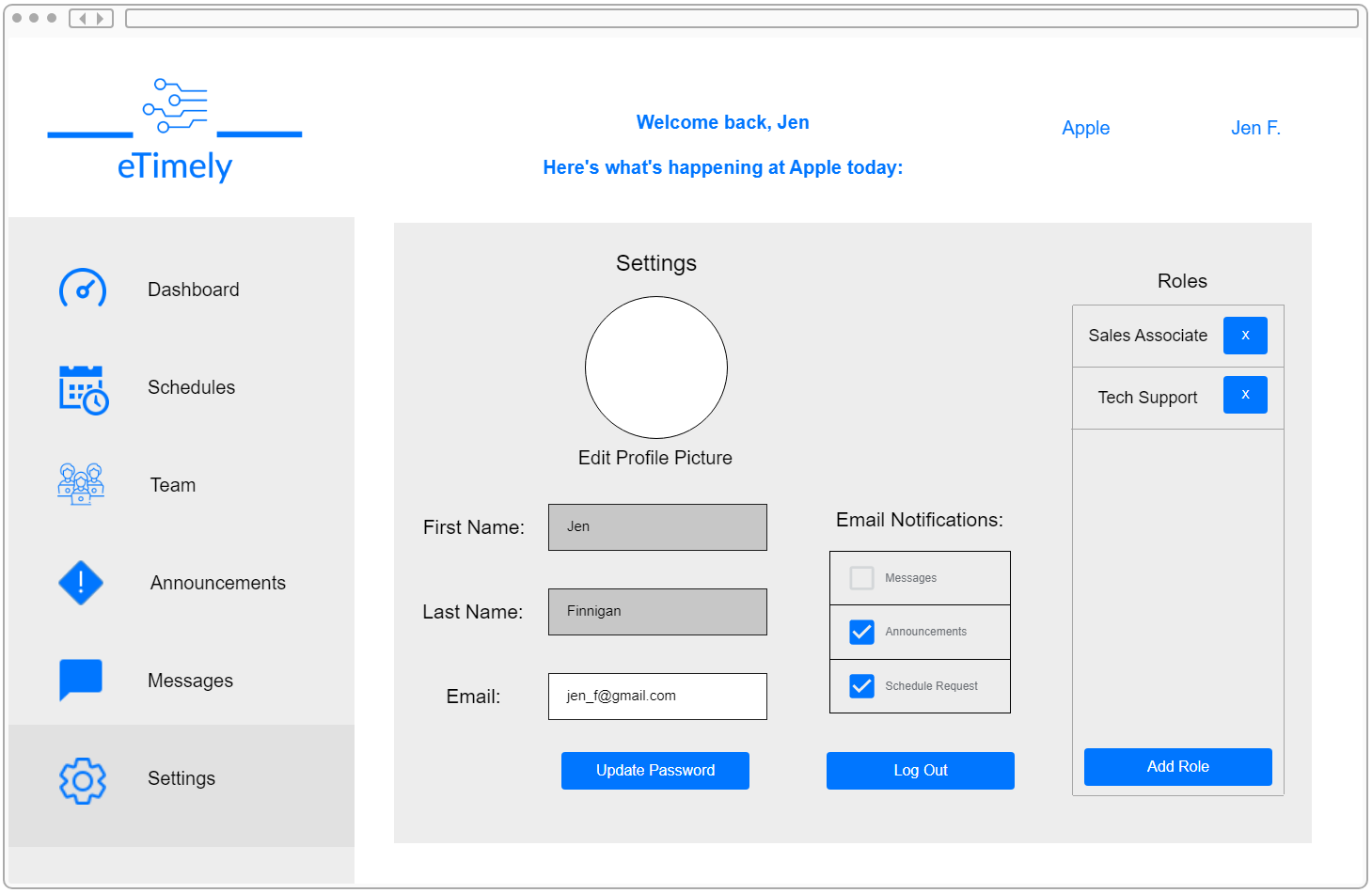
**4.7.3.6 Business Messages Tab**



**4.7.3.7 Business Announcements Tab**



**4.7.3.8 Business Settings Tab**



Appendix A: References

The following table summarizes the documents referenced in this document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Reference Number** | **Document Name** | **Description** | **Location** | |
| 1 | React Documentation | Basic documentation of React. Refers to the compatible web browsers | <https://reactjs.org/docs/react-dom.html> |
| 2 | Firestore Quotas and Limits | Documentation outlining the limitations of Firestore. | <https://cloud.google.com/firestore/quotas> | |
| 3 | JSON Web Token (JWT) Documentation | Documentation outlining how Json web tokens work. | https://jwt.io/introduction | |

Appendix B: Key Terms

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Actors | The people or external systems acting on eTimely |
| Availability | The specified times and days of the week an employee is willing to be scheduled. |
| Business | A company that has signed up for and who is using eTimely. |
| Employee | A singular person who works at the business who is using eTimely. |
| Organization | A business account and all associated staff accounts. |
| Role | Specific job title assigned by a business to a staff user. |
| Staff | All employees of a business. |
| User | Any person using the application, generally someone using a business or staff account |