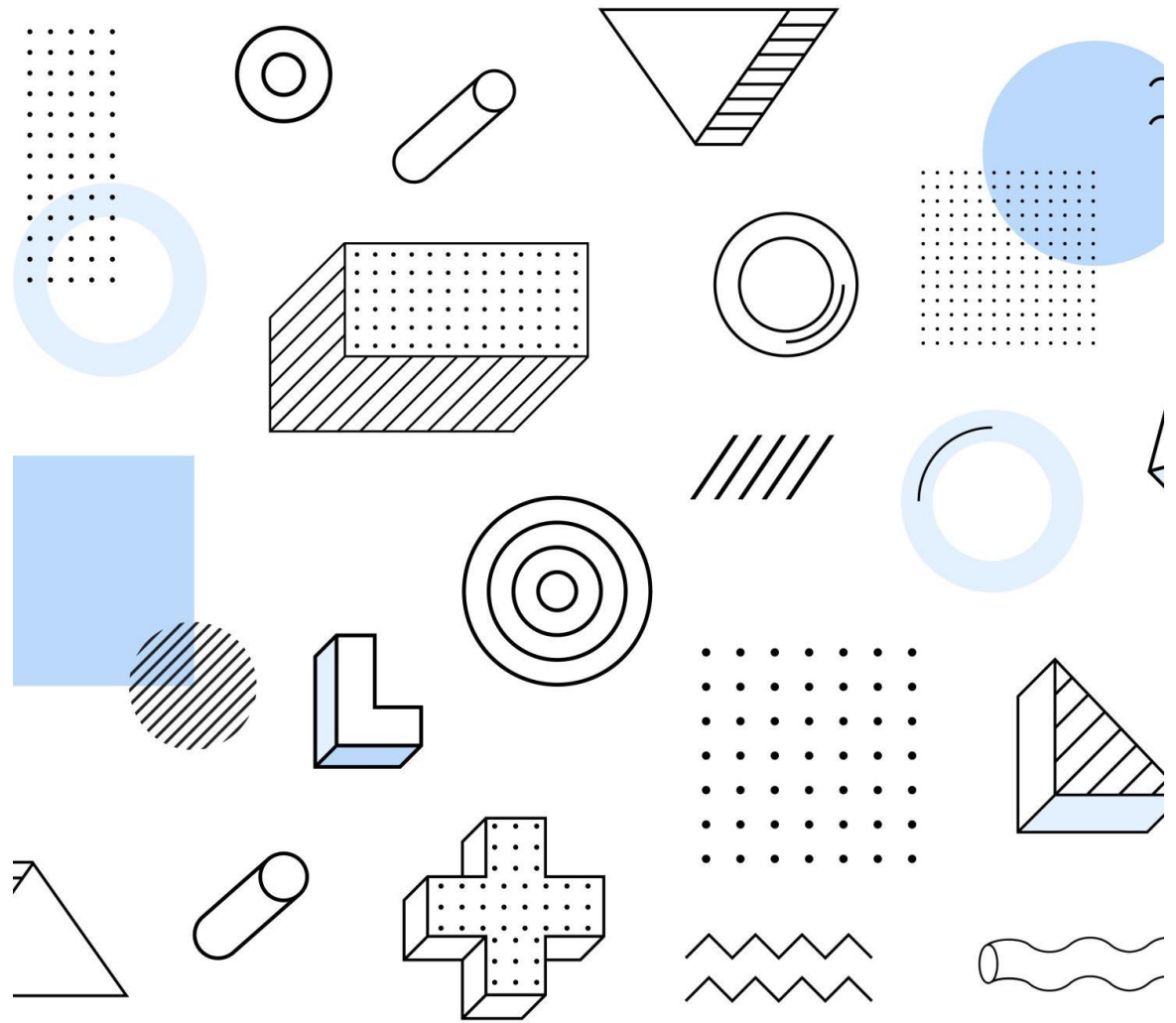# Design Specification eTimely

Matt Korte, Caleb Obi, Chris Gumieny, & Samia Chowdhury
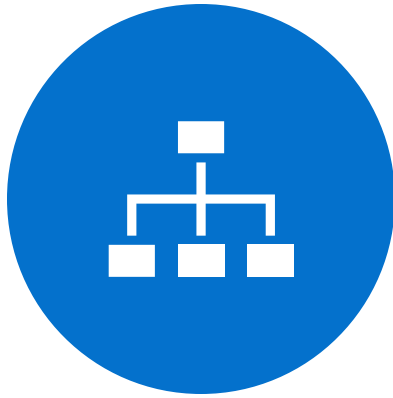
Introduction

Purpose

LAYOUT, EXPLAIN, AND TRACK ARCHITECTURE & DESIGN

DEFINE AND TRACK USE CASES

GUIDE THE DEVELOPMENT TEAM

Assumptions

Internet connection

No information will be stored locally with user

Firestore

Database structure

Constraints

React v17.0.2

Limits tools and libraries we can use

Not supported on all browsers
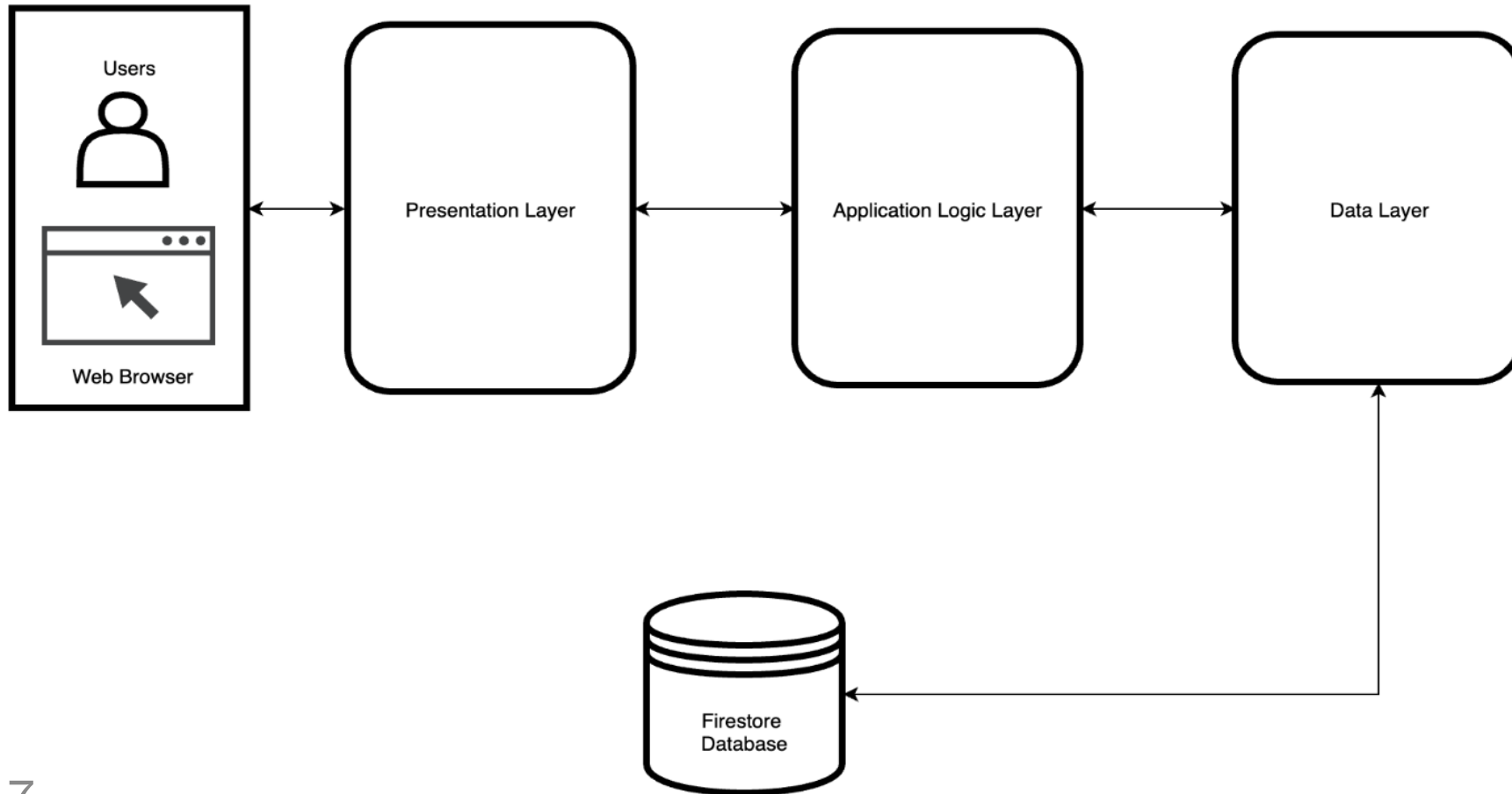
NodeJS v16.13.2

Used for Backend Development
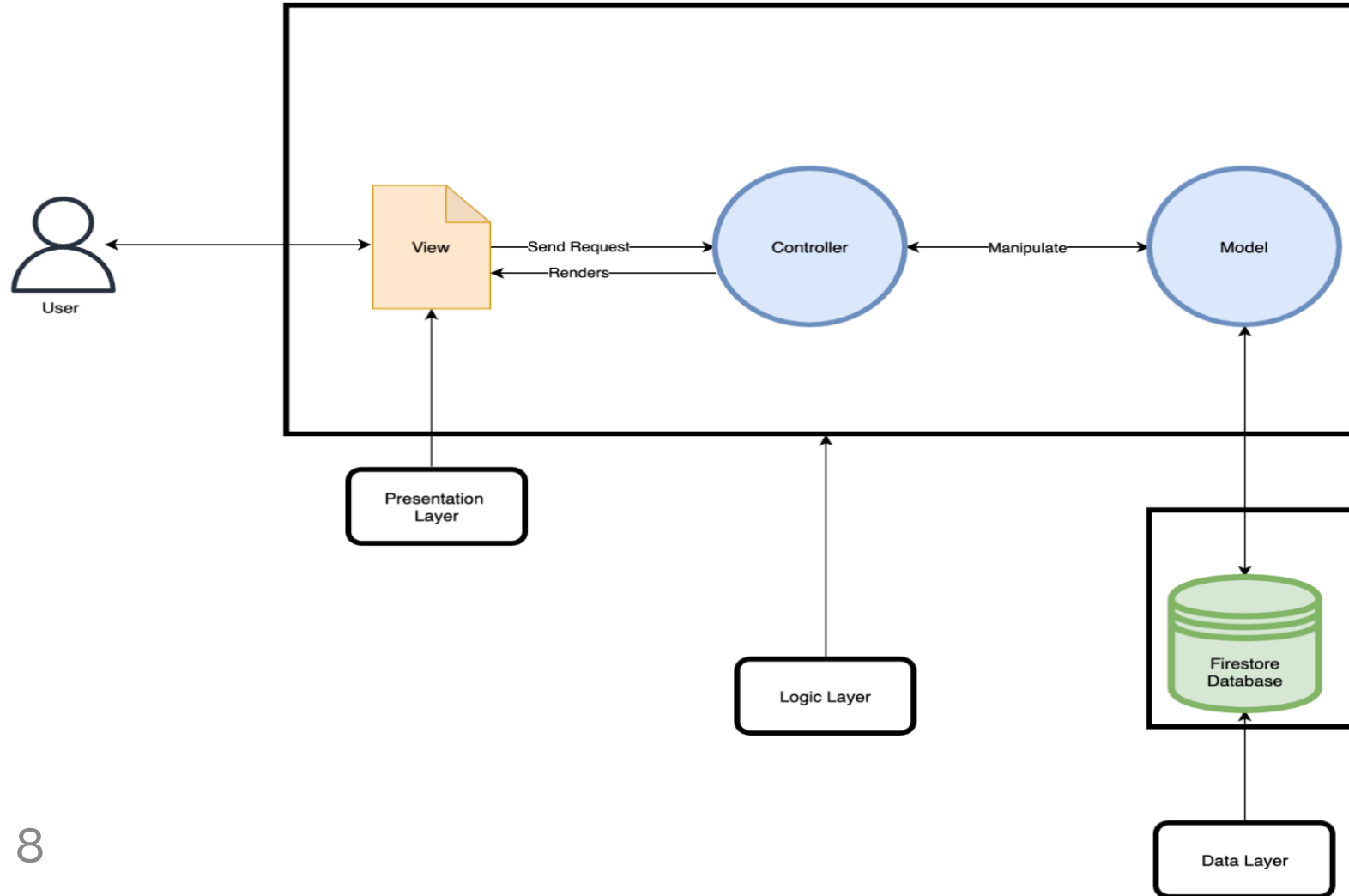
Firestore

Limits on data stored, reads/writes

Architectural Design
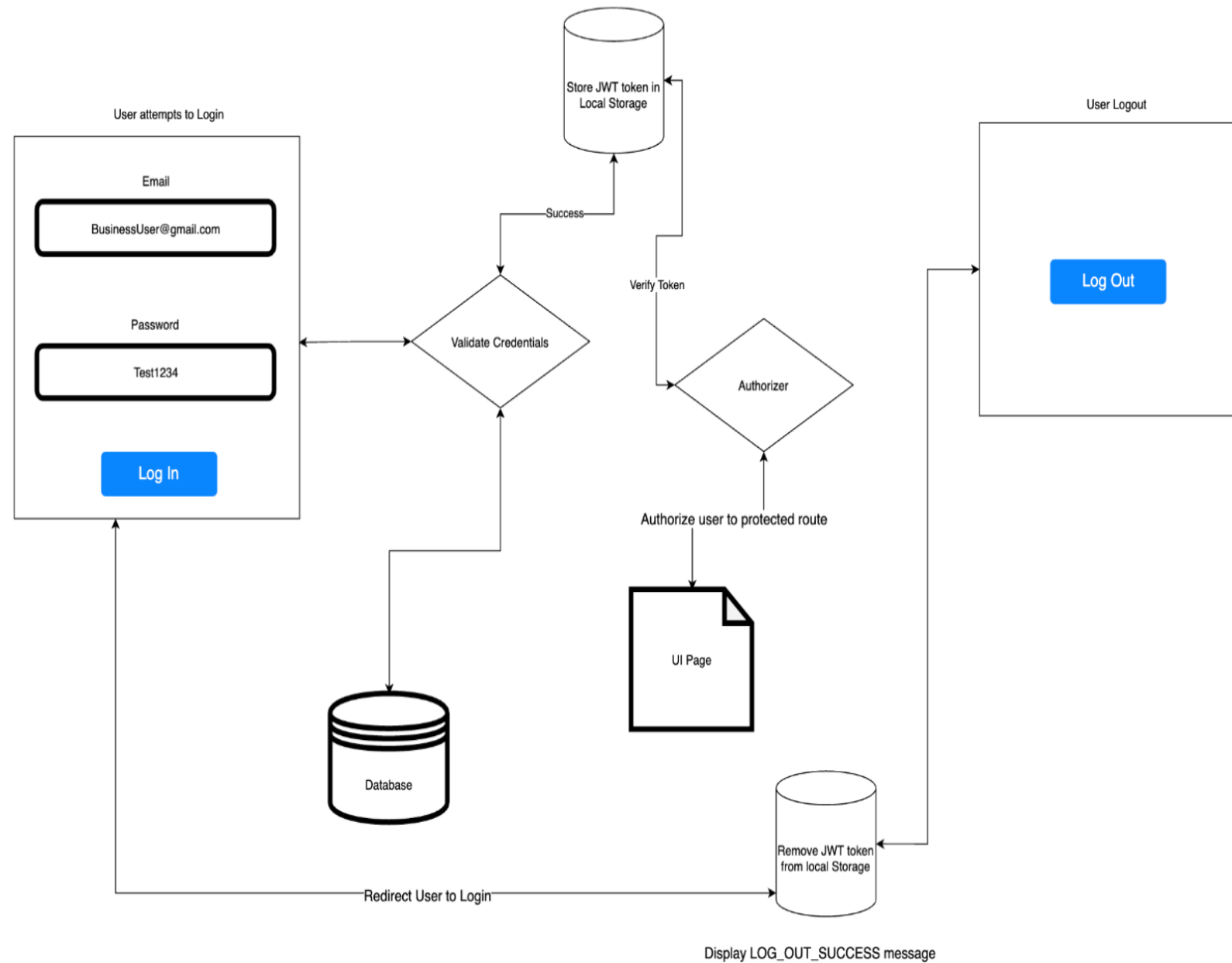
Software Architecture – Layered Architecture
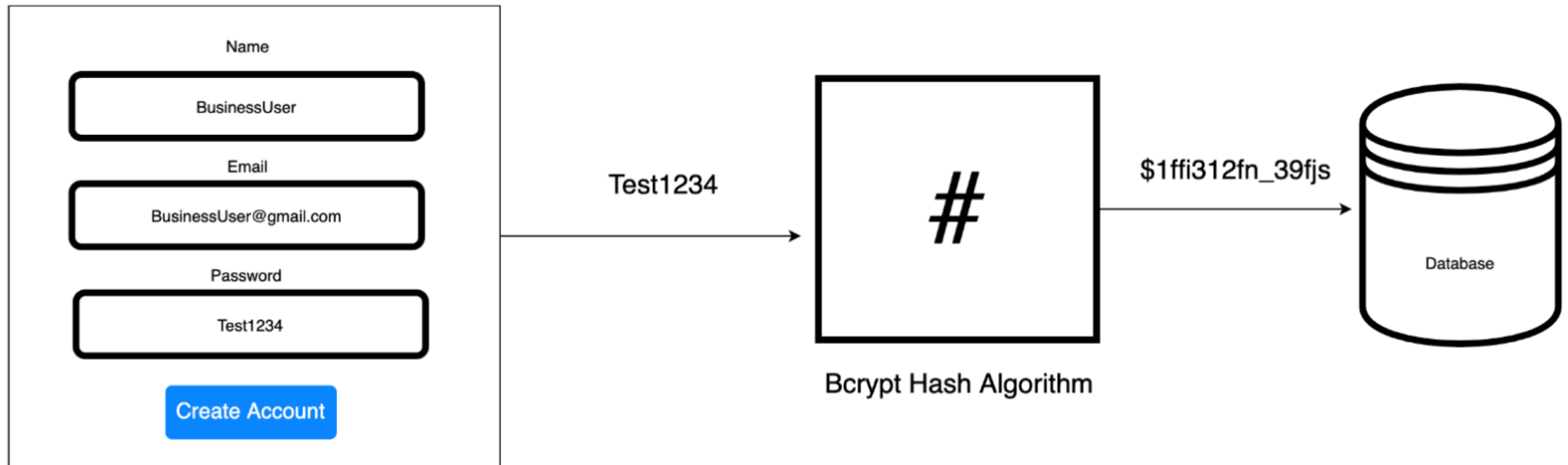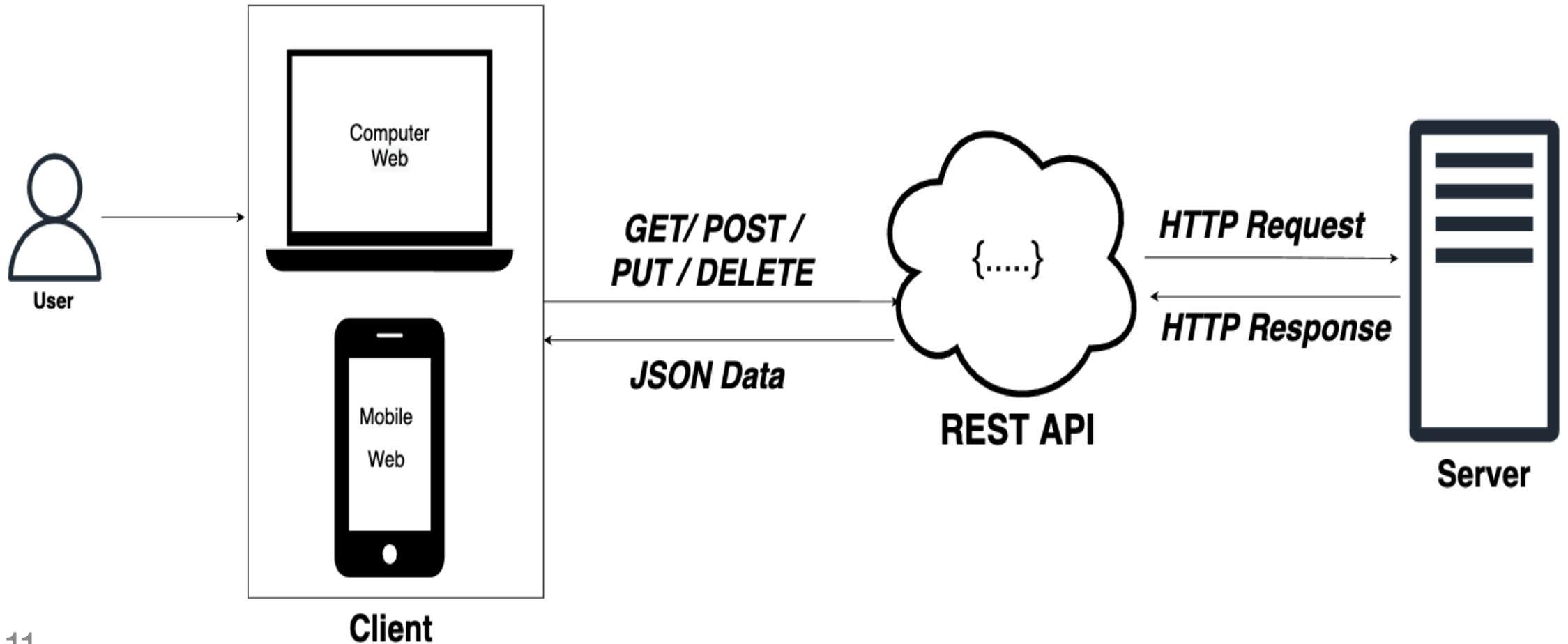
Software Architecture - MVC

Security Architecture – Web Tokens

Security Architecture – Password Encryption

Communication Architecture

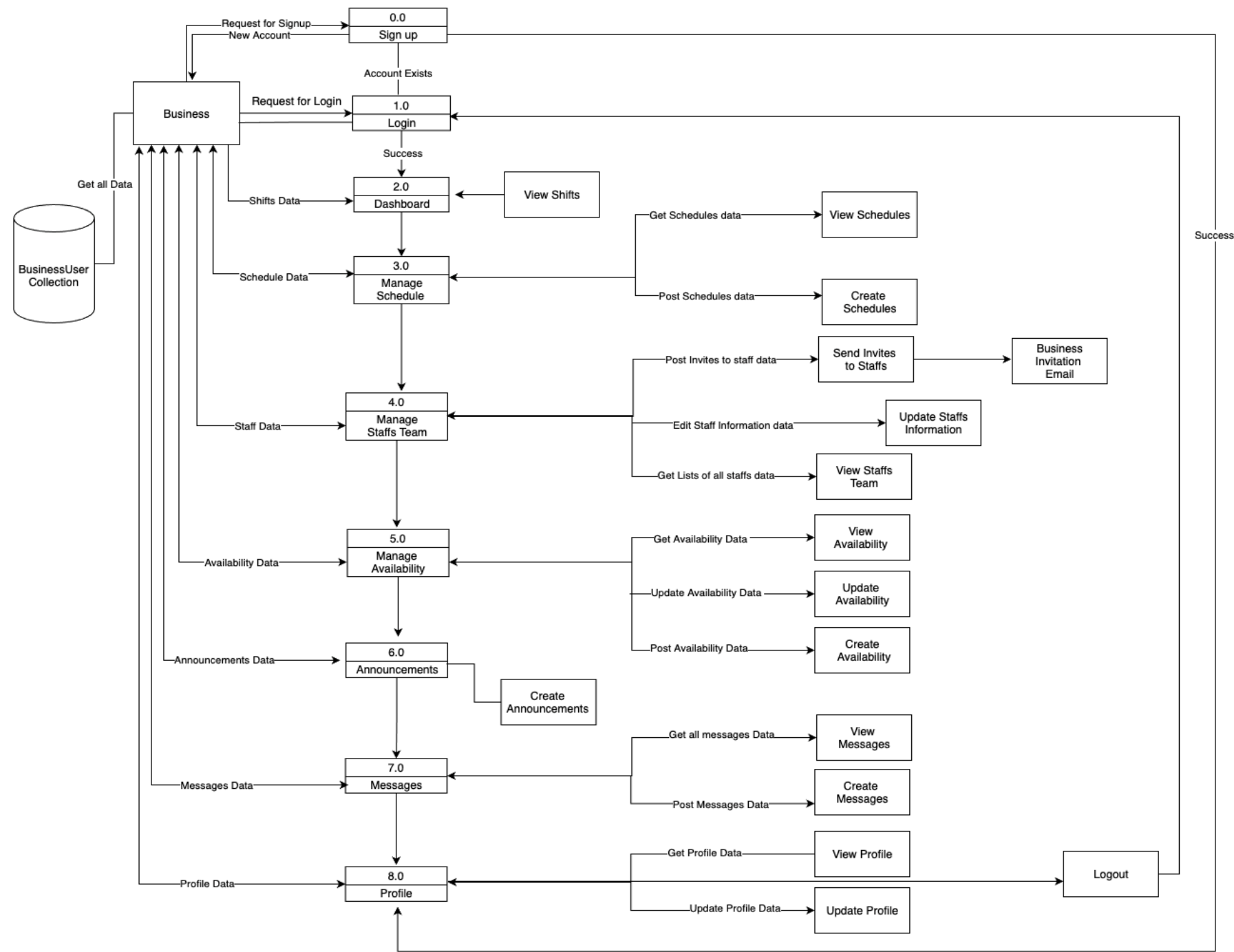# System Design

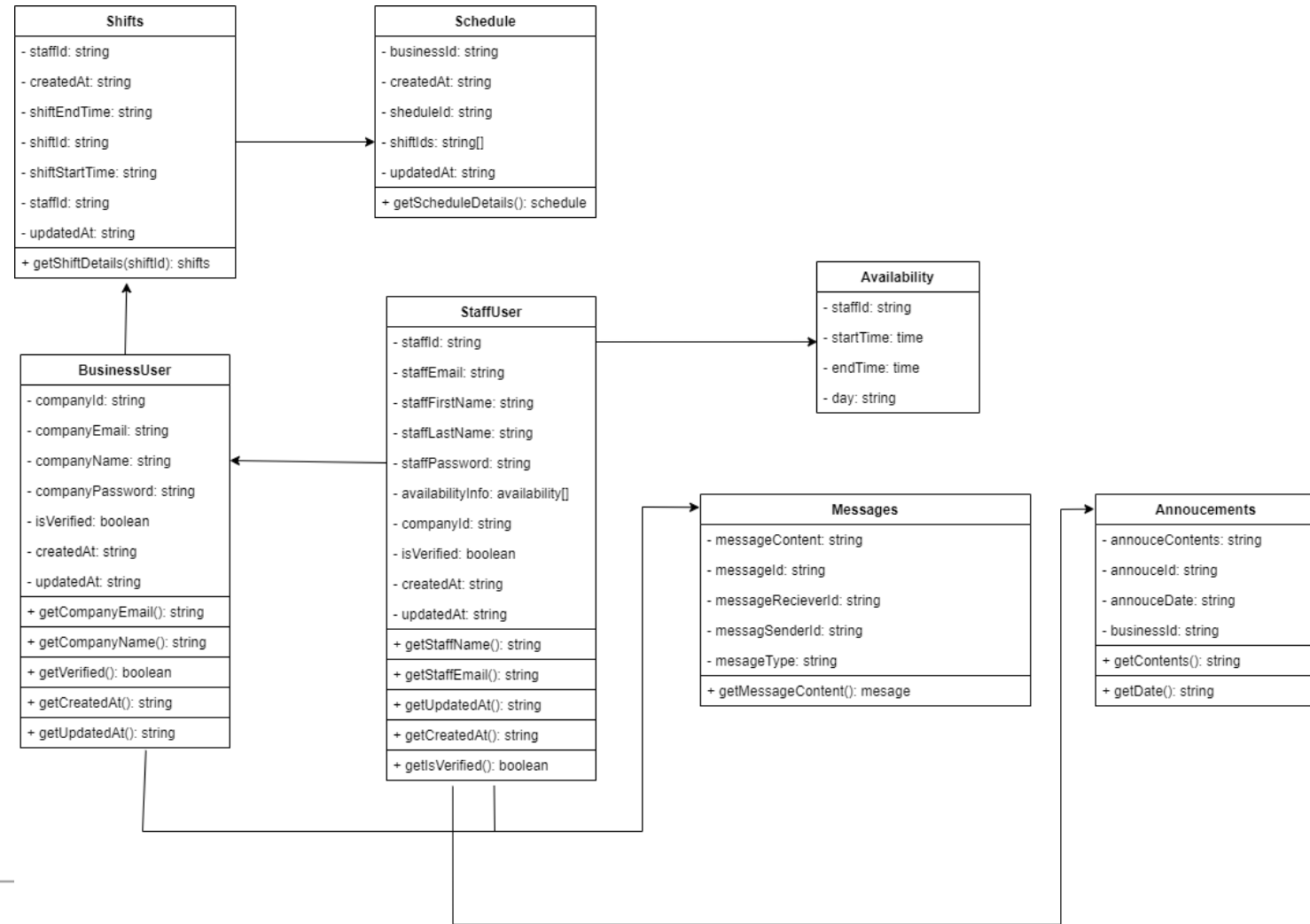# Data Flow Diagram - Business User

Class Diagram

Shifts

- staffId: string
- createdAt: string
- shiftEndTime: string
- shiftId: string
- shiftStartTime: string
- staffId: string
- updatedAt: string

+ getShiftDetails(shiftId): shifts

Schedule

- businessId: string
- createdAt: string
- sheduleId: string
- shiftIds: string[]
- updatedAt: string

+ getScheduleDetails(): schedule

Availability

- staffId: string
- startTime: time
- endTime: time
- day: string

StaffUser

- staffId: string
- staffEmail: string
- staffFirstName: string
- staffLastName: string
- staffPassword: string
- availabilityInfo: availability[]
- companyId: string
- isVerified: boolean
- createdAt: string
- updatedAt: string

+ getStaffName(): string
+ getStaffEmail(): string
+ getUpdatedAt(): string
+ getCreatedAt(): string
+ getIsVerified(): boolean

BusinessUser

- companyId: string
- companyEmail: string
- companyName: string
- companyPassword: string
- isVerified: boolean
- createdAt: string
- updatedAt: string

+ getCompanyEmail(): string
+ getCompanyName(): string
+ getVerified(): boolean
+ getCreatedAt(): string
+ getUpdatedAt(): string

Messages

- messageContent: string
- messageId: string
- messageRecieverId: string
- messagSenderId: string
- mesageType: string

+ getMessageContent(): mesage

Annoucements

- annouceContents: string
- annouceId: string
- annouceDate: string
- businessId: string

+ getContents(): string
+ getDate(): string

14

Use Cases

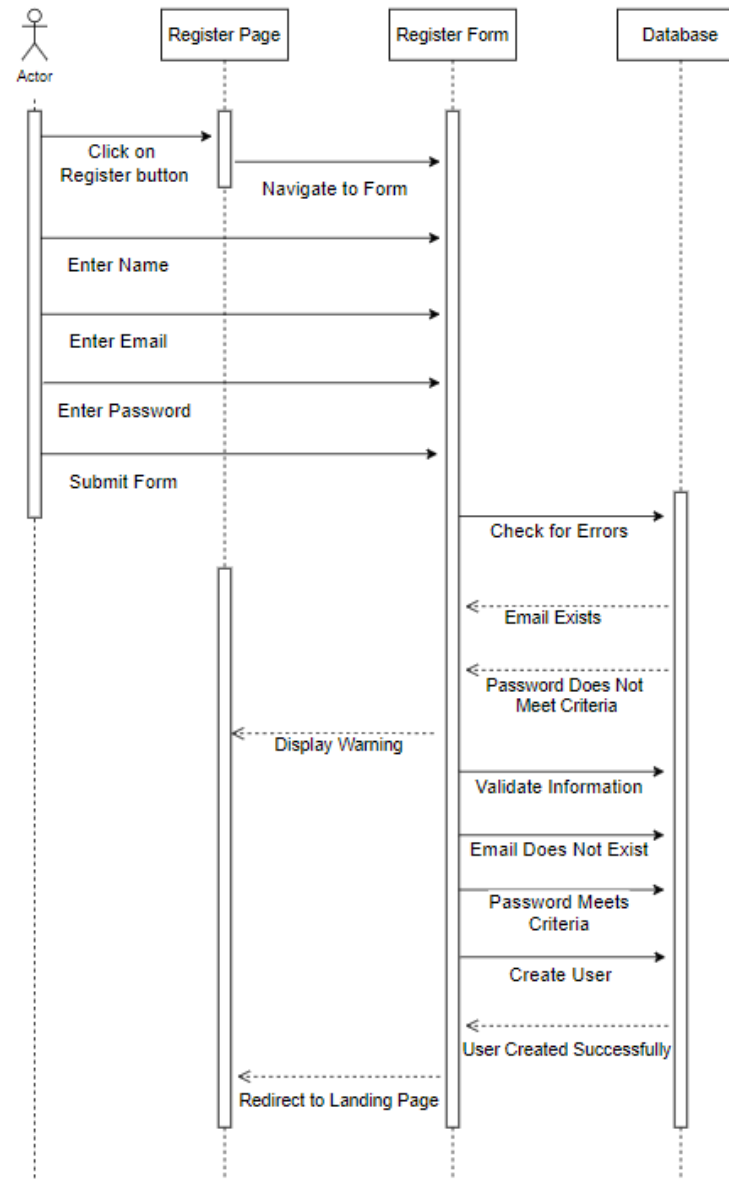| Use Case ID: UC-1		Use Case Name: Business Sign-Up		
Created By: Matt Korte		Date Created: 2/24/2022		
Last Updated By: Samia Chowdhury		Last Revision Date: 3/3/2022		
Actors	New Business Users			
Description	New business user creating a business account			
Trigger	User clicking on the "Create Account" 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.			

15

Use Cases Cont.

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

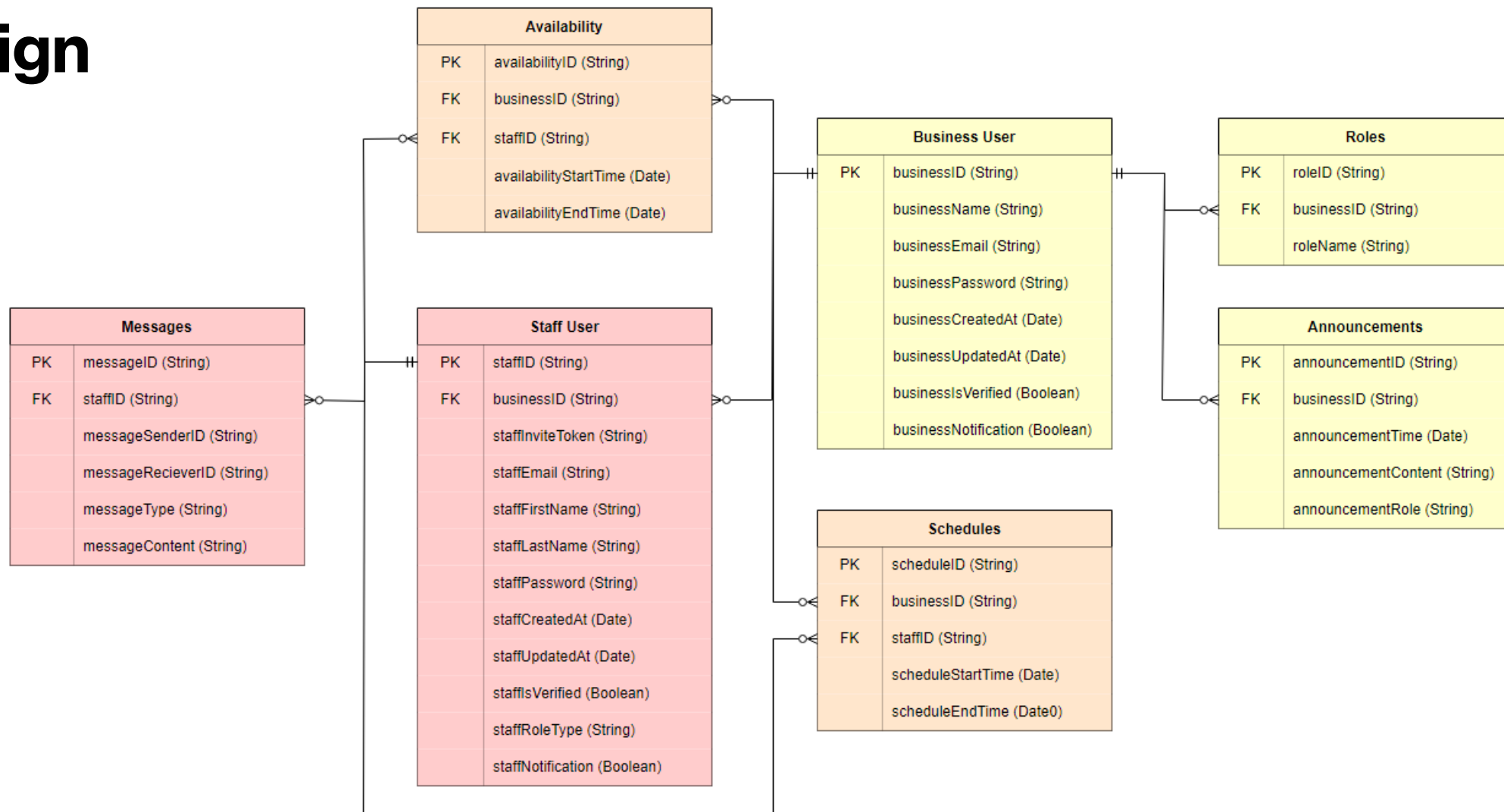|	
Actors	Business Users
Description	Business users create the schedule for their staff
Trigger	The business user navigates to the scheduling screen
Preconditions	The user is signed into a business account
Postconditions	The schedule information is saved in the database, to later be called by the staff 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	1. Repeat steps 1-4 of the normal flow 2. User clicks the "Cancel" button resulting in none of the changes being saved and returns user to regular schedule page window
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 Cases Cont.

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

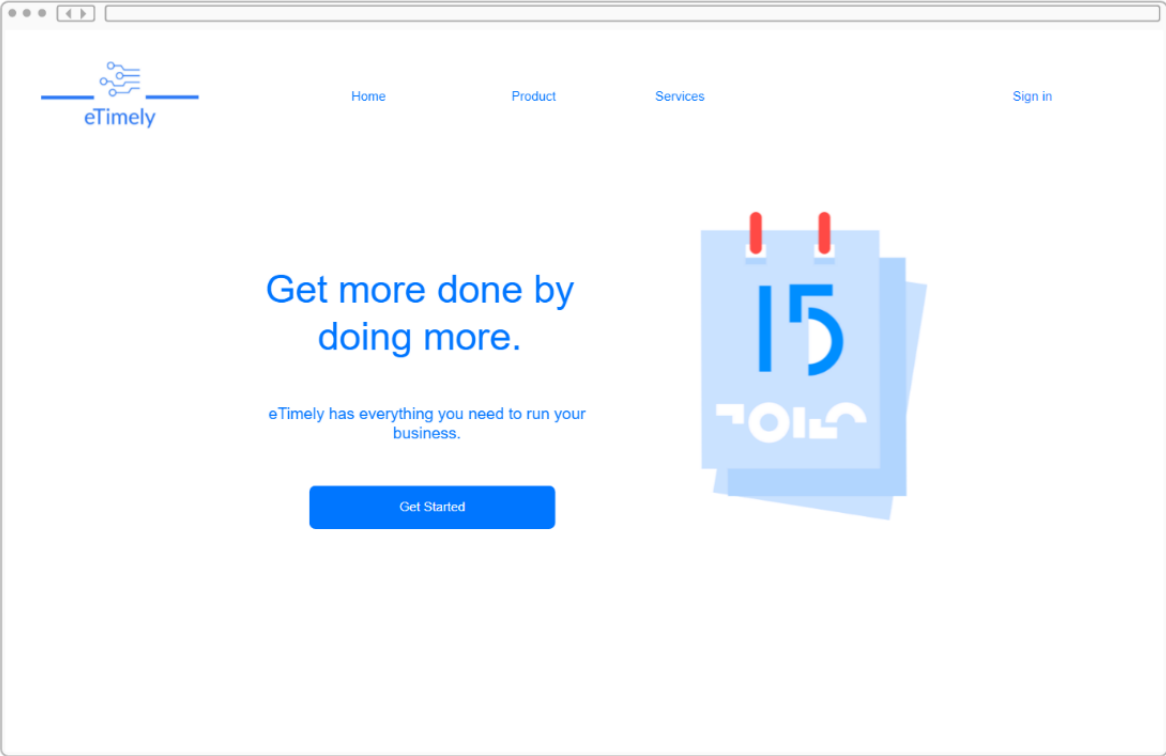|	
Actors	Staff Users
Description	Staff user updating the days and times they are available to work
Trigger	Clicking the "Availability" button on the left navigation
Preconditions	The user is logged into a staff account
Postconditions	Staff Availability is updated to reflect the times inputted by the 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

User Sign Up - Sequence Diagram

Database Design

Availability

PK	availabilityID (String)
FK	businessID (String)
FK	staffID (String)
	availabilityStartTime (Date)
	availabilityEndTime (Date)

Business User

PK	businessID (String)
	businessName (String)
	businessEmail (String)
	businessPassword (String)
	businessCreatedAt (Date)
	businessUpdatedAt (Date)
	businessIsVerified (Boolean)
	businessNotification (Boolean)

Roles

PK	roleID (String)
FK	businessID (String)
	roleName (String)

Messages

PK	messageID (String)
FK	staffID (String)
	messageSenderID (String)
	messageRecieverID (String)
	messageType (String)
	messageContent (String)

Staff User

PK	staffID (String)
FK	businessID (String)
	staffInviteToken (String)
	staffEmail (String)
	staffFirstName (String)
	staffLastName (String)
	staffPassword (String)
	staffCreatedAt (Date)
	staffUpdatedAt (Date)
	staffIsVerified (Boolean)
	staffRoleType (String)
	staffNotification (Boolean)

Announcements

PK	announcementID (String)
FK	businessID (String)
	announcementTime (Date)
	announcementContent (String)
	announcementRole (String)

Schedules

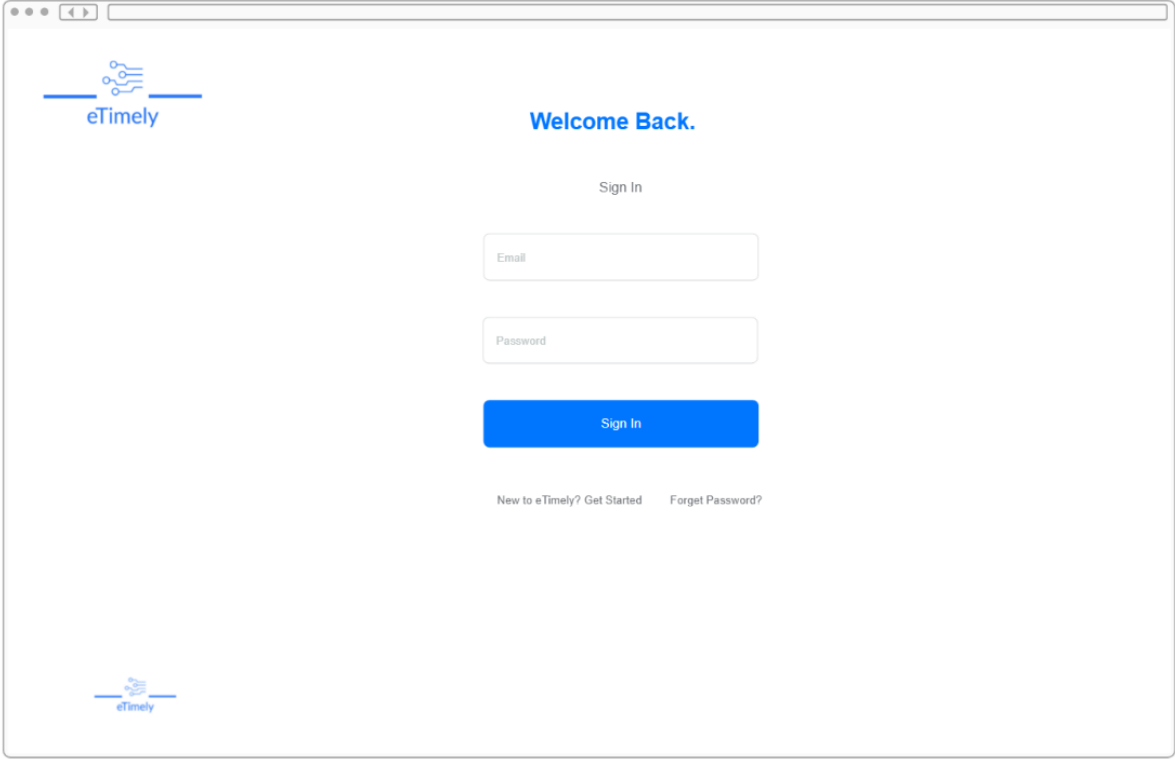|	
PK	scheduleID (String)
FK	businessID (String)
FK	staffID (String)
	scheduleStartTime (Date)
	scheduleEndTime (Date0)

User Interface Design
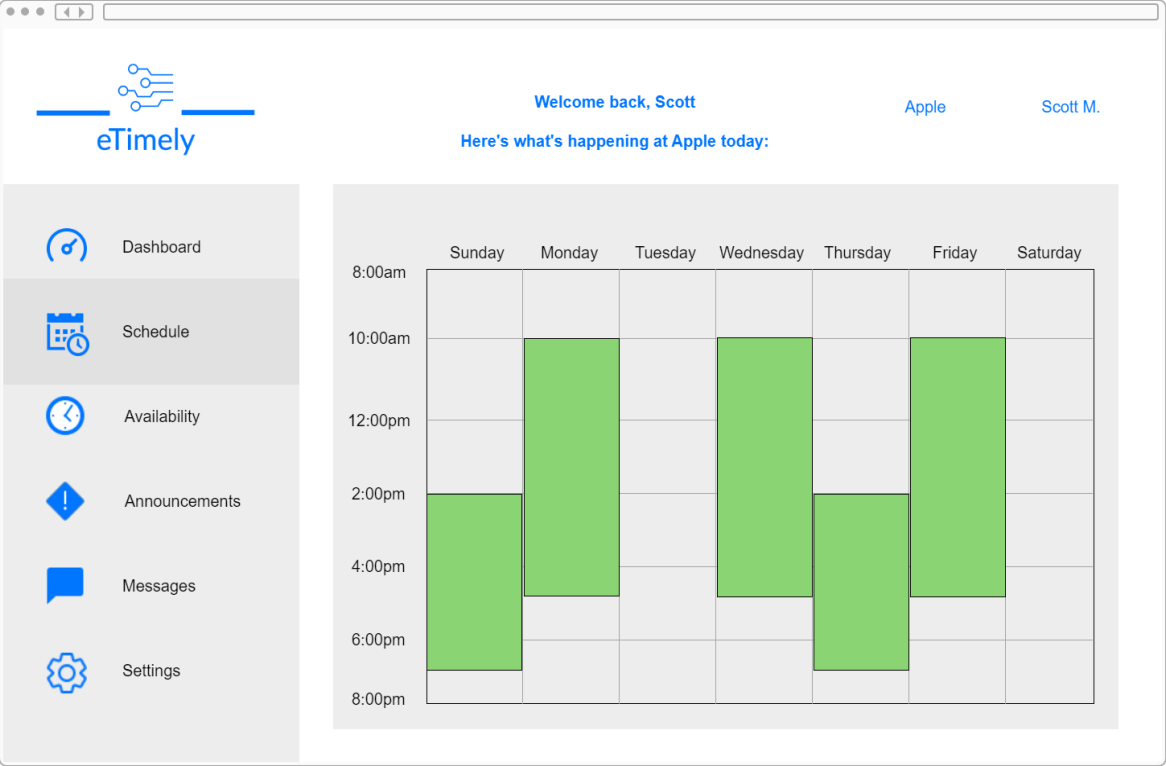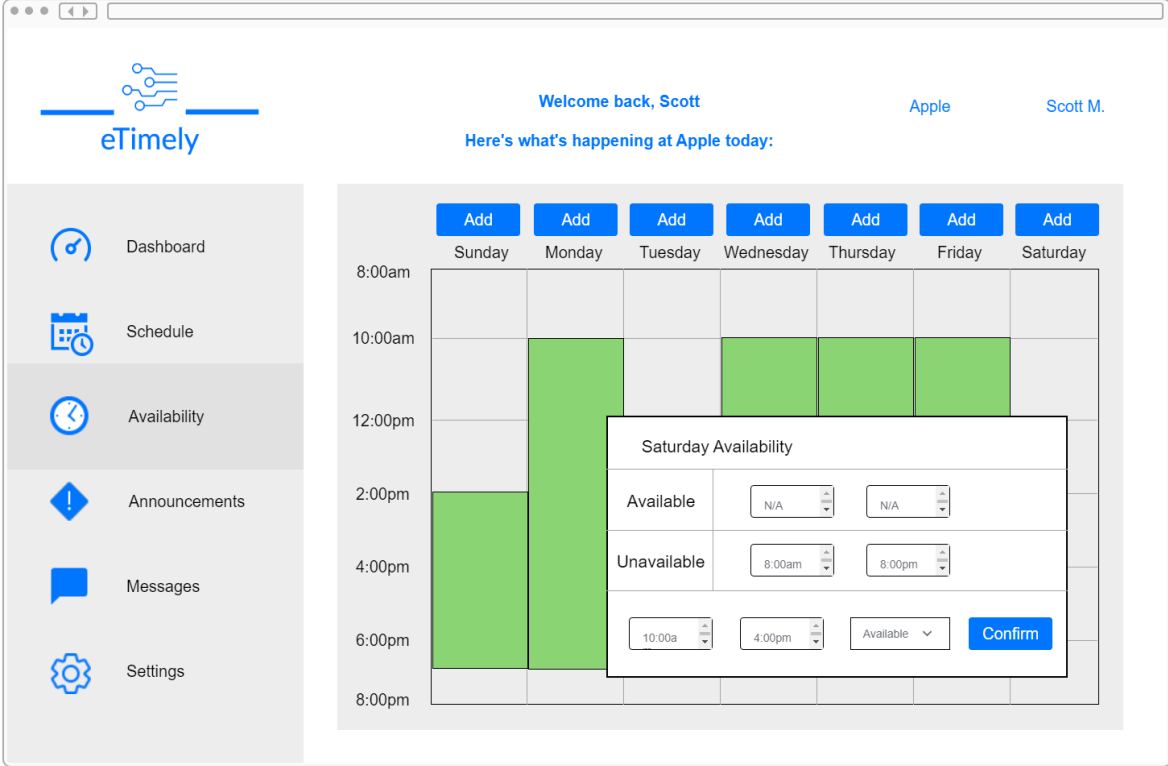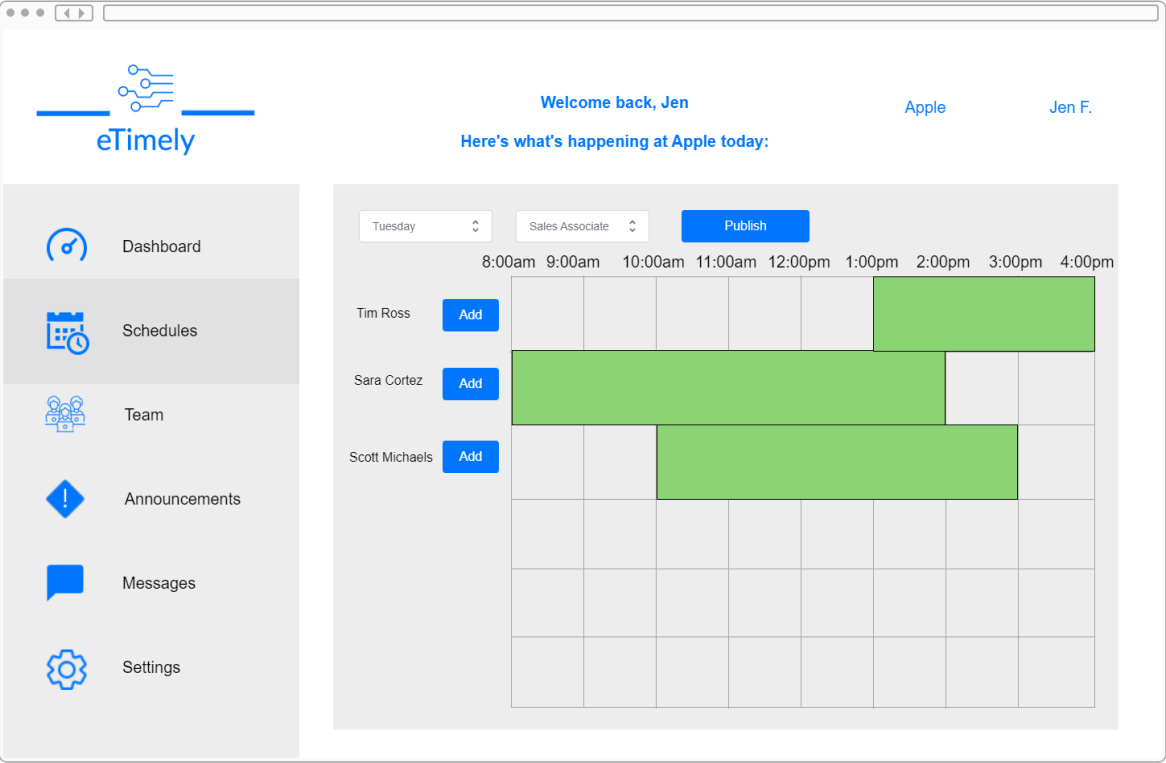
Default View
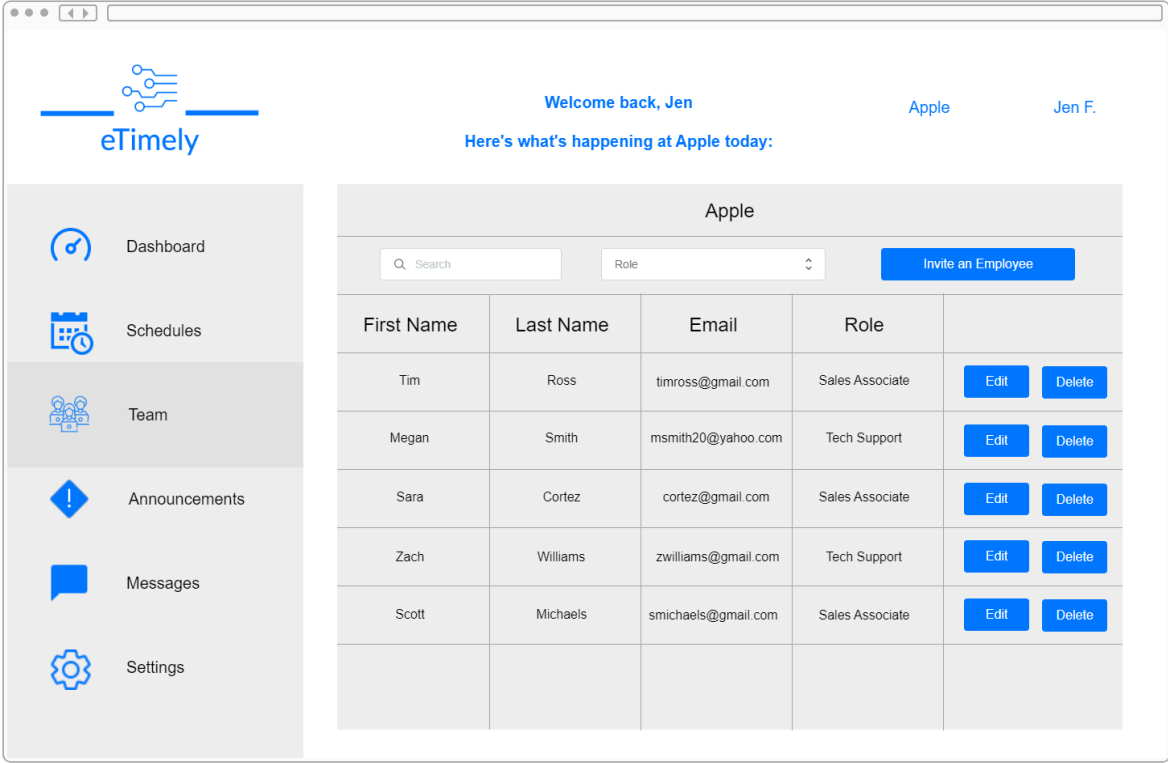

Home Page

Login Page

Staff View

Staff Schedule Tab

Staff Availability Window

Business View

Business Schedule Tab

Business Team Tab

23

References

- https://reactjs.org/docs/react-dom.html
- https://cloud.google.com/firestore/quotas
- https://jwt.io/introduction
- https://mui.com/getting-started/supported-platforms/

Questions?