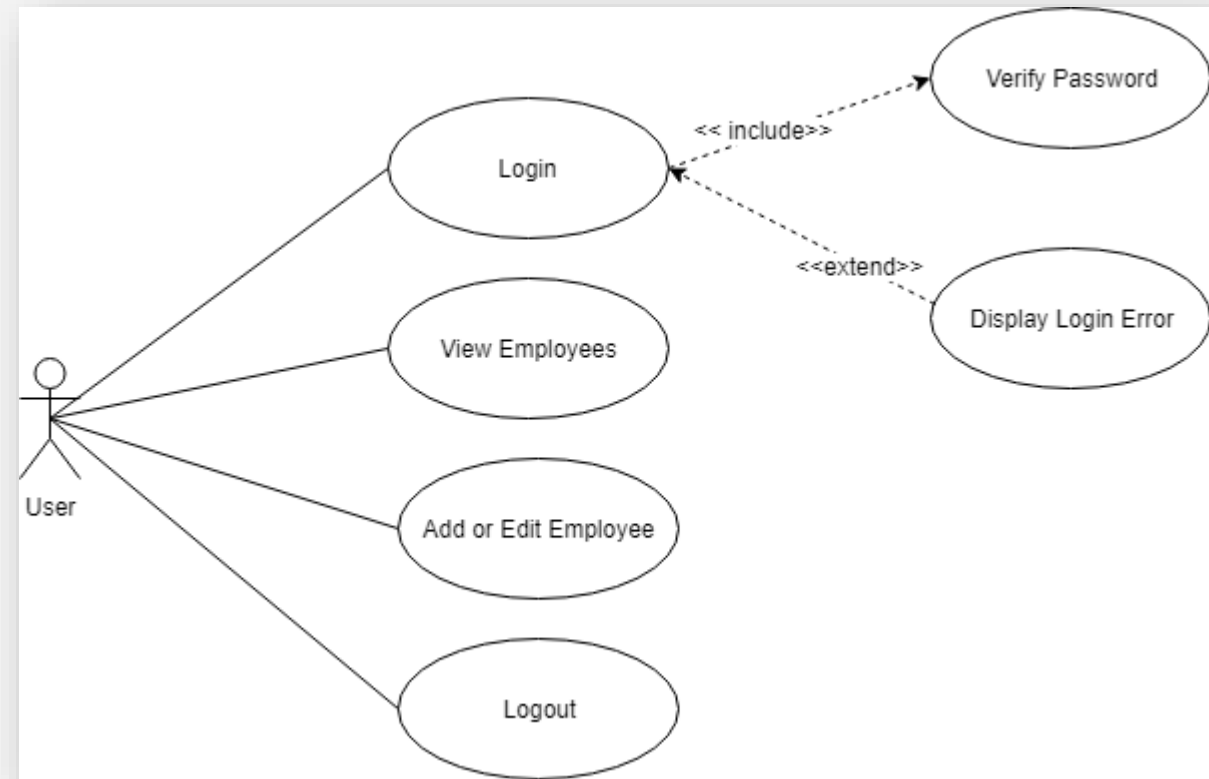# Full Stack Coding Challenge

Baton Rouge – Client Innovation Center
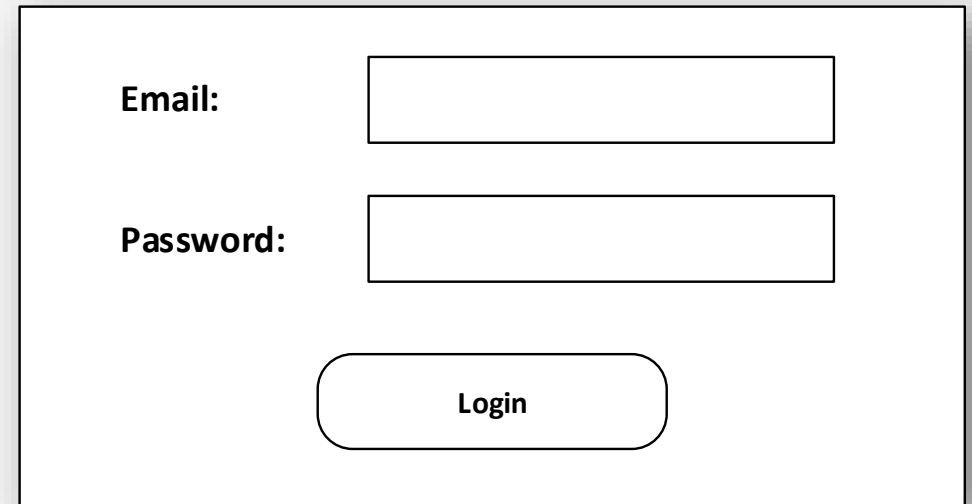
Use Case

- Create a login page for user authentication.

  ❖ Validate the user exists.

  ❖ Display error if the user does not.

- Upon a successful login, the user shall view a list of employees.

- The user shall have the ability to add new employees.

- The user shall have the ability to update existing employees.

Employee Login Page

- The user shall not login if the email and password do not meet minimal requirements.

- If the user login is unsuccessful, an error shall display on the login page.
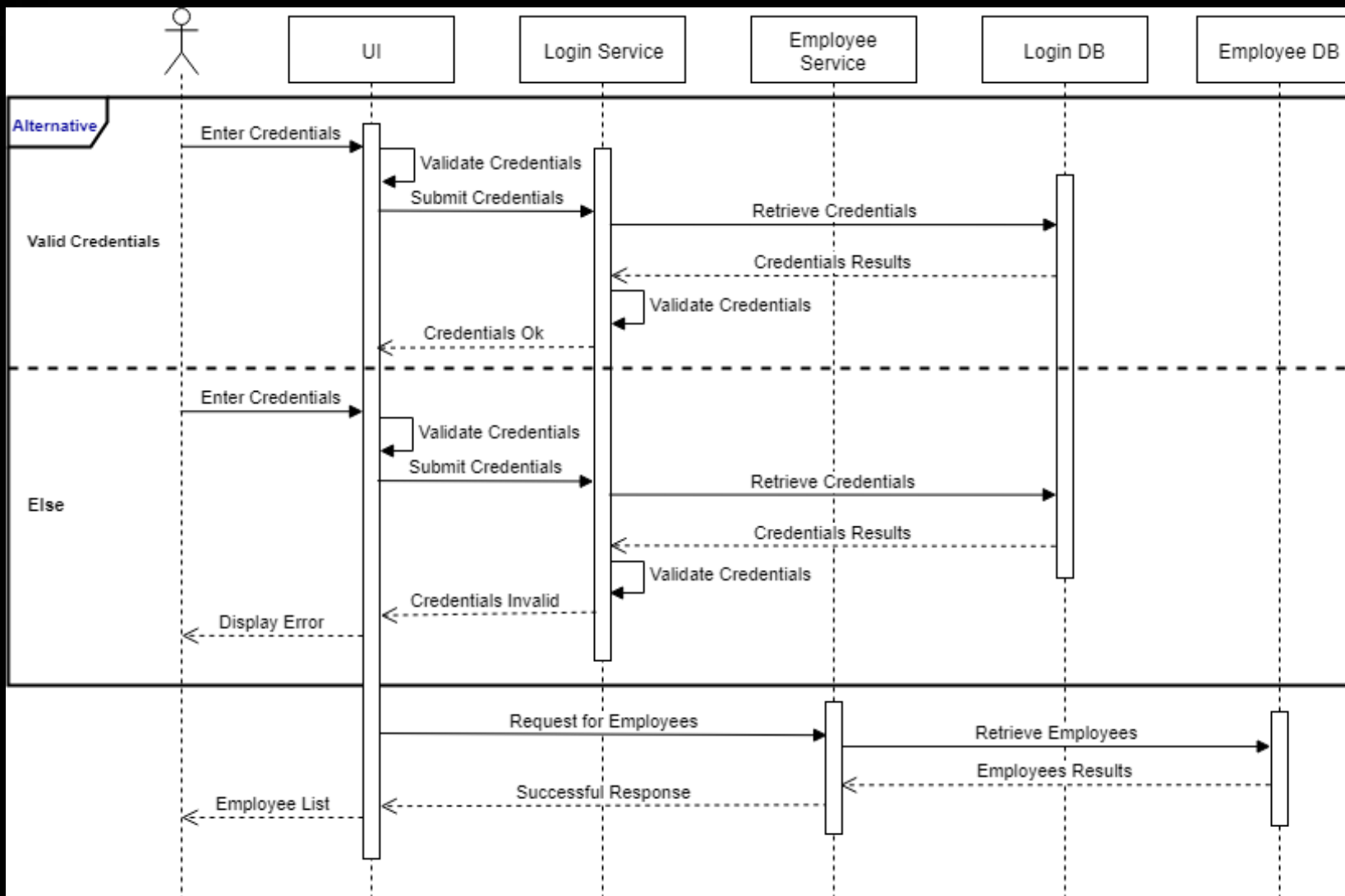
Email:

Password:

Login

Data Entry Lengths and Validation Information

- Email: Minimum 8, Maximum 35 (alpha numeric)

- Password: Minimum 8, Maximum 35 (alpha numeric)

Login Sequence Diagram

Employee List Page

- Upon a successful login, the application shall display the employee list page.

- The employee list shall be sorted by the employee names.

- The add employee button shall navigate to the employee add/edit page.

- The employee name link shall navigate to the employee add/edit page.

Add Employee

Name	Email Address
Jane Doe	jdoe@gmail.com
Jane Doe	jdoe@gmail.com
Jane Doe	jdoe@gmail.com
Jane Doe	jdoe@gmail.com
Jane Doe	jdoe@gmail.com

Employee Update Sequence Diagram
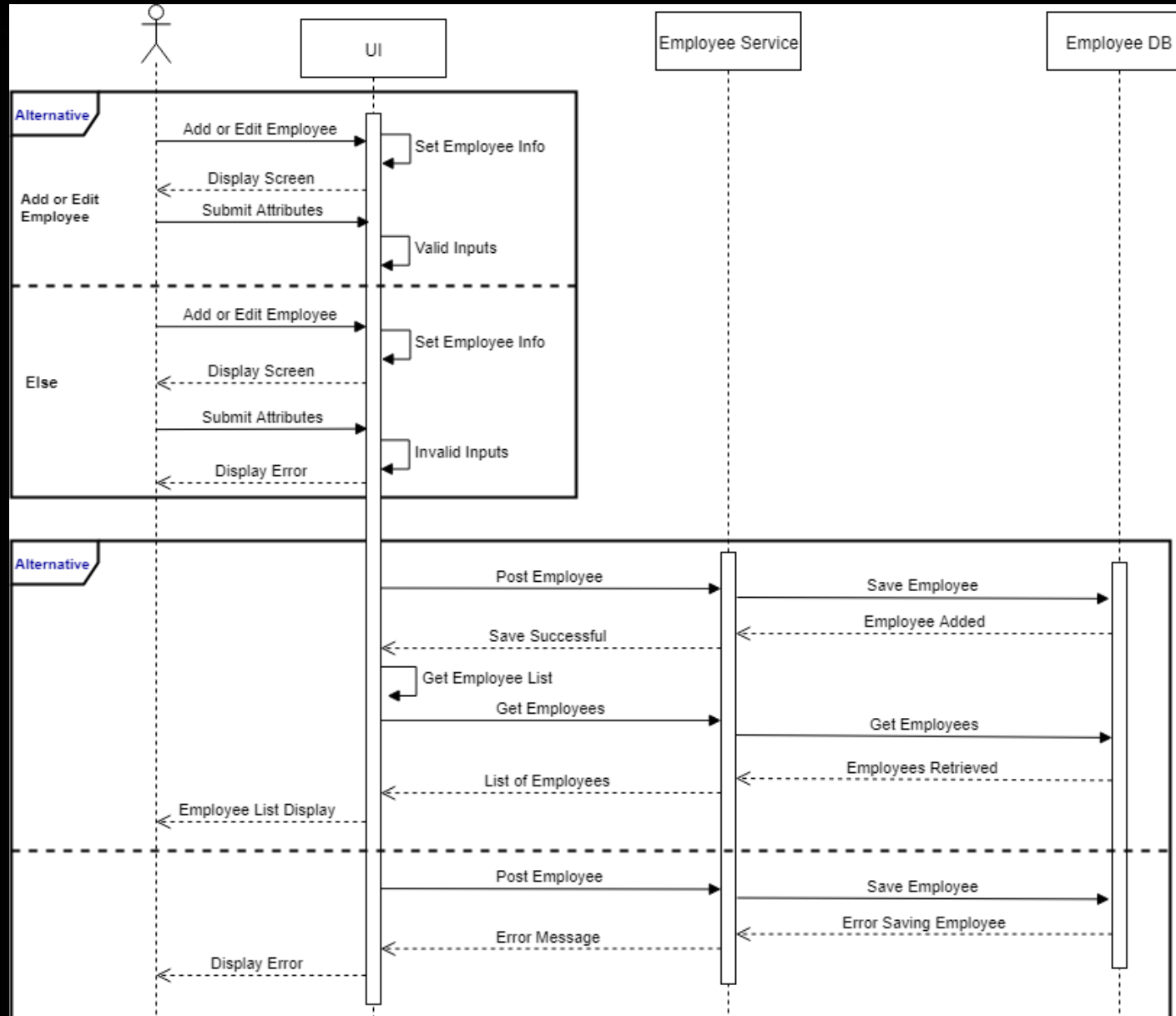
Employee Add/Edit Page

- All fields are required.

- User cannot save the employee information unless all fields are populated.

- Upon clicking the Cancel button, the user shall navigate to the employee list page.

- Upon clicking the Save button, the entered information shall update the database and navigate to the employee list page.

Employee - New

First Name:

Last Name:

Address:

City:

State: ▼

Zip/Postal Code:

Home Phone:

Cell Phone:

Email:

Save   Cancel

Data Entry Lengths and Validation Information

- First and Last Names: Minimum 2, Maximum 35 (alpha, spaces allowed)

- Address: Minimum 10, Maximum 50 (alpha numeric , spaces allowed)

- City: Minimum 5, Maximum 50 (alpha , spaces allowed)

- State: Dropdown, value 2 characters (Dropdown)

- Zip/Postal Code: Minimum 5, Maximum 9 (numeric only)

- Home/Cell Phone: 10 characters (numeric only)

- Email: Minimum 10, Maximum 50 (alpha numeric , email validation)

Stacks to Deliver

IBM

Angular Front End

- **A**ngular

- **N**odeJS

React Front End
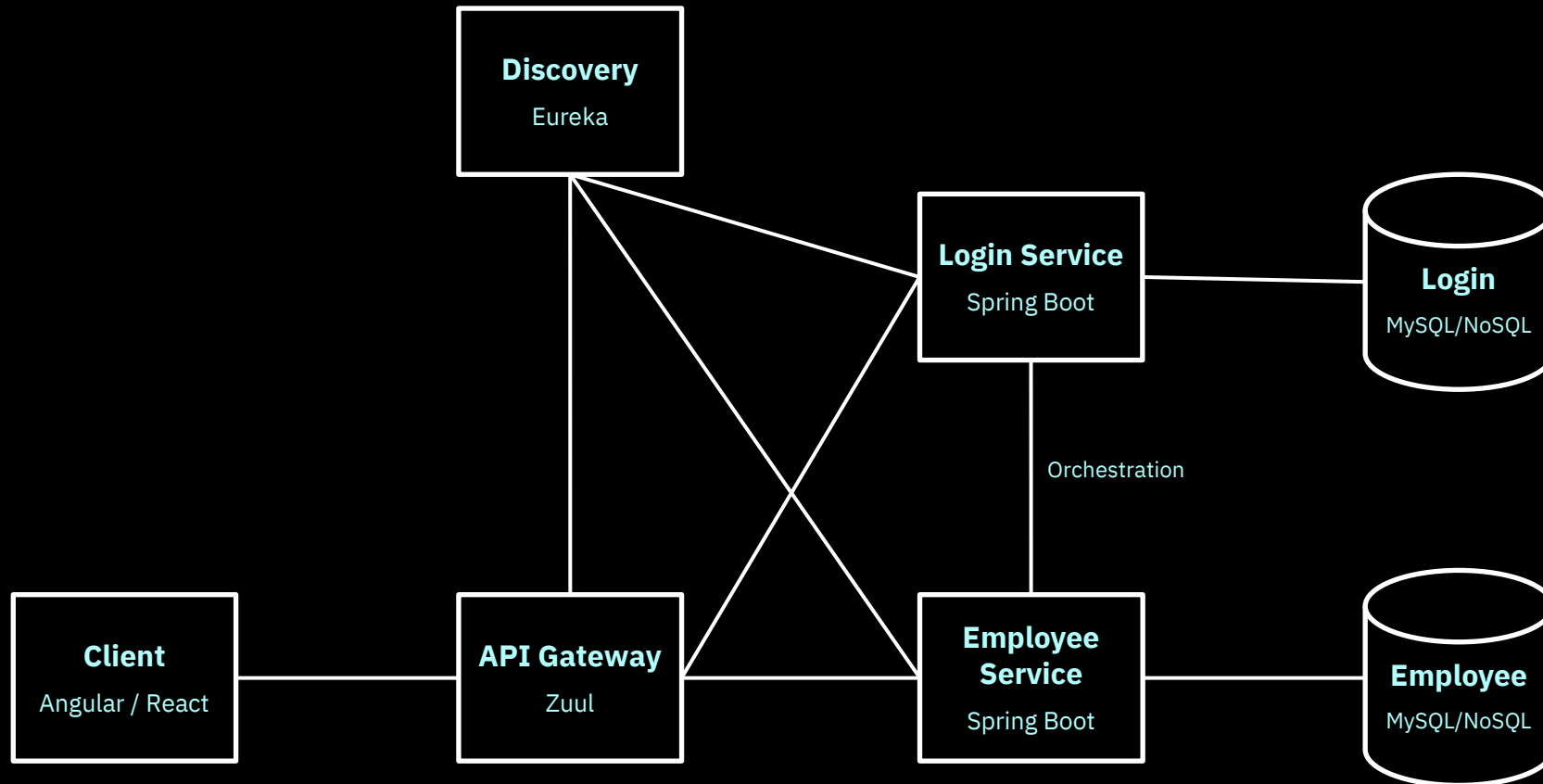
- **R**eact

- **N**odeJS

Java Backend

OR

- MySQL / NoSQL

- Spring Boot

- JPA/Hibernate

- API Gateway

- Service Discovery

- Microservices
  - ❖ Login Service
  - ❖ Employee Service

.Net Backend

- MySQL

- Web APIs

- EF Core

- Ocelot

- Steeltoe/Eureka

- Microservices
  - ❖ Login Service
  - ❖ Employee Service

High Level Architecture for Services

Requirements

Java Backend

- JDK 1.8
- Spring Boot
- JPA
- Hibernate
- Eureka (Discovery)
- Zuul (API Gateway)

Front End

- Visual Studio Code
- Spring Tool Suite (STS)
- NodeJS
- Angular
- ReactJS

OR

.NET Backend

- .NET Core 2.2+
- Web APIs
- EF Core
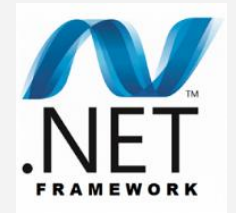- Ocelot (API Gateway)
- Steeltoe/Eureka

Tech Stack Counterparts
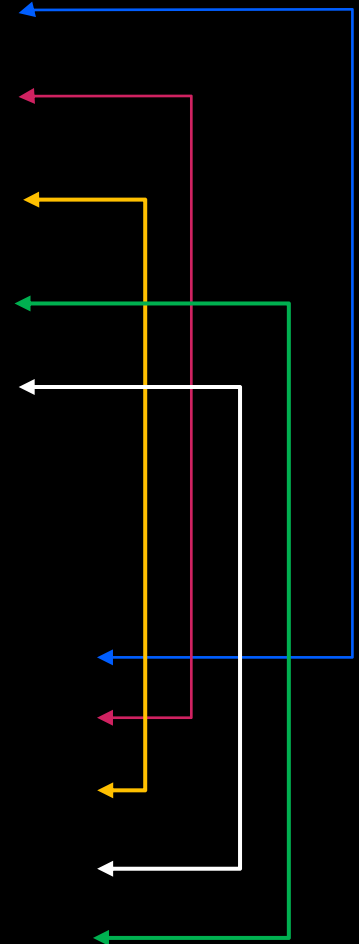
Technology Exposure

- Docker / Docker Compose
- SpringBoot
- .Net
- JPA
- Hibernate
- NodeJS
- ReactJS
- AngularJS
- Javascript ES5/ES6
- Jenkins
- Kubernetes

Deliverables

Before you start ...

Prerequisites

- IDEs
  - ➢ Spring Suite Tool IDE
  - ➢ Visual Studio Code
  - ➢ Or, your choice
- JDK 1.8 or later
- NodeJS
- Docker Hub Account: https://hub.docker.com/
- IBM Cloud Account: https://cloud.ibm.com
- IBM GitHub Account: https://github.ibm.com/
- Udemy Course - Git for Geeks: Quick Git Training for Developers (3.5 Hours)
- Fork the Git repository: https://github.ibm.com/jeffreys/full-stack-coding-challenge

Code stubs are included.

Deliverables

Week	Udemy	Hours	Task
1	• Docker for the Absolute Beginner – Hands On – DevOps • Microservices with Spring Cloud	3.5 4.5	• Create MySQL Docker Images (Login and Employee) • Complete Login Service and Containerize • Complete Employee Service and Containerize • Create and Run in Docker Images • Test Docker Images (Postman and MySQL WorkBench)
2	• Microservices with Spring Cloud (continued) **Start:** • Full Stack: Angular and Spring Boot (12.5 Hours)	4.5	• Implement Eureka Discovery Service and Containerize • Implement Zuul API Gateway Service and Containerize • Validate Eureka Discover Service identified: Login, Employee, and API Gateway Services. • Test Services via Zuul API Gateway
3 – 4	• Full Stack: Angular and Spring Boot • Go Java Full Stack with Spring Boot and React	12.5 11.5	• Implement and Containerize Angular UI • Ensure screen requirements are implemented • Test Angular UI against service components • Repeat above steps for the React UI
5 – 6	**DEMONSTRATIONS** *UI and Service components must be running in Docker containers.*		

Deliverables – Week 1

Databases Implementation

- Create MySQL Docker Images:
  - Login DB
  - Employee DB

- Run Images in Docker Container

- Test Connectivity to DBs with MySQL Workbench

Services

- Implement Login and Employee Services

- Create Docker Images per Service

- Run services in Docker Container

- Test services with Postman or tool of choice

Push Images to Docker Hub

Weekly deliverables should be committed to your code repository and added to the deployment of the entire application stack using Docker Compose.

Udemy Courses:
- Docker for the Absolute Beginners – Hands On – DevOps
- Microservices with Spring Cloud

Deliverables – Week 2

API Gateway and Discovery

- Implement Discovery Service

- Implement API Gateway Service and configure to interface with the Discovery Service

- Modify Login and Employee Services to interface with:

  – Discovery Service

  – API Gateway

- Create/Modify Docker Images of Services

- Run services in Docker containers

- Test services with Postman or tool of choice

Push Docker images to Docker Hub

Weekly deliverables should be committed to your code repository and added to the deployment of the entire application stack using Docker Compose.

Udemy Courses:
- Microservices with Spring Cloud (continued)

Deliverables – Weeks 3 to 4

UI Development

- Implement UI using Angular

- Create Docker Image of UI

- Run Image in Container

- Implement UI using ReactJS

- Create Docker Image of UI

- Run Image in Container

Push Docker images to Docker Hub

At this point, all code should be committed to your code repository. The entire application stack (DB, Services, and UIs) should be deployed by running Docker images and using Docker Compose.

Udemy Courses:
- Full Stack: Angular and Spring Boot (12.5 Hours)
- Go Java Full Stack with Spring Boot and React (11.5 Hours)

Bonus Deliverables

IBM Cloud Deliverable

By now, you should have successfully accomplished delivering a full-stack application. Now let's **Journey to the Cloud**.

- Create your IBM Cloud Account

- Install IBM Cloud CLI

- Upload your Docker Images to the IBM Cloud Image Registry

- Create a free cluster on IBM Cloud

- Deploy your application to the IBM Cloud using Kubernetes.

Udemy Courses:
-   Kubernetes for the Absolute Beginner – Hands-on

Jenkins Deliverable

With Jenkins, build a pipeline to:

- Checkout code base from Git Repository

- Compile Code (Java projects)

- Build Docker Images from code base

- Launch application with Docker Compose

  - Databases

  - Services

  - UI (Angular or React)

Udemy Courses:
- Jenkins 2 Bootcamp: Fully Automate Builds to Deployment 2019

Other Tutorials and References

Angular: https://angular.io/tutorial/

ReactJS: https://reactjs.org/

Spring Initializer: https://start.spring.io/

Spring Boot: https://spring.io/projects/spring-boot/

Tutorials Point: http://www.tutorialspoint.com/

W3 Schools: https://w3schools.com/