# Case Study Outline

| Ordinal | Project Phase | Objective | Purpose | Description |
|---|---|---|---|---|
| 1 | Ideation | To create an idea to begin to design | To allow time to explore potential application concepts. | Students will use time to sketch ideas of what may be a feasible application to implement before the end of this course. |
| 2 | Concept Approval | To pitch the idea created during ideation to an instructor. | To allow instructor to consider the feasibility of the project. | Students will use their instructor as a resource to decide the feasibility of their application. |
| 3 | Wireframe Design | To create a graphical prototype of the expected application | To allow students to first sketch and design before implementing | Students will use Balsamiq.com to generate a Wireframe to mock the design of their front end application |
| 4 | Backend Web Service: Data Models & UML | To create a logical prototype of the expected back-end application implementation | To create containers for each of the Model objects to be persisted in the database | Students will use Spring to annotate each respective Model class. |
| 5 | Backend Web Service: Controller Logic | To create a logical implementation of the expected back-end controller layer of the application | To create a controller which can receive and respond to requests from a front end application | Students will use Spring to annotate each respective Controller class. |
| 6 | Backend Web Service: Service Logic | To create a logical implementation of the expected back-end service layer of the application | To create a service which is used by a controller to handle the business logic of web-transactions | Students will use Spring to annotate each respective Service class. |
| 7 | Front End: Implementation | To create an implementation of the proposed Wireframe design | To create a client-facing application that enables interactions with the backend web service anc controller | Students will use Angular, React, or Vue to implement a client-facing application which crosses origins with their webserver. |
| 8 | Frontend Web Service: Data Models & UML | To create a front-end representation of the expected back-end service layer's data. | To create JSON models representative of the requests being sent to the webserver | Students will use Angular to model the RESTful resources being transacted between the front and backend applications. |
| 9 | Frontend Web Service: Routing Requests | To create a front-end router which routes requests from the client | to route incoming requests from backend controller to respective front-end controller endpoint. | Students will use in-app routing to navigate to views. |
| 10 | Database Integration | To create a structured set of data | to persist data across application instances. | Students will use Heroku to create and configure a live database instance. |
| 11 | Server Cloud Deployment | To expose the web-server for public use. | To allow students to have a live web server which can be interacted with by public clients | Students will deploy their Spring boot applications to a persistent EC2 Instance using Heroku. |
| 12 | Application Cloud Deployment | To expose the web-application for public use. | To allow students to have a live web application which can be viewed by a public client | Students will deploy their Front-end applications to a persistent container using Netlify. |