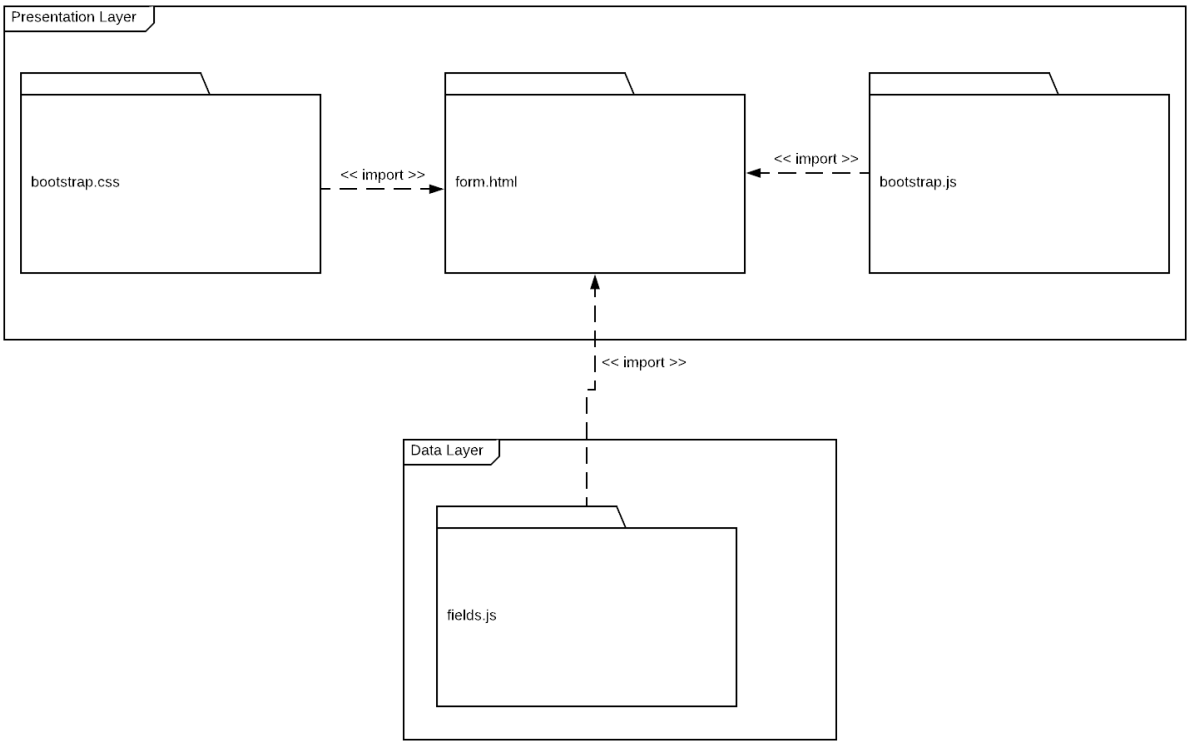
Nicholas Anderson
Carli Martinez
Brandon Thomas
Ryan Wallace

Shift Creator
Team Project – D.4 Design
CS 386 – Software Engineering
Spring 2018
Prof. Dr. Marco Gerosa

1. Description

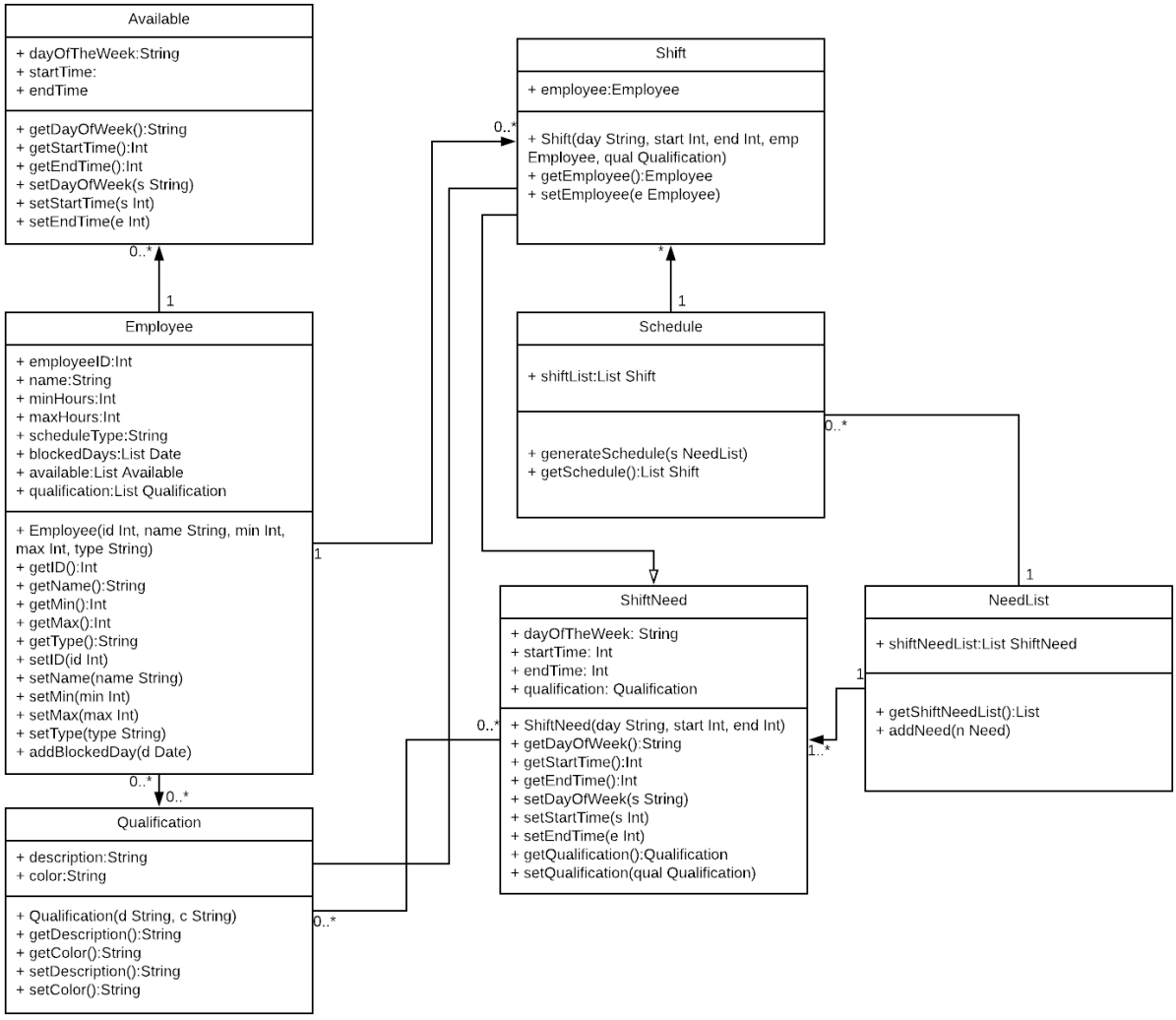
Going to website, choosing the hours of business of when it is open. Putting names of employee and days they are working. It creates a list of the week with each employee shift in each day.

2. Architecture



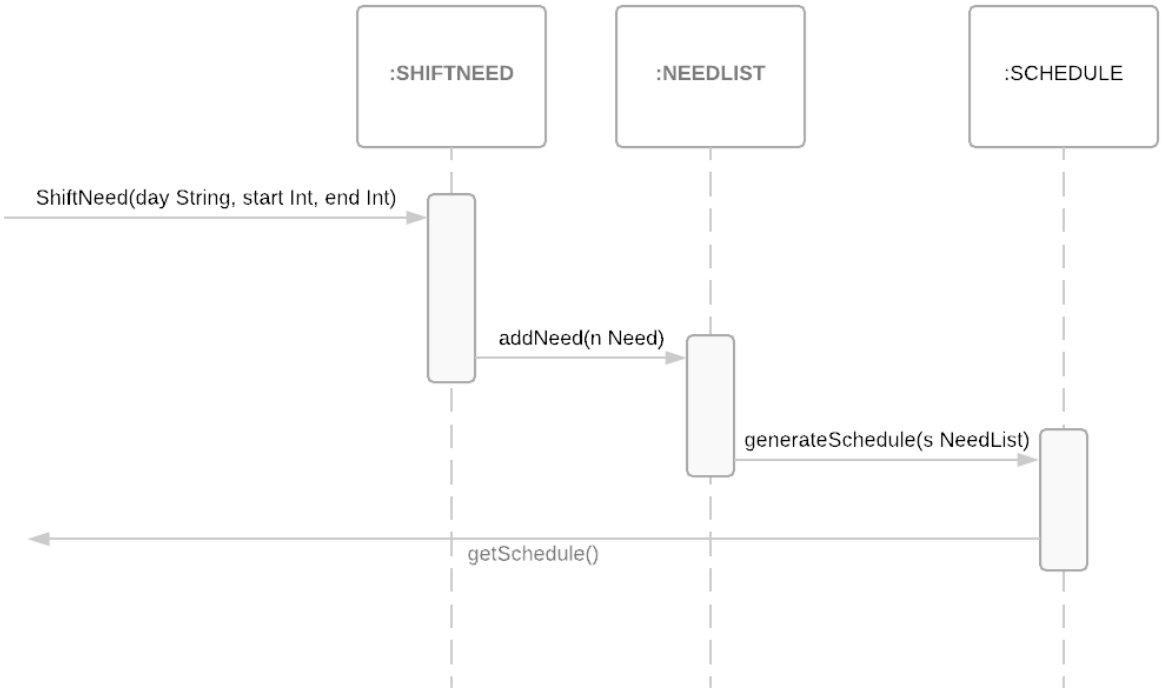
The lack of the ability in javascript to import other javascript files largely prevents separation into packages. ShiftCreator being a web application has standard separation of html and css in the presentation layer to the user. Bootstrap files are providing the theming for the front end. All data and manipulation is being performed in the classes provided by fields.js.

3. Class Diagram



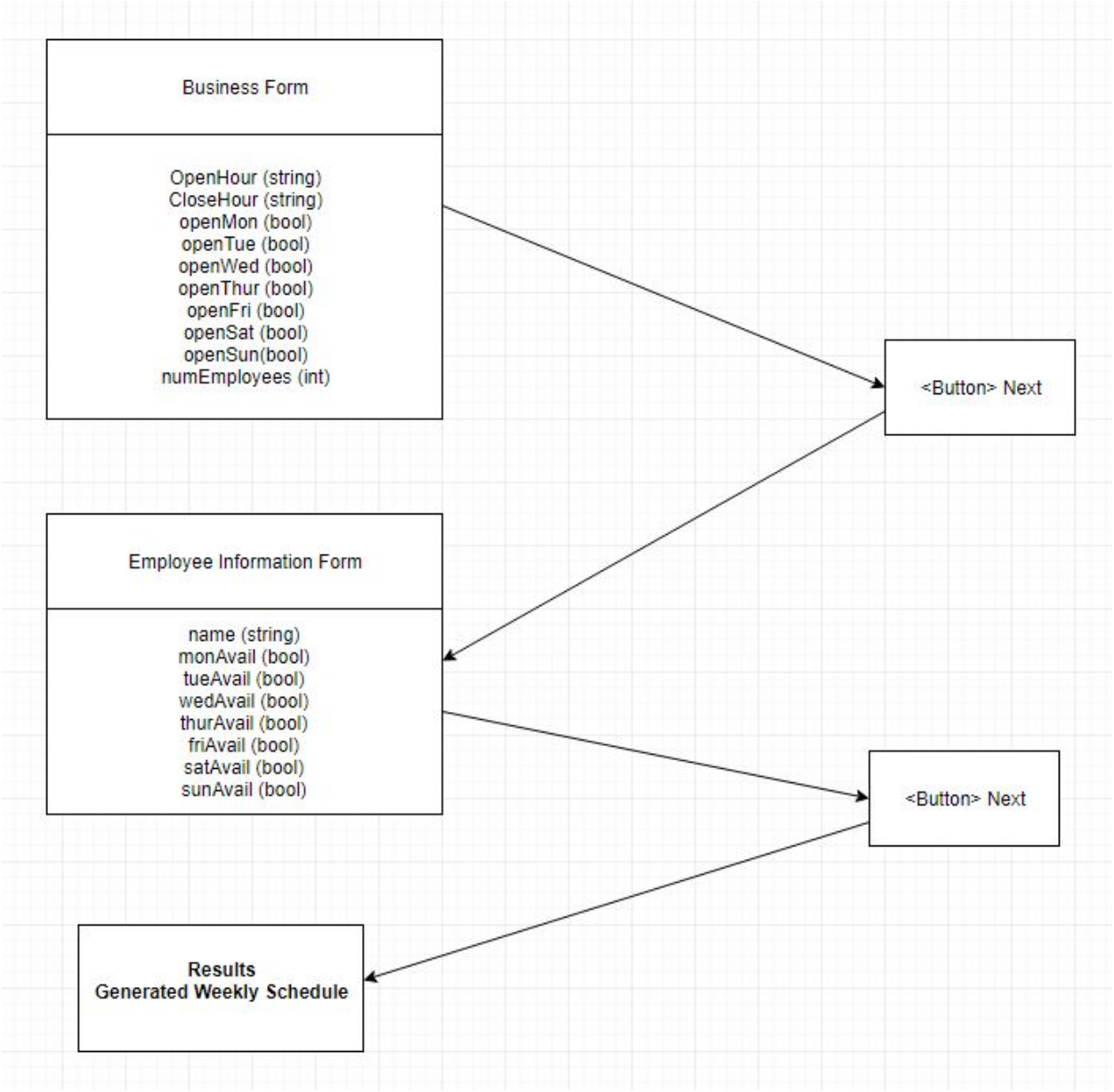
4. Sequence Diagram

Use Case: New business volume requires another shift to be created and requires the schedule to be re-generated.

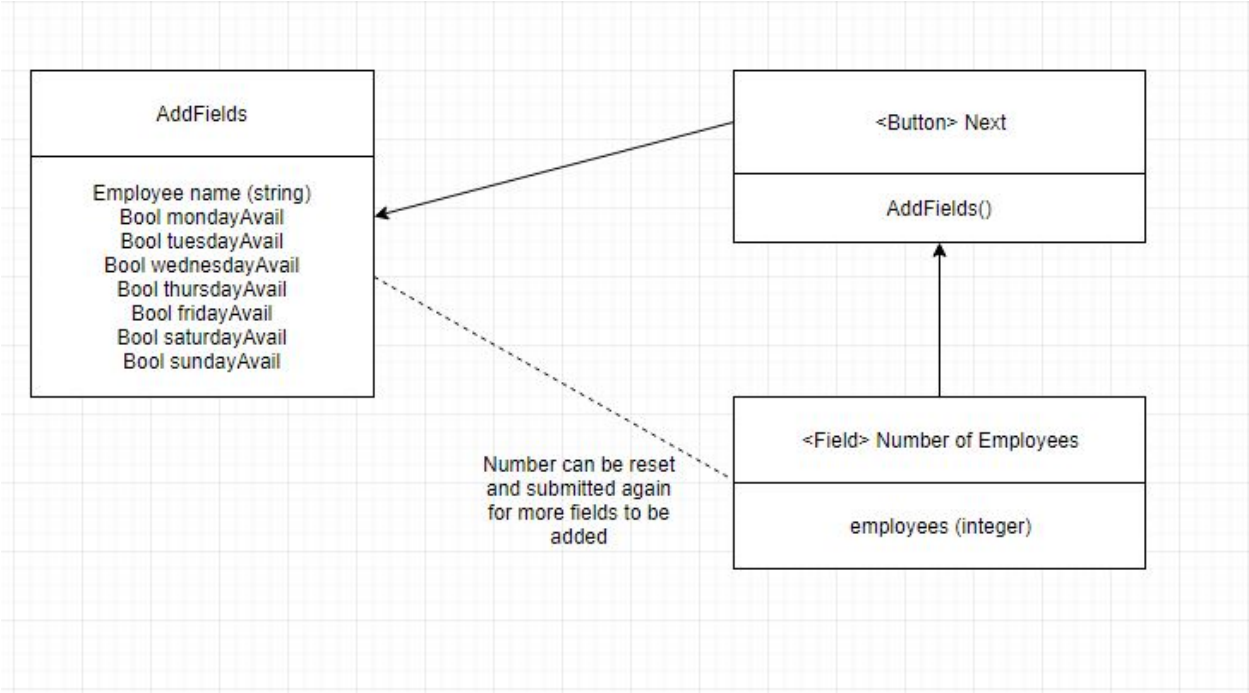


5. Design Patterns

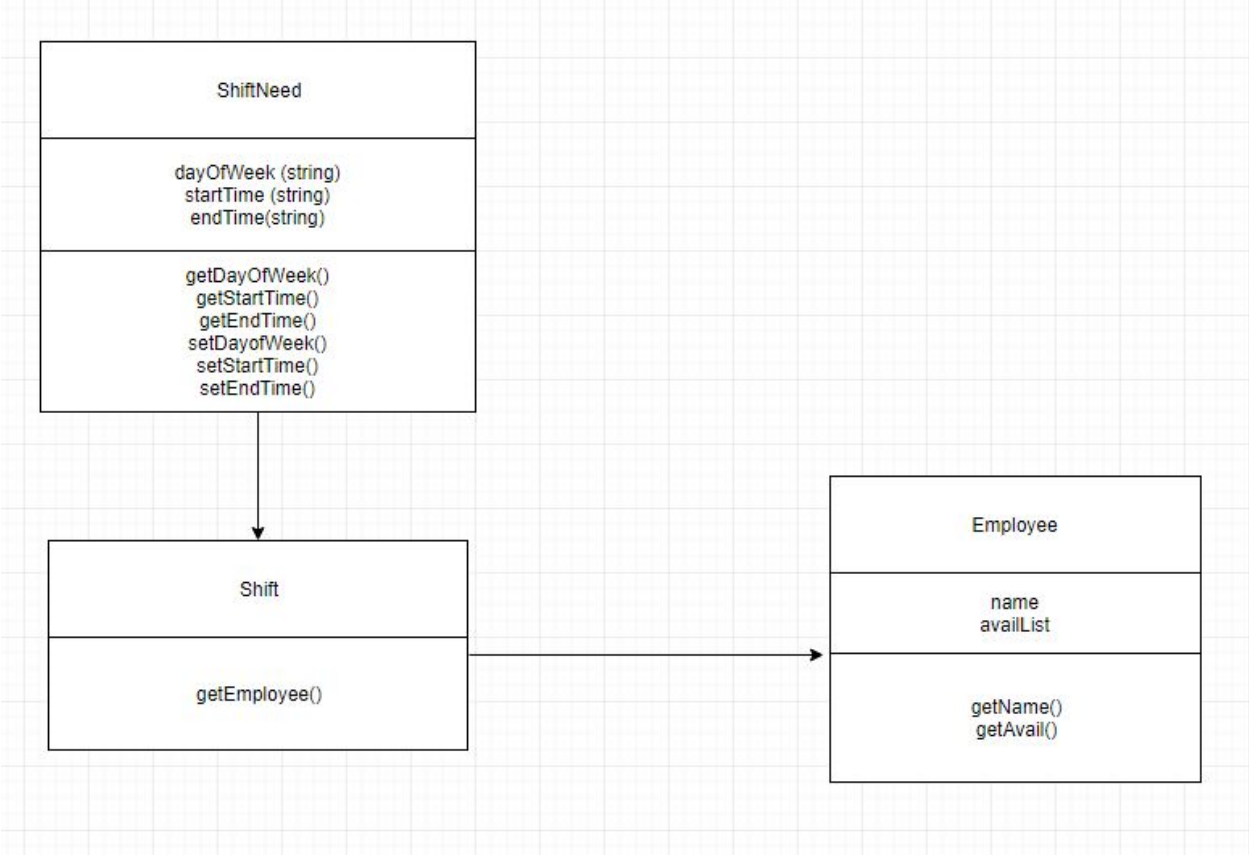
1. **Behavioral -** This design pattern represents the behavior of the input experience. The user competes part of the form, clicks ‘next’, some code processes part or all of what was just entered, the next part of the form is generated, or if the user is done with the form, the results are generated.
   https://github.com/csmartinez/job-scheduler/blob/master/js/fields.js



2. **Creational -** This design pattern represents a piece within the behavioral experience diagram above. More specifically, it shows how fields are dynamically added within the webpage based on a number the user enters before clicking the ‘next’ button. It creates a smooth and flexible experience, allowing the user to go back and regenerate more fields with ease, without causing them to lose their existing input. https://github.com/csmartinez/job-scheduler/blob/master/js/fields.js



3. **Structural -** This is the basic structure of how we generate schedules. Based on what the user enters about shifts, a shift need is created to represent and open/empty block of time. Structurally, a shift it is a type of shift need that has the ability to call getEmployee() to have an employee assigned to it.
https://github.com/csmartinez/job-scheduler/blob/master/js/fields.js



6. Design Principles

Using the SOLID principle from class, starting with single responsibility principle which our website is doing one thing with each function. The next principle is the open and closed principle of where we simplifying our cases to class. The liskov substitution principle, have not needed it at this part of project. The Dependency Inversion principle where using our site to have source for the script to run the schedule part of the project.

7. Group Participation

Nicholas Anderson - Updated class diagram and created sequence diagram. 25%
Carli Martinez - Created and set up D4 document. Created design patterns UML diagram with descriptions. Published our project to the world wide web. 25%
Brandon Thomas - Description and Design Principles 25%
Ryan Wallace - Ran weekly meeting, helped fill in D.4 25%