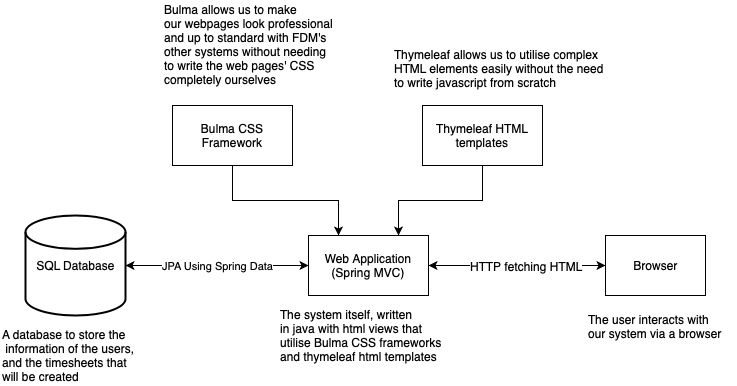
**FDM Timesheets - Architecture and Design**

**Task 1 - Architecture Decisions**

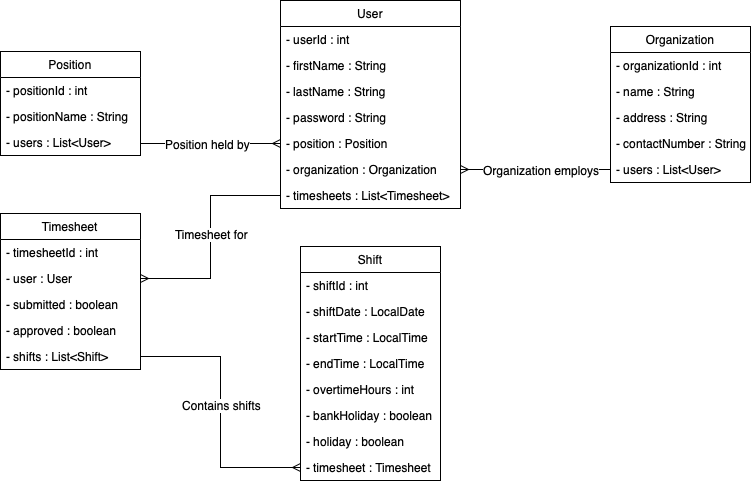
****

We decided to construct our project’s architecture as above as it suited the needs of an online timesheet submission system well. We chose to create an SQL database to store the necessary information for the users of the system. An SQL database is persistent, and is easy to add data to and read data from using hibernate.

We used a number of open source resources to improve the look and functionality of our web application. One of which was Bulma, a CSS framework that allowed us to make professional-looking web pages far more quickly and efficiently than if we were to write the CSS from scratch. This allowed us to create a product that resembled the design of the company’s website.

Thymeleaf was added to our HTML pages in order to manipulate the data that would be sent and retrieved from the database more conveniently. Thymeleaf would also allow us to more easily implement impressive-looking page elements, such as graphs and inputs for the timesheets without the need to write complicated JavaScript ourselves. Due to the nature of the application, it was decided that a browser would be the best way for users to access the application.

**Task 2 - Design Decisions**



Shown here is a class diagram showing how the various classes interact with each other and how all the data is defined and stored.

* Each Timesheet object has seven instances of the Shift object (representing each day of the week) which are stored in ***ArrayList<Shift>***.
* The User object has **ArrayList<Timesheet>** where each instance of Timesheet is stored. The user that the shift belongs to is also recorded along with their required information, and can have many timesheets stored that belong to them. This is necessary as a user will create a new timesheet every week.
* Each User will also be working at a certain Organization. Each Organization will store the information of the Users that it work for it in **ArrayList<User>**.

The Shift class was initially designed to have the variables ***startTime*** and ***endTime*** as ***java.util.LocalDateTime*** type. This raised a problem where a Shift (which represents a single working day) could span over multiple days. To resolve this, we decided to use the newer ***java.time*** library’s ***LocalDate*** and ***LocalTime*** types. We added a separate variable ***shiftDate*** of type *LocalDate* which only stored a date. We also changed *startTime* and *endTime* variables to type *LocalTime*types which only stored time, preventing a Shift from spanning multiple days.