Architecture/Design Document

Dwebble
Classroom Scheduling Application

Group 12
Ibrahim Abu Anz
Elias Julian Marko Garcia
Aiden Cornstubble

Table of Contents

Change History

Version: 1
Modifier: Ibrahim Abu Anz
Date: 04/02/2020

---

1 Introduction

The primary purpose of the project on developing the Classroom Scheduling application is to reduce the human-effort to class schedules manually for an educational institute or a university. While making a classroom schedule, there are several rules, constraints, preferences to be followed and it becomes a complex and lengthy job for a human to prepare accurate schedules. So, the web based Classroom scheduling application is being developed to make the process easier, faster, more accurate, and user-friendly. Thus, it will help significantly in saving time, effort, and cost to make classroom schedules.

This document aims at presenting explanations about the architecture and design of the Classroom Scheduling application to address the concerns and interests of various stakeholder of the project. The major stakeholders of the project are,

- **End-users –** The Classroom Scheduling application will be used by instructors or teachers, and university staffs responsible to make the schedules and check the schedules. Hence, they have some functional and non-functional requirements from the system. For example, staffs would like to add rules, constraints for classes and schedules, teachers or instructors can add their preferences and availability details into the system. The system must automatically product a set of prioritized schedules for an admin staff who will be responsible to select a suitable schedule. Then the system will send the same to all instructors according to their personalized schedules.
- **Project Manager –** The project manager is responsible to orchestrate different aspects like cost, time, resources, and scope of the project to complete it successfully. A project manager is also responsible for overall planning, and execution of the project, controlling and monitoring the project progress, and to manager project team members. A project manager should be responsible to handle risks and communication required for the project.
- **Developer –** The developer is responsible for the coding, and unit testing part of the application. The developer follows the designs prepared by the designers and must adhere to the standards of coding and testing.
- **System analyst –** The system analysist is responsible to analyze the requirements of the application, and preparing the system requirement specification for the design, development, and validation of the system.
- **Testers -** Testers are responsible to prepare the test plan for the system and to carry out integration testing and system testing. They are also responsible for the verification and validation of the system against the system requirements specifications.
- **Designers -** Designers are responsible for designing the user interface, user interactions of the application.
- **Technical writers –** They are responsible for preparing the manuals, and training modules for the end users.

- **IT Staffs –** They are responsible for general IT jobs like setting up the development environment, helping in troubleshooting the IT infrastructure and network connectivity required for the project. They are also responsible for the maintenance tasks required for the system to keep it operational after the implementation.

System Architecture and Design document has the primary aim to hid the underlying complexities of a system and provides different abstraction levels or views to the different stakeholders so that they can understand the system from their perspectives. There are different diagrams to model functional and non-functional requirements of a system from different perspectives.

1. **Logical View**
   It is focused on the implementation of various functional requirements of the concerned system for different stakeholders. Usually, the view is described using several building blocks by hiding the underlying structures of processes, codes and other complexities. Building blocks are for subsystems, classes, components and so on. UML diagrams like class diagram, sequence diagrams are used to present the logical view of a system.
2. **Process View**
   It is focused on giving runtime view of a system. At the runtime, there are various processes and threads interacting with each other in the system to execute various system operations. It also helps in understanding the scope of various parallel processes in the system competing for resources.
3. **Development View**
   It is focused on the implementation and deployment part of the system. The view gives information about various static artefacts like graphics, interfaces, source code, packages, of a system. It also gives the idea of implementation of these static items into the physical system.
4. **Use Case View**
   It is the central theme of the overall architectural view of a system. Other than giving information about the use cases of a system, it also gives information about validation of the use cases. Use case view gives information about the behavioral aspects of the system through the descriptions of different functionalities.

2 Design Goals

There may be multiple designs possible for a system. And the value of a design is determined by the priorities of various stakeholders. For example, under a circumstance, end-users may require basic functionalities and minimalistic design. And in an another circumstance, end users of the same system required advanced functionalities rather than minimalistic design.
However, the design priorities for the Classroom Scheduling Application are:

- The design should be minimalistic

- The design should ask for minimal and guided inputs from end users
- The design should provide feedbacks to users
- The design should be consistent and accessible for all users
- The design should adhere to the roles of end users
- The design should high the complexity of the application from the end users

3 System Behavior

Understanding system behavior gives a 'black-box' view of the system where the underlying codes, and processes don't come into the picture. It gives a description or understanding of what a system does rather than giving information about how the system does it.
The Class Scheduling application should allow support two types of end users depending on the roles, admin staff and instructors. So, there will be options to either create a schedule or to check a schedule. An admin staff can add rules and constraints. Then the system will create multiple schedules and will allow the admin staff to select one. And on the other hand, an instructor can add preferences and availability details, and can download personalized scheduled as selected by an admin staff.

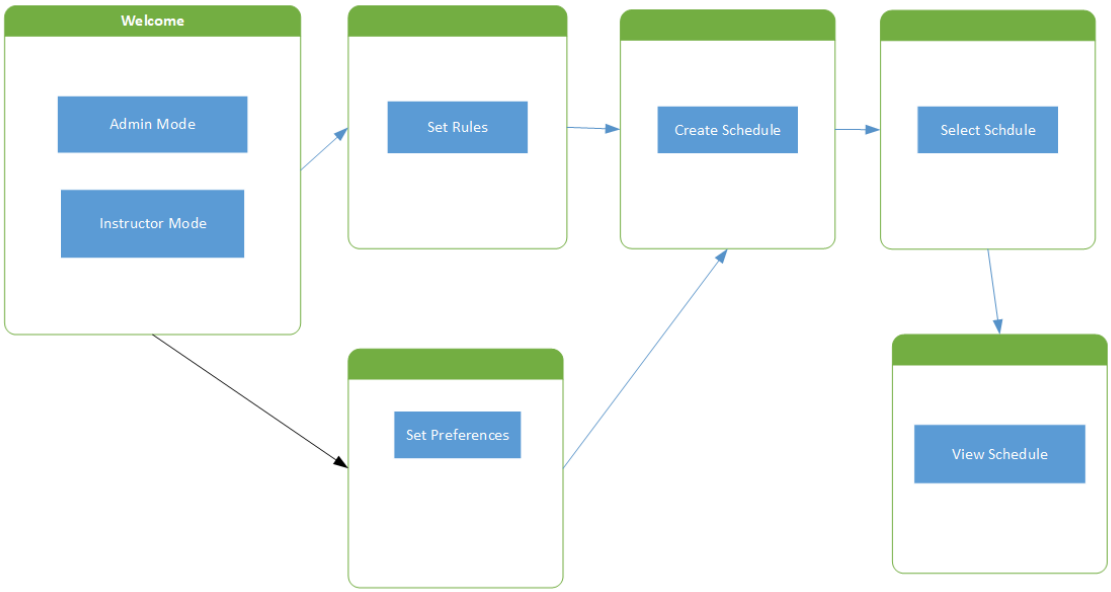

Figure 1: System Behaviors

4 Logical View

The logical view of the Class Scheduling application has been captured through the high-level, mid-level architectural design and detailed class design. At first the high-level

design gives an understanding of the three major components of the system, then the mid-level design helps to understand the connections and communication between the components described in the high-level design. And the detailed class design gives an understanding of the underlying system domain and classes of the system.

4.1 High-Level Design (Architecture)

The high-level view or architecture consists of 3 major components:

- **Front-end Client**
  The frond-end client can be a web client accessed from a mobile device or a PC. It allows an end user to interact with the Classroom Schedule application through a web browser. The web client requests to access data from the database through the web application. And end-users also get results from the web application on the front end client.
- **Web Application**
  In the three-tier architecture, the web application consisting of the application logic sits between the Front-end client and the back-end database. It accepts requests coming from the front-end client, decodes the requests, processes it. If necessary, then it communicates with the backend database to fetch data. Then it sends back the results to the front end client.
- **Database Server**
  The database server sits behind the web application. It contains raw data required for the web application. It takes data access requests from the web application, processes the request by running queries on the data and returns the results to the web application.

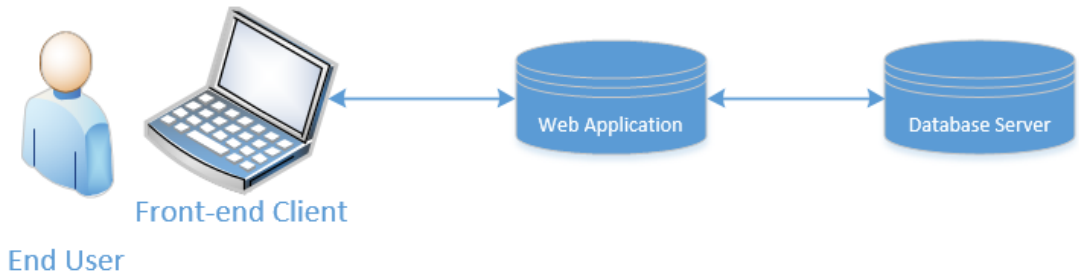

Figure 2: System Architecture

4.2 Mid-Level Design

The mid-level components and the relationships between the components have been shown in the following figure.

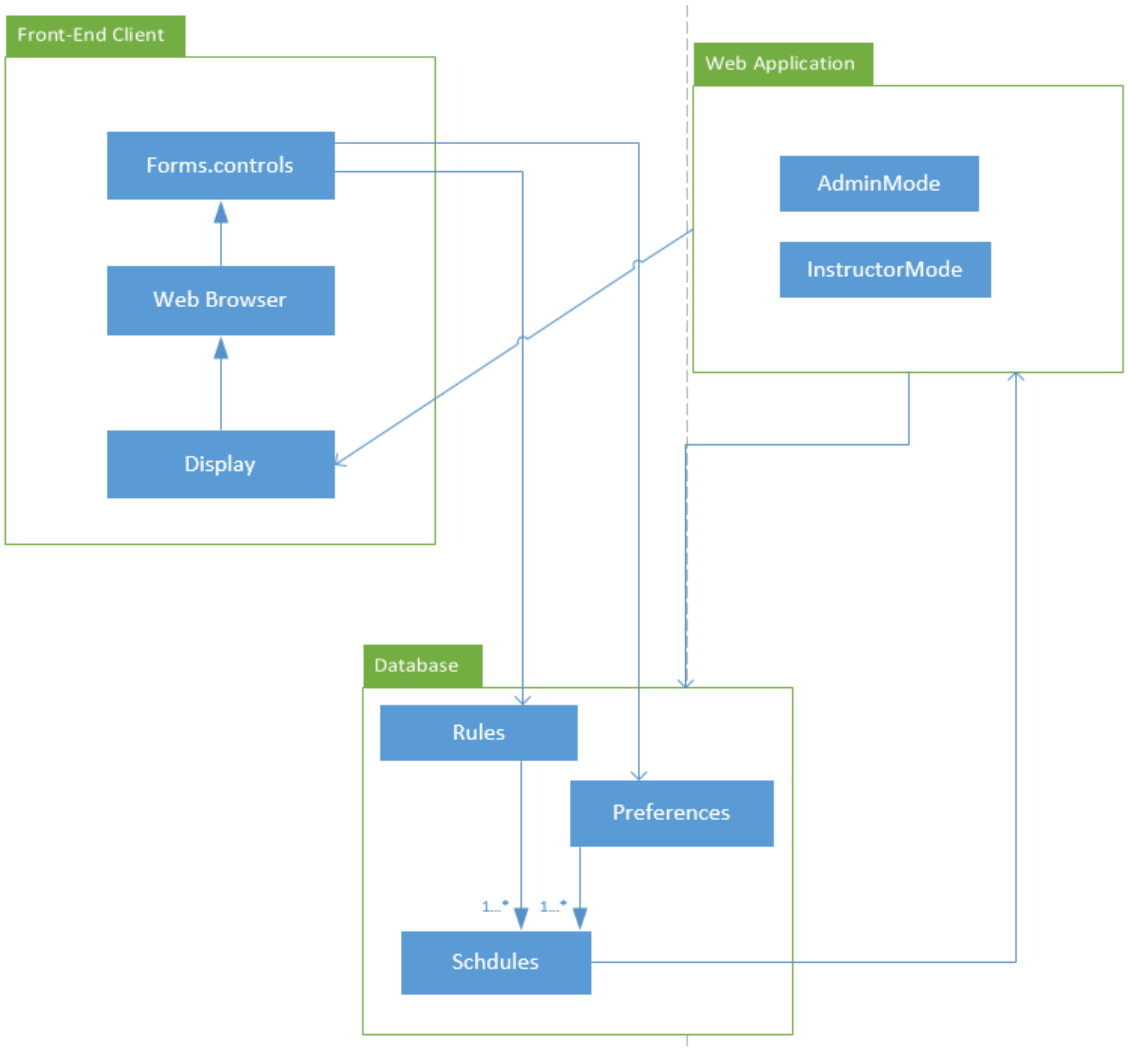

Figure 3: Mid-level components and relationships

The sequence diagram for AdminMode is given below,

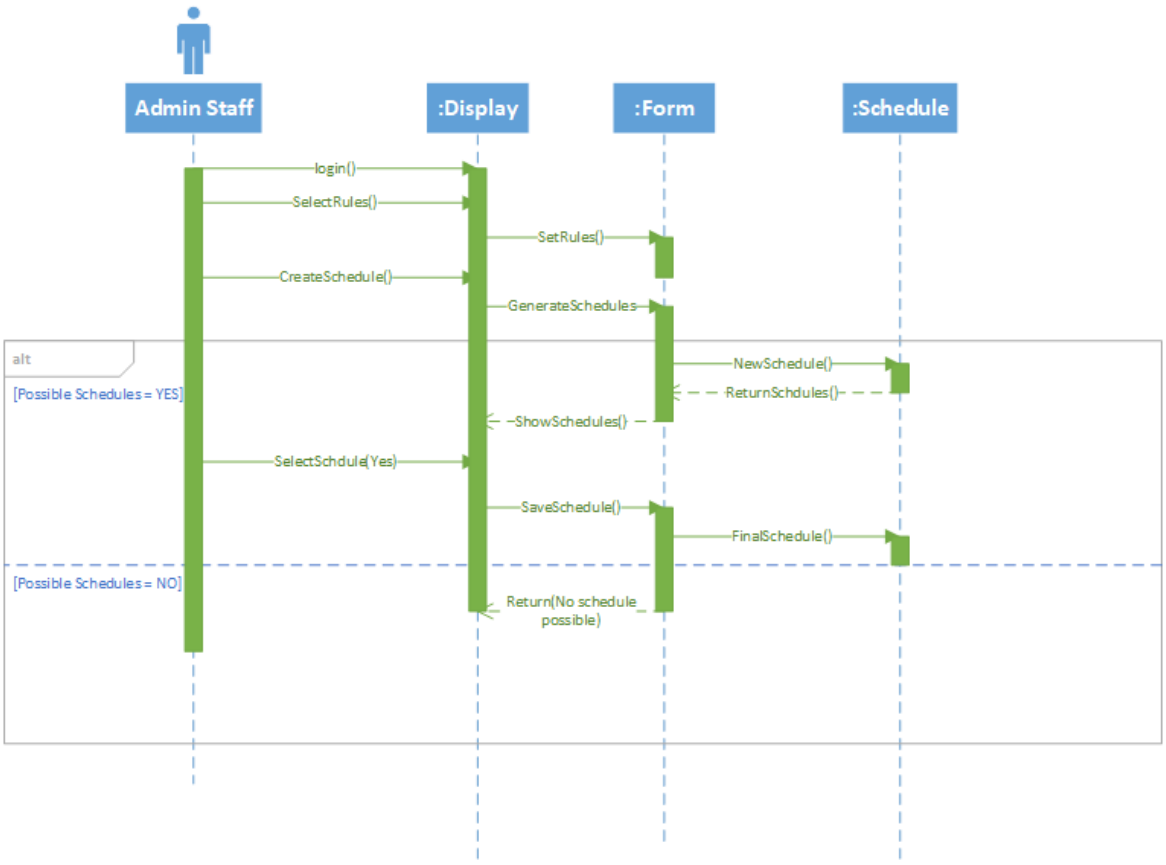

Figure 4: Sequence Diagram for the AdminMode

The sequence diagram for the InstructorMode is given below,

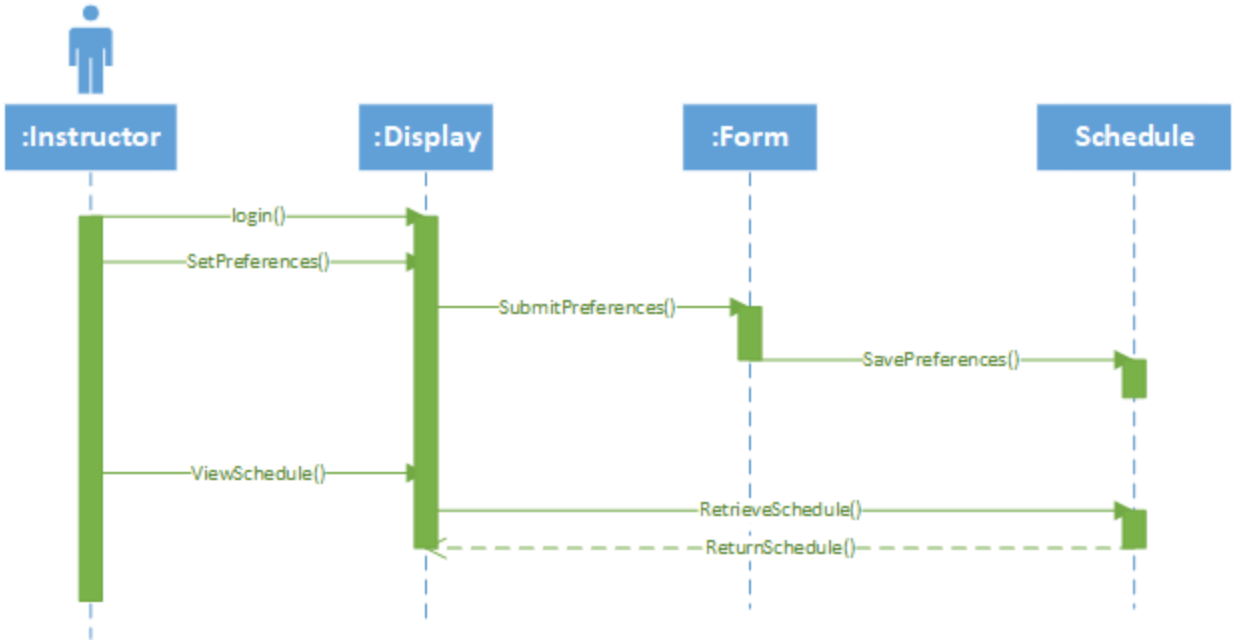

Figure 5: Sequence Diagram for Instructor Mode

4.3 Detailed Class Design

The detailed class diagram is given below. It is a static structural diagram to describe key classes of the underlying system domain. However, these are not the same as the implemented classed during the system development. Rather it helps in showing possible system classes or entities, attributes and operations of the classes. It helps to understand the structure of the system.

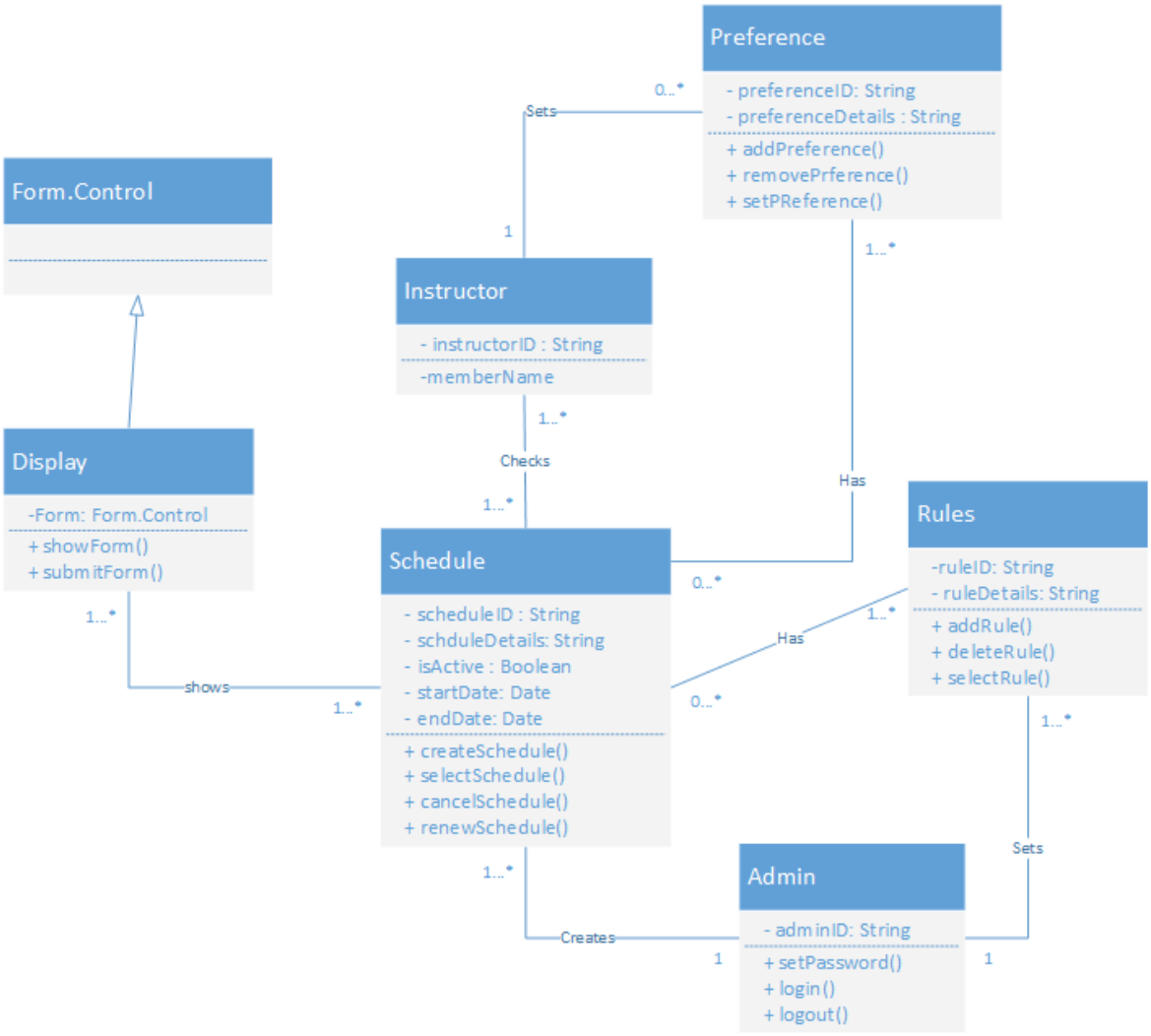

Figure 6: Detailed Class Diagram

5 Process View

The process is creation of schedule by an admin staff, selection of schedule and display of the schedule to instructors. Here an end user can be either an admin staff or an instructor. An end user provides input through input interface available on a front end client. The control then goes to the Class Scheduler application. From there the control reaches to the control output thread to display the result. And finally results or schedules are displayed on the display interface of the front end client.

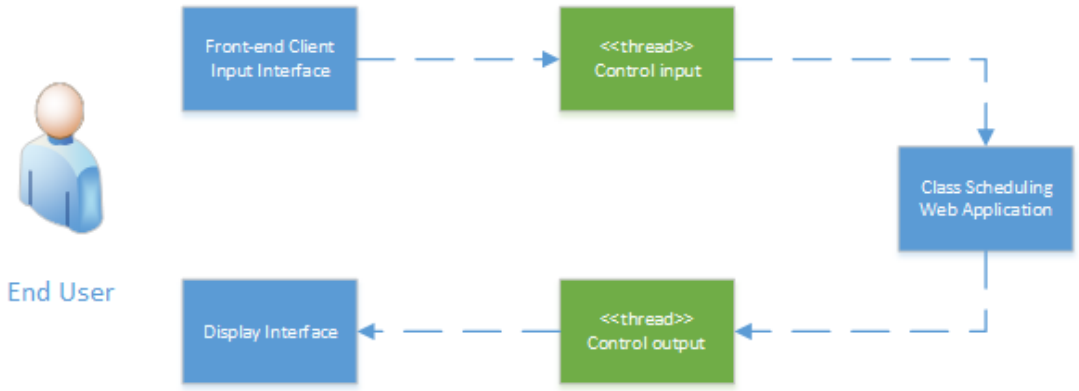

Figure 7: Process Control View

6 Physical View

TBD

7 Use Case View

The use case view with significant use cases are given below. It shows the critical use case of the proposed system, external actors, associations between use cases and actors and dependencies between use cases.

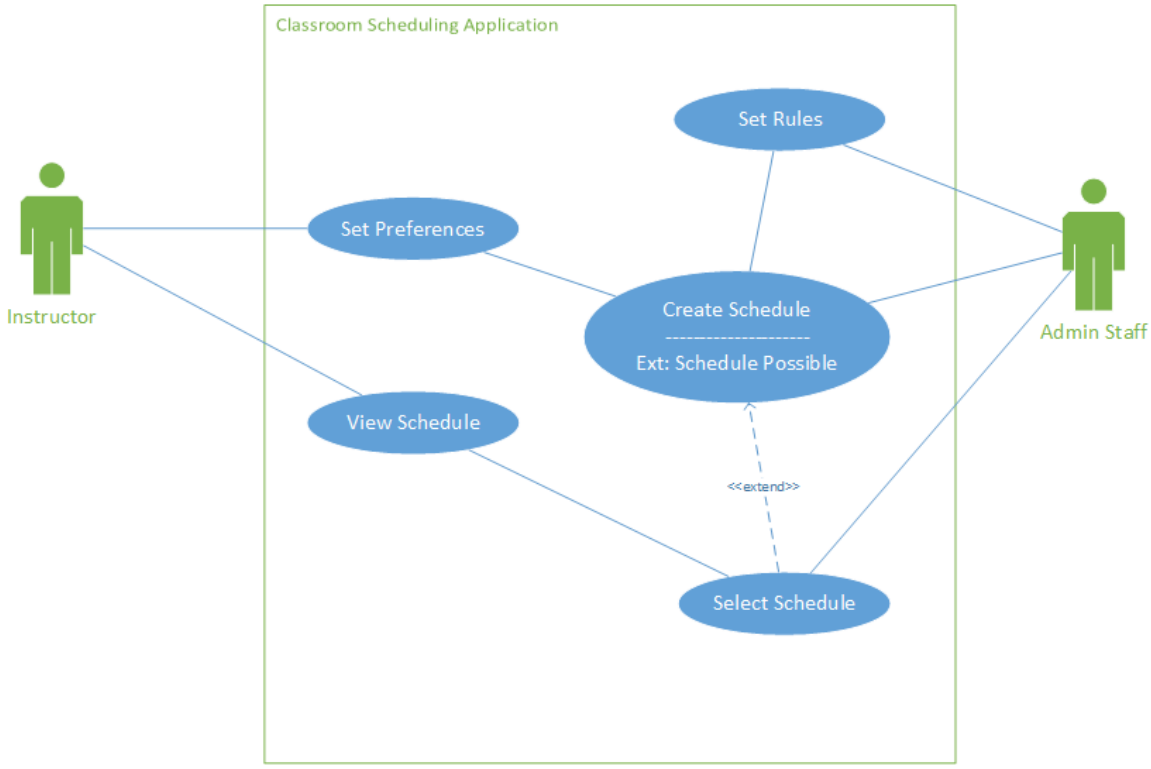

Figure 8: Use Case View