<u>Iteration Plan:</u>
<u>Iteration #1</u>

Group 12

Kyle Nickerson,
Catherine Rendell &
Ben Morrison

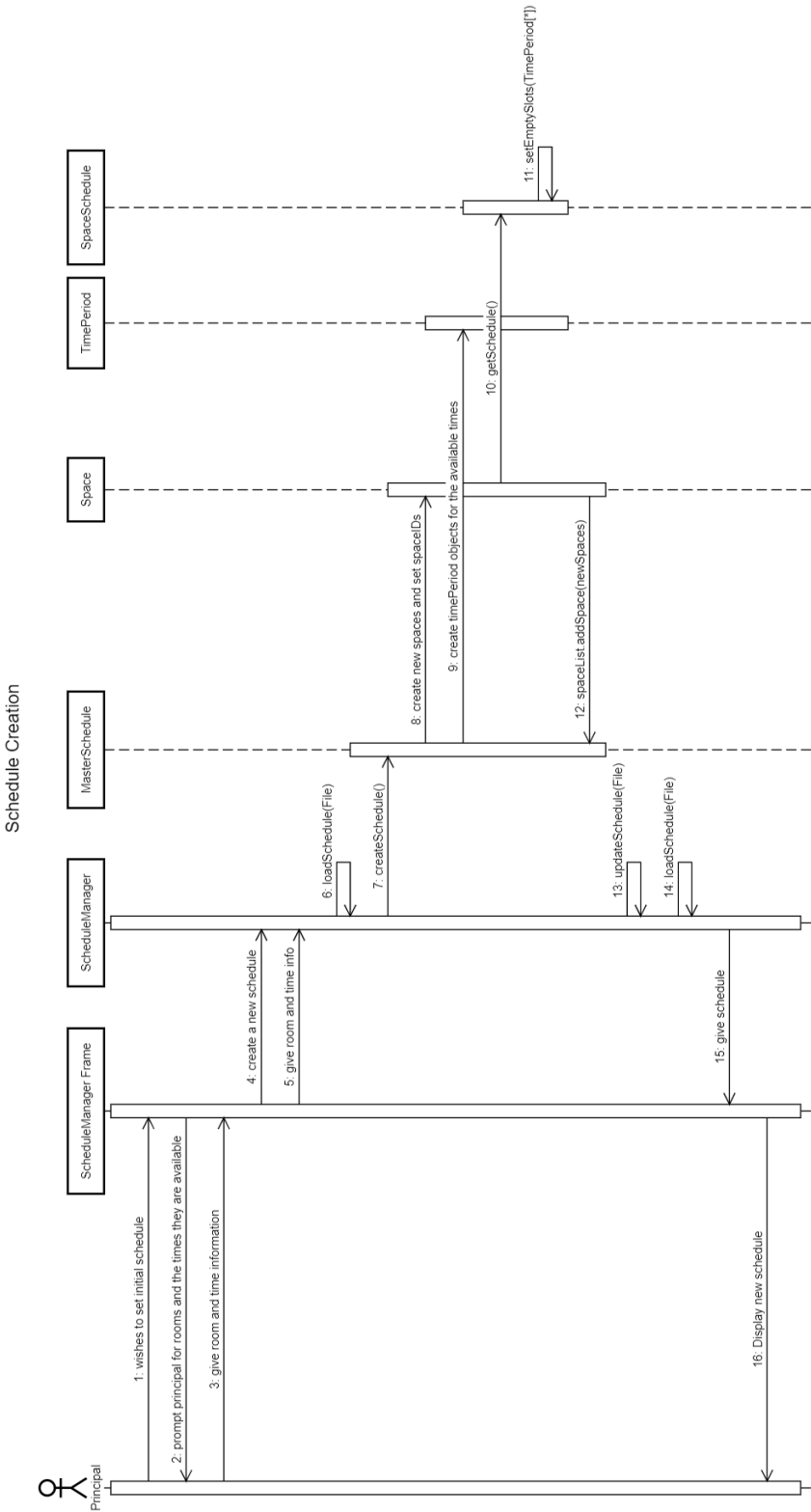Github repository: https://github.com/BigTuna08/cs3716

<u>Plan</u>

We will be using a feature driven development approach, because the 3 main features of our system- the schedule creation, ability to request rooms and the ability to decide requests- are not closely related. Also we have a limited time frame for this iteration, and the feature driven approach will allow us to have working code to show the client faster. In our first iteration we are going to be implementing the schedule creation feature. We chose this because it is a part of the essence of our system. The other two main features deal with requests, but it only makes sense for requests to happen after the schedule has been created (because it contains the info of what rooms/times can be requested).

<u>Use Case: Principal creates schedule</u>

1. Principal starts system

2. System displays list of possible actions

3. Principal selects option to create schedule

4. System prompts principal to enter room names

5. Principal enters room names

6. System prompts principal for each rooms availability

7. Principal enters each rooms availability

8. The system displays blank schedule for confirmation

      8a. Principal confirms schedule is correct

      8b. Principal rejects schedule

            1. Repeat steps 4-8 as needed

9. System stores the empty schedule

Sequence for Schedule Creation

Schedule Creation

	ScheduleManager Frame	ScheduleManager	MasterSchedule	Space	TimePeriod	SpaceSchedule

Principal

1: wishes to set initial schedule

2: prompt principal for rooms and the times they are available

3: give room and time information

4: create a new schedule

5: give room and time info

6: loadSchedule(File)

7: createSchedule()

8: create new spaces and set spaceIDs

9: create timePeriod objects for the available times

10: getSchedule()

11: setEmptySlots(TimePeriod[*])

12: spaceList.addSpace(newSpaces)

13: updateSchedule(File)

14: loadSchedule(File)

15: give schedule

16: Display new schedule

Required Classes

ScheduleManager

createSchedule()
loadSchedule(File)
addSpace(Space)
removeSpace(Space)
addEvent(Event)
removeEvent(Event)
getSchedule()

SpaceSchedule

addEvent(Event)
getEvents():Event[*]
getTimeSlots():TimeSlot[*]
getEmptySlots():TimeSlot[*]

timePeriod
1..*

TimePeriod

dayOfWeek: char
startDate: Date
endDate: Date
startTime: long
endTime: long

duration(): double
overlaps(TimePeriod): boolean

1
manages

1
schedule

MasterSchedule

addSpace(Space)
removeSpace(Space)
addEvent(Event)
removeEvent(Event)

spaceList
*

Space

spaceID: String

getSchedule(): SpaceSchedule
blackout(TimePeriod)
addEvent(Event)