NORTHERN ARIZONA
UNIVERSITY

Nicholas Anderson
Carli Martinez
Brandon Thomas
Ryan Wallace

Shift Creator
Team Project – D.2 Requirements
CS 386 – Software Engineering
Spring 2018
Prof. Dr. Marco Gerosa

1. Positioning

1.1. Problem Statement

The problem of tediously creating a work schedule for employees affects managers; the impact of which is wasting valuable time trying to create a schedule that works for every employee.

1.2. Product Position Statement

For managers who tediously create work schedules for their employees, *ShiftCreator* is a scheduling app that saves you time by generating your employee schedules for you; unlike *Work Schedule Maker*, our product works with everyone to create the perfect schedule.

1.3. Value Proposition

ShiftCreator is a work shift scheduling software that automatically generates the weekly work schedules of multiple employees to save time for the manager and be a more streamlined experience that automatically schedules employees given the necessary information, and if the schedule is not ideal, a new one can be randomly generated in seconds.

2. Stakeholders

Scheduler/Manager - Main user of the software. Will use *ShiftCreator* to enter employee shift requirements (minimum hours, availability), generate schedules, and edit schedule, if needed, before publishing to employees. They will be main source of application feedback during development.

Director/Owner - Decision maker for the purchasing of software for the department or business. This person is not always going to be the scheduler for the organization. They will need to be shown presentations and demonstrations of the prototype during development.

IT Department - For sufficiently large organizations, an IT department will have approval of what software is installed on company computers. Will need to work with representative to ensure *ShiftCreator* complies with any organizational security requirements. Also possibly assist with installation of *ShiftCreator* if only IT department allowed administration rights to install software.

Employees - Shift schedules will be managed by *ShiftCreator*. Could provide feedback during testing as to how the generation of schedules is working for staff.

3. Functional Requirements

1. Enter employee's information including availability, minimum hours, maximum hours, flexible or set schedule, and any qualifications (lead, certifications).
2. Enter business operating hours and required staff per hour of operation.
3. Generate suggested schedule(s) for employees.
4. Edit suggested schedule for employees before publishing.
5. Prevent scheduling employees on vacation or other blocked off days.
6. Display schedule in PDF or spreadsheet format.
7. Switch employee shift assignments after generation at employee shift change request.
8. Highlight different shifts or employee types on schedule.

4. Non-functional requirements

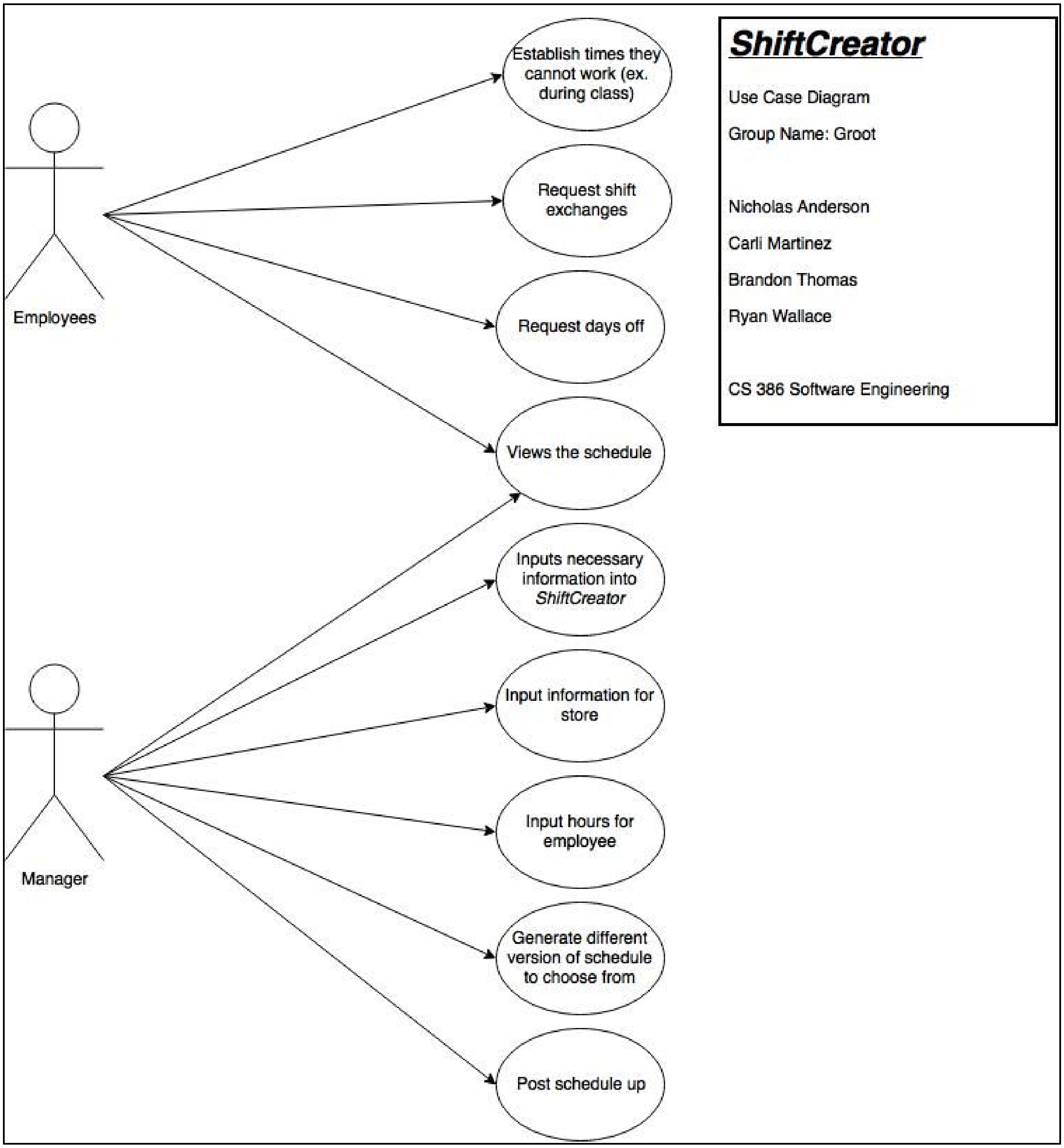
1. Easy of use
2. Flexibility
3. Appealing
4. Reuseable
5. Affordable
6. Portable

5. MVP

A. Inputting information of the store hours
B. Inputting information of employee's Availability
C. Generating a Week Schedule with the data
D. Create different versions
E. Viewable Schedule
F. Print out Schedule

6.1. Use Cases - All

https://drive.google.com/file/d/1qFNKLYPRwhPSUyK_VHHVcP3vZVroXbvw/view?usp=sharing



6.2. Use Case Descriptions - All

Name: Views the schedule

Brief Description: The main goal of the software, viewing the schedule that is generated from the *ShiftCreator* software that will be the weekly schedule of work shifts composed of the employees and their shifts organized by day, Monday through Sunday.

Flow of Events:

1. Employee or manager clicks on a button or opens a PDF or Excel Spreadsheet to view the schedule.
2. Employee looks for their name in the schedule and takes note of when they are scheduled.

Key scenarios: Opening the schedule via web, PDF, excel spreadsheet or e-mail attachment

Special Requirements: A computer, web-access or the appropriate version of Microsoft Excel

Preconditions: Having the correct information used before generation of the schedule, for example the maximum number of hours an employee can work reflected onto the schedule

Post-conditions: A path for the employee to be able to notify the manager if something is wrong and the ability for the manager to change an existing schedule to reflect that

Extension points: No extension points for this use case

Sketch of the corresponding user interface:

Week of February 19th - 25th, 2017

	Mon	**Tue**	**Wed**	**Thur**	**Fri**	**Sat**	**Sun**	**Hours**
Ed	4-7pm		4-7pm	10am-5		3-7pm	3-7pm	19.00
Amy		3-7pm		3-7pm	3-7pm	10am-5	10am-5	22.00
Mark	10am-5	10am-5	10am-5		10am-5			20.00
Janet					4-7pm	4-7pm	4-7pm	16.00

Name: Generating version of weekly Schedule

Brief Description: After inputting information of store hours and employee’s availability, it generates a version of the weekly schedule.

Flow of Events: After generating a schedule, it show it through a viewable document that can then be changed or regenerate another schedule.

Key scenarios: Getting a schedule.

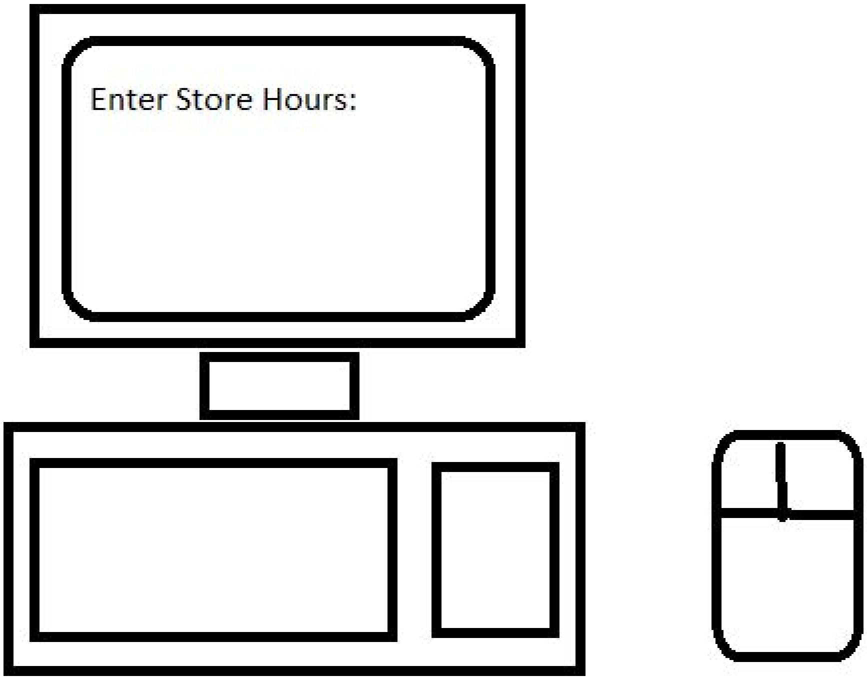
Special Requirements: Needed information for the store and employee availability.

Preconditions: Hours of operation of the store for constraints of developing weekly schedule.

Post-conditions: Saved option of each employee and possible version of schedule.

Extension points: Future or fired employee, hours of operation changing, updating any information dealing with scheduling.

Sketch of the corresponding user interface:



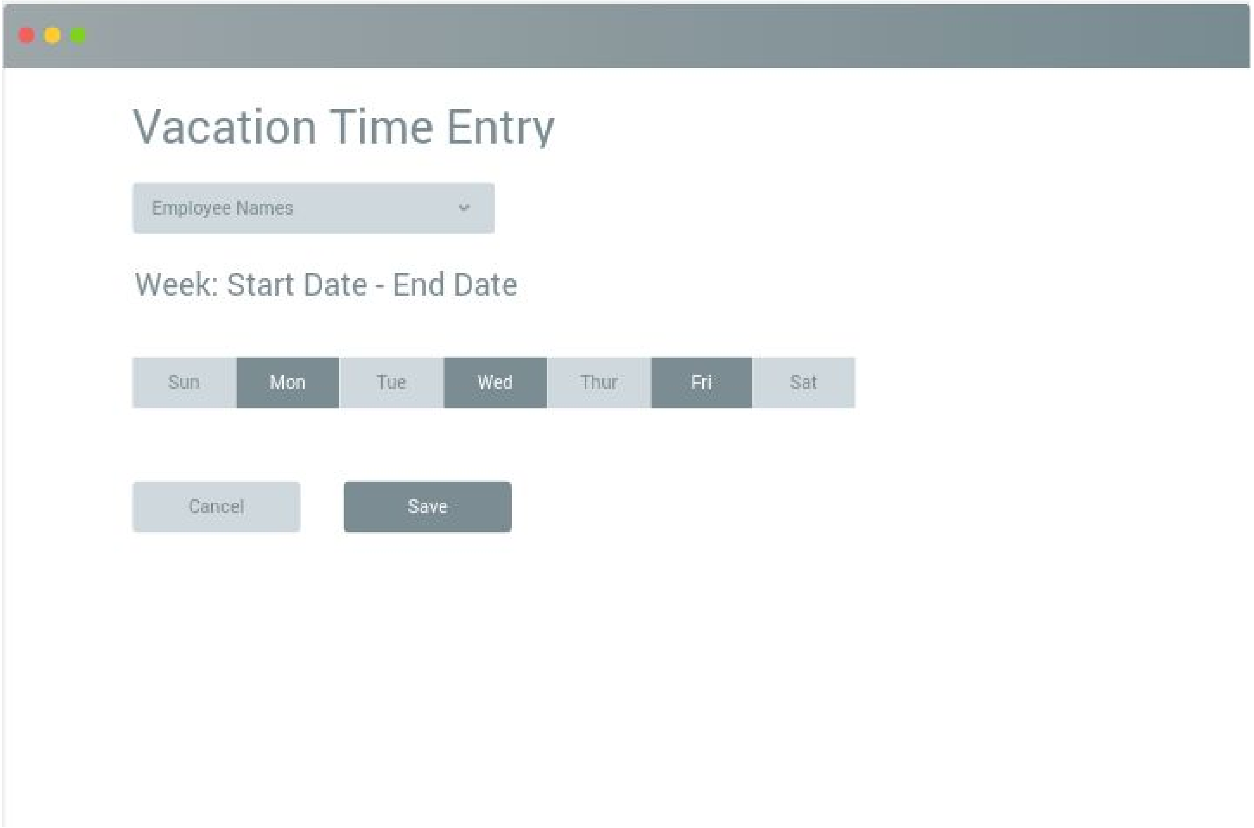
Name: Block employee vacation days
Brief Description: To prevent scheduling employees on days they cannot work outside of their normal availability, the scheduler will enter days employees cannot work.
Flow of Events:

1. The scheduler opens a vacation day entry window.
2. Selects an employee.
3. Selects day employee cannot work.
4. Clicks button to save entry.
5. ShiftCreator stores the employee and vacation days to list.

Key scenarios: Employee cannot work a day next week
Special Requirements: Entry should be easy for the user.
Preconditions: A scheduler has an employee request for a day or days they cannot work.
Post-conditions: *ShiftCreator's* list of vacation days is updated with the new employee and date(s) if saved or remains unchanged if canceled.
Extension points:

- At 3, the scheduler can enter additional days.
- At any point, user can hit close to cancel entry.

Sketch of the corresponding user interface:



Name: Enter business operating hours and staff requirements
Brief Description: Many businesses are not open 24 hours a day. *ShiftCreator* needs to have the hours of operation entered.
Flow of Events:

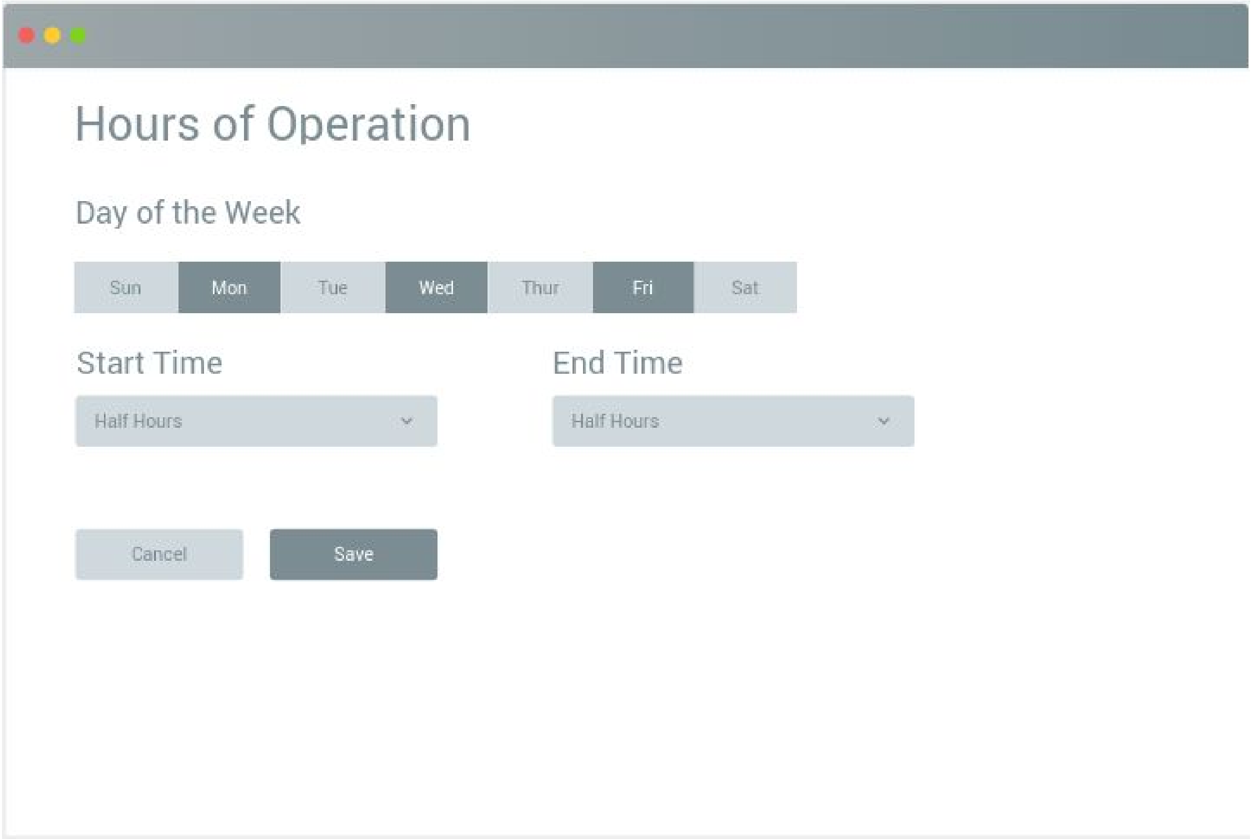
1. Scheduler opens a window displaying business information
2. User selects a day of the week
3. User enters starting time of opening shifts
4. User enters ending time of closing shifts
5. Clicks button to save entry.
6. Shift Creator stores the beginning and end time of scheduled shifts for that day of the week.

Key scenarios: The scheduler business data is being entered for the first time or the business changes hours of operation.

Special Requirements: Entry should be easy for the user.
Preconditions: The scheduler has a list of new business operating times.
Post-conditions: *ShiftCreator* stores new start and end shift times for each day of the week if saved or remains unchanged if canceled.
Extension points:

- At 5, user clicks another day of the week to enter hours of operation for.
- At any point, user can hit close to cancel entry.

Sketch of the corresponding user interface:

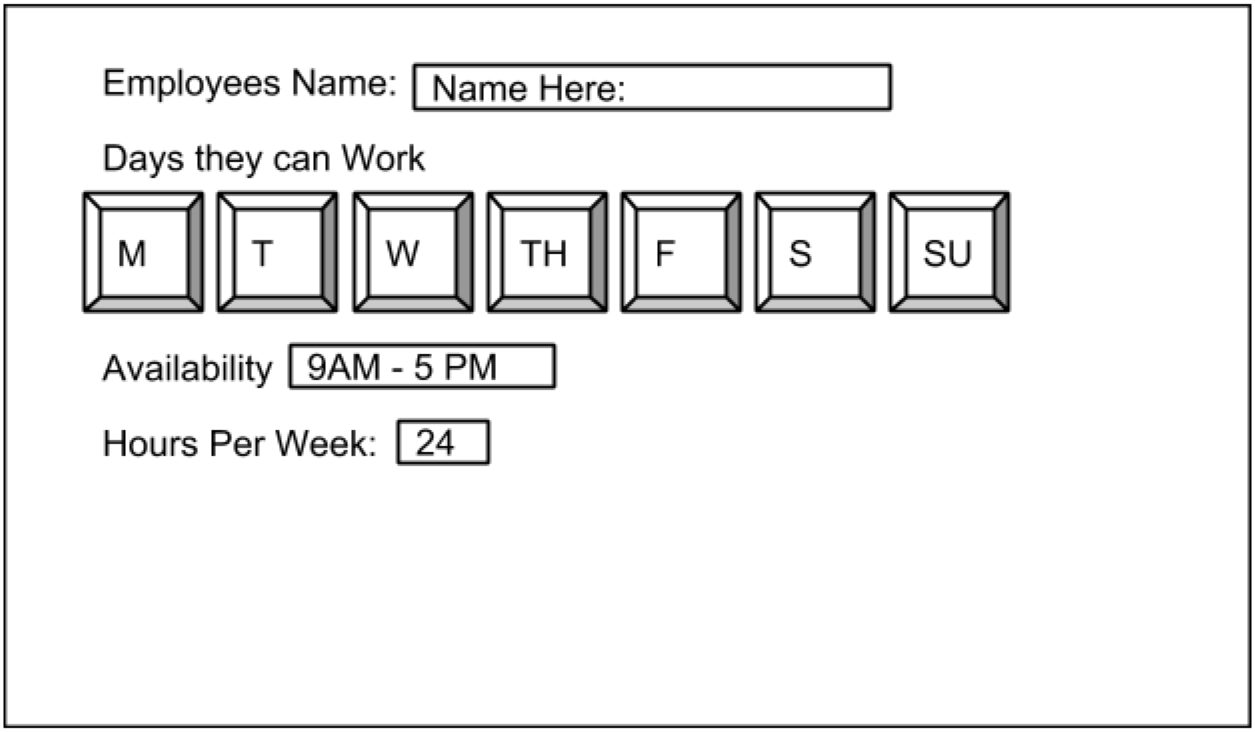


Name: Enter an employee's work availability
Brief Description: *ShiftCreator* needs to know an employee's work availability in order to create the schedule that works for the employee.
Flow of Events:

1. The scheduler opens a employee information/availability entry window
2. Select the days an employee can work. (M, T, W, TH, F, S, SU)
3. Select hours of availability for each work day.
4. Click the save/apply button
5. ShiftCreator stores the employee and their availability for future shift generation

Key scenarios: A employee is being added for the first time or is changing the employees availability.
Special Requirements: Should be easy to use and straightforward.
Preconditions: A textual description that defines a constraint on the system when the use case starts
Post-conditions: ShiftCreator has a list of the employees and their availability

Sketch of the corresponding user interface:



Employees Name: Name Here:

Days they can Work

Availability 9AM - 5 PM

Hours Per Week: 24

7. User Stories - All

User story: As an employee, I want to be scheduled around my requested availability times, so I can be able to come into work without neglecting my needs and obligations.
Priority Level: High
Estimated hours it will take to accomplish this: 7 (Assuming software has already been started and can operate without this functionality)

User story: As a manager, I want to schedule my employees quickly so I don't have to spend a lot of time on it.
Priority Level: Medium
Estimated hours it will take to accomplish this: 3-4

User story: As a manager, I want to enter our operating hours and staff requirements, so that I don't worry a lack of staffing.
Priority Level: High
Estimated hours it will take to accomplish this: 3-4

User story: As a manager, I want to see a automatically generated schedule options, so I save time by only tweaking it.
Priority Level: High
Estimated hours it will take to accomplish this: 9

User story: As a manager of Cline Library, I want to be able to print the schedule so everyone is can see who is working when.
Priority Level: Medium
Estimated hours it will take to accomplish this: 2 - 6 (If our program is built into excel or on the internet browser printing will be an easy thing to accomplish because those programs already have built in printing functionality.)

User story: As an Employee, I want to easily request time off so that I do not conflict with the schedule and will get my shift covered.
Priority Level: High
Estimated hours it will take to accomplish this: 13 (Assuming software has already been started and can operate without this functionality)

User story: As a Shift lead at Sodexo dining services, I want to easy to use, for creating a weekly schedule.
Priority Level: Medium
Estimated hours it will take to accomplish this:
Possibly about no more than 24 hours as little as 12 hours.

User story: As a Supervisor of two units at Sodexo dining services, I want simplicity, for scheduling people.
Priority Level: High
Estimated hours it will take to accomplish this:
Should be about around two days.

---

8. Trello

https://trello.com/b/xWYGWFAX

9. Group participation:

Nicholas Anderson - Entered initial requirements 3 and 4, created 2 use cases and diagrams, added use cases to Trello board, took notes for weekly meeting, uploaded weekly meeting and d2 to github. -25%

Carli Martinez - Set up user stories templates for everyone and added two, created use case diagram on Draw.io with two actors and a few use cases to start, wrote a use case description and set up use case description templates for everyone, set up Trello and added use cases there, wrote value proposition 25%

Brandon Thomas - Non-functional requirements, MVP, Use case, Use case Description, and Users stories.- 25%

Ryan Wallace - Created the problem statement and Product Position Statement. Created 2 Use case Descriptions, and Users stories. - 25%