SDD: 401 Senior Project - USchedule

6.1. Introduction

This Software Design Description Document presents the architecture and detailed design for the software for the USchecule project. This Project is a Web Application which provides an application to generate work schedules for employees in an organization.

The application will take the residents' preferences and then generate a call schedule by the constraints inputted by the administrator. The application will also be generalized to be usable by other industries such as restaurants, retail, etc. The purpose is to make scheduling efficient, simple and scalable.

6.1.1 System Objectives

The objective of this application is to provide administrators with a new webapp that can take in the residents' preferences and then generate a call schedule based off the constraints inputted by the user admin. The current scheduling systems that exist are expensive and performing this task by hand is tedious. This application will reduce time and cost for users as the schedule will be generated within seconds.

6.1.2 Hardware, Software, and Human Interfaces

6.1.2.1 Graphical User Interface - Node.js and React on AWS
We are using Visual Studio Code version 1.38 and up as our code editor.

6.1.2.2 Server - Python/Flask on AWS Lambda

6.1.2.3 Algorithm - We are using Google OR Tools for help with our algorithm

6.1.2.4 Database - MySQL on AWS Relational Database Services
We are using MySQL Workbench for our database.

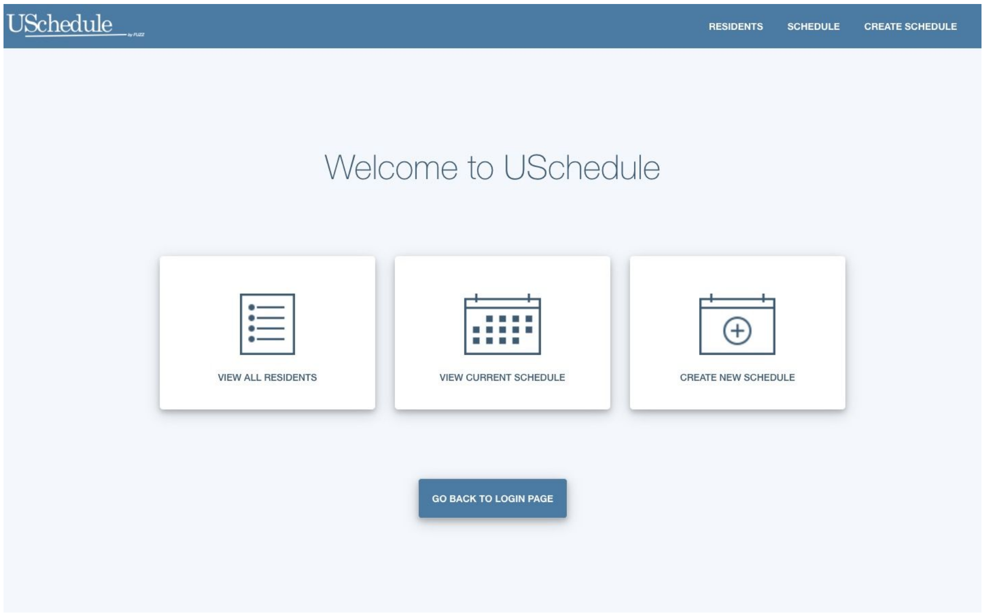
6.1.2.5 Operating Systems - MacBook Pro/Air
MacOS Mojave 10.14 and up

6.1.2.6 Design Mockups - Sketch
We are using the application Sketch to make mock ups of our front end design.

6.1.2.7 Other Resources
We are using Trello to organize and prioritize tasks for the project. We have been using Google Drive as as our platform for organizing and working on written documents.

6.2 Architectural Design

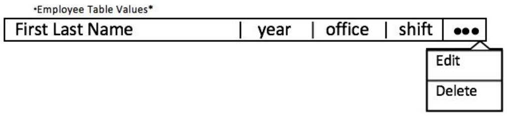
With the design of our system, our users will be interacting with our front end web application. Our front end will be interacting directly with the Algorithm. The algorithm will be communicating with each of the needed resident, organization, and shift classes. Those classes will be communicating with the database to grab all data relating to that specific class.



UI Layout

Log In			
Home Page	Scheduling Page	Resident Management Page	Output Page
♦ Schedule ⊕ Create Button ⊕ View Residents ⊕ Log Out	♦ List of residents ⊕ Options for each resident ⊕ Float preference drop down ⊕ Vacation preference	⊕ Add new resident □ Employee table values ⊕ Remove resident ♦ Confirm pop up ♦ Edit screen ⊕ Filter ⊕ Generate schedule ♦ Error page	♦ Schedule generates ⊕ Download button

Key
♦ UI Element
⊕ Button
□ Field



6.2.1 Major Software Components

6.2.1.1 The GUI displays a Login Page

- The login page will communicate with the user login data in the database to verify correct username and password information has been entered. If forgot password button is selected, it will communicate with the database to grab email information for that user. It will also work to store new user data in the database when creating a new account. The login button will direct the user to the home page.

6.2.1.2 The GUI displays a Home Page

- The home page will allow the user to visit the scheduling and resident page. It will also have a button allowing the user to view the current schedule. When selected, this will communicate with the database to retrieve the schedule which will be stored as a JSON object.

6.2.1.3 The GUI displays a Scheduling Page

- The scheduling page will take in the inputted schedule constraints and store them until the schedule has been generated.

6.2.1.4 The GUI displays a Resident Management Page

- The resident page will communicate and add/edit/pull/delete data from the resident database.

6.2.1.5 The GUI displays an Output Page

- The output page will communicate with the algorithm which communicates with the related classes which communicate with the database. This algorithm will generate the schedule.

6.2.2 Major Software Interactions

6.2.2.1 Login Page

- The login page will send username and password data to the database. The database will communicate whether or not this data is valid.

6.2.2.2 Home Page

- The home page current schedule button will grab the JSON object for the schedule from the database. The object will include days, shifts, employees, and organization.

6.2.2.3 Scheduling Page

- The scheduling page will grab the resident data from the database to display all current residents to be scheduled. The scheduling page will take in constraints for each resident which will be used as constraints in the algorithm which generates the schedule.

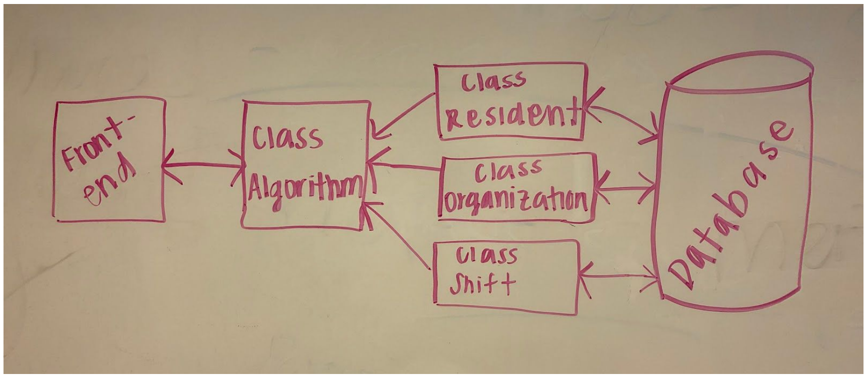
6.2.2.4 Resident Management Page

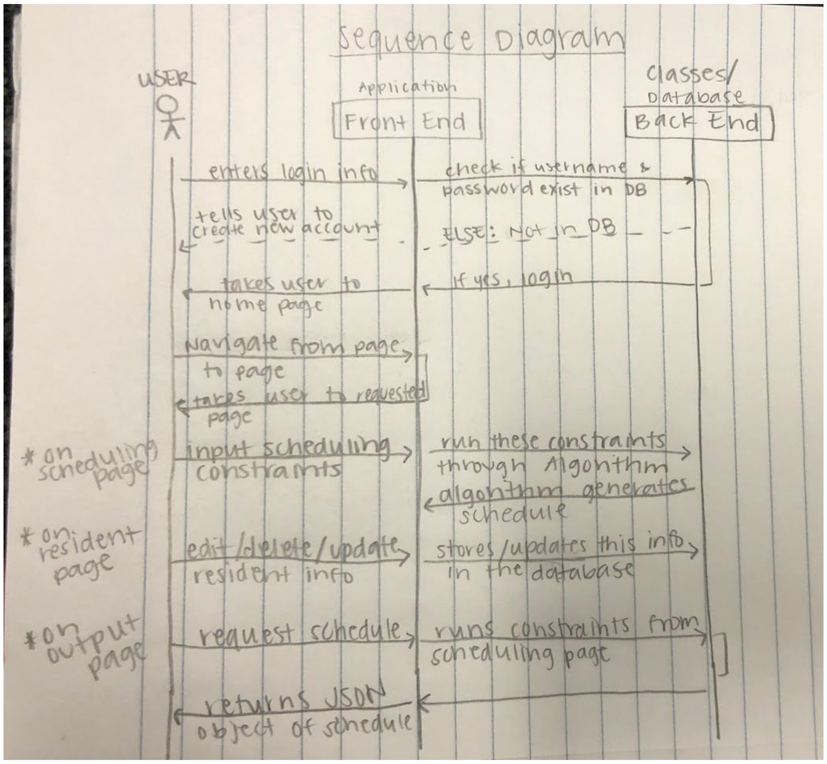
- The resident management page will send resident data to be stored in the database. It will also edit or delete current resident data from the database.

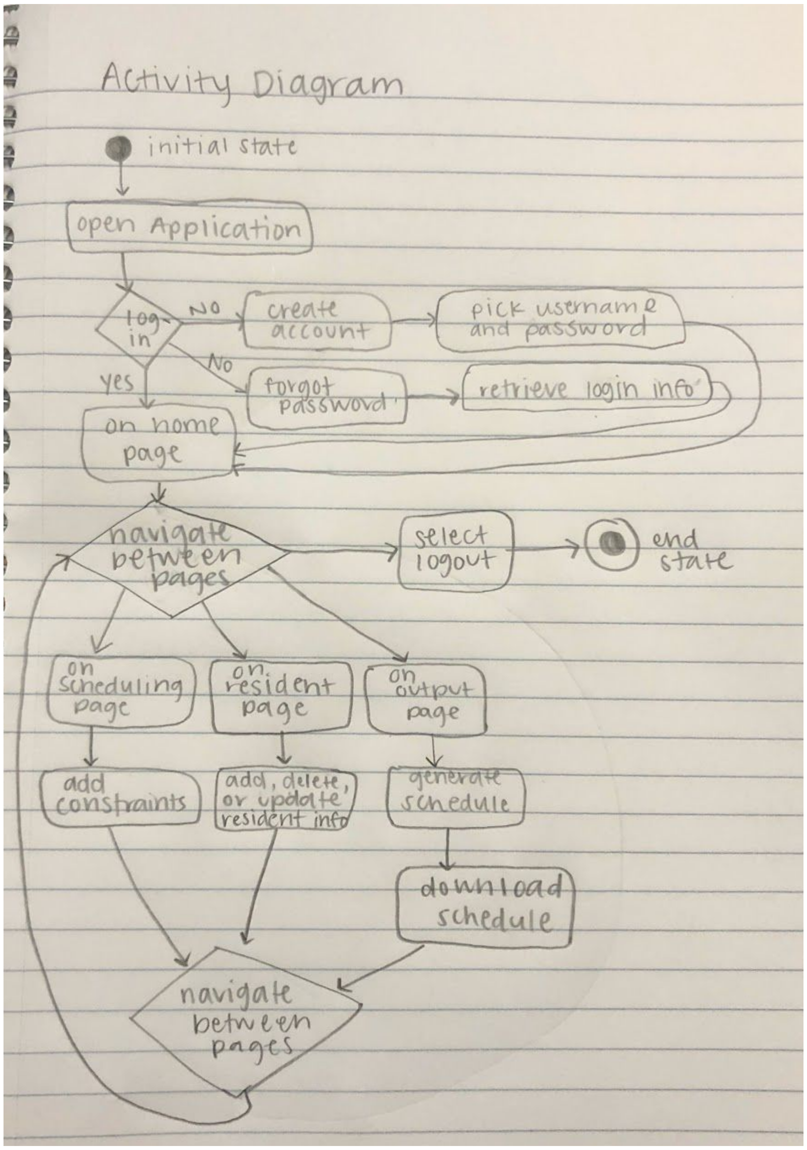
6.2.2.5 Output Page

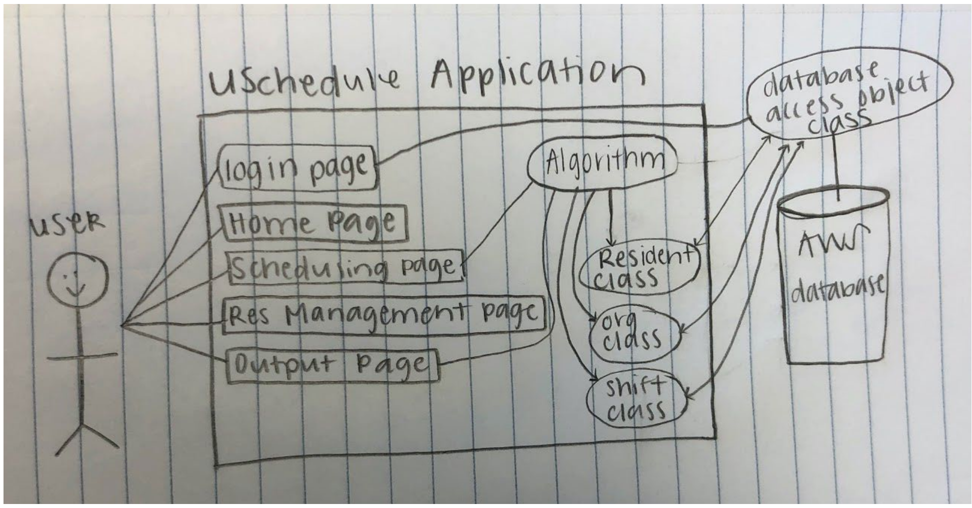
- The output page will grab the created JSON object for the schedule which was generated from the algorithm.

6.2.3 Architectural Design Diagrams




Sequence Diagram
USER
Application
Front End
Classes/ Database
Back End
enters login info
check if username & password exist in DB
tells user to create new account
ELSE: Not in DB
takes user to home page
if yes, login
Navigate from page to page
takes user to requested page
* on scheduling page
input scheduling constraints
run these constraints through Algorithm
algorithm generates schedule
* on resident page
edit/delete/update resident info
stores/updates this info in the database
* on output page
request schedule
runs constraints from scheduling page
returns JSON object of schedule


Activity Diagram
initial state
open Application
log-in
No
create account
pick username and password
No
forgot password
retrieve login info
yes
on home page
navigate between pages
select logout
end state
on scheduling page
on resident page
on output page
add constraints
add, delete, or update resident info
generate schedule
download schedule
navigate between pages


USchedule Application
database access object class
login page
Algorithm
user
Home Page
Scheduling page
Resident class
AWS database
Res Management page
org class
Output Page
shift class

6.3. CSC and CSU Descriptions

6.3.1 Class Descriptions

6.3.1.1 Detailed Class Description 1

6.3.1.n Detailed Class Description n

6.3.2 Detailed Interface Descriptions

6.3.3 Detailed Data Structure Descriptions

6.3.4 Detailed Design Diagrams

6.4 Database Design and Description

6.4.1 Database Design ER Diagram

6.4.2 Database Access

6.4.3 Database Security