Title: Flexible Employee Work Scheduler

Author: Kelly Lamb

CST-452 Capstone Project Requirements Document

Grand Canyon University

Instructor: Professor Amr Elchouemi

Revision: 1.0

Date: July 31, 2022

ABSTRACT

Creating a weekly & daily schedule of employees with constrained availability in a constantly changing work force requirement is a meticulous and time-consuming process fraught with potential for error. This application will assist in creating a schedule that has both flexible work demands and flexible employee availability. The result is to allow a scheduling manager the ability to define skeleton shifts per department, modify said shifts per daily/hourly demand, track and apply employees with training and availability to those shifts – evenly and fairly distributing shifts and hours reducing the need to track overtime (too many hours per day or too many days per week) and create a weekly schedule report with indicators to show manual intervention needed. This document is to provide the Final Architectural Plan (Design Specifications) to build the application.

History and Signoff Sheet

Change Record

Date	Author	Revision Notes
2022/07/03	Kelly Lamb	Initial draft for review/discussion
2022/07/10	Kelly Lamb	Include Actual Screenshots
2022/07/17	Kelly Lamb	New ERD, SQL, More Screenshots
2022/07/24	Kelly Lamb	New Functionality - Employee, Availability & Training
2022/07/31	Kelly Lamb	New Functionality - Templates, Template Details

Overall Instructor Feedback/Comments

Overall Instructor Feedback/Comments

Integrated Instructor Feedback into Project Documentation

☐ Yes ☐ No

TABLE OF CONTENTS

Design Introduction

1. Provide the high-level design of the proposed solution or business case with supporting narrative text. This design should include mock-up screen shots for the proposed user interface, pseudocode, or flowcharts that show the logic for the program, as well as the anticipated process flow. The purpose of the solution/business case design is to allow the stakeholder to approve the concepts before committing resources to the technical design.

2. Use the template to list the project deliverables that are to be included external to this Design Specification (Data Dictionary, API Design, etc.).

Deliverable Acceptance Log					
ID	Deliverable Description	Comments	Evaluator (internal or external as applicable)	Status	Date of Decision
1	Scheduler_data_dictionary.xlsx	Spreadsheet containing tabs specifying database table structures	Kelly		
2	Scheduler.sql.txt	A text file containing the SQL to create the MySQL Database Schema	Kelly		
3	Scheduler_sprint_backlog.xlsx	Spreadsheet containing Functional, Non-Functional, and Technical Use Cases	Kelly		
4	Lean Canvas docx	Contains Metrics and information for application and customers	Kelly		
5	Test Cases.xlsx	Spreadsheet listing test cases	Kelly		
5	Traceability Matrix.xlsx	Spreadsheet listing test requirements and related tests cases for quality assurance	Kelly		

Detailed High-Level Solution Design

1. Provide a detailed overview of how the proposed design fits into the overall solution.
    a. Create diagrams to logically and physically depict the system. This can be illustrated using UML Component, UML Deployment, and UML Activity diagrams or simply block diagrams done in a drawing tool such as Visio.
    b. This section should also include any solution configuration changes that will be required to develop and implement the proposed solution.
    c. Describe the approach and resources required to assure non-functional requirements (such as security and performance) will be met with this solution.
    d. The purpose of the detail solution architecture is to provide sufficient information for a developer to produce the system.
2. Provide a detailed inventory of hardware and software technologies that will be used in the solution:
    a. List any Frameworks or third party libraries that will be used.
    b. List any Proof of Concepts to be completed (POC) to ensure that the technologies and frameworks selected are the best fit for purpose, cost effective, and proper to solve the problem. This section should also be updated with the purpose/rationale for the POC and the results of the POC.

Use the templates below to list the Proof of Concepts, hardware, and technologies.

Proof of Concepts		
Description	**Rationale**	**Results**
1. Develop Environment	Industry Standards	Reliable performance for other applications
2. Database	Proven Reliability, Availability, and Performance	Reliable performance for other applications
3. Manual Run-through Process	Test process to see logical outcome	Works in theory

Hardware and Software Technologies
1. Language: PHP
2. Framework: Laravel
3. Library: PHP Spreadsheet
4. IDE: Visual Studio
5. Database: MySQL
6. Template Engine: Blade
7. Server: MAMP (Apache Tomcat)

8. Logger
9. Bootstrap
10. Composer

Pseudo Code Algorithm – Schedule Generator:

1. Define User
2. Define Departments
3. Define Positions within Departments
4. Define Shifts for those positions
5. Define Templates and fill with a list of shifts (details)
6. Define a target week (day by day) and fill with a list of templates(details)
7. Define Employee – Availability and fill with a list of trained positions (emp. Training)

Algorithm Loop

1. Create loop through target weeks from lowest day number to max
2. Find all shifts associated to this day by linking back through target week and associated template details
3. Loop through employee training position and check employee availability – assign first one to an array of day, position, time range
4. Break from employee training position loop and continue with shift loop
5. If no employee found for shift – put an indicator in the array (not found)
6. Continue target week loop until each day completes
7. Print report

Logical Solution Design:

Provide the proper diagrams and drawings that represent the high-level logical solution design.

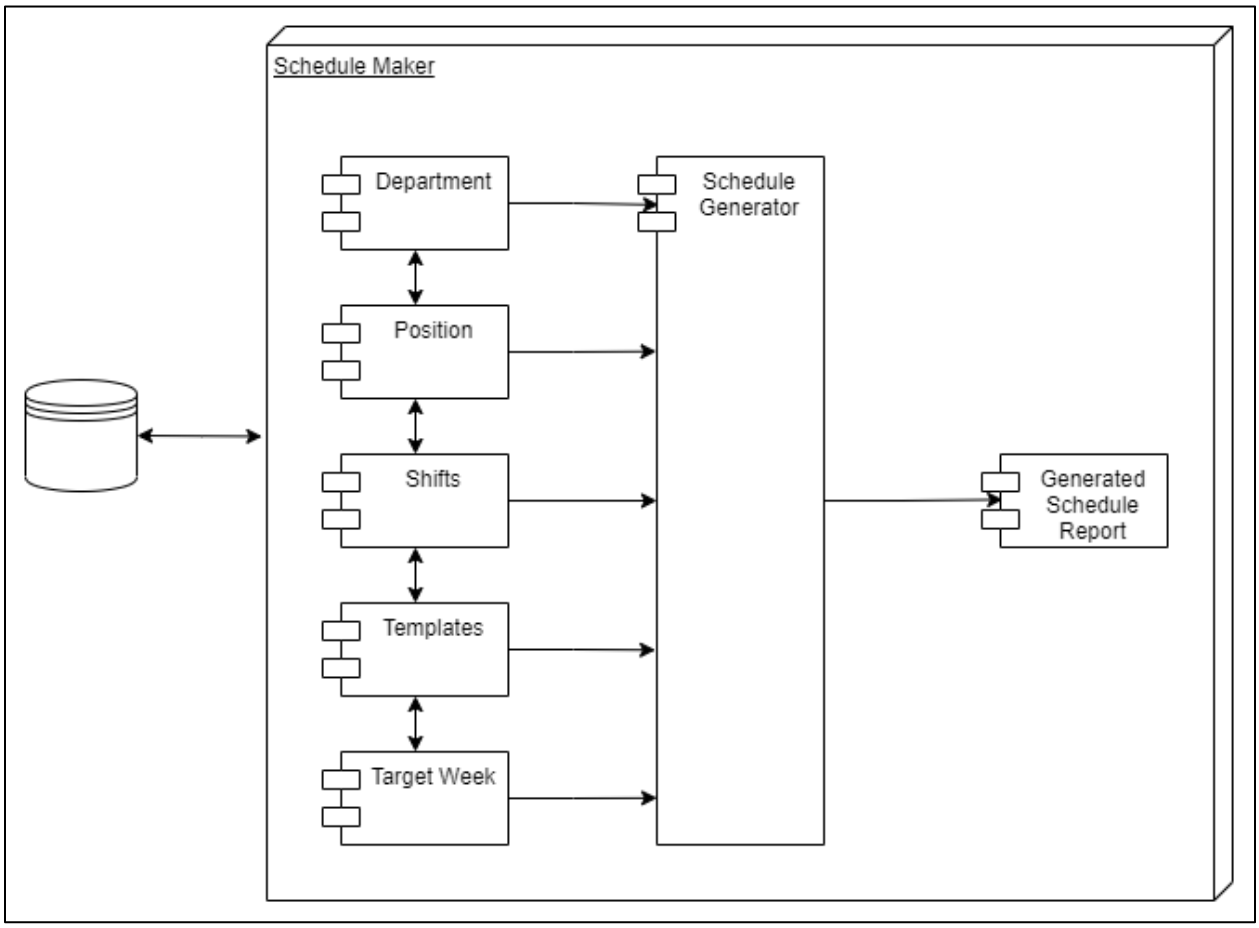


Image: Logical Architecture Diagram

Physical Solution Design:

Provide the proper diagrams and drawings that represent the high-level physical solution design.

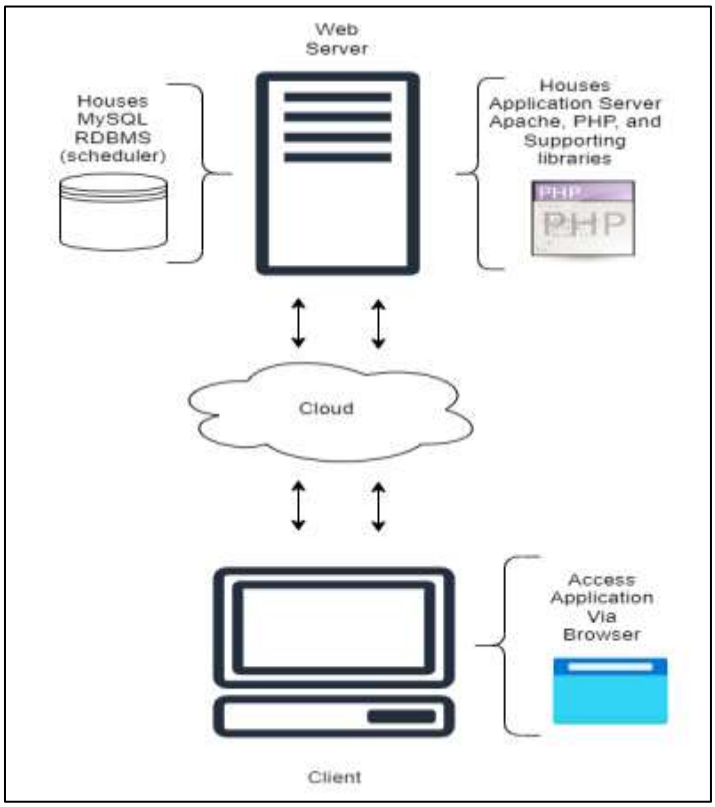


Image: Physical Solution Design Diagram

Detailed Technical Design

General Technical Approach:
You should, in words, describe your approach and design here. You should also summarize any meeting notes, brainstorming sessions, etc. that you want to retain through the design of your project.

1. Create a development environment using stable, recent, and compatible versions of PHP, Laravel, MAMP, Composer, and PHP Spreadsheet (Preferably using Visual Studio or PHP Storm).
2. Create the MySQL Database using scripts that correspond to the data dictionary specifications.
3. Develop code (Objects) to handle Database Services (CRUD, DAO / DTO), Business Services (Report Generation), Controllers and Forms (MVC).
4. Create Repository on either BitBucket or GitHub to store code and design
5. Deploy and Test locally using a local version of MAMP
6. Deploy remotely to either Azure or Heroku (either can house the PHP application and MySQL database).

Key Technical Design Decisions:
Any final technical design decisions, such as framework decisions, etc., should be documented here. This should list the technology/framework, its purpose in the design, and why it was chosen. If necessary, the proper Proof of Concepts should be defined and implemented to validate the technical decision.

1. As listed above, PHP, Laravel, and MySQL databases are tried-and-true solutions with a large user base, documentation – basically, the de facto standard for web applications development and supported by many hosting sites.
2. Blade templates works well with Bootstrap styling (as proof of concepts from previous projects).
3. Free (low cost) solutions to keep in mind – avoid using costly licenses

Database ER Diagram:
Image file of your ER database diagram.

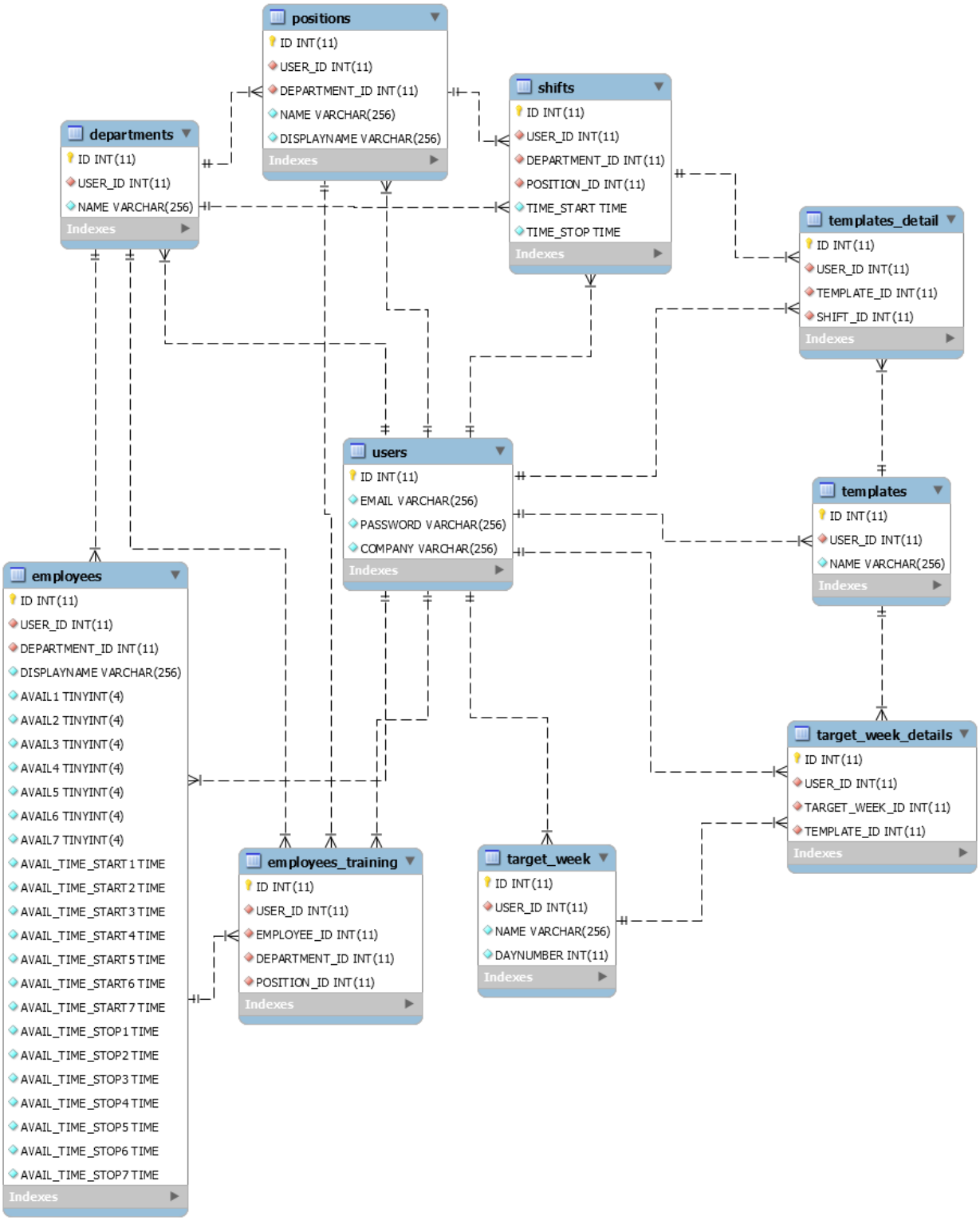


Image: ER Diagram

Database DDL Scripts:
This should contain the DDL script showing all database constraints, etc.

Filename: scheduler.sql.txt

```
-- phpMyAdmin SQL Dump
-- version 5.0.4
-- https://www.phpmyadmin.net/
--
-- Host: localhost:3306
-- Generation Time: Aug 01, 2022 at 12:28 AM
-- Server version: 5.7.24
-- PHP Version: 7.4.16

SET FOREIGN_KEY_CHECKS=0;
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `scheduler`
--
CREATE DATABASE IF NOT EXISTS `scheduler` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
USE `scheduler`;

-- --------------------------------------------------------

--
-- Table structure for table `departments`
--
-- Creation: Jul 04, 2022 at 03:11 AM
--

DROP TABLE IF EXISTS `departments`;
CREATE TABLE IF NOT EXISTS `departments` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `NAME` varchar(256) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `USER_ID_IDX1` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `departments`
--

TRUNCATE TABLE `departments`;
--
-- Dumping data for table `departments`
--

INSERT INTO `departments` (`ID`, `USER_ID`, `NAME`) VALUES
(1, 1, 'Office'),
(2, 1, 'Maintenance');

-- --------------------------------------------------------

--
-- Table structure for table `employees`
--
-- Creation: Jul 16, 2022 at 08:35 AM
--
```



```
DROP TABLE IF EXISTS `employees`;
CREATE TABLE IF NOT EXISTS `employees` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `DEPARTMENT_ID` int(11) NOT NULL,
  `DISPLAYNAME` varchar(256) NOT NULL,
  `AVAIL1` tinyint(4) NOT NULL,
  `AVAIL2` tinyint(4) NOT NULL,
  `AVAIL3` tinyint(4) NOT NULL,
  `AVAIL4` tinyint(4) NOT NULL,
  `AVAIL5` tinyint(4) NOT NULL,
  `AVAIL6` tinyint(4) NOT NULL,
  `AVAIL7` tinyint(4) NOT NULL,
  `AVAIL_TIME_START1` time NOT NULL,
  `AVAIL_TIME_START2` time NOT NULL,
  `AVAIL_TIME_START3` time NOT NULL,
  `AVAIL_TIME_START4` time NOT NULL,
  `AVAIL_TIME_START5` time NOT NULL,
  `AVAIL_TIME_START6` time NOT NULL,
  `AVAIL_TIME_START7` time NOT NULL,
  `AVAIL_TIME_STOP1` time NOT NULL,
  `AVAIL_TIME_STOP2` time NOT NULL,
  `AVAIL_TIME_STOP3` time NOT NULL,
  `AVAIL_TIME_STOP4` time NOT NULL,
  `AVAIL_TIME_STOP5` time NOT NULL,
  `AVAIL_TIME_STOP6` time NOT NULL,
  `AVAIL_TIME_STOP7` time NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `DEP_ID_IDX2` (`DEPARTMENT_ID`),
  KEY `USER_ID_IDX2` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `employees`
--

TRUNCATE TABLE `employees`;
--
-- Dumping data for table `employees`
--

INSERT INTO `employees` (`ID`, `USER_ID`, `DEPARTMENT_ID`, `DISPLAYNAME`, `AVAIL1`, `AVAIL2`,
`AVAIL3`, `AVAIL4`, `AVAIL5`, `AVAIL6`, `AVAIL7`, `AVAIL_TIME_START1`, `AVAIL_TIME_START2`,
`AVAIL_TIME_START3`, `AVAIL_TIME_START4`, `AVAIL_TIME_START5`, `AVAIL_TIME_START6`,
`AVAIL_TIME_START7`, `AVAIL_TIME_STOP1`, `AVAIL_TIME_STOP2`, `AVAIL_TIME_STOP3`,
`AVAIL_TIME_STOP4`, `AVAIL_TIME_STOP5`, `AVAIL_TIME_STOP6`, `AVAIL_TIME_STOP7`) VALUES
(1, 1, 1, 'Kelly Lamb', 1, 1, 1, 1, 1, 1, 1, '00:00:00', '00:00:00', '00:00:00', '00:00:00',
'00:00:00', '00:00:00', '00:00:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00',
'23:59:00', '23:59:00'),
(2, 1, 1, 'Ashley Lamb', 1, 1, 1, 1, 1, 1, 1, '00:00:00', '00:00:00', '00:00:00', '00:00:00',
'00:00:00', '00:00:00', '00:00:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00',
'23:59:00', '23:59:00'),
(3, 1, 2, 'Pancho Padilla', 1, 1, 1, 1, 1, 1, 1, '00:00:00', '00:00:00', '00:00:00', '00:00:00',
'00:00:00', '00:00:00', '00:00:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00',
'23:59:00', '23:59:00'),
(4, 1, 2, 'Luis Aparicio', 1, 1, 1, 1, 1, 1, 1, '00:00:00', '00:00:00', '00:00:00', '00:00:00',
'00:00:00', '00:00:00', '00:00:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00',
'23:59:00', '23:59:00'),
(5, 1, 2, 'Kenneth Schweibinz', 1, 1, 1, 1, 1, 1, 1, '00:00:00', '00:00:00', '00:00:00',
'00:00:00', '00:00:00', '00:00:00', '00:00:00', '23:59:00', '23:59:00', '23:59:00', '23:59:00',
'23:59:00', '23:59:00', '23:59:00');

-- --------------------------------------------------------

--
-- Table structure for table `employees_training`
--
-- Creation: Jul 16, 2022 at 08:35 AM
--
```



```
DROP TABLE IF EXISTS `employees_training`;
CREATE TABLE IF NOT EXISTS `employees_training` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `EMPLOYEE_ID` int(11) NOT NULL,
  `DEPARTMENT_ID` int(11) NOT NULL,
  `POSITION_ID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `EMP_ID_IDX1` (`EMPLOYEE_ID`),
  KEY `DEP_ID_IDX3` (`DEPARTMENT_ID`),
  KEY `POS_ID_IDX1` (`POSITION_ID`),
  KEY `USER_ID_IDX3` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `employees_training`
--

TRUNCATE TABLE `employees_training`;
--
-- Dumping data for table `employees_training`
--

INSERT INTO `employees_training` (`ID`, `USER_ID`, `EMPLOYEE_ID`, `DEPARTMENT_ID`, `POSITION_ID`)
VALUES
(1, 1, 1, 1, 1),
(2, 1, 2, 1, 2),
(3, 1, 3, 2, 3),
(4, 1, 4, 2, 4),
(5, 1, 5, 2, 5);

-- --------------------------------------------------------

--
-- Table structure for table `positions`
--
-- Creation: Jul 16, 2022 at 08:31 AM
--

DROP TABLE IF EXISTS `positions`;
CREATE TABLE IF NOT EXISTS `positions` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `DEPARTMENT_ID` int(11) NOT NULL,
  `NAME` varchar(256) NOT NULL,
  `DISPLAYNAME` varchar(256) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `DEP_ID_IDX` (`DEPARTMENT_ID`),
  KEY `USER_ID_IDX4` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `positions`
--

TRUNCATE TABLE `positions`;
--
-- Dumping data for table `positions`
--

INSERT INTO `positions` (`ID`, `USER_ID`, `DEPARTMENT_ID`, `NAME`, `DISPLAYNAME`) VALUES
(1, 1, 1, 'General Manager', 'GM'),
(2, 1, 1, 'Assistant Manager', 'MOD'),
(3, 1, 2, 'Supervisor', 'Lead'),
(4, 1, 2, 'Assistant', 'Asst'),
(5, 1, 2, 'Mechanic', 'Mech');

-- --------------------------------------------------------

--
-- Table structure for table `shifts`
```



```
--
-- Creation: Jul 16, 2022 at 08:32 AM
--

DROP TABLE IF EXISTS `shifts`;
CREATE TABLE IF NOT EXISTS `shifts` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `DEPARTMENT_ID` int(11) NOT NULL,
  `POSITION_ID` int(11) NOT NULL,
  `TIME_START` time NOT NULL,
  `TIME_STOP` time NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `DEP_ID_IDX4` (`DEPARTMENT_ID`),
  KEY `POS_ID_IDX2` (`POSITION_ID`),
  KEY `USER_ID_IDX5` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `shifts`
--

TRUNCATE TABLE `shifts`;
--
-- Dumping data for table `shifts`
--

INSERT INTO `shifts` (`ID`, `USER_ID`, `DEPARTMENT_ID`, `POSITION_ID`, `TIME_START`, `TIME_STOP`)
VALUES
(1, 1, 2, 3, '06:00:00', '14:00:00'),
(2, 1, 2, 4, '06:00:00', '14:00:00'),
(3, 1, 2, 5, '08:00:00', '16:00:00'),
(4, 1, 1, 1, '06:00:00', '18:00:00'),
(5, 1, 1, 2, '15:00:00', '23:00:00');

-- --------------------------------------------------------

--
-- Table structure for table `target_week`
--
-- Creation: Jul 16, 2022 at 08:34 AM
--

DROP TABLE IF EXISTS `target_week`;
CREATE TABLE IF NOT EXISTS `target_week` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `NAME` varchar(256) NOT NULL,
  `DAYNUMBER` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `USER_ID_IDX6` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `target_week`
--

TRUNCATE TABLE `target_week`;
--
-- Dumping data for table `target_week`
--

INSERT INTO `target_week` (`ID`, `USER_ID`, `NAME`, `DAYNUMBER`) VALUES
(1, 1, 'Mon', 1),
(2, 1, 'Tue', 2),
(3, 1, 'Wed', 3),
(4, 1, 'Thur', 4),
(5, 1, 'Fri', 5),
(6, 1, 'Sat', 6),
(7, 1, 'Sun', 7);
```



```
-- --------------------------------------------------------

--
-- Table structure for table `target_week_details`
--
-- Creation: Jul 16, 2022 at 08:34 AM
--

DROP TABLE IF EXISTS `target_week_details`;
CREATE TABLE IF NOT EXISTS `target_week_details` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `TARGET_WEEK_ID` int(11) NOT NULL,
  `TEMPLATE_ID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `TARGET_WEEK_ID_IDX1` (`TARGET_WEEK_ID`),
  KEY `TEMPLATE_ID_IDX1` (`TEMPLATE_ID`),
  KEY `USER_ID_IDX7` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `target_week_details`
--

TRUNCATE TABLE `target_week_details`;
--
-- Dumping data for table `target_week_details`
--

INSERT INTO `target_week_details` (`ID`, `USER_ID`, `TARGET_WEEK_ID`, `TEMPLATE_ID`) VALUES
(1, 1, 1, 1),
(2, 1, 1, 2),
(3, 1, 1, 3),
(4, 1, 2, 1),
(5, 1, 2, 2),
(6, 1, 2, 3),
(10, 1, 4, 1),
(11, 1, 4, 2),
(12, 1, 4, 3),
(13, 1, 5, 1),
(14, 1, 5, 2),
(15, 1, 5, 3),
(16, 1, 6, 1),
(17, 1, 7, 1),
(18, 1, 3, 1),
(19, 1, 3, 2),
(20, 1, 3, 3);

-- --------------------------------------------------------

--
-- Table structure for table `templates`
--
-- Creation: Jul 16, 2022 at 08:32 AM
--

DROP TABLE IF EXISTS `templates`;
CREATE TABLE IF NOT EXISTS `templates` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `NAME` varchar(256) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `USER_ID_IDX8` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `templates`
--

TRUNCATE TABLE `templates`;
--
```



```
-- Dumping data for table `templates`
--

INSERT INTO `templates` (`ID`, `USER_ID`, `NAME`) VALUES
(1, 1, 'Office Only'),
(2, 1, 'Maintenance Only'),
(3, 1, 'Mechanic Only');

-- --------------------------------------------------------

--
-- Table structure for table `templates_detail`
--
-- Creation: Jul 16, 2022 at 08:33 AM
--

DROP TABLE IF EXISTS `templates_detail`;
CREATE TABLE IF NOT EXISTS `templates_detail` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `USER_ID` int(11) NOT NULL,
  `TEMPLATE_ID` int(11) NOT NULL,
  `SHIFT_ID` int(11) NOT NULL,
  PRIMARY KEY (`ID`),
  KEY `TMP_ID_IDX` (`TEMPLATE_ID`),
  KEY `SHIFT_ID_IDX1` (`SHIFT_ID`),
  KEY `USER_ID_IDX9` (`USER_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `templates_detail`
--

TRUNCATE TABLE `templates_detail`;
--
-- Dumping data for table `templates_detail`
--

INSERT INTO `templates_detail` (`ID`, `USER_ID`, `TEMPLATE_ID`, `SHIFT_ID`) VALUES
(1, 1, 1, 4),
(2, 1, 1, 5),
(3, 1, 2, 1),
(4, 1, 2, 2),
(5, 1, 3, 3);

-- --------------------------------------------------------

--
-- Table structure for table `users`
--
-- Creation: Jul 04, 2022 at 02:38 AM
--

DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `ID` int(11) NOT NULL AUTO_INCREMENT,
  `EMAIL` varchar(256) NOT NULL,
  `PASSWORD` varchar(256) NOT NULL,
  `COMPANY` varchar(256) NOT NULL,
  PRIMARY KEY (`ID`),
  UNIQUE KEY `USER_EMAIL_IDX` (`EMAIL`),
  KEY `PASSWORD_IDX` (`PASSWORD`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

--
-- Truncate table before insert `users`
--

TRUNCATE TABLE `users`;
--
-- Dumping data for table `users`
--
```



```
INSERT INTO `users` (`ID`, `EMAIL`, `PASSWORD`, `COMPANY`) VALUES
(1, 'kl@kl.com', '11111111', 'Fun Park');

--
-- Constraints for dumped tables
--

--
-- Constraints for table `departments`
--
ALTER TABLE `departments`
  ADD CONSTRAINT `USERS_ID_FK1` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `employees`
--
ALTER TABLE `employees`
  ADD CONSTRAINT `DEPARTMENT_ID_FK3` FOREIGN KEY (`DEPARTMENT_ID`) REFERENCES `departments`
(`ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `USERS_ID_FK8` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `employees_training`
--
ALTER TABLE `employees_training`
  ADD CONSTRAINT `DEPARTMENT_ID_FK4` FOREIGN KEY (`DEPARTMENT_ID`) REFERENCES `departments`
(`ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `EMPLOYEE_ID_FK1` FOREIGN KEY (`EMPLOYEE_ID`) REFERENCES `employees` (`ID`) ON
DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `POSITION_ID_FK2` FOREIGN KEY (`POSITION_ID`) REFERENCES `positions` (`ID`) ON
DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `USERS_ID_FK9` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `positions`
--
ALTER TABLE `positions`
  ADD CONSTRAINT `DEPARTMENT_ID_FK1` FOREIGN KEY (`DEPARTMENT_ID`) REFERENCES `departments`
(`ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `USERS_ID_FK2` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `shifts`
--
ALTER TABLE `shifts`
  ADD CONSTRAINT `DEPARTMENT_ID_FK2` FOREIGN KEY (`DEPARTMENT_ID`) REFERENCES `departments`
(`ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `POSITION_ID_FK1` FOREIGN KEY (`POSITION_ID`) REFERENCES `positions` (`ID`) ON
DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `USERS_ID_FK3` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `target_week`
--
ALTER TABLE `target_week`
  ADD CONSTRAINT `USERS_ID_FK6` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `target_week_details`
--
ALTER TABLE `target_week_details`
  ADD CONSTRAINT `TARGET_WEEK_ID_FK1` FOREIGN KEY (`TARGET_WEEK_ID`) REFERENCES `target_week`
(`ID`) ON DELETE CASCADE ON UPDATE CASCADE,
```



```
  ADD CONSTRAINT `TEMPLATE_ID_FK2` FOREIGN KEY (`TEMPLATE_ID`) REFERENCES `templates` (`ID`) ON
DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `USERS_ID_FK7` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `templates`
--
ALTER TABLE `templates`
  ADD CONSTRAINT `USERS_ID_FK4` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Constraints for table `templates_detail`
--
ALTER TABLE `templates_detail`
  ADD CONSTRAINT `SHIFT_ID_FK1` FOREIGN KEY (`SHIFT_ID`) REFERENCES `shifts` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `TEMPLATE_ID_FK1` FOREIGN KEY (`TEMPLATE_ID`) REFERENCES `templates` (`ID`) ON
DELETE CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `USERS_ID_FK5` FOREIGN KEY (`USER_ID`) REFERENCES `users` (`ID`) ON DELETE
CASCADE ON UPDATE CASCADE;
SET FOREIGN_KEY_CHECKS=1;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```


Flow Charts/Process Flows:

You should insert any flow charts or UML Activity diagrams here. Flow charts should document algorithms or workflows that will be implemented in your program.

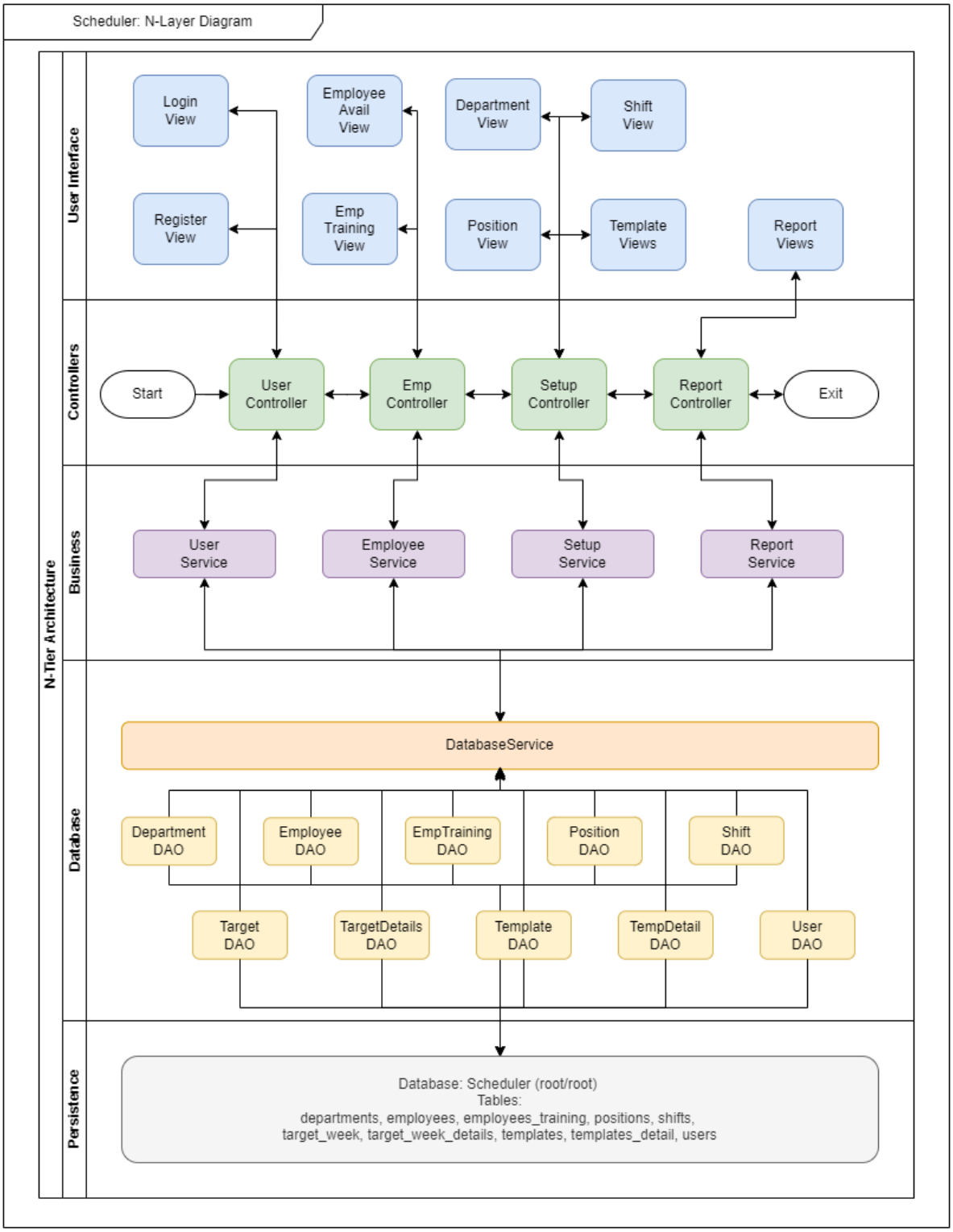


Image: UML N-Layer Architecture Diagram

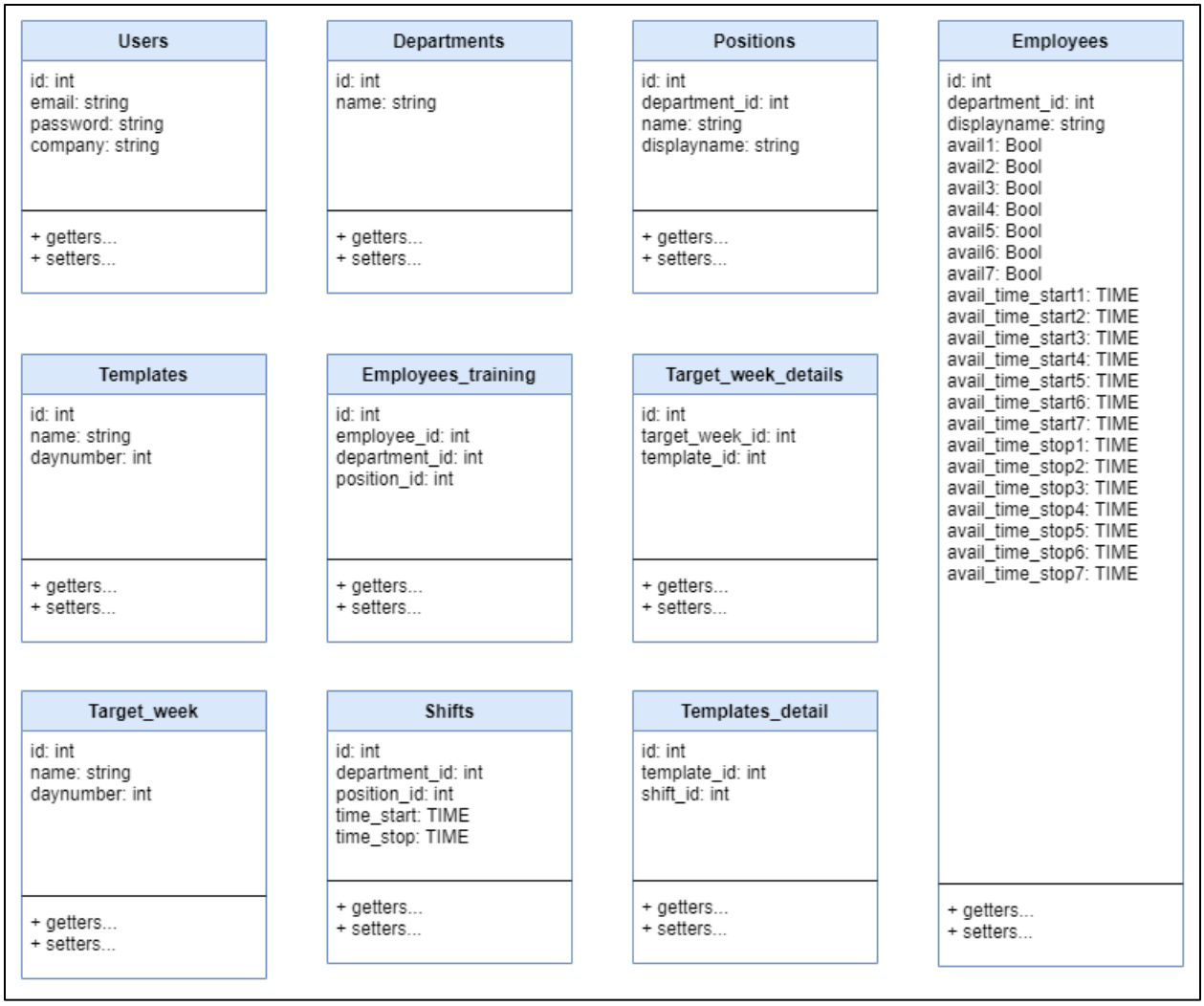


Image: UML Class Diagrams (Models)

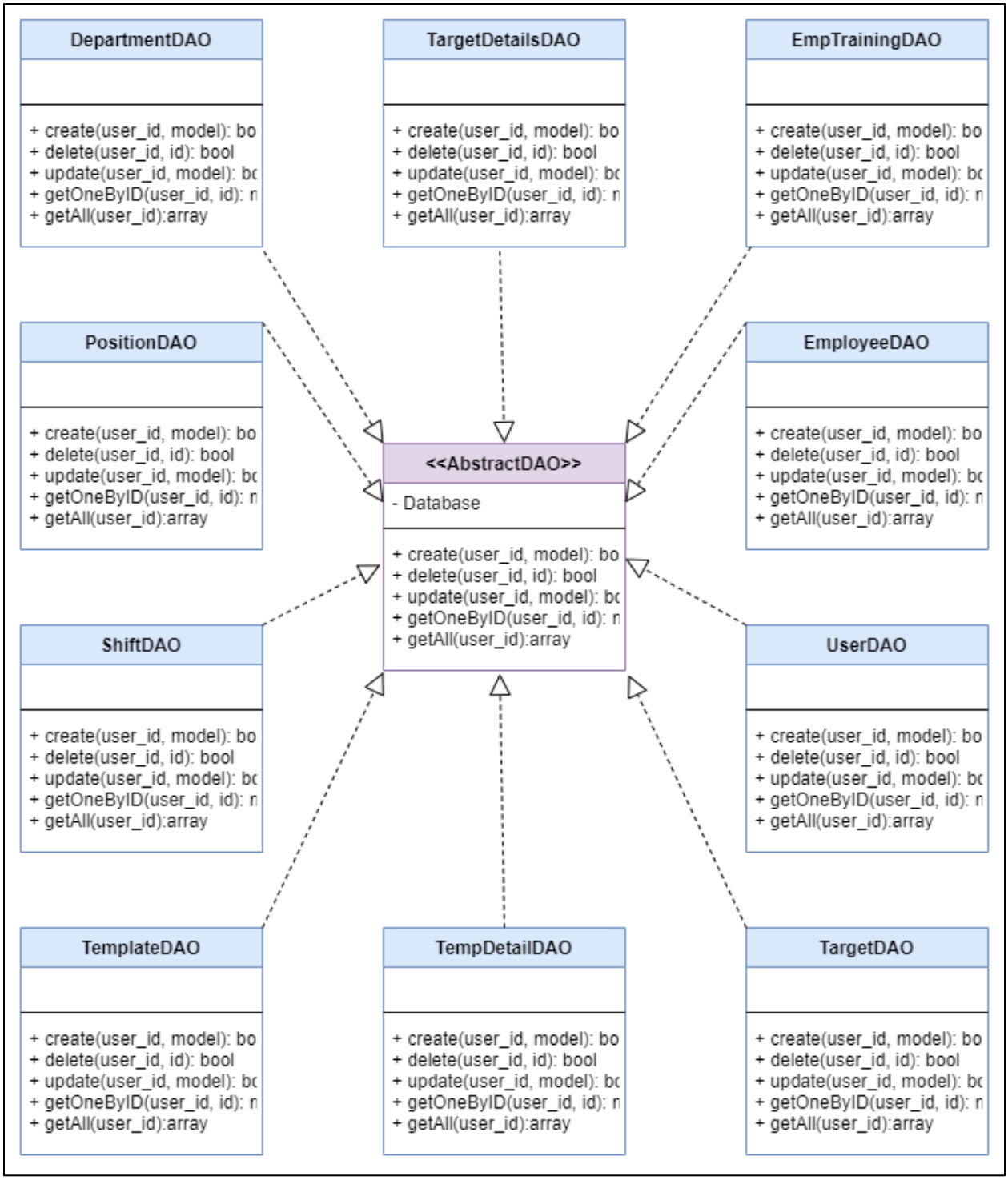


Image: UML Class Diagrams (DAO – Extends Abstract DAO – Utilized in Database/Business Services)

Sitemap Diagram:
Image file of your Sitemap diagram.

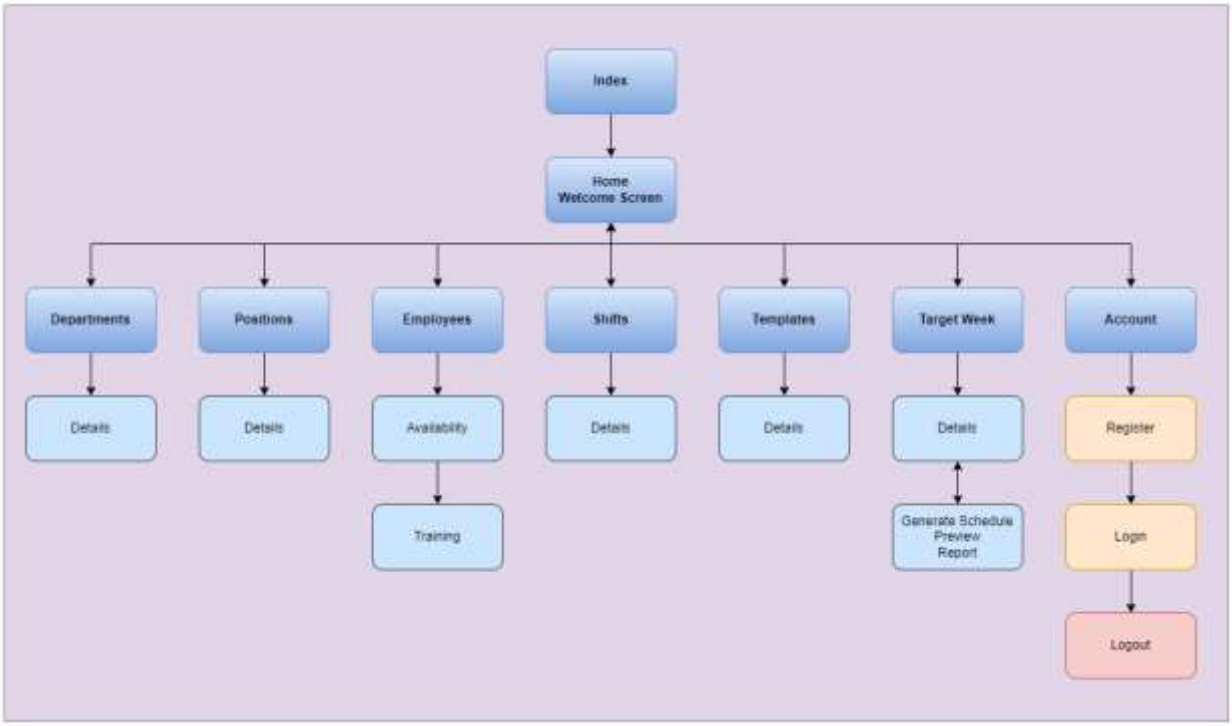
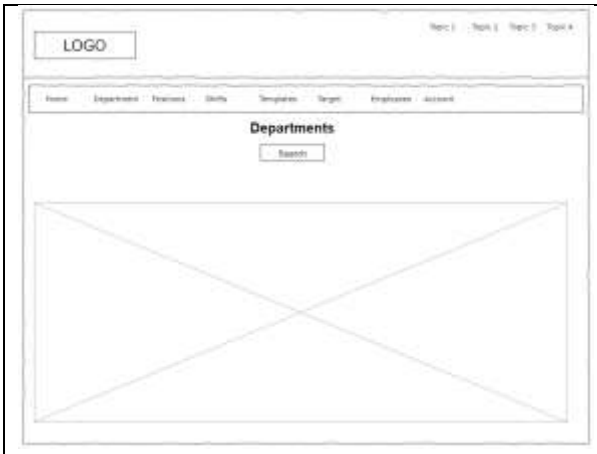
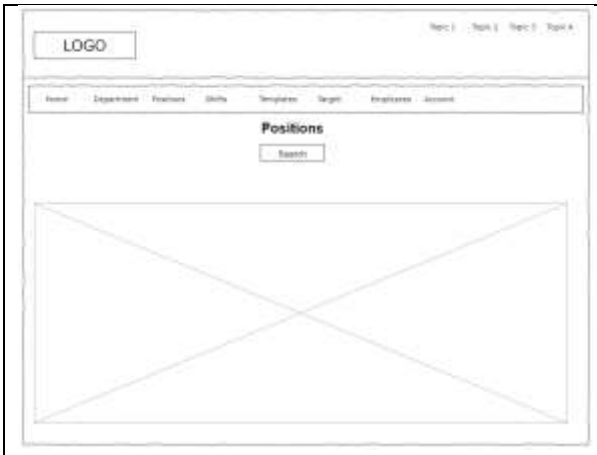
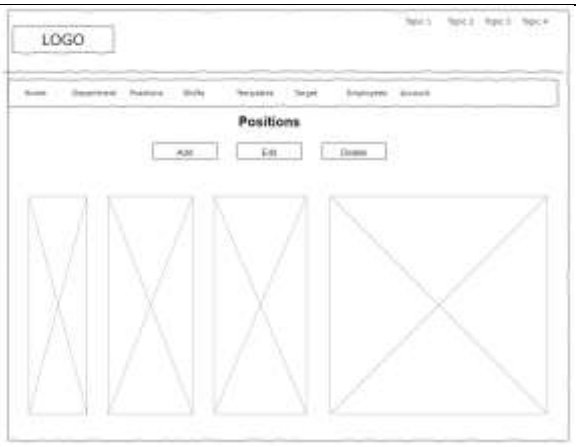
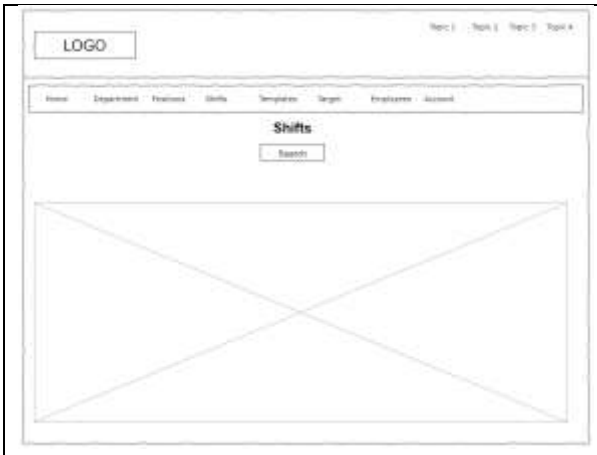
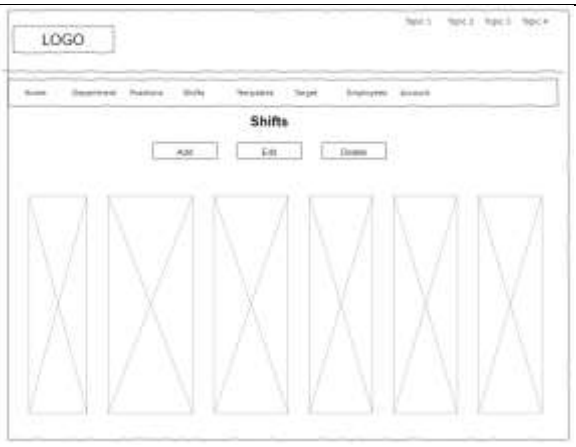


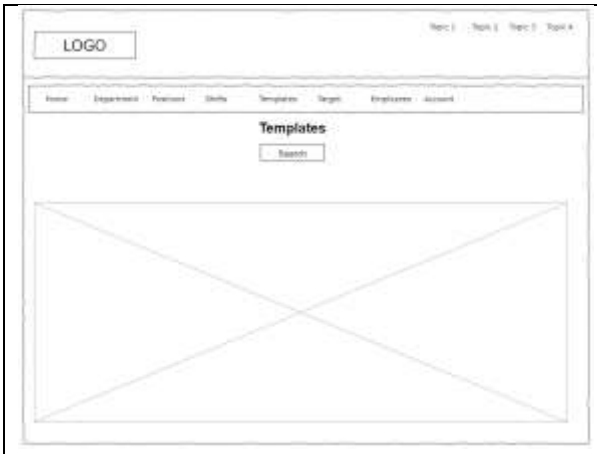
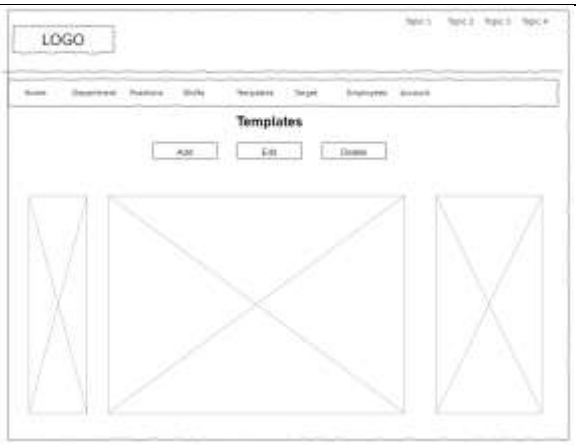
Image: Site Map

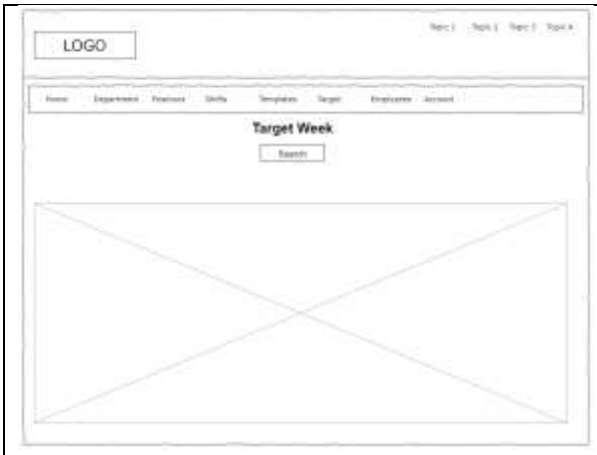
User Interface Diagrams:
You should insert any wireframe drawings or white board concepts that were developed to support your application. If you have no supporting documentation, explain the rationale why you are able to leave this section as N/A.

Wireframes

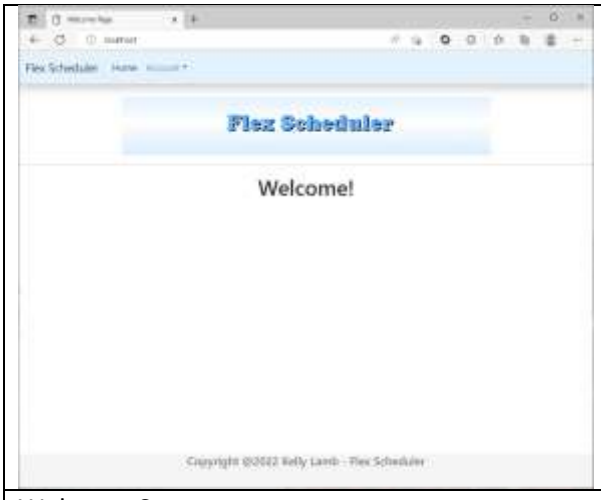
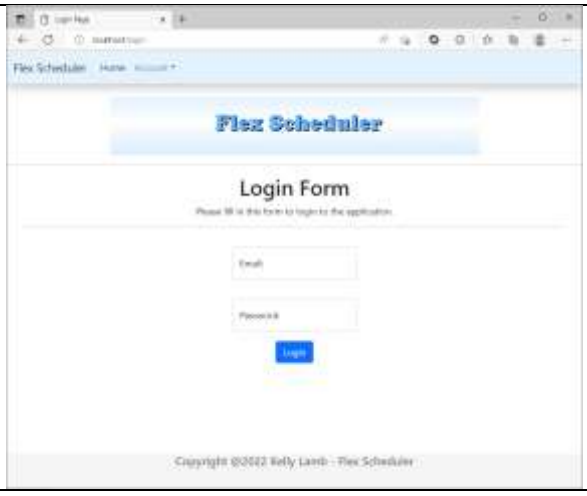
	
Welcome Screen	Login / Register Screen

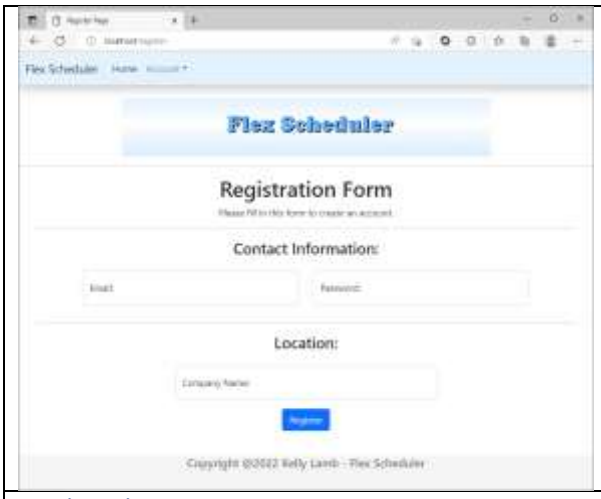
	
Departments Main (Listing / Search)	Departments Detail (Create / Edit)
	
Positions Main (Listing / Search)	Positions Detail (Create / Edit)
	
Shifts Main (Listing / Search)	Shifts Detail (Create / Edit)

	
Templates Main (Listing / Search)	Templates Detail (Create / Edit)

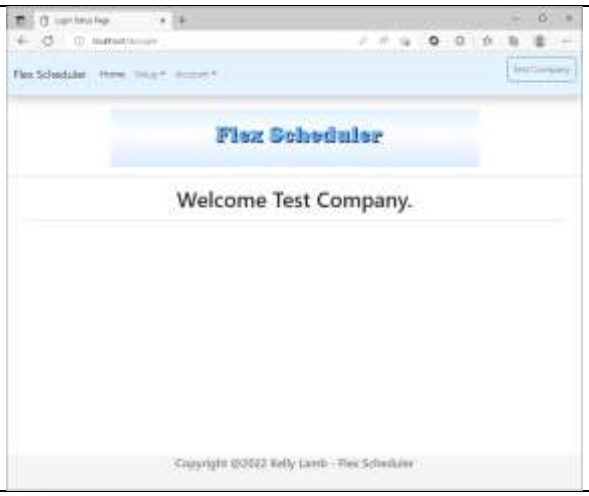
	
Target Week Main (Listing / Search)	Target Week Detail (Create / Edit)

Screenshots

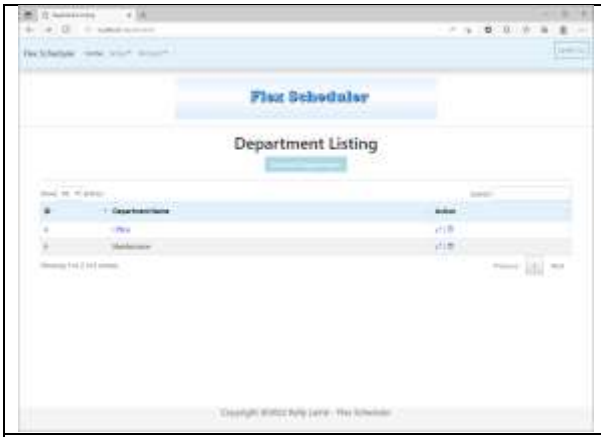
	
Welcome Screen	Login Screen



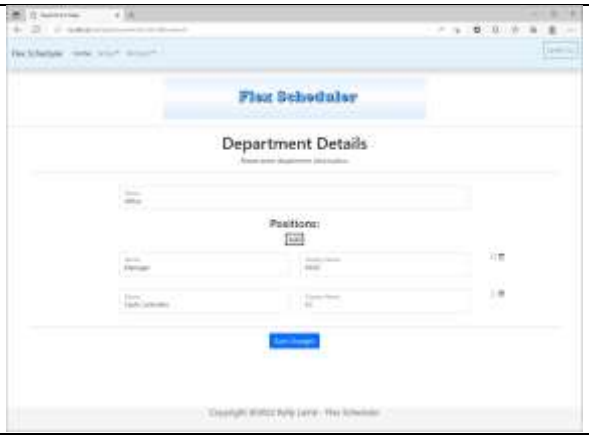
Registration Screen



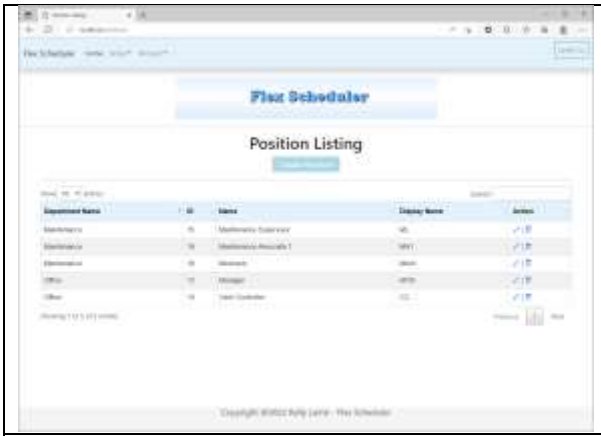
Welcome Screen after success login



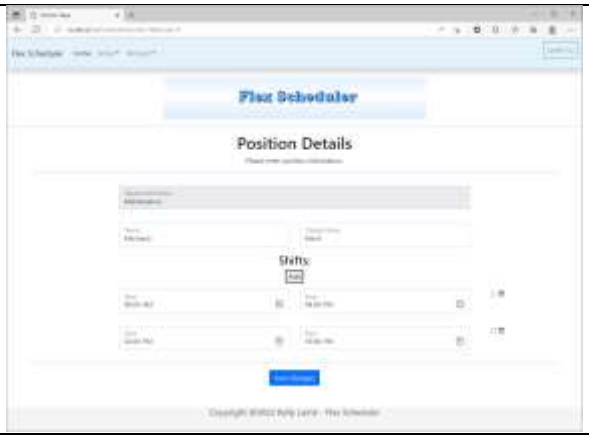
Listing - Departments



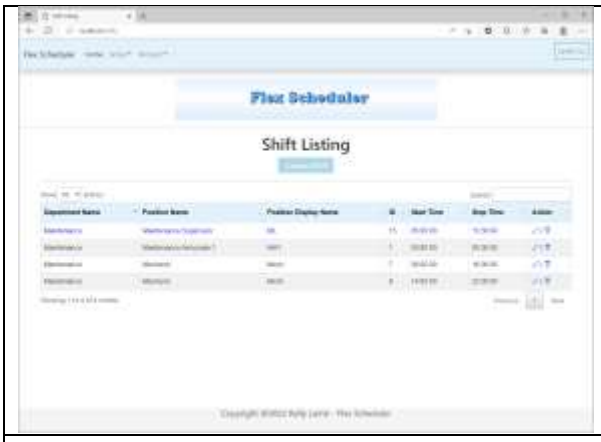
Details - Departments



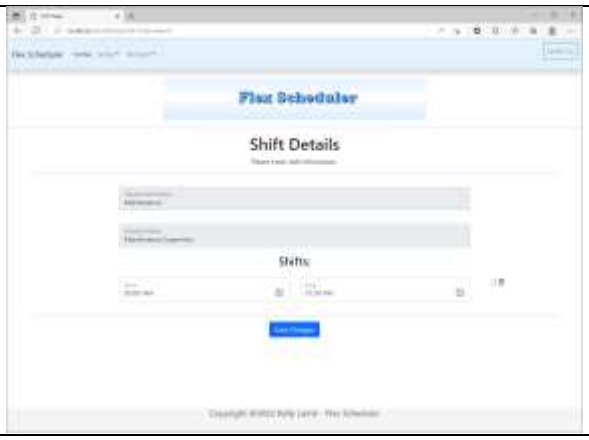
Listing - Positions



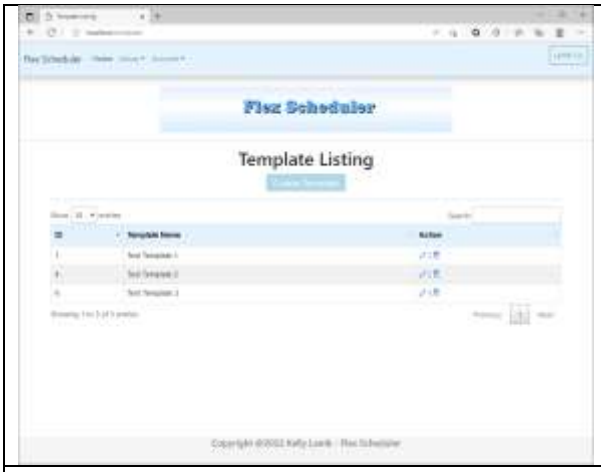
Details - Positions



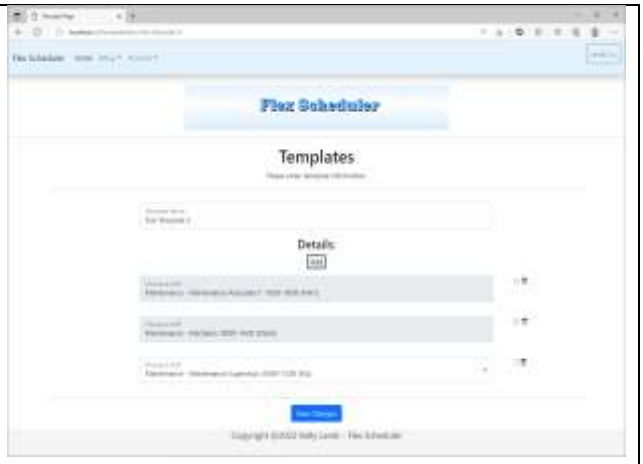
Listing - Shifts



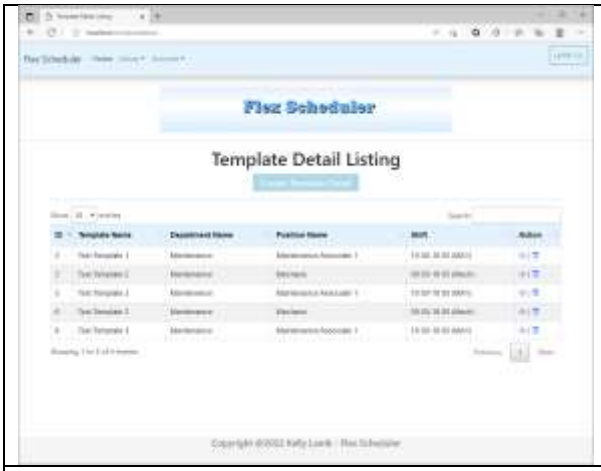
Details - Shifts



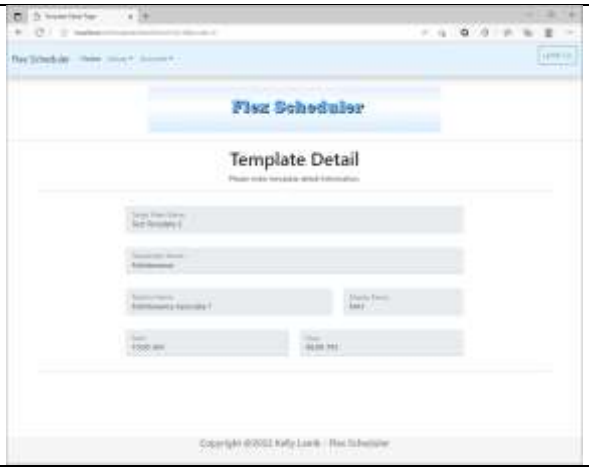
Listing - Templates



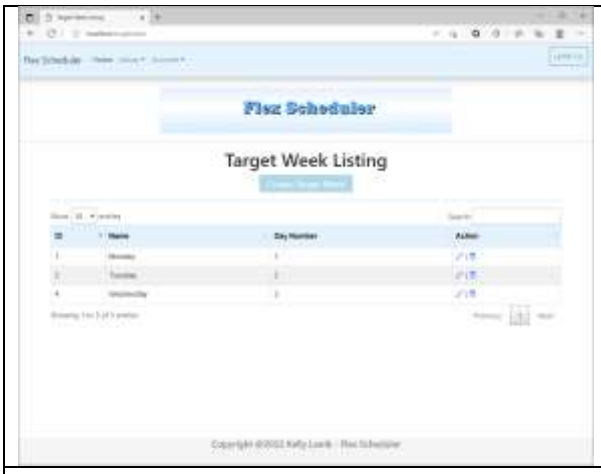
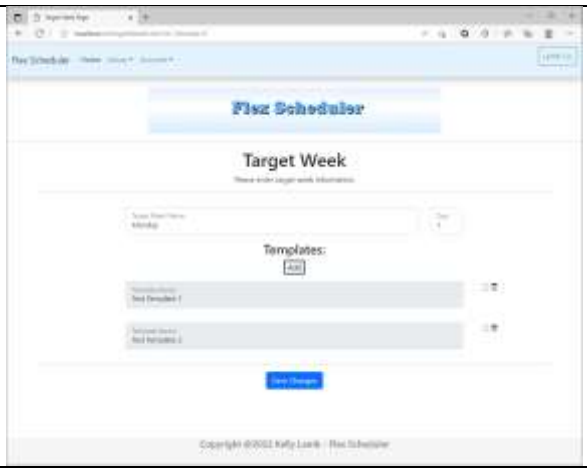
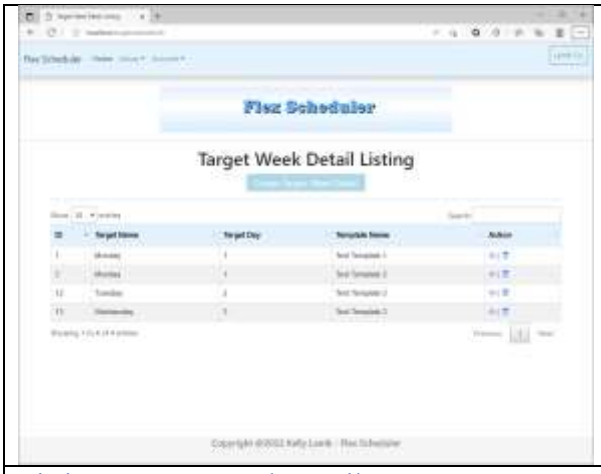
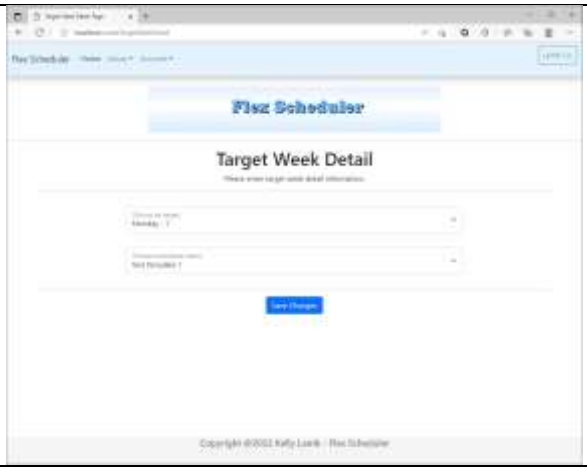
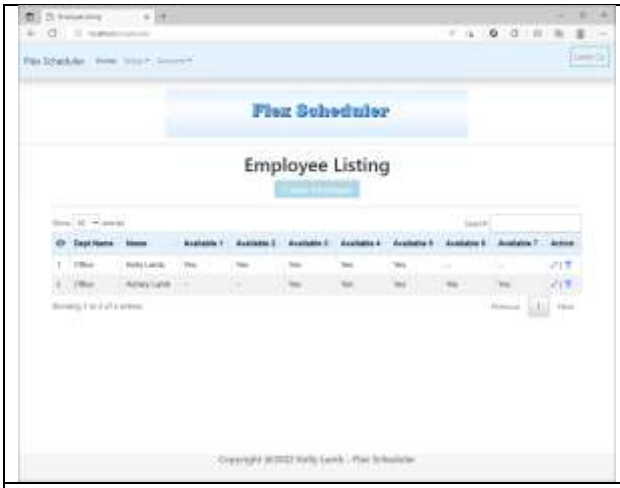
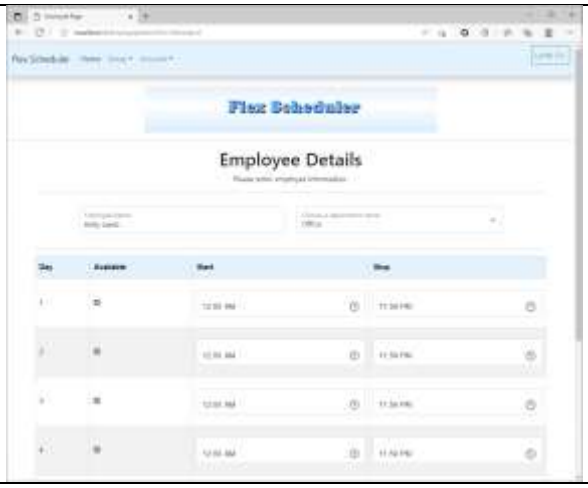
Detail - Templates

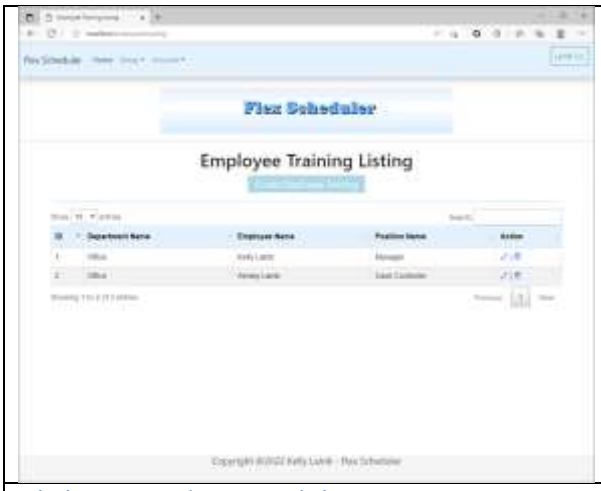


Listing – Templates Detail

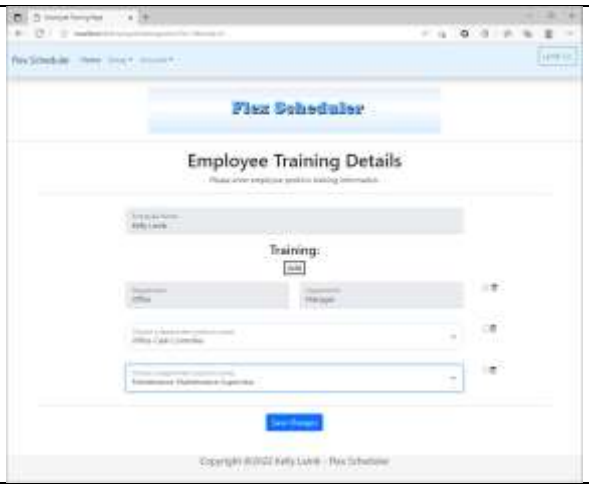


Detail – Templates Detail

	
Listing – Target Week	Details – Target Week
	
Listing – Target Week Details	Details – Target Week Details
	
Listing - Employees	Details - Employees

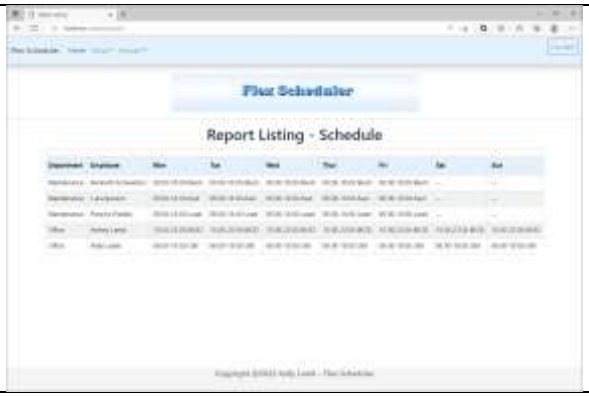


Listing – Employee Training



Details – Employee Training

Report – Create Schedule



Report – Schedule Generated

UML Diagrams:
You should insert any UML Class diagrams and UML Sequence diagrams here. If you have no supporting documentation, explain the rationale why you are able to leave this section as N/A.

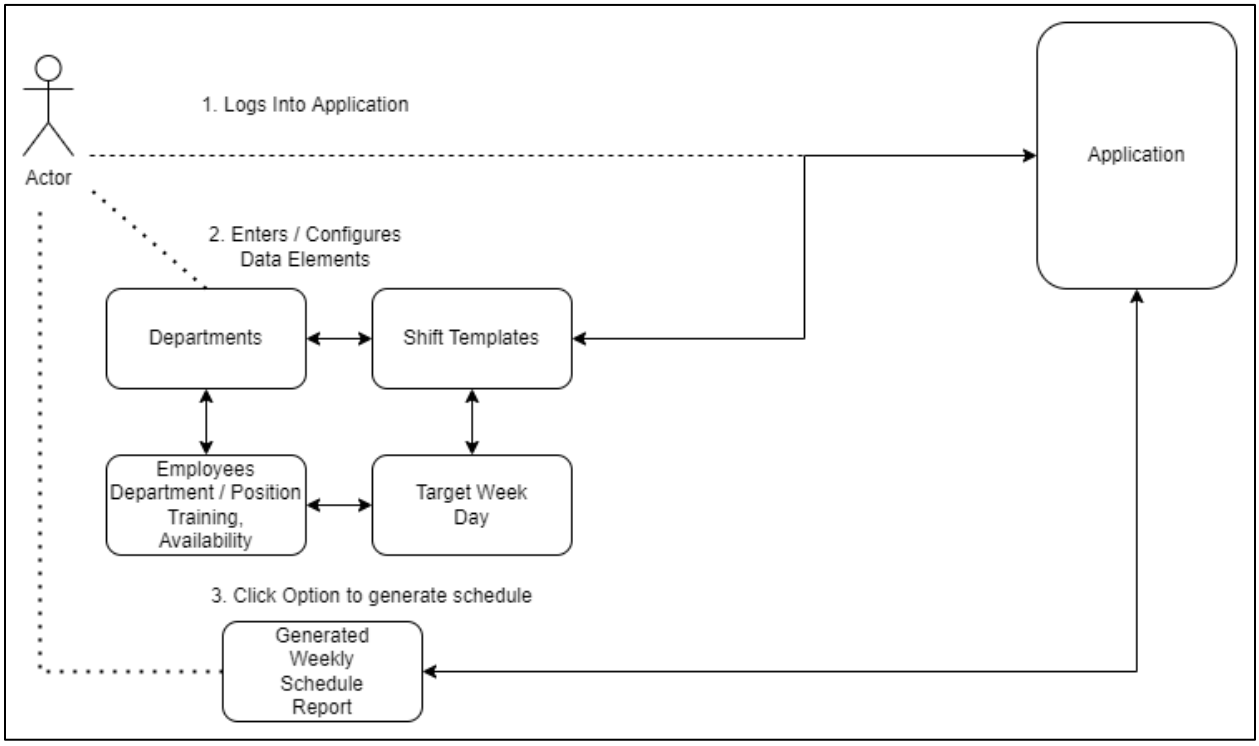


Image 1. Use Case – Sequence Diagram

Service API Design:
This section should fully document any third party Service Interface APIs being consumed or application specific Service API's being published, how to access the service, what parameters are required by the API, and the detailed JSON data format specification that could be used by a third party developer to integrate with the service and API.

N/A: No external third party services will be required for this project.

NFR's (Security Design, etc.):
This section should outline how non-functional requirements will be supported by the design.

1. One user = one account (schedule work area)
2. User must register and information stored in database
3. At time of login, authenticate login against the database (store login info in global context)
4. All functionality is associated with valid account login (authorized via global context)

Operational Support Design:
This section should fully document how your design supports monitoring and logging.

1. PHP / Laravel provides a built-in logging mechanism which, once configured, will store information to see all transactions throughout the processes and help to identify and resolve any issues (stored on site in storage/logs directory).
2. May, in later iterations, alter logging mechanism to send logs to Loggly and use provide monitoring via Uptime Robot (both external licensed, fee-based options).

Other Documentation:

Reports Design

Provide a listing of the reports that the system will provide, if applicable. If not, state that the system does not produce any reports and provide additional documentation as described in the handbook.

1. An HTML grid displaying proposed schedule
2. A downloadable (Spreadsheet) for the proposed schedule
3. Listings for editing the following areas:
    a. Departments
    b. Positions
    c. Employees
        i. Availability
        ii. Training
    d. Shifts
    e. Shift Templates
    f. Target Week Schedules

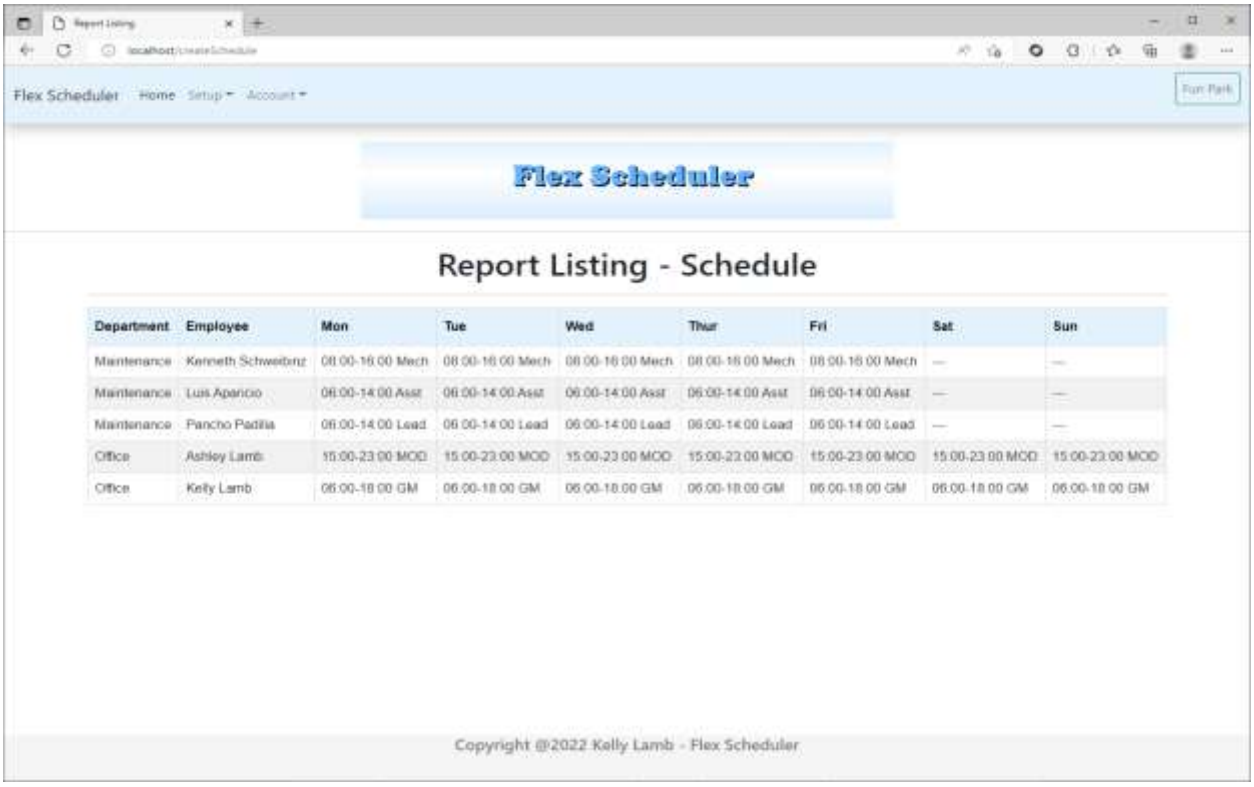
File Date: 06/26/2022 18:48:45

Last Updated: 06/25/2022 2:12:27 PM

June 27 to July 03, 2022

Employee Name	Mon 6-27	Tue 6-28	Wed 6-29	Thu 6-30	Fri 7-01	Sat 7-02	Sun 7-03
Golf / Arcade	10 - 10	10 - 10	10 - 10	10 - 10	10 - 11	10 - 11	10 - 10
Ride hours	12 - 7	12 - 7	2:30 - 9:30	12 - 7	2:30 - 9:30	1 - 9	1 - 9
Snack Bar hours	Closed	Closed	Closed	Closed	CLOSED	1 - 7	1 - 7
Lamb, Kelly	8 - 5 GM	8 - 5 GM	8 - 5 GM	8 - 5 GM	8 - 5 GM	OFF	OFF
Aleman, Veronica	7 - 3:30 CC	7 - 3:30 CC	INVENTORY	Prize Room	N/A	8 - 4 CC	REQUEST OFF
Blancas, Aracely	REQUEST OFF	N/A	9 - 5 CC	7amPD/CC	9 - 5 MOD	10 - 6 MOD	8 - 3 MOD
Gomez, Lisa	N/A	N/A	REQUEST OFF	8am PD/SCH	8 - 4 CC	GOLF	8 - 3 CC
Hernandez, Richard	3 - 10:30 MOD	3 - 10:30 MOD/Orient	OFF	OFF	2:30 - 9:30 IB	4:30 - 11 CD	3 - 10:30 MOD(TR)
Mejia, Alan	N/A	N/A	6 - 10:30 SCH/TR	N/A	6 - 11:30 L	6 - 11:30 L	12:15 - 8 L
Morales, Omar	REQUEST OFF	N/A	4 - 10:30 MOD(TR)	3 - 10:30 MOD	5 - 11:30 MOD(TR)	4 - 11:30 MOD(TR)	REQUEST OFF
Ortiz, Diane	N/A	N/A	N/A	N/A	5 - 11:30 MOD(TR)	4 - 11:30 MOD(TR)	3 - 10:30 MOD(TR)
Schweibinz, Kenneth	9 - 5 MECH	9 - 5 MECH	OFF	9 - 5 MECH	OFF	9 - 5 MECH	9 - 5 MECH
Rodriguez, Anthony	3:30 - 7 MECH	OFF	3:30 - 7 MECH	OFF	3:30 - 7 MECH	OFF	OFF
Chavez, Fabian	N/A	N/A	N/A	N/A	N/A	2 - 6 PS	2 - 6 PS
Merino, Brian	N/A	N/A	REQUEST OFF	N/A	N/A	N/A	N/A

Image: HTML Report for proposed schedule – downloadable to spreadsheet



Actual Image vs Mock-Up above

Project Task	Hours	Cost
Establish Development Environment	8	$ 800.00
Establish Repository	1	$ 100.00
Establish Deployment Server	4	$ 400.00
Define Database &Tables	8	$ 800.00
Define DAO	8	$ 800.00
Define Data Layer Services	8	$ 800.00
Define Business Layer Services	8	$ 800.00
Define Security Layer Services	4	$ 400.00
Define User Interfaces	16	$ 1,600.00
Local Test	2	$ 200.00
Deploy Test	2	$ 200.00
Estimated Cost ($100.00/Hour)	**69**	**$ 6,900.00**

Table: Cost Estimation Schedule

Version 1.0: (Iteration / Sprint 1.0)

Start Date: 07/03/2022

End Date: 07/10/2022

Project Task	Hours	Start Date	End Date
Establish Development Environment	8	7/3/2022	7/4/2022
Establish Repository	1	7/4/2022	7/4/2022
Establish Deployment Server	4	7/4/2022	7/4/2022
Define Database &Tables	8	7/5/2022	7/5/2022
Define DAO	8	7/6/2022	7/6/2022
Define Data Layer Services	8	7/7/2022	7/7/2022
Define Business Layer Services	8	7/8/2022	7/8/2022
Define Security Layer Services	4	7/9/2022	7/9/2022
Define User Interfaces	16	7/9/2022	7/10/2022
Local Test	2	7/10/2022	7/10/2022
Deploy Test	2	7/10/2022	7/10/2022

Table: Time Estimation Table

Appendix A – Technical Issue and Risk Log

1. Use the template to identify and monitor project issues and risks.

Issues and Risk Log								
Issue or Risk	**Description**	**Project Impact**	**Action Plan/Resolution**	**Owner**	**Importance**	**Date Entered**	**Date to Review**	**Date Resolved**
I/R	What is the issue or risk?	How will this impact scope, schedule, and cost?	How do you intend to deal with this issue?	Who manages this issue?				
R	Security (Employee Info)	Low	Someone obtains names and times available to work at a specific location, training	Developer Access by secured login only	High			
R	Hosting Site Downtime	Low	Local system for backup purposes	Client Export data for backup on local system	Low			
I	Insufficient Employees to fill schedule requirements	Low	Alert Client	Client User must resolve shortages or reduce scheduling requirements	High			

Appendix B – References

PHPSpreadsheet (n.d.). Welcome to PhpSpreadsheet. https://phpspreadsheet.readthedocs.io/en/latest/

Sparta (n.d.). Laravel Excel. https://laravel-excel.com/

Appendix C – External Resources

GIT URL:	*https://bitbucket.org/klambgcu/cst-452/src/master*
Hosting URL:	*https://flex-scheduler.herokuapp.com/*

