Medical School LIC Scheduler

SRS Document

Version 1.5

Revision History

Date	Version	Description	Author
04/April/2019	1.0	Initial draft	Katie Ortstadt
21/April/2019	1.1	Integrate suggestions. Still need to add Excel/CSV files	Huy Bui
25/April/2019	1.2	Updated server info. Minor revisions. Added UC08, UC09, UC10	Alexander Parris
25/April/2019	1.3	Added example Excel/csv file images	Justin Herold
25/April/2019	1.4	Editing/Updates	Justin Herold
26/April/2019	1.5	Final review/updated data dictionary/added example student schedule	Zach Alaniz

Table of Contents

Software Requirements Specification

1.   Introduction

1.1   Purpose

This Software Requirements Specification (SRS) describes the functional and nonfunctional requirements for software release 1.0 of the TCU/UNTHSC Medical School LIC Scheduler. This document is intended to be used by the members of the project team who will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here are committed for release 1.0.

1.2   Intended Audience and Reading Suggestions

Numerous audiences rely on this SRS.
- The LIC administrator needs to confirm requirements.
- Project managers base their estimates of schedule, effort, and resources on the requirements.
- Software development teams need to know what to build.
- Testers use it to develop requirements-based tests, test plans, and test procedures.
- Maintenance and support staff use it to understand what each part of the product is supposed to do.
- Documentation writers base user manuals and help screens on the SRS and the user interface design.

1.3   Product Scope

This Software Requirements Specification pertains to the TCU/UNTHSC LIC matching and scheduling system which will be applied to the Texas Christian University and University Northern Texas combined medical school. This system will be developed by a team of five TCU computer science students. The system will allow medical students to create clerkship schedules for their second year at the TCU/UNTHSC medical school. The system handles actual schedule creation, and exports the schedules as an excel or .csv file.

1.4   Definitions, Acronyms, and Abbreviations

See the project's Glossary for a comprehensive list of terms.

1.5   References

Yellow Walkman, Vision Document, Version 2.0
https://docs.google.com/document/d/1hvIxEQU5bO59SyzB9nmS1Y2dVMwIcJL4OgjJ43QuPtU/edit

Yellow Walkman, Software Development Plan, Version 3.0

https://docs.google.com/document/d/1QvNL3HxcYLdBNrWpZnOnXhFUKca4pvUlKnKJir60NuE/edit

1.6   Overview

The rest of this document outlines the software requirements for the LIC Scheduler. It includes specifications on reliability, useability, security, etc. This document also explains the layout of the database and what reports the software shall generate. This guide should be referenced any time additional development is made on the project.

2. Overall Description

2.1 Product Perspective

The TCU/UNTHSC LIC Scheduler program is a brand new software system intended to create student schedules for the medical school. It will be used with the very first class of medical students, and, as such, there is no existing process or software for scheduling. The context diagram as shown in Figure 1 illustrates the external entities and system interfaces. The project is considered complete, but may be changed in the future depending on the medical school's needs.
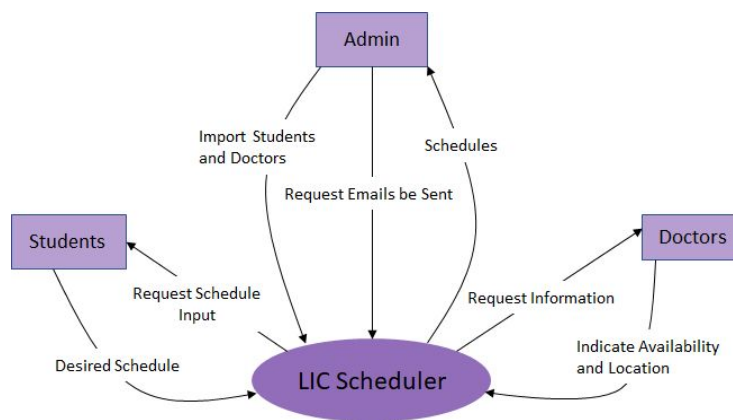

Figure 1

2.2 User Classes and Characteristics

Table 1 shows is a table of user classes for our system.

User Class	Description
Administrator	The Administrator is in charge of overseeing the scheduling process. There is currently one administrator. This administrator inputs student and doctor information and decides when to begin the scheduling process The schedules will be downloaded and distributed by the administrator.
Doctors	Doctors who use our system are those in the DFW area who have agreed to mentor a medical student. There are approximately 500 doctors who will use the system. They only use the LIC program to indicate their location and availability.
Students	The medical school Students will use the system to design their schedules. There will be about 60 students, and they will only need to access the program once.

Table 1

2.3 Operating Environment

OE-1: The program shall operate correctly with the following web browsers: Firefox, Google Chrome, and Safari.

OE-2: The program shall operate on a server which can run our Spring Boot Java program.

OE-3: The program shall permit user access from the corporate intranet; from a VPN Internet connection; and by Android, iOS, and Windows smartphones and tablets.

2.4   Design and Implementation Constraints

CO-1: The system's development shall conform to the Agile Methodology and the Agile Manifesto.

CO-2: All HTML code shall conform to the HTML5 standard.

2.5   Assumptions and Dependencies

AS-1: Each medical school student will be paired with seven different doctors from all seven specialties.

AS-2: There will be a sufficient number of doctors so that all medical school students can fulfill their clerkship requirements.

DE-1: The operation of the LIC Scheduler depends on the Gmail SMTP functioning correctly to send out doctor and student emails.

DE-2: The LIC Scheduler depends on Java JDK 8.0 or higher to be installed.

DE-3: The LIC Scheduler's online reliability depends on the hosting platform, which is currently AWS.

3.   Specific Requirements

3.1   UC02 - Doctor Enters/Edits Information

Brief Description:

This use case documents how doctors can enter and edit their information (e.g: available time, location, etc) into the database of the system. Entering and editing information involves the same course of steps.

Primary Actor: Doctor

Level: User goal level

Stakeholders and Interests: Doctors and admins

Preconditions: Doctor has the unique link to the prefilled form containing his/her information. This form pulls data from the system's database.

Postconditions: The information entered is saved in database. The doctor gets notified about the changes he/she has made.

Trigger: Doctor clicks on the link to his/her unique form.

Main Success Scenario:

1. The system retrieves and displays all the current information related to the doctor (availabilities, general location, address, number of students willing to teach)
2. The doctor edits any information that he/she wants and submits the information back to the system.
3. The system collects and verifies the information with the user.
4. The system saves the information to database.
5. The system notifies the doctor that it has saved the information.

Extensions:

1a. The system fails to retrieve the information from the link

> 1a1. The system identifies the error (the link has expired, user has been removed previously, corrupted data in database, or unknown error)
> 1a2. The system notifies the user and provides suggestions for fixing the problem
> 1a3. The system saves the log and notifies the LIC administrator.

3a. The system detects incorrect format

> 3a1. The system identifies the wrong-formatted fields
> 3a2. The system notifies the user to change those fields according to set format.
> 3a3. The system returns user to Step 2 in Main Success Scenario

4a. The system fails to save the information to database

> 4a1. The system notifies user to wait for a few minutes before trying to submit again.
> 4a2. The system saves the error log and notifies the admin.

Priority: High

Secondary Actors: N/A

Non-Functional Requirements:

1. The link should work on all browsers, or at least all the major browsers, e.g. Chrome, Internet Explorer, Edge, Firefox, Safari

Open Issues:

N/A

3.2   UC03 - Admin Views Schedules

Brief Description:

This use case allows admins to view the schedules of doctors and students. The admin will want to easily view the schedules generated by the algorithm.

Primary Actor: Admin

Level: User goal level

Stakeholders and Interests: Admin

Preconditions: The admin must have logged onto the system, and the students must have already created their schedules.

Postconditions: No changes should be made to the database after this use case is completed.

Trigger: This use case begins when the admin selects the "view schedules" option.

Main Success Scenario:

1. The system inquires whether to display students or doctors

2. The user indicated whether they would like to view students or doctors
3. The system will display a list of all of the selected group
4. The admin indicates which particular student or doctor they are interested in viewing.
5. The system displays the indicated person's schedule.

Extensions:

3a. There is no schedule for a doctor or student that is chosen.

      3a1. The system shall display an error message and continue to display all students and doctors.

Priority:

Secondary Actors: None

Non-Functional Requirements: The schedules should be viewable in all major internet browsers. This use case should not be dependent upon any specific hardware or browser.

Open Issues:

3.3   UC04 - Student Selects Clerkships

Brief Description:

This use case allows students to select their 2-week recurring schedule.

Primary Actor: Student

Stakeholders and Interests: Admin and students

Preconditions: The doctors' location and availability data is already in the database. The students' names, emails, and phase 1 doctors are already in the database. The students have been sent a unique link.

Postconditions: Clerkships are saved in the database.

Trigger: This use case begins when the student is directed to the clerkship selection page.

Main Success Scenario:

1. The system loads the clerkship selection page, along with the student's phase 1 clerkship info.
2. The student indicates which clerkship they would like to schedule, and their prefered location.
3. The system shall retrieve all clerkships of the type selected from the database.
4. The system shall populate a calendar with available time slots.
5. The student shall select an available time slot.
6. The system shall indicate that the selection was made.
7. The student and system shall repeat 2 - 6 until the student has selected time slots for all seven clerkships.
8. The student indicates that they want to submit their schedule.
9. The system shall verify that the desired schedule is still available.
10. The system alters the database to reflect the schedule created.
11. The system sends the schedule to the student via email.

12. The system redirects the student to a submission success page.

Extensions:

3a. There are no more clerkships of that type available.

1. 3a1. The system shall redirect the student to an error page that will display the problem, and the admins contact information.
2. The system shall inform the admin that an error has occurred.

3b. There are no more clerkships available in that location.

1. The system shall inform them to choose another location that is available.

9a. The desired schedule is not available.

1. The system shall clear the clerkships from the schedule that are taken. The system shall display an explanation message, and ask the student to re-pick the unavailable clerkships.

Secondary Actors: None.

Non-Functional Requirements: This use case should not be dependent upon any specific hardware or browser. It should work in all major browsers.

Open Issues: N/A

3.4   UC06 - Admin Exports Schedules

Brief Description:

This use case describes the process of an admin exporting student and doctor schedules, after they are generated.

Primary Actor: Admin

Stakeholders and Interests: Admin, students, and doctors

Preconditions: Student and doctor schedules have already been generated. Also an admin has logged into the system with a valid account.

Postconditions: A .csv or excel file will be saved to the admin's computer.

Trigger: Admin indicates that they would like to export generated schedules.

Main Success Scenario:

1. Admin indicates that they would like to export generated schedules.

2. The system displays a screen with an option to export doctor, student, or all schedules.

3. The admin selects doctor, student , or all; as well as the export format. Then submits their request.

4. The system fetches the desired schedules from the database and generates an export file.

5. The system uploads the file onto the admins browser.

Extensions:

1a. Schedules have not been generated

1a1. The admin will be alerted that not all schedules have been created. No .csv will be downloaded.

Secondary Actors: None

Non-Functional Requirements: The schedules should be downloadable in all major internet browsers. This use case should not be dependent upon any specific hardware or browser.

Open Issues: N/A

3.5   UC07 - Admin Imports Users

Brief Description:

This use case will describe the beginning of the LIC schedule generation process, from importing the students and doctors, to clearing and re-inputting the users into the database. This includes sending unique links to the students and doctors.

Primary Actor: Admin

Stakeholders and Interests: Admin, students, doctors

Preconditions: The admin has signed in to the admin portal and a no-reply email has been linked to the system.

Postconditions: The database is updated with the uploaded information.

Trigger: The admin is logged into the system and would like to import data.

Main Success Scenario:

1. Admin indicates they would like to import enrolled students and/or participating doctors.
2. System displays page allowing an admin to input file containing and doctors.
3. Admin uploads the student or doctor input file.
4. System stores the student or doctor information into the database.
5. User indicates that they want to send unique links to the doctors and/or students
6. System sends emails to the group(s) that the admin wants to send emails to

Extensions:

4a. The admin uploads an incompatible file

1a1. The system indicates an incorrect file and requests a different upload.

Secondary Actors: None

Non-Functional Requirements: The system must be able to handle a .csv file and excel files. This use case should not be dependent upon any specific hardware or browser.

Open Issues: N/A

3.6   UC08 - Super Admin Adds Admin

Brief Description:

This use case will describe the actions taken for an admin with Super Admin permission to add another admin in the database

Primary Actor: Super Admin

Level: Medium

Stakeholders and Interests:

Preconditions: The super admin has signed in to the admin portal.

Postconditions: The admin has been added to the database

Trigger: The super admin is logged into the system and would like to add an admin

Main Success Scenario:

1. Super Admin indicates they would like to add an admin.
2. System displays page allowing an admin to enter the new admin's information
3. Super Admin enters valid information in the correct fields.
4. System validates the information and adds the admin to the database
5. System displays a success message

Extensions:

4a. The admin enters in a username or email that has already been used

1a1. The system indicates that the username has already been taken.

Priority:

Secondary Actors:

Non-Functional Requirements:

Open Issues:

3.7   UC09 - Super Admin Deletes Admin

Brief Description:

This use case will describe the actions taken for an admin with Super Admin permission to delete another admin in the database

Primary Actor: Super Admin

Level: Medium

Stakeholders and Interests:

Preconditions: The super admin has signed in to the admin portal and the admin to be deleted is in the database.

Postconditions: The admin has been removed from the database

Trigger: The super admin is logged into the system and would like to delete an admin

Main Success Scenario:

1. Super Admin indicates they would like to delete an admin.
2. System displays page allowing an admin to enter an admin's username
3. Super Admin enters valid username in the username field.
4. System validates the username
5. System removes the admin from the database
6. System displays a success message

Extensions:

4a. The admin enters in an invalid username

1a1. The system indicates an incorrect username has been uploaded.

Priority:

Secondary Actors:

Non-Functional Requirements:

Open Issues:

3.8   UC10 - Admin Sends Unique Links to Doctors and Students

Brief Description:

This use case will describe the process of admin sending out mass emails with unique links to Doctors and Students after he/she imports users (UC07).

Primary Actor: Admin

Level: Admin Level

Stakeholders and Interests: Admin, Doctors, Students

Preconditions: The admin has signed in to the admin portal and a no-reply email has been linked to the system. The database is updated with the uploaded information.

Postconditions: Emails with unique links are sent to all the doctors and students in the database

Trigger: The admin is logged into the system and would like to send mass unique links to students and doctors.

Main Success Scenario:

1. Admin indicates they would like to send unique links to students and/or participating doctors.

2. System sends emails to a certain number of emails to either doctors or students with their respective unique links through their emails.

3. System notifies admin that emails have been sent.

Extensions:

Priority: Medium

Secondary Actors:

Non-Functional Requirements:

Open Issues:

4. Data Requirements

4.1 Logical Data Model
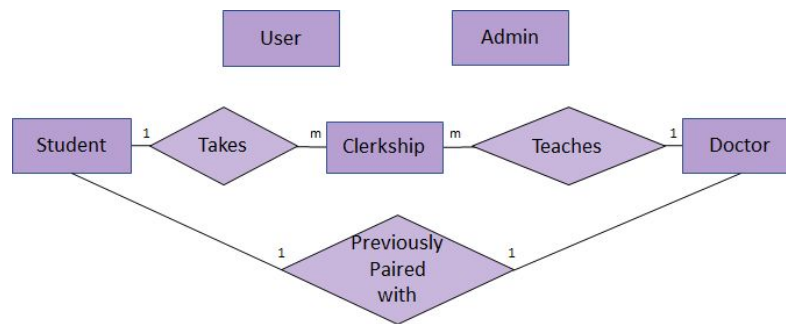
Figure 2

4.2 Data Dictionary

Student Table:

Data Element	Description	Data Type	Values
id	A unique id for each student	String	Randomly-generated string
email	The student's email	String	Email

name	Student's name	String	
hasSchedule	Check if a student has made their schedule yet	Boolean	
phase1Doc	One-to-one link to the phase one doctor	Doctor	
clerkships	One-to-many link to the student's clerkships	Clerkship	
beenEmailed	Flag whether this student has been emailed yet	Boolean	

Doctor Table:

Data Element	Description	Data Type	Values
id	A unique id for each doctor	String	Randomly-generated string
email	The doctor's email	String	Email
name	Doctor's name	String	
available	Check to see if the doctor has any open availabilities	Boolean	
availabilities	Binary string with '1' indicating available and '0' indicated not available. Each spot is mapped to a time.	String	Binary string (only 1's and 0's)
numberOfDaysAvail	Number of available days	int	
specialty	The doctor's specialty	Specialty	Enum type Specialty
location	The doctor's location	Location	Enum type Location
address	The doctor's physical address	String	
numStu	The number of students a	int	

	doctor is willing to take on		
hasStu	Checks if a doctor already has a student or not	int	
clerkship	One-to-many link to the clerkships	Clerkship	
phase1Stu	One-to-one link to the phase 1 student	Student	
hasPhase1	Indicates if the doctor has a phase 1 student or not	Boolean	
beenEmailed	Flag whether doctor has been emailed yet	Boolean	

Clerkship Table:

Data Element	Description	Data Type	Values
id	A unique id for each clerkship	String	Randomly-generated string
student	Many-to-one link to the student taking the clerkship	Student	
doctor	Many-to-one link to the doctor leading the clerkship	Doctor	
date	Date of the clerkship	Date	
startTime	The start time of the clerkship	String	
endTime	The end time of the clerkship	String	
description	The description of the clerkship for LCMS+, in this case the address	String	
time	Clerkship time for week 1	TimeSlot	

time2	Clerkship time for week 2	TimeSlot	
specialty	The specialty of the doctor	Specialty	Enum type Specialty
day	Day of the week when the clerkship occurs	int	
eventType	A tag for LCMS+	String	"Clinic"

Table 4

User Table:

Data Element	Description	Data Type	Values
id	A unique id for each user	String	Randomly-generated string
email	The user's email	String	Email
password	The user's password	String	
name	The user's first name	String	
lastName	The user's last name	String	
active	Indicate if the user is active	int	
roles	The user's role in the system	Role	

Table 5

Admin Table:

Data Element	Description	Data Type	Values
id	A unique id for the admin	String	Randomly-generated string
stuFile	The name of the previously-uploaded student information file	String	
docFile	The name of the previously-uploaded doctor	String	

	information file		

Table 6

4.3 Reports

The program shall produce two reports: a Comma Separated Values file of all clerkships for uploading to LCMS+, and a visual Excel file of all schedules.

Doctor csv file example:

	A	B	C	D	E	F	G	H	I	J
1	Doctor Name	Title	Student	Description	Day	Week	Start Time	End Time	Location	Event Type
2	Quintina Wastie	Neurology	Earle Wloch		Wednesday	week 1	13:00:00	17:00:00		Clinic
3	Orrin Moffatt	Internal Medicine	Earle Wloch		Thursday	week 1	13:00:00	17:00:00		Clinic
4	Orrin Moffatt	Internal Medicine	Earle Wloch		Thursday	week 2	13:00:00	17:00:00		Clinic
5	Shani Atcherley	Family Medicine	Mari Mengue		Thursday	week 1	8:00:00	12:00:00		Clinic
6	Shani Atcherley	Family Medicine	Mari Mengue		Thursday	week 2	8:00:00	12:00:00		Clinic
7	Lacey Toke	Internal Medicine	Mari Mengue		Saturday	week 1	13:00:00	17:00:00		Clinic
8	Lacey Toke	Internal Medicine	Mari Mengue		Saturday	week 2	13:00:00	17:00:00		Clinic
9	Lotty Lovewell	Surgery	Mari Mengue		Friday	week 1	13:00:00	17:00:00		Clinic
10	Lotty Lovewell	Surgery	Mari Mengue		Friday	week 2	13:00:00	17:00:00		Clinic
11	Sherry Pleven	Psychiatry	Mari Mengue		Saturday	week 1	8:00:00	12:00:00		Clinic
12	Pren Comley	Family Medicine	Gonzales Harcase		Monday	week 1	13:00:00	17:00:00		Clinic
13	Pren Comley	Family Medicine	Gonzales Harcase		Monday	week 2	13:00:00	17:00:00		Clinic
14	Aimil O'Feeny	Neurology	Gonzales Harcase		Wednesday	week 1	8:00:00	12:00:00		Clinic
15	Randi Roskeilly	Family Medicine	Dalia Tarbatt		Monday	week 1	8:00:00	12:00:00		Clinic
16	Randi Roskeilly	Family Medicine	Dalia Tarbatt		Monday	week 2	8:00:00	12:00:00		Clinic
17	Elton Honig	Psychiatry	Maiga Childers		Wednesday	week 1	8:00:00	12:00:00		Clinic
18	Maynord Niesing	Pediatrics	Mari Mengue		Tuesday	week 1	8:00:00	12:00:00		Clinic
19	Maynord Niesing	Pediatrics	Mari Mengue		Tuesday	week 2	8:00:00	12:00:00		Clinic
20	Maynord Niesing	Pediatrics	Maiga Childers		Tuesday	week 1	8:00:00	12:00:00		Clinic
21	Maynord Niesing	Pediatrics	Maiga Childers		Tuesday	week 2	8:00:00	12:00:00		Clinic
22	Bertrand Toffoletto	Neurology	Maiga Childers		Thursday	week 1	13:00:00	17:00:00		Clinic
23	Lombard Lampert	OBGYN	Maiga Childers		Tuesday	week 1	13:00:00	17:00:00		Clinic
24	Simon Shirtcliffe	Surgery	Joby Readwin		Wednesday	week 1	8:00:00	12:00:00		Clinic
25	Simon Shirtcliffe	Surgery	Joby Readwin		Wednesday	week 2	8:00:00	12:00:00		Clinic
26	Fleurette Ebdin	OBGYN	Earle Wloch		Friday	week 1	13:00:00	17:00:00		Clinic
27	Fleurette Ebdin	OBGYN	Joby Readwin		Friday	week 1	13:00:00	17:00:00		Clinic
28	Rowen Eudall	Pediatrics	Gonzales Harcase		Saturday	week 1	8:00:00	12:00:00		Clinic
29	Rowen Eudall	Pediatrics	Gonzales Harcase		Saturday	week 2	8:00:00	12:00:00		Clinic

doctorsSchedule-2

Student csv file example:

	Student Name	Title	Doctor	Description	Day	Week	Start Time	End Time	Location	Event Type
1	Student Name	Title	Doctor	Description	Day	Week	Start Time	End Time	Location	Event Type
2	Earle Wloch	Pediatrics	Jess Peachey		Wednesday	week 1	8:00:00	12:00:00	FortWorth	clinic
3	Earle Wloch	Pediatrics	Jess Peachey		Wednesday	week 1	8:00:00	12:00:00	FortWorth	clinic
4	Earle Wloch	Neurology	Quintina Wastie		Wednesday	week 1	13:00:00	17:00:00	Denton	clinic
5	Earle Wloch	OBGYN	Fleurette Ebdin		Friday	week 1	13:00:00	17:00:00	KellerSouthLakeAlliance	clinic
6	Earle Wloch	Surgery	Allistir Larraway		Monday	week 1	13:00:00	17:00:00	Dallas	clinic
7	Earle Wloch	Surgery	Allistir Larraway		Monday	week 1	13:00:00	17:00:00	Dallas	clinic
8	Earle Wloch	Internal Med	Orrin Moffatt		Thursday	week 1	13:00:00	17:00:00	Mansfield	clinic
9	Earle Wloch	Internal Med	Orrin Moffatt		Thursday	week 1	13:00:00	17:00:00	Mansfield	clinic
10	Earle Wloch	Psychiatry	Leyla Franzetti		Tuesday	week 1	8:00:00	12:00:00	Mansfield	clinic
11	Earle Wloch	Family Medi	Itch Fenelon		Saturday	week 1	8:00:00	12:00:00	Denton	clinic
12	Earle Wloch	Family Medi	Itch Fenelon		Saturday	week 1	8:00:00	12:00:00	Denton	clinic
13	Mari Mengue	OBGYN	Brnaba Woodroofe		Wednesday	week 1	8:00:00	12:00:00	Denton	clinic
14	Mari Mengue	Neurology	Clywd Ivison		Wednesday	week 2	8:00:00	12:00:00	Mansfield	clinic
15	Mari Mengue	Pediatrics	Maynord Niesing		Tuesday	week 1	8:00:00	12:00:00	KellerSouthLakeAlliance	clinic
16	Mari Mengue	Pediatrics	Maynord Niesing		Tuesday	week 1	8:00:00	12:00:00	KellerSouthLakeAlliance	clinic
17	Mari Mengue	Surgery	Lotty Lovewell		Friday	week 1	13:00:00	17:00:00	FortWorth	clinic
18	Mari Mengue	Surgery	Lotty Lovewell		Friday	week 1	13:00:00	17:00:00	FortWorth	clinic
19	Mari Mengue	Internal Med	Lacey Toke		Saturday	week 1	13:00:00	17:00:00	Arlington	clinic
20	Mari Mengue	Internal Med	Lacey Toke		Saturday	week 1	13:00:00	17:00:00	Arlington	clinic
21	Mari Mengue	Psychiatry	Sherry Pleven		Saturday	week 1	8:00:00	12:00:00	FortWorth	clinic
22	Mari Mengue	Family Medi	Shani Atcherley		Thursday	week 1	8:00:00	12:00:00	Dallas	clinic
23	Mari Mengue	Family Medi	Shani Atcherley		Thursday	week 1	8:00:00	12:00:00	Dallas	clinic
24	Gonzales Harcase	Neurology	Aimil O'Feeny		Wednesday	week 1	8:00:00	12:00:00	Denton	clinic
25	Gonzales Harcase	OBGYN	Marketa Pillington		Tuesday	week 1	8:00:00	12:00:00	KellerSouthLakeAlliance	clinic
26	Gonzales Harcase	Pediatrics	Rowen Eudall		Saturday	week 1	8:00:00	12:00:00	Arlington	clinic
27	Gonzales Harcase	Pediatrics	Rowen Eudall		Saturday	week 1	8:00:00	12:00:00	Arlington	clinic
28	Gonzales Harcase	Surgery	Karry Towsie		Friday	week 1	8:00:00	12:00:00	Arlington	clinic
29	Gonzales Harcase	Surgery	Karry Towsie		Friday	week 1	8:00:00	12:00:00	Arlington	clinic

studentsSchedule-9

Doctor xlsx file example:

Weekly Schedule

TCU/UNT Medical School Scheduling
Doctor: Quintina Wastie
Email: qwastieb@123-reg.co.uk

WEEK 1:

Time/Period	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Morning:				LEAPS			
Afternoon:			Title: Neurology Location: Denton Student: Barbra Waister				

WEEK 2:

Time/Period	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Morning:							
Afternoon:				LEAPS			

Quintina Wastie's Schedule-2019 Orrin Moffatt's Schedule-2019 Shani Atcherley's Schedule-2019 Lacey Toke's Schedule-2019 Lotty Lovewell's Schedule-2019 Sherr

Student xlsx file example:

Report ID	LIC-RPT-1
Report Title	Student Schedules CSV
Report Purpose	The Admin needs to upload the student schedules to LCMS+, the student portal. Events in LCMS+ can be uploaded as .csv files. The report is a csv file compatible with LCMS+.
Report Users	Admin
Data Source	Database of student clerkships
Frequency and Disposition	Report is generated on demand by the Admin. The report is a .csv file that is automatically downloaded to the admin's device.
Visual Layout	Data spreadsheet (comma-separated list)
Report Body	Fields shown and column headings: ● Student Name ● Title (Clerkship Specialty) ● Doctor ● Description (Physical address of the clinic) ● Day (Day of the week, eg. "Monday") ● Week (Whether this clerkship occurs on week 1 or week 2) ● Start Time

	• End Time • Location • Event type (For LCMS+, always "clinic")
Interactivity	Not interactive
Security Restrictions	Only a user logged onto an admin account can request the report

Table 7

Sample Student Schedule

	Sun	Mon	Tues	Wed	Thur	Fri	Sat
Morning	N/A		Neuro			Surg.	Pedi.
Afternoon	N/A	Psych.		Fam. Med.	LeAPS	Int. Med.	OBGYN

Week 2

	Sun	Mon	Tues	Wed	Thur	Fri	Sat
Morning	N/A					Surg.	Pedi.
Afternoon	N/A			Fam. Med.	LeAPS	Int. Med.	

Report ID	LIC-RPT-2
Report Title	Visual Representation of Schedules
Report Purpose	The Admin may want to view all the student and doctor schedules, as well as save the schedules for record-keeping. This report downloads an excel file with a visual representation of each schedule.
Report Users	Admin
Data Source	Database of student clerkships
Frequency and Disposition	Report is generated on demand by the Admin. The report is an excel file that is automatically downloaded to the admin's device.

Visual Layout	Data spreadsheet
Report Body	The report appears as a two-week calendar visual with the clerkships filled in. The student's name and email are included at the top. The doctor, specialty, and location are included with each listed clerkship.
Interactivity	Each tab on the spreadsheet represents a different student.
Security Restrictions	Only a user logged onto an admin account can request the report

Table 8

4.4 Data Acquisition, Integrity, retention, and disposal

DI-1: The system shall retain student information (names, emails, phase 1 doctors) until a new student or doctor information file is uploaded.

DI-2: The system shall retain doctor information (names, emails, specialty) until a new doctor information file is uploaded.

DI-3: The system shall retain clerkship information (times, specialty, doctor, student, etc.) until a new student or doctor information file is uploaded.

DI-4: The system shall retain admin login information (username, password) indefinitely.

DI-5: The system shall discard all clerkship data (doctor, student, time, specialty) after generating the two final reports (CSV and Excel reports).

5. External Interface Requirements

5.1 User Interfaces

The LIC Scheduler does not follow specific interface standards. Future development should just focus on matching existing UI as best as possible. Because few people interact with the system, however, styling is not emphasized. The only requirement is that the website use official TCU/UNTHSC Medical School logos when appropriate.

All Administrator web pages must include a header with a link back to the main admin panel.

5.2 Software Interfaces

Because there is no existing precedence for an LIC scheduling system, all the software used for this program was created within our Spring project. The LIC scheduler generates a .csv data output compatible with the student portal LCMS+, but this file must be uploaded independently by the user. Our system does not directly interface with LCMS+ in any way.

5.3 Hardware Interfaces

Our project is not an embedded system, and thus does not have any hardware interfaces.

5.4 Communications Interfaces

CI-1: The LIC Scheduler shall send an email to each doctor in the database with a unique link directing them to their availability form.

CI-2: The LIC Scheduler shall send an email to each student in the database with a unique link directing them to their scheduling page.

6. Quality Attributes

6.1 Usability

USE-1: An administrative user should be able to successfully navigate the system with less than 1 hour of training, provided by the user manual.

USE-2: A doctor user should be able to successfully navigate the system with no training. The GUI for the doctor form should be straightforward and well-explained.

USE-3: A student user should be able to successfully navigate the system with no training. The GUI for the doctor form should be straightforward and well-explained. Additionally, a "how-to" guide is provided on the page.

USE-4: At least 90% of students should be able to successfully create their schedules without errors. A "contact admin" button will be provided for those who encounter errors.

6.2 Reliability

AVL-1: The LIC Scheduler shall be available at least 90% of the time during the scheduling window.

AVL-2: After schedules have been created for the year, the LIC Scheduler can be taken offline until the next year's schedules must be made.

6.3 Performance

PER-1: The system shall accommodate a total of 100 users and a maximum of 60 concurrent users during the peak usage time.

PER-2: Excluding the email function (which may take several minutes), response times after requesting a page should be less than 5 seconds.

6.4 Security

SEC-1: Administrators shall be required to log on to the LIC Scheduler to view any administrative tasks

SEC-2: Doctor forms will each have a unique link that is shared only with the appropriate doctor

SEC-3: Student scheduling pages will each have a unique link that is shared only with the appropriate student

6.5 Safety

Our program should not cause any additional safety concerns beyond the normal risks of accessing the internet. It is purely a software product.

7.   Appendix A: Analysis Models

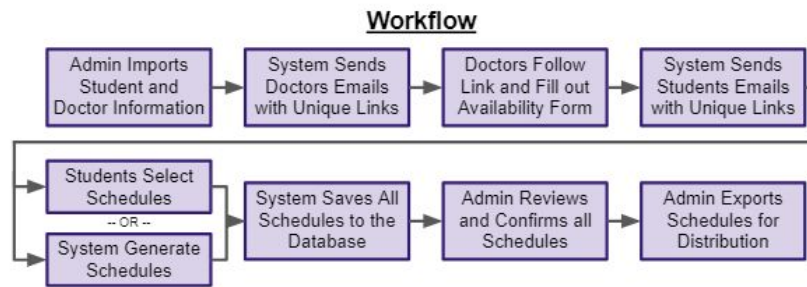Figure 2 is a workflow diagram which demonstrates how the system is used.

Figure 2