Software Requirements Specification

for

Lions Final Exam Scheduler (L.I.F.E.S.)

Version 1.0

Prepared by



|  |  |  |
| --- | --- | --- |
| Team Leader | Scott Smoke | ssmoke@una.edu |
| SQA | Riley Smith | msmith9@una.edu |
| SQA | Jordan Beck | jbeck@una.edu |
| Engineer | Joshua Ford | jford@una.edu |
| Technical Writer | Jeffrey Allen | jallen2@una.edu |

Instructor: Patricia Roden, Ph.D

Course: CS455 Software Engineering

|  |  |
| --- | --- |
|  |  |
|  |  |

Contents

RevisioNs iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 References and Acknowledgments 2

1.6 Overview 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

4 Other Non-functional Requirements 7

4.1 Safety and Security Requirements 7

4.2 Software Quality Attributes 7

5 Other Requirements 8

6 client-developer agreement 8

Appendix A – Data Dictionary 9

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| 1.0 | Jeffrey Allen | Initial draft  Added document’s purpose  Added definitions and acronyms section  Added references and acknowledgements  Added user documentation  Defined intended audience  Added overview of document  Added Product Overview  Added application perspective  Added product functionality  Edited References  Added line numbers  Added Name of Application  Removed unnecessary sections | 2/20/15 |
| 1.1 | Scott Smoke, Jeffrey Alllen | Made revisions to every section added in sections 4.3 and 4.4 | 2/27/15 |
|  |  |  |  |

# 

# Introduction

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

## Document Purpose

The purpose of this document is to give a detailed description of the requirements for the Lions Final Exam Scheduler (L.I.F.E.S.) software package. It will illustrate the purpose and complete declaration for the functionality of the system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a client for its approval and a reference for developing the first version of the system for the development team. Upon agreement of this specification document between the client, Dr. Patricia Roden, and the Tune Squad, both parties will provide their signatures (see section 6).

## Product Scope

L.I.F.E.Sis a desktop application with a graphical user interface which will allow for two administrative users and multiple general users in the University of North Alabama’s Office of the Registrar to generate a final exam schedule. The application will make use of the data from a previous similar semester to determine trends in popular class times. The L.I.F.E.S. application can be installed on a computer with a CD-ROM or a flash drive.

The goal of this software is for a user to be able to display a final exam schedule that has been generated using the software in a similar format displayed on the UNA website as of the year 2015. Previously, a person had to compose schedules by hand which could take days or even weeks to produce a schedule with minimal conflicts. This software attempts to speed up the process by providing a single user with an easy and intuitive work environment by providing a small subset of data to produce a high quality exam schedule. After a final exam schedule is created, only an administrator will be able to finalize the schedule that has been generated.

## Intended Audience

This document is intended to be read by the client professor, VPAA, the individuals located in the Office of the Registrar that will be using this software, and the Tune Squad development team.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Wish | A desirable level of achievement that may not be attainable through available means contained in a language statement |
| User | Someone who interacts with the desktop application |
| UNA | University of North Alabama |
| SRS | Software Specification Document |
| Developer | A member of the Tune Squad |
| Admin/Administrator | System administrator who is given specific permission for managing and controlling the system |
| VPAA | Vice President of Academic Affairs |
| LIFES | Lions Final Exam Scheduler |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## References and Acknowledgments

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

[2] University of North Alabama Official Colors, http://www.una.edu/graphic-standards/print-web/

## Overview

In section 2 the application’s functionality is described at a high-level. In section 3, a detailed description of its requirements is presented, including functional requirements and use cases. Afterwards, located in section 4, are the non-functional requirements such as authentication, error handling, and help messages.

# Overall Description

## This section of the SRS describes application at a high-level. The application will be explained in its context to show how it interacts with other systems and introduce basic functionality of it.

## Application Perspective

L.I.F.E.S. is a desktop application for the University of North Alabama. Its only dependency is an external printer in the event a user wishes to print a final exam schedule generated by the application.

## Product Functionality

After successfully authenticating themselves during the login processes, a user will be able to generate final exam schedules. The result of generating a final exam schedule will be based on two separate data files provided by the user. The first file provides the application with details which define the time constraints of the exam schedule. This file can either be created manually by using the software, or it can be loaded from a separate input file. The second file must be one that is loaded from a separate file, which will provide the application with the data gathered from a previous semester.

The result of generating a final exam schedule from the two files will immediately be displayed in a similar format the University’s website presents the schedule of final exams (see figure 1). A user can either view a single date of class schedules or the schedule in its entirety.

Depending upon the type of user utilizing the application, an administrator will have the option to finalize a schedule that has been generated. This will label the project as “Administratively Approved,” which will apply a seal of approval to all printed documents associated with the finalized schedule.

A user that has questions about the software can refer to the help menu. This menu will demonstrate to the user how to use the application.

## Users and Characteristics

The users of this application will be the VPAA and individuals at the Office of the Registrar. All users should be familiar with opening desktop applications and maneuvering file systems. There are two categories of users, which are either considered an administrative or a general user. At maximum there can be two administrators. Privileges are the separating factor between these two different categories of users. The privileges of each category are described below.

#### General users have the privilege to authenticate themselves as valid users. These general users are users who have been verified by an administrator and may operate the application. General users will be able to open, view, run schedule, reschedule, and modify the schedule.

Administrators have all the same privileges as the general users with the addition of creation and deletion of general users, reset passwords, finalize exam schedule and unlock accounts.

## Operating Environment

The environment in which the software will be developed and operated upon is in an HP Pro running a 64-bit running a Windows 7 operating system.

## Design and Implementation Constraints

**Programming Language Proposal**

Any hardware requirements should be limited to the HP pro.

## User Documentation

The user manual for this application can be found electronically by first clicking the tab labeled “Help” in the menu bar of the application, and then clicking “User’s Manual”. A hard copy of the manual will accompany the final submission of the project.

# Specific Requirements

Descriptions of all functional requirements are described in this section. There are detailed descriptions of all the interfaces in the system and all of the application’s features.

## External Interface Requirements

This section provides a detailed description of all inputs into and outputs from the application. It also contains a description of the hardware and software interfaces. Basic prototypes of the user interface are also provided in this section.

### User Interfaces

A first-time user of L.I.F.E.S. should see the log-in page when he/she opens the application (See Appendix 1). If the user has not been registered by the VPAA, he/she should not be able to access any other interface. If the user is not a first-time user, he/she should be able to login directly when the application is opened. Here the user chooses the type of search he/she wants to conduct.

The main interface will have a table that will display a generated schedule. Above the table there will be two buttons to select the two input files. Below the table there will be a button to generate an exam schedule and a button to reschedule the currently displayed schedule and a swap button. The main interface will contain a menu bar with the following menu options: File, Help, and Log Out. Under the file option there will be the sub options: new, open, save, save as, print, and close. The main interface will contain the following dialogs.

Login Dialog

The login dialog will ask for the username and password and will have the buttons login and cancel.

Open File Dialogf

The open file dialog will prompt the user to enter a file name or search for the required file. This dialog will contain the buttons open and cancel.

Save File Dialog

The save file dialog will ask the user for the filename and the format to save the file. S dialog will contain the buttons save and cancel.

Print Dialog

The print dialog will ask for the printer to print to.

Error Dialog

The error dialog will display the error and have an ok button.

Data input dialog

The data input dialog will prompt the user to enter the data that would otherwise be in the first file that is required for the exam schedule.

All colors and logos used in L.I.F.E.S. will adhere to the University of North Alabama graphics standards. Examples of these interfaces are in the appendix.

### Hardware Interfaces

The product will be able to communicate with printers through libraries available to the programming language of choice.

### Software Interfaces

The product will be able to communicate with the Windows 7 operating system and link

with libraries to link with a printer.

## Functional Requirements

The L.I.F.E.Sapplication will allow its users to create a final exam schedule that will make use of data from previous similar semesters to determine trends in popular class times. The application has two levels of users: General Users and Administrators.

### Authentication System

The user will be prompted for a username and password. The username must be the users’ University of North Alabama email. The password must be between seven and nine characters long. The password must start with an alphabetical character followed by any number of alphanumeric characters and may contain any of the following \*,# and $. The password is not case sensitive.

#### General User

##### Open Schedule

A user will be able to open a previously generated final exam schedule. They will be able to choose the entire schedule or a single day.

##### Generate Schedule

The user will generate a schedule using two valid input files.

##### Reschedule

When a user generates a final exam schedule, the option to rerun the final exam schedule will be available.

##### Modify Schedule

A user will be able to swap final time slots.

##### Save Schedule

The generated final exam schedule will be able to be saved in three different formats. These formats are either plain text, PDF, or a comma separated file.

##### Print Schedule

#### Administrative User

The administrator can perform all the general user task in addition to the following:

##### Create/Delete Users

##### Reset Passwords

##### Finalize Exam

##### Unlock Account

# Other Non-functional Requirements

## Safety and Security Requirements

* There are two types of users: administrator and general users. Both types of users will require login information including a username and password.
  + Usernames will be that person’s UNA email address.
  + Passwords will consist of 7-9 characters with the first character being alphabetic and the rest being alphanumeric or the symbols \*, #, $. Passwords are not case sensitive.
  + The list of username and passwords will be stored in an encrypted text file.
* Users have four attempts to enter a correct username and password. Upon the fourth failed attempt to login that user is locked out and has to be unlocked by an Administrator.
* If a user enters the wrong login credentials an error message will display whether the username or password was incorrect.
* Exam schedules can only be finalized by an Administrator.

## Software Quality Attributes

* Flexibility – The program will allow for manual modification to the exam schedule.
* Usability – The user interface will be designed to be easy to learn and use. There will also be an included user manual in the program’s help menu.
* Reliability – The program will generate an exam schedule that will always work, having no time conflicts.
* Testability – The software will be written with testability in mind. Each module of the software will be written with test cases in mind to allow for the finding of faults.
* Availability – The software will be available on a CD-ROM/Flashdrive.
* Maintainability – The software will achieve maintainability through the use of modules.

## Input File specifications

This application uses military time as a standard format.

### First file:

The first file provides the application with details which define the time constraints of the exam schedule. This file can either be created manually by using the software, or it can be loaded from a separate input file.

#### Number of days to schedule

This integer value will be between 3 and 5.

#### Beginning time of the first exam of the day

The beginning time of each exam is 0700. The final time for exams is either 5 or 5:15 p.m..

#### Length of time for each exam

The minimum is one hour fifteen minutes for each exam, and there is no maximum. Every exam can begin on either the quarter hour, half hour, or hour.

#### Length of time between exams

The minimum is ten minutes and the maximum is thirty minutes.

#### Length of time for a lunch period

This value will be optional, and there are no limits for this value.

### Second file:

The format of this file is a CSV file with the first column specifying the day and time of the class, and the second field containing the total number of students in the class.

## Output File

The first line will contain the semester and the year. The second line will specify the class data file used to generate the schedule. Then, all the data from the first file or the manually inputted data will be included in this file. Then the generated schedule which will use standard times. The format of this file will either be plain text, PDF, or CSV.

## Input Errors

Every error dealing with the two input files will be given in a pop up dialog that will list the line number and the type of error.

# Other Requirements

6 Client-Developer Agreement

|  |  |  |
| --- | --- | --- |
|  |  |  |
| *Patricia L. Roden, Ph.D, Client* |  | *Date* |
|  |  |  |
| *Scott Smoke, Team Leader* |  | *Date* |
|  |  |  |
| *Riley Smith, SQA* |  | *Date* |
|  |  |  |
| *Jordan Beck, SQA* |  | *Date* |
|  |  |  |
| *Joshua Ford, Engineer* |  | *Date* |
|  |  |  |
| *Jeffrey Allen, Technical Writer* |  | *Date* |

Appendix A – Data Dictionary

Appendix B – Interface Prototypes

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

Figure 1.1

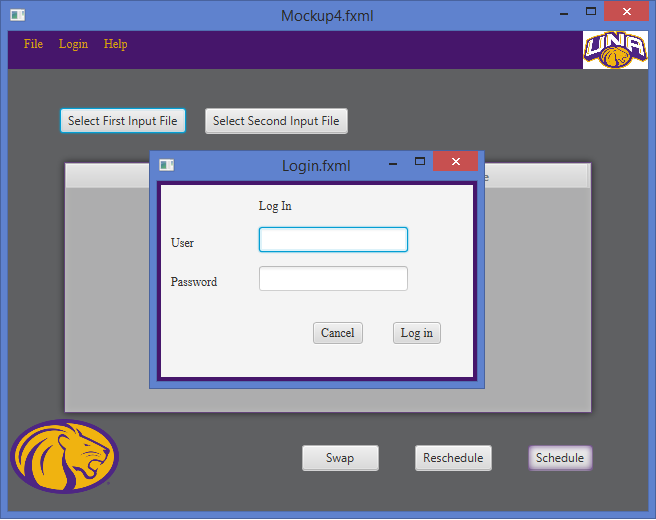
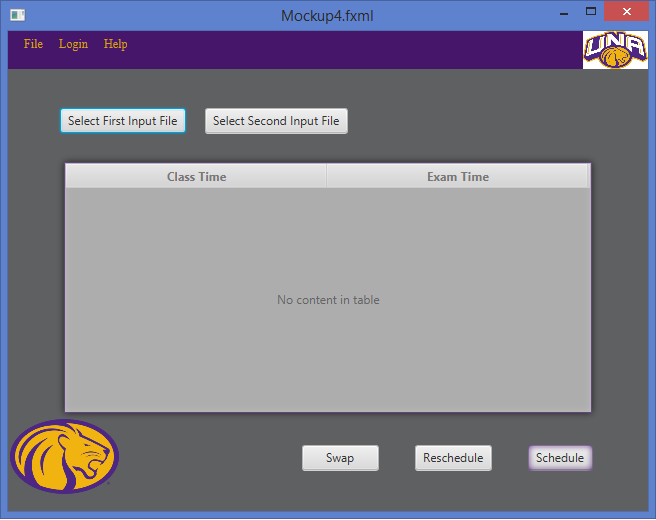
**

Figure 1.2

**

