**Conestoga** **College** **–** **SET**

**Capstone Project**

**SRS**

**Jennifer Klimova (6950851)**

**Nathan Nickel (6763692)**

**Wes Thompson (6992143)**

**Alex Martin (6693659)**

**Date** **of** **Submission** **:** **Document** **Revision** **:**

**2016-12-14** **Revision** **A2**

PD   
 (INFO 3020)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Description** **of** **Change** | **By** |
| A1 | 2016-12-10 | • Initial release of document | Jennifer Klimova  Nathan Nickel  Alex Martin  Wes Thompson |
| A2 | 2016-12-14 | • Total revision of document | Jennifer Klimova  Nathan Nickel  Alex Martin  Wes Thompson |

Contents

[1. Introduction 4](#_Toc469504814)

[1.1 Purpose 4](#_Toc469504815)

[1.2 Scope 4](#_Toc469504816)

[1.3 Definitions, Acronyms and Abbreviations 4](#_Toc469504817)

[1.4 References 5](#_Toc469504818)

[1.5 Overview 5](#_Toc469504819)

[2. Overall description 6](#_Toc469504820)

[2.1 Product perspective 6](#_Toc469504821)

[2.1.1 System Interfaces 6](#_Toc469504822)

[2.1.2 User Interfaces 6](#_Toc469504823)

[2.1.3 Hardware Interfaces 12](#_Toc469504824)

[2.1.4 Software Interfaces 12](#_Toc469504825)

[2.1.5 Communication Interfaces 12](#_Toc469504826)

[2.1.6 Memory Constraints 12](#_Toc469504827)

[2.1.7 Operations 12](#_Toc469504828)

[2.1.8 Site Dependencies 12](#_Toc469504829)

[2.2 Product functions 12](#_Toc469504830)

[2.2.1 Manager 12](#_Toc469504831)

[2.2.2 Employee 12](#_Toc469504832)

[2.2.3 Function Descriptions 13](#_Toc469504833)

[2.3 User Characteristics 18](#_Toc469504834)

[2.4 Constraints 18](#_Toc469504835)

[2.5 Assumptions and Dependencies 18](#_Toc469504836)

[2.6 Apportioning of Requirements 18](#_Toc469504837)

[2.7 Software System Attributes 18](#_Toc469504838)

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of the Lily Scheduling System. It will explain the purpose and features of the system including: the interfaces of the system, what the system will do, and the constraints under which the system must operate. This document is intended for both the stakeholders and the developers of the system and will be a guide for creation of the software.

## 1.2 Scope

This software will be an online and mobile scheduling application known as Lily. This system will allow scheduling managers to easily create and share their schedule with their employees. The system will also allow employees to easily check their schedule and make shift request.

More specifically, this system is designed to allow the schedule manager to easily manage their schedule by allowing them approve or deny shift requests with the click of a button. This allows the employees to negotiate shift requests amongst each other and send them to the manager for approval once both employees accept.

## 1.3 Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Schedule Manager | The person(s) who create and maintain the schedule |
| Employee | The person(s) who use the scheduling system |
| LSS | The LSS is the Lily System’s Scheduler acronym that is used throughout the document. |

## 1.4 References

[1]IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

[2]Clarke, S. (2013). SET Guidelines for Documenting Requirements in an Agile way. Retrieved December 12, 2016, from <https://conestoga.desire2learn.com/content/enforced/62087-D2LGroup-0059_P00_X_X_D2LGroup-0059_1_X/Standards/Agile/SET-Agile-SRS-Guideline.pdf>

## 1.5 Overview

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

# 2. Overall description

This section will give an overview of the system's purpose and functionality.  It will also talk about the design constraints of the system and any assumptions that were made.

## 2.1 Product perspective

The Lily System's scheduler called Lily has 3 types of interfaces which include the user, system, and the software. These three interfaces will be explained and described.

### 2.1.1 System Interfaces

This system will be interacting with software like Excel, Word, and itself.

### 2.1.2 User Interfaces

The user will be utilizing the scheduler through the Lily system. There are a few features of which are available to the user. These features include the creation of schedules, employee profiles, instant work change requests, instant messaging, a review page for each employee, and a day to day schedule for the employee. The scheduler allows managers and supervisors to update and create schedules easily and efficiently. There is also a feature to print the schedules out and special templates which can be scanned for future references.

Upon entering the site or application, the user will be seeing similar to Figure 1.

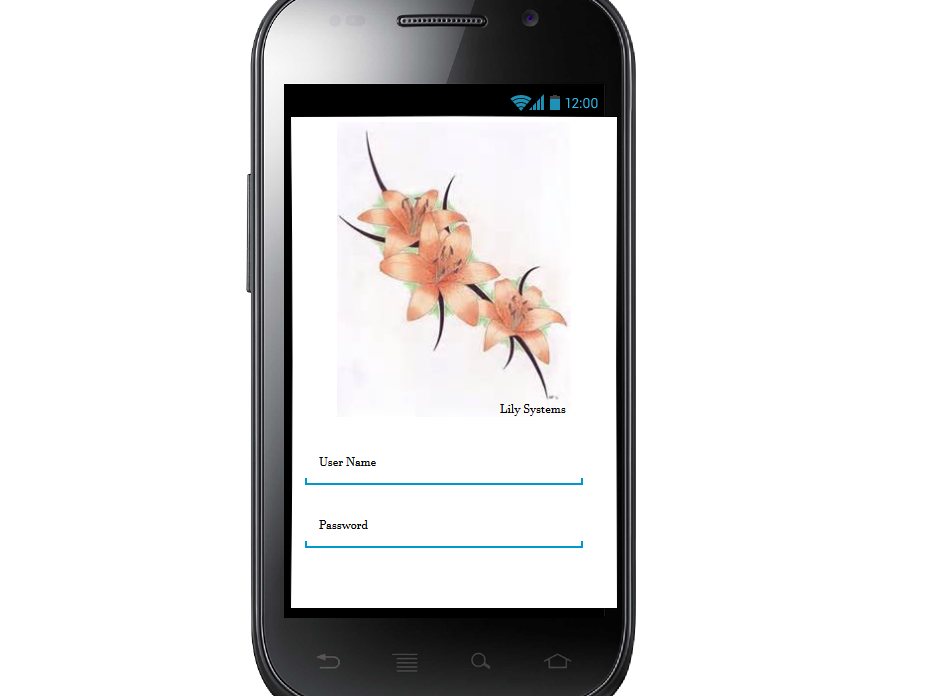


Figure 1

When the user will login they then they will see Figure 2. Here the user will be able to see their schedule, and any new notifications about any requests made.

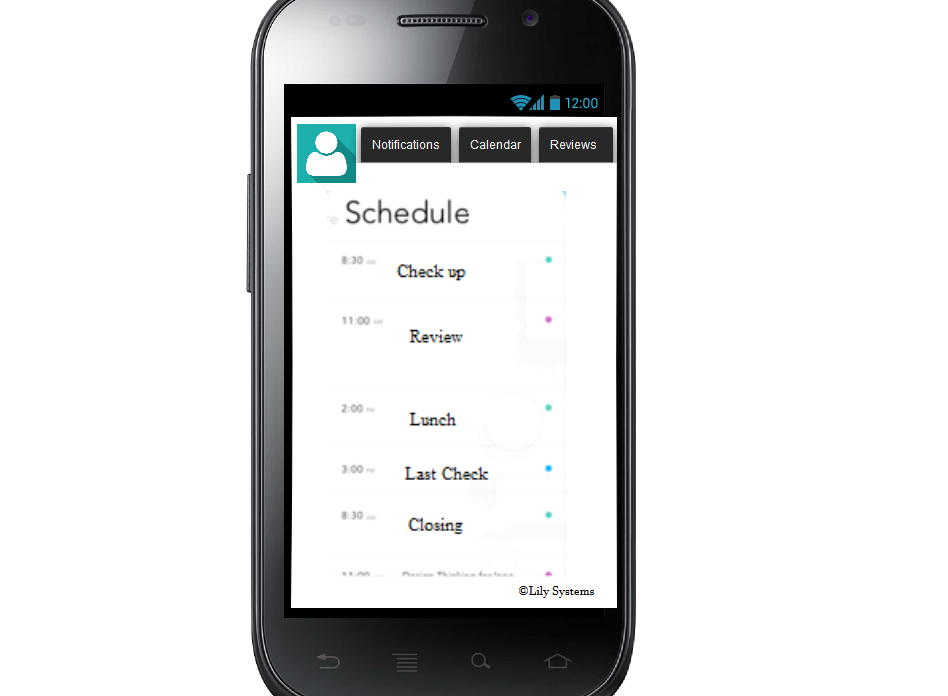


Figure 2

The user can check out their profile page which will give them an overview of their review as well.

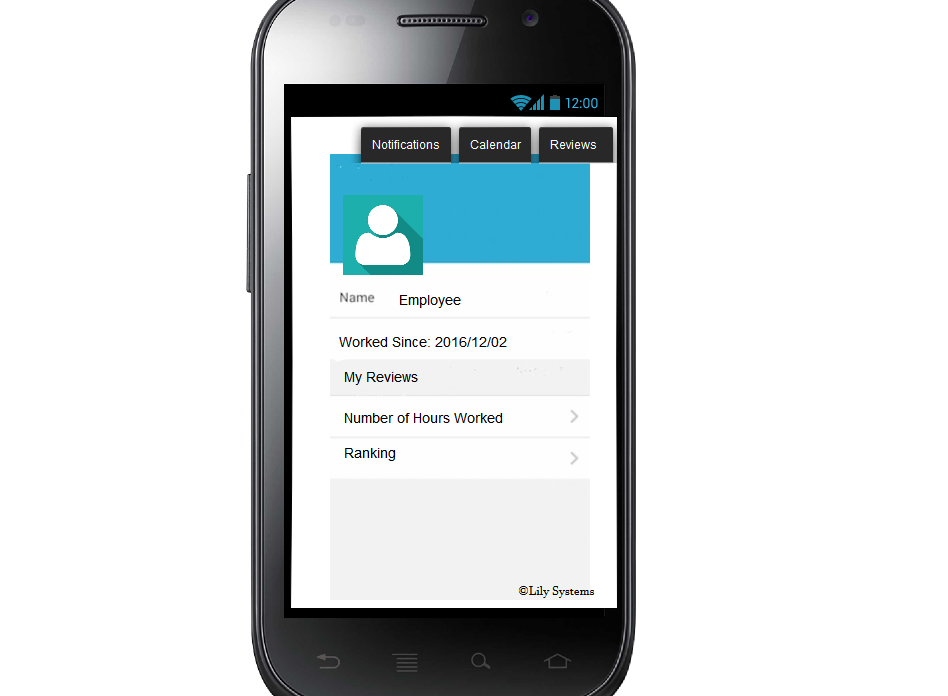


Figure 3

#### 2.1.2.1 Updating Calendar

The higher security level people will be able to update the calendar exactly the way they need it in real time, where all of their employees are able to see the changes also in real time. The interactive calendar (Refer to figure 4) allows the managers to even colour code their data appropriately.

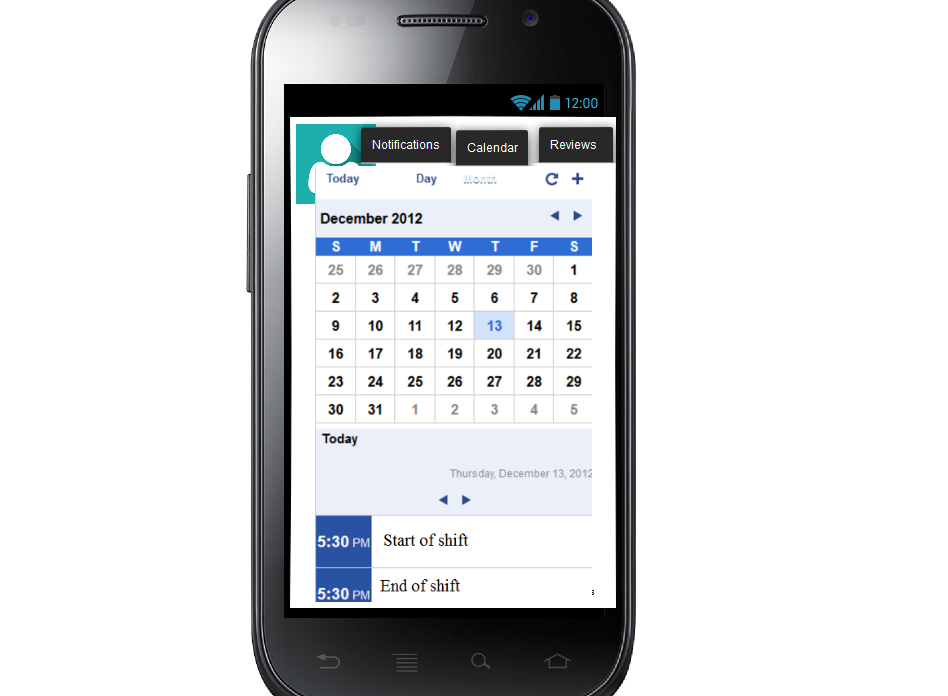


Figure 4

#### 2.1.2.3 Making a Request

The system shall allow instant shift change requests between employees and managers (see figure 5). The manager will be able to approve or reject said request appropriately. In the case the shift is approved, the system shall automatically make the changes on the calendar.

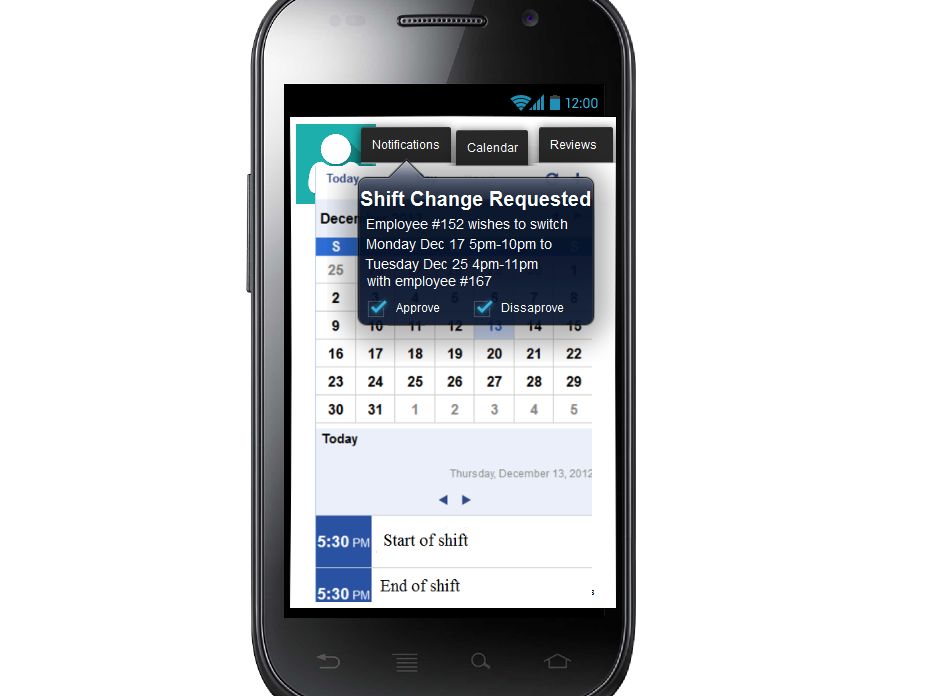


Figure 5

#### 2.1.2.4 Instant Messaging

Like the instant shift change requests, the system shall allow employees and managers to instant message each other in the cases that there is information that needs to be relayed. The system shall also record all of the conversations and store them for security reasons.

#### 2.1.2.5 Employee Reviews

The system shall allow for employee reviews that will be conducted by the managers. Employees and managers are all able to create a review for an employee of the establishment, and each review will be monitored for any indecency by the system. Managers are also able to approve and disapprove any reviews made by other employees in case of any missed inappropriate reviews. The system will also calculate the star amount that an employee therefore will gain, according to their reviews.

#### 2.1.2.6 Scanning Calendars

The system shall allow the manager to scan specially made scanning paper for the LSS. Upon scanning the sheets, the system will transfer all of the information into the online calendar. The manager is able to ask the system to also transfer the information into an Excel spreadsheet.

### 2.1.3 Hardware Interfaces

The LSS does not require any specially made hardware interfaces. The LSS can be used on a mobile device or on a computer that supports current browsers.

### 2.1.4 Software Interfaces

The LSS as of current does not require any API’s or other software interfaces. The LSS can be used on current browsers and application

### 2.1.5 Communication Interfaces

There are no special communication protocols used in the LSS.

### 2.1.6 Memory Constraints

The memory constraints cannot be fully documented at this time.

### 2.1.7 Operations

For general operations see section 2.2.

### 2.1.8 Site Dependencies

The LSS does not require any on site dependencies. It can be utilized on any device that can run current browsers, or on a mobile device with the mobile application.

## 2.2 Product functions

The LSS will have the following options for the user to interact with: creating/updating/deleting calendars, accepting/declining shift changes, creating new employees, updating/deleting existing employees, create/update/delete/review reviews of employees, notifications, instant messaging, profiles, and finally any calendar scanning or uploading from separate files.

### 2.2.1 Manager

The manager will have access to the following options: creating/updating/deleting calendars, accepting/declining shift changes, creating new employees, updating/deleting existing employees, create/update/delete/review reviews of employees, notifications, instant messaging, profiles, and finally any calendar scanning or uploading from separate files.

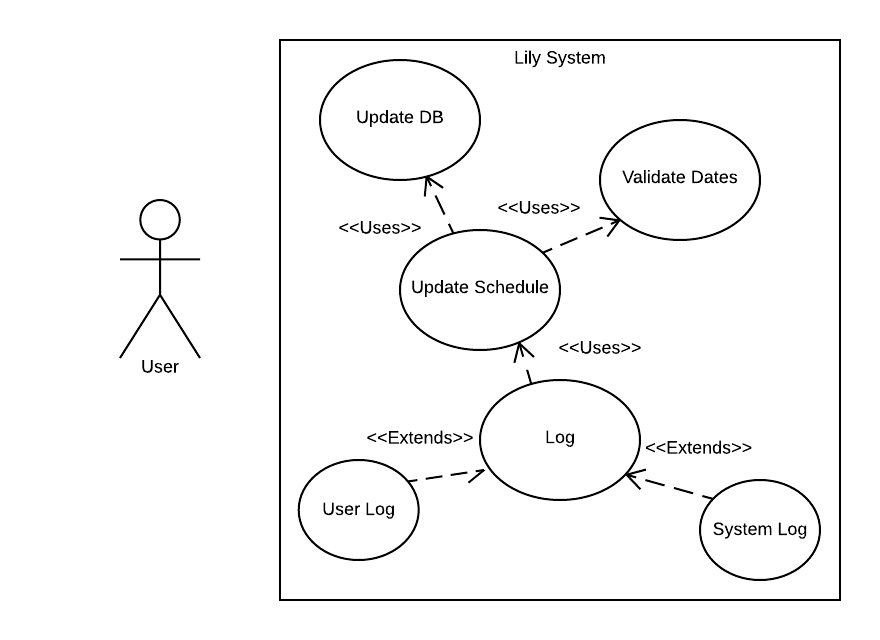
### 2.2.2 Employee

The options that the employee has are more limited, they are able to view calendars, request shift changes, create/update/delete/review reviews of other employees, notifications, instant messaging, and modify their profiles.

### 2.2.3 Function Descriptions

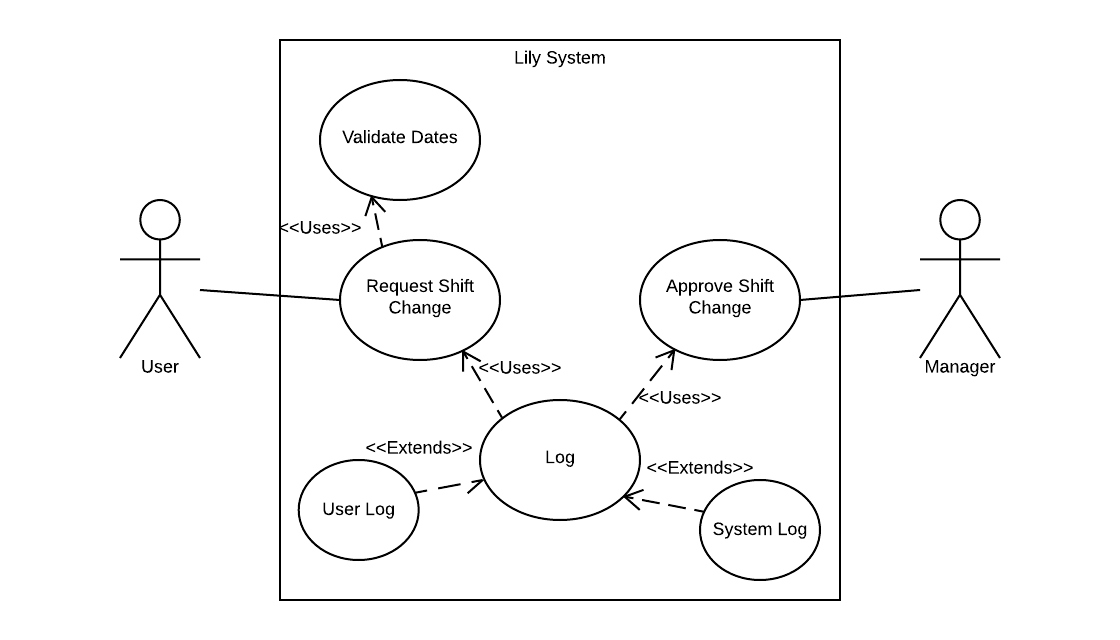
The functions below described, are creating/updating/uploading of a schedule, the accepting/declining of shift changes, creation of employees and their profiles.

#### 2.2.3.1Updating Calendars



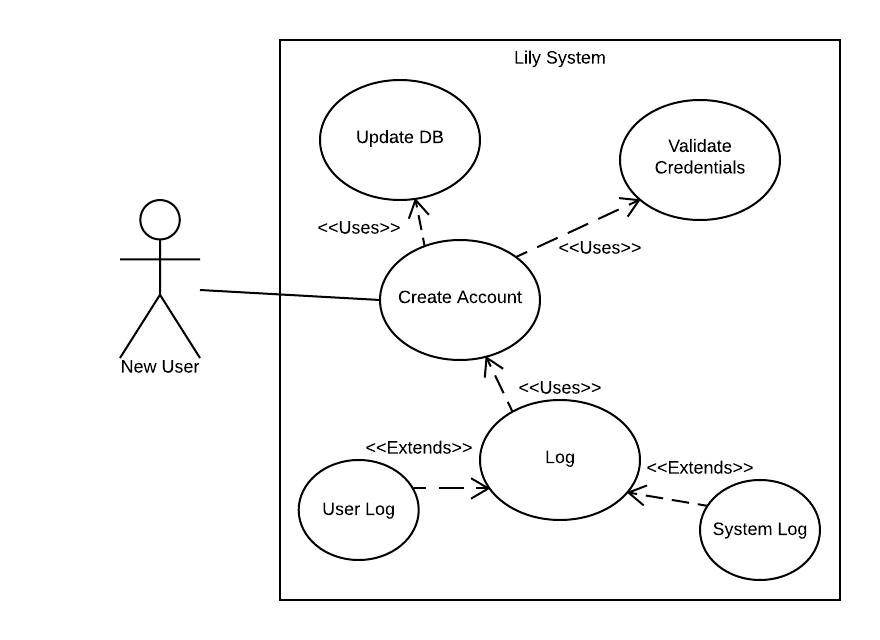
|  |
| --- |
| Flow of activities for scenario of User updating a schedule |
| Main Flow:  1.       User adds events to a schedule  2.       System validates the information added ie. Dates are realistic and time frames make sense  3.       User saves the updated schedule  4.       System logs the actions |
| Exception Conditions:  2.   The user is presented with an error message describing the faults with their input  3.   If the schedule was not successfully saved in the database the user will be notified |

#### 2.2.3.2 Accepting/Declining Shift Changes



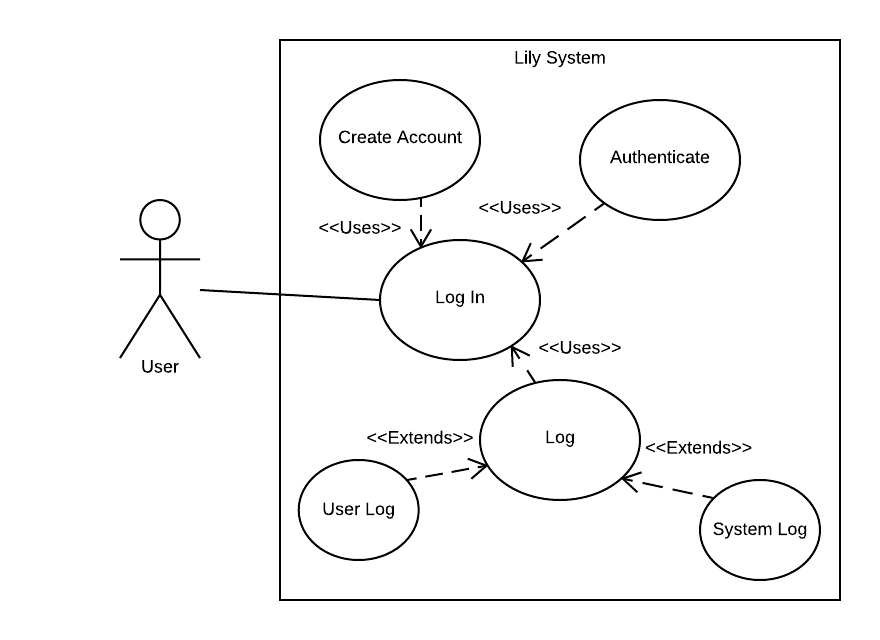
|  |
| --- |
| Flow of activities for scenario of User requesting shift change and Management reviewing request |
| Main Flow:  1.       User requests a shift change with the desired dates, times and reasons included  2.       System validates user's input  3.       Request is sent to management for approval  4.       Management chooses whether to approve or decline request  5.       System logs the actions |
| Exception Conditions:  2.   User is presented will any incorrect input they entered  3.   User is notified if their request failed to send |

#### 2.2.3.3 Creating New Employees



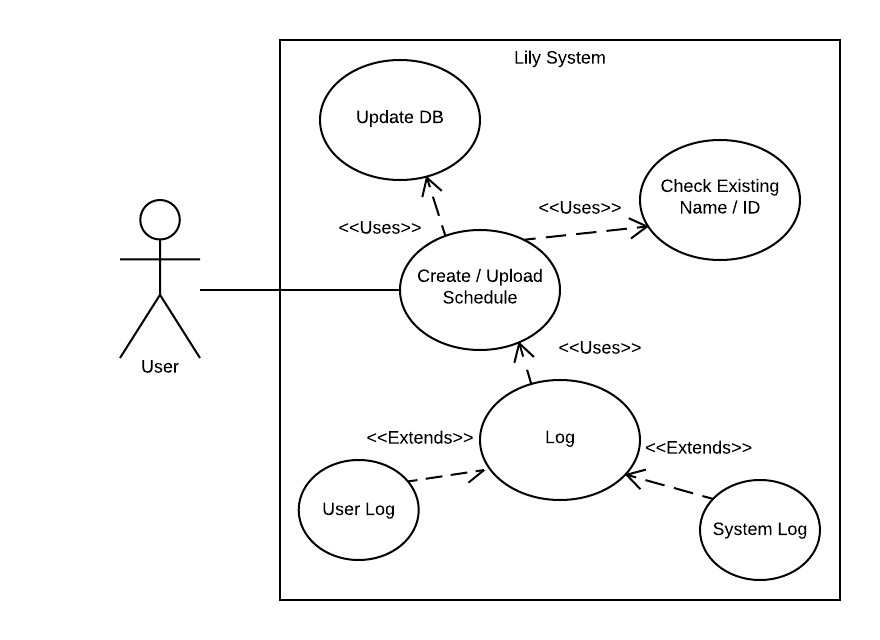
|  |
| --- |
| Flow of activities for scenario of New User creating an account |
| Main Flow:  1.       New user enters information into the system  2.       System validates the information from the user  3.       New user account is created and information is added to the database  4.       System logs the actions |
| Exception Conditions:  2. The user is presented with any incorrect information in their sign up form  3. User is presented with an error message detailing the reason why the account could not be          created and saved to the database |

#### 2.2.3.4 Profiles



|  |
| --- |
| Flow of activities for scenario of User log in |
| Main Flow:    1.       If this is a new user, then user will create a new account    1a. If existing user then they enter in their credentials  2.       The system will authenticate their credentials  3.       User’s credentials are correct and they are granted access to the system’s features  4.       System logs the actions |
| Exception Conditions:  2. If the user’s login credentials to not match any database entries then they are not granted          access to the system and will be presented with a message indicating the issue. |

#### 2.2.3.5 Calendar Creation/Upload



|  |
| --- |
| Flow of activities for scenario of User creating / uploading a new schedule |
| Main Flow:  1.       If the user creates a new schedule they will be prompted to name it    1a. If the user uploads a schedule they will select an appropriate file type from their computer          Computer and will need to name it (default name will be filename)  2.       The system will add the schedule to the database  3.       System logs the actions |
| Exception Conditions:  2.   If the schedule name already exists there is a conflict and the user will be prompted to change         the schedule name or cancel the creation of the schedule |

## 2.3 User Characteristics

The Lily scheduling system is designed to be easy to learn and use for a variety of users. The Employee users are expected to be Internet literate and be able to use basic site navigation and tools. The Schedule manager is expected to be computer/mobile literate and to be able to navigate the site and use the scheduling tools

## 2.4 Constraints

The Lily scheduling application will not be able to import a schedule from another scheduling app. The scheduling app will also not be supported on Windows Phone, older versions of Android.

## 2.5 Assumptions and Dependencies

Our application will need to rely on a web hosting service in order to remain active and online. Users will need to have a computer or mobile device in order to use our application.

## 2.6 Apportioning of Requirements

For the features that will be included in the LSS, a few of them may need to be “put off” to another release version. These features are: one of the mobile versions of the application may be transferred to another release (whether it be the android or apple is undecided); the special scanning paper feature for the LSS will be done in a future release; the review page feature will be considered for future release; instant messaging will be another feature that will be postponed in the event that there is a time constraint.

## 2.7 Software System Attributes

The requirements in this section specify the reliability, availability, security, maintainability, and portability of the of the software system. These are non-functional requirements. The Lily scheduling application should reliably be accessible online. The Lily scheduling application should sync scheduling information between the browser and mobile versions reliably and almost instantly. The system shall keep all user information secure for the users only sharing user information with the appropriate users. The system shall be able to be repaired if broken and upgraded with new features in the future. The system should be portable enough to allow for ports to potentially be made to new operating systems as demands change.