**Software Requirements**

**Specification for**

**Learnr**

**Version 1.0**

**Prepared by**

**Tanzila Ahmed**

**Nelson Batista**

**Samuel Cohen**

**Tan Paul**

**CCNY**

**May 2016**

**Table of Contents**

Introduction .........................................................................................................................1

Purpose .............................................................................................................................. 1.1

Scope ................................................................................................................................. 1.2

Definitions, Acronyms, and Abbreviation ............................................................................1.3

References ......................................................................................................................... 1.4

Overview ............................................................................................................................ 1.5

General Description ........................................................................................................... 2

Use Case Model Survey .................................................................................................... 2.1

Assumptions and Dependencies ....................................................................................... 2.2

Specific Requirements ...................................................................................................... 3

Use Case Reports ............................................................................................................ 3.1

Non­Functional Requirements .......................................................................................... 3.2

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to provide an overview with the purpose, scope, references, case model survey, assumptions, dependencies, case reports, and nonfunctional requirements of the Software Requirements Specification for the Learnr application. The document explains the functional features of Learnr including interface details, design constraints, and performance characteristics.

**1.2 Scope**

<​Learnr​>

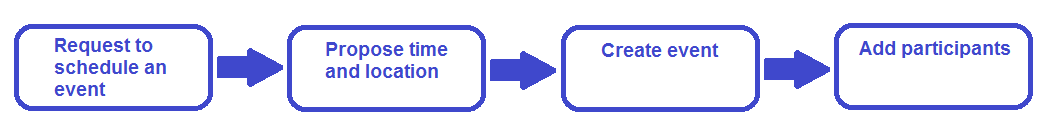
The Learnr application is a scheduling tool that can be used quickly and easily to find time and place to meet with people who are interested in studying in a group. The Learnr application enables college students an easier way to organize, administer and manage all student, course, and meeting related information and outcomes. The Learnr is an integrated application system of modules and functions that provide the most convenient method for users. Furthermore, all the data is saved using SQL database format which allows to store and retrieve large amounts of records quickly and efficiently.

**1.3 References**

(1) Schach, Stephen R. (2011). “Object­ Oriented and Classical Software Engineering,” 8th Edition. The McGraw­Hill Companies.

**1.4 Overview**

The Learnr will be very simple with only four steps to create a study group. The four steps for the process are:



**1.      Request to Schedule an Event**

An event can be scheduled by a participant or coordinator by filling out the title of the event, a description of the course and materials that will be covered, and the location of the meeting.

**2.      Propose Time and Location**

Since college students have classes and other commitments, it’s difficult to coordinate time to study together, hence, the event creator can give a time and location.

**3.      Create Event**

System admins will then process the request for the event, assign a coordinator for the event that is qualified to help students for the given subject, and put forth the prerequisites needed for participants to join. The event will then be made public so participants can search for the events related to their subject of choice and request to join, given that they satisfy the prerequisite for the subject.

**4.      Add Participants**

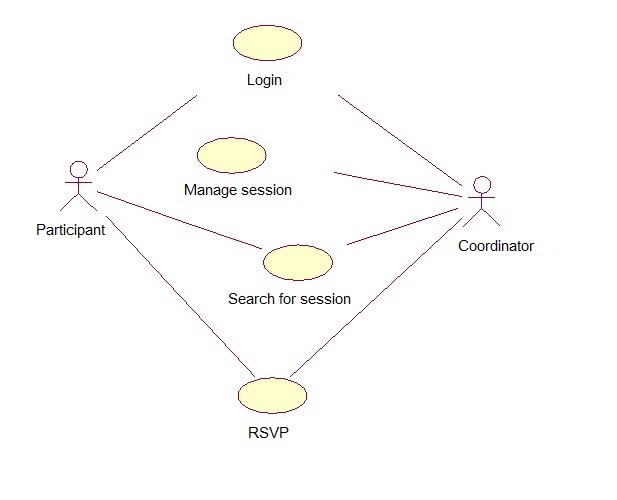
System admins will then accept their request. If the number of participants exceeds the maximum, then system admins can increase the limit and assign more coordinators to manage the group.

Learnr is intended to provide a means for entering the necessary information for class group study scheduling, searching for the sessions, and requesting to join sessions. This information of the upcoming sessions, with their descriptions, times, and the locations will be stored in a database. The user profile information, such as academic history and contact information, will be stored in another database. The user will be able to search for the session on the web interface based on subject, time, and location, which will return the upcoming sessions based on the user selected parameters, and creating web pages with scheduling details about the session.

**2. Use-Case Diagram & Description**

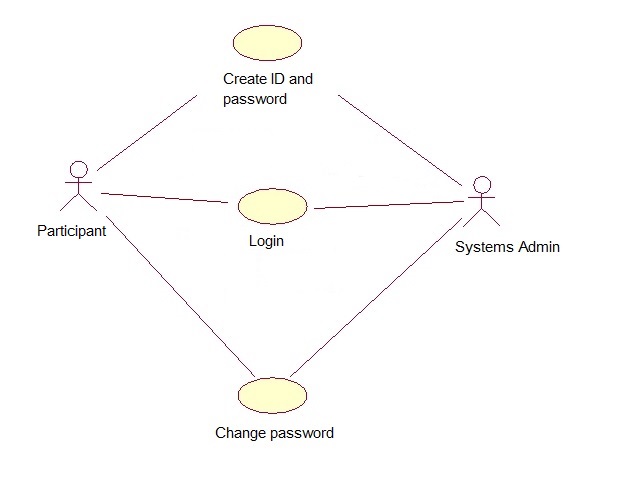
Use case diagrams represent the functional interactions of a system. The figures represent the users, which are external to the system and interact with the system through interfaces. The users of the Learnr application are the participant, coordinator, and the systems admin. The participant uses the system to search for study sessions that are created by systems admin. The ovals represent the individual use cases that the system performs to provide the services that the users desires. The primary use cases of the Learnr application system are: Search for Sessions, Registration and Login, and Manage Sessions. The participant interacts with the following use cases: Registration and Login, Search for Sessions, and RSVP. The systems admin interacts with the following use cases: Create Session, Manage Session, Assign Coordinator, and manage RSVP.

Use Case Diagram of System Analysis:

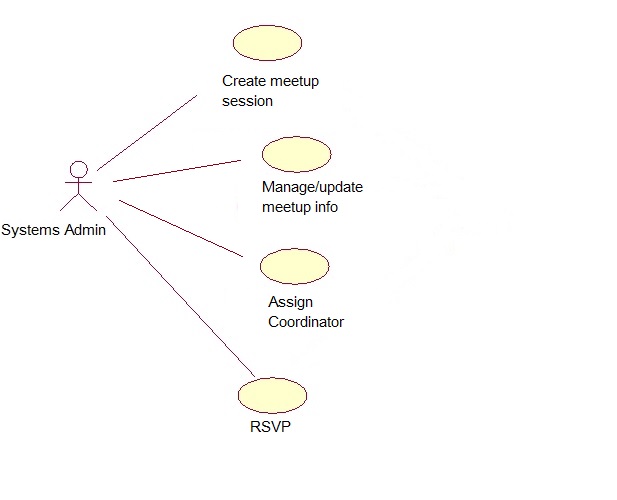


This use case diagram above shows the general function of the system. Some of the high level use cases are shown below by decomposing them into their own use case diagrams (e.g. Registration and Login, which decomposes to lower-level use cases: Create ID and Password, Login, Change Password).

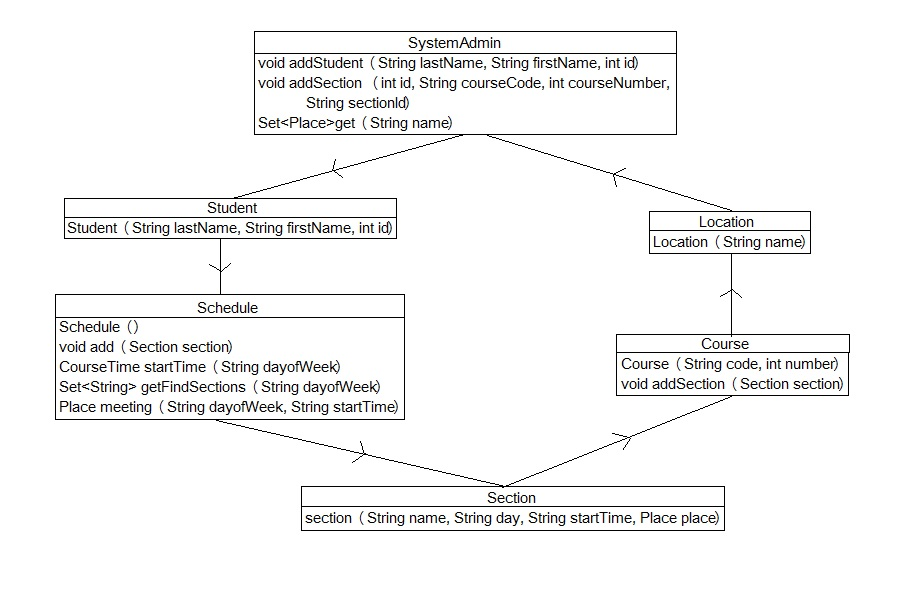
Use Case Diagram for Registration and Login:



Use Case Diagram for Management:



The following UML diagram shows the system process and partial code:



**3. Data Flow Diagram**

The three essential components of the Learnr application are:

*The database:*

To store and look up study sessions, there must be a database that contains all of the

information about a study session. This should be implemented as a SQL database with

attributes for each session such as time, class, coordinator, and any additional notes about a

session.

*The web interface:*

Our database provides an interface for the users to look at study sessions. We have

decided to implement this as a web application, using web framework Flask and set up a basic web server with different pages using Python, HTML, and CSS.

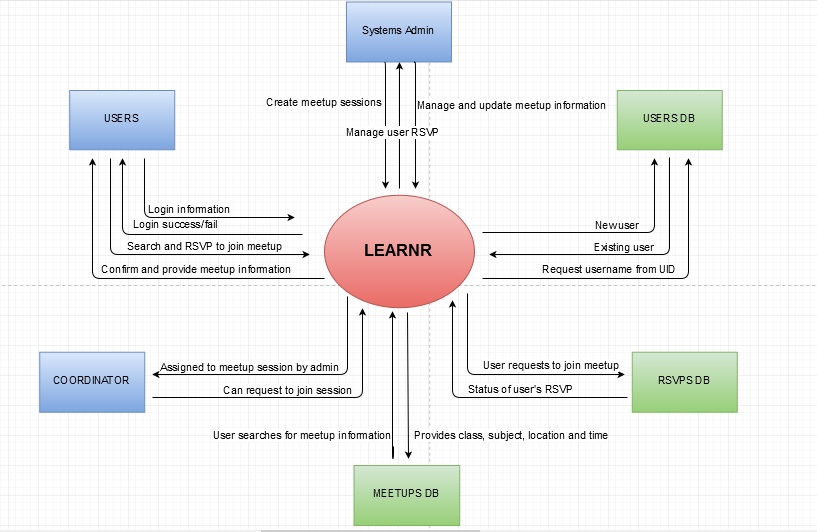
*User Profile:*

For all users, we require a user profile. This holds information such as major, year, email, a messaging service, as well as a unique id. As with many of our data, this can be implemented in the form of a

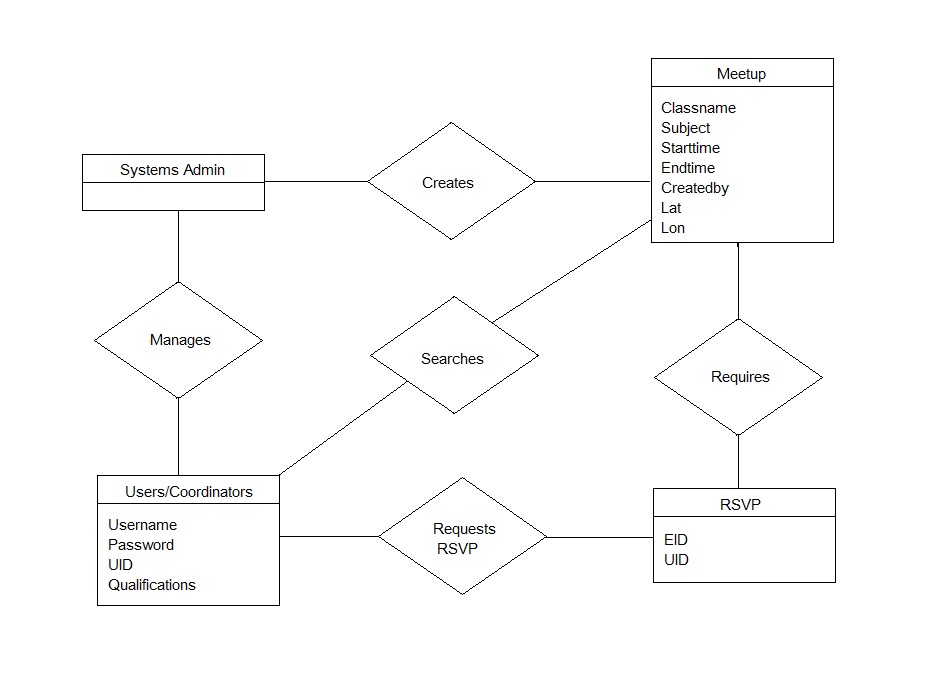
database of users. The web interface can then provide a way to view user details.

The Data Flow Diagram below shows the processes that each user will face, and it also shows the databases where all the data will be stored and retrieved as per needed.

Data Flow Diagram of Learnr:



**4. ER Diagram**



The ER Diagram above describes the relationship that each component shares with other components. The ER model is composed of entity types (such as Systems Admin, Users/Coordinators, Meetup, and RSVP) and specifies relationships that exists between those entity types.

**5. Pseudocode for Use-Cases**

**6. GUI Snapshot**