

Lewis' take on PSD3 requirements

Session object - date/time, room no, session capacity, tutor/lecturer, students attending, name, optional, 1 of N, course, duration

Login possibilities - tutor / admin / student / lecturer

Options for different users

Tutor - availability, no of sessions

Student - sign up

Admin - set up sessions, authorize session changes?

Lecturer - request sessions

Basic Use Cases:

Tutor > availability input > max no of sessions, times available, subjects can teach

Lecturer > request sessions > name, course, optional, tutor/lecturer, duration, frequency, equipment needed?

Questions

1. General system questions

- **Local computer system** - What is meant by “local computer system”. Is the system going to be deployed only on the computers inside the cs department. Is it going to be used only on a single machine (it doesn’t make any sense)
- **Future of the system** - do you expect any upgrades of the system once is finished. Is the system going to be related with other systems in the future.

2. General Questions

- How you define a time slot? Do all the slots start at say 14:05 and finish at 14:55?
- The attendance of the students has to be checked? What is the procedure?

3. Users

- **Undergraduate students**
 - What if there are no suitable slots that the student can attend?
 - What if a student wants to swap his lab slot? What is the procedure?
- **Lecturers**
 - Will the lecturer submit a file with the number of required sessions for a given course?
- **Teaching administrators**
 - Again is the admin going to be prompt with a series of questions when creating a new session slot, or he will submit a file?
 - How is the teaching administrator going to be notified that a session needs to be set up.
- **Tutors**
 - To indicate available slots are the tutors submitting a file (in some format), or the system will supply the lab sessions for the tutor and then he/she can choose available slots from the list of sessions?
 - What is the cardinality of the relationship session<->tutor
 - Once the tutors indicate their available slots, how are they allocated to the lab sessions? What is the max number of tutors for a given lab session? Are the tutors assigned to labs only or they can be assigned to example class as well?
- What details should we keep for each role?
- How the users will log in to the system. Are they going to use their lab user/pass or they should register exclusively for this system
- Is the system going to have some sort of admins that can log in as student/tutor/lecturer/teaching admin?

4. Initial Requirements

- **Definition of teaching session**
 - Course/Session type/Timestamp/Venue/Max students
 - Who are the tutors, if any? Who is the lecturer?
- **Manual timetabling of these sessions**
 - Is the system going to prompt a sequence of questions in order the user to fill in session information or the user should fill in a text/excel/xml file with the session information?
- **Attendance monitoring at sessions**
 - Is the attendance going to be monitored via beeper? If so who is responsible to submit the information from the beeper to the system?
 - What if the student is sick, is this case related to the system?
- **Bulk export of some/all data as CSV files.**
 - What information could be exported in CSV file? Attendance? Sessions for a given course? Tutor<-->labs relationship?

5. Additional Requirements

Requirements specification

- **General use cases**
 - Logging in using MyCampus credentials
 - Logging out “-exit” command
 - Should have responsive command prompt with some generic functions for all users
 - Should have “-help” command
- **Undergraduate students use cases**
 - Should be able to see list of courses
 - Should be able to sign up for a session associated with a course
 - If the student doesn't have a suitable slot - should be able to contact teaching administrator - sending a simple message.
 - Should be able to request a swap to another session
 - If there is any changes(like changing the venue for a lecture) the students need to get notification by email.
 - should be able to see to which session has been allocated to
- **Lecturers use cases**
 - Can set up a session
 - Should choose between submitting a CSV file or fill in the required

information from the command line

- Can supply a file with the following information using a CSV file:
 1. Session type (lab/example class/tutorial)
 2. Course name
 3. Lecturer name
 4. Number of tutors required
 5. Duration
 6. Max/Min students
- Can follow the command line instructions to fill in the required information
- Should be able to edit or delete session information

- **Teaching administrators use cases**

- Should be notified when a lecturer request a new session to be arranged
- Should receive session information from the lecturer
- Pass the session information to another system (out of the project scope), which is responsible to set up timetable slots and to book rooms
- Should be able to review messages from students requesting swap of sessions or notifying for slot clashes.

- **Tutors use cases**

- should be able to submit CSV files with available times
- should be able to state specializations
- should be able to state min/max required or preferred number of hours
- should be able to send message requesting session swaps
-

- **System admin use cases**

-

Title	Show list of courses
Description	<p>Normal process flow: A student is successfully logged into the system. When typing “show courses” a list of available courses should be supplied. Each course should be associated with an id so that the user can select the course from the command line.</p> <p>Error handling: If there are no available courses for this student appropriate message should be shown.</p>

Title	Sign up for session
Description	<p>Normal process flow: When the student has selected a given course he should be supplied with the sessions associated with this course. Each session should have the following information:</p> <p>Session ID Session type (lab/example class/tutorial) Course name Lecturer name Number of tutors if any Duration Max/Min students, Number of students enrolled for the session</p> <p>Then the student should be able to sign up for a session using the following command “sign up session #num”. If signed up successfully appropriate message should be shown. If the student doesn’t have suitable sessions he/she should be able to contact the teaching administrator (See ticket “contact teaching administrator”).</p> <p>Error handling: If the student tries to sign up for session that is already full - appropriate message should be supplied.</p>

Title	Contact teaching administrator
Description	<p>Normal process flow: All user should be able to send messages to the teaching administrator. Once a user is successfully logged into the system he should be able to write the following command - "contact teaching admin". Then the user will be asked to write a message. To terminate the input the user should type "CTRL + D". Then the user will be asked if he wants to send the message or to discard it. In both cases appropriate messages should be shown.</p> <p>Error handling: If for some reason the user is unable to send the message, either because there is no connection to the database or something else - appropriate error message should be shown.</p>

Title	Teaching admin - setting up session
Description	<p>Normal process flow: Teaching administrators should receive notifications from lecturers regarding new sessions that should be set up. A notification should include CSV file with the session information. Then the teaching administrator should type "set up session #id" so that the CSV file could be passed to the system actually responsible for setting up time slots for the session and booking rooms.</p> <p>Error handling: Any error returned by the outer system should be reported to the user. the session already exists - the information for that session should be overwritten.</p>

Title	Request a swap of sessions
Description	<p>Normal process flow: Once a student or a tutor is successfully logged into the system he/she should be able to request a swap of session. This can be done via sending a message to the teaching administrator. The user should type “swap session”, then he/she will be asked to write a message to request the swap. Once the user is finished writing the message, he/she should use “CTRL+D” in order to terminate the input. Then the user will be asked whether to send the message or to discard it.</p> <p>Error handling: If for some reason the user is unable to send the message - appropriate error should be shown.</p>

Title	Setting up sessions
Description	<p>Normal process flow: Successfully logged in lecturers should be able to set up new sessions. Each session should have the following required fields.</p> <ol style="list-style-type: none"> 1. Session type (lab/example class/tutorial) 2. Course name 3. Lecturer name 4. Number of tutors required 5. Duration 6. Max/Min students <p>Typing the command “set up session” the lecturer will be asked whether the session will be set up manually through the command line, or via CSV file.</p> <ol style="list-style-type: none"> 1. Setting up session manually - The user will be prompted with a sequence of questions in order to fill the session information. The obtained session information then will be saved as a CSV file. 2. Setting up session via CSV file - The user will be supplied with a CSV file template, which can be found in the file system (email probably?). Then the user will be asked to state the path to this CSV file, once the file is ready to submit. <p>Once the lecturer is finished with filling the information for the session, he/she should be able to notify the teaching administrator. Typing the</p>

	<p>following command <i>“notify TA”</i> followed by the path to the CSV file that contains the information about this session.</p> <p>Error handling:</p> <ol style="list-style-type: none"> 1. If the user has supplied invalid file the system should return appropriate error message 2. If the user has typed incorrect information in the CSV file or during the sequence of questions - again appropriate error should be shown, as well as the request for setting up the session should be rejected.
--	--

Title	Edition and deletion of sessions
Description	<p>Normal process flow:</p> <ol style="list-style-type: none"> 1. Edition The lecturer should be able to see the list of sessions added to the system using the command <i>“list sessions”</i>. Then the user should be able to select, which session he wants to edit. The system then will suggest two ways of editing <ol style="list-style-type: none"> 1. Edit via command line - the system will iterate through the session fields and suggest the user to edit each of them. For each field the user should select whether to edit or go to the next field. When editing a field the user should be able to save or discard changes. 2. Edit the CSV file - The user should be supplied with a CSV file. Once finished editing the file, the user should be able to save changes specifying the path of edited CSV file. 2. Deletion Once the user has listed the session that have been added to the system, he/she should be able to select which one he/she wants to delete using the command <i>“delete session #id”</i>. <p>Error handling: The system should ignore changes that are violating the session files types. If the user tries to submit file with incorrect format an error should be shown.</p>

Title	Stating specialization and stating min/max required or preferred number of hours
Description	<p>Normal process flow: The tutors should be able to state their specialization and preferred number of hours. Typing the command “set up number of hours” or “set up specialization” the tutor will be asked whether the specialization and the number of hours will be set up manually through the command line, or via CSV file.</p> <p>Setting up spec/hours manually - The user will be prompt with a sequence of questions in order to fill the spec/hours information. The obtained spec/hours information then will be saved as a CSV file.</p> <p>Setting up spec/hours via CSV file - The user will be supplied with a CSV file template, which can be found in the file system (email probably?). Then the user will be asked to state the path to this CSV file, once the file is ready to submit.</p> <p>After writing the specialization and the number of hours, the tutor should be able to set up them. This will be done by typing “CTRL + D” in order to terminate the input.</p> <p>Error handling: If for some reason the tutor is unable to set up the specialization and number of hours (or unable to upload the CSV file) or he type an incorrect text, then an appropriate error should be shown.</p>

Title	Teaching admin - reviewing messages from students requesting swap of sessions or notifying for slot clashes.
Description	<p>Normal process flow: Teaching administrators should receive notifications from students regarding swaps or sessions or slot clashes. A notification should include a message with the information - swap a session (or slot clash) with a reason why the student needs to do this operation . Then the</p>

	<p>teaching administrator should be able to change the session of the student.</p> <p>Error handling: An error message should appear if there is problem with editing a session (see “<i>edition and deletion of session</i>” for more information).</p>
--	---

Title	Teaching admin - any changes(like changing the venue for a lecture) the students and tutors (or lecturers) need to get notification by email.
Description	<p>Normal process flow: Teaching administrators should receive notifications regarding changing and editing a session. The teaching administrator should type “<i>edit session #id</i>” so that the session could be changed. then the teaching administrator need to type “<i>send notification #id</i>” in order to send a notification about the changes. This will send an email to the students and tutors notifying them about the changes.</p> <p>Error handling: An error message should appear if there is problem with editing a session or sending an email (see “<i>edition and deletion of session</i>” for more information).</p>

Title	Help command
Description	<p>Normal process flow: Whenever a user is logged in the system when typing “<i>help</i>” a list of available commands, as well as description for them, should be supplied to the user.</p> <p>Error handling: No error are expected here.</p>

Title	Students and tutors - check session allocation
Description	<p>Normal process flow:</p> <p>Student and tutors should be able to see a list of sessions they have</p>

	<p>been allocated to. The command for that is “<i>show my sessions</i>”.</p> <p>Error handling: If the user is not allocated to any session - a message should be shown</p>
--	---

Title	System command prompt
Description	<p>Normal process flow:</p> <p>When a user is successfully logged in the system there should be a command prompt that the user will be communicating with. The command prompt should be expecting new commands from the user until the user actually closes the program or type “<i>exit</i>”.</p> <p>Error handling: Whenever the user tries to enter invalid commands - an error message should be shown, suggesting to type the “<i>help</i>” command in order to see the list of legal commands.</p>

key words: ta (teaching administrator), students (undergraduate students), tutors, lecturers,