Software Requirements Specification

for

AVAILSYS

Prepared by

- IIT2019204
MITTA LEKHANA REDDY
- IIT2019208
DHANUSH VASA
- IIT2019234
PRAVALLIKA KODI
- IIB2019030
KANDAGATLA MEGHANA SANTHOSHI

IIIT Allahabad

30-04-2021

Table of Contents

Introduction **1**
- **Purpose** **1**
- **Document Conventions.** **1**
- **Intended Audience** **1**
- **Product Scope** **2**
- **References** **2**

Overall Description **2**
- **Product Perspective** **2**
- **Product Functions** **3**
- **User Characteristics** **4**
- **Operating Environment** **5**
- **Design and Implementation Constraints** **5**
- **User Documentation** **5**
- **Assumptions and Dependencies** **5**

External Interface Requirements **5**
- **User Interfaces** **5**
- **Hardware Interfaces** **6**
- **Software Interfaces** **6**
- **Communications Interfaces** **7**

System Features **7**
- **4.1 System features for Admin :** **8**
- **4.3 System features for Students :** **11**

Other Nonfunctional Requirements **12**
- **Performance Requirements** **12**
- **Safety Requirements** **12**
- **Security Requirements** **12**
- **Software Quality Attributes** **12**
- **Business Rules** **13**

Other Requirements **13**

1. Introduction

"AVAILSYS" is basically a web based application, which will enable Faculty of University with features related to Booking of certain labs and also resources for their respective sections. Followed by relaying that Booking Information to the faculty, respective teaching assistance and students.

1.1 Purpose

The purpose of this web based application is to allow Faculty of University Labs the ability to Book labs at a comfortable time slot and also not having the issues of overlapping lab time slots with other faculties labs.

From Student's point of view,

- Up to date notification is not always relayed from their respective Faculty after assigned a lab for them.

From Faculty

- Unsure as to if a lab has already been booked by another faculty and hence leads to hesitation and may also lead to overlapped booking of labs.
- Requesting for resources which are out of stock due to no real communication between the admin department and Faculty.
- No real means of communication to relay the exact details of lab timings to students as lab booking is not guaranteed.

Hence, this project is an attempt to overcome these drawbacks in a University Labs Department.

1.2 Document Conventions.

Main Section Titles

Font: Times Face: Bold Size: 18

Subsection Titles

Font: Times Face: Bold Size: 14

Standard Text
Font: Arial Face: None Size: 11

1.3 Intended Audience

The main audience of this project are the faculty of Universities who are looking for a efficiently organised system which allows the faculty to maintain and allocate labs for their students

with ease. Also request resources based on real time storage data on the availability of resources without any worry of any shortage.

The secondary audience of this project is are the students of the respective faculty as they would be informed about the booked lab immediately when the Faculty have assigned it to them. Other than being informed at a later date or not at all for the faulty and gives them the time to prepare.

1.4 Product Scope

The scope of the “AVAILSYS” is to create an online web based application for the users to check the availability and book the lab at their confort. This system will be an alternative to the traditional offline way of checking available labs and booking them. The scope of the sub-system which is described in this document is the user interface. The system will have two major types of users: Faculty and Admin.

This will be explained in ‘Functionality requirements’ section :

- ➢ The first ones are Faculty who check and reserve a slot and resources prior to the requirement using the application.
- ➢ The second ones are Admin who manages the data bases and maintains the application from the university.
- ➢ Students receive an Email after his/her faculty books a slot for the lab.

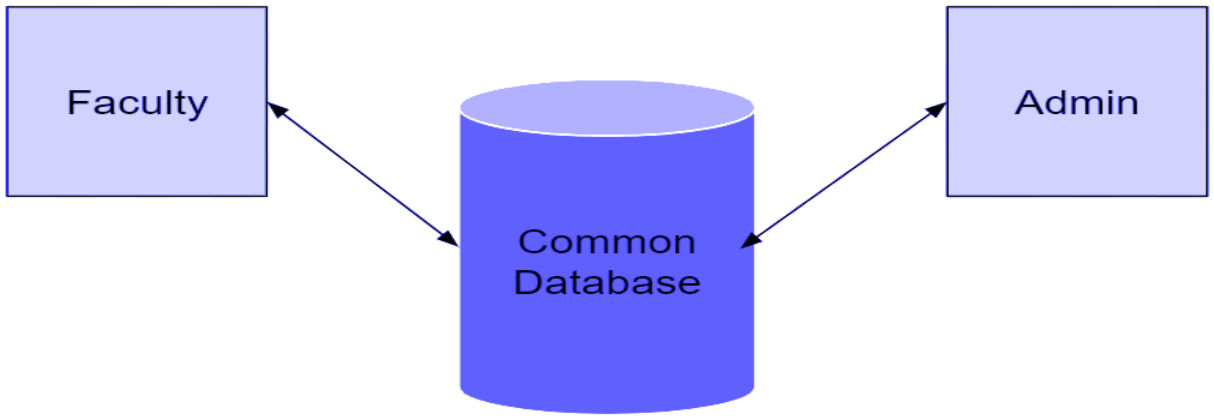
1.5 References

- *https://courses.cs.ut.ee/MTAT.03.306/2018_fall/uploads/Main/team5.pdf*
- *https://edgeryders.eu/t/list-open-source-software-for-resource-scheduling-and-booking/6629#heading--1-4*
- IEEE SRS Format

2. Overall Description

2.1 Product Perspective

“AVAILSYS” *is an independent stand-alone system.It is totally self-contained .*



The system considers faculty and Admin perspectives. It has “AVAILSYS” – where faculty are able to book labs and use system functions described in his document, while for the administrative use there is a separate environment and all relevant operations run. Both systems are based on the common database. They are integrated with each other, meaning the relevant data is exchanged. The subsystem “User Interface” is an element of “AVAILSYS”and it represents the web based application for faculty to book labs online according to their choice of selection and time of availability.

2.2 Product Functions

“AVAILSYS” supports the following use cases:

USE CASES	**DESCRIPTION OF USE CASES**
ADMIN:	
Manage Student database	Admin can add, remove, and edit any students details.
Login	Allows Admin to Login
Manage Faculty database	Admin can add, remove, and edit any Faculty details
Cancel the Slot	Allows the Admin to cancel any reserved slot
Manage Timeslot	Admin assigns the allowed time slots for - Faculty to select from.
Manage Labs	Admin assigns the available for Faculty to select from.
Manage Resources Database	Allows the Admin to update the Resources Database according to the availability of resources.
FACULTY:	
Register Faculty	Allows a faculty member to register into the application.
Manage Sections	Allows the faculty to add, remove or edit students details to their respective sections
Login	Allows Faculty to login
Schedule a Slot	Allows the Faculty to Book an available time slot for lab

Cancel the Slot	Allows the Faculty to cancel the reserved slot that they have made
Schedule Resources	Allows the Faculty to schedule resources for the lab
Receive Emails	The Faculty receives an email upon successful booking of the lab with details of when and where the lab is being held.
STUDENT:	
Receive Emails	The Student receives an email upon successful booking of the lab with details of when and where the lab is being held.

2.3 User Characteristics

There are 3 USER levels in our “AVAILSYS” lab scheduling web based application:

a. ADMIN
b. FACULTY
c. STUDENT

A. ADMIN:

Admin has the access to control what is happening in the application. Admin is solely responsible for managing the Labs availability, resources and validation of the Faculty and students. Hence, Admin can view the all the booked labs report, students assigned to these labs and analyze them and take the decision accordingly. Admin is required to have experience in managing a similar management portal and must and foremost have in depth knowledge about the faculties and students in the University.

B. FACULTY:

Faculties sole responsibility is to find an available time slot for a lab at their convenient time. The Faculty is slightly restricted compared to the Admin as the faculty cannot cancel other faculties labs insead is obliged to only deal with his/her labs allocated booking only. Faculty is also responsible for updating his/her section and the students in the section. As a result the Faculty is required to know basic knowledge of how to use a web based application.

C. STUDENT:

Students really have no use with the web based application as the only thing that the student does is wait for an email from the “AVAILSYS” application to notify him/her about ther lab where and when it is. Student requires the ability to check his email at regular intervals.

2.4 Operating Environment

The environment in which “AVAILSYS” is operated on is a web browser that is connected to a physical server which allows the user to access “AVAILSYS” from any operating system like Microsoft OS, Linux and ChromeBook OS. As long as the user interface for the system shall be compatible with any time of web browser such as Mozilla Firefox, Google Chrome, and Microsoft EDGE.

2.5 Design and Implementation Constraints

I. **Memory :** System will have only 10GB space of data server,
II. **Language Requirement :** Software must be only in English language.
III. **Budget Constraint :** As there is no budget, “AVAILSYS” is intended to be very best of the abilities and work for all basic functionalities. UI is intended to be minimalistic.
IV. **Implementation Constraints :** Needs to be a web based application or an android based application.

2.6 User Documentation

1. A video demonstration of the functionality of the Admin interface.
2. A video demonstration of the functionality of the Faculty interface.
3. A video demonstration of the functionality of the Student interface.
4. A Read_Me .txt file explaining how to set up and run “AVAILSYS” web based application.

2.7 Assumptions and Dependencies

It is assumed that the system developed will work perfectly. That's going to be developed under the Windows OS, and Apache Server with MySQL database. If incase of any difficulties, SRS should be flexible enough to change accordingly.

3. External Interface Requirements

3.1 User Interfaces

FACULTY	
PAGE NAME	**DESCRIPTION**
Registration	Faculty registers by providing details and credentials.
Login	Faculty logins into the system.

Dashboard	It provides functional features for faculty. 1. Manage Sections 2. Schedule a slot 3. Cancel a slot
Manage Sections	Faculty will be able to add sections,modify student details and remove sections.
Schedule a slot	Faculty will be able to schedule a slot.
Cancel a slot	Faculty will be able to remove a certain slot booking
ADMIN	
Login	Admin logins into the system
Dashboard	It provides functional features like faculty 1. Manage faculty 2. Manage students 3. Manage labs 4. Manage resources 5. Cancel slot
Manage faculty	Admin will be able to manage faculty details and remove faculty.
Manage students	Admin will be able to manage students details and remove students.
Manage labs	Admin will be able to allocate specific time slots for specific labs
Manage resources	Admin will be able to update real time resource data
Cancel slot	Admin will be able to remove a certain slot.

3.2 Hardware Interfaces

"AVAILSYS" will be running on browsers like Google Chrome, Microsoft Edge and Mozilla Firefox as it is a web based application.

3.3 Software Interfaces

Development End:

Javascript, HTML, CSS, Bootstrap, OS(windows) , Python Django framework

Database Server:

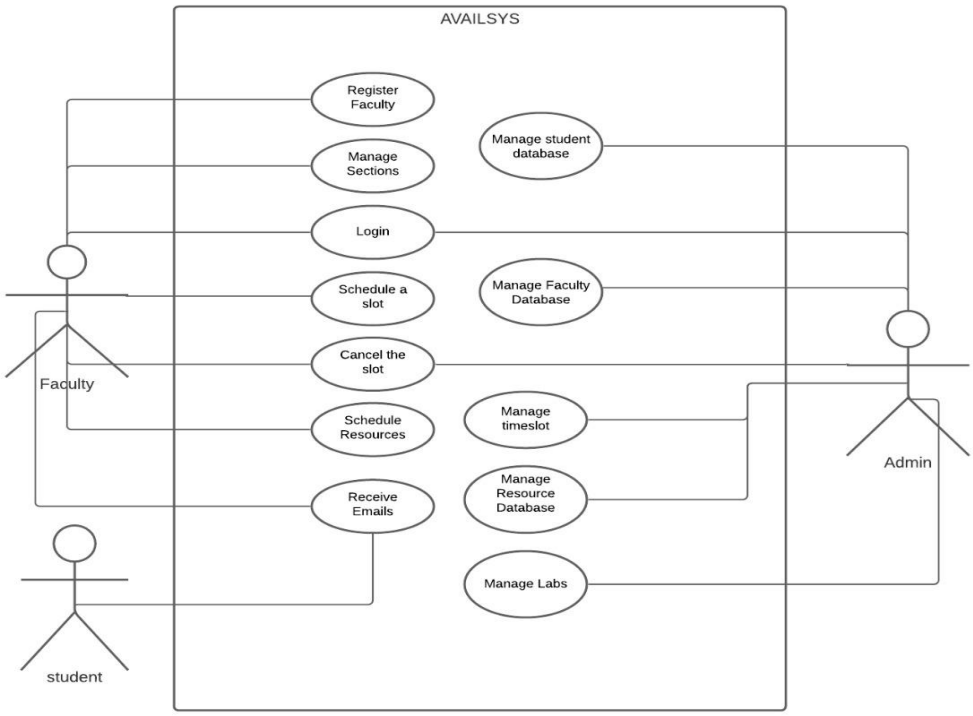
DBSqlite3 DataBases, OS(windows)

3.4 Communications Interfaces

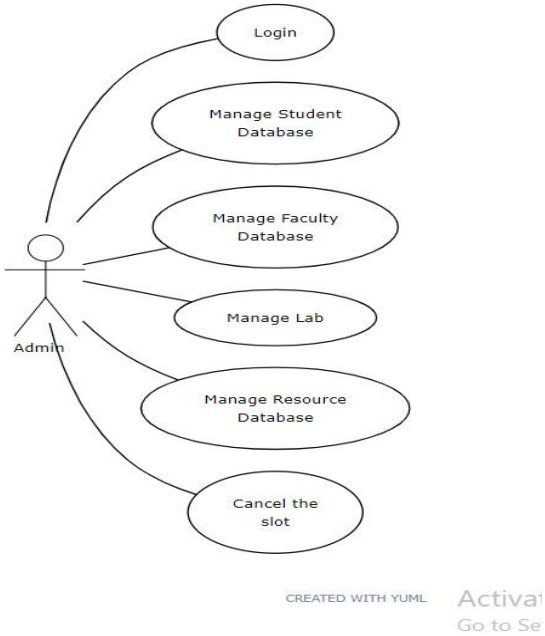
The System shall be using HTTP/HTTPS for communication over Internet and for intranet communications, it shall use TCP/IP protocol.

4. System Features

We describe functional requirements with the help of use-case diagrams of each actor.We can be classified into three sections for Administrators(Admin),Faculty and Students.



4.1 System features for Admin :



4.1.1 Use Case -1 : Login

Primary Actor *: Admin*
Precondition *: Internet Facility and should be a part of organisation.*
Main scenario : *Admin fills the details in the registration form and logins into the system.Get verification from administrator side.*
Alternate scenario : *It can be possible to create duplicate registration with similar details.*

4.1.2 Use Case -2 :Manage Student Database

Primary Actor *: Admin*
Precondition *: Internet Facility and students' successful detail registrations.*
Main scenario : *Students self register their details during reporting or joining in the organisation with self declaration of true details , and the admin gives a section as per number.*
Alternate scenario : *it can be possible that the student finishes or drops from the organisation and updates of regular results of the student.*

4.1.3 Use Case -3 :Manage Faculty Database

Primary Actor : *Admin*
Precondition : *Internet Facility and faculties' successful registration.*
Main scenario : *Faculty should themself register during their joining in the organisation , Admin allocates sections to faculty.*
Alternate Scenario : *it can be possible that the faculty leaves the organisation and regular update for the section allotted for faculty is to be made.*

4.1.4 Use Case -4 : Manage Lab

***Primary Actor** : Admin*
***Precondition** : Internet facility and labs' availability should be registered.*
***Main scenario** : Admin should register the available labs into the system and also the corresponding resources.*
***Alternate scenario** : there may be a possibility that the lab may be under renovation.*

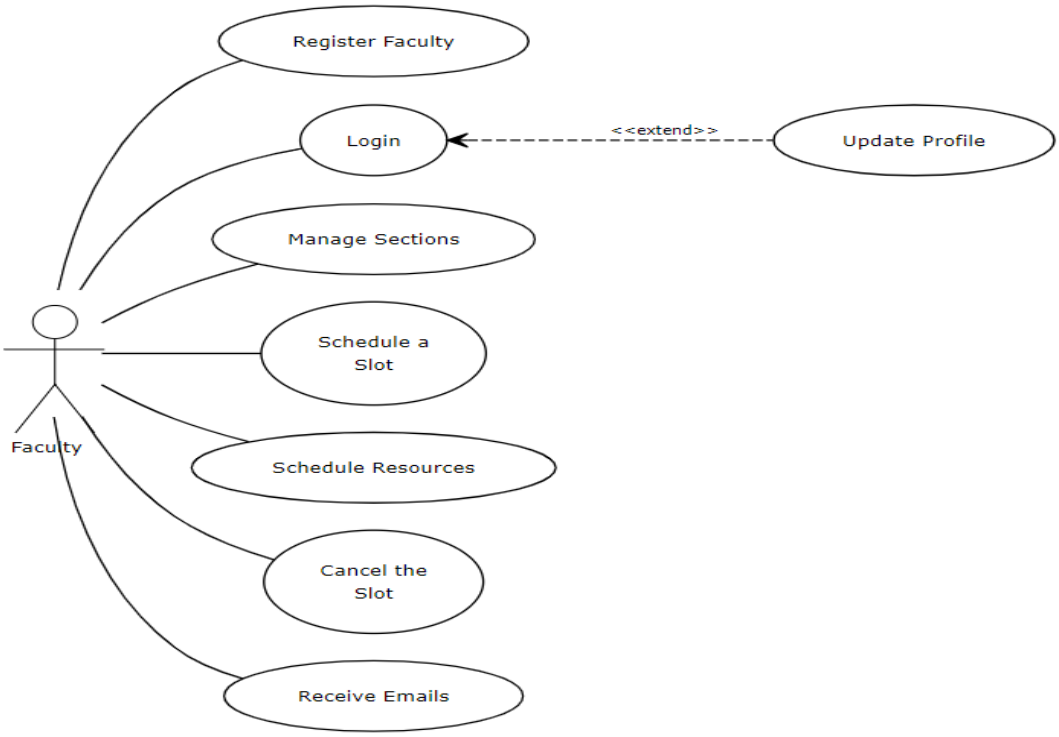
4.1.5 Use case -5: Manage resource Database

***Primary Actor**: Admin*
***Precondition:** Internet Facility and record the data of time slots.*
***Main scenario:** Admin should be able to record the data from students, faculty, and live check lab availability.*
***Alternate scenario:** there may be a possibility of updates in student, faculty databases.*

4.1.6 Use case -6: Cancel the Slot

***Primary Actor**: Admin*
***Precondition:** Internet Facility and a good awareness of lab resources.*
***Main scenario:** Admin has to cancel the slot opted by the faculty if there are any changes the resources required by the section according to labs.*
***Alternate scenario:** there may be a chance of cancellation of the slot without informing the faculty and students.*

4.2 System features for Faculty :



4.2.1 Use Case -7: Register Faculty

Primary Actor: *Faculty*
Precondition: *Internet facility and must be the part of the organization.*
Main Scenario: *Faculty fills details in the registration form and register. Get verification from the administrator side.*
Alternate Scenario : *It can be possible to create duplicate registration with similar details . so, users may get a pop-up message of duplicate registration.*

4.2.2 Use Case -8 : Login

Primary Actor: *Faculty*
Precondition: *Faculty should be registered and have a stable internet connection.*
Main Scenario: *Faculty logins using his credentials .*
Alternate Scenario : *Faculty gets an error message , if he enters wrong credentials.*

.

4.2.3 Use Case -9 : Manage Sections

Primary Actor: *Faculty*
Precondition: *Faculty must be logged in using his credentials .*
Main Scenario: *Faculty manages sections , if he wants to add or remove sections from database.He edit section details like students email id's.*

4.2.4 Use Case -10 : Schedule a slot for lab

Primary Actor: *Faculty*
Precondition: *Faculty must be logged in using his credentials .*
Main Scenario: *Faculty must select the section to book slot according to the requirement.According to availability, faculty is able to book slot.*
Alternate Scenario:*Faculty can face an error , if the slot is booked or capacity is insufficient.*

4.2.5 Use Case -11 : Schedule resources

Primary Actor: *Faculty*
Precondition: *Faculty must be logged in using his credentials .*
Main Scenario: *Faculty must schedule resources if necessary depending on the course he teaches.*
Alternate Scenario:*Faculty can face an error , if the resource capacity is insufficient.*

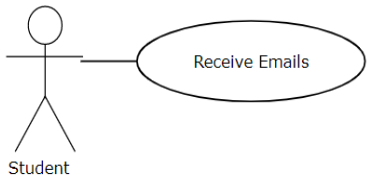
4.2.6 Use Case -12 : Cancel Slot

Primary Actor: *Faculty*
Precondition: *Faculty must schedule a slot for a section.*
Main Scenario: *Faculty have an option to cancel the slot booked , irrespective of the reason anytime before the slot begins.*

4.2.7 Use Case -13 : Receive Email

Primary Actor: *Faculty*
Precondition:*Faculty must schedule a slot and schedule resources if necessary.*
Main Scenario:*Faculty and corresponding section students will receive a confirmation Email of slot schedule.*

4.3 System features for Students :



4.3.1 Use Case -15: Receive Email

Primary Actor: *Students*
Precondition: *Faculty must schedule a slot and schedule resources if necessary.*
Main Scenario: *Faculty and corresponding section students will receive a confirmation Email of the slot schedule.*

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Version	Description	Prioritization Attribute
V1	Every scheduling submission	High

	& modification should be updated in websites's common database within 10 seconds after each submission & modification activity.	
V1	Results for cross checking of availability of labs in the internal database and Faculty's choice shall be in 5 seconds .	High
V1	Webpage UI load time should be within 2 secs	High
V1	Redirection page load time shall be within 2 secs	High

5.2 Safety Requirements

If there is any extensive damage to a wide portion of the database due to catastrophic failure , such as a disk crash , the recovery method restores a post copy of the database that was backed up to archival storage typically on our systems and reconstruct a more current state by reapplying the operations of managing slots and resources from the backed up database ,up to the time of failure.

5.3 Security Requirements

Admin will be able to log in to the Availsys application and have access to it .Access to it will be protected by a user login screen that requires credentials .Faculty profile details and credentials are encrypted.

5.4 Software Quality Attributes

Correctness: Our software will provide 95% accuracy.We may face failure because of sudden unexpected hardware function failures.

Reliability : Hosting service is in safe hands and most reliable servers.Data collected is encrypted , no security issues are faced.

Interoperability : Availsys can be developed and we can add any developments if required in the future in the dashboard interface provided for faculty and admin.

Usability: As it is a hosted application , it can be accessed from anywhere and anytime . Our website can adapt any screen without any screen resolutions issues.

5.5 Business Rules

Not applicable as it is a non-profitable website for synchronization of the labs for faculty and students.

6. Other Requirements

Admin and Faculty share a common database to store mandatory data to perform the required functions of software. Faculty shares his/her details , along with the details of students for communication purposes are stored in a common database along with credentials.Admin details and credentials are stored in a separate database.

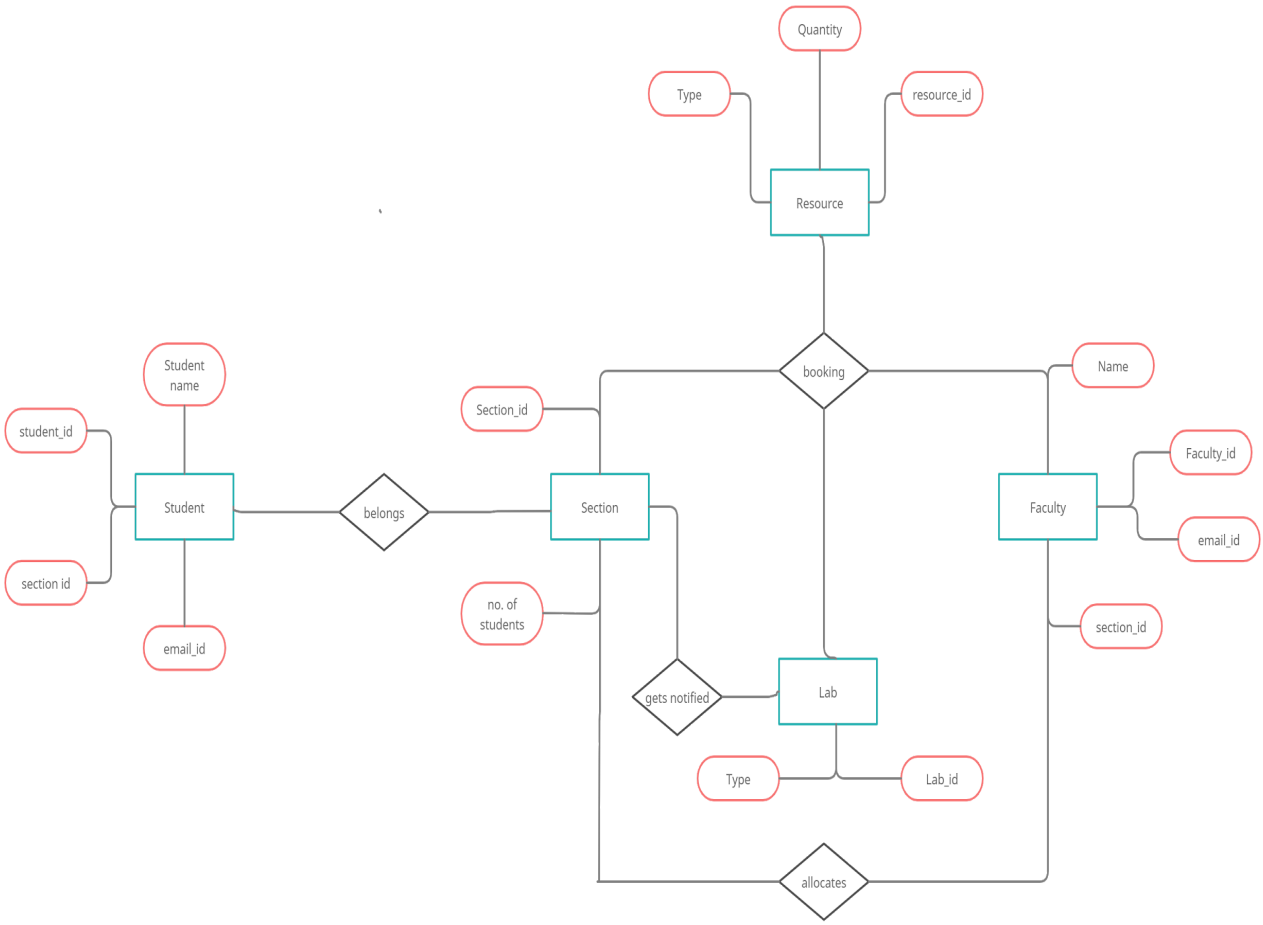
Appendix A: Glossary

Lab, Faculty,Students

Appendix B: Analysis Models

<*Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

E - R Diagram for AVAILSYS



: