E-Classroom Requirements Specification

Version 1.0

April 19, 2021

Project by:
Denada Rama
Denajd Hoxha
Marilena Shkurti
Natanael Xhelilaj
Rashid Serjani
Romario Braho

Table of Contents

1. Executive Summary

1.1 Project Overview

In this project, we aim to construct and deploy a free web & mobile service for the students of Epoka University, regarding the “Mentorship Program” of the school.

The "Mentorship Program" is a great initiative, designed for students offering support and help to other students that want to discover their interests and become high performing individuals. However, the program has continuously encountered difficulties through the years for various issues coming from both parties. As a result, especially during the time of pandemic, students who have applied for study guidance are being left without a response or supervision, and the students who have applied for mentoring, have not been assigned any tasks or students.

We’re offering a solution to this problem and a better approach for the coordination team, by building a software service where students are free to apply to available courses, or apply to become a mentor and open courses themselves, with some little guidance and supervision from the assigned coordinators of the service.

1.2 Purpose and Scope of this Specification

The purpose of the software is to provide a set of solutions to the students of Epoka University that wish to follow the “Mentorship Program”.

The purpose of this specification is to assess the state of our service, to define its requirements and functions in a high-level language and to document the entire process based on the software’s design, implementation and deployment.

This document addresses requirements related, and in scope of:

- In depth documentation of the features of the software (from user side & admin side)
- Technical overview of the system processes
    - User, Functional, Non-Functional and Domain Requirements
    - Means of designing the software
    - Development & Testing
    - Deployment & Maintenance
- Definition of users’ means of using and accessing the service
    - User Scenarios/Use Cases
- Description of possible dependencies and constraints

Aspects out of the scope for this specification:

- Rules & Regulations of the university that address the criteria of the program

2. Product/Service Description

2.1 Product Context

E-Classroom is an extremely helpful software which serves the purpose and mission of the “Mentorship Program”, in that it provides all necessary functionalities and benefits needed for the program to properly work. Given the difficulties encountered through the years, this online service fulfills the scope of requirements and also takes into consideration the rules and regulations, by providing the rightful privacy and security to all its users.

This service gives an easy, flexible and efficient solution to the mentor-students, who will be able to manage their own courses and classes, have a comfortable interaction with the participants and get their feedback. The course participants will have access to a huge number of documentations, helpful resources and instructions that will be provided by the mentors.

2.2 User Characteristics

If we divide the program in three major subsections, we are able to give a more detailed description of what the software provides for all parties.

Students:

1. See the available courses offered in their department & request to enrol in the ones they need help with
2. Follow the instructions of the mentor based on what they're assigned to study and complete
3. Elaborate on any questions or uncertainties to the private/group chat with the mentor
4. See the mentor's timetable for their free hours during the week
5. Request a one-to-one appointment, etc
6. Rate the mentor at the end of the course & see the Honours Dashboard for best-performance students during the semester

Mentors:

1. Create courses based on the subjects they feel confident to teach
2. Accept enrolment invitations on their classes - either one student or a group of students, up to them to decide
3. Assign reading lists, watching lists, quizzes to the students
4. Create a chat to communicate
5. Weekly update their free hours on their timetable
6. Keep a private to-do list on what things have been achieved until now, what tasks are being done and what are the future goals for the course
7. Retrieve the Certificate of Completion based on the ratings of the students & the course given

Academic Advisors:

1. Make decisions on the mentor requests, based on criteria like GPA, course performance, etc.
2. View mentor achievements throughout the year
3. Invite students to become mentors

2.3 Assumptions

It is assumed that only advisors have access to students' information regarding GPA and performance, in order to evaluate mentor requests & invitations.

It is assumed that the achievements of the mentors are confidential and only available to the advisor, unless wished differently from the mentor themselves.

It is assumed that the mentor understands the concept of privacy/confidentiality and does not share information regarding the course participants.

It is assumed that mentors will actively keep up with the work and interact with the course participants.

It is assumed that the mentors will generally follow the same workflow when it comes to posting activities and resources weekly.

It is assumed that by the end of the course/semester, the participants will give a fair & unbiased feedback and rating on their mentor.

It is assumed that all students who wish to participate in the program have access to a digital device like PC, laptop and/or mobile.

It is assumed that all students have strong internet service at the time of usage.

2.4 Constraints

This system will be potentially constrained by:

1. The fact that there has to be a wide range of courses provided by mentors.
2. The fact that the number of regular users (students) will probably always outcome the number of available mentors.
3. The students are constrained to only use their Epoka mails to join the program.
4. Potential constraint in the software performance since it depends on the usage traffic.
5. The need of a fast internet connection and strong mobile data signals.
6. Having all the students, but especially mentors, understand the way the service works and how to use it best.

2.5 Dependencies

-To be updated

Dependencies that affect the requirements:

- This service is dependent on a successful communication between the mentors and the students.
- This service is dependent on a strong capability from mentors to manage their own classes and optimize studying.
- Students can only follow courses depending on what the mentors have created, regardless of what subject they need most help with.
- Students can join the course depending on whether the mentor will accept them or not.

3. Requirements

Priority Definitions

The following definitions are intended as a guideline to prioritize requirements.

- Priority 1 – The requirement is a “must have” as outlined by policy/law
- Priority 2 – The requirement is needed for improved processing, and the fulfillment of the requirement will create immediate benefits
- Priority 3 – The requirement is a “nice to have” which may include new functionality

3.1 Functional Requirements

	Requirement	**Comments**	**Priority**	**Date Rvwd**	**SME Reviewed / Approved**
BR_01	The system should provide a web application which will be used by all types of users	This will be their main platform of operation	2		
BR_02	The system should provide the users with a cross-platform mobile application.	This will give them an alternative way of accessing the services provided by the system	2		
BR_03	The system should handle multiple account types	Based on the status of the user, (student, mentor, advisor), each will have their own view of the system	2		
BR_04	The system should handle unregistered users	Students that have not registered in the Mentorship program will only be able to see the Register option and not have access to any published course	2		
BR_05	The system should restrict account registration	Students and mentors should only be able to register with their Epoka mails	2		
BR_06	The system should require the department & role information when registering	Users should be required to specify the department they belong to and the type of service they’re applying for (student or mentor)	2		
BR_07	The system should not give access to the users applying as mentors, until advisor approval	All users that wish to become mentors have to be evaluated by their respective advisor, based on mentor criteria	2		

	Requirement	Comments	Priorit y	Date Rvwd	SME Reviewed / Approved
BR_08	The system should not give access to students that have requested to enroll in a course, until mentor approval	Students that wish to participate in a course will be able to do so only if the mentor agrees and accepts their request	2		
BR_09	The system should make available to the students to explore courses within their own department	Students can enroll only in courses that are published within the current semester by mentors in their department	2		
BR_10	Students should be able to access all courses they've enrolled in	This includes the courses that are currently being taught and the courses that have been completed	2		
BR_11	Within a course, students should be able to have access to all tasks and resources uploaded by the mentor	Tasks and resources including reading lists, links to tutorials, uploaded videos, compressed files, etc.	2		
BR_12	Within a course, students should have options available as to locally download the resources or access them on the browser	Since the resources may be compressed files or book attachments, the students should be able to use them properly	3		
BR_13	Within a course, students should view the uploaded tasks and resources structured in academic weeks	Specifically, each academic semester lasts approx. 16 weeks, so the files that the mentor assigns should correspond to the academic week they were uploaded (+ specific date and time)	3		
BR_14	Students should be able to access basic information about the mentor of a course they're enrolled in	Basic information includes: their full name, a preferred picture and bio, all courses the mentor is teaching for the current semester, the mentor's timetable for the current week	3		
BR_15	Students should be able to see the week's agenda within a course they're enrolled in	The agenda will include already appointed meetings, in the calendar, & free consultation hours that the mentor has entered	2		

	Requirement	**Comments**	**Priorit y**	**Date Rvwd**	**SME Reviewed / Approved**
BR_16	Students should be provided with an option to request an appointment with the mentor, within a course they're enrolled in	They have to specify the preferable date and time for the appointment, the type (one-to-one or online), and briefly describe the topics they wish to discuss	2		
BR_17	The system should provide the mentors the option to create a course based on their specifications	The specifications include the type of the course, code and name, department, etc.	2		
BR_18	The system should make available to the mentor all the courses they have created	All courses include the courses they're currently teaching and the courses they have finished	2		
BR_19	Mentors should have access to all students that are following their classes	This information should include all students, specifying the course code they're part of	2		
BR_20	The system should provide to the mentor the option of sending an email to students enrolled in their classes	This option should be provided in the same list where mentors can access all students	3		
BR_21	The system should provide to the mentor the option of removing a particular student from the course they're enrolled in	This option should be provided in the same list where mentors can access all students	3		
BR_22	The system should provide to the mentor the week's activity on all courses	Week's activity includes all posts and resources shared with the classes	3		
BR_23	The mentor should be able to decide what personal information they're comfortable sharing with the enrolled students	This means they can edit their profile anytime they want, they choose whether they want a profile picture or not & they choose what their bio says	2		
BR_24	The system should make available to the mentor to update their timetable at anytime	The general timetable keeps information about meetings of all the courses that the mentor is currently teaching	2		

	Requirement	**Comments**	**Priorit y**	**Date Rvwd**	**SME Reviewed / Approved**
BR_25	The system should provide to the mentor the option of accepting/declining all student requests for enrollment	The mentor decides which of these students and how many of them can follow the course that they're offering	2		
BR_26	The system should provide to the mentor an honor certificate upon course completion	The certificate will include the course, semester & student rating. The mentor should be able to download it anytime	2		
BR_27	The system should provide to the advisor the option of accepting/declining mentor requests	The advisor should evaluate the ability of a student to teach a course, based on criteria such as GPA and performance, and give them access to create courses	2		
BR_28	The system should provide to the advisor the option of inviting students to join the program as mentors	The students should receive an automated email from the advisor, inviting them to become mentors	3		

3.2 Non-Functional Requirements

3.2.1 User Interface Requirements

The main application shall be a web application, which can be seen either with any type of web browser and the other alternative shall be the mobile application providing similar services.

The main page of the web application, shall be a simple login interface, where it will ask the user for their Epoka mail and password. Based on the given credentials, the necessary system constraints will apply. The user will gain access to the system, in case of proven authenticity, otherwise an error message of invalid credentials will be displayed.
If the user is not registered, they will only have access to the general information about the Mentorship Program and will be prompted to join it.

As part of the system structure different system modules are included, depending on the type of user they were registered as. Such modules are dashboard module, courses module, students module, mentor module, timetable module, etc.

Once the user is logged in the web application, they shall have access to specific modules of the system. The system will redirect the regular user (students) to the courses module. This module will provide information to the user such as the courses they're currently enrolled in, the courses they've completed and the option to explore new courses. The system will redirect the admin (mentor) to the dashboard module, with course information, week's activity, to-do list and other actions.

Another module provided by the system is the timetable module. This module is a calendar of the week's agenda, including meetings and consultations. The students shall be able to view the timetable for each course they're currently enrolled in, the mentor shall be able to edit their own agenda that shall be shown in their profile and the timetable for every course they're teaching.

The most important module shall be the course module, where students will have access to assignments, tasks, reading lists, tutorials and other resources provided by the mentor of the course, structured in academic weeks.

By the end of the course, the mentor shall be awarded with a certificate of completion, auto-generated based on the students' evaluation of the mentor. The mentor shall be able to access and download their achievements at any time.

3.2.2 Usability

Accessibility & Learnability
The software shall be easy to access remotely and at all times, since both students and mentors will be able use the application from any of their devices.
Specific error messages will be displayed, by also identifying the specific action, that caused the error. The application is specified for certain users, thus the system will know, when a certain action is not allowed.

Flexibility
The software shall be easy to update in order to accommodate new requirements
The software shall be designed in such a way that the isolation and management of errors is possible

Effectiveness
The software shall provide the mentors with practical tools of managing their data and with a convenient way of communicating with their class participants.

3.2.3 Performance

- *3.2.3 to 3.3 To be rediscussed before updated*

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

- Static numerical requirements may include the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.
- Dynamic numerical requirements may include the number of transactions and tasks and the amount of data to be processed within certain time period for both normal and peak workload conditions.

All of these requirements should be stated in measurable form. For example, "95% of the transactions shall be processed in less than 1 second" rather than “an operator shall not have to wait for the transaction to complete”.

3.2.3.1 Capacity

Include measurable capacity requirements (e.g., the number of simultaneous users to be supported, the maximum simultaneous user load, per-user memory requirements, expected application throughput)

3.2.3.2 Availability

Include specific and measurable requirements for:

- Hours of operation
- Level of availability required
- Coverage for geographic areas
- Impact of downtime on users and business operations
- Impact of scheduled and unscheduled maintenance on uptime and maintenance communications procedures
- reliability (e.g., acceptable mean time between failures (MTBF), or the maximum permitted number of failures per hour).

3.2.3.3 Latency

Include explicit latency requirements, e.g., the maximum acceptable time (or average time) for a service request.

3.2.4 Manageability/Maintainability

3.2.4.1 Monitoring

Include any requirements for product or service health monitoring, failure conditions, error detection, logging, and correction.

3.2.4.2 Maintenance

Specify attributes of the system that relate to ease of maintenance. These requirements may relate to modularity, complexity, or interface design. Requirements should not be placed here simply because they are thought to be good design practices.

3.2.4.3 Operations

Specify any normal and special operations required by the user, including:

- periods of interactive operations and periods of unattended operations
- data processing support functions
- backup and recovery operations

- safety considerations and requirements
- disaster recovery and business resumption

3.2.5 System Interface/Integration

Specify the use of other required products (e.g., a database or operating system), and interfaces with other systems (e.g., UWHires package interfaces with PubCookie and ODS, HEPPS system interfaces with Budget system). For each interface, define the interface in terms of message format and content. For well-documented interfaces, simply provide a reference to the documentation.

Outline each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

3.2.5.1 Network and Hardware Interfaces

Specify the logical characteristics of each interface between the product and the hardware or network components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols (e.g., signal handshake protocols).

3.2.5.2 Systems Interfaces

Example systems interface requirements:

3.2.6 Security

3.2.6.1 Protection

Specify the factors that will protect the system from malicious or accidental access, modification, disclosure, destruction, or misuse. For example:

3.2.6.2 Authorization and Authentication

Specify the Authorization and Authentication factors. Consider using standard tools such as PubCookie.

3.2.7 Data Management

Specify the requirements for any information that is to be placed into a database, including

- types of information used by various functions
- frequency of use
- data access rules
- data entities and relationships
- integrity constraints
- data retention
- valid range, accuracy, and/or tolerance
- units of measure
- data formats
- default or initial values

3.2.8 Standards Compliance

Specify the requirements derived from existing standards, policies, regulations, or laws (e.g., report format, data naming, accounting procedures, audit tracing). For example, this could specify the requirement for software to trace processing activity. Such traces are needed for some applications to meet minimum regulatory or financial standards. An audit trace requirement may, for example, state that all changes to a payroll database must be recorded in a trace file with before and after values.

3.2.9 Portability

If portability is a requirement, specify attributes of the system that relate to the ease of porting the system to other host machines and/or operating systems. For example,

- Percentage of components with host-dependent code;
- Percentage of code that is host dependent;
- Use of a proven portable language;
- Use of a particular compiler or language subset;
- Use of a particular operating system;
- The need for environment-independence - the product must operate the same regardless of operating systems, networks, development or production environments.

3.2.10 Other Non-Functional Requirements

Please provide all necessary non-functional requirements, similar to the requirements explained in the lesson slides or in the textbook.

3.3 Domain Requirements

Everything related to the domain that might be needed in the project shall be mentioned in here. Sometimes the domain Requirements might be thought as part of either functional or non-functional requirements.

4. User Scenarios/Use Cases

4.1 User Scenarios

Notice: We are using the abbreviations U – referring to students (users), A – referring to mentors (admins).
We refer to User if it's not specified whether they're a student or a mentor, otherwise we specify the user type.

Scenario U1: User is not logged in

1. User is presented with a Login/Signup option.
2. User is presented with detailed info about the program.

Scenario U2: User is registering for the first time

1. User is asked to enter their active Epoka mail
2. User is asked to enter and confirm a valid password
3. User is asked to specify the department they're in
4. User is asked to choose the type of account they want to create (student or mentor)
5. User account is pended until the respective advisor approves if the type of account is "mentor"
6. User is redirected to explore courses if the type of account is "student"
7. Sign out button is added to the dashboard

Scenario U3: User fails to register

1. User is prompted the specific error that is occurring (wrong email, not Epoka mail, invalid password, etc.)
2. User retries to enter the right credentials

Scenario U4: User is successfully logged in

1. User is asked to enter their Epoka mail
2. User is asked to enter their password
3. If their credentials are matched in the database, they are authorized to be redirected
4. User gets redirected to the dashboard of the website

Scenario U5: User failed to login

1. User is asked to enter their Epoka mail
2. User is asked to enter their password
3. The credentials that were entered are not found in the database
4. User is displayed an error message "Your credentials were not correct, please try again"
5. User tries to enter the credentials again

Scenario U6: Student requests to enroll in a course

1. Student is able to see all courses published for the current semester by mentors in their department
2. Student wants to participate in a specific course
3. Student requests to enroll in the course
4. Student waits for the mentor's confirmation

Scenario U7: Student is accepted to attend a course

1. Student has gained access to the course by the mentor
2. Student can now see the dashboard of the course, containing assignments and resources posted by the mentor in academic weeks

3. Student has access to all services within the course provided for students

Scenario U8: Student wants to read the assignments of a course

1. Student can see the assignments structured in academic weeks
2. Student is only able to view, open and download the resources
3. Student will have two options in the corner of each assignment: open in new tab or download locally
4. Student chooses to open the files in the web browser and the files will be shown in a new tab if the browser supports the file type
5. Student chooses to download the files in their local device and they'll be able to access the files where they store it

Scenario U9: Student wants to see the mentor's profile

1. For each course enrolled in, student will be able to view basic information for the mentor of the course
2. Student can see a profile picture, followed by a brief bio description
3. Student can see all the courses the mentor is teaching in the current semester, followed by a link to the course and an option to request enrollment in other courses
4. Student can see the mentor's timetable for the current week, with the agenda for all their courses together

Scenario U10: Student sees the timetable within a course they're enrolled in

1. For each course enrolled in, student will be able to view the week's agenda provided by the mentor for that specific course
2. The timetable will include the meeting that have already been decided
3. The timetable will include free consultation hours provided by the mentor if the student wishes to request an appointment

Scenario U11: Student requests an appointment with the mentor

1. Student is prompted to check the course's timetable before setting a preferrable date and time for the appointment
2. Student is asked to enter the date and time for the appointment
3. Student is asked to specify the type of appointment (one-to-one or online meeting)
4. Student is asked to briefly describe the topics they wish to discuss in the meeting
5. Student waits for mentor confirmation

Scenario A1: Mentor creates a new course

1. Mentor is asked to specify the course code and course name
2. Mentor is asked to specify the department the course belongs to (for documentation purpose only)
3. Mentor is only able to teach courses within their department
4. Mentor publishes the course and waits for student requests

Scenario A2: Mentor views all courses they've taught

1. Mentor will be able to access all courses at any time
2. Mentor will access the courses in two different sections: the courses they're currently teaching and the courses they've completed
3. Mentor will be able to edit and post only on the ongoing courses
4. Mentor will be able to open access any specific course by being redirected to a new tab

Scenario A3: Mentor sees all the students that are following their courses

1. Mentor will have access to a table containing the names of all students that are currently enrolled in the courses and the respective course code
2. Mentor has an option to email to each student, in their Epoka mails
3. Mentor has an option to remove any student

Scenario A4: Mentor chooses to send email to a student

1. Mentor presses the “mail” option in the All-Students table
2. Mentor is redirected to their Epoka compose mail option in Gmail account, with the student’s mail as recipient
3. Mentor is able to compose and send the desired email from there

Scenario A5: Mentor chooses to remove a student from a specific course

1. Mentor presses the “remove” option next to student name in the All-Students table
2. Mentor is prompted to reconfirm the removal of the student
3. Once the mentor chooses “Remove”, the student is deleted from the student list and will no longer have access to course materials

Scenario A6: Mentor sees all students that have requested to enroll

1. Mentor will have access to a table containing all student requests for the courses of the current semester
2. Mentor will see the student’s full name, Epoka mail, and the course code they’ve requested to enter
3. Mentor will have an option to accept/decline the student
4. Once the mentor accepts a request, they’re giving the student access to course materials and the student will show on the list

Scenario A7: Mentor opens and edits “My Profile”

1. Mentor will have access to their profile and choose what will be visible to the students
2. Mentor will be able to update their profile picture at any time
3. Mentor will be able to edit their bio
4. Mentor will be able to add or delete course links

Scenario A8: Mentor edits “My Timetable”

1. Mentor will have access to their personal agenda for the current week
2. Mentor will have all the courses they’re currently teaching listed in a table
3. Mentor will be able to drag and drop the courses in the timetable in the preferred time of the day
4. Once the mentor chooses “Update timetable”, the timetable will be automatically updated and shown in their profile, when accessed by students.

Scenario A9: Mentor sees the current week’s activity in dashboard

1. Mentor will access the week’s activity in their dashboard
2. The activities include course materials poster, meetings held, and other resources
3. The activities will be listed by date, starting from the most recent post

Scenario A10: Mentor posts materials in a course

1. Mentor will access the homepage of each course they’re teaching for the semester
2. Mentor will access the recourses and activities structured in academic weeks
3. Mentor will have options to add, edit, open, copy, delete or change settings of the entire academic week

4. Mentor has the same options for each individual material within the academic week
5. After they post the new materials that they wish to share with the class, they choose "Make visible" option
6. The new materials are made public to the course participants

Scenario A11: Mentor sees all the students following a specific course

1. Mentor will have access to a table containing the names of all students that are currently enrolled in the specific course they're accessing
2. Mentor has an option to email to each student, in their Epoka mails
3. Mentor has an option to remove any student

Scenario A12: Mentor edits "Course Timetable"

1. Mentor will have access to the timetable for the specific course
2. Mentor will have to specify already arranged meetings and free hour consultations
3. Mentor will be able to drag and drop the meeting or the consultation in the timetable in the preferred time of the day
4. Once the mentor chooses "Update timetable", the timetable will be automatically updated and shown in "View Timetable" module of the students.

Scenario A13: Mentor receives appointment request

1. Mentor gets a notification for a requested appointment by a student
2. Mentor sees the student that made the request, the time of appointment, the type and a brief topic description
3. Mentor is prompted to choose whether to accept or decline the appointment
4. If mentor accepts, the appointment is added to the "Course Timetable" agenda

Scenario A14: Mentor receives certificate upon course completion

1. Mentor has access to their achievements
2. Upon course completion and student ratings, the "My achievement" module is automatically updated
3. The module shows the course name, semester taught and overall student rating
4. Mentor has the option to download the certificate
5. The certificate will be in the pdf format, containing mentor, course and rating information
6. The certificate will be issued by the CPAO of Epoka University and will be valid for life.

APPENDIX

The appendixes are not always considered part of the actual Requirements Specification and are not always necessary. They may include

- Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
- Supporting or background information that can help the readers of the Requirements Specification;
- A description of the problems to be solved by the system;
- Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.

When appendixes are included, the Requirements Specification should explicitly state whether or not the appendixes are to be considered part of the requirements.

Appendix A. Definitions, Acronyms, and Abbreviations

Define all terms, acronyms, and abbreviations used in this document.

Appendix B. References

List all the documents and other materials referenced in this document.

Appendix C. Requirements Traceability Matrix

The following trace matrix examples show one possible use of naming standards for deliverables (FunctionalArea-DocType-NN). The number has no other meaning than to keep the documents unique. For example, the Bargaining Unit Assignment Process Flow would be BUA-PF-01.

For example (1):

Business Requirement	Area	Deliverables	Status
BR_LR_01 The system should validate the relationship between Bargaining Unit/Location and Job Class.---Comments: Business Process = "Assigning a Bargaining Unit to an Appointment" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
		BUA-PF-01 Derive Bargaining Unit-Process Flow Diagram	Accepted
BR_LR_09 The system should provide the capability for the Labor Relations Office to maintain the job class/union relationship.---Comments: Business Process = "Maintenance" (Priority 1)	BUA	BUA-CD-01 Assign BU Conceptual Design	Accepted
		BUA-PF-02 BU Assignment Rules Maint Process Flow Diagram	ReadyForReview

For example (2):

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_01	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_01	1	BUA	BUA-PF-01	Derive Bargaining Unit-Process Flow Diagram	Accepted

BizReqID	Pri	Major Area	DevTstItems DelivID	Deliv Name	Status
BR_LR_01	1	BUA	BUA-UCD-01	BU Assign LR UseCase Diagram	ReadyForReview
BR_LR_01	1	BUA	BUA-UCT-001	BU Assignment by PC UseCase - Add Appointment and Derive UBU	Reviewed
BR_LR_01	1	BUA	BUA-UCT-002	BU Assignment by PC UseCase - Add Appointment (UBU Not Found)	Reviewed
BR_LR_01	1	BUA	BUA-UCT-006	BU Assignment by PC UseCase - Modify Appointment (Removed UBU)	Reviewed
BR_LR_09	1	BUA	BUA-CD-01	Assign BU Conceptual Design	Accepted
BR_LR_09	1	BUA	BUA-DS-02	Bargaining Unit Assignment DB Modification Description	Accepted
BR_LR_09	1	BUA	BUA-PF-02	BU Assignment Rules Maint Process Flow Diagram	Accepted
BR_LR_09	1	BUA	BUA-UCD-03	BU Assign Rules Maint UseCase Diagram	Reviewed
BR_LR_09	1	BUA	BUA-UCT-045	BU Assignment Rules Maint: Successfully Add New Assignment Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-051	BU Assignment Rules MaintUseCase: Modify Rule	Reviewed
BR_LR_09	1	BUA	BUA-UCT-053	BU Assignment Rules MaintUseCase - Review Assignment Rules	Reviewed
BR_LR_09	1	BUA	BUA-UCT-057	BU Assignment Rules MaintUseCase: Inactivate Last Rule for a BU	Reviewed
BR_LR_09	1	BUA	BUA-UI-02	BU AssignRules Maint UI Mockups	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-021	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-027	BU Assignment Rules Maint TestCase: Modify Rule - Success	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-035	BU Assignment Rules Maint TestCase: Add New Rule (Associated Job Class Does Not Exist) - Error Condition	ReadyForReview
BR_LR_09	1	BUA	BUA-TC-049	BU Assignment Rules Maint TestCase: Modify Rule - Error Condition	ReadyForReview

For example (3):

BizReqID	CD01	CD02	CD03	CD04	UI01	UI02	UCT01	UCT02	UCT03	TC01	TC02	TC03	TC04
BR_LR_01			X		X		X			X		X	
BR_LR_09	X			X		X			X		X		X
BR_LR_10	X			X					X		X		
BR_LR_11		X											

Appendix D. Organizing the Requirements

This section is for information only as an aid in preparing the requirements document.

Detailed requirements tend to be extensive. Give careful consideration to your organization scheme. Some examples of organization schemes are described below:

By System Mode
Some systems behave quite differently depending on the mode of operation. For example, a control system may have different sets of functions depending on its mode: training, normal, or emergency.

By User Class
Some systems provide different sets of functions to different classes of users. For example, an elevator control system presents different capabilities to passengers, maintenance workers, and fire fighters.

By Objects
Objects are real-world entities that have a counterpart within the system. For example, in a patient monitoring system, objects include patients, sensors, nurses, rooms, physicians, medicines, etc. Associated with each object is a set of attributes (of that object) and functions (performed by that object). These functions are also called services, methods, or processes. Note that sets of objects may share attributes and services. These are grouped together as classes.

By Feature
A feature is an externally desired service by the system that may require a sequence of inputs to affect the desired result. For example, in a telephone system, features include local call, call forwarding, and conference call. Each feature is generally described in a sequence of stimulus-response pairs, and may include validity checks on inputs, exact sequencing of operations, responses to abnormal situations, including error handling and recovery, effects of parameters, relationships of inputs to outputs, including input/output sequences and formulas for input to output.

By Stimulus
Some systems can be best organized by describing their functions in terms of stimuli. For example, the functions of an automatic aircraft landing system may be organized into sections for loss of power, wind shear, sudden change in roll, vertical velocity excessive, etc.

By Response
Some systems can be best organized by describing all the functions in support of the generation of a response. For example, the functions of a personnel system may be organized into sections corresponding to all functions associated with generating paychecks, all functions associated with generating a current list of employees, etc.

By Functional Hierarchy
When none of the above organizational schemes prove helpful, the overall functionality can be organized into a hierarchy of functions organized by common inputs, common outputs, or common internal data access. Data flow diagrams and data dictionaries can be used to show the relationships between and among the functions and data.

Additional Comments
Whenever a new Requirements Specification is contemplated, more than one of the organizational techniques given above may be appropriate. In such cases, organize the specific requirements for multiple hierarchies tailored to the specific needs of the system under specification.

There are many notations, methods, and automated support tools available to aid in the documentation of requirements. For the most part, their usefulness is a function of organization. For example, when organizing by mode, finite state machines or state charts may prove helpful; when organizing by object, object-oriented analysis may prove helpful; when organizing by feature, stimulus-response sequences may prove helpful; and when organizing by functional hierarchy, data flow diagrams and data dictionaries may prove helpful.