Epoka e-Learner Requirements Specification

Version 1.0

May 14, 2021

CEN 302 – Software Engineering

Faculty of Architecture and Engineering

Department of Computer Engineering

Sonia Memaj, Kevin Kurtaja, Nos Qirjako, Romina Hoxha, Marinela Alishollari, Ledia Leka

Table of Contents

1. Executive Summary

1.1 Project Overview

Having to study remotely has made the use of online learning platforms an absolute necessity. Epoka university offers well-organized structures for this purpose however since all of the school-related activities are carried out online some deficiencies are made evident. The biggest problem being the use of too many platforms across all courses. This including Google Classroom, Epoka LMS, EIS, and Epoka official site.
What this project aims, is to create an improved learning experience for Epoka University students, based on the already existing online platforms that the university uses. We intend on creating a mobile application to facilitate the student experience. Our application will offer a space where students can access all the necessary information during their studies and also create a communication channel between them and the professors. One of the goals is to make this interaction easier for a smoother learning experience.

1.2 Purpose and Scope of this Specification

The purpose of this project is to create a mobile application that will be an extension of Epoka Interactive System. This will provide students with the utilities that EIS offers. At the same time some of the information offered from Epoka's official site will be included. The students will also be able to access their courses and lectures which will be updated by the professors.

In scope:

- Creating a simple and efficient user experience for the users of this application (Students and Lecturers).
- Updating the legislation accordingly to guarantee the privacy and data safety of its users.
- The retreival of official documents from the university with the appropriate signatures, seal and specifications.

Out of Scope:

- Managing and having access to student data as they are considered confidential information.
- Creating an actual connection to EIS. The project will be working with static data which will be used as test data to check the quality and functionality of the application. So, an actual connection could only be made by the university officially once the project is complete.

2. Product/Service Description

2.1 Product Context

This product will be connected to EIS and Epoka LMS to get the necessary student information and the lectures for each course.

2.2 User Characteristics

Users include:

1. **Students:**
    - Views courses, lectures, grades, attendance, finances.
    - Gets a specific timetable view based on year, department, and group.
    - Can request documents.
    - Can view events or news relating to the university.

- Can select courses.
- Can make posts on their course page.
- Can view or request transportation (request was made available past year because of the pandemic, not sure if it will be a permanent feature).
- Can access the school library digital catalog.
- Can communicate with the professors through private comments.

2. Lecturers/Professors:

- Views and edits courses, lectures, attendance.
- Gets a specific timetable view.
- Can view events or news relating to the university.
- Can access the school library digital catalog.
- Can view or request transportation (request was made available past year because of the pandemic, not sure if it will be a permanent feature).
- Can communicate with the students through private comments.

3. Advisor:

- Has the same privileges as the professor.
- Can see more information about the students including their GPA.
- Can make course approvals.

All users will be logged in with Epoka email account. Lecturers who were once students will be logged in with their new Epoka email account and with this account will have lecture privileges. The advisor will be a professor with that same account, they will just be granted more privileges.

2.3 Assumptions

The application will need a persistent network connection. Users will also need a mobile device to run this application on. They need to be using either iOS or Android.

2.4 Constraints

1. Mobile-based frameworks and programming languages.
2. Security concerns. Considering there will be a lot of personal information on the users, security is a main priority.
3. Memory space.

2.5 Dependencies

4. Most of the updates will be made by the professors, so students will rely heavily on them for their grades, attendance etc.
5. Other updates will be made by the university staff which will be represented in the database.
6. We need a simple user interface(prototype) to start testing the connection.

1. Requirements

1.1 Functional Requirements

EL_GR => E-Learner General Requirements

EL_SR => E-Learner Student Requirements

EL_PR => E-Learner Professor Requirements

EL_AR => E-Learner Advisor Requirements

Req#	Requirement	Comments	Priority	Date Rvwd	SME Reviewed / Approved
EL_GR_01	Users have different privileges but same base view.	Business Process = “Maintenance	3	20/04/21	Bob Dylan, Mick Jagger
EL_GR_02	Users should be able to log in and log out, no sign up available.		1	20/04/21	
EL_GR_03	After leaving the app users should not be logged out.		1	20/04/21	
EL_GR_04	Each user has their own profile, they cannot edit their profile data.	Same data as their Epoka email account.	1	20/04/21	
EL_GR_05	Users can access emergency tab.		1	20/04/21	
EL_GR_06	Users can see all of the events which will be updated in real time.		3	20/04/21	
EL_GR_07	Users can access the full timetable, the first view they get will be personalized.		2	20/04/21	
EL_GR_08	Grade and GPA will be calculated automatically.		1	20/04/21	
EL_AR_01	Advisors can confirm course selection.		1	20/04/21	
EL_AR_02	Advisors can confirm students eligible for GP.		2	20/04/21	
EL_AR_03	Advisors cannot edit student’s information.		1	20/04/21	
EL_PR_01	Professors cannot create courses. Courses are created automatically with data from the course selection.		1	20/04/21	
EL_SR_01	Students can access only selected courses.		1	20/04/21	

1.2 Non-Functional Requirements

1.2.1 User Interface Requirements

In addition to functions required, describe the characteristics of each interface between the product and its users (e.g., required screen formats/organization, report layouts, menu structures, error and other messages, or function keys).

1. Firstly, there is the login in view containing fields for username and password, forgot password button and continue with Gmail button.
2. After logging in the opening view will be the same for each user. There are two options so far. Having a feed for the news, a bottom navigation bar and a sidebar menu which contains all of the necessary tabs (courses, attendance etc). Or having small box like components with all of that information. More information on the views will be given in the sketches.

1.2.2 Usability

7. Learnability:
    - The application should be easy to use. A lot of the functionalities are similar to platforms that the students are already familiar with, like Epoka LMS, EIS and Google Classroom, so the users won't need specific training.
    - When first logging in to the app there will be tips to guide the users through the app.
    - In case of error message, the cause for the error will be displayed.

1.2.3 Performance

8. Depends on the user's internet connection speed and strength.
9. Depends on the hardware that the user uses to access the application.
10. It also depends on the number of users that can access the application at the same time.

3.2.3.1 Capacity

The database will be working with static data. The first tests will be done with a small amount of users, then later on we will increase the number of users close to the actual number of students and academic staff that will be using the application.

3.2.3.2 Availability

11. Application should be available at all times.
12. It will be only for the intended users (students and professors of Epoka).
13. There should be as little downtime as possible, even outside of working hours, as students need to be able to access the application at all times.
14. Accessible at any geographical location.
15. In case of planned downtime due to maintenance the users should be notified beforehand.

3.2.3.3 Latency

- Depends on the internet connection strength which is not covered by the system.
- Depends on the size of the database.
- Latency for documents to be uploaded.
- App should open for at most 1.5 s.

1.2.4 Manageability/Maintainability

3.2.4.1 Monitoring

- When user enters correct credentials, they are logged in. If not, an error message will be displayed telling them that either their password or username was not correct. There will be an option in case password is forgotten. It will be reset using email.
- App performance monitoring will be done through app statistics and user ratings.

3.2.4.2 Maintenance

- Updates should be documented while preserving system integrity.
- Users will be informed in case of errors or downtime due to maintenance.

3.2.4.3 Operations

16. The users require login at any time.
17. Each user will have different accessibilities based on their needs. They are explained more in detail in the requirements specification.
18. Different types of users will be able to access some of the same tabs (timetable, calendar, library etc).
19. Each user should have access to notifications based on updates or recent posts.

1.2.5 System Interface/Integration

3.2.5.1 Network and Hardware Interfaces

The application will run on mobile phones (Android and iOS). It can run on most phones without a problem, the mobile phone is the only hardware required.

3.2.5.2 Systems Interfaces

The professor interface can make assignment posts where the student interface will be able to upload the document to be saved to a firebase storage. The professor interface can make a post where the student interface can comment on, increasing the communication. The student interface can make the same thing.

1.2.6 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:

20. Activity and error logging.
21. Password encryption.
22. Users will be logged using their Epoka mail.
23. Professors can only access the grades and attendance of students for their courses.
24. Advisors will have some more privileges like the students' GPA and some of their general personal information.
25. Other personal student information like the ones required on the documents given by registrar's office can be accessed only by them. It will be protected in the database.

3.2.6.2 Authorization and Authentication

- Single-factor authentication.
- Username and password login or login with Gmail, only with Epoka emails.
- In case of password change, the new password can be set via email.
- Each category of user is authorized with their specific accessibilities.

1.2.7 Data Management

All the information that will be dealt with in this application, is specific to students and professors. So, the attributes will be specific to the users.

1.2.8 Standards Compliance

Considering the personal and sensitive information this application will handle, it is necessary that it will comply to a set of regulations that protect these data. To protect the privacy of the users, the application will comply with EU's GDPR (General Data Protection Regulation) which addresses data protection, transfers and privacy.

1.2.9 Portability

The application can be used on a mobile phone (Android iOS) that has an internet connection.

3.3 Domain Requirements

- Users can access the lecture and transport timetables.
- They can access general information about each faculty and department.
- They can access the library system.

4. User Scenarios/Use Cases

1. User Scenario 1 – finance approval
    a. Student makes payment.
    b. Finance makes approval.
    c. Changes are made in the system.
        i. Semester fee is considered paid. It shows in the finance section.
        ii. Courses are approved by finance. It shows in course selection.
        iii. Documents are approved by finance. It shows in the document request section.
2. User Scenario 2 – course selection / approval
    a. Finance approves the courses.
    b. Students select the courses they will take for the semester.
    c. If the selection is correct then they receive advisor approval.
3. User Scenario 3 – document request
    a. Students select document type, number of copies and add notes(optional).
    b. Students select submit button.
    c. If they have made the payment, they will receive finance approval.
    d. Once they have the approval, they will receive a notification that the payment is approved.
    e. They receive registrar approval.
    f. When the documents are ready, they will get a notification (and/or email) that the documents are ready to pick up.
4. User Scenario 4 – making post
    a. Users click on post section in the tab of the specific course.
    b. Users click on the add button to add any type of file.
    c. After finishing writing the post, they click on the post button.
    d. After the post is made other users can comment on the post. They cannot add files in the comments.
5. User Scenario 5 – submitting grades
    a. Professors add type of grade.
    b. They add weight of grade.
    c. They add the grade, class average is calculated automatically.
    d. They add date and of the grade.
    e. Course progress bar is updated automatically.
    f. After changes are saved, the grades will appear to the students on the grades section and on the course section (only the current average).
6. User Scenario 6 – submitting grades
    a. Professors add attendance week.

b. Professors add lecture topic.
c. They add the attended hours and date and time.
d. Once changes are saved the attendance shows to the student on the attendance tab and on the specific course tab (the current attendance).

APPENDIX

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

26. Sample input/output formats, descriptions of cost analysis studies, or results of user surveys;
27. Supporting or background information that can help the readers of the Requirements Specification;
28. A description of the problems to be solved by the system;
29. 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.