Software Engineering Project (SEPRO) Practical Assignment 2021/2022

SOFTWARE PROJECT No. 1

UAM Vehicle Sharing Software System

PRACTICAL ASSIGNMENT No. 1

Technical Report: Initial Software Requirements Specification of a Software Project

The Autonomous University of Madrid (UAM) has announced the eleventh edition of the competition for IT Innovation Projects. These projects must be associated with improvements in mobility in private transport to go to UAM. The objective of this competition, named *EcoUAM*, is to develop a web software project to specify the requirements that a software with these features should have, allowing the management of private transport between students, teaching staff and researchers, as well as UAM administration and services staff. The system should use the Google Maps or Apple Maps tools. After the project has been awarded, in a second phase, the implementation will be carried out, which is out of the scope of this project.

This interactive system will oversee to provide information and assistance to these UAM members about private vehicles which are available to be used as a means of transportation when arriving or departing the UAM. *EcoUAM* is part of the UAM's spirit to foster a fluid, sustainable and collaborative way of mobility to or from the University campus within the framework of the goals of the UAM Ecocampus Office.

This assignment is a compulsory task. Each team acts as a software company, which submits its original proposal to the UAM competition. Each company shall write a technical report and propose a name for the software project that best fits for the proposal.

During this first phase of the project, the UAM shall firstly inform you a set of ideas of what it would like to have for its students, faculty, and staff, but you must assess its inclusion and the way in which the University has delegated you the definition of this project as teams of experts that you are. Secondly, within the competition framework of *EcoUAM,* UAM shall study the different proposals to determine the one that best fits the objectives set by the competition jury. The second phase of the project will be evaluated according to the results obtained during the first phase with the purpose to provide the selected software system to the rest of the Spanish public universities that stand up for a more sustainable mobility to reduce greenhouse gas emissions and to stop the deterioration of the environment.

The proposal of the web application should optimise the management and use of private transport considering the following aspects: a driver user can publish that she owns a car and would like to carpool in a particular date and time, informing itinerary, available seats and price per seat; a passenger user can look available rides up and reserve one seat for a ride; the system carries out the reservation management (and its archival storage), as well as the payment management defined in the application; notifications sent by the application; integration of Google Maps and/or Apple Maps in the corresponding functionalities; query user profiles by means of published rides; messages delivery and receipt between users.

In addition, UAM students, faculty and staff can sign up in this carpool service and pay the corresponding payments for its use. With this purpose, UAM provides a system for the automatic authentication for all the users of the application, which returns the authenticated user and her role in the University. This UAM available authentication system shall be integrated in the mentioned application, which shall include all the features that the software development team will consider convenient for a carpool system in the UAM. Moreover, there exists an available component: the payment gateway, which shall be used in the application for the defined payment system. Therefore, external systems provided by UAM are the authentication platform and the payment gateway.

As mentioned, to cover these aspects which the University provides, the list of ideas in the following environment is given:

- Users Management.
- Publication of Rides.
- Ride Seat Booking Management.
- Payment Management.

In these descriptions, the following **notation** for each environment is used:

[Explanatory Notes]
< *To be completed by the working team of software engineers* >

USERS MANAGEMENT

1. Only authenticated users will be able to access the application (except for the login page).
2. When users log in, the system will verify the access data (UAM email and password) entered by them with the UAM authentication system.
   2.1. If the authentication is accepted, the system will assign a session to the user (authenticated user).
   2.2. <*The working team must specify what happens if the authentication is not accepted*>.
3. When users log out, the system will delete their session (unauthenticated user). Note: Do not delete the system profile.

4. When users access the system for the first time, an "empty" profile will be created (Note: The system knows who the user is by the information retrieved by the UAM authentication system).
   [In addition, users will also be associated with a virtual payment account (see Payment Management environment).]
5. The system will allow users to fill in (edit / modify) their profile data: name, age, profile (student, faculty, or staff), place of residence, place of study/work at UAM, *<the working team must specify additional information to be included in the profile>*.
6. Authenticated users will be able to see the rides published and reserved by them (historical) as well as the payments they have made for each ride.

PUBLICATION OF RIDES

7. Authenticated users will be able to publish ride for carpooling (from now on, **drivers**):
   7.1. They must indicate place of departure, date and time, number of persons admitted, whether smoking is allowed in the vehicle, destination, and price per seat. They can also include (optional) a textual description with pick-up points along the ride and/or type of vehicle (motorcycle, car, minivan, van, etc.).
   7.2. Optionally, users can indicate whether it is an occasional or recurring ride.
      7.2.1. For frequent rides drivers must indicate how often they take them and a date when they estimate to quit taking this ride (maximum 2 months).
      7.2.2. The system will register all the rides configured by the user (a single ride for sporadic rides, and all possible rides for recurring rides).
8. The system will consider each repetition as an independent ride.
9. Drivers will be able to change the number of available seats if it does not affect a reserved seat (that is, a seat that has been reserved cannot be removed).
10. Drivers will be able to modify any information of the ride if no reservation has been made for it. When recurring rides are modified, drivers must indicate whether the modification affects just one specific ride or all rides from the given date on.
11. Drivers will be able to see the rides they have published (both those already made and those that are still pending) *<as software engineers you will evaluate the possibility of keeping historic data>*.
    11.1. Drivers will be able to enter the code given by passengers on the rides made that have pending payments (see the Ride Seat Booking Management environment).
       11.1.1. If the code entered is correct, the system confirms the journey made by the driver and the passenger (it initiates the transfer of passenger-driver credits, see Payment Management environment).
12. If a driver modifies a ride with booked seats, the system will check if the modification is feasible *<as software engineers you must define the criteria used to decide whether this is feasible>*.

13. Drivers can cancel rides (they can be completely cancelled or, in case they are recurring, just for one day).
    - 13.1. When a ride is cancelled, the blocked credits of the passengers are refunded (see Payment Management environment).

RIDE SEAT BOOKING

14. Users (hereinafter, **passengers**) will be able to see the available rides <*it is left to each team to decide if they must be authenticated or not in order to see the rides*>.
    - 14.1. To see the rides passengers must indicate the pick-up point, date and time, and the arrival point.
    - 14.2. In addition, they will be able to define filters for ride details such as smoking, type of vehicle, etc.
    - 14.3. [The system will show passengers the rides which match the selected criteria, showing a simplified travel information (in a mobile application there is not much space to show everything). This will also depend on whether you are authenticated or not.]
15. (Authenticated) passengers will be able to see the ride details.
16. Passengers can make seat bookings when they are seeing the ride details.
    - 16.1. For recurring journeys, the system will require passenger to confirm if the reservation is for a particular ride or it is intended to be a recurring booking (in this case the final date must be provided).
    - 16.2. The system will calculate the total cost of the booking and display it to the user.
    - 16.3. The system will verify that the user has the required credit in her virtual payment account.
        - 16.3.1. If there are sufficient credit, the system will block this amount of credit (see Payment Management environment).
            - 16.3.1.1. The system will send a reservation request to the driver.
            - 16.3.1.2. If the driver accepts the request, the system will generate a booking voucher for each ride, and it will associate the voucher with the itinerary and the passenger. Each document will have a unique code which will be only visible to the passenger.
            - 16.3.1.3. If the driver does not accept, the passenger credit that have been blocked for that ride will be unblocked.
        - 16.3.2. If there are not enough credits, the user cannot book the ride(s).
    - 16.4. The system will indicate the occupation of the booked seat(s) in the ride.
17. Passengers may cancel any reservations made before the beginning of the ride. <*It is left to the team's choice to establish penalties for cancellations at the time of commencement of the ride*>.
18. Passenger cannot reserve seats on different rides that overlap in time.
19. Passengers will be able to check the rides they have reserved (both those from the past and the ones for the future). <*The team will evaluate the possibility to include a reservations history or possible states of reservations*>.

PAYMENT MANAGEMENT

20. Registered users will have a virtual payment account (VPA) associated with their identity (profile).
21. VPAs will initially have a balance of 0 credits. One VPA credit is equivalent to €1. Credits can be divided into cents.
22. VPAs will manage available credits as follows:
    22.1. Available credits are the amount of credits that the user can use at a given moment.
    22.2. Blocked credits are user's credits reserved by the system as a guarantee of payment for booked rides. Blocked credits cannot be used by the user unless they are unlocked.
    22.3. Total credits are the total amount of credits associated with the account at a given moment (the sum of available + blocked credits).
23. Authenticated users can transfer credits to their VPA from their bank account through an operation by means of virtual POS (online payment).
    23.1. Transferred credits will be added to the available ones.
    [The team should not define how the payment between POS and the bank account works. With this purpose, the team will consider it as a module which will be called by indicating only the amount to be transferred and VPA to send the money to. The module will just indicate whether it has succeeded or not].
24. When users book a ride, the system will block at the VPA the number of credits equivalent to the cost of the ride.
25. Users will be able to request the reimbursement of all or part of their "available" credits in their VPA to a bank account.
    [As in the transfer of credits to the VPA, the team must consider that this is done by means of a call to a module to which the number of euros to be transferred and the bank account to which it will sent will be passed. The module will indicate if it has been successful or not.]
    25.1. In case of success, the system will deduct the reimbursed amount of available VPA credits.
26. When a driver confirms (see Tracking Subsystem) that he/she has made a ride with a passenger, the system will transfer to the driver's VPA the credits blocked in the passenger's VPA associated with the ride. The transferred credits will be added as available credits of the driver and will be deducted from the blocked credits of the passenger.
27. When a ride is cancelled, the corresponding blocked credits are refunded to its respective owners (unblocking them, subtracting them from the blocked credits and adding them to the available credits).

<Moreover, each team should mandatorily complete the following environments:

- *Notifications for actions for seat reservation, ride cancellation, etc. (Note: These are notifications within the application; they cannot be sent by mail or SMS).*

- *Use of Google Maps or Apple Maps whenever the team considers it is pertinent.*
- *Functionality to allow users see the profiles of other users as they see the ride offers that are published.*
- *Sending and receiving messages.>*

The division of the software system to be conceived into subsystems is as follows, with these four subsystems being mandatory:

- Users Management Subsystem (login, profile creation/modification, search for published rides and history, logout).
- Rides Publication Subsystem (ride publication for carpooling by the driver, modification of published rides, confirmation of rides, cancellation of rides).
- Ride Seat Booking Management Subsystem (selection of a ride by passengers, booking management by passengers, booking cancellation).
- Payment Management Subsystem (based on prepaid system).

On the one hand, the proposed solution should be based on this division. On the other hand, the interactive software system will comprise a highly usable web application for users of the service, which allows them to use a carpool service in the UAM. It will offer a mobile user interface with additional features for geolocation, which allow users to consult any available information from the web. In addition, its integration with Google Maps or Apple Maps will increase the application interactivity, offering a very usable interface for some services, such as departure and arrival point, and notification map, among others.

To review already existing proposals related to this technology, you should consult at least the following references, and especially the applications associated with them, must be consulted:

https://www.uam.es/uam/ecocampus

https://www.uam.es/uam/ecocampus/movilidad

http://compartir.org/

http://www.viajamosjuntos.com/

https://www.blablacar.es/

https://amovens.com/

https://www.uber.com/

These references should be extended, looking for information on how to manage carpool services in different rides and a couple of applications that implement it.

The elaboration of a **technical report** of the project is requested. This report shall:

1. Clearly describe the problem to solve, the need to be covered and the solution to be implemented. To do this, you must elicit and document software requirements to be created, analyse, and evaluate current technology and media (platforms, architecture, programming languages, etc.) to create this application.
2. Include all kinds of benefits to be obtained from the project.
3. Include mock-ups representing the design of the interaction, as well as the user interface of the proposed web information system for the mobile version.

The methodological procedure to prepare the proposed software system must include at least the following techniques:

1. Research on the Internet about other similar systems, analysing the functionality of these systems and describing them in a structured way by specifying the purpose of each functionality, the advantages and disadvantages of each software system that has been analysed and ideas drawn for the project through a comparative analysis among competitors. The functionalities of these systems that improve the proposed functionality or contribute as an extra functionality that is considered to provide an added value, shall be described as requirements of the proposed project.
2. Apply the brainstorming technique to generate novelty ideas about the ideal functionalities of the interactive information system the team shall propose in the call 2021-2022 of this competition. In addition, these functionalities must be grouped into subsystems.
3. Design the model using low-fidelity prototypes and/or tools for mock-ups development for the system proposed by the team.

The teamwork to be followed for the inception of this project must lead at least to the following deliverable specified in the Appendices of the technical report:

1. Development of documents that record the outputs of requirements elicitation techniques applied.
2. Minutes of team meetings.
3. Use of tools for the management of collaborative documents, by means either OneDrive, or Google Drive, to organize and conceive all the documentation about the software project collaboratively.

The report must conform to the following basic criteria, which shall be consider for its evaluation:

1. The report length must be between 25 and 35 pages.
2. It must include a cover page with company logo and information identifying the document: Project Name, Document Type, Document Identification, version and status of the document as well as the table revision control, including version, date, content or major modifications and purpose.

3. The report should follow the recommended structure in SEPRO subject during the class. **It is not intended to be exhaustive** in terms of the number of issues discussed. The quality of the argument will be assessed.
4. For the assignment, the quality of the technical report shall be considered from the consistency and coherence point of view.
5. You should cite in the references section other information sources from which data and information have been obtained (specialized magazines, Internet, etc.). These references should also be properly mentioned in the text of the report.
6. All appropriate techniques for information/knowledge acquisition and requirements elicitation should be used.

Appendix A provides a possible organization scheme for the final document.

Assessment Criteria

1. Document structure
2. Quality in drafting and formal presentation of the document
3. Consistency and coherence of arguments
4. Alignment with teacher recommendations
5. Use of information sources
6. Registry of results of the use of techniques for information/knowledge acquisition and requirements elicitation.

Due Date

SEPRO-2391: March 2nd, 2022.
SEPRO-2451&SEPRO-2392: March 3rd, 2022.

Tools for the Practical Assignment

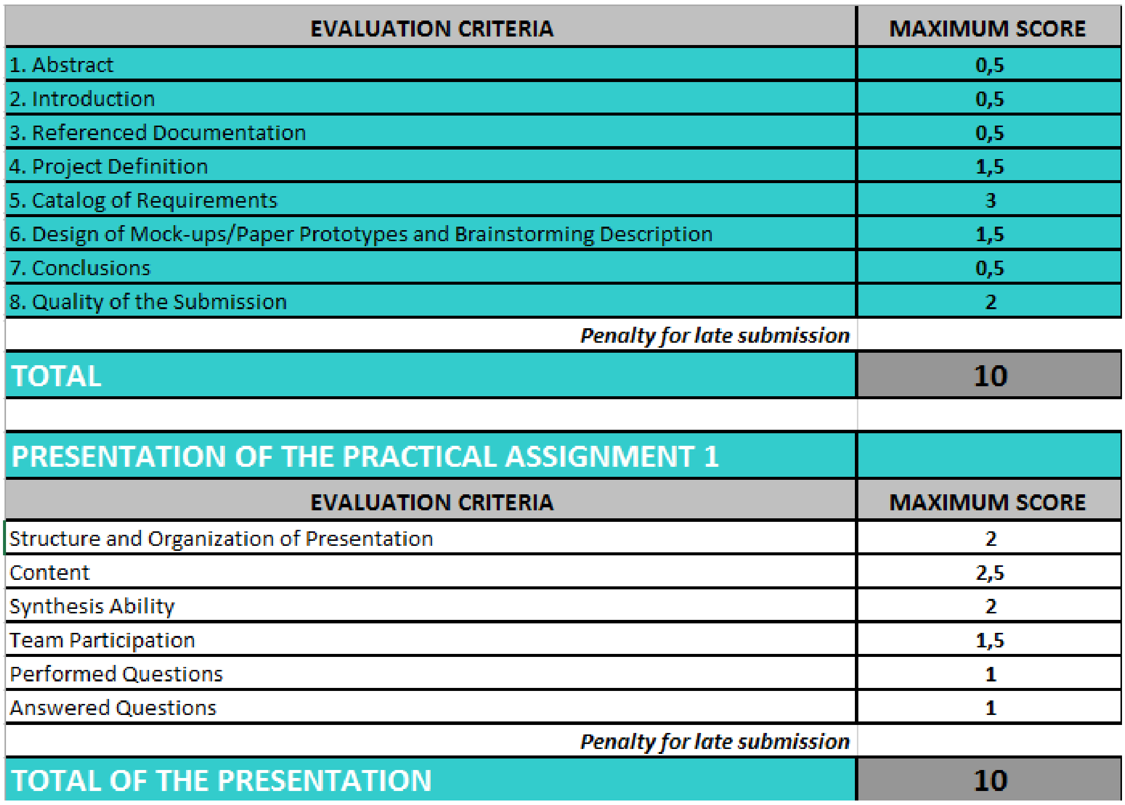
Any word processor and mock-ups development tools.

Deliverables

The technical report including the initial elicitation of software requirements and the mock-ups (low-fidelity prototype) representing the conceptual interaction design and graphical user interface, as well as the corresponding appendices for the outputs from requirements elicitation techniques and minutes from team meetings. The Reflection Document that corresponds to the performance of this Practical Assignment Nº 1 must also be delivered. These documents should be submitted through the Moodle for SEPRO assignments delivery.

Rubric

The evaluation criteria for this Practical Assignment No. 1 are listed in Table 1.



EVALUATION CRITERIA	MAXIMUM SCORE
1. Abstract	0,5
2. Introduction	0,5
3. Referenced Documentation	0,5
4. Project Definition	1,5
5. Catalog of Requirements	3
6. Design of Mock-ups/Paper Prototypes and Brainstorming Description	1,5
7. Conclusions	0,5
8. Quality of the Submission	2
Penalty for late submission	
TOTAL	10
PRESENTATION OF THE PRACTICAL ASSIGNMENT 1	
EVALUATION CRITERIA	MAXIMUM SCORE
Structure and Organization of Presentation	2
Content	2,5
Synthesis Ability	2
Team Participation	1,5
Performed Questions	1
Answered Questions	1
Penalty for late submission	
TOTAL OF THE PRESENTATION	10

Table 1. Rubric for Practical Assignment Nº 1

Marks

Marks shall be published four weeks after the submission date of the technical report.

Appendix A. Outline of Final Document

TECHNICAL REPORT OF THE INITIAL SOFTWARE REQUIREMENTS SPECIFICATION OF A SOFTWARE PROJECT

FRONT PAGE (cover page including the company logo and name, information identifying the document: project name, document type, identification, and version)

Change log table (including version, date, contents or main modifications and purpose)

Summary (Problem description and delimitation, document goals, public destination, project goals, beneficiaries of the project, system goals, functionality/services of the software system, system users, conclusions, i.e., project importance, contributions, and benefits). THE SUMMARY MUST BE SELF-CONTAINED.

CONTENTS

1. Introduction

Section (which can be split into subsections) composed by paragraphs with a coherent storyline. They must describe the following aspects:

- **Description of the problem. Delimitation of the problem**
- **Definition of the computable need of the group/s of people affected by the project (context, importance)**
- **Document goals, recipients**
- **Project goals, project beneficiaries, system goals, functionalities/services of the software system, users**
- **Methods used/procedure. Relevant antecedents for the report**
- **Document explanation**
- **Explanation of the logical structure of the document**

2. Project Definition

Paragraph with an introduction to the question. A comparison of existing work with the proposal can also be incorporated adequately here or in the introduction.

2.1. Goals and Functionality

Section (which can be split into subsections) composed by paragraphs with a coherent storyline. They must describe the following aspects:

- **Detailed project goals**
- **Scope**
- **Description of the application. Main objectives and scope**
- **Functionalities of the application (division of the system into subsystems and description of the tasks performed by each subsystem)**

2.2. Initial Catalogue Requirements

Paragraph with an introduction to the following questions.

2.2.1. Functional Requirements

- **Specification of the functional requirements of the application by subsystems, with labels.**

2.2.2. Non-Functional Requirements

- **Specification of the non-functional requirements of the application by type, with labels.**

3. System Interaction. Conceptual and Visual Design

Section (which can be split into subsections) composed by paragraphs with a coherent storyline. They must describe the following aspects:

- **Description of the concept of the system**
- **Description of the user interface (mock-ups)**
- **Description of the interaction of the user with the system (interaction/navigation through the windows)**

4. Conclusions

Section (which can be split into subsections) composed by paragraphs with a coherent storyline. They must describe the following aspects:

- **Global summary related to the introduction (SRS and design of the user interface and system interaction)**
- **Discussion of the proposed system**
- **Discussion of the contributions of the proposed system**
- **Benefits to be obtained**
- **Limitations of the proposal**
- **Future work**

References

[1] ……
[2] ……
:

Appendices

Appendix A. Title 1

Appendix B. Title 2

:

<u>Notes</u>: Keep a consistent format for the references. All the references cited in the text as [1], [2], etc. Figures and tables must be numbered, and they must have titles and references to them in the text. Appendices must be referenced in the text. Pages must be numbered. Orthography must be revised, as well as the document format.