UserUSERS MANAGEMENT

1. Only authenticated users will be able to access the application (except for the login

page).

* The page will have a login interface.
* To access the page users have to write username and password in the login interface.
* When the user writes username, password and other data fields correctly and clicks the login button the application will become available.

2. When users log in, the system will verify the access data (UAM email and

password) entered by them with the UAM authentication system.

1. Data written in the login interface (username, password and other data fields) will be processed by the user authentication system.

2.1. If the authentication is accepted, the system will assign a session to the user

(authenticated user).

2.1. If UAM authentication system returns a valid response a new user session will be created and the application will be shown to the user

2.2. <The working team must specify what happens if the authentication is not

accepted>.

2.2. If the authentication system returns a not valid response the login interface will show again with a Invalid Username or Password message.

3. When users log out, the system will delete their session (unauthenticated user).

Note: Do not delete the system profile.



* To log out, the user must click on the logout button. This button will only be displayed if the user is already signed in.
* The user session will end. The system will save all user data in case something has changed and then it will delete the current session.
* The application will stop being available.
* The login interface will show up for a new user to login.

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).]

* If the user hasn’t used the application before, a new user profile with the user data will be created when the login button is pressed.
* A new virtual payment account will be created by the system with basic payment information of the user

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>.

5.

* To modify the user profile data, the user must press the settings button then press their profile picture and their profile page will show, from there, press the edit profile button and the edit profile interface will show.
* A new interface will be shown with the current profile data in different fields. Fields available will be name, age, profile (student, faculty or staff), residence, place of study/work
* Users will be able to change the profile data by clicking on the different fields and entering the new data.
* When profile data is changed, the save changes button must be pressed to save the profile data in the application database.

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.

6. Authenticated users can see a list with all published and reserved rides (by the current user). Rides will be organized on a historical order on the list. Payments of each ride will be shown next to each ride data.

PUBLICATION OF RIDES

7. Authenticated users will be able to publish ride for carpooling (from now on,

drivers):

* In the drivers main page there will be a button to create a ride once the button is pressed the ride creation page will be displayed

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.).

* The ride creation page will contain all the fields needed to create the ride
* Once the user fills in all the ride details, the publish ride button will activate and he will be able to publish his ride.
* The ride then will get stored on the external rides database

7.2. Optionally, users can indicate whether it is an occasional or recurring ride.

* In the end of the ride creation page, the user will find a slider indicating if the ride is recurring or not.

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).

* Once the ride is marked as recurring, a button for each day of the week will be shown and the user will be able to select the days of the week he is going to make this ride (we assume that the recurrence of the rides are weekly and that if there are some specific days the user will not make the ride he will cancel those days manually later)
* Also, when marked as recurring, a date will be asked to the user indicating when will he stop making the ride

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).

* The application will store in the database the ride and will later allow the user to see the rides that he has taken and published.

8. The system will consider each repetition as an independent ride.

* When a ride that is marked as frequent is published, the application will publish each of the rides one by one and treat them independently.

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).

* In the driver main page he will have to press the list button, then the my rides button.
* In the my rides page you can access the main details of each ride. When a ride is pressed the ride's complete details are shown, and if the ride is published by that user, he can change the details there and then press the button to save changes.
* If the changes are valid, a page showing that the changes have been made shows, but if they are not, a n error page is shown.

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.

* Following the steps on the previous description the driver can access the ride modification page where if no passenger has reserved his spot, the driver can make any change he wants.
* Once the change is made, if the ride is recurring, a pop up will show that will ask the driver if the change should be applied to all the rides or just the single one edited.

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>.

* In the driver main page he can press the list button that will take him to his rides page.
* In that page he will find two tabs, past and future rides. By pressing on the one he wants to see he will be able to visualize the past or the future rides he has been on, will be on or published.

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).

* Once the ride time is past, the passengers will be able to see a pop up with a code once they access their current ride.
* Once the time of the ride is past, the driver will have an option to input the passengers codes to confirm their ride. From the details of the current ride accessed in the rides page, he will find an option that says scan confirmation codes that will allow him to input the passengers codes in the text field and the system will confirm the payment from the passenger to the driver.

13. Drivers can cancel rides (they can be completely canceled or, in case they are

recurring, just for one day).

* The user would have to enter in the my rides page, then he would have to enter the details of a ride that he has published, from there a cancel ride button will show, and when it is pressed on a recurring ride, a message asking to apply the cancelation for one ride or for all will show

13.1. When a ride is canceled, the blocked credits of the passengers are refunded

(see Payment Management environment).

* Once the ride is canceled by the driver, the system will automatically refund the passenger’s credits and the ride will appear on the system as canceled and never done.

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>.

The system will show a button for the user to interact and will display all available rides.

* The available rides button will show on the main screen.
* For users to see the rides they will need to be authenticated.

14.1. To see the rides passengers must indicate the pick-up point, date and time,

and the arrival point.

In the rides display screen a filter interaction system to find rides, with a field to choose date, time origin and arrival.

* To change the filters the user will need to press the filter screen.
* Once these filters are used the system will only display rides that satisfy the search.

14.2. In addition, they will be able to define filters for ride details such as smoking,

type of vehicle, etc.

The system will also show filters for details such as smoke and car type.

* Users need to be logged in and have accessed the rides page.
* User will use search filters to change pick-up point, date and time,

and the arrival point. At the same time the user can modify these details filters.

* The application searches for the rides that satisfy the conditions.

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.]

When filters are applied the system will display all the results that match the specified criteria.

* Users need to be authenticated.
* Application search for rides that satisfy the search.
* Users see the available rides displayed on the screen with the corresponding drivers dates and times.

15. (Authenticated) passengers will be able to see the ride details.

When the user is authenticated the system will display the option to see the ride detail of a specific ride the user selects on the rides menu.

* The user needs to be authenticated.
* The user has to enter the rides menu. The system will display the available rides.
* The user must press a ride to open the rides detail menu.
* The system will access the database and show all the ride information to the user.

16. Passengers can make seat bookings when they are seeing the ride details.

If a user is on a ride details screen a button can be pressed to book a seat on the desired car. The system will add the passenger to the respective ride.

* The user needs to be authenticated.
* The user will need to access a ride detail screen (as in the previous requirement)
* The system will show the user a button to book a seat. When the button is pressed, the system will access the database to add the user to the ride.

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).

When the book seat button is pressed by the user a pop up message will be shown asking if the ride is recurrent. If so, the system will repeat the process for all the recurrent rides of that trip.

16.2. The system will calculate the total cost of the booking and display it to the

user.

The system will calculate and show the user the total cost of the book.

* A user that has been previously authenticated
* The user is in the process of booking a ride.
* 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.

The system will access the database to find the user balance and check if the user booking has enough credits to fulfill the book cost.

16.3.1. If there are sufficient credit, the system will block this amount of

credit (see Payment Management environment).

If the user has enough credits to pay the book, the system will change the user balance account in the database, where the cost in credits will change from available credit to blocked credit.

16.3.1.1. The system will send a reservation request to the driver.

After making a successful booking by a user, the system will show a reservation request on the request menu of the driver client of the application.

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.

If a reservation request is done, it will show to the driver application.

* A request for a ride has been made by a user.
* The system shows the rider a new request popup on the request menu icon in the main page.
* The driver enters the request menu and the system shows all reservation requests pending for the driver.
* The driver accepts the request. The system adds the user to the corresponding rides in the database.
* The system generates a booking voucher with a unique code and sends it 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.

If a reservation request is done, it will show to the driver application.

* A request for a ride has been made by a user.
* The system shows the rider a new request popup on the request menu icon in the main page.
* The driver enters the request menu and the system shows all reservation requests pending for the driver.
* The driver rejects the request. The system removes the request and unblocks the previously blocked money of the passenger in the database.

16.3.2. If there are not enough credits, the user cannot book the ride(s).

The user is authenticated

* The user access a ride detail page
* The user tries to book a ride with the book ride button.
* The system checks in the database if there is enough credit.
* If there is not the system would not allow the booking. Feedback of the cancelation of the process is shown in the user interface.

16.4. The system will indicate the occupation of the booked seat(s) in the ride.

When a user sees a ride detail screen of a specific ride, the occupation of the seats will be shown.

* When a book is validated, the system will add the user to the ride.
* The system will check for the number of users of a ride on the database and will display them showing some visual feedback on how many seats are left on the ride detail page.

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>.

A passenger can press the cancel button on the details of the my rides page before the beginning of the ride.

* The user accesses the my rides button in the main page.
* The system shows all current rides booked or done by the user.
* The user selects the ride desired to cancel. The system displays the ride details with a cancel button. The user can press the cancel button.
* The system will access the database to change the user balance and the available seats.
* If the cancellation has been made within 24-3 hours before the ride a 30% of the price would be not returned.
* If the cancellation has been made within 3-0 hours before the ride there would not be a refund.

18. Passengers cannot reserve seats on different rides that overlap in time.

* A passenger reserves a seat on one ride at a specific hour.
* The passenger tries to reserve another seat on another ride in that same hour.
* The system cancels the operation and shows to the user some feedback response indicating why the procedure could not be achieved.

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>.

* User is authenticated.
* The user presses the my rides button in the main page.
* The system displays past history of rides. A button is also displayed to change to the future history rides.
* The system will display an icon next to each future ride showing the state of the book (pending, accepted or rejected)

PAYMENT MANAGEMENT

20. Registered users will have a virtual payment account (VPA) associated with their

identity (profile).

* Registered Users have a profile section with the VPA settings menu
* The sub-section has a form for non configured accounts
  + The form asks for the credit card number, expiration date, card pin
  + The user has to validate its identity
  + The card remains set for future payments in time
* Accounts with already configured VPA will have an option to add more cards

21. VPAs will initially have a balance of 0 credits. One VPA credit is equivalent to €1.

Credits can be divided into cents.

* The User has a balance section on his profile
* The current credit is displayed in units and 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).

* The user has a “Deposit” option on its balance section
* The user enters the desired amount to add to the account
* The system proceeds with the POS tool

23.1. Transferred credits will be added to the available ones.

* The new credit is computed and displayed

[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.

* The user has enough credits to book a ride
* The system checks that the credits required are sufficient
* The credits are blocked at the user VPA
* The blocked credits are reflected in the user balance section

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 be sent will be passed. The module will indicate if it

has been successful or not.]

* The user has a “Withdraw” option on its balance section
* The user enters the desired amount to subtract from the account
* The system proceeds with the reimbursement module

25.1. In case of success, the system will deduct the reimbursed amount of available

VPA credits.

* The new credit is computed and displayed

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.

* The driver confirms a drive with a specific passenger is over
* Credits blocked from the passenger VPA are transferred to the drivers VPA
* New balances are computed in both accounts

27. When a ride is canceled, 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)

* The driver cancels a ride
  + Some passenger cancels a certain ride
  + The passenger specifies the reasons to cancel the ride
* The credits are unblocked and remain in the passengers VPA
* The new balance is re-computed
* Drivers can change the details of the rides or cancel the ride. For changes that affect the ride such as changing the date, the driver can make a petition to change the time and the users will get a notification where they will have to vote if they agree with the change, if all the passengers agree, the change is made, if not, the passengers who do not agree can cancel their spot in the ride without any penalization.
* Once the ride is canceled by the driver, the system will automatically refund the passenger’s credits and the ride will appear on the system as canceled and never done.

29-**Notifications for ride seat booking**

* 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).
* A request for a ride has been made by a user.
* The driver accepts the request.

| Functional Requirements | Non-Functional Requirements |
| --- | --- |
| 1.  2.  3.  4.  6.  7.  8.  9.  10.  12 killed  13.  21  22  24  26  14.  15.  16.  18  19 | 5  11  17  20 - Security/Bank Conventions  23 - Security/Transactions  25 - Security/Transactions  27 - Reliability/Response to Errors |