Project: Architecture for Cultural Heritage Management System

Version	Deliverable phase II

Deliverable Template

Date	
Team Name/ID	QDTeam

Team Members		
Name and Surname	**Matriculation number**	**E-mail address**
~~Andrea Avallone~~	~~*N/A*~~	~~*andrea.avallone@student.univaq.it*~~
Daniele Fossemò	*280966*	*daniele.fossemo@student.univaq.it*
Fabio Capitanio	*268133*	*fabio.capitanio@student.univaq.it*
Gianluca Rea	*278722*	*gianluca.rea@student.univaq.it*

Table of Contents

Challenges/Risk Analysis

Risk	Date the risk is identified	Date the risk is resolved	Explanation on how the risk has been managed
technology to implement for localization	29/11/2021	2/12/2021	The team has considered different solutions in order to find the one which is most cost-effective. The proposed solution uses UWB indoor localization technologies.
which kind of app implementation	29/11/2021	2/12/2021	Web-app implementation in order to have a compromise on accessibility and usability of the system, focused on 2 points: the possibility of cross-platform, which in this way is ensured with less effort, and the consistent choice of a single work framework, without the need to adapt multiple technologies to communicate with each other to allow different mobile apps.
lack of knowledge on a specific domain (IoT and indoor localization technologies)	29/11/2021	never end	study the different solutions used on this kind of problem and go into detail of chosen technologies relying also on academic articles [1], [2]
Data Privacy and treatment of user's information	2/01/2022	20/01/2022	the only sensitive information of the user managed by the system is the Email address, necessary both as a contact tool and as a support and assistance tool. During the visit, each user will have a unique token, which will not allow the location system to directly access user data.

Requirements Refinement

For each requirement (both functional and non-functional) is defined the degree of priority (1-5), so how much the fulfilment of the requirement is important to let the system work properly in accord with his goal and constraints; and the degree of difficulty (1-5), so how much the fulfilment of the requirement will cost in terms of time and money for his implementation.

Functional Requirements

1. **FR1 - Visitor and artworks localization** - priority: 5, difficulty: 5
   The system should be able to localise both the visitors in a near-real-time way and the artworks in the museum. This is in order to:
   1.1. Provide navigation support for visitors during their visitation.
   1.2. Give details of the artworks to the visitors near to them.
   1.3. Allow visitors to locate and save different points of interest using a dedicated search engine system (eg artworks).
   1.4. Get statistics data of the single visitor (eg: preferences, time spent on different artwork, etc) to allow the system to generate recommended visit inter-museum routes and suggested artworks (in the same or nearby museum).
   1.5. Get statistics data of the visitors, such as crowd movement (in a continuous way), number of people visiting, time spent on average by people on different artworks, etc.
   1.6. Get statistics data of the artworks, such as rating, how much content is created on that artworks, etc.

2. **FR2 - Support to administration**- priority: 4, difficulty: 3
   The system should provide an interface to the administration team in order to:
   2.1. Provide the CRUD function on the artworks, the IoT components in the system (and configure them), the users registered to the system, the time slots for booking of tickets and the number of tickets available per day.
   2.2. Provide a page with the statistics data of crowd, booked tickets and artworks.

3. **FR3 - Booking system** - priority: 4, difficulty: 2
   The system should allow visitors to book tickets on a dedicated webpage. In particular in this webpage:
   3.1. Is possible to buy a ticket for a certain day and contextually indicate one of the time slots defined by the administration as the one in which is intended to make the visit.
   3.2. Is possible to see how much the museum is occupied on a certain day and in the specific time slots.
   3.3. Is possible to see how many tickets are available on a certain day. The number of tickets available in a museum in a day is defined by the administration team which chooses this quantity in accordance with statistical data they own.
   Furthermore:
   3.4. If a visitor who has booked the ticket tries to enter when the museum is filled (in the museum are allowed a maximum number of 500 visitors), he will be asked to wait until a place becomes free.
   3.5. If a visitor doesn't visit the museum on the day he has booked the ticket (for any reason), can request a refund of the ticket.
   Also, the system should provide a ticket office at the entrance for visitors who can't buy

tickets online. In order to give some advantages to visitors who book the ticket online, the system should:

3.6. divide the queue for buying tickets to the ticket office from the queue for the validation of tickets. So the visitors who book the ticket online can go directly to the second queue while the others have to do also the queue to buy the ticket

3.7. consider the ticket bought at the ticket office as tickets which can't be refunded.

4. **FR4 - Personal content** - priority: 3, difficulty: 3
The system should allow the visitors to generate personal content and share them with the community. In particular the system:

4.1. Should have a space that allows visitors to rate both the museum and the artworks individually, with a score from 1 to 5 and optionally with text, photos and/or videos.

4.2. Should show, in addition to the information of the museum and the artworks, the respective content generated by other visitors.

4.3. Also, the visitors should be able to create personal inter-museum routes and share them in a dedicated public space where they can be rated by other visitors.

5. **FR5 - Other services** - priority: 1, difficulty: 3
The system should be able to provide the following services:

5.1. generate recommended visit inter-museum routes and suggest artworks to the visitors based on their preferences

5.2. generate recommendations on the nearby museum to visit in case the museum is full

5.3. gather weather conditions

Non-functional Requirements

1. **NFR1 - Usability** - priority: 5, difficulty: 5
The system is designed in order to give a full-immersive experience to the visitors. For this reason, it's important that the web app is easy to use and the overall experience (booking, authentication, visitation, localization of artworks and generation and reading of content) feels smooth and satisfying to the visitors.

2. **NFR2 - Security** - priority: 5, difficulty: 1
Due to GDPR, the system should not store any sensitive data related to the visitor. Also, the storage system should guarantee a high degree of security. Finally, the museum has to tutelage itself from the theft of IoT devices by malicious visitors.

3. **NFR3 - Reliability** - priority: 4, difficulty: 5
For the same reason as the first point, the system should be able to guarantee a low failure probability, which means the IoT devices should work almost without failures with a maximum load of 500 people in the museum. If the system fails, there won't be a loss of people nor of money (except for booking in case of a refund), but there will be a temporary disservice, which must be resolved as fast as possible by administration remotely or physically based on the nature of the problem (software or hardware).
To lower the probability of failure, the system should provide different ways to make things right, such as augmentative IoT devices or changing settings about the booking system.

4. **NFR4 - Performance** - Priority: 4, difficulty: 3
The system should be able to localise the artworks and the visitors with high accuracy and

precision. Also, the system should be able to authenticate the visitors in a max of 1 second.

5. **NFR5 - Accessibility** - priority: 3, difficulty: 2
   The system should be able to give access to a set of basic features to the visitors that don't have smartphones (eg: booking and authentication). Also, these visitors should be localised by the system (anonymously) for the sake of crowd monitoring by the administration.

Informal Description of your system and its Software/System Architecture

Description of the system

The system tries to meet most of all the two main requests of giving an immersive experience for visitors and giving an efficient way to the administration team both to manage statistical data of museums, artworks and visitors and to monitor in a near-real-time the crowd in each museum.
To achieve these two interlaced results the system tries to achieve the following target:
Retrieve data with localization of visitor and artworks, process and use them both to improve the user's experience and to provide statistical data to the administration team; guarantee the management operations of museum and artworks in the system, a booking service and ad-hoc recommendation for visitors using machine learning techniques.

Sub-systems

The system is logically divided into three main groups:

1. **Localization**
   This part is responsible for the high-accuracy localization of visitors and artworks. To achieve these results are used the following components:
   1.1. **Ultra-Wideband indoor localization technologies (UWB):**
   To localise visitors, the system uses the UWB technologies, which consist of a set of trackers plugged in the power source available in the museum and a set of receivers (tag). The Tags are divided into two typologies, the first labelled as "visitor-dedicated" (VD) and the second labelled as "artwork-dedicated" (AD).
   The first is given to the visitors at the entrance (after ticket validation) by the member staff (which also have to occupy about their return back and about maintenance of both AD and VD tags) and are associated to visitors account through ticket ID (so that notification can appear on visitor's smartphone without a direct association from smartphone to tag). In particular, the association work in three phases:
   - the visitor validate the ticket
   - before/during/just after the ticket validation the member staff activate the tag (with eventually an apposite QRcode attached to the tag)
   - The event of the activation of the tag and the event of the validation of the ticket are caught by the system which pairs the ID of the tag with the ID of the user owner of the ticket (through the ticked ID).

   The second is associated with the artworks (the associations are stored in the Database component). When a VD tag is near an AD tag, then an event will be caught and managed by the message broker component.
   1.2. **People counter**:
   To monitor the crowd in the museum and check in every moment that not more than 500 visitors are inside the museum, at the entrance and the exit of the museum there will be placed people counters. In addition, at the entrance, there will also be a QR reader which is used to authenticate the booked tickets.
   1.3. **QR code reader**:
   To authenticate visitors in a fast and easy way the system uses QR code readers at the entrance. The visitors who have booked the ticket with the web app have just to use

the QR code to authenticate the ticket. Those who don't before have to buy the ticket at the ticket office and then validate it at the QR code reader.

2. **Cloud**

This part is responsible for giving all the services about storage and process of data using a cloud services provider. The data are stored and processed to give an immersive experience to visitors and provide statistical data to the administration team:

2.1. **Database:**
The data are given from the localization subsystem and the back office component are stored in a cloud database.

2.2. **data processing engine:**
The stored data are processed by programs running in external machines for the fulfilment of general-purpose functions.

2.3. **Message broker:**
A publish/subscribe broker running in external machines that catch events raised by the localization subsystem and is responsible for the streaming of data flow.

2.4. **Machine learning model:**
The statistical data collected from visitors and stored in the cloud database are used by a validated model which uses machine learning techniques to give recommendations to visitors during their visitations, such as the suggestion of artworks (in the same or nearby museum) or the generation of recommended inter-museum visit routes in accord of the user preferences.

3. **Web app**

This part is responsible to provide the services of the system both to visitors and the administration team:

3.1. **Visitor account:**
The visitor can register to the app to get all the services the system previously described during their visitation and to get access to other services like those provided by the following two components.

3.2. **Booking page**:
The tickets can be booked online by visitors registered to the web app. Contextually to the buying, the visitor can see statistics about the influx during the day they are interested, eventually divided into time slots. The tickets will be provided with a QR code which will be authenticated at the entrance with the appropriate QR code readers component.
For visitors who don't book tickets online(for any reason, such as tourists that go around through the city and have decided at the last moment to visit the museum but don't have any device to book tickets), there is the ticket office at the entrance, in which a member staff sells the tickets through a member-staff dedicated booking page. The queue for the ticket office is divided from the queue for the QR code readers and the tickets sold by the ticket office aren't refundable.

3.3. **Community space**:
The visitors have access to a space dedicated to community content in which they can enjoy content created by other visitors and create content themself. Also, this content can provide feedback (in the shape of statistical data) to the administration team because of the rating system.

3.4. **Back office:**
The administration team uses the reserved back office both to do the CRUD function

they need for museums, artworks, tickets and booking data (eg: time slots) and to monitor the crowd and look at statistical data.

Architectural patterns

The architectural patterns used by the system are

1. **Publish/Subscribe:**
   The system reacts by the message broker component to the events produced by the visitors during their visitation, to give them the services linked to the localization, such as the providing of details of artworks near to them.

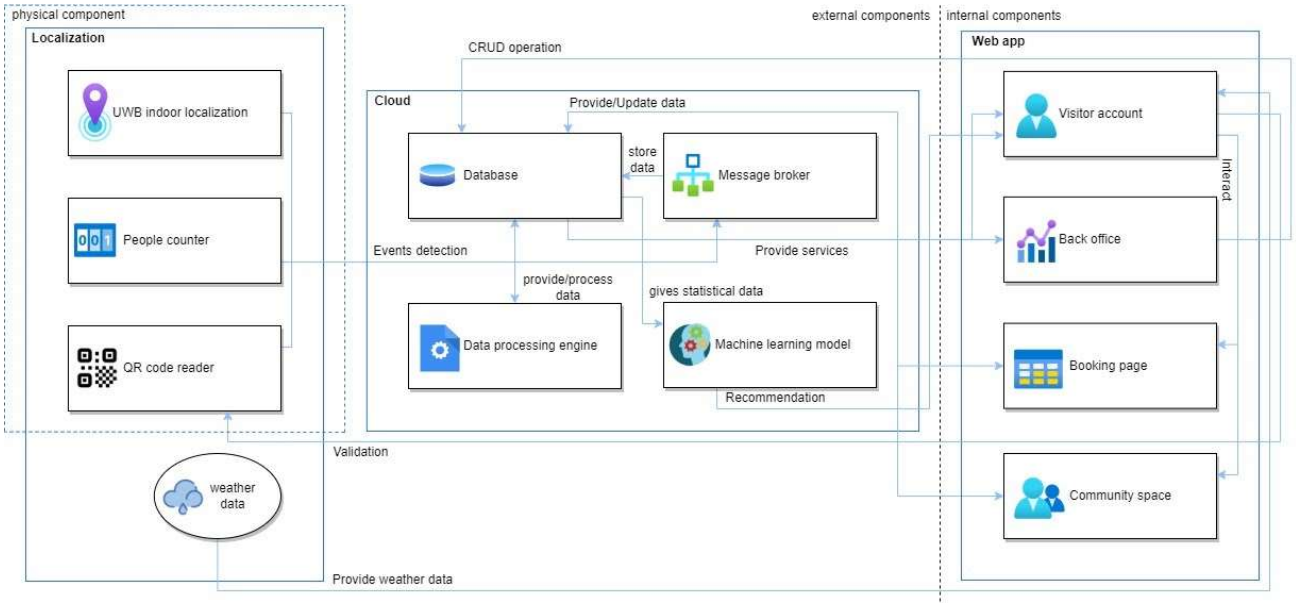
2. **Client-server**:
   The web app subsystem is structured with a request/response system, in which the visitors send requests through their visitor account to get the services provided by the system and the administration team sends requests through the back office to get access to the function of CRUD, monitoring the crowd and looking at statistical data.

3. **Cloud computing**:
   The system relies on a cloud service for data storage and fulfilment of data management and processing, as defined by the cloud component.

System boundaries

In the architecture defined until now is possible to divide the components by those that use externally provided technology integrated into our system, and by others that are implemented internally. Generally, the localization subsystem uses components provided externally. For example, consider the UWB indoor localization component: It's not just the hardware used for the localization, but also the software which uses the techniques based on distances and angle degree of the tag to the receivers. This component is considered as a black-box which gives the system the coordinates in near real-time of the visitors. The same reasoning can be done for the QR code reader, the people counter and the weather data which are provided by an external service and integrated into our system.
Finally, also the Cloud component is provided by an external service.

Overall system architecture



How does the system satisfy the requirements?

The following list is showed how the system tries to satisfy both the functional and the non-functional requirements:

1. **Functional Requirements**
   - 1.1. **FR1: Visitor and artworks localization:**
     This requirement is fulfilled by the Localization sub-system, the message broker and the cloud components. The UWB indoor localization guarantees the near-real-time localization of the visitors, the message broker detects the event of interest, and the cloud component processes them to give services like the giving of detail of artwork near to visitors. Also, the UWB indoor localization statistical data are stored in the cloud storage and services such as the save/search of artworks of interest are provided by a dedicated space in the visitor account; while the recommendation is given by the machine learning model component.
   - 1.2. **FR2: Administration support:**
     The functions needed by the administration such as the CRUD operation, the monitor

of the crow and the view of statistical data is satisfied by the back office dedicated component.

1.3. **FR3: Booking system:**
The functions described in these requirements are provided by the Web app subsystem, specifically by the interaction of the visitor account and the booking page components. Also, the number of tickets and the time slots are defined by the administration team with the back-office component.

1.4. **FR4: Personal Content:**
The functions described in these requirements are provided by the Web app subsystem, specifically by the interaction of the visitor account and the community space components.

1.5. **FR5: Other services**:
The other services are provided by the machine learning model component (recommendation), the interaction of visitor and community space components (generate inter-museum visit routes and suggest artwork) and the external weather data provider which is displayed in the web app subsystem.

2. **Non-functional requirements**

2.1. **NFR1: Usability:**
The UWB indoor localization component gives the visitor an easy way to get access to all the functions of the museum, in fact, they only have to validate the ticket, get the tag given by the member staff and use the web app with a personal account to enjoy the visit.
In addition, the booking page component and the QR authentication give the visitor a fast way to access the museum, and finally, the experience is improved by the community space component.
The way in which the tags are associated with the users shouldn’t create waiting time (or at least it should be very little), because the tag can be activated before the ticket validation (the system, once caught the event of tag activation or ticket validation will wait for the other one. If more than one tag is activated they are queued by the system in order to pair to a user ID through ticket validation).

2.2. **Security:**
The visitor account component doesn't need the personal information of the visitor (just email, username and password, and to other users in the community space is shown only username), also the tag of UWB component gives only information about user position. Also, the tags are associated with a user ID, so in case of theft (which gets out of control of member staff assigned to return back of tags), the administration team can request the police to go back to the concerned visitor with the given email. Finally, the cloud service is chosen according to the fulfilment of this requirement.

2.3. **Reliability:**
The UWB indoor localization component uses advanced technologies with a simple implementation which represent one of the most cost/effective solutions for indoor localization. The fault probability of this system should be at least lower than almost every other indoor localization technology.

2.4. **Performance:**
The UWB indoor localization component also uses technologies with better accuracy results. Finally, the authentication is provided by the QR code reader, which from the moment the QR code is shown to the reader properly, needs at least 1 second for the authentication process.

2.5. **Accessibility:**

The web page subsystem guarantees to all visitors the function of booking both with a smartphone, laptop and personal computer. To those who don't use these instruments, there is a ticket office in the entrance, which acts as a booking page component. The authentication is made by a QR authenticator, so those who don't have a smartphone can print the QR code previously to the visit.
Finally, the function of localization is provided by the tag, so the administration can monitor the crow at any moment, both to those who have a smartphone and to those who have not.

Design Decision

Localization technologies

Concern (Identifier: Description)		*Con#1: Which kind of technologies should be used for the localization of visitors and artworks?*
Ranking criteria (Identifier: Name)		*Cr#1: Usability* *Cr#2: Performance* *Cr#3: maintainability* *Cr#4: Cost*
Options	**Identifier: Name**	*Con#1-Opt#1 : Smartphone localization*
	Description	*The system uses a wifi indoor triangolation for the localization of visitors, and reacts when the them are near to the artworks.*
	Status	*This option is rejected*
	Relationship(s)	
	Evaluation	*Cr#1: For visitors this option it's easy to use, but it works only if they have a smartphone which is connected to the wifi of museum and has enough battery.* *Cr#2: The wifi triangulation has not a satisfable precision* *Cr#3: This option should be easy to implement and mantain.* *Cr#4: This option should have a medium cost*
	Rationale of decision	*This option could fail on the attempt of satisfy the Cr#1 because of the visitor's smartphone conditions, and does not satisfy the Cr#2*
	Identifier: Name	*Con#1-Opt#2 : BLE Beacons*
	Description	*The system uses the bluethoot communication of the smartphones to identify and track the visitors*
	Status	*This option is rejected*
	Relationship(s)	
	Evaluation	*Cr#1: For visitors this option it's easy to use, but it works only if they have a smartphone which have bluethoot active and has enough battery.* *Cr#2: On the range of the receiver station, the percision is acceptable* *Cr#3: The implementation of this choice involves a complete setup of an infrastructure of receiving stations, and the logical implementation with the system* *Cr#4: This option should have a medium cost*
	Rationale of decision	*Same as for Opt#1, this option could fail on the attempt of satisfy the Cr#1 and does not satisfy the Cr#2*
	Identifier: Name	*Con#1-Opt#3 : UWB indoor localization only on visitors*
	Description	*The system uses the ultra windband indoor localization technologies which consist in a set of trakers and a set of tag. The tag are continuosly localazied by the trakers and will be provided to the visitors. Instead for artworks there will be a localization based on a modelization of museum, which is divided into plans and rooms. Each room is divided in areas, which are squares determined by coordinates defined by the administration team. So, the artworks (and the area in which the administration team wants the tag of visitors arises up events) is localized by the area in which it's inside.*
	Status	*This option is rejected*
	Relationship(s)	
	Evaluation	*Cr#1: For visitors this option it's easy to use. Furthermore the visitors will be localized also if they don't have a smartphone (or it's out of energy), so the administration can monitor the crowd at any moment. Instead the artwork localization system may be of difficult use for the administration team.* *Cr#2: This UWB indoor localization technologies have an high accuracy, in order of meters or even centimeters. instead the artwork localization system may be have a certain degree of approssimation, defined by the use made by the administration team.* *Cr#3: The implementation may be hard, but the hardware have a good durability and also the tag are responsible only for visitors localization.* *Cr#4: this option may have an high cost, but the artworks localization system should contain them.*
	Rationale of decision	*This option does not completely satisfy any of the criteria, but only partially.*
	Identifier: Name	*Con#1-Opt#4 : UWB indoor localization to visitors and artworks*
	Description	*The system uses the ultra windband indoor localization technologies which consist in a set of trakers and a set of tag. The tag are continuosly localazied by the trakers and are divided in two kind of tag, the first given to visitors (which can assiciate them to their smartphone) and the second associated to the artworks. when a tag associated to the visitors is near to one associated to an artworks, they arise up an event that will be managed by the system.*
	Status	*This option is decided*
	Relationship(s)	
	Evaluation	*Cr#1: For visitors this option it's easy to use. Furthermore the visitors will be localized also if they don't have a smartphone (or it's out of energy), so the administration can monitor the crowd at any moment.* *Cr#2: This UWB indoor localization technologies have an high accuracy, in order of meters or even centimeters.* *Cr#3: The implementation may be hard, but the hardware have a good durability.* *Cr#4: this option have an high cost.*
	Rationale of decision	*This option satisfy the first two criteria, partially the third and don't satisfy the last. However this is the only solution that satisfy the most prioritized criteria (the first two), so it's the one taken for the system*

App implementation

Concern (Identifier: Description)		*Con#2: Which type of app implementation choose?*
Ranking criteria (Identifier: Name)		*Cr#1: Usability* *Cr#2: Accessibility* *Cr#3: Efficiency* *Cr#4: Time to production*
Options	***Identifier: Name***	*Con#2-Opt#1 : smartphone app and dedicated web page*
	Description	*The system provide both an app dedicated for smartphones which has all the services described in the system and a web page with just basic services, so login, details of museums, booking, and access to the community space.*
	Status	*This option is rejected*
	Relationship(s)	
	Evaluation	*Cr#1: This option should satisfy this criteria due to the fact that advanced services (localization support, creation of personal content, etc.) are developed specifically for smartphone enviorment. Also the web page should provide an easy way to get access to base services.* *Cr#2: This option should be accessible by almost the majority of people thanks to implementation in both smartphone and web scenarios* *Cr#3: This option should be satisfy for the same reason of Cr#1* *Cr#4: Due to the fact that the system should provide a dedicated app for the smartphones (Android, IoS, etc) and a dedicated web page, this option need a very large amount of time.*
	Rationale of decision	*This option is rejected because of the complexity of his implementation, which enlarge time of production (and relatevely costs). In addition the Con#2-Opt#2 is considered a better cost-effective option than this.*
	Identifier: Name	*Con#2-Opt#2: webapp*
	Description	*The system provide only a webapp with both the basic and the advanced services.*
	Status	*This option is decided.*
	Relationship(s)	
	Evaluation	*Cr#1: The web app is developed giving the priority on smartphone scenario, in order to give the most immersive experience during the visitation, so this criteria should be preserved.* *Cr#2: This option should be accessible by almost the majority of people thanks to the fact it consider both the smartphone and the web scenarios.* *Cr#3: This option should be satisfy for the same reason of Cr#1, but less than the #Con2-Opt#1 (which provide a smarphone dedicated app) in some of advanced services.* *Cr#4: This option should request a reasonable time to production.*
	Rationale of decision	*Even if the satisfability of third criteria get more difficulty, this option should be the better cost-effective option for this concern. Also the first two criteria should be covered, so the system proceed on this option.*

Data management system

Concern (Identifier: Description)		*Con#3: which kind of data management system choose?*
Ranking criteria (Identifier: Name)		*Cr#1:Performance* *Cr#2:Security* *Cr#3:Reliability* *Cr#4:Cost* *Cr#5:Migration*
Options	***Identifier: Name***	*Con#3-Opt#1: Internal MySQL server*
	Description	*The system use an internal MySQL server deployed by a database developer*
	Status	*This option is rejected.*
	Relationship(s)	
	Evaluation	*Cr#1: Performances depends on implementation by the database developer and the current hardware.* *Cr#2: The safety part must be managed internally, applying the appropriate security procedures* *Cr#3: the reliability of the system must be optimized by carrying out good system maintenance and having good fault tolerance mechanics* *Cr#4: The cost of the system varies according to the priority given to the criteria described above.* *Cr#5: The system migration is managed internally, so it is adaptable to any situation.*
	Rationale of decision	*This option has been rejected because the satisfability of the first three criteria depend much on how is effectively implemented. Due to the fact that this choose is very important for the overall system, the option is rejected in order avoid that risk.*
	Identifier: Name	*Con#3-Opt2: Cloud services*
	Description	*The system use an external MySQL and NOSQL server by a provider of cloud services.*
	Status	*This option is decided.*
	Relationship(s)	
	Evaluation	*Cr#1: The cloud management of resources enable to use the necessary power load to have a performance that allows the system to run correctly. The possibility of scaling the power modularly makes it applicable to the System Scalability context..* *Cr#2: Security is managed by the cloud service provider, and allows for a standard without implementation work.* *Cr#3: The cloud service allows fault tolerance and backup service techniques, allowing rapid system recovery in case of failure.* *Cr#4: The cost may vary, in order to satisfy mainly the Performance and Reliability points, but it remains applied as a subscription for the entire duration of use of the system.* *Cr#5: The migration is managed by the cloud service with the predefined methodologies that are made available. This reduces the possibility of a technology change during use.* *Cr#6: The mix of Nosql and Mysql guarantees the scalability of the system as the number of users and museums grows.*
	Rationale of decision	*While the precedent option has been rejected in order to avoid risk on a very important element of the overall system, this option garantuee a solution with low risk due to the fact that the services are known a priori. In addition a provider of cloud services gives a set of services that can be useful in the overall system.*

Booking system

Concern (Identifier: Description)		*Con#4: How the system have to manage the booking system?*
Ranking criteria (Identifier: Name)		*Cr#1: Usability* *Cr#2: Reliability ("at any moment for a given museum, only 500 people are allowed inside")* *Cr#3: Accessability*
Options	***Identifier: Name***	*Con#4-Opt#1 : Time slot of visitation*
	Description	*Visits are divided by time slot (defined by the administration team) within which visitors can make their visitation. In particular visitors can buy tickets by the dedicated booking page (selecting a free time slot) or thanks to the ticket office at the entrance (which will sell tickets only if there are free place in the time slot). The queue to the ticket office will be devided from the queue to the QR reader, and after buy the ticket at ticket office the visitor have to go to the queue of QR reader.*
	Status	*Has this option is rejected*
	Relationship(s)	
	Evaluation	*Cr#1: This option may force the visitor to decide a unliked time slot and may ruin the immersive experience due to definition of time slot done by the administration team* *Cr#2: This option satisfy the Cr#2, because the system can check that in any time slot there aren't more than 500 people.* *Cr#3: This option satisfy the Cr#3, because people which can't book tickets online (because don't have any device, or are tourists which decide to visit the museum at the last moment and don't have a smartphone at place) can ticket them to the ticket office.*
	Rationale of decision	*This option may ruin the overall experience for visitors if time slot are too short and from the other hand may lower profit if time slot are too long. In addition this option force the visitor to visit the museum in a particulary time slot, whether they like it or not.*
	Identifier: Name	*Con#4-Opt#2 : max number of tickets and time slot only for statistics*
	Description	*For each day the administration team choose a max number of salable tickets and a set of time slot. When visitors buy the ticket indicate in which time slot they think to visit the museum, so that the system can make statistics on how much the museum is crowded in a certain time slot.These statistics are showed to other visitors which are alerted that if there are already 500 people in the museum in a certain moment, they have to wait until a place is freed. They are also alerted that in the unfortunate case in which they can't enter the museum because it's full until the closure, they can request the refund of ticket. As Con#4-Opt#1 there will be a ticket office at the entrance for them who doesn't booked online the tickets, and the queue of ticket office is divided from the queue to the QR reader (so after buy the ticket at the ticket office the visitor have to go to the queue of QR reader). At visitors who have booked the ticket by ticket office won't be guarantee the refund of ticket.*
	Status	*Has this option is dediced*
	Relationship(s)	
	Evaluation	*Cr#1: This option is easy to use for visitors, which are free to choose as they wish as long as salable tickets are avaiable.* *Cr#2: This option may don't satisfy the Cr#2 because a visitor can wait for entering for a long time or even don't enter, but the system try to contain the probability of failure showing statistics on time slot to visitors when they booking tikets. In the worst case in which the visitor can't enter (probability of this case is dependent on how many tickets are salable each day, so the administration team can adjust this number in order to low this probability) the system allow the visitors to get a refund, which is a loss of money for the system but a way to feed the visitors.* *Cr#3: This option satisfy the Cr#3, because people which can't book tickets online (because don't have any device, or are tourists which decide to visit the museum at the last moment and don't have a smartphone at place) can ticket them to the ticket office. Note that visitors which have booked tickets online have an advantaje over those which have used ticket office, because they go directely to QRcode queue and are entitled to refund.*
	Rationale of decision	*This option satisfy the first and the third criteria. On the other hand the second criteria may don't be satisfy in particular cases, but the system try to contains the probability that these cases arise up using a maximum number of ticket and statistics on time slots.*

Broker

Concern (Identifier: Description)		*Con#1: Which IoT broker should be implemented for the publish/subscribe protocol?*
Ranking criteria (Identifier: Name)		*Cr#1: Scalability* *Cr#2: Maintainability* *Cr#3: Security*
Options	***Identifier: Name***	*Con#1-Opt#1 : Self Hosted Brokers*
	Description	*The Broker is installed on an internal server*
	Status	*the option is rejected*
	Relationship(s)	
	Evaluation	*Cr#1: In This option the scalability is hard to obtain and there is the need of someone with deep knowledge of the system always present* *Cr#2: This option is hard to maintain* *Cr#3: This option have good security*
	Rationale of decision	*This option could fail on Cr#1 and the Cr#2 because of the need of deep knowledge of the entire system by the admin*
	Identifier: Name	*Cr#1-Opt#2: Managed Brokers*
	Description	*The Broker is present on the cloud*
	Status	*the option is accepted*
	Relationship(s)	
	Evaluation	*Cr#1: In this option the scalability is easily to obtain using the cloud resources* *Cr#2: The mantainability is easy to maintaint trough the cloud* *Cr#3: This option have good security*
	Rationale of decision	*This option sastify all the criteria with focus attention of the scalability aspect of it*

Views and Viewpoints

Stakeholders

- **Visitor**: The end-user, the one who will use the various features made available by the system. Its priorities will be the ease of use of the system and the interaction with it. The Concerns of this Stakeholder are mainly related to the visual presentation of the system, and to the number of actions that are allowed within it.
- **Museum Operator**: This Stakeholder identifies the privileged user of the system, who is assigned the management operations of the End User operations. Here, too, Concern is the visual presentation and actions allowed by the system.
- **Core Developer**: Its function is to develop and maintain the system and the interaction between its components. Its priority is high ease of management of the system, and the ability to adapt according to requests. The Concern of this Stakeholder will be the architecture of the system in its entirety.
- **System Integrator**: The task of this Stakeholder is to connect and implement services external to the system, to allow its use and management within it. His Concern is aimed at the integration capacity of the system.
- **Data Analyst**: Its role is to develop and implement a Data Analysis system that processes the information produced by the system and produces customized statistics to enhance the use of the system by the End User. His Concern concerns the collection and use of system data.
- **Database Developer**: It deals with the management of data within the system so that it is easily accessible and usable by the various Stakeholders who need it. Concern focuses on the data structure underlying the system, and its management.
- **Customer**: The purchaser of the system, the one who uses the services made available by the system for his or her company or company. Its main concern is the quantity and quality of the services offered by the system.

Concerns

- **Usability**: Ease of use of the system, end-user interaction with the various services.
- **Efficiency**: System speed and reliability, stability in daily use and during a high workload.
- **Security**: Security of the information contained in the system, robustness to harmful and dangerous external accesses.
- **Back-Office Management**: Possibility of system management by system supervisors.
- **Data Management**: Collection, exchange and storage of data by the system.
- **Budgeting**: Costs from the use and maintenance of the system.
- **Scalability**: Possibility of expanding the system to allow its use in contexts of different sizes.
- **Maintenance**: Complexity of system maintenance during its use.
- **Data Analysis**: Ability to analyze system data, for the purpose of improving the service itself.
- **Fault Tolerance**: Resistance to system failures, ability to limit the disservice in case of system anomalies.

Stakeholder / Concern Interconnection Table

System Development Section

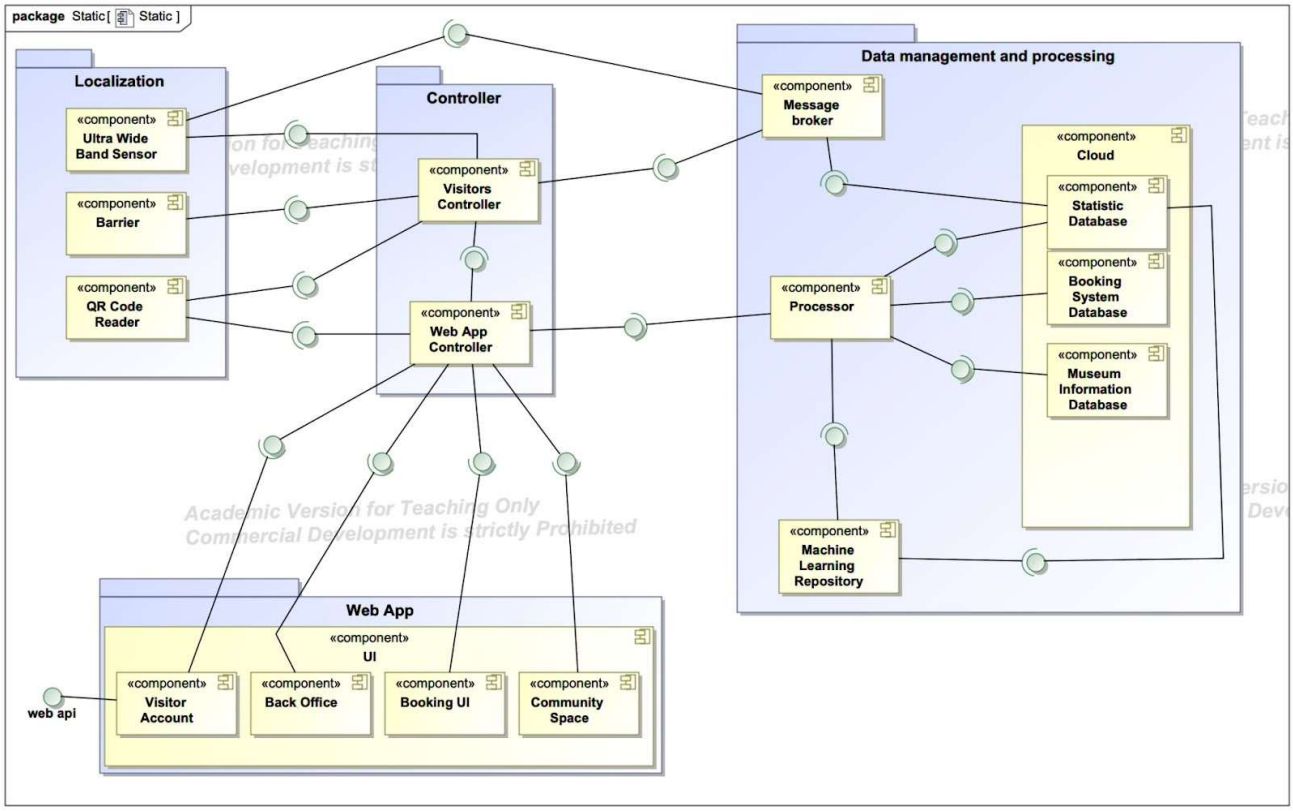
Concern	Core Developer	System Integrator	Data Analyst	Database Developer
Usability	X			
Efficiency	X	X		
Security	X	X		
Data Management			X	X
Scalability		X		X
Maintenance	X			X
Data Analysis			X	
Fault Tolerance	X	X		

System Use Section

Concern	Customer	Museum Operator	Visitor
Usability		X	X
Efficiency			X
Security	X		
Back-Office Management		X	
Budgeting	X		
Scalability	X		

UML Static and Dynamic Architecture View

Component Diagram



The architecture is statically defined in the component diagram. The component diagram is divided into 4 main sections:

- Localization
- Controller
- Data management and processing
- Web App

The localization components are used for the localization of the visitor and the ticket validation. In this package, we can find even the component responsible for the localization of artworks and crowds for statistical use. The controller package is the main link between all the parts of the system. In this package, we can find the components responsible for the management of the web app user request (visitors or the museum team) or the automated requests generated by the localization system. The data management and processing components are those used for the storage and/or analysis of all requests made by other parts of the system. The last package is the web app where all the visitors can check the museum or book tickets while the team of each museum can add or update the information about the museum and all the artworks present inside of it.

Localization

- **Ultrawide-band-sensor:** This component is the static description of the technology used for the localization of the visitor in the museum for statistical analysis
- **Barrier:** This component is used for managing the entrance and the exit of the museum by the visitors and from which we can calculate the numbers of people in the edifice.
- **QR Code:** This component is the component that will be used for ticket validation by the visitor.

Controller

- **Visitor Controller:** This component is used for the management of the automated request made by the visitor while staying in the museum
- **Web App Controller:** This component is the static description of the controller that manages both visitor requests made in the web app and the requests made by the team of the specific museum for the managing of the artworks and the booking system.

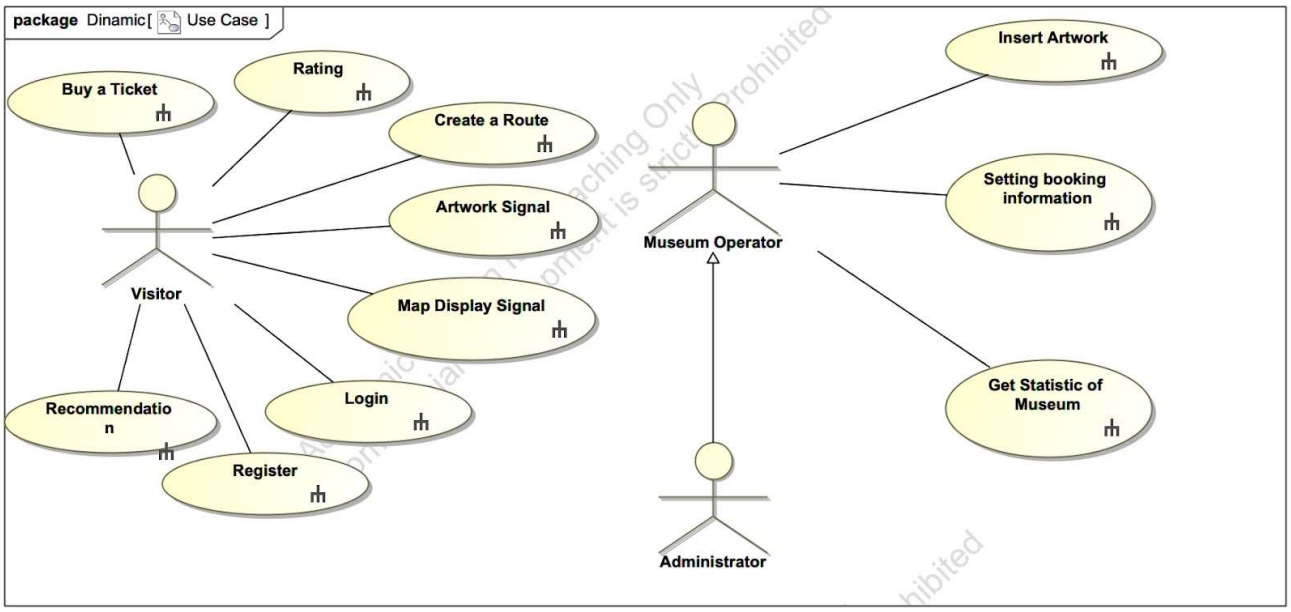
Data management and processing

- **Processor:** This component has the duty to manage requests and processing data.
- **Message broker:** This component manages the message request made by or for the IoT devices in the museum
- **Cloud:** Inside this component, there are various database systems used both for storage and analysis purposes
    - **Museum Statistic Database:** this is the component where the statistic of the specific museum is storage for later uses
    - **Booking System Database:** This is the component where all the booking data is saved and from which we can extract the information for the booking choices.
    - **Museum Information Database:** This is the component where the team of the museum saves and update information for the museum or the specific artwork
    - **Individual Statistic Database:** This is the component where the statistic of the users is saved for future use.
- **Machine Learning model:** This component is used to perform machine learning algorithms on the data saved on the cloud

Web app

- **Web app:** Inside this component, there are various sections of the web app used by different actors
    - **Visitor Account:** This component is where each visitor can manage and personalize his experience
    - **Back Office:** This component is where each museum team can visualize the statistics for the respective museum and where they can manage the artworks and their respective info.
    - **Booking:** This component is where the visitor can book the museum visit based on their search and/or filters.
    - **Community Space:** This component is the static description of the communal space where all the visitors can share their experiences with each other.

Use Case Diagram



Actors

- **Visitor:** Is the tourist who visits the museum and books the tickets.
- **Museum Operator:** Is the operator who has to manage the artwork and other information on the system.
- **Administrator:** This actor has the same power as the museum operator but at a system level, so he can access all the museum dashboards and can manage users.

Use-Case Description

Name	Buy a ticket
Actors	Visitor
Trigger	The visitor search for a ticket
Description	The use case describes how a visitor can search and buy a ticket for visiting a museum

Name	Rating
Actors	Visitor
Trigger	The visitor makes a rate about a part of the museum
Description	The use case describes how a visitor can rate a museum, an artwork or all the artwork and even how he can rate an individual route

Name	Create Route
Actors	Visitor
Trigger	The visitor creates a route using a form
Description	The use case describes how a visitor can create a route inside a museum, that differs from the default one

Name	Insert Artwork
Actors	Museum Operator (or Admin)
Trigger	The museum operator inserts a new artwork using a form
Description	The use case describes how a museum operator can insert a new artwork displayed in the museum inside the system.

Name	Setting Booking Information
Actors	Museum Operator (or Admin)
Trigger	The museum operator change the booking setting using a form
Description	The use case describes how a museum operator can update the museum booking setting for future use.

Name	Get Statistic of Museum
Actors	Museum Operator (or Admin)
Trigger	The museum operator enters the statistics data page of his museum
Description	The use case describes how the system calculates all the museum statistics for displaying it to the museum operator

Name	Artwork Signal
Actors	Visitor
Trigger	The visitor is near an artwork
Description	The visitor is near an artwork. The IoT devices send signal the system and this supply the visitor with all the information about it

Name	Map Display Signal
Actors	Visitor
Trigger	Visitor is moving in the museum
Description	The sensors signal to the system the user movement and the system gives back to the user a route to follow

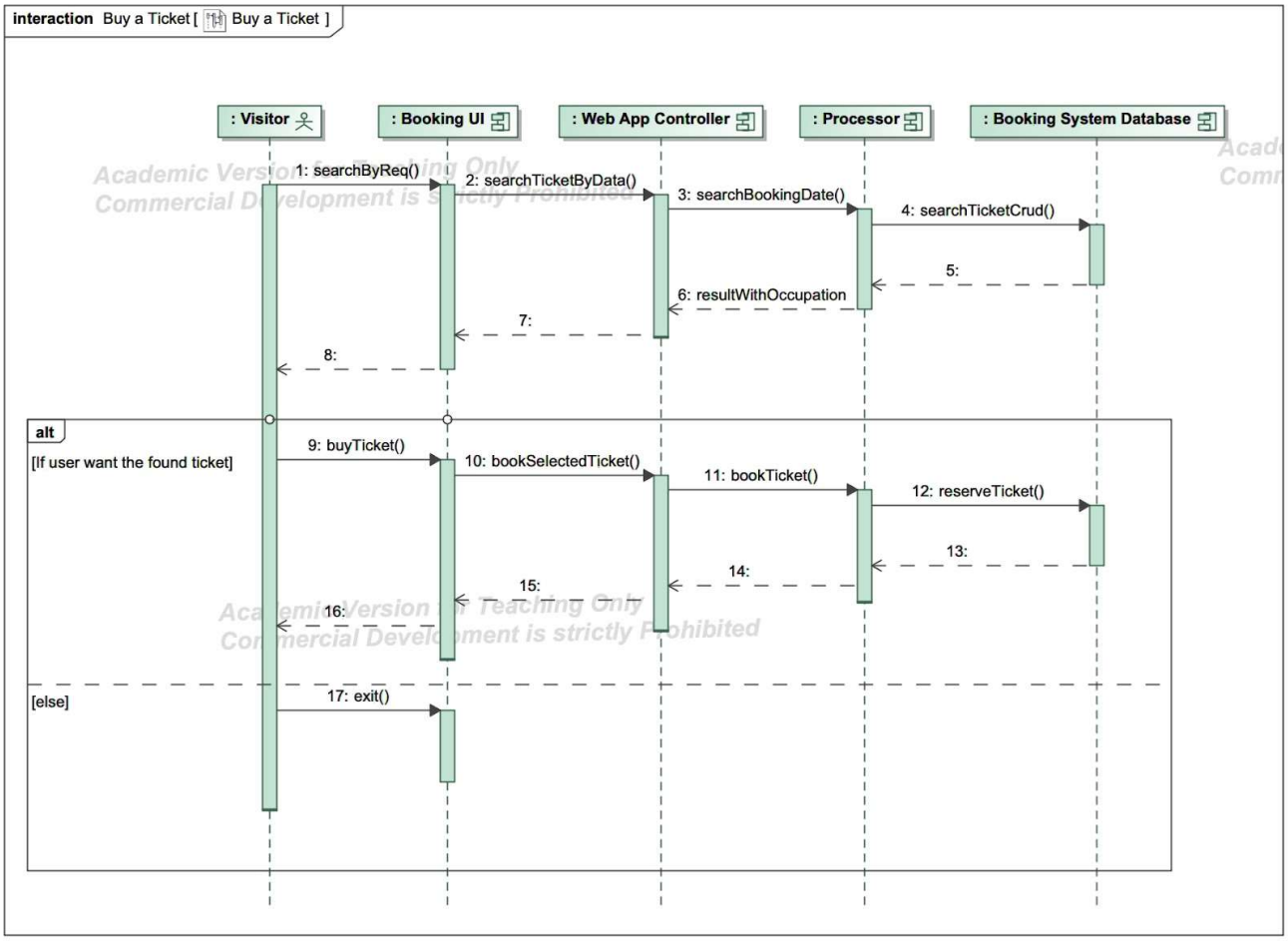
Name	Login
Actors	Visitor
Trigger	Visitor login in the web app
Description	The system authenticates the user

Name	Register
Actors	Visitor
Trigger	Visitor register to the system
Description	The visitor register himself in the web application

Name	Recommendation
Actors	Visitor
Trigger	Visitor check the recommendation section
Description	The system generates using machine learning recommendation of museums for the specific user

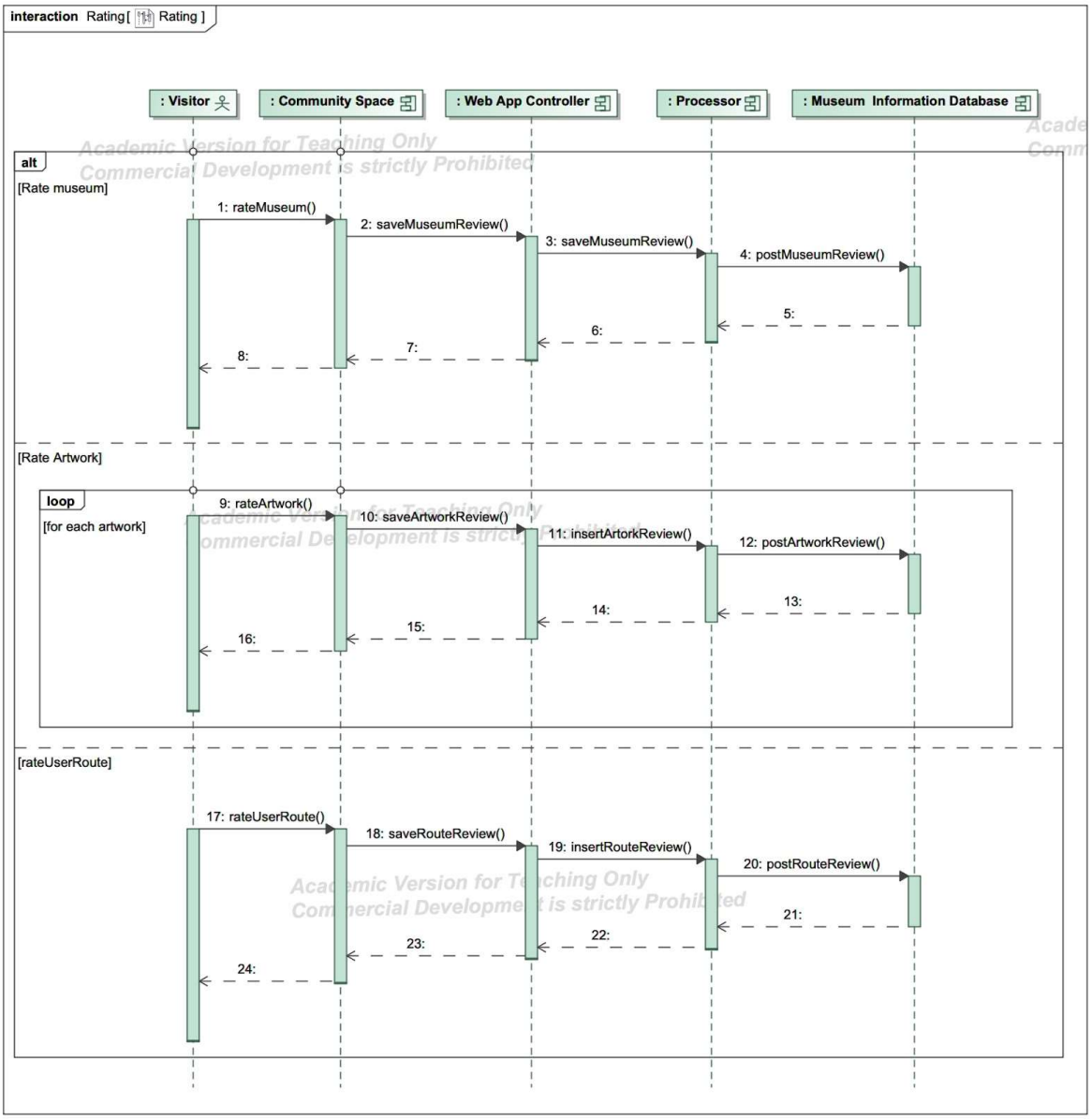
Sequence Diagram

Buy a ticket



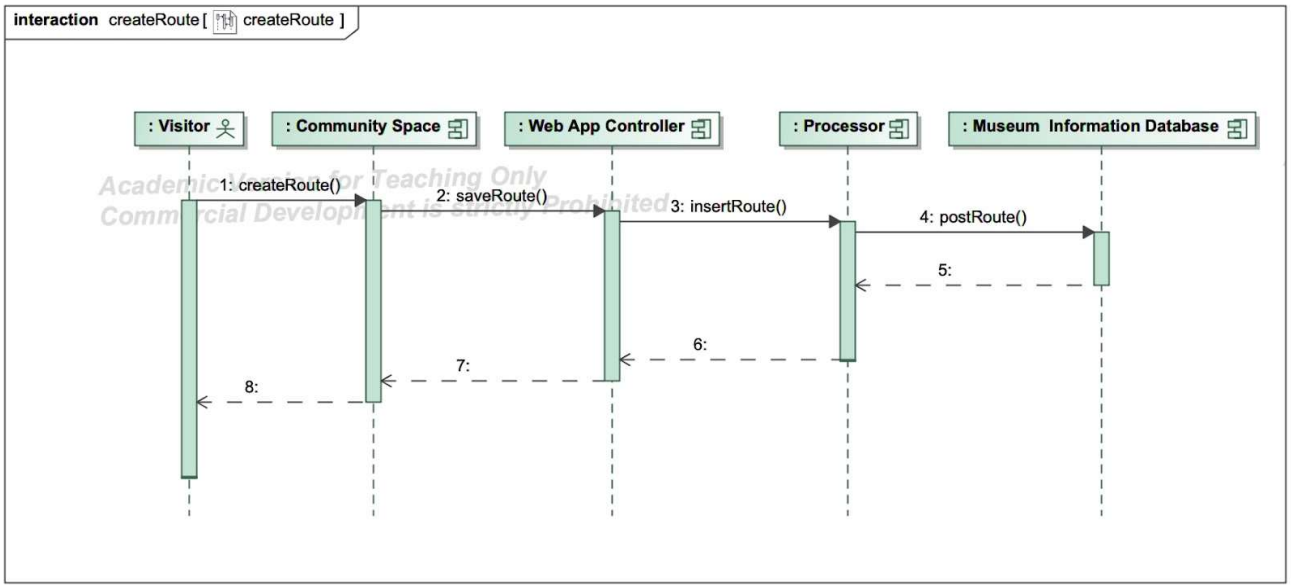
In this sequence, we can find the flow that manages the booking of a ticket. The visitor search by using some parameters, this request is managed first by the booking component and then by the web app controller. The web app controller then requests the booking system database through the processor. The result of the last request is the results of available tickets and the occupation of the museum based on the parameters entered by the user.
At this point, the visitor has two choices. The first one is the booking of a ticket based on the result of the previous interaction. The visitor makes this request to the booking component and this sends the request to the web app controller. The web app controller then makes the reservation of the ticket on the booking database through the processor. The alternative of booking a ticket after the results of the tickets is to exit from the booking.

Rating



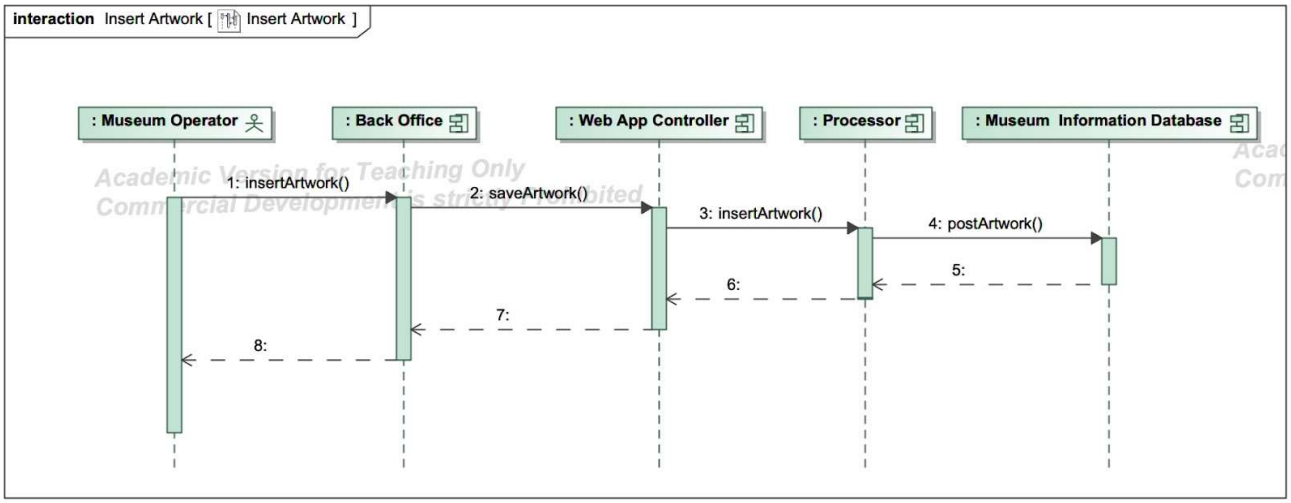
The visitor in the system has different alternative things to rate. The first one is the museum in general. This request is made on the community space by the visitor, then the community space calls the web app controller which inserts the rating in the museum information database through the processor. The second rate possibility is the rating of the artworks. The visitor can rate each artwork present in the museum with the possibility of adding one or more photos or videos about the specific artwork. Each rate request is managed by the community space components that send the request to the web app controller. Then another request is made by the web app controller through the processor to the museum information database to save the review of the artwork. The third one is the rate of the museum maps made by other users. This request is first managed by the community space. The community space then sends the request to the web app controller which sends the request through the processor with the review to the museum information database for writing it on the system.

Create Route



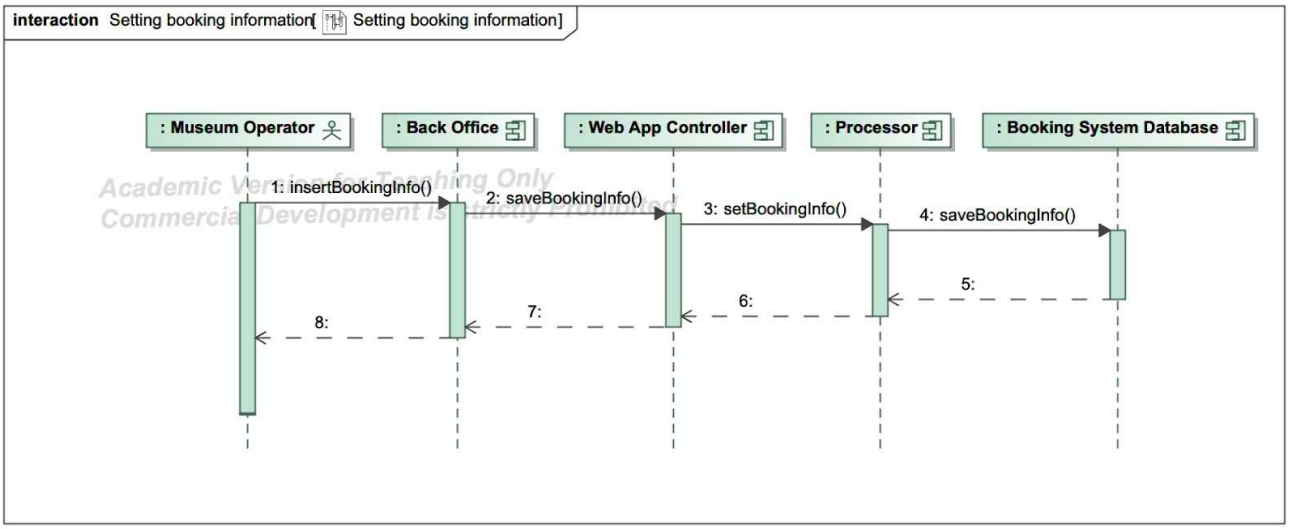
The visitor can create an independent route and make it visible to the other user in the community space. After the route is complete the visitor post it on the community space. The community space components send the request to the web app controller. The latter sends the request through the processor to the museum information database to save it. In the previous sequence diagram, we described how other visitors can rate the route made by another visitor.

Insert Artwork



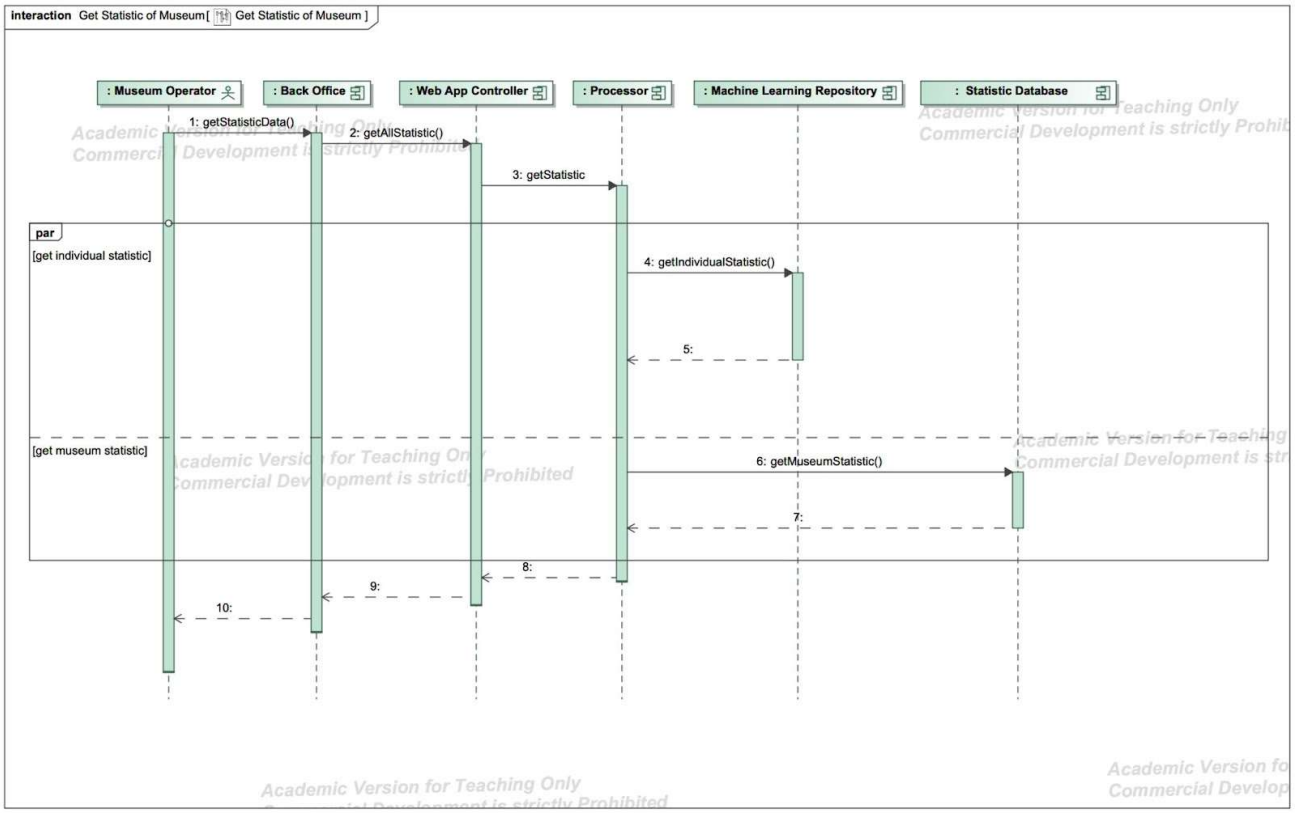
This sequence diagram describes the upload of the information for a new artwork inside a specific museum. The museum operator can insert the new artwork information inside the back office component. The back office component then sends those requests to the web app controller and proceeds to save the artwork information through the processor on the museum information database component.

Setting Booking Information



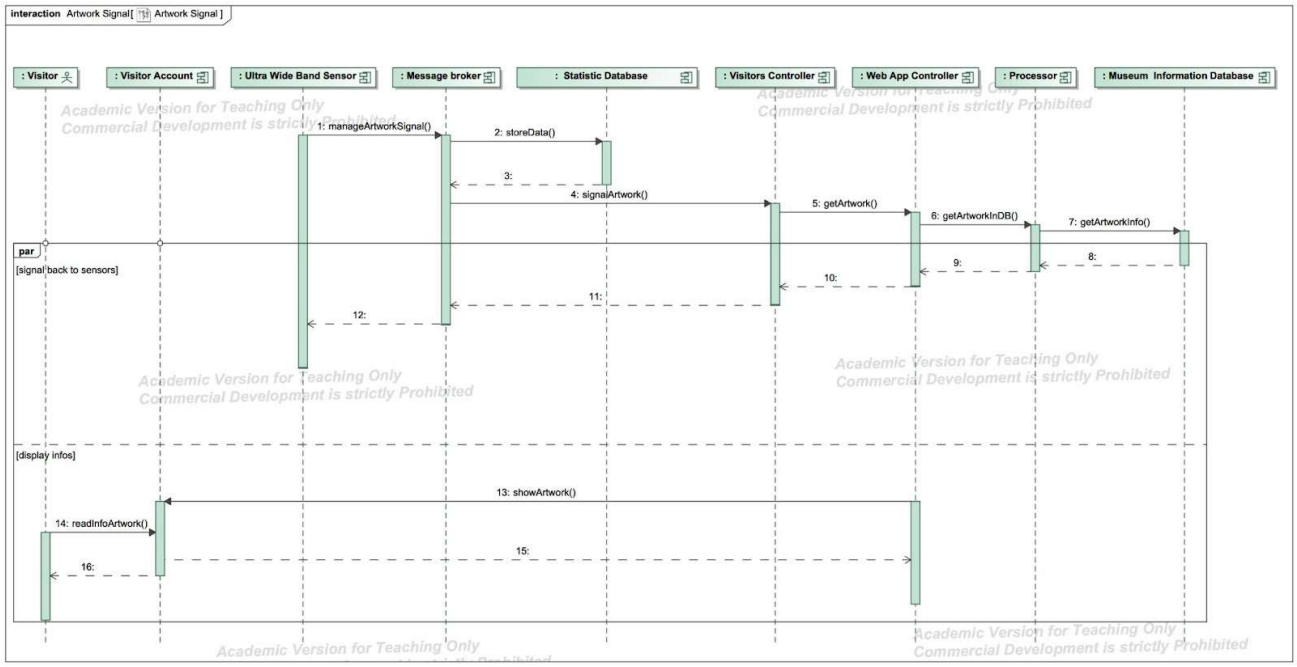
In this sequence diagram, we can see how the museum operator can insert or update the booking setting of their museum. The booking information such as the time slots is inserted in the back office component. The back office component then sends the request to the web app controller. This last component then proceeds to insert or update the information in the booking system databasethrough the processor. A message of completion is sent back to the back office for the museum operator.

Get Statistic of the Museum



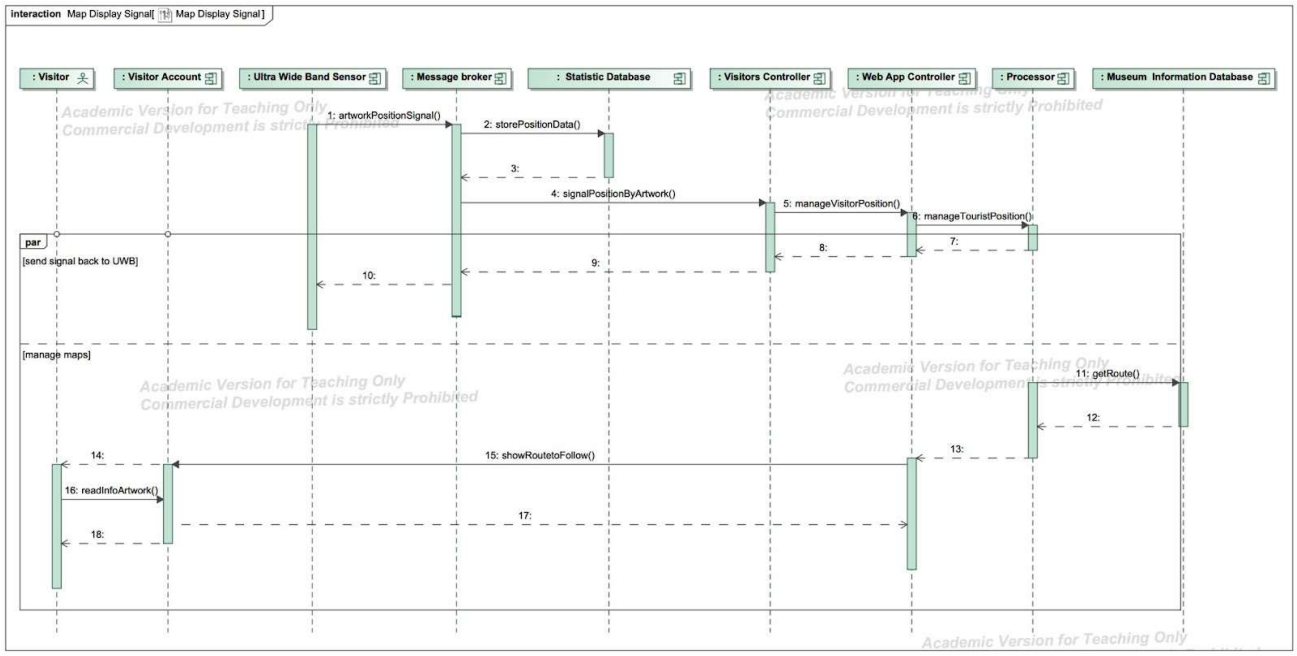
In this sequence diagram we see how the statistic page in the back office of the museum operator is calculated. The museum operator asks to see the data of their museum by the back office page. The back office then asks the controller to accumulate the data for the display. Then the web app controller makes requests in parallel through the processor. The first request is sent to the machine learning model component which is used to calculate the statistics of the museum using the visitor's statistics of the people who visited and saw the artworks. This calculation is made by this component by asking the individual statistic database for the actual data. Then the machine learning model sends back the worked data to the web app controller. The second request made by the web app controller is sent to the museum statistic database where more general statistics are stored using also IoT devices. After the end of this parallel request, the web app controller sends back the data for the page to the back office.

Artwork signal



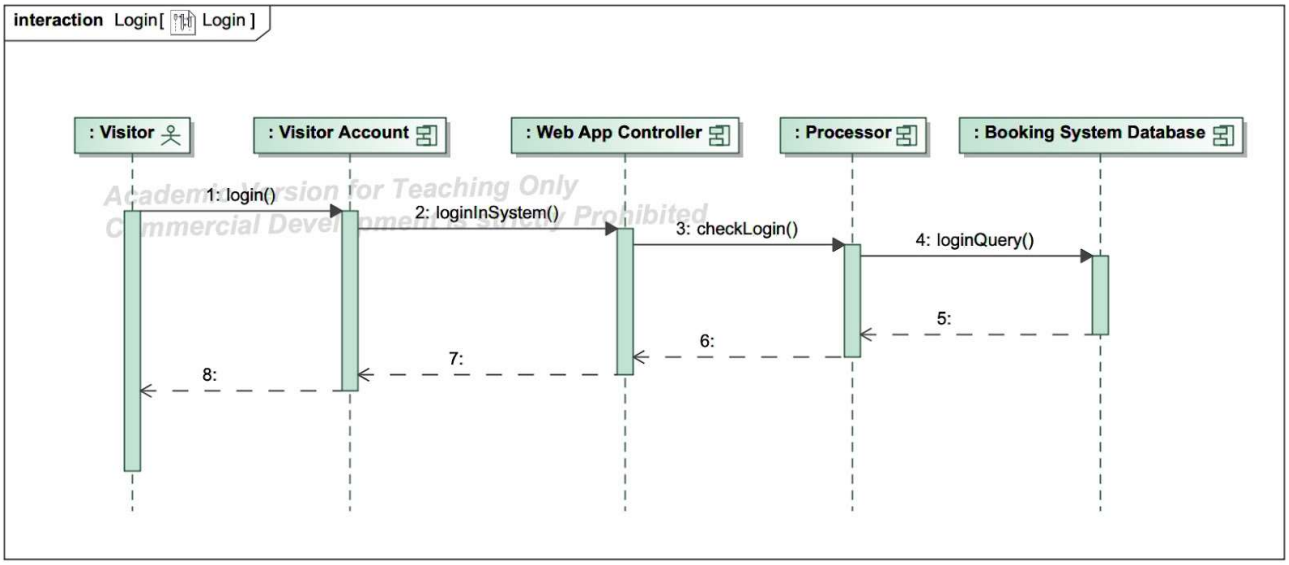
In this sequence diagram, we see how the display of information about what he is seeing is made while the visitor is standing near the artwork. First, the Ultra wideband sensor via the publisher/subscriber paradigm has an interaction with the message broker who stores data for statistical usage in the museum statistic database. After this, the message broker informs the visitor controller to start the procedure for information retrieval about the artwork. The visitor informs the web controller to get the specific artwork. The web controller gets the information needed from the museum information database through the processor. At this moment two parallel procedures come into play. The first one is the ok message back from where the signal started while the web app controller sends the information gathered in the previous moment to the visitor account and this last component is used by the visitor.

Map Display Signal



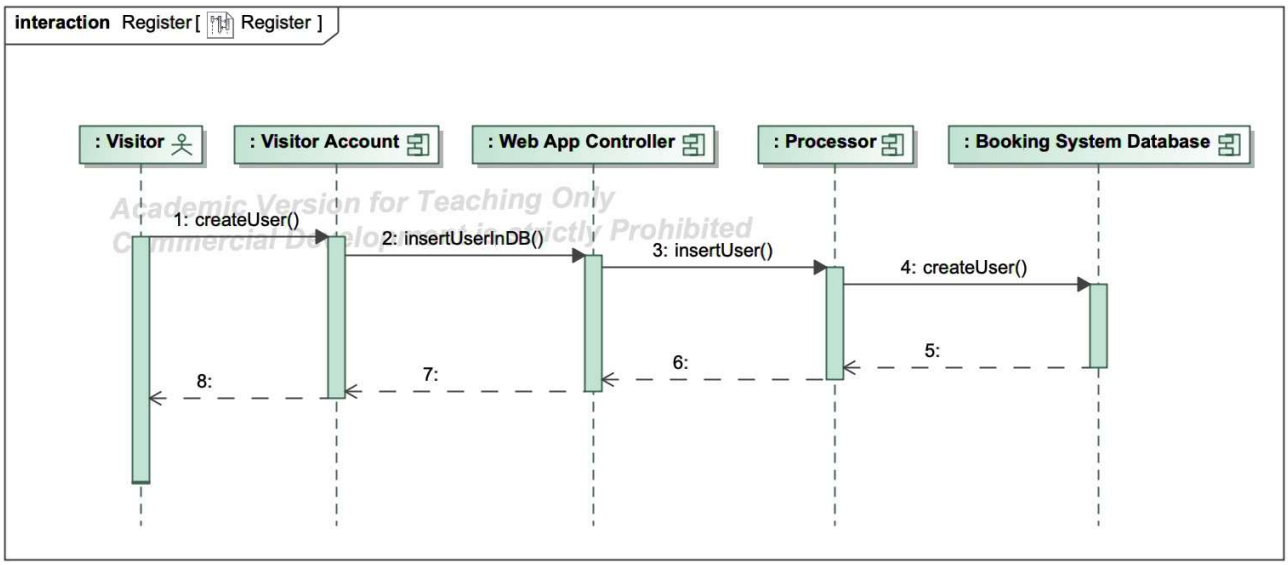
In this sequence diagram, we see how the display of information about the route is made while the visitor is standing near an artwork. First, the Ultra wideband sensor via the publisher/subscriber paradigm has an interaction with the message broker who stores data for statistical usage in the museum statistic database. After this, the message broker informs the visitor controller to start the procedure for information retrieval about the route. The visitor informs the web controller to get the right route to show. The web controller gets the information needed from the museum information database through the processor. At this moment two parallel procedures come into play. The first one is the ok message back from where the signal started while the web app controller sends the information gathered in the previous moment to the visitor account. The visitor is informed with a notification of the updated route via the visitor account component and this last one is used by the visitor to see the change.

Login



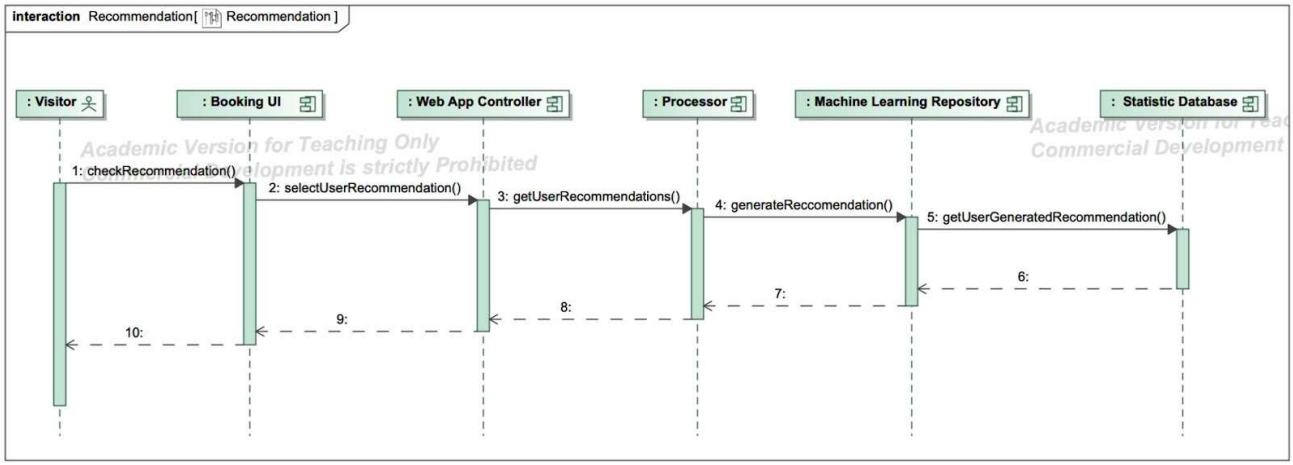
In this sequence diagram we see how the visitor logged into the system. The visitor inserts his data into the visitor account. The visitor account sends a request to the web app controller. The web app controller sends a request to the processor who checks the data in the booking system database. Then the return message is sent back to the visitor account who unlocks the system to the visitor

Register



In this sequence diagram, we see how the visitor registers himself into the system. The visitor inserts his data into the visitor account. The visitor account sends a request to the web app controller. The web app controller sends a request to the processor who inserts the data in the booking system database. Then the return message is sent back to the visitor account who is logged now in the system.

Recommendation



In this sequence diagram, we see how the visitor can see the recommendation museums. The visitor opens the section of the booking UI for the recommendation. Booking UI sends a request to the web app controller to get the User Recommendation. The web app controller sends a request to the processor who calls the machine learning repository. The machine learning repository calculates from the statistic database user recommendation. The calculated results are sent back to the Booking UI where it's displayed to the visitor.

From Architecture to Code

We implemented the prototype using laravel and python. To simulate the project we created 3 museums and different artworks and users. We used python to simulate people walking inside the museum near artworks. For this purpose, we used different libraries that are needed for running the project.

Python Libraries:

- Flask
- Requests
- Queue
- Threading
- Random
- Datetime
- Time

Repository of the prototype: https://github.com/GianlucaRea/QDMuseum
Video Demo of the prototype: https://www.youtube.com/watch?v=aU1NdMnqf3Q

In the demo we show how users interact with the system. The museum operator can do any crud operation of the artworks or the time slots of the museum. The visitor can easily buy a ticket and validate it using also the QR Code. Particularly in the demo we show how the admin can see the people in the museum in real time with the presence of the artwork in the map.

Summary

The main goals of the project, so the development of a system which lets the visitors have an immersive experience and the administration to manage the crowd and statistical data, should be reached thanks to the technologies used by the system and how the architecture is shaped. Also the main risks, challenges and both functional and non-functional requirements identified should be all resolved, thanks to the design decision and the research of solutions made in the preliminary phases, such as the research of the different types of technologies for indoor localization.

If well implemented, the UWB has very high accuracy with a very good trade-off of costs/benefits, and the way the visitors are associated to tags (so through the coupling phase possible thanks to ticket validation) guarantees both privacy terms and an efficient way to give localization services to visitors. Also, in order to have a good level of performance, the system uses an external Cloud service for data management and processing. These solutions make possible the app implementation as a web app, in fact, the flow of data for localization services are managed by the back-end, and not with direct communication between the tag and the device used by visitors. This means that the system reaches any kind of visitor (smartphone, laptop, tablet, etc.) and the costs of implementation are low compared with other solutions which reach the same range of visitors.

Finally the Prototype, despite having bare graphics and only simulates the UWB, shows how the system developed is architected and how the main goals of the project are satisfied effectively (such as the fruition of localization services by visitors and admin, booking tickets and validation and dashboard for administration team). Some parts of the architecture have been ignored or partially developed by this prototype (like minor services such as weather data or the social part) due to the less priority for the system.

References

[1]: Zhang, Yiyi & Gao, Shuai & Xiao, & Meng, Liqiu & Li,. (2020). Cost-Effective Wearable Indoor Localization and Motion Analysis via the Integration of UWB and IMU. Sensors. 20. 344. 10.3390/s20020344.

[2]: Zhao, Yanyang & Wang, Ligen & Frigon, Jean-Francois & Nerguizian, Chahé & Wu, Ke & Bosisio, Renato. (2006). A Software Defined Radio Receiver Architecture for UWB Communications and Positioning. 255-258. 10.1109/CCECE.2006.277399.