University of Pretoria
Software Engineering - COS 301

A-Recognition

Team Singularity
3 May 2019

Team Members:

Richard McFadden	**u17026662**
Adrian le Grange	**u17056782**
Jarrod Goschen	**u17112631**
Alessio Rossi	**u14137934**
Kyle Olivier	**u15001319**

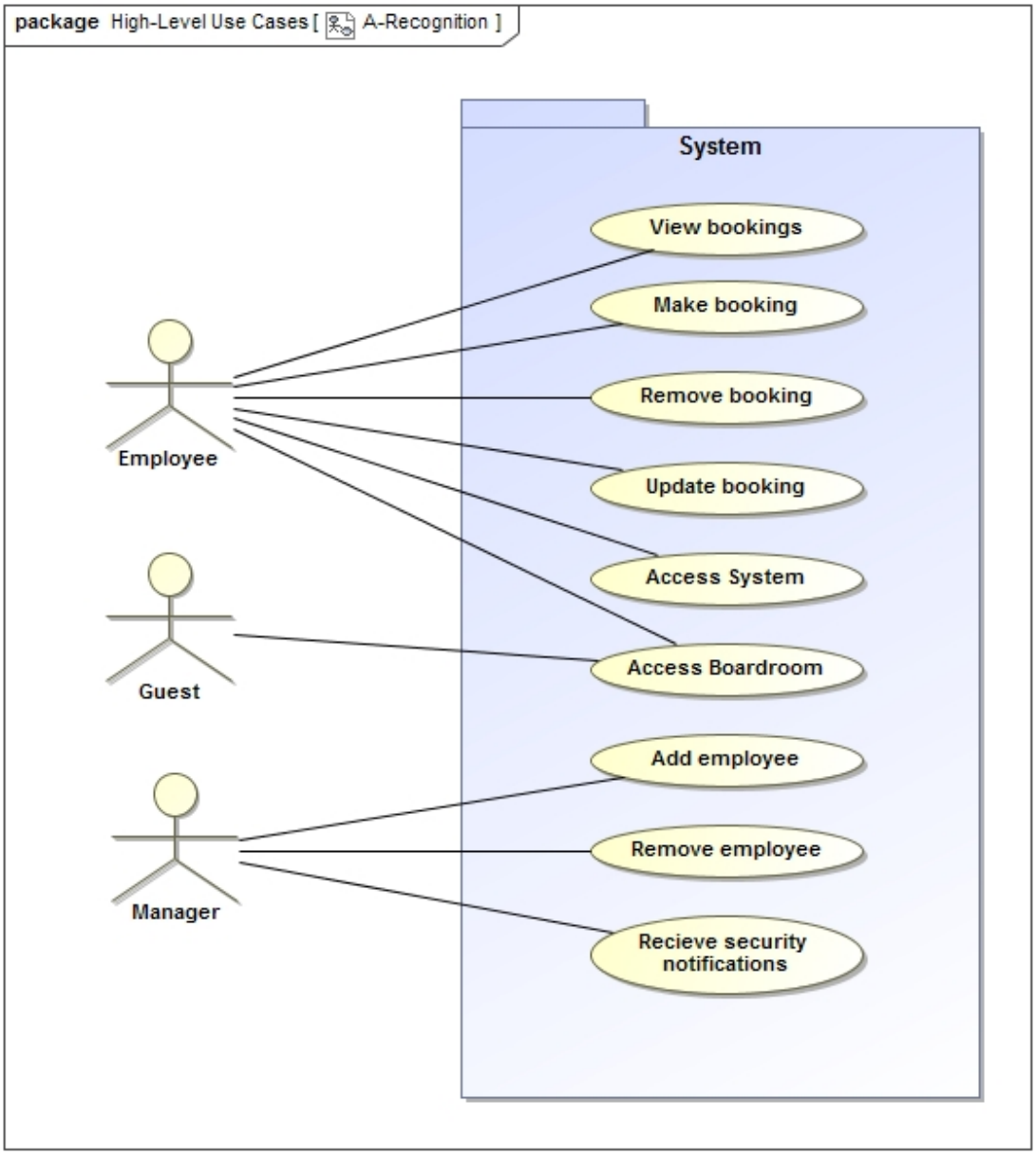
Contents

1 Introduction

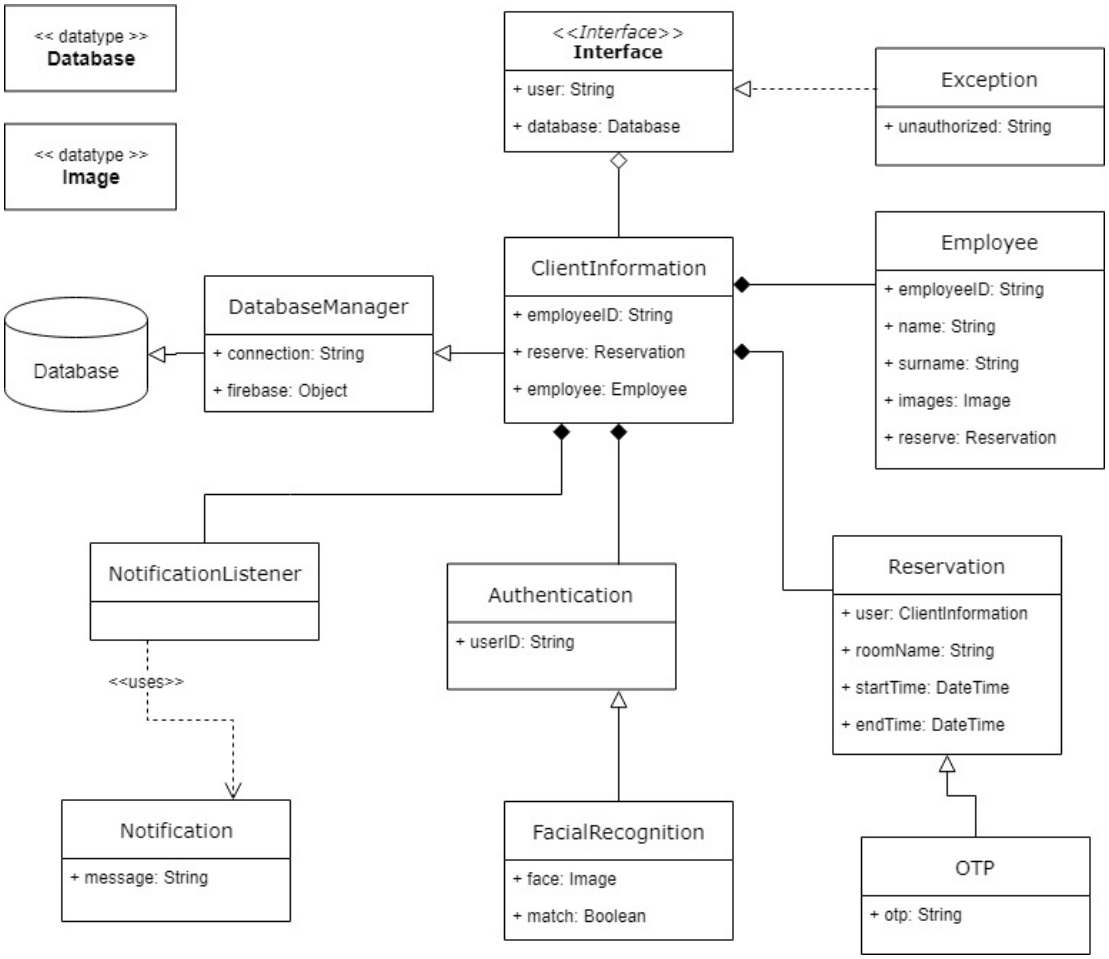
The aim of this project is to develop an access management suite to control who may enter specified areas of an environment. These areas will be identified as "Rooms" throughout this specification. The project's objective is to allow for access to be provided mainly through the use of facial recognition, while still allowing guests to be provided access through a secondary system that generates one-time pins.

The need for this system has arisen due to the apparent disregard for current booking systems and a lack of an elegant way to enforce them as well as the intent to extend the systems functionality further into an integrated security system for the premises.

The scope of our project entails the incorporation of the previously mentioned access management system with the companies currently utilized booking systems, ideally allowing for seamless integration without changing current booking procedures.


package High-Level Use Cases [A-Recognition]
System
View bookings
Make booking
Remove booking
Update booking
Access System
Access Boardroom
Add employee
Remove employee
Recieve security notifications
Employee
Guest
Manager

2 Domain Model



3 Architectural Design

The overall A-Recognition system architecture is primarily a client-server architecture because the work environment requires a high-security, and all business operations need to be verified server-side and not client-side. The client-server architecture allows for a secure system where multiple clients (employees or guest users) can connect to a single server. Using this architecture, we can maintain that the clients do not have access to each others information as confidentiality is a large security concern for EPI USE - and the server processes all requests, ensuring that access cannot be given without the appropriate validation by the authorization system.

By using this monolithic model, one can offload the work of heavy tasks to the server, so the clients can be low power and thus low cost. The client side is intended to be deployed on a simple user interface with very few interactions and a camera to capture the user. As such very cost effective computers can be used. This means the user interface will be cheaper to maintain, replace, and will consume less electricity, saving costs.

The A-Recognition system will employ both an interactive subsystem as well as a database subsystem. The interactive subsystem will be governed by events triggered by a user in which their will perform many of the tasks depicted in the functional requirements. Users with administrator privileges will be able to perform tasks such as adding a new employee to the system and providing their information, that will be processed through the Client Information Subsystem and will fall within the interactive sphere of the overall system. Other functional requirements that will require interaction will be the process of booking a meeting room and adding the relevant personnel to the meeting rooms. This will require the user to inspect a meeting room as well as attempt to make a booking (if it isn't already booked).

A database subsystem will be employed by pulling the relevant data from the databases to provide a quicker and more efficient access when a user is attempting to gain access through a facial recognition scan. It will also be involved within the process of adding employee data as it will be submitted to the database while shielding the rest of the system from the database implementation.

Our proposed system will be employing a structured approach to provide a high cohesion and low coupling between subsystems. Due to the nature of the proposed subsystems, a component based (or modular) architectural style will be applied. This will allow us to seperate the means of concern between each subsystem (or module) to allow an effective sense of reusability as well as a far more effective sense of effectivity when providing fixes.

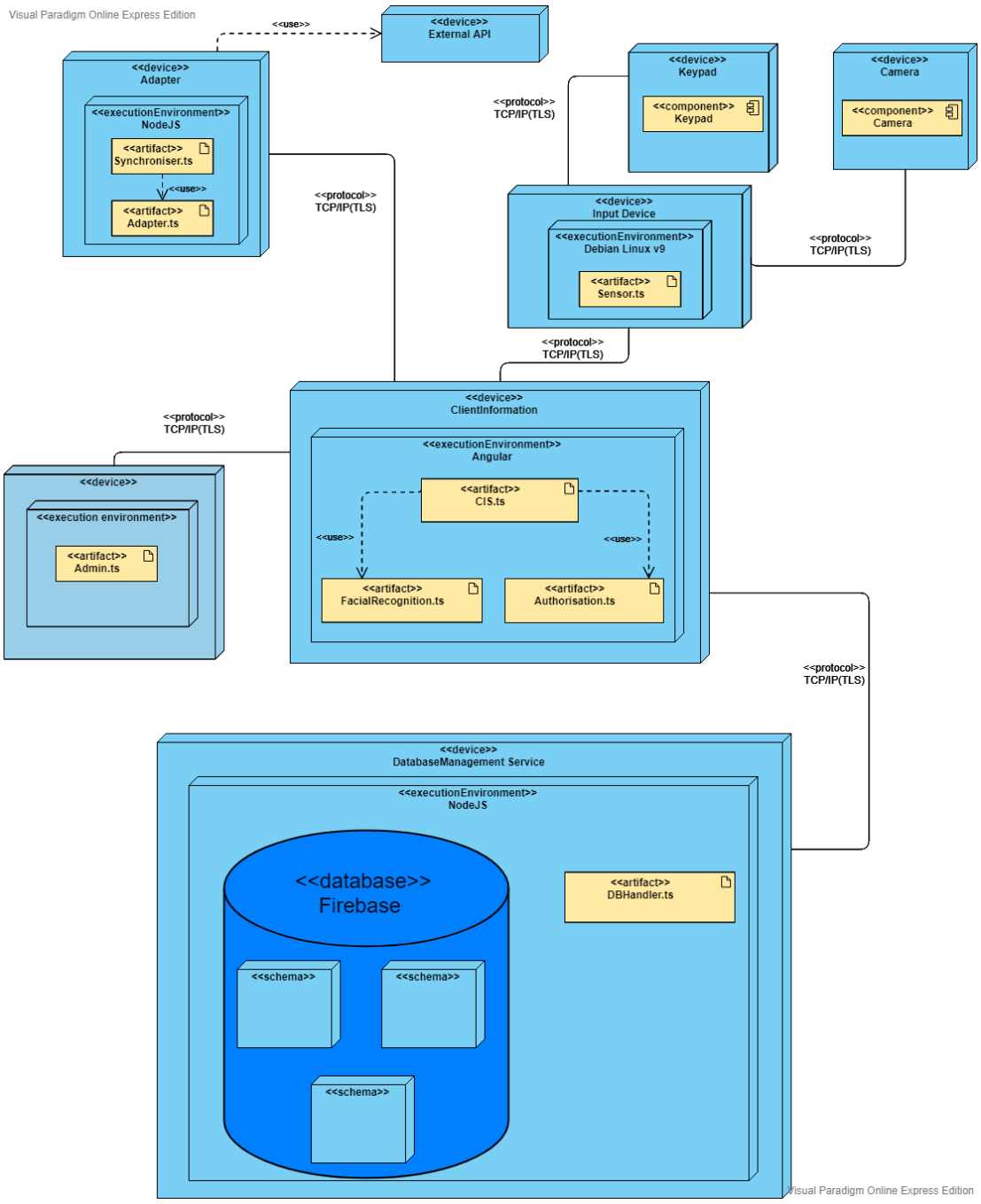
The benefit of using a modular based style allows the overall system to be pulled apart and separate subsystems used in other system that may be developed in the future. This allows a breakdown of the application into reusable functional components that are able to communicate over a network to provide data to other subsystems.

By using a component or modular based architectural style, a number of benefits and advantages will appear. This includes, as mentioned above, a sense of reusability by allowing many components to be reused in many other systems in the future. The context of each subsystem will therefore not be entirely specific to the entire system as different environments per subsystem may be defined. This will allow a greater sense of reliability as each component is independent and will improve flexibility by using a low coupling technique.

A layered approach or architectural style will also be used as the various subsystems will be separated (similar to component based) into varying layers. This will allow a user interface to be independent and not interfere with other layers such as authentication or the database manager. The process of a layered approach can be seen when a user attempts to access a room that has been booked. The user will provide their data via the facial recognition software in which it will be passed to the authentication module (or layer). This will then be compared to data that was retrieved from the database via the database manager layer to determine the validity of the user that has attempted access. In the case of a successful authentication, the user will then be granted access to the relevant meeting room. The system will be be divided essentially into three global layers, the server (authentication and database manager), the client (client information system and notification) and the browser or interface layer. These global layers will communicate with the other layers via the network to provide the relevant information between them.

A client-server approach will also be used in which the client will interact with the server. This is achieved by a client providing their data (their facial images) via the input source outside of a meeting room. The client data (the input source of the facial data) will then be compared to data that was retrieved for each individual employee that was added to a booked room to determine the success of the authentication measures (the server).

3.1 Deployment Model



4 User characteristics

We have identified three main user types that will interact with the system, they are as follows:

1. **Registered Employee**:
   Registered Employees are members of the system, they have their data store in the system and can utilize the full functionality of the access and booking systems. These users are able to have access granted via facial recognition as well as create reservations for a room.

2. **Guest**:
   Guests are users that are only using the facility temporarily or employees that have yet to be added to the system. Guests do not have the ability to gain access via facial recognition, instead guest users must request a one-time pin to be able to access restricted rooms.Both registered employees and guests may utilize the OTP functionality.

3. **Administrator**:
   Administrators are registered employees with added administrative rights. They are able to register employees as well as set access rights for employees.

5 System Use Cases

5.1 Employee

U1.1

Use Case Name:	**Access booked boardroom**
Business Actor:	Employee
Description:	Employee attempts to enter booked boardroom
Preconditions:	Employee provided identification at the boardroom
Postconditions:	Employee is granted or denied access
Basic Workflow:	1. The system determines if the Employee is included in the booking for they boardroom. 2. The Employee is informed on if access was granted or denied.

U1.2

Use Case Name:	**Access System (Login)**
Business Actor:	Employee
Description:	Employee logs into system by providing identification
Preconditions:	Employee is not logged in
Postconditions:	Employee is logged in and can access booking functionality.
Basic Workflow:	1. Employee positions face in front of a camera connected to the system 2. On successful identification the Employee is logged into the system 3. The bookings screen is displayed to the Employee
Alternative Workflow:	2.a. On unsuccessful identification the Employee is displayed with an option to log in with a OTP. 2.b If the OTP option is selected a screen waiting for input will appear, else the Employee can keep trying to log in with facial recognition. 3.a If identification fails due no action is taken and no login is made.

U1.3

Use Case Name:	Make booking
Business Actor:	Employee
Description:	Employee attempts to book a boardroom
Preconditions:	Employee is logged into system
Postconditions:	Employee created a booking or no action is taken
Basic Workflow:	1. Employee clicks on the button for making a booking 2. The system displays the bookings for all the boardrooms and a place to input the desired room, time and the employees allowed in the boardroom during the booked session. 3. The Employee enters details 4. If the requested boardroom is available during the specified time, a button is displayed that allows them to place the booking. 5. The employee clicks the button and the booking is made.
Alternative Workflow:	4.a. Should the boardroom not be available during the specified time the user will not be able to make the booking.

U1.4

Use Case Name:	Remove booking
Business Actor:	Employee
Description:	Employee removes one of their bookings
Preconditions:	Employee is logged into system
Postconditions:	Employee removes booking or no action is taken
Basic Workflow:	1. Employee clicks on the button for viewing bookings 2. The employee can then click on a booking 3. If the Employee created that booking the will have an option to delete it presented to them. 4. The Employee chooses to remove the booking and the booking is removed.
Alternative Workflow:	3.a. If the Employee selects another employee's booking, there will not be a remove booking option displayed.

U1.5

Use Case Name:	View bookings
Business Actor:	Employee
Description:	Employee views bookings on the system
Preconditions:	Employee is logged into system
Postconditions:	Bookings is displayed to the Employee
Basic Workflow:	1. Employee clicks on the button for viewing bookings 2. Bookings are displayed to the Employee

5.2 Manager

U2.1

Use Case Name:	Add Employee
Business Actor:	Manager
Description:	Manager registers a new Employee
Preconditions:	Manager is logged into the system
Postconditions:	Employee removes booking or no action is taken
Basic Workflow:	1. The Manager chooses from his home screen that he wants to add an employee 2. The Manager can then enter the details of the new Employee and then take the necessary pictures 3. The Manager confirms the operation 4. The Employee is then registered on the system.
Alternative Workflow:	3.a. If all details were not provided, the Manager will be notified until all details are provided.

U2.2

Use Case Name:	Remove Employee
Business Actor:	Manager
Description:	Manager removes an Employee
Preconditions:	Manager is logged into the system
Postconditions:	Employee account is disabled on the system
Basic Workflow:	1. The Manager chooses from his home screen that he wants to remove an employee 2. The system prompts the manager to enter the employee ID 3. The Manager confirms the operation 4. The Employee's account is then disabled on the system.
Alternative Workflow:	2.a. The employee name can also be entered to search for employees with that name and the Manager can then select the desired user.

U2.3

Use Case Name:	Recieve security notifications
Business Actor:	Manager
Description:	Manager recieves alerts on repeated OTP attempts, unauthorized access to booked boardrooms and attempted access to Manager functionality.
Preconditions:	Manager is logged into the system
Postconditions:	Notifications is displayed
Basic Workflow:	1. Upon any event that requires attention a notification is sent to the Manager 2. The system then displays the messages on the Manager home sceen
Alternative Workflow:	2.a. If the Manager is not logged in to the system the messages are stored and displayed upon login.

5.2.1 Guest

Use Case Name:	**Access boardroom**
Business Actor:	Guest
Description:	Guest attempts to access boardroom
Preconditions:	Guest received OTP a fixed amount of time before the booked time
Postconditions:	Guest is allowed or denied access to the boardroom
Basic Workflow:	1. An Employee makes a booking and adds Guest to booking. 2. The Guest is notified of the time, place and boardroom of the booking. 3. Some fixed time before the booked time an OTP is sent to the Guest. 4. The Guest then enters the OTP at the boardroom 5. Access is then granted upon successful OTP entry
Alternative Workflow:	Should the Guest attempt to enter the wrong room and enter their OTP, access will be denied and they will have to contact the person that made the booking to guide them to the venue and grant them access.

U3.1

6 Sub-Systems

6.1 Authentication

6.1.1 Requirements

R1.1 The authentication subsystem shall decide whether a given user is authorized for access to a room.

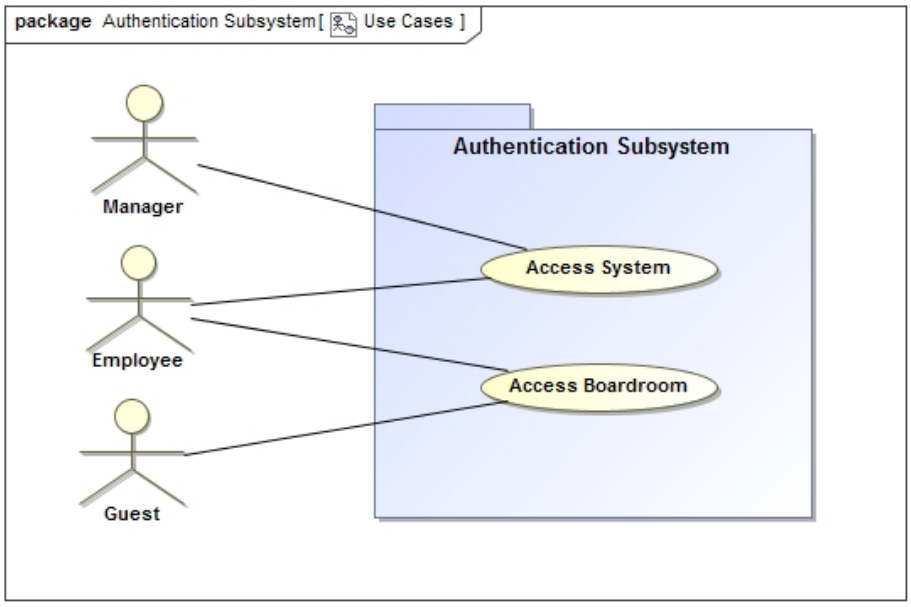
R1.1.1 The subsystem shall grant access to a given room by recognizing authorized users using facial recognition.

R1.1.2 The subsystem shall grant access to a given room if supplied a valid OTP issued to an authorized user.

R1.2 The authentication subsystem shall ensure integrity of the system and secure user information.

R1.2.1 The system shall validate all requests in the system that involve the changing, adding or removing of critical information.

R1.2.2 The system shall encrypt all sensitive data transferred over the network to ensure privacy and data security.



	UC1.1	UC1.2	UC1.3	UC1.4	UC1.5	UC2.1	UC2.2	UC2.3	UC3.1
R1.1									
R1.1.1	x								
R1.1.2		x							x
R1.2									
R1.2.1			x	x		x	x		
R1.2.2		x						x	

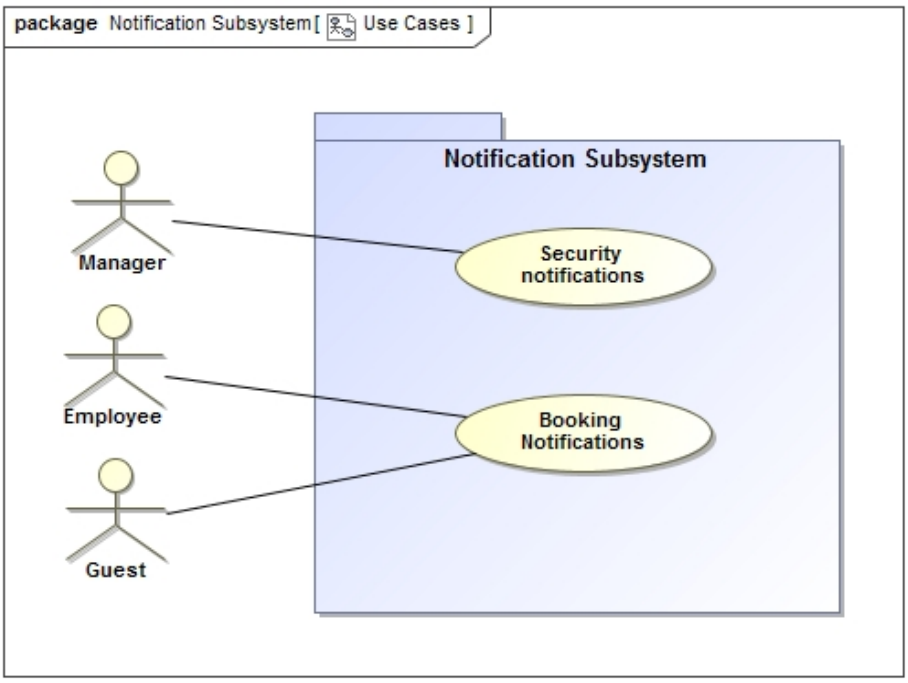
6.2 Notification

R2.1 The notification subsystem shall notify the admins of abnormal activity.

- R2.1.1 The system shall notify admins of system failures or errors unable to be handled during run-time.
- R2.1.2 The system shall notify admins of repeated attempts to access a room from an unauthorized individual.

R2.2 The system shall notify users when appropriate

- R2.2.1 The system shall notify a user if their access request was accepted or denied and if denied provide a reason.
- R2.2.2 The system shall notify a user if it is unable to resolve identification via facial recognition and prompt them to request an OTP.



	UC1.1	UC1.2	UC1.3	UC1.4	UC1.5	UC2.1	UC2.2	UC2.3	UC3.1
R2.1									
R2.1.1								x	
R2.1.2								x	
R2.2									
R2.2.1	x	x							
R2.2.2		x							

6.3 Client Information

R3.1 The Client Information subsystem shall act as an facade that manages other subsystem requests.

R3.1.1 The system shall forward requests to Authentication to ensure system integrity before executing the commands.

R3.1.2 The system shall contact the notification subsystem upon a requests completion, prompting the correct notification.

R3.2 The system shall handle errors thrown by the systems "core".

R3.2.1 The system shall take appropriate action upon having an error thrown,contacting the notification subsystem or re-trying the request.

R3.2.2 The system shall keep track of failures and failed access attempts, contacting notification to report issues where appropriate.

	UC1.1	UC1.2	UC1.3	UC1.4	UC1.5	UC2.1	UC2.2	UC2.3	UC3.1
R3.1									
R3.1.1	x	x	x	x		x	x		x
R3.1.2	x	x	x	x		x	x		x
R3.2									
R3.2.1	x	x	x	x		x	x		x
R3.2.2								x	

6.4 Database Management

R4.1 The Database Management subsystem shall provide all necessary CRUD functionality for the system.

R4.1.1 The system shall be extensible in its design and not take on business logic in its functionality.

R4.1.2 The system shall be responsible for interacting with external integrated systems such as office 365 to ensure cross-platform synchronization and seamless integration.

R4.1.3 The system shall be responsible retrieving tokens generated on external storage platforms such as Firebase.

	UC1.1	UC1.2	UC1.3	UC1.4	UC1.5	UC2.1	UC2.2	UC2.3	UC3.1
R4.1									
R4.1.1	x	x	x	x	x	x	x	x	x
R4.1.2	x	x	x	x	x	x	x	x	x
R4.1.3	x	x	x	x	x	x	x	x	x

6.5 Administration

R5.1 The Administration subsystem shall provide all functionality associated with employee management.

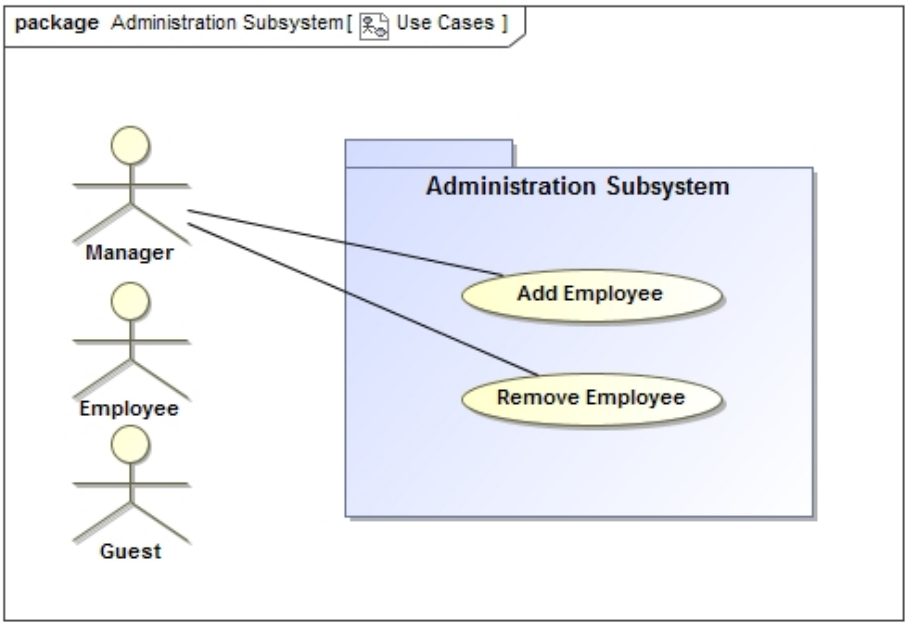
R5.1.1 The system shall provide the ability to add employees to the system, revoke or change employee access rights and set access rules.

R5.2 The Administration subsystem shall provide information about the activities of the system.

R5.2.1 The system shall allow the admin to view system logs, so that they may see who accessed a room, how they accessed the room and when they did.

R5.2.2 The system shall provide notifications to the admin, alerting them of system issues or unauthorized attempts to gain access to a room.

R5.3 The Administration subsystem shall ensure security by validating the credentials of those attempting to use the administrative subsystem, ensuring they have the proper authority.



	UC1.1	UC1.2	UC1.3	UC1.4	UC1.5	UC2.1	UC2.2	UC2.3	UC3.1
R5.1									
R5.1.1						x	x		
R5.2									
R5.2.1								x	
R5.2.2								x	
R5.3						x	x	x	

6.6 Reservation

R6.1 The Reservation subsystem shall provide the necessary information and functionality for booking a room.

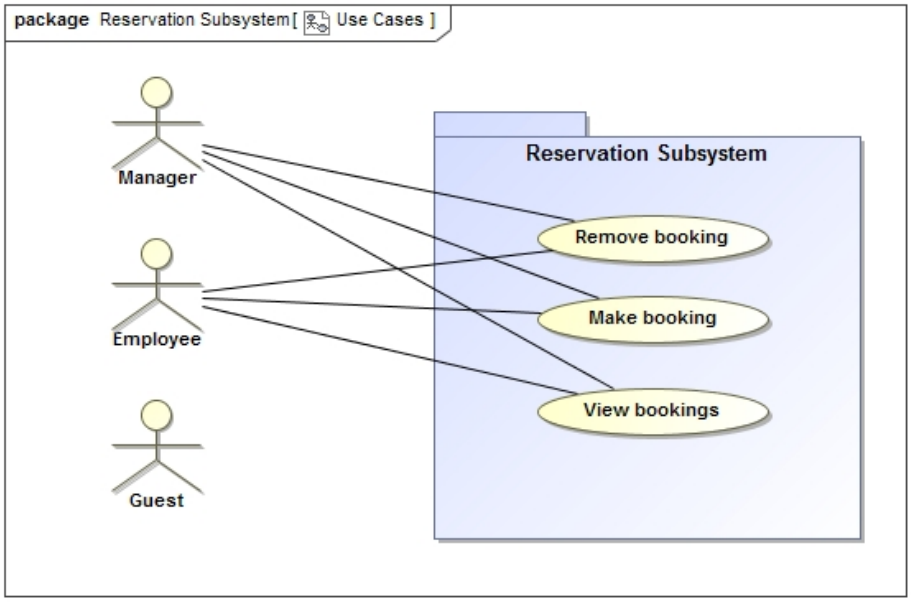
R6.1.1 The system shall show users what rooms are currently booked and which are open for booking on a specified date and time.

R6.1.2 The system shall allow users to book a room on a specified date and time, provided they have access rights for that room.

R6.2 The Reservation subsystem shall integrate with preexisting utilities such as office 365 calender, to ensure that current work flow and procedures in the company remain unimpeded.

R6.2.1 The system shall synchronize with bookings made on Google calender and reflect bookings made on external systems in this system.

R6.3 The Reservation subsystem shall work with the Authentication subsystem to ensure that access is granted or denied in a quick and efficient manner based off of current bookings.



	UC1.1	UC1.2	UC1.3	UC1.4	UC1.5	UC2.1	UC2.2	UC2.3	UC3.1
R6.1									
R6.1.1			x	x	x				
R6.1.2			x						
R6.2									
R6.2.1			x	x	x				
R6.3	x	x	x	x	x				

7 Non-functional Requirements

7.1 Quality Requirements

7.1.1 Security

- The system is required to adhere to the Protection of Personal Information Act (POPI) as well as adhere to the General Data Protection Regulation (GPDR) to protect the valuable information of employees and users.
- The system will provide a secure database to maintain the records of the employees of the organization while also maintaining strict access to users that have administrator rights.
- Access to the database through the system will be only accomplished by approved personnel (administrators) to allow a secure access stream to alter records
- One Time Pin's (OTP's) will only be administered to guests or clients that have been approved or invited by an employee of the organization.

7.1.2 Performance

- The proposed response time for the system to detect a face as well as allow or deny access should be below 5 seconds within the final product and below 10 seconds during development.
- The time taken to notify security or the person that had booked the relevant meeting room should be as efficient as possible to allow efficient detection.
- One Time Pin (OTP) generation will be provided in a timely manner to guests that wish to gain access.

7.1.3 Maintainability

- The system should allow effective maintainability through the use of a loosely coupled subsystems as well as modular programming.

7.1.4 Availability

- The system is expected to have an up time of at least 99% as it is a crucial part of the everyday work flow to the business.

7.1.5 Usability

- The system must be effective and friendly to the administrator to allow the simple procedure of adding employees or for users that wish to use the system to book a meeting room.
- It must also provide an effective learning curve as well as a sense of memorability for future use.

7.1.6 Reliability

- The system will need to provide accurate comparisons to allow access or to restrict said access.
- The system must also be maintained and provide its service with a high priority of uptime as to allow effective use.
- The system must also provide fault tolerance and be able to recover under failure and ensure uptime.
- Due to the system taking a component modular design, the independent subsystems will be more reliable as they will be loosely coupled.
- Appropriate error control procedures (error messages) will be implemented in a per subsystem approach to enforce a modular design.

7.1.7 Scalability

- The system will be able to process multiple users attempting to gain access into more than one meeting room at a single time.
- The system will be able to generate multiple One Time Pins (OTP's) to relevant guests that wish to gain access.
- The systems modular design allows the facial recognition functionality to be reused in a fully fledged access control system.

7.1.8 Portability

- The system will provide modules that may be used in other aspects to allow an effective integration into other systems.

7.1.9 Integration

- A loosely coupled system will be able to fit into other projects with ease as subsystems will be separate modules in a component based design. This will allow a high cohesion basis of communication between the subsystems.

7.1.10 Modifiability

- The subsystems will be able to be easily modifiable to implement changes in the future.
- The modular approach allows the subsystem to be altered with little compromise to other subsystems.

7.2 Constraints

7.2.1 Software Constraints

- The system is required to use the Angular Framework with regards to the front-end development
- The system is required to use NodeJS as the programming language for the back-end of the system
- The system is required to use Firebase as a real time database strategy to efficiently match the facial data against that retrieved from Firebase
- The system is required to make use of the Firebase database in regard to storing facial data for all employees by submitting data added by administrator users
- The system is required to generate One Time Pin's (OTP's) that will be limited to a four digit combination
- The system will be integrated with API calls through Google as well as Microsoft and must be able to cater between the relevant client systems
- The system must comply with POPI regulations
- The system must be integratable into the EPI USE network system

7.2.2 Hardware Constraints

- Standard cameras from web cameras to specifically build camera's will be used for the facial capture
- A user will have to stand within a meter from the camera for an accurate image capture

8 System trace-ability matrix

	Authentication	Notification	Client Information	Database Management	Administration	Reservation
UC1.1	x	x		x		x
UC1.2	x	x	x	x		x
UC1.3		x	x	x		x
UC1.4		x	x	x		x
UC1.5			x	x		x
UC2.1		x	x	x	x	
UC2.2		x	x	x	x	
UC2.3		x			x	
UC3.1	x	x	x	x		x

9 Technologies

9.1 Version Control: Git (Hosted on github.com)

We chose git and Github as it is easy to work with while still providing us with all the functionality we need for our version control purposes. As an added bonus our team is experienced and comfortable with git.

9.2 Project Management: Zenhub

Once again the seamless integration into Github combined with tons of nice to have features and an easy to use interface made us choose to use Zenhub for our project management.

9.3 Continuous Integration: Travis CI

Travis CI provides a free service to open source projects, integrates seamlessly with Github and Slack (What we use for our team communication). It is also very user-friendly allowing us to dedicate more time to actual development.

9.4 Front-End Framework: Angular 7

We decided to use Angular as it reduces the amount of code needed for the front end development allowing us to spend more time fine-tuning our system.

9.5 Data Storage: Firebase/Firestore

We chose firestore as it is a real-time database and provides us with the speed we need to make the facial recognition as fast as possible.

9.6 Face Detection: OpenCV (For now)

We aim to use OpenCV as it supports many different languages and should be very quick to detect faces. As the project progresses this might be changed depending on performance and integration requirements.

9.7 AI Facial Recognition: EasyFaceNet (For now)

We plan to use EasyFaceNet as we have limited experience in AI and building a good facial recognition AI can take months. In order to not reinvent the wheel we looked into FaceNet which provides a TensorFlow implementation for facial recognition. Further research revealed EasyFaceNet which provides a really easy interface to FaceNet and promises to allow for very good facial recognition without having in-depth knowledge of Deep Neural Networks. Facenet might however be used if we feel like EasyFaceNet does not provide the performance or accuracy that we need.