Abstract—The attendance of students in Instituto Superior Técnico is currently collected manually using attendance sheets. On the other hand, looking for unoccupied rooms to study requires students to physically check one room at the time until they find one suitable in terms of occupancy. These two tasks require a lot of human intervention and time that could be better spent in academic activities. In this document, we propose a solution to address these problems in the context of an academic institution. Our proposal was implemented as a prototype, used for its validation. Through building a prototype that used indoor positioning technologies based on Wi-Fi, we were able to automate the student attendance registration and the estimation of the number of occupants in rooms used by the student population. The prototype was developed using open source technologies and took advantage of the most common handheld systems currently used by students and teachers and the existing Wireless Local Area Network of the university campus where it was deployed. We performed experiments to verify the correctness of the location estimations generated by the prototype and its corresponding performance when exposed to an increasing number of potential users. The obtained results showed that the prototype was able to generate room-level location estimations and it could support at least 7000 users over the duration of 1 minute. We also provide an overview of the related work in the areas of location-based services, indoor positioning systems and systems that manage the attendance of students.

1. Introduction

Nowadays, we spend around 90 percent of our time indoors\(^1\). Knowing about this reality, many innovative companies have begun to offer services specially designed for indoors environments.

On the other hand, more than 60 percent of the global population have access to mobile devices, wherein more than one third of these devices are smartphones\(^2\). Not only smartphones but also tablets and other handheld systems offer a wide range of sensors and connectivity options, with special emphasis on Wireless Fidelity (Wi-Fi) and Bluetooth (BT) - two of the technologies that have been explored in building positioning systems for the indoor environment.

In order to take advantage of the increasing smartphone penetration in the student and staff population, academic institutions have already started launching mobile applications that enable their users to access a wide range of academic services [1]. However, these previously referred technologies are not fully explored in higher education institutions since there are still tasks being performed manually that could benefit from them.

1.1. Motivation

At the current time, in lectures under the responsibility of Departamento de Engenharia Informática (DEI)\(^3\), the attendance of students is mainly collected through an attendance sheet that is passed from hand to hand among the students during the lecture. This sheet usually has the full name and the identification number of the students that are supposed to attend the lecture, requiring each student to sign next to his identification information. Although this practice is voluntary in the majority of the lectures, being just mandatory in laboratory classes, teachers also need to report the total number of students that have participated in the lecture for statistical purposes. This task requires a lot of human intervention, which consumes time that could be better spent in lecture activities.

On the other hand, Instituto Superior Técnico (IST) has a policy of letting its students use unoccupied class rooms, i.e. rooms that are not being used for academic activities at that time, as study rooms. Currently, a student needs to visually check the schedule that is displayed next to each class room’s door to see if it is not being used for classes, and then he also needs to check if the room’s occupancy is acceptable for him.

The previous two cases have a common need: the estimation of the number of occupants at a given location.

\(^{3}\)https://fenix.tecnico.ulisboa.pt/departamentos/dei
Although these cases happen in the particular case of IST at the current time of writing, they could also apply in other higher education institutions not only in Portugal but at a global level.

1.2. Proposed Solution

Considering the aspects referred above, this dissertation addresses the problem of collecting student attendance at lectures and estimating the number of occupants in class and study rooms. As a solution, we propose a low-cost system, the AtOcu (Attendance and Occupation) platform, that makes use of the most common handheld systems currently used by students and teachers to provide services based on their location to automate the student attendance registration and the room occupancy estimation.

1.3. Thesis Contribution

In the course of the development of this dissertation, we have designed, implemented and evaluated a prototype of the AtOcu platform that automates the student attendance registration and the room occupancy estimation. This prototype was developed for the IST - Taguspark scenario: it interacts with the local academic management system, the FenixEdu\(^4\), and is integrated with a positioning system designed for indoor environments, the SmartCampusAAU platform, that takes advantage of the existing Wi-Fi infrastructure. Globally, the prototype contributed to explore and determine the suitability of building services based on location to register attendance and estimate room occupancy on the top of an indoor positioning system based on Wi-Fi. However, the overall proposed solution can be extended to other scenarios and education institutions where similar situations may occur, specifically involving the abovementioned services.

1.4. Outline

This document is structured as follows. Section 2 surveys previous work in the field of Location-Based Service (LBS), indoor positioning and attendance and occupancy systems. A description of the proposed architecture is presented in Section 3. The process and the choices made during the implementation of the proposed architecture are described in Section 4. Section 5 describes the set of tests performed over the implemented solution and the corresponding results. Finally, a summary about the research and work developed, as well as future work, is presented in Section 6.

2. Related Work

2.1. Location-Based Services

Services that take into account the location of an entity, or multiple entities, are known as LBSs [2]. A location can be described as a symbolic or physical type, and as an absolute or relative type [3]. Symbolic location is described using terms easily understood by humans while physical location is described using coordinate systems. On the other set of terminologies, an absolute location is described using a shared reference point among all locations, while a relative location is described according to its own frame of reference.

2.1.1. Architecture. LBSs can be developed by adopting a layered architecture. An example of this architecture is presented in Figure 1. This architecture comprises six layers:

1) The Sensors layer is responsible for detecting a variety of physical phenomena and collecting raw data.
2) The Measurements layer transforms the raw data collected from sensors into measurement types.
3) The Fusion layer uses the measurement information to determine the target’s location information.
4) The Arrangements layer infers the spatial relationships between the detected targets.
5) The Contextual Fusion layer combines location information, arrangement information and contextual information.
6) The Activities layer adds semantic information to the contextual information in order to make inferences about the state of targets.

2.1.2. Security and Privacy. Regulatory strategies, privacy policies, anonymity and obfuscation (also known as entropy) are methods that can be used to ensure location privacy in LBSs [5]. Regulatory strategies are often based on government rules on personal information and its use. Privacy policies are agreements between the users of the services and the entity that manages their location information data. Anonymity can be enforced by using pseudonyms and techniques of grouping targets to create ambiguity. Finally, obfuscation is a method that focus on reducing the quality of the location information data.

2.2. Techniques for Indoor Positioning

Indoor Positioning Systems (IPSs) use techniques as triangulation, fingerprinting, proximity and vision analysis to provide location information [6], [12].

2.2.1. Triangulation. To estimate the target position, triangulation uses geometric properties. Specifically, there are two types of triangulation, lateration and angulation, and these techniques need to estimate the angles and the distances, correspondingly, among the target object and a set of reference points [6].

2.2.2. Fingerprinting. Fingerprinting, also known as scene analysis, is a pattern matching technique consisting of two stages - offline and online stage - and it is applied to Radio Frequencies (RFs) technologies. Firstly, it is necessary to build a radio map of the site, which is performed during the offline stage. It consists of collecting a set of signal features that are location dependent (also known as fingerprints) such as Received Signal Strength (RSS). This map establishes a relationship between location coordinates and the signal strength from reachable reference points. While using the IPS, in the online stage, the location of the target is estimated by matching live signal strengths against the radio map that was previously built.

2.2.3. Proximity. Unlike the previous described techniques, instead of detecting the location of the target, the proximity technique detects its closeness to proximity sensors, i.e. the target is detected within a limited range from the sensor.

2.2.4. Vision-Analysis. Visioning positioning is a technique that estimates locations from images received by one or multiple reference points [8].

2.3. Indoor Positioning Systems

An IPS is a system that provides location information that could be used to develop LBS. IPSs can be implemented using two different architectures, depending on where the location information is produced: self-positioning or infrastructure positioning architectures. In self-positioning architectures the target’s location is estimated by the target itself aided by the infrastructure, while in infrastructure positioning architectures the target’s location estimations are made by the infrastructure from the data it receives from the target [8].

2.3.1. Wi-Fi. The Wi-Fi is a Wireless Local Area Network (WLAN) technology that has been explored in recent years to build positioning systems. IPSs based on Wi-Fi have the advantage, in most of the cases, of using an already existing infrastructure for data communications, namely the Access Points (APs) of the WLAN as reference points. WLAN-based systems usually use trilateration or fingerprinting techniques to estimate positions.

One of the first examples of an IPS using WLAN was RADAR [9]. Another example is the SmartCampusAAU [10] platform. It offers an open software platform that supports the creation of LBSs by taking advantage of the existing WLAN.

2.3.2. Bluetooth. BT is a wireless technology standard that is suitable for exchanging data over short distances. Today, BT version 4 includes not only the classic BT protocol, but also the Bluetooth Low Energy (BLE) protocol. The latter is the most suitable for indoor positioning purposes mainly because of its low power requirements and low cost. Notwithstanding, there are IPSs based on the classic BT protocol, for instance the Topaz location system [11].

2.3.3. Infrared. Similarly to other wireless technologies, Infrared Radiation (IR)-based systems also need a transmitter and a receiver. The transmitter, an IR emitter, is carried by the target of the system, and it is usually a device capable of emitting an unique signal that can identify its user [12]. After being detected by a receiver, the signal is interpreted and the location of its transmitter is estimated taking into account the location of the receiver.

2.3.4. RFID. Radio Frequency Identification (RFID) is often used to track and identify objects. This technology is often applied using proximity techniques, involving RFID tags and readers. RFID tags can be either active or passive [13]. Active tags have a dedicated power supply, allowing them to send signals up to ranges of 100 meters. Passive tags, on the other hand, do not have dedicated power supplies, which requires them to be powered and activated by signals emitted by external devices (the reader). For this reason, passive tags work at a shorter range than active tags.

2.3.5. Vision-Based. Two types of vision positioning systems that can be used to estimate the indoor location of target objects were identified. The main difference between both types is basically the use, or not, of tags. TRIP is an example of an IPS that use tags to assist the identification of target objects [14]. This system can be paired with inexpensive cameras (e.g. web-cams) and it is able to provide the location and the orientation of the tagged target. On the other hand, other vision-based systems do not require a tag to be attached to the target, for instance the system proposed by Stillman, Tanawongsuwan and Essa [15].

2.4. Attendance and Occupancy systems

2.4.1. Bluetooth-based. BT is the wireless technology chosen in [16] to develop MITSAT, which is a student attendance tracking system. This system requires each student to use a BT transmitter with an unique identifier to interact with BT receivers attached to APs spread across the indoor environment. The receivers detect that a student has entered a class room by using the Received Signal Strength Indicator (RSSI) measurement. [17] presents an intelligent lecture assistant that provides a solution for the scenario of
class rooms attendance. This assistant involves two modules installed in the lecturer’s and in the students’ handheld devices. The attendance information is taken by the lecturer using his device, which requires that each of the students’ devices previously had connected to his own through Wi-Fi or BT.

2.4.2. RFID-based. RFID is one of the most used set of wireless technologies to build attendance systems. Smart Attendance System [18] is a web-based application that makes use of RFID technology to simplify attendance recording in combination with relational databases that store the attendance information. This system defines different levels of access to the attendance information in terms of user main role (e.g. student, lecturer or staff), and offers additional functionality, apart of attendance recording, such as notes distribution and reminders. Similarly, the attendance system proposed in [19] also makes use of the same base technology, requiring students to flash their student identifiers to a RFID reader upon entering a class room and afterward the attendance records are made available online so that the lecturers can access them.

3. Architecture

We propose the architecture shown in Figure 2. The system comprises three basic components: an IPS, the AtOcu client and the AtOcu server.

3.1. AtOcu Client

The AtOcu client, detailed in Figure 3, runs in the handheld system of each user, student or teacher, and it is mainly responsible for:

- Estimating the location information of its own user with the assistance of the IPS;
- Contributing to the occupancy information collection effort;
- Collecting the attendance of its user;
- Informing the user about his current location;
- Providing room occupancy information;
- Providing the number of attendees during class;
- Informing the user about his current or next classes.

By estimating the user’s location information at the handheld system level, we are opting for a self-positioning architecture.

The AtOcu client begins operating once the user logs in by supplying his credentials through the Interface module. This module manages a Graphical User Interface (GUI) that allows the user to interact with the AtOcu client and visualize the contextual information such as: user’s current location information; schedule of the actual and remaining classes of the user; occupancy information of the rooms located on the proximity of the user; current status of the student, e.g. if the student is currently attending a class.

Once the user is logged in, the AtOcu client request the user’s schedule from the AtOcu server. This particular communication is established through the Communication module, which comprises the necessary logic to interact with the AtOcu server and interpret its responses. More details about the communication between these two components are given later in Subsection 3.3. The user’s schedule comprises a variable number of events associated with the courses the user is enrolled in and it is the fundamental data resource for the attendance management functionality.

To perform location estimations, the AtOcu client interacts with an IPS that uses the sensors commonly available in handheld systems in order to estimate location information. It should be noted that the AtOcu client is limited to perform location estimations within the physical limits of its user’s university campus, and therefore, once the user is detected on campus, the AtOcu client starts producing location estimations until the user leaves the campus.

The Persistence module manages the access to the local database where the entities required by other modules are
The Context module contextualizes the location information previously gathered by the Location module with the user’s schedule so that the AtOcu client can determine which action must be performed. When inside the campus but outside scheduled classes, the AtOcu client is periodically estimating the current location of its user, which is then anonymized and sent to the AtOcu server through the Communication module - this information is used to reflect the general occupancy data of the whole campus.

On the other hand, the Context module also organizes the user’s schedule that it has received from the Communication module in order to analyze the date and time of the upcoming classes, as well as the location where they will take place, so that it can determine if the user is attending or missing the scheduled class. In case the user is attending the scheduled class, the Context module sends the attendance information to the AtOcu server through the Communication module. If the user is found to be missing the scheduled class, the Context module postpones the verification of the user’s location, without exceeding the duration of the class.

3.2. AtOcu Server

The AtOcu server, detailed in Figure 4, is a central entity mainly responsible for:

- Gathering the users’ schedule information;
- Managing the users’ credentials;
- Storing and aggregating the data generated by AtOcu clients.
- Providing a global and aggregated view over the occupancy and attendance data previously collected;

The Communication module allows the AtOcu server to direct requests to the Academic Management System in order to have access to its users’ schedule information. The characteristics of this module are dependent of the Academic Management System being used in the university where the AtOcu platform is to be used at.

The access to the local database is done through the Persistence module.

The Application Programming Interface (API) module offers the interface through which the AtOcu client sends requests to the AtOcu server.

The Entity module concentrates all the logic to manage the entities that are being stored in the local database and that are the core data that enables the AtOcu platform to offer the proposed LBSs. After new entities are created, they are sent to the Persistence module in order to be stored in the local database.

Among the requests the AtOcu server is supposed to receive through the API module, the emphasis lies on the requests directly involved with occupancy and attendance data. The AtOcu server is expected to receive anonymous location updates with a certain periodicity from various AtOcu clients. These updates are sent to the Entity module so that proper entities could be created to represent such updates. Then, instead of sending these new entities to the Persistence module for long-term storage purposes, they are sent directly to the Cache module. The Cache module is responsible for retaining data for a certain time, determined by an expiration time. The expiration time guarantees that the user contribution to the occupancy count is discarded if its AtOcu client does not send another update within a certain period of time.

On a regular basis, the Entity module executes a process that takes a snapshot of the current occupancy status of the university campus. The process requests from the Cache module the cache entries related to the occupancy data so that it can create entities to represent the aggregated information. This information is then sent to the Persistence module for storage.

Differently from the management of the occupancy data, upon receiving requests from the various AtOcu clients to store attendance data, the Entity module creates entities that represent attendance records so that they can be sent to the Persistence module for storage.

Ultimately, the Interface module allows its users, the administrator of the AtOcu platform or building management personnel, to access the aggregated attendance and occupancy data being collected by the AtOcu platform through a GUI.

3.3. Communication

The proposed solution uses the Internet as the communication medium between the various clients and the centralized server. The AtOcu server adopts the Representational State Transfer (REST) architectural style and makes use of Hypertext Transfer Protocol (HTTP) to provide a network-based API to the AtOcu client.

As previously presented in Subsection 3.1, the occupancy and attendance data is generated in the AtOcu client and then sent to the AtOcu server. To protect the privacy and the integrity of the data being transmitted, the communication must be established in a secure fashion, since the communication medium is not controlled by our solution.
Figure 5. High level architecture of the implemented prototype.

For that purpose, we use the Transport Layer Security (TLS) protocol to encrypt the connections established when the AtOcu client accesses the API provided by the AtOcu server. Using TLS allows us to protect the privacy and the integrity of the occupancy and attendance information being sent to the AtOcu server by each of the AtOcu clients, which also helps preserve the anonymity of the users of the system. Furthermore, TLS can be used to guarantee that the AtOcu client just exchange data with the AtOcu server - by identifying the AtOcu server with a TLS certificate.

4. Implementation

A prototype of the AtOcu system was developed considering the Taguspark campus of IST. The AtOcu client and the AtOcu server were developed, respectively, for the Android Operative System (OS) and using the Play Framework. The Academic Management System on place is the FenixEdu system and we have integrated the SmartCampusAAU platform into the proposed solution as the chosen IPS.

The FenixEdu system is used locally for academic and administrative management, which allowed us to obtain information about courses, their schedules and enrolled students. This information is accessible through a web API, the FenixEdu API.

We have chosen to use the SmartCampusAAU platform as IPS, since it supports self-positioning out of the box, offers room-level accuracy and it is based on Wi-Fi, which makes it possible to take advantage of the existing WLAN infrastructure on campus.

4.1. AtOcu Client

4.1.1. Location module. The implemented Location module comprises a started service called LocationServiceCapsule and a class called SmartLocationManager.

The indoor positioning functionality of the SmartCampusAAU platform is exposed through the LocationService class of its library, which is a service itself. Through this service, we are able to download the radio map and start and stop indoor positioning. Given that the AtOcu client needs to periodically estimate the current location of its user even when the user is not interacting with its GUI, we created the started service LocationCapsule that binds to LocationService and runs in the background.

To make the Location Capsule service gather location estimations on the background, we used the AlarmManager class - it is available on the Android platform and provides access to the system alarm services, allowing us to schedule an alarm responsible for starting the Location Capsule service. These alarms are scheduled when the user is detected on campus and they were configured to run periodically, with the periodicity being equal to 60 seconds.

The Location Capsule service registers a listener with the LocationService to receive location notifications that are then used to update the internal state of the Smart Location manager.

The user is detected on campus when his handheld system connects to the local WLAN.

Every time the Smart Location manager receives a new location notification, it retrieves its symbolic name from the Persistence module and uses the Communication module to send the location update to the AtOcu server. This notification aims at contributing to the current occupancy of the location in which the user was detected.

4.1.2. Context module. The implemented Context module comprises a started service called AttendanceService, the ReschedulerReceiver broadcast receiver and the classes AttendanceManager and ContextualManager.

The Contextual manager manages the current contextual interpretation of the user’s status. This status can tell us if the user is attending or missing a scheduled class or instead, if he is outside of classes.

To collect the attendance of an user at a particular scheduled class, we firstly need to check if the user’s location matches the location where the class takes place during its duration. For that purpose, we used the AlarmManager, similarly to how we solved the gathering of location estimation in the background. The Attendance manager is used to get the scheduled class that will happen sooner from the Persistence module and then to schedule an alarm to start the Attendance service to check the user’s attendance at that particular class.

When the Attendance service starts, it waits until the requested location estimation is generated and through its symbolic location it can check if the user’s location matches the location where the class is supposed to be taking place.

In case there is a location match, the Attendance service notifies the Contextual Manager that the user is currently...
attending a class and then requires the Persistence module to change the state of the respective check-in event to attended. In case there is no location match, the Attendance service uses the Attendance manager to reschedule the alarm for later and the Contextual manager is notified that the user is missing a class.

4.1.3. Communication module. The implemented Communication module comprises the started service AtocuSynchronizerService and the class AtocuApiManager.

The AtocuApiManager class is an intermediate class used by other modules to request the AtOcu server through its REST API. It offers a custom HTTP client configured to use TLS.

The AtocuSynchronizerService service is started right after the user has logged in to request institutional, personal and schedule information from the AtOcu server as well as the mapping between absolute and symbolic locations. On the other hand, the AtocuSynchronizerService service is also started while the user is attending a class in order to request the AtOcu server to store his corresponding attendance.

4.1.4. Persistence module. The Persistence module is the combination of the ActiveAndroid Object Relational Mapper (ORM) with the classes Course, SymbolicLocation and CheckInEvent. The check-in events represent locally the scheduled classes that are received from the AtOcu server. The courses entities represent the courses the user is enrolled in or teaching, depending on his role. The symbolic locations are entities that represent the mapping between the absolute locations, represented by geographic coordinates, and a symbolic name.

4.1.5. Interface module. The Interface module comprises two activities, the login and main activities.

The login activity, implemented in the class LoginActivity, is responsible for verifying if the user is logged in into the AtOcu client; if not, it then leads the user through the login process that uses the authentication and authorization mechanisms that exist in the FenixEdu system.

After the login process is finished, the AtOcu client receives the identity and anonymity tokens from the AtOcu server. The identity token is used to access endpoints of the API that require the user to be identified at the server side, e.g. while requesting schedule information. The anonymity token was created to be used to access the API endpoint that requests the AtOcu server to register that a user was detected at a given location, i.e. to contribute to the room occupancy effort.

The main activity, implemented through the MainActivity class, manages a set of fragments to structure the GUI through three screens: the status screen, the rooms screen and the classes screen.

The status screen is the first screen that the user sees after initiating the AtOcu client. In this screen, the user is able to consult personal and contextual information, which includes the user’s status. The user’s status changes to show that user is currently attending or missing a class.

The classes screen lists the next classes the user has to attend on his schedule and the current number of students that are attending a class. This screen also reacts to status updates, namely by changing the color of the current scheduled class that is taking place. If the user is attending the scheduled class, its representative item listed on the top of classes screen is displayed in green and red otherwise, i.e. if the user is missing the class.

The rooms screen lists all the available rooms that are at a distance up to 10 meters of the last location in which the user was detected. In this screen, we also used colors to differentiate all the rooms being shown in terms of the ratio between the room’s current occupancy and capacity so that the user can easily decide which room he wants to use.

4.2. AtOcu Server

The development of the AtOcu server took advantage of the architectural organization, specifically through the Model-View-Controller (MVC) pattern, offered by the Play Framework. It provides the classes Model and Controller that encapsulate, respectively, the logic of the models and controllers components and a template engine to generate views.

4.2.1. Communication module. The Communication module’s logic was encapsulated in the class FenixEduApiManager. The FenixEduApiManager class configures the API client provided by the official FenixEdu Software Development Kit (SDK) with our OAuth 2.0 client credentials obtained when registering the AtOcu platform with the FenixEdu instance. These credentials are used in every requests that attempt to invoke private endpoints of the FenixEdu API that involve user's personal and courses information.

4.2.2. Cache module. The Play Framework provides a cache API so we did not need to implement one from scratch. In the AtOcu server implementation, we used cache to store temporarily HTTP responses that are common to multiple AtOcu clients, such as the mapping between symbolic and absolute locations, and to retain anonymous location updates received from each active AtOcu client.

4.2.3. Entity module. Each one of the entities being managed by the AtOcu server were implemented as models in the Play Framework. These models, classes that extend the Model class, encapsulate all the logic to create, update, delete and retrieve the entities from the Persistence module.

The users of the AtOcu platform are represented as user entities on the AtOcu server side. Classes, shifts and courses are entities that combined represent the user’s schedule and are used to support the attendance registration LBS. Locations entities represent both rooms and passages. Essentially, a passage location is a place through which the
user passes by when entering or leaving a room. A location entity is created after its respective physical location is mapped by the SmartCampusAAU mobile application and sent to the AtOcu server through its API module. Occupants entities were designed to represent anonymous users that are contributing to generate room occupancy estimations.

Right after the AtOcu server is deployed, the system schedules a job to perform the creation of occupancy snapshots for all the locations available and it is implemented to run at every new hour. These snapshots take into account the maximum and minimum occupancy values obtained in the last hour.

4.2.4. Persistence module. On the server side architecture of the AtOcu platform, we used the Ebean ORM as the Persistence module.

4.2.5. Interface module. The Interface module of the AtOcu server was implemented through a web GUI, the AtOcu website, that allows its users, the administrator of the AtOcu platform or building management personnel, to use a Web browser to access some of the information generated by the AtOcu platform. The web GUI comprises a set of views that were implemented in HyperText Markup Language (HTML), Cascading Style Sheets (CSS), Javascript and Scala programming languages.

The AtOcu Website comprises six pages and its site map is represented in Figure 6.

To access the AtOcu website, the user needs to supply an username and password. We implemented a simple authentication scheme instead of the login scheme adopted in the AtOcu client because the AtOcu platform’s administrator or the building management personnel might not have IST credentials. After a successful login, the user is redirected to the overview page, which allows the user to consult general numbers such as the number of: registered and active users; courses being monitored; classes that were lectured; rooms currently mapped. The user can then access the listed pages to consult the locations mapped and their respective occupancy history, and the courses being monitored and their respective attendance records.

4.2.6. API module. The API module comprises a Router component and the Locations, Users, Courses and Classes controllers that are responsible for generating the responses to the requests received from AtOcu clients and from web clients accessing the AtOcu website. Each one of these controllers have the specific methods to process the requests towards the entities with the same name.

The Router is a component of the Play framework that is used to intercept incoming HTTP requests and invoke the corresponding actions that are implemented by controllers. All the aforementioned controllers extended the Controller class of the Play Framework so that they can be recognized by the Router component.

The set of all endpoints constitutes the web API provided by the AtOcu server, that we called AtOcu API. This API is consumed by the AtOcu client and by the web client that accesses the AtOcu website.

Whenever the AtOcu client invokes any of endpoints of the API, it needs to include the entity token or the anonymity token (when registering an occupant) in the query parameters, otherwise its request is ignored.

4.3. Security

In order to enable Hypertext Transfer Protocol Secure (HTTPS), we deployed the AtOcu server with Nginx as a front end web server. To avoid additional costs with the development of this prototype, we decided to generate a self-signed TLS certificate for the AtOcu server. The certificate served then to enable HTTPS with Nginx, and since it is a self-signed certificate, we also needed to embedded it on the Atocu client so that the client can verify and trust the AtOcu server.

5. Evaluation

5.1. Positioning Tests

The positioning tests were designed to evaluate the correctness of the location information produced by the implemented AtOcu prototype in combination with the SmartCampusAAU solution.

These tests were performed in two scenarios in which two types of contiguous rooms of different sizes were selected - specifically offices and lecture halls. Before proceeding with each test, we mapped the rooms and the passage locations that are part of their surroundings with the SmartCampusAAU app.

After creating the radio map, we defined 18 and 108 stationary positions that covered, respectively, the entire floor area of each office and lecture hall. To perform a positioning test, we moved sequentially to each one of the defined positions and collected a location estimation using the AtOcu client while staying stationary. This test was performed five times for each scenario.

The first scenario involved two contiguous offices, office 2-11-3 and office 2-11-5, on the second floor of the IST - Taguspark main building (Figure 7). In the first scenario, we obtained correct estimations in all tests in 50% of the
defined positions for the office 2-11-3 and in 55.56% of the defined positions for the office 2-11-5.

The second scenario involved two contiguous lecture halls, A4 and A5, on the ground floor of the IST - Taguspark main building (Figure 8). In the second scenario we obtained substantially better results, given that the system generated correct estimations for all tests in 90.74% and 89.81% of the defined positions for, respectively, the lecture hall A5 and lecture hall A4.

The results obtained for the positioning tests demonstrated that the performance of the system was not uniform across the whole area of the rooms tested. However, the positions in which the system performed poorly were relatively near passage points such as doors where the user is not expected to stay for long periods of time so that the system does not detect him as an occupant. For this reason, and considering that in the majority of the positions tested the system produced a correct location estimation, we consider that the results obtained confirm that the AtOcu platform reached room-level accuracy when integrated with the SmartCampusAAU platform.

5.2. Load tests

Although the location information is estimated at the client level, as well as the core of the business rules of the AtOcu platform, the AtOcu server is still required to serve a variable number of AtOcu clients as a resource provider and as a central repository. Therefore, we decided to test this central component of the proposed system to analyze how it behaves under an increasing number of clients. Particularly, we tested the server to support at least half of the previously mentioned number.

We used the Loader® cloud-based load and scalability testing service. Using this service, we ran a set of tests where we specified the total number of clients to invoke the respective API endpoint over the duration of 60 seconds. Each one of the endpoints was tested from 1000 clients to 7000 clients over the duration of 60 seconds and we collected the average response time in milliseconds of each test.

We obtained the results presented in Figure 9. The average response times of the endpoint that returns a cached response were relatively constant during the course of the tests, varying between 80 and 81 milliseconds. Although slightly higher, the average response times of the tests performed for the endpoint requiring the AtOcu server to write or update values from the database did not change significantly, varying between averages response times of 85 to 94 milliseconds.

The only exception in these tests were observed for the third endpoint, where the average response times showed a rapid increase between 5000 and 6000 clients per test, from 337 to 1553 milliseconds and reached its upper limit at 7000 clients per test with an average response time of 1934 milliseconds.

The obtained results showed that the AtOcu server is able to scale to respond to the requests produced by a number of clients that is higher than the actual number of students and teachers located at the Taguspark campus of IST.

5.3. Client Response Time tests

We also performed a response time test targeting the endpoint /locations/occupancy/ by using different handheld systems running the AtOcu client, to obtain a more accurate result from the client perspective.

The average response times obtained by the smartphone and the tablet were, respectively, 80 and 89 milliseconds.

6. Conclusion

This dissertation addressed the problem of collecting student attendance at lectures and estimating the number of occupants in class and study rooms. To solve this problem,
we designed the architecture of the AtOcu platform to be modular, extensible and IPS independent. A prototype was then developed for the IST - Taguspark scenario that took advantage of its WLAN and an existing IPS, the SmartCampusAAU solution, to estimate the location of its users and consecutively estimate room occupancy information. On the other hand, the prototype was also able to interact with the FenixEdu platform, a local management system, to obtain schedule information that was combined with the user’s location information to provide the attendance registration service. In order to evaluate the proposed solution, we developed a set of experiments that were applied to the prototype in order to investigate the correctness of the its location estimations and the performance of its main components. The obtained results showed that the prototype was able to generate room-level location estimations and that it could scale to accommodate an increasing number of users using its services. Considering the evaluation of the prototype, we concluded that the propose solution showed that IPSs, specifically those based on the Wi-Fi technology, can be utilized to build LBSs that automate the attendance registration and the room occupancy estimation in the context of higher education institutions.

References


Figure 9. Results of the load tests for the AtOcu server.