Attendance Registration and Occupancy Estimation using Indoor Positioning Systems

Fábio André Raposo Ribeiro

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Prof. Fernando Henrique Côrte-Real Mira da Silva
Prof. Ricardo Jorge Feliciano Lopes Pereira

Examination Committee

Chairperson: Prof. Paolo Romano
Supervisor: Prof. Fernando Henrique Côrte-Real Mira da Silva
Member of the Committee: Prof. Miguel Filipe Leitão Pardal

November 2015

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.

Keywords: location-based service, indoor positioning system, wi-fi, attendance registration, room occupancy estimation

Resumo

A recolha de presenças dos alunos do Instituto Superior Técnico é feita através da utilização de folhas de presenças. Por outro lado, para procurar por salas livres, os alunos têm que se deslocar sala a sala e verificar se as mesmas têm um número aceitável de ocupantes. Estas duas tarefas consomem tempo e recursos humanos que poderiam ser melhor utilizados em actividades académicas. Neste documento, propomos uma solução para endereçar estes problemas no contexto de instituições académicas. A nossa solução foi implementada e consequentemente validada através de um protótipo. Através do desenvolvimento de um protótipo que recorreu a tecnologias baseadas em Wi-Fi, conseguimos automatizar o registo de presenças dos alunos e a estimação do número de ocupantes em salas que são usadas pela população estudantil. O protótipo foi desenvolvido através da utilização de tecnologias de código aberto e tirou partido dos dispositivos móveis que são comummente utilizados por estudantes e professores, e da rede local sem fios existente no campus universitário para o qual foi desenvolvido. Foram realizadas experiências para verificar a correcção das estimativas de localização geradas pelo protótipo e o seu respectivo desempenho quando exposto a um número crescente de utilizadores. Os resultados obtidos mostraram que o protótipo foi capaz de gerar estimativas de localização adequadas a salas e conseguiu suportar, pelo menos, 7000 utilizadores distribuídos ao longo de um minuto. Foi ainda descrito o estado de arte de serviços baseados em localização, sistemas de posicionamento interiores e sistemas que gerem as presenças de estudantes.

Palavras-chave: serviço baseado em localização, sistema de posicionamento interior, wi-fi, registo de presenças, estimativa de ocupação de salas

Contents

Abstract iii

Resumo v

List of Figures xi

List of Tables xiii

Acronyms xv

1 Introduction 1

1.1 Motivation . . . 1

1.2 Proposed Solution . . . 2

1.3 Thesis Contribution . . . 3

1.4 Outline . . . 3

2 Related Work 5

2.1 Location-Based Services . . . 5

2.1.1 Categories . . . 5

2.1.2 Architecture . . . 6

2.1.3 Security and Privacy . . . 7

2.2 Techniques for Indoor Positioning . . . 7

2.2.1 Triangulation . . . 8

2.2.2 Fingerprinting . . . 9

2.2.3 Proximity . . . 9

2.2.4 Vision-Analysis . . . 9

2.2.5 Dead reckoning . . . 9

2.2.6 Comparison . . . 10

2.3 Indoor Positioning Systems . . . 11

2.3.1 Indoor Positioning System Architecture . . . 11

2.3.2 GPS and cellular-based systems . . . 11

2.3.3 Wi-Fi . . . 12

2.3.4 Bluetooth . . . 12

2.3.5 Infrared . . . 13
2.3.6 RFID . . . 13
2.3.7 Zigbee . . . 14
2.3.8 Ultra-Wide Band . . . 14
2.3.9 Ultrasound . . . 14
2.3.10 Vision-Based . . . 15
2.3.11 Dead-Reckoning . . . 15
2.3.12 Comparison . . . 15
2.4 Attendance and Occupancy systems . . . 17
2.4.1 Bluetooth-based . . . 17
2.4.2 RFID-based . . . 17

3 Architecture **19**
3.1 System Requirements . . . 19
3.2 System Architecture . . . 21
3.2.1 AtOcu Client . . . 21
3.2.2 AtOcu Server . . . 24
3.3 Communication . . . 27
3.4 Concluding Observations . . . 28

4 Implementation **29**
4.1 Implementation Scenario . . . 29
4.2 Implementation Options . . . 30
4.2.1 Indoor Positioning System . . . 30
4.2.2 Software Tools . . . 32
4.3 Implemented Architecture . . . 34
4.3.1 SmartCampusAAU Integration . . . 34
4.3.2 AtOcu Client . . . 37
4.3.3 AtOcu Server . . . 44
4.3.4 Security . . . 54
4.4 Concluding Observations . . . 55

5 Evaluation **57**
5.1 Prototype Deployment . . . 57
5.2 SmartCampusAAU Integration . . . 58
5.2.1 Positioning Tests . . . 58
5.3 AtOcu server . . . 63
5.3.1 Load Tests . . . 63
5.4 AtOcu client . . . 64
5.4.1 Response Time Tests . . . 64

6 Conclusions **67**

6.1 Future Work 68

Bibliography **75**

List of Figures

2.1 Location stack 8

3.1 High level architecture of the proposed system. 22
3.2 AtOcu client's detailed architecture. 22
3.3 The location stack layers in the AtOcu client. 25
3.4 AtOcu server's detailed architecture. 25

4.1 Instituto Superior Técnico - Taguspark campus 30
4.2 Mapping between MVC components and AtOcu server's modules. 34
4.3 High level architecture of the implemented prototype. 36
4.4 SmartCampusAAU's partial data model. 36
4.5 Entities managed by the AtOcu client. 41
4.6 Login sequence from the user perspective. 43
4.7 AtOcu client GUI. 45
4.8 Entities managed by the AtOcu server. 48
4.9 Site map of the AtOcu Website. 49
4.10 AtOcu website overview page. 50
4.11 AtOcu website locations page. 50
4.12 AtOcu website occupancy history page. 51
4.13 AtOcu website courses page. 51
4.14 AtOcu website attendance records page. 52
4.15 Interaction between AtOcu components and FenixEdu during user's login. 55

5.1 Handheld systems utilized to run the AtOcu client. 58
5.2 Blueprint sections of the rooms used for positioning tests. 60
5.3 Results of the positioning tests. 61
5.4 Results of the positioning tests per tested position. 62
5.5 Results of the load tests for the AtOcu server. 65
5.6 Results of the response time tests for the AtOcu client. 66

List of Tables

2.1 Comparison of positioning methods. 10
2.2 Comparison of IPS metrics and proprieties. 16
2.3 Comparison of IPS techniques and methods. 16

4.1 Software solutions used to build the AtOcu prototype. 35
4.2 Inference of the user's main role. 45

List of Acronyms

A-GPS Assisted GPS

ACID Atomicity, Consistency, Isolation, Durability

AoA Angle of Arrival

AP Access Point

API Application Programming Interface

BLE Bluetooth Low Energy

BT Bluetooth

Cell-ID Cell Identification

CSS Cascading Style Sheets

DEI Departamento de Engenharia Informática

DGPS Differential GPS

DoA Direction of Arrival

DR Dead Reckoning

DSI Direção de Serviços de Informática

GPS Global Positioning System

GSM Global System for Mobile

GUI Graphical User Interface

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IDE Integrated Development Environment

IPS Indoor Positioning System

IR Infrared Radiation

IST Instituto Superior Técnico

JSON JavaScript Object Notation

LBS Location-Based Service

LoS Line-of-Sight

MVC Model-View-Controller

NFC Near Field Communication

ORM Object Relational Mapper

OS Operating System

PS Positioning System

RAM Random Access Memory

REST Representational State Transfer

RF Radio Frequency

RFID Radio Frequency Identification

RSS Received Signal Strength

RSSI Received Signal Strength Indicator

SDK Software Development Kit

TDoA Time Delay of Arrival

TLS Transport Layer Security

ToA Time of Arrival

UHF Ultra High Frequency

URI Uniform Resource Identifier

URL Uniform Resource Locator

UWB Ultra-Wide Band

VPS Virtual Private Server

Wi-Fi Wireless Fidelity

WLAN Wireless Local Area Network

XML Extensible Markup Language

Chapter 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. One example is Google Inc. that has, in 2011, expanded its service Google Maps to support indoor maps that enables its users to navigate within buildings such as airports, shopping malls and universities[2].

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[3]. 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 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]. Instituto Superior Técnico (IST), following this trend, has launched the Técnico Lisboa mobile application[4] - in 2014 for the Android platform and in 2015 for the iOS platform, which are the most popular mobile Operating Systems (OSs). 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)[5], 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. The main objective of this practice is to draw correlations

[1] http://www.ericsson.com/res/docs/2013/real-performance-indoors.pdf (2015-02-03)
[2] Google Indoors Maps, http://www.google.com/maps/about/partners/indoormaps/ (2015-02-03)
[3] http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536 (2015-02-03)
[4] Técnico Lisboa Mobile, https://mobile.tecnico.ulisboa.pt/
[5] Departamento de Engenharia Informática, https://fenix.tecnico.ulisboa.pt/departamentos/dei

between students' attendance of lectures and their respective grades. 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. The existing system has some drawbacks that compromise its main objective:

- The act of passing a paper sheet from hand to hand is a possible cause of distraction among the students during the lecture;
- The teacher might forget to collect the attendance sheet;
- The posterior analysis of the attendance information has to be performed manually;
- The attendance sheet might get lost;
- Students can sign for their absent colleagues or sign mistakenly in wrong places;
- Students might not find their name on the attendance sheet.

As previously suggested, this task requires a lot of human intervention, which consumes time that could be better spent in lecture activities.

On the other hand, 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. Knowing that a lot of students usually use these spaces to study, it is often difficult to find empty rooms or rooms with just a few students. 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. Therefore, their differentiation lies on the context of the occupant that can be having a scheduled class or just using a room for other academic purposes.

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.

Moreover, outside the academic context, attendance registration can also apply to businesses interested in keeping track of their employees and occupancy registration can also allow building management personnel to analyze not just the current occupation of common areas such as bathrooms, cafeterias or passage positions, but also its variation over time.

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 Taguspark campus of the IST: it interacts with the local academic management system, FenixEdu[6], 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. For instance, the AtOcu platform could be adapted for the industry so that businesses could keep track of their employees through the attendance registration functionality. It could also be adapted to interact with building management systems so that they could behave accordingly to the occupancy estimation of the locations under the scope of their operation, e.g. by turning on the lighting or ventilation equipment when a user is detected inside a room.

On the other hand, the AtOcu platform was designed to be modular to ease the extension of its functionalities, e.g. by adding new services on the top of the services it already offers, and to be independent on the technology used for generating indoor location estimations.

1.4 Outline

This document is structured as follows. Chapter 2 surveys previous work in the field of Location-Based Service (LBS), indoor positioning and attendance and occupancy systems. A detailed description of the proposed architecture is presented in Chapter 3. The process and the choices made during the implementation of the proposed architecture are described in Chapter 4. Chapter 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 Chapter 6.

[6] FenixEdu, http://fenixedu.org/

Chapter 2

Related Work

In the next sections the main related work is explored to substantiate the proposed solution. Section 2.1 starts by introducing services developed on top of location information. The following section describes the techniques and methods utilized to estimate positions indoor. Then, Section 2.3 presents a set of systems and their respective base technologies used to provide location information. Finally, Section 2.4 introduces a few examples of attendance and occupancy systems.

2.1 Location-Based Services

Services that take into account the location of an entity, or multiple entities, are known as Location-Based Services (LBSs) [2]. ”Entity” and ”location” are two concepts that are worth clarifying. A entity is any object[1] whose location information is of interest. 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 (such as living room if a Positioning System (PS) is used in the context of a house) 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.

A LBS involves at least two entities that can be either stationary or moving, and they may also switch between these states. One of these entities is the target of the service, i.e. its location information is used to offer the service. Any of the entities involved can be the recipient of the location information.

2.1.1 Categories

In terms of the recipient of the location information, LBSs can be classified either as location-tracking services or as location-aware services [4]. In location-tracking services, the user’s location information is provided to an external entity, while in location-aware services, this information is used by the service to provide contextual information to the user.

[1] An object can be seen as a human, although, in reality, the human is perceived through the use of an external device that make it possible to know his localization.

LBSs can also be classified in terms of the entity that evokes the service (reactive or proactive), in the distinction of the user and the target (self or cross-referencing), and in the number of participant targets (single or multi-target) [5]. More specifically, reactive LBSs are directly invoked by the user while proactive services are automatically initiated by predefined events. When the user and the target of a LBS coincide, we say this service is self-referencing. On the other hand, cross-referencing services provide the target's location information for another user. When the focus of a LBS is on tracking only one target's location, then it is classified in single-target. Conversely, multi-target LBSs are focused on the relationship between multiple targets.

2.1.2 Architecture

LBSs can be developed following a monolithic architecture for faster results, although this approach lacks flexibility [6]. An alternative is to adopt a layered architecture that could serve any kind of LBS or application. An example of this architecture is presented in Figure 2.1. This architecture comprises six layers [7]:

1. The Sensors layer is responsible for detecting a variety of physical phenomena and collecting raw data that can be used to obtain location information.

2. The Measurements layer transforms the raw data collected from sensors into measurement types (e.g. distances or angles).

3. The Fusion layer uses the measurement information to determine the target's location information and presents an interface so that the upper layers can access this information. In this layer a specific representation for the location information could be chosen (e.g. symbolic or physical location) as well as the corresponding coordinate system.

4. The Arrangements layer infers the spatial relationships between the detected targets (e.g. relationship of proximity).

5. The Contextual Fusion layer combines location information, arrangement information and contextual information (e.g. target's schedule or indoor temperature) in order to enable the upper layers to recognize states or sequences of events.

6. The Activities layer adds semantic information to the contextual information in order to make inferences about the state of targets (e.g. the target is missing a scheduled appointment at a specific location). This layer is intended to provide an interface to directly support user tasks.

Derivations of this architecture are found in the literature. For example, a new layer named Intentions has been added at the top of the Activities layer, being responsible for deciding which actions to take based on the information provided by the Activities layer and users' preferences [8].

Another layered approach, yet simpler and more generic, is defined by just three layers [9], specifically:

1. The Positioning layer involves the deployment and configuration of the location infrastructure. This layer is also responsible for collecting raw data using sensors and its subsequent transformation into location information.

2. The Middleware layer hides the complexity of the positioning layer and offers an interface for the upper layers.

3. The Application layer uses the data provided by the Middleware layer to offer a set of LBS.

There are also architectures that aim at supporting multiple positioning technologies that require complex data fusion modules [6]. These modules provide mechanisms for raw data fusing coming from different technologies and location information obtained from different techniques.

2.1.3 Security and Privacy

Location information can be used to infer other personal information, and therefore it is important to allow the users of location-based systems to have full control of their personal location information [10].

Regulatory strategies, privacy policies, anonymity and obfuscation (also known as entropy) are methods that can be used to ensure location privacy in LBSs [11]. 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.

These methods are not always effective in guaranteeing total privacy protection. In the case of anonymity, it is still possible to analyze patterns of user habits, which could be used to infer user identity [10].

In some services, the target's location is often disclosed to service providers. It is important to obtain the consent of users not only in self-referencing LBSs but also in any service that collects any sort of personal location information. To obtain a consent, service providers should inform users of all implications of such disclosure [8]. Specifically, users must be informed of: who has access to the disclosed location information; the steps taken to protect the location information; other uses that the service provider has for the location information; and the risks involved in disclosing the location information.

In order the promote the protection of user privacy, the Wireless Association, CTIA[2], has developed a set of best practices and guidelines, that should be considered while developing a new LBS [12].

2.2 Techniques for Indoor Positioning

Indoor Positioning Systems (IPSs) use techniques as triangulation, fingerprinting, proximity and vision analysis to provide location information [13, 14]. These techniques aim to mitigate the measurement

[2]Cellular Telephone Industries Association, http://www.ctia.org/

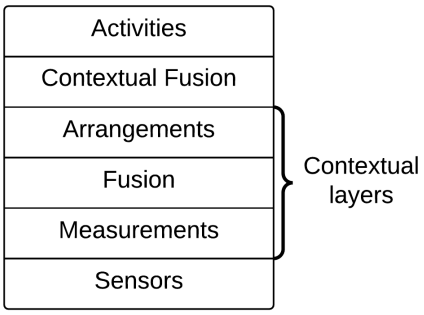


Figure 2.1: Location stack adapted [7].

errors caused by multipath fading, Line-of-Sight (LoS) path unavailability, among other wireless propagation interferences [15]. Lately, dead-reckoning techniques have also been explored, taking advantage of the set of sensors found in modern handheld systems (e.g. smartphones) such as inertial, motion and geomagnetic sensors. To get better performance, more than one of these techniques can be combined.

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 [13]. The target object can be a handheld system such as a smartphone. Lateration can still be refined into trilateration when the technique needs to use three reference points, and multilateration when it uses more than that amount of points.

In trilateration, the most used methods to estimate the distance among the target object and a reference point are Time of Arrival (ToA) and Time Delay of Arrival (TDoA). ToA takes into account that the distance between the target and a reference point is directly proportional to the propagation time and it requires, at least, three reference points. To use ToA, the internal clocks of all reference points involved must be synchronized. TDoA does not require this type of synchronization, instead, it determines the location of the target by looking at the difference in time that a signal, transmitted by the target, arrives at multiple reference points [13].

Angulation, on the other hand, can be applied using Angle of Arrival (AoA), also known as Direction of Arrival (DoA). This method needs, at least, two reference points, and the target's position is determined by the angle of the lines from each of the reference points to the target.

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 Frequencys (RFs) technologies. Firstly, it is necessary to build a radio map of the site. To do so, the offline stage is performed. 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. The process of pattern recognizing of the fingerprints can be made by using probabilistic methods or various machine learning algorithms, such as k-nearest neighbors, Neural Networks, or Support Vector Machines [16].

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 [3]. The location of the proximity sensors is previously known [3]. There are two scenarios where it is possible to determinate a relative location of the target. The first scenario happens when the target is detected by a single sensor, and therefore the target's location is the same as the sensor. The second scenario involves more than one sensor detecting the presence of the target. In this case, the location of the target is determined by the location of the sensor that detects a stronger signal from the target.

On the other hand, there are technologies that explore the proximity technique by inverting the roles of the target and the sensors as previously described. For instance, Near Field Communication (NFC) is one of those technologies where a sensor is attached to the moving target and the references points (previously referred as sensors) are stationary.

2.2.4 Vision-Analysis

Visioning positioning is a technique that estimates locations from images received by one or multiple reference points [17]. In this case, the target objects do not need to carry a tracked device. Instead, one or multiple cameras are placed in the indoor environment where indoor positioning is to be enabled and the target objects are identified in the captured images. Finally, the position estimation of the target object is obtained by looking in a pre-measured database.

2.2.5 Dead reckoning

The current generation of handheld systems, such as smartphones and tablets, employ a diversity of sensors that contribute to the creation of a wide range of applications [18]. Given the target's current

[3] Closeness means the sensor detects the object within a limited range.

position, its consequent positions can be estimated using sensors such as accelerometers, gyroscopes, and magnetometers (inertial, motion and geomagnetic sensors, correspondingly) [19].

This technique, which is commonly known as Dead Reckoning (DR), does not provide absolute locations, but rather provides relative locations to the starting location provided by other system. The direction of the target's movement is usually measured by a magnetometer functioning as a compass and the displacement can be provided by an accelerometer. Although these methods can be improved by using the remaining sensors and mathematical models, they will have accumulated errors over time due to the precision of the sensors as well as imprecisions caused by external interferences (e.g. magnetic interferences affect magnetometers). It is necessary to periodically correct the location information by using other positioning technologies as described later in Subsection 2.3.11.

Knowing that different movements (e.g. walking or running) can have an impact on the data collected by sensors, it is still difficult to define all the unknown variables that can determine the target's location (e.g. stride length).

2.2.6 Comparison

A comparison of the various methods for location estimation are presented in Table 2.1. Wireless positioning systems usually employ triangulation methods that take into account the characteristics of radio signals. The propagation of these signals are often susceptible to ranging errors that are a result of the nonexistence of LoS and synchronization issues between transmitters and receivers. In the case of fingerprinting methods, their accuracy is dependent on the quality of the training data collected at reference points. This training data is used to build radio maps that can become outdated if there are any posterior changes in the wireless infrastructure, in the building infrastructure, or even in the environment dynamics (e.g. people movements, relative humidity level [20]). Proximity methods, on the other hand, quantize the space into cells around each transmitter, which limits their granularity. The performance of vision-based methods are affected by changes in ambient light as well as occlusions that might occur. Finally, DR methods are prone to a variety of errors associated with each of the sensors used, but their cumulative property is what causes the most significant errors, i.e. knowing DR methods estimate relative locations, their errors accumulate over time.

All the methods described in this subsection are susceptible to different types of errors, which suggest that the development of positioning systems could benefit from the use of multiple measurements techniques to improve the location estimation.

Table 2.1: Comparison of the methods used to estimate target's location [8].

Method	**Measurement**	**Sources of Errors**
Angulation	AoA	No LoS
Lateration	ToA, TDoA	No LoS, synchronization
Fingerprinting	RSS	Infrastructural and environmental changes
Proximity	Signal strength	Space quantization
Vision Analysis	Video Frames	Occlusion
Dead Reckoning	Acceleration, direction	Cumulative

2.3 Indoor Positioning Systems

In this subsection we present the base technologies most used to develop commercial and academic, IPSs. An IPS is a system that provides location information that could be used to develop LBS. Their architecture is not strongly defined since they could be built only to serve a specific service, or they could be more modular and provide a number of services to be used by other services. In other words, they can implement any set of the layers described in Section 2.1.2.

It is important to notice that in order to surpass the limitations of a single technology, a system may combine multiple technologies, which extends the applicability of IPS to a wider range of indoors environments.

2.3.1 Indoor Positioning System Architecture

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 [17].

Thus, self-positioning architectures are more recommended for contexts in which it is necessary to preserve the privacy of the targets. Apart of the architecture side, the software side also plays an important role by managing, i.e. controlling access and distributing, the target's location information [17].

2.3.2 GPS and cellular-based systems

Global Positioning System (GPS) is the most used and successful positioning system, being widely used by military and civil applications [21] at a global scale. Nevertheless, GPS alone is not suitable for use in indoor positioning since there is a poor coverage of satellite signal for indoor environments and the radio waves used (microwaves) are attenuated by the structure of buildings and the objects inside them. There is also Differential GPS (DGPS), which is an enhancement of GPS that provides a better accuracy, but also suffers from the same indoor related problems of its base technology.

Assisted GPS (A-GPS) is an iteration over the GPS technology and it uses support stations to speed up the access to orbital information, overcoming GPS's weak signals in some environments. The communication between the A-GPS receivers and the support stations are usually done over cellular networks or Wireless Fidelity (Wi-Fi). SnapTrack was a commercial product exploring this technology [13].

There are many ways to determine the location of a target using cellular-based technologies and Cell Identification (Cell-ID) is one of them. This method uses the location of the base transceiver station that the target is using, at a given moment, to determine its location.

GPS and cellular-based systems are not widely used indoors, although they are used in most of the available outdoor positioning systems, supporting a wide range of LBSs.

2.3.3 Wi-Fi

The Wi-Fi, IEEE 802.11 standard [22], is a Wireless Local Area Network (WLAN) technology that has been explored in recent years to build positioning systems. Although Wi-Fi has a greater range outdoors, in the context of positioning it has been used mainly indoors. 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. The most common problems with this technology is related to the general interferences that effect RF technologies, mainly caused by the disposition of objects, the movement of the users on the site and the overlapping of APs [17].

One of the first examples of an IPS using WLAN was RADAR [23]. The proposed system uses the triangulation technique applied to RSS and signal-to-noise ratio, providing 2-D absolute location information with an accuracy of 4 meters [17]. SmartCampusAAU [24] offers an open software platform that supports the creation of LBSs by taking advantage of the existing WLAN. This platform requires the user, or the developer, to build the indoor radio map of the site where indoor positioning is to be enabled. Redpin [25] is another open source solution based on Wi-Fi that could take into account the RSSs of other radio technologies, namely Bluetooth (BT) and Global System for Mobile (GSM), to improve location estimations. Google Indoor Maps[4] is built on the top of Google Maps[5] and takes advantage of the fingerprinting technique over Wi-Fi measurements to offer indoor positioning. It was recently made available in Instituto Superior Técnico (IST) Alameda campus. Both SmartCampusAAU and Google Indoor Maps include support for representing floor plans.

2.3.4 Bluetooth

BT is a wireless technology standard, currently managed by the Bluetooth Special Interest Group[6], that is suitable for exchanging data over short distances. For this reason, it has been incorporated into a wide diversity of handheld systems. 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 [26]. On the other hand, these different BT protocols are not compatible among each other, even though they both use the same 2.4 giga-hertz RF.

The link layer of BLE technology has five states, namely standby, initiating, connection, scanning and advertising, where the last two are the most suitable states for implementing an IPS [27].

Contreras, Castro and Torre have also presented a performance evaluation of an IPS using BLE and introduce different working profiles that balance accuracy with energy consumption (i.e. there is a trade-off between location acquisition delay and respective energy consumption) to adapt to the needs of specific applications [27]. Zhao et al. compares the location accuracy between Wi-Fi and BLE-based

[4]Google Indoor Maps, https://www.google.com/maps/about/partners/indoormaps/
[5]Google Maps, https://www.google.pt/maps
[6]Bluetooth Special Interest Group, https://www.bluetooth.org/

systems with approximate indoor environmental conditions and concludes that BLE is around 27% more accurate than Wi-Fi [28].

Recently, new commercial products using BLE to provide indoor positioning have appeared on the market. Indoo.rs[7] and Estimote[8] are two examples. These kind of products include BLE compatible beacons and Software Development Kits (SDKs) that allow developers to create LBSs for handheld systems based on Android[9] and iOS[10].

2.3.5 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 [14]. 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.

IR is a short-range technology suitable for selective reception of signals [13]. Active Badge system [29] is an example of the use of this technology in the context of indoor positioning.

2.3.6 RFID

Radio Frequency Identification (RFID) is often used to track and identify objects (e.g. goods or people). This technology is often applied using proximity techniques, involving RFID tags and readers. RFID tags can be either active or passive [30]. 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. A RFID system usually consisting of two pieces of hardware: tags and readers. The RFID tags are essentially transponders responsible for storing a small amount of data (e.g. unique tag identifier). This data is transmitted to readers that are within the range of the tag, using radio signals. The reader decodes the signals received and usually pass them to a proper software layer that interprets them and estimate the respective location of the target.

The accuracy of the RFID systems depend on the techniques and methods used to estimate location information as well as the density of tag deployment [14]. These systems can provide just symbolic location based on the proximity of the tags detected, or they can achieve higher accuracy if applied using methods such as AoA and TDoA.

SpotON [31] and LANDMARC [32] are two examples of indoor positioning systems based on RFID technology that use active tags. More recently, an indoor location methodology using a Ultra High Frequency (UHF) passive RFID system was proposed for construction projects [33].

[7] Indoo.rs, http://indoo.rs/
[8] Estimote, http://estimote.com
[9] Android, http://www.android.com/
[10] iOS, https://www.apple.com/ios/

NFC[11] is a low-cost and bidirectional wireless communication technology, based on RFID technology. It allows devices to share information at a maximum rate of 424 kilobits per second at less than 4 centimeters. NFC can be used to perform contactless transactions[12], access diverse digital content and serve as a communication bridge between different electronic devices.

NFC Internal is an indoor navigation system based on NFC that orients users of NFC enabled mobile devices [34]. Users are able to know their location by touching with their mobile devices on NFC reference tags spread across a building at known positions.

2.3.7 Zigbee

ZigBee, currently at version 3.0[13], is a wireless communication standard based on IEEE 802.15.4 [35], operating at 2.4 giga-hertz. This technology is intended for applications with a low throughput and low power consumption profile [14].

Sugano et al. implemented an IPS based on ZigBee that uses the Received Signal Strength Indicator (RSSI) measurement [36]. Given that RSSI measurement is affected by multipath fading, Hu, Cheng and Zhang propose an algorithm that use a moving average calculation to smooth the signal propagation exponent and the averaged RSSI to mitigate that phenomenon and therefore it contributes to improve ZigBee-based IPS, in terms of position estimation, while maintaining the energy consumption and the low data rate [37].

2.3.8 Ultra-Wide Band

Ultra-Wide Band (UWB) is also a radio-based technology, whose signals have a bandwidth of at least 500 mega-hertz and a time resolution in the range of nanoseconds, which makes it less affected by radio related interferences, such as multipath fading and the nonexistance of LoS, and, for that reason, it can use methods such as ToA, TDoA and AoA to provide a better precision and accuracy than the previously presented radio technologies [14].

UWB-based technology transmits signals over multiple bands of frequencies at the same time, on the interval [3.1, 10.6] giga-hertz, and can be used in areas covered by other RF-based technologies without significant interference [13].

An example of an IPSs based on UWB is Ubisense [38] with an accuracy of 15 centimeters.

2.3.9 Ultrasound

IPS based on ultrasound make use of lateration techniques (e.g. ToA) to estimate the target's position. Although ultrasound-based system use mechanical waves rather than electromagnetic waves, their accuracy is also affected by the nonexistence of LoS and other interferences such as multipath propagation [14].

[11] Near Field Communication, http://nfc-forum.org/what-is-nfc/about-the-technology/
[12] NFC is compatible with the ISO/IEC 14443 standard for contacless integrated circuit cards.
[13] ZigBee version 3.0, http://zigbee.org/zigbee-for-developers/zigbee3-0/ (2014-12-02)

Active Bat [17] and Cricket [39] are well-known systems in the literature that use ultrasound as the base technology to provide indoor positioning.

2.3.10 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 [40]. 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 [41]. Instead, they use face recognition algorithms to identify the target and hence they are more unobtrusive.

2.3.11 Dead-Reckoning

Since DR just gives the target's relative position, an absolute position must be found by applying other positioning technologies. For instance, Sharp and Yu described a system that uses known positions as checkpoints to update the position given by DR methods [19]. To reduce the errors associated with the identification of these checkpoints, positioning techniques based in the WLAN infrastructure or in NFC reference points located in the coverage area can be used. The idea of checkpoints can also be applied to estimate an absolute location to initialize a system that uses DR methods.

2.3.12 Comparison

Tables 2.2 and 2.3 present a summary of the technologies used indoors for positioning purposes that were described in this subsection in terms of the position estimation techniques, measurements methods, and a set of the following evaluation criteria:

- Accuracy is often referred as an uncertainty measure that defines that area where the target could be placed;
- Scalability represents the number of targets that could be tracked by an IPS;
- Commercial Availability means that there are commercial IPSs using this technology;
- Self-positioning means that the location information is generated by the target itself;
- Cost of a IPS installation and maintenance;
- Target that could be detected by the IPS;

Wi-Fi, BT (especially BLE) and RFID using passive tags (where we also include NFC) are the most suitable technologies to build a low cost IPS. The first two have also the advantage of being compatible with the majority of the handheld systems that are commercially available, which opens the possibility of using these technologies in combination with DR techniques to improve the positioning accuracy.

Table 2.2: Comparison of IPS technologies in terms of metrics and proprieties [13, 14, 17].

Technology	Accuracy	Scalability	Commercial Availability	Self-positioning	Cost	Target
Wi-Fi	1-10 m	high	yes	yes	low	Smartphones, tablets
BT	1-5 m	high	yes	yes	low	Smartphones, tablets
RFID	4 cm-5 m	medium	yes	yes	low	Specific tags, smartphones (NFC)
UWB	1 cm-1 m	low	yes	no	high	Specific tags
Zigbee	1-10 m	low	yes	yes	medium	Specific tags
IR	1 cm-5 m	low	yes	yes	medium	Specific tags
Ultrasound	1 cm-1 m	low	no	no	high	Specific tags
Vision-based	1 cm-1 m	low	no	yes	high	Any visible object

Table 2.3: Comparison of IPS technologies in terms of techniques and methods [13, 14, 17].

Technology	Technique(s)					Method(s)				
	Trilateration	Angulation	Fingerprinting	Proximity	Vision Analysis	ToA	TDoA	AoA	RSS	Video frames
Wi-Fi	x	x	x	x		x	x	x	x	
BT	x		x	x		x			x	
RFID	x		x	x					x	
UWB	x	x				x		x	x	
Zigbee	x			x					x	
Infrared	x			x		x				
Ultrasound	x					x	x			
Vision-based		x			x					x

2.4 Attendance and Occupancy systems

Some of the technologies presented in Section 2.3 appear as possible candidates to build attendance and occupancy systems capable of working indoors. Attendance is the act of someone being present at a given event occurring at a given time and location. For this purpose, attendance systems can also be used to infer the occupancy rate of the event it is supposed to serve.

In the literature, most of attendance systems take advantage of the user's biometric characteristics [42, 43]. More recently, RF-based technologies have also been explored in this context.

2.4.1 Bluetooth-based

MITSAT is a student attendance tracking system based on BT technology [44]. 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 RSSI measurement. Saad et al. described an intelligent lecture assistant that provides a solution for the scenario of class rooms attendance [45]. 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. A similar approach that uses BT is described by Jamil [46].

2.4.2 RFID-based

RFID is one of the most used set of wireless technologies to build attendance systems [47–51]. Smart Attendance System [47] 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 by Kassim et al. [48] 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. The work developed by Al-Barhamtoshy, Altalhi and Mashat [49] suggests that the unique identifier of the RFID tag used by each student could be renewed every year to increase student's privacy. In the same work, the solution adopted is said to have reduced wasted time related with attendance checking. Patel proposes a system that uses the RFID technology in conjugation with face recognition [51]. This system firstly detects the user through the RFID subsystem, similarly to the previous described systems, and then verify the student identifier by capturing a photo of the student and consequently analyzing it.

Subpratatsavee et al. proposes a system that takes advantage of the sensors found in modern handheld systems [50]. It combines the NFC sensor and the camera of the lecturer's mobile device to develop an attendance system. When entering a class, each student carrying their unique NFC tag must

touch the lecturer's mobile device (functioning as a NFC reader) while the embedded camera is used to take a photograph of the student. Afterward, the data coming from the two sensors are combined and used to verify each student's identity.

Chapter 3

Architecture

In this Chapter, we present the architecture of the AtOcu platform. The system's requirements are presented in Section 3.1. The high level architecture and the various components that compose the AtOcu platform are described in Section 3.2. In Section 3.3, we explore the characteristics of the communication between the various components of the system. Finally, concluding observations are presented in Section 3.4.

3.1 System Requirements

Universities campuses are polyvalent environments, having multiple types of spaces to be used by teachers, students and staff. The present solution will just focus on places that could be used as classrooms or for study purposes inside university campuses.

Students and teachers will be the targets of the proposed system since their location information is going to be used for automating the attendance registration. These entities are also users of the system, inasmuch as they will be able to use the system by accessing attendance and occupancy information.

Given the places where information will be collected, the system must estimate the targets' location with room-level accuracy. The system must also estimate the occupancy of such places and make this information available to its users.

Higher education institutions usually have a large number of students and academic staff and although not all of the students are attending lectures at the same time, the system must be able to handle an increasing number of parallel interactions (e.g. attendance recording or occupancy queries) up to the maximum bound defined by the total number of potential users. In other words, the system must scale for the total number of students and teachers of the higher education institution where it is being deployed at.

Since collecting personal location information might raise privacy concerns among those being tracked, the system must enforce the anonymity of its users unless they are attending lectures where the recording of their attendance is mandatory. In terms of geographical limits of operation, the system must just determine the location of its targets within the limits of the university campus.

On the other hand, the system must work with minimum user intervention, i.e., the system must guarantee that the occupancy and attendance information is automatically gathered without explicit user intervention - but still with his consent.

The system must take advantage of the infrastructure commonly found in university campuses, such as the WLAN, and the handheld systems commonly used by its target users, namely smartphones and tablets, in order to lower the involved costs.

The system must allow its administrator or building management personnel of the target university to overview the occupancy and attendance data being collected, without compromising the students' anonymity.

The system must also be developed in a modular way so that its functionalities could be easily extended.

In summary, the requirements of the system can be defined in functional and non-functional requirements. In terms of **non-functional requirements**, the system must:

- Generate location information with room-level accuracy and within the physical limits of the university campus;
- Scale to handle the total number of students and teachers of the university in which it is to be deployed at;
- Rely on the infrastructure of the target university and on the personal handheld systems of the students and teachers;
- Enforce the anonymity of students and teachers when collecting occupancy information;
- Be easy to use;
- Be modular and extensible;
- Collect user's location information only with his consent.

On the other hand, in terms of **functional requirements**, the system must:

- Automate room occupancy estimations;
- Periodically register the room occupancy estimation;
- Provide room occupancy information;
- Automate student attendance registration;
- Provide the number of attendees during class;
- Inform students and teachers about their next scheduled classes;
- Provide a global and aggregated view over the occupancy and attendance data previously collected to the system's administrator or building management personnel.

3.2 System Architecture

In response to the system requirements and recalling the architectures discussed in Chapter 2, we propose the architecture shown in Figure 3.1. The system comprises three basic components: an IPS, the AtOcu client and the AtOcu server.

There is also an external component that interacts with our system: the Academic Management System of the home university where the system is to be deployed. The represented Academic Management System is the entity responsible for managing academic data. For the purpose of this project, we consider this entity will be able to provide specific information about courses, namely course identification information, respective schedules, enrolled students and assigned teachers. Essentially, the Academic Management System should be seen as a black box.

On the other hand, in this project our objective is not to propose a new IPS; rather we will make use of an existing system that provides the necessary flexibility to build LBSs on top of it. Since the chosen IPS must take advantage of the infrastructure commonly found in university campi, with emphasis on the campus in which the system proposed is to be deployed, we decided to make the choice of the IPS an implementation option that is described later in Chapter 4. For that reason, the IPS is also going to be seen as a black box at the architectural point of view.

3.2.1 AtOcu Client

The AtOcu client, detailed in Figure 3.2, 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. Our choice is supported by two reasons: first, the fact that the user's location information is produced in his own device facilitates the enforcement of privacy preserving measures; second, we are pushing the location estimation to the nodes responsible for collecting raw data and producing measurements, which better distributes the load of the system instead of producing these estimations on a centralized infrastructure.

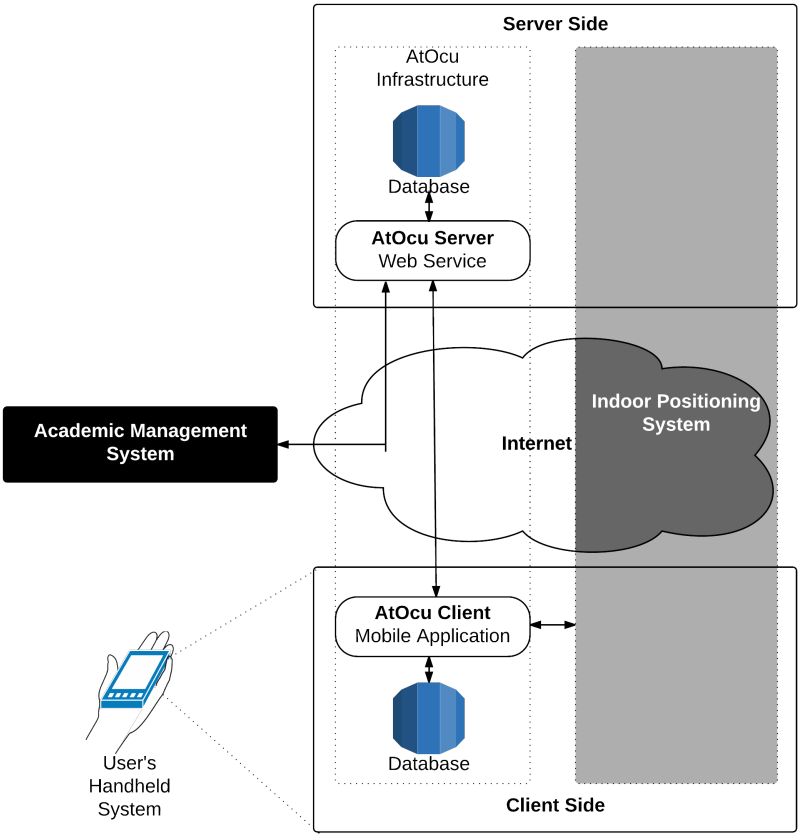


Figure 3.1: High level architecture of the proposed system.

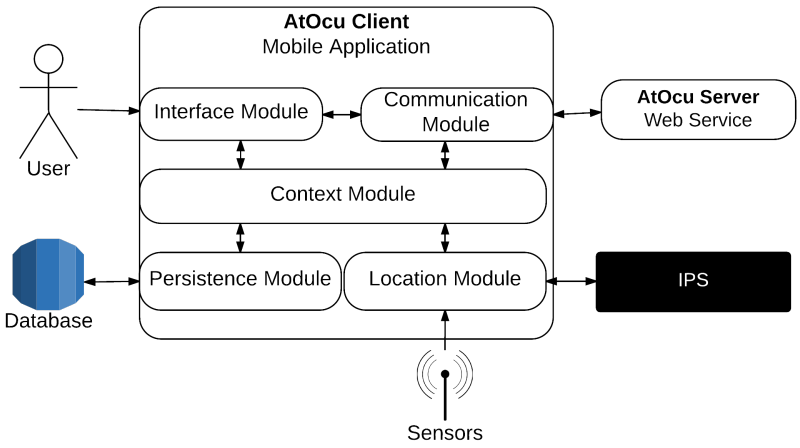


Figure 3.2: AtOcu client's detailed architecture.

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 following contextual information:

- The 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 requests 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 Section 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. Therefore, the Communication module sends the received schedule to the Context module for further analysis.

To perform location estimations, the AtOcu client interacts with an IPS that uses the sensors commonly available in handheld systems. As we have previously suggested in Chapter 2, technologies such as Wi-Fi, BT (where we also include BLE) and NFC are the most suitable ones, not only in terms of the costs involved but also because it is possible to achieve room-level accuracy with such technologies. Depending on the technology - within the range of the previously suggested options - and techniques used by the IPS, locations can be represented internally in different forms. For that reason, we have included a Location Module that is responsible for interpreting these IPS specific representations and eventually translating them into a representation used by the AtOcu platform. 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 choice of the approach to follow in order to detect that the user is on campus depends on the implementation scenario and thus it is discussed in Chapter 4.

The Persistence module manages the access to the local database where the entities required by other modules are stored.

The Context module contextualizes the location information previously gathered by the Location module with the user's schedule it has received from the Communication module 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. The periodicity of the location updates is parameterized so that it could be adjustable to reach a balance between the energy efficiency of the handheld systems and the granularity of updates.

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. This way, the Context module can retain the scheduled class that will happen sooner and send the remaining schedule to the Persistence module for storage purposes. After the Context module detects the retained class has started, it checks if its user's current location information matches the location of the class. In other words, the Context module determines 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, and retrieves the next scheduled class from the Persistence 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.

Another important point we have taken into consideration while designing the detailed architecture of the AtOcu client was the location stack introduced in Subsection 2.1.2 of the previous chapter. The mapping between the layers suggested by this architecture and some of the modules of the AtOcu client is represented in Figure 3.3 and could be interpreted as follows. The handheld system's sensors detect the physical phenomena (Sensors layer) and through the respective interfaces it is possible to access the measurements (Measurements layer). The Location module, in combination with the chosen IPS, is responsible for converting the measurement information into a specific symbolic location adopted in the AtOcu platform (Fusion layer) and infer spatial relationships between different locations (Arrangements layer). The Context module, on the other hand, has the necessary logic to combine the estimated user's location received from the Location module and the user's schedule (Contextual Fusion layer) to infer the user's status and consequentially decide what action it should execute (Activities layer).

3.2.2 AtOcu Server

The AtOcu server, detailed in Figure 3.4, 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.

Given the characteristics of the data being managed by the AtOcu server, the database used in the server side is a relational database so that the Atomicity, Consistency, Isolation, Durability (ACID) properties over the entities being managed by the AtOcu server can be guaranteed, with special emphasis

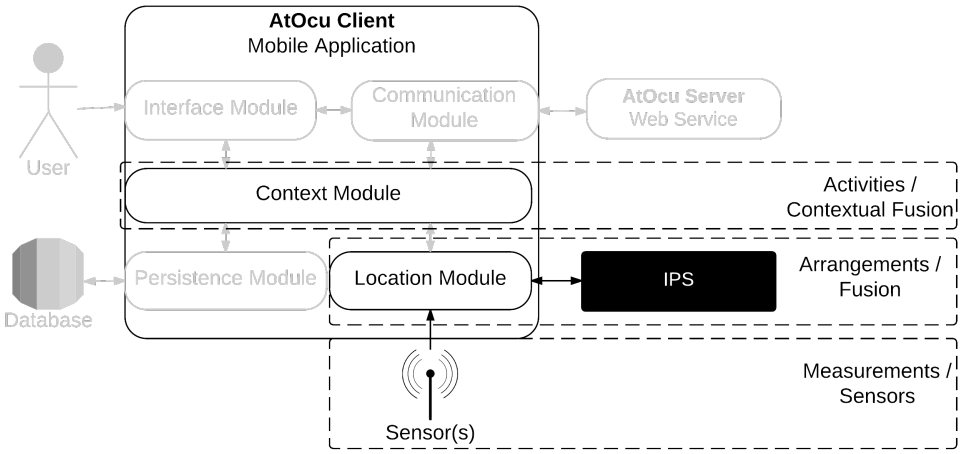


Figure 3.3: The AtOcu client and the corresponding mapping between its modules and the location stack layers.

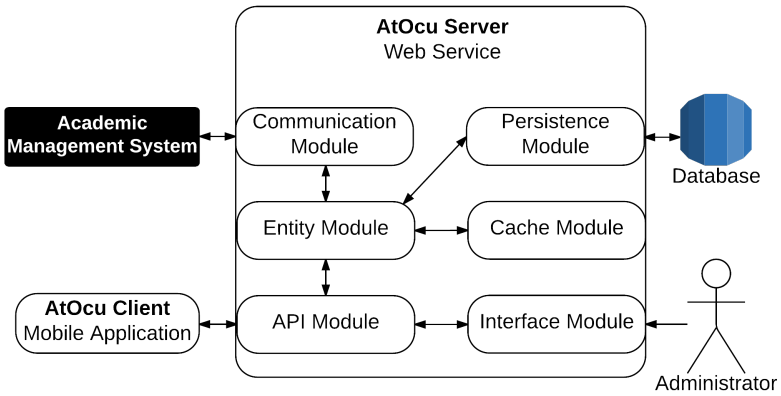


Figure 3.4: AtOcu server's detailed architecture.

on the attendance and occupancy data. The access to this 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. Both the API and the Communications modules constitute points of interaction with external components. The difference between the two modules lies in the component that performs the request. If the request is performed by an external component, then the API module is the module responsible for generating a proper response. If instead the AtOcu server is the component that performs the request, then it must be done using the Communication module.

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. When a user signs up for the first time, the Entity module creates a new entity that represents that user and requests his schedule from the Academic Management System through the Communication module. After analyzing the user's schedule, the Entity module creates new entities that represent the courses that the user is attending during the current academic term and its respective classes. Depending on the chosen IPS, it might be also necessary to create entities that represent the locations where the classes are being held. 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. This period of time applied to the location updates on the server side is also parameterized and, more importantly, it is consistent with the periodicity of the communications received from the AtOcu clients. By being consistent, we mean the expiration time is longer than the time period of received location updates to accommodate delays caused by the communication medium, but not excessively longer so that it negatively biases the freshness of the occupancy data. When a cache entry reaches its expiration time, the Cache module ensures it is destroyed.

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. For each mapped location, the aggregated information comprises the current occupancy of the location and the date and time when the snapshot was taken.

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. However, there is still some post-processing given to the attendance records. The AtOcu server has to receive attendance records from both students and teachers of a particular class to deduce that this class has effectively occurred. Therefore, the Entity module runs a process periodically to inspect the classes that have, or not, happened, wherefore that information can be added to the record for future reference.

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. The Interface module requests the information from the API module, just as the AtOcu client does. However, the Interface module may only request aggregated attendance and occupancy data in order to maintain users' anonymity.

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. A subset of the constraints included in REST [52] are suitable for our solution:

- Client-Server architecture - the client and the server, represented respectively by the AtOcu mobile application and the AtOcu web service, have different roles and concerns, which guarantees the separation of concerns principle;
- Stateless - the AtOcu server does not store or manage the session state of the AtOcu clients. Instead, the AtOcu client is the component responsible for maintaining its own session state and thus each request sent must contain all the information needed to be successfully processed by the AtOcu server;
- Cache - there is a variable number of AtOcu clients that might make the same requests to the AtOcu server. Therefore, the AtOcu server caches its responses so that they can be reused in later responses for the same requests;
- Uniform interface - all components must interact through a single uniform interface, which contributes to isolate the changes that can be made at the implementation level;
- Layered system - the architecture of the AtOcu platform is composed of hierarchical layers, as we have previously suggested in Figures 3.2 and 3.4.

As previously presented in Subsection 3.2.1, 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. 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. The TLS protocol is a protocol designed to provide communications security over the Internet and 1.2 is its latest version [53]. It is common to refer to HTTP over TLS as Hypertext Transfer Protocol Secure (HTTPS). 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.

For the purpose of the proposed solution, we assume that the handheld system where the AtOcu client is running at is not compromised and its owner, the user of the system, is not going to try to reverse engineer the mobile application in order to take advantage of the LBSs offered. We also assume the web service provided by the AtOcu server is running on a secure machine.

3.4 Concluding Observations

Considering the requirements listed in Section 3.1, the proposed architecture was designed to provide the main LBSs that constitute the objectives of this project: student attendance recording and room occupancy estimation.

Students and teachers are users and, at the same time, targets of the AtOcu platform.

We have opted for a self-positioning architecture and interpreting the user's context at the AtOcu client level, which better distributes the load of the system than centralizing these tasks at the AtOcu server side. On the other hand, we have adopted the REST architectural style that allows the AtOcu platform to handle an increasing number of parallel interactions between a variable number of AtOcu clients and the AtOcu server.

The estimation of the user's location and the respective interpretation of his context performed by the AtOcu client that runs on the his own handheld system also contributed to preserve the user's anonymity. Considering the same objective, we have adopted an encrypted connection between the components of the AtOcu platform as well as storing aggregated data at the AtOcu server side, unless a disclosure is required, as with attendance registration.

Another architectural feature is that the AtOcu platform automates the student attendance recording and room occupancy estimation and thus it does not require explicit user intervention to perform these tasks.

Finally, both components of the AtOcu platform were modularized in order to make it easier to extend the proposed solution, for instance by offering new LBSs, and more importantly, to make the platform IPS independent.

Chapter 4

Implementation

In this Chapter, we present the implementation of the AtOcu prototype. We start by describing the implementation scenario that characterizes the context of the development process in Section 4.1. In Section 4.2, we introduce the chosen IPS and the software tools we have used to support the development of the prototype. The implemented architecture is presented in Section 4.3, which includes all the aspects involving the development of all the parts of the AtOcu platform, the adaptations performed in the chosen IPS and security measures. Finally, concluding observations are presented in Section 4.4.

4.1 Implementation Scenario

To develop the prototype of the AtOcu system, we have considered the existing physical and information infrastructure of IST. More specifically, we have considered the Taguspark campus[1], located in Oeiras, Portugal. Its facilities[2] occupy a total of 19380 m^2 and comprise a single building with three floors (Figure 4.1) that are used for a diverse set of academic activities.

The main building of the Taguspark campus is almost totally covered by a WLAN based on the IEEE 802.11 standard [22] - comprises a total of 55 APs spread across its indoor area - in furtherance of providing Internet access to its users. This access is granted to users who possess specific personal credentials associated with their affiliation to IST or to any other user who has access to the Eduroam network [3], of which the IST is a part.

IST uses the FenixEdu platform [4] for academic and administrative management. The FenixEdu platform is a family of products that are modular and configurable so that they could be customized to the needs of each institution that uses them. The FenixEdu Academic is one of these products, and it is a Student Information System solution, that manages the institution's back office information, including information that is required for our solution, such as information about students, teachers, their respective schedules and courses. This information is accessible through a web API, the FenixEdu API.

[1] Taguspark campus, https://tecnico.ulisboa.pt/misc/vv-taguspark/tour.swf
[2] https://tecnico.ulisboa.pt/en/about-IST/facts-figures/ (2015-07-30)
[3] Eduroam, https://www.eduroam.org/
[4] FenixEdu, http://fenixedu.org/

The FenixEdu API contains public and private endpoints, any of each provides access to information about the resources managed by the FenixEdu platform. External applications can access public endpoints without any previous consent, while the access to private endpoints is dependent on a previous authorization granted by the resource's owner. To manage this authorization process, the FenixEdu platform uses the OAuth 2.0 [1].

Considering the high level architecture presented in 3.2, FenixEdu corresponds to the Academic Management System component that our solution interacts with in order to obtain information about courses, their schedules and enrolled students.

Both infrastructures, the WLAN and the FenixEdu instance deployed in IST, are managed by the Computer and Network Services, commonly known as Direção de Serviços de Informática (DSI) [5].

4.2 Implementation Options

4.2.1 Indoor Positioning System

To choose an IPS suitable for our implementation scenario, we took into account the different technologies previously presented in Section 2.3. Of all the IPSs presented, the ones based on Wi-Fi (Subsection 2.3.3) are the most suitable choices to serve our prototype since there was already a WLAN infrastructure in place. Therefore, we did not need to deploy and configure new hardware, which would increase the cost of our solution.

From all the solutions based on Wi-Fi we have surveyed, we decided to further analyze the SmartCampusAAU [24] and the Redpin [25] IPSs because their software is open source and available for download [6][7]. These options were preferred over commercial IPSs because they do not result in additional costs with external services. On the other hand, both solutions use the fingerprinting technique and offer room-level accuracy so they comply with the architecture previously discussed in Chapter 3.

Figure 4.1: Instituto Superior Técnico - Taguspark campus

[5] DSI, http://dsi.tecnico.ulisboa.pt/en/
[6] SmartCampusAAU, http://smartcampus.cs.aau.dk/
[7] Redpin, http://redpin.org/

The Redpin IPS adopts a collaborative approach, by allowing its users to create, modify and use a radio map that was previously created by other users. Redpin comprises two components: the Sniffer component is responsible for collecting RSS measurements from different wireless devices in range to create a fingerprinting; the Locator component, on the other hand, stores the previously collected fingerprints in a central repository and contains the logic to locate mobile devices. Apart from collecting just the RSS of all Wi-Fi APs in range, the Sniffer component also collects RSS measurements from the active GSM cell and the identification of all non-portable BT devices. The measurements are then sent to a central server, which is responsible for supporting the mobile clients in the process of estimating their location given the known fingerprintings. Although the Locater was initially implemented as a central server, the authors of Redpin suggested that running the Locater on the mobile device would be beneficial considering the users' privacy. The client side of the Redpin solution that runs on the mobile device was initially developed for the Symbian Operating System (OS), but other implementations, namely for the Android and iOS OSs, are also available.

Similarly to Redpin, SmartCampusAAU also uses crowdsourcing to build radio maps. This solution supports both self-positioning and infrastructure positioning, whose differences were previously presented in Subsection 2.3.1. In its infrastructure positioning mode, the APs that are part of the WLAN infrastructure need to be capable of performing signal strength measurements - a task performed by mobile devices in its self-positioning mode.

The SmartCampusAAU platform comprises a mobile application, the SmartCampusAAU app, and a server, the SmartCampusAAU backend. The SmartCampusAAU app is used to build radio maps, and depending on the positioning mode the Wi-Fi measurements are performed by the mobile device that runs the mobile application or by the routers that are part of the WLAN infrastructure. The SmartCampusAAU backend, on the other hand, comprises two services, the Radio Map service and the Wi-Fi Sniffer service. The Radio Map service is responsible for handling the radio map created by SmartCampusAAU app in a centralized fashion, while the Wi-Fi Sniffer service is the entity that performs position estimations in the infrastructure positioning mode.

SmartCampusAAU mobile application can also be used to enable indoor navigation through the previously collected locations. The SmartCampusAAU mobile application is available for the most popular mobile OSs, namely Android, iOS and Windows Phone [8]. The Android version supports both positioning modes, while the iOS and Windows Phone versions just support the infrastructure positioning.

After analyzing the characteristics of both solutions, we have inspected the source code of both solutions and tried to run both on our means. Not only the source code of the SmartCampusAAU platform appeared to be clearer than the Redpin solution, but we were also successful in our attempts to run it locally, which was not the case with the Redpin platform. Apart from the clarity, the SmartCampusAAU platform is also an extensible solution, since it also includes a library that could be imported into third party mobile applications, making the access to the indoor positioning functionality centralized and organized.

On the other hand, the proposed architecture discussed in Chapter 3 requires the used IPS to support

[8]Windows Phone, https://www.windowsphone.com/

self-positioning, and SmartCampusAAU platform offers this feature out of the box, even if limited to a particular OS, the Android OS, while the Redpin platform does not.

For these reasons, we have chosen to use the SmartCampusAAU platform.

4.2.2 Software Tools

We have selected a set of software platforms, libraries and tools to help us developing the prototype of the AtOcu system. Our choices were made not only by taking into account the technical characteristics of the tools, but also our personal experience and community trends.

The AtOcu client was developed for the Android mobile OS, which runs on a variety of handheld systems, with emphasis on tablets and smartphones. Android is the mobile OS with the largest worldwide market share according to International Data Corporation[9], a fact that also occurs in IST as a previous survey had suggested [1]. On the other hand, the SmartCampusAAU mobile application version that meets our requirements, i.e. adopts a self-positioning architecture, is just available for Android.

For storage purposes, we have used the SQLite[10] database in the AtOcu client. This database is self-contained and does not require a separate server process, which makes it suitable for handheld systems. Another advantage is that SQLite is offered by default on the Android platform and thus it is fully supported.

To ease the management of SQLite database records, we opted to use an Object Relational Mapper (ORM). From all the options available, we chose the ActiveAndroid[11] solution because of its simplicity of use and configuration.

To take full advantage of the Android platform through its own APIs, we have used the Java programming language[12]. Another advantage of using Java in Android development is related with the large number of well tested open source libraries and tools supported by communities of developers that we can use to accelerate the development of software solutions.

To structure and layout the user interface of the AtOcu client, we have used Extensible Markup Language (XML), which is already used in Android by default for the same purpose.

The HTTP responses of the FenixEdu API are mainly served in the JavaScript Object Notation (JSON) format. Considering that HTTP request-response scheme is also used in the AtOcu platform for the exchange of data between its components, we decided to use the same response format. For that reason, we used the Gson library[13] to convert Java objects to their JSON representation and vice versa in both the AtOcu client and the AtOcu server.

To communicate with the AtOcu server through its REST API, we used the Retrofit[14] solution that provides a configurable HTTP client and it allows us to turn an API accessed through HTTP into a Java interface. On the other hand, Retrofit also makes it possible to download data in the JSON format, and thus it can be combined with the Gson library.

[9] http://www.idc.com/prodserv/smartphone-os-market-share.jsp (2014-12-02)
[10] SQLite, https://www.sqlite.org/
[11] ActiveAndroid, http://www.activeandroid.com/
[12] Java, http://www.java.com/
[13] Gson, http://code.google.com/p/google-gson
[14] Retrofit, http://square.github.io/retrofit/

The AtOcu server is the entity of our system that has to interact directly with the FenixEdu API. The FenixEdu API can be accessed through official SDKs in Java, PHP[15] and Python[16] programming languages. These SDKs facilitate the application of the OAuth 2.0 protocol flow by the external application accessing the private endpoints of the FenixEdu API. Although we could also have developed our own module of software to communicate with the FenixEdu API, it would fall outside the scope of our work. For that reason and considering we are already using the Java programming language to develop the AtOcu client, we decided to use the same language on the server side and take advantage of the official Java SDK to invoke the FenixEdu API.

To develop the AtOcu server we decided to use a web application framework that followed the REST architectural style. Among the various open source options available, we decided to use the Play Framework[17] because it supports the Java programming language and also follows the Model-View-Controller (MVC) architectural pattern that has some similarities with the proposed architecture for the AtOcu server. The MVC pattern is characterized by the models, views and controllers concepts [54] that can be associated, respectively, with the Entity, Interface and API modules. This association is presented in Figure 4.2. In brief: models are representations of the information manipulated by the application and for that reason the domain logic is concentrated in these components; views render data from models and represent them usually through an user interface; controllers, on the other hand, are the components that respond to events generated by external entities (e.g. client applications) based on models and views.

To develop the Interface module of the AtOcu server, we used HyperText Markup Language (HTML), Cascading Style Sheets (CSS), Scala and Javascript. The first two were chosen because they are part of the standard technologies proposed by the World Wide Web Consortium[18]. Scala[19] is the main programming language used in the Play Framework and it has a dedicated template engine that makes it possible to generate any text-based format. For this reason, we used Scala to generate dynamically HTML. Javascript is a scripting language that was chosen based on its simplicity of integration with HTML. For that reason, it was used to generate charts embedded within HTML on the client-side.

Recalling that the AtOcu server is responsible for storing attendance and occupancy data received from all the AtOcu clients, we had to choose a database that was compliant with the ACID properties and that was supported by the chosen web application framework. Given these requirements, we were able to find several open source options, so we decided to use one that we were already familiar with from previous academic projects: the PostgreSQL relational database system [20]. To access the database on the server side of the AtOcu platform, we decided to use the Ebean ORM[21] that is already integrated in the Play Framework.

To enable HTTPS on the server side, we used the web server Nginx[22] because we had previous

[15]PHP, https://secure.php.net/
[16]Python, https://www.python.org/
[17]Play Framework, https://www.playframework.com/
[18]World Wide Web Consortium standards, http://www.w3.org/standards/ (11-10-2015)
[19]Scala, http://www.scala-lang.org/
[20]PostgreSQL, http://www.postgresql.org/
[21]Ebean ORM, http://ebean-orm.github.io/
[22]Nginx, http://nginx.org/ (11-10-2015)

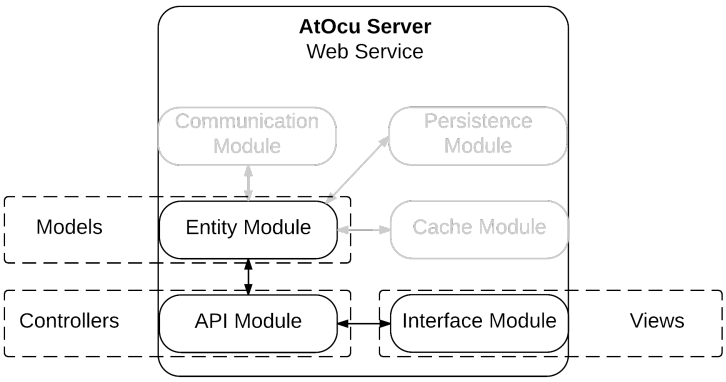


Figure 4.2: Mapping between MVC components and AtOcu server's modules.

experience in configuring it in similar contexts.

We used the Android Studio[23] as the Integrated Development Environment (IDE) for Android development. This platform has the advantage of having the Android SDK tools included. To work with the Play Framework, we have used the Intellij IDEA[24] community edition IDE. The Android Studio IDE is based on Intellij IDEA and for that reason, both IDEs share a common user interface and similar functionalities, which eased the development of the AtOcu prototype.

A summary of the technologies used is presented in Table 4.1.

4.3 Implemented Architecture

The architecture discussed in Chapter 3 has two components that are dependent on the implementation scenario. These components are the Academic Management System and the IPS. In Figure 4.3 we present the implementation architecture where these components are identified. 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. It is also reinforced the idea that our prototype takes advantage of the existing WLAN infrastructure.

4.3.1 SmartCampusAAU Integration

From the selected IPS, we have used the SmartCampusAAU mobile application, the SmartCampusAAU library for Android and the Radio Map service of the SmartCampusAAU backend. Recalling the fact that SmartCampusAAU platform uses the fingerprinting technique, we just use the SmartCampusAAU mobile application during the offline stage. This mobile application runs on the handheld system of the agent responsible for building the Wi-Fi radio map of the university campus where the system is to be deployed at. In order to map a new location, this agent needs to position its handheld system in the target location and collect new RSS measurements from that position. Then, these measurements are

[23] Android Studio, https://developer.android.com/sdk/
[24] Intellij IDEA, https://www.jetbrains.com/idea/

Table 4.1: Software solutions used to build the AtOcu prototype.

Name	Version	Type	AtOcu component
Android	4.4	mobile OS	client
Android SDK tools	21	SDK	client
Android Studio	1.1.0	IDE	client
SQLite	3.7.11	database management system	client
Retrofit	1.9.0	HTTP client	client
ActiveAndroid	3.1.0	ORM	client
Scala	2.11.1	programming language	server
Ebean	3.3.3	ORM	server
Intellij IDEA	14.0.4	IDE	server
PostgreSQL	9.3.9	database management system	server
FenixEdu Java SDK	2.3.1	SDK	server
Nginx	1.8.0	web server	server
Java	7	programming language	client and server
Gson	2.3.1	library	client and server

sent to the to the Radio Map Service that gathers them to build a new radio map or to update an existing one. These measurements and the corresponding meta information, such as the location information, are stored in a relational database in the Server side - a subset of the Radio Map service's data model that is important for the AtOcu platform is represented in Figure 4.4. After mapping the desired locations, the Radio Map service is able to provide the radio map previously built upon requests received from the AtOcu clients that use the SmartCampusAAU library for that purpose. Once the AtOcu clients receives the radio map, they are able to estimate the location of their user.

To ensure that the AtOcu platform just generates location estimations within the physical limits of IST - Taguspark campus, the radio map must just include locations from the indoor environment of the respective campus.

From all the components from the SmartCampusAAU platform that were used in the AtOcu prototype, we have just performed a minor adaptation of the SmartCampusAAU mobile application. After adding a new location to the radio map, we have the possibility of supplying symbolic information that characterizes that location. In the original SmartCampusAAU mobile application, the symbolic information includes a title, description, an uniform resource locator and the type of the location. Considering the proposed architecture, we needed the location's identification and capacity, so we have converted the title field to the identification field and the description field to the capacity field. Then, we needed the SmartCampusAAU mobile application to communicate with the AtOcu server in order to send both the identification and capacity of the newly added location, as well as its respective coordinates (latitude, longitude and altitude). The altitude coordinate represents the floor of the mapped location. The reason we also send the locations' coordinates to the AtOcu platform is that the radio map keeps a mapping between them and the recorded signal strength information, and therefore given the signal strength we can access the location's identification and capacity.

Although we have access to the entire SmartCampusAAU platform's source code, we decided not to run the Radio Map service in our own infrastructure and use the SmartCampusAAU server that was

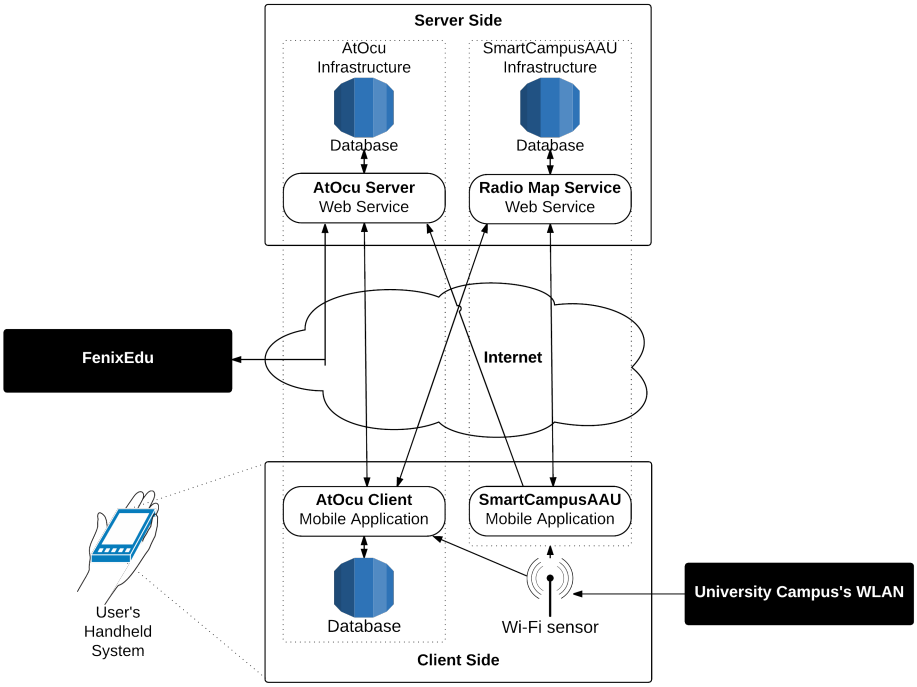


Figure 4.3: High level architecture of the implemented prototype.

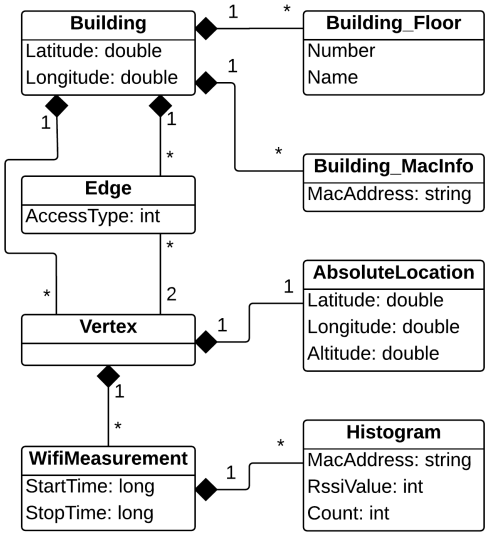


Figure 4.4: SmartCampusAAU's partial data model [24].

running at Aalborg University [25] during the development of our prototype for reasons of simplification. This decision does not impact the objectives nor the contributions of our work since the AtOcu platform was designed to be IPS independent. While developing this prototype we have assumed that the aforementioned server and the Radio Map service it offers were safe and did not compromise the data generated or used by the AtOcu prototype.

4.3.2 AtOcu Client

Android applications are composed of four types of essential components, namely activities, services, content providers and broadcast receivers. From these components, we have used activities, services and broadcast receivers to develop the AtOcu client.

From the user interface point of view, an activity can be used to represent a single screen of an application. In order to make the interface as modular as possible, we can also use another application component called fragment to represent portions of the user interface within an activity. For that reason, it is possible to combine multiple fragments in a single activity, which modularizes the development of user interfaces in Android. In order to create an activity, we have to create a subclass of `Activity`.

Services, on the other hand, are used to perform long-running operations in the background without providing an user interface. These components can be started from other components such as fragments or activities. To create a service, we need to create a subclass of `Service`. Services that are started from an application component and, once started, run in the background indefinitely are called started services - indefinitely in this context should be interpreted as "until the system decides to finish the service". On the other hand, bound services are services that run as long as an application component is bound to it.

Location Module

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

The indoor positioning functionality of the SmartCampusAAU library is exposed through the LocationService class, which is a service itself. Through this service, we are able to download the radio map and start and stop indoor positioning if we invoke, respectively, the methods `enableIndoorPositioning()`, `startWifiPositioning()` and `stopWifiPositioning()` of the `LocationService` service. This service was implemented as a bound service and thus we needed to bind another application component to it before using its interface. 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 (e.g. the handheld system enters the sleep mode), we created the started service Location Capsule that binds to `LocationService` and runs in the background. This service was implemented in the `LocationServiceCapsule` class. Our objective was to encapsulate the `LocationService` within this started service so that there is only one application component in the AtOcu client that interacts directly with the SmartCampusAAU platform.

[25] Aalborg University, http://www.en.aau.dk/

We implemented two variations of how the Location Capsule service gathered location estimations on the background. Our first approach was to make it to run indefinitely while the user was on campus. For that purpose, the service requested a wake lock from the `PowerManager`, which is a class from the Android platform that manages the control of the power state of the device, in order to force the device's CPU to run even if the handheld system's display goes off, preventing it from entering the sleep mode.

Our second approach consisted in using the `AlarmManager` class - available on the Android platform. This class 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. This approach does not require the AtOcu client to hold uninterruptedly the wake lock during the permanence of the user on campus and thus it requires less CPU time to provide the same service and consequently it is more energy efficient. For these reasons, we opted for the second approach.

Independently of the approach we choose to periodically generate location estimations, the Location Capsule service registers a listener with the `LocationService` to receive location notifications. Locations notifications are received through one of the listener's callback methods, the `onLocationChanged()`. For that reason, a newly received location notification is used by the Location Capsule service to update the internal state of the Smart Location manager, implemented in the `SmartLocationManager` class.

The user is detected on campus when his handheld system connects to the local WLAN. This was done through an implemented class that extended the `BroadcastReceiver` class that is another Android component which allow us to register our application for system events. In this case, we registered the AtOcu client for the `WIFI_STATE_CHANGED`, `STATE_CHANGED`, `CONNECTIVITY_CHANGE` system events, which are sequentially fired when the handheld system connects or disconnects from a wireless network.

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.

Context Module

The implemented Context module comprises the Attendance started service, 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. The user's status is persisted through the `SharedPreferences`, which is a class that allows us to save and retrieve persistent key-value pairs in the Android platform. By default, the data can just be accessed by other components of the AtOcu client.

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. There are two occasions when the AtOcu client schedules alarms through `AlarmManager`: once

the user's schedule is available or updated and after a scheduled class is finished. 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. By default, all alarms previously scheduled through `AlarmManager` are canceled when the handheld system shuts down. To prevent this from happening, we implemented the `ReschedulerReceiver` class. This class also extends the `BroadcastReceiver` class. In this particular case, we registered the AtOcu client for the `BOOT_COMPLETED` system event, which is fired after the Android system has completed the boot process and thus our application can reschedule the alarm that was previously canceled.

When the Attendance service starts, it firstly checks if the Location module is generating location estimations. If not, the Location module is requested for a single location estimation. Then, the Attendance service 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. Since user's schedule falls outside the scope of the `ContextualManager` class, the Attendance service also schedules an alarm that will signal the end of the schedule class, so that the future status of the user is updated accordingly.

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. The time of the rescheduling is dynamically computed based on the start time of the class and its duration so that it is performed up to four more times after the first failed attempt and it is never performed after half of the class duration. For instance, for a class with a duration of 90 minutes that started at 10:00 and whose first attempt to determine if the user was at the desired location failed, the hypothetical attempts would be performed at 10:11, 10:22, 10:33 and 10:45 (seconds are not considered for simplicity purposes). If all the attempts fail, the Attendance service proceeds to schedule a new alarm based on the next check-in event.

The attendance checking process proceeds until there are no scheduled classes left.

Communication Module

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

The `AtocuApiManager` class is used by the other modules of the AtOcu client to send requests to the AtOcu server through its REST API. These modules just need to instantiate the `AtocuApiManager`, whose constructor automatically configures a custom HTTP client, and invokes its asynchronous methods. Although the Retrofit solution offers a HTTP client out of the box, we needed to configure a custom HTTP client so that all request are sent over TLS to AtOcu server. For that purpose, we generated a keystore - by using the key and certificate management utility called keytool[26] - that contains the certificate of the server and included it into the AtOcu mobile application as an raw resource. Therefore, the

[26]keytool, https://docs.oracle.com/javase/7/docs/technotes/tools/windows/keytool.html

custom client can trust the certificate from the previous referred keystore and verify if the hostname of the server being contacted matches the hostname listened in the certificate.

The `AtOcuApiManager` is also responsible for configuring an instance of the `WebView` to communicate over HTTPS with AtOcu server, for login purposes, as it is later described in more detail in the Interface module.

Using Retrofit, we defined an interface to represent the REST API provided by the AtOcu server. In this interface, we used the `@GET` and `@POST` notations and its parameters, respectively GET and POST HTTP methods, to describe each HTTP request and respective parameters that are part of the API. In order to convert the JSON content form of the AtOcu server's responses to Java objects, we used the Gson converter that is integrated into Retrofit. We created a Java class to represent each one of the JSON responses. After receiving a response, the Gson converter automatically instantiates the corresponding object so that AtOcu client can use its content to provide its services. More information about the REST API is provided later on Subsection 4.3.3.

The majority of the requests performed by the AtOcu client are meant to communicate regularly which location the user is at - and thus contributing for the room occupancy service. The remaining requests are performed when the user logs in for the first time or when there are check-in events that the user has already attended and need to be sent to the AtOcu server for storage purposes.

The first type of requests are made directly by the Context module, while the two other types are made by an intermediate, the `AtocuSynchronizerService` service. This 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.

The user is require to intervene so that the AtOcu client can download his complete schedule information. A course can have multiple type of classes - in Departamento de Engenharia Informática (DEI), the most common types of classes are theoretical, laboratory or practical classes - and depending on the type of the class, it could happen that the student has more than one shift that he can choose to attend. On the other hand, courses can be lectured by more than one teacher, and when that is the case, the teachers have to decide who is responsible for a subset of the shifts available.

Since we were unable to get the association between teachers or students and the set of shifts they are, respectively, teaching and attending, we invite the user of the AtOcu client to inform the platform which shifts to consider while serving the attendance service. The shift selection is performed through the Interface module and it is started after the `AtocuSynchronizerService` service downloads the courses the user is associated with and their respective shifts. The shifts presented to the user are just the ones that have more than one option for each type of class. After the shifts are selected, the `AtocuSynchronizerService` service requests the AtOcu server to register this information and downloads the remaining schedule information, specifically the classes associated with the subset of shifts selected.

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. In case the AtOcu server is unreachable, the check-in events remain stored through the Persistence module so that they

can be sent later when the `AtocuSynchronizerService` service is again started.

Persistence Module

The AtOcu client manages check-in events, courses and symbolic locations. The first two entities are only related with attendance registration LBS while symbolic locations are entities that allow us to interpret the location estimation made by SmartCampusAAU library and thus they are used by both LBSs offered by the AtOcu platform. In relation to the local database, these entities are stored as tables and are represented in Figure 4.5.

The Persistence module is the combination of the ActiveAndroid ORM with the classes `Course`, `SymbolicLocation` and `CheckInEvent`. Therefore, the ORM was used to set up the local database and create the necessary tables from abovementioned class files that represent each one of the three entities. These classes extended the `Model` class from ActiveAndroid that has part of the logic that allows the mapping between our classes and SQLite tables and each class attribute needed to be annotated with `@Column` to be recognized as table columns. Then, to create a new data entry, we just need to create a new instance of any of the classes that extend the `Model` and call the `save` method to store it on the database. Once the tables are created, other modules are able to query the existing tables.

The check-in events represent locally the scheduled classes that are received from the AtOcu server. These events share the same identification of the scheduled classes they represent, and contain other contextual information that is used by the AtOcu client to know exactly when the event is happening (date), how long it lasts (duration) and where it happens, which is provided by their association with symbolic locations. The date represents simultaneously the year, month and day of the scheduled class, as well as the hour and minute, and it is represented in the number of milliseconds since the standard epoch date used in Java[27].

When the AtOcu client detects that its user is attending a scheduled class, the corresponding check-in event is marked as such. This check-in event persists in the local database until the AtOcu client successfully sends this information to the AtOcu server.

The courses entities represent the courses the user is enrolled in or teaching, depending on his role. These entities are mainly used to inform the user about the classes he needs to attend.

The symbolic locations are entities that represent the mapping between the absolute locations, represented by geographic coordinates, and a symbolic name.

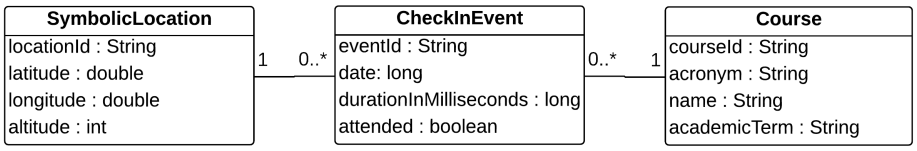


Figure 4.5: Entities managed by the AtOcu client.

[27]Standard Java epoch is January 1, 1970, 00:00:00 GMT as described in the documentation of the `Class Date`, https://docs.oracle.com/javase/7/docs/api/java/util/Date.html (2015-09-28)

Interface Module

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

Knowing that the students and teachers of IST are the targets and users of the AtOcu prototype, we took advantage of the authentication and authorization mechanisms being used by the FenixEdu platform with the AtOcu prototype. The authentication is provided by IST's Central Authentication Service, while the authorization is done through the OAuth 2.0 protocol [1]. The first mechanism allow us to verify the user accessing the AtOcu client is who he claims to be, while the second one enables the AtOcu platform to access the FenixEdu's API. For that purpose, we registered the AtOcu client within the IST system running the local FenixEdu instance, where we requested the Information and Curriculum scopes, which allow us to, respectively, access the user's personal information, name and username, and the user's curricular information, namely the courses the user is enrolled in or teaching, depending on his current role.

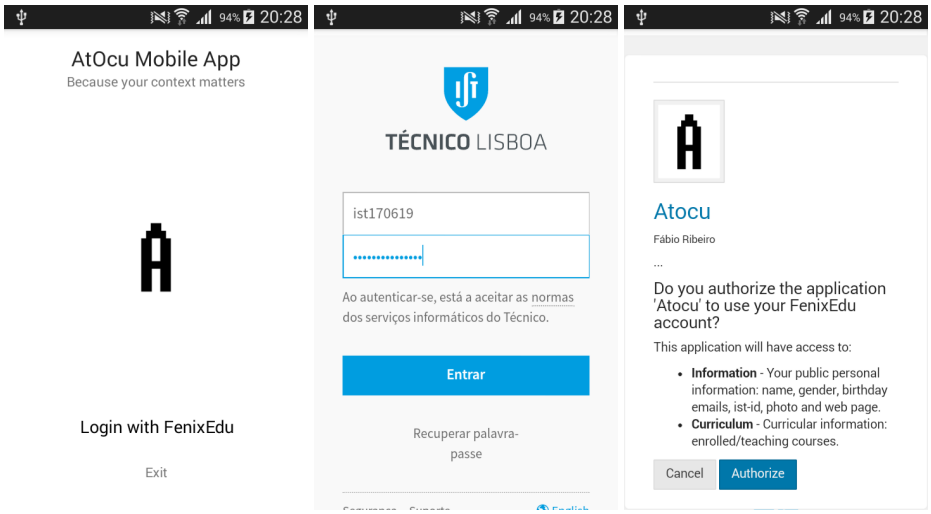
The login activity, implemented in the class `LoginActivity`, is responsible for verifying if the user is logged in into the AtOcu client; if not, it then leads the user through the login process.

The login process is lead by the login activity through its GUI represented in Figure 4.6. When the user initiates the AtOcu client for the first time, or after being logged out, he is given the possibility of logging in through FenixEdu (Figure 4.6(a)). In case the user decides to proceed, the login activity requests an instance of the `WebView` class from the Communication module, specifically from the `AtocuApiManager` class. This instance functions as an embedded browser and it is used to request the authentication page from the AtOcu server. The AtOcu server's response involves redirecting the user to the FenixEdu authentication web page, where he needs to enter his IST credentials (Figure 4.6(b)), namely his username and respective password, and then authorize the requested scopes (Figure 4.6(c)). The authorization is just performed once during the first login.

After the authorization process is finished, the embedded browser is redirected to an Uniform Resource Identifier (URI) with a known pattern. This pattern contains two tokens: the identity token and the anonymity token. 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. Therefore, the identity token is personal to each user. On the other hand, 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. Contrary to the identity token, the anonymity token is shared among the users of the AtOcu platform and thus it does not compromise their anonymous contribution to the room occupancy estimation.

The embedded browser was previously configured by the `AtocuApiManager` class to detect the redirection and extract the tokens from the known pattern and persist them locally so that the AtOcu client is the only application able to access them.

The main activity, implemented through the `MainActivity` class, manages a set of fragments to structure the GUI, as presented in Figure 4.7. Each fragment implements a screen and there are currently three screen available: the status screen (4.7(a)), the rooms screen (4.7(b)) and the classes screen (4.7(c)).



(a) Initial login screen. (b) Authentication form. (c) Authorization form.

Figure 4.6: Login sequence from the user perspective.

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. The personal information includes its name, username and role, which are static fields, i.e. they are not supposed to change over time. On the other hand, the contextual information section contains the education institution acronym and a set of dynamic fields:

- fields that depend on the user's current position, specifically the geographic coordinates, floor and name of the location in which the user is estimated to be;
- the term field that varies when the user enrolls in a new academic term;
- the status field that varies depending on the user's current location and schedule.

Whenever the user's status changes to show that user is currently attending or missing a class, the status field expands to show information about the course and the schedule of that particular class that is currently happening.

The classes screen lists the next classes the user has to attend on his schedule. 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. In case the user is a teacher, he can also consult the current number of students that are attending the class.

The rooms screen lists all the available rooms - rooms that are not currently occupied with classes - 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.

The navigation between these screens is done through a navigation menu (4.7(d)). There is also an options menu that can be accessed from any screen (4.7(e)) that allows the user to stop or start the gathering of his position, update his schedule and log out of the AtOcu client.

Additionally, we implemented a dialog that appears over any one of the previously referred screens that notifies the user to select the shifts he will be attending (4.7(f)).

4.3.3 AtOcu Server

The development of the AtOcu server took advantage of the architectural organization, specifically through the 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. The Play Framework also includes other components, such as the Router and the Cache API, that were used to support the development of the several modules that comprise the AtOcu server's architecture as it is explained throughout this subsection.

Communication Module

The Communication module's logic was encapsulated in `FenixEduApiManager` class. This class configures the API client provided by the official FenixEdu 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. The invoked API endpoints through the API client are the following:

GET /about returns basic information about IST and it is a public endpoint.

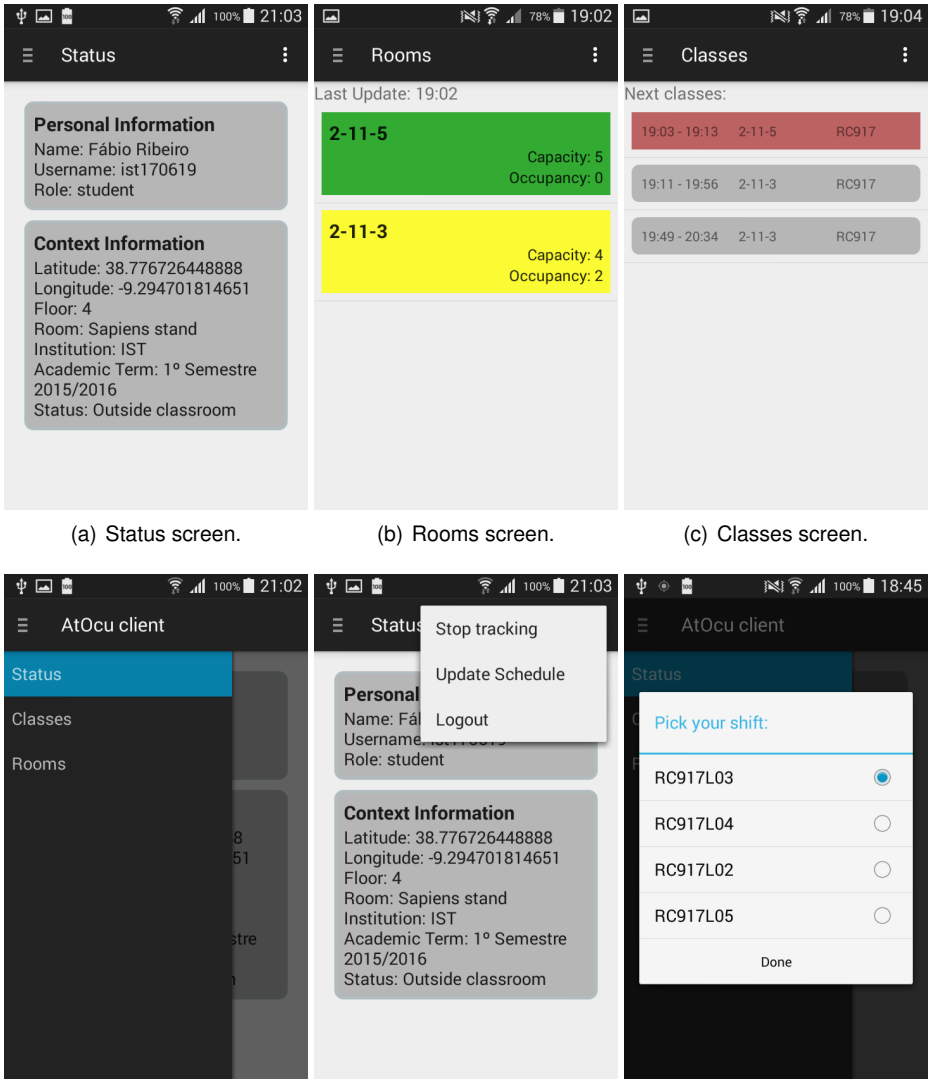
GET /person returns user's personal information and it is a private endpoint of the personal scope.

GET /person/courses returns user's courses information and it is a private endpoint of the curricular scope.

GET /courses/{id}/schedule returns information regarding the schedule of a particular course and it is a public endpoint.

Private API endpoints are invoked for each user, while public API endpoints are just invoked once per resource, since the AtOcu server uses the Entity module to later create the entities that represent the data retrieved and consequentially are stored on the local database through the Persistence module. The methods of the API client that correspond to these endpoints return JSON objects, and for that reason we used the Gson converter to automatically instantiate the corresponding Java object that allows us to directly access their fields.

The user's role is automatically inferred from the data retrieved from the FenixEdu API. The endpoint '/person' could return a combination between three types of roles: 'teacher', 'student' and 'alumni'. Since the targeted users of AtOcu platform are students and teachers and based on our experimentations with the FenixEdu API, for each user we infer his main role as represented in Table 4.2.



(a) Status screen. (b) Rooms screen. (c) Classes screen.

(d) Navigation menu. (e) Options menu. (f) Shift selection dialog.

Figure 4.7: AtOcu client GUI.

Retrieved roles	Inferred role
student	student
student, alumni	student
teacher	teacher
alumni, teacher	teacher
student, alumni, teacher	teacher

Table 4.2: Inference of the user's main role.

The `FenixEduApiManager` class is only used in two distinct situations: when users log in for the first time so that the AtOcu server can download all the required private data to offer attendance registration and occupancy estimation LBSs; and right after the AtOcu server is started to request the current academic term.

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.

Entity Module

Each one of the entities being managed by the AtOcu server were implemented as models in the Play Framework and they are presented in Figure 4.8, as well as the relationships between them. 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, that among personal information also holds OAuth access and refresh tokens that are used to access the FenixEdu API and an entity token that is sent to the AtOcu client so that this component can access the AtOcu API.

Classes, shifts and courses are entities that combined represent the user's schedule and are used to support the attendance registration LBS. In IST, a course comprises a variable number of shifts and a student is not necessarily enrolled in all shifts and thus the need arises to also manage shift entities.

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. Having these two types of locations were necessary to reduce the sparsity of the mapped locations that could cause the system to generate wrong location estimations when the user is not inside a room.

The SmartCampusAAU platform makes use of the fingerprinting technique which involves identifying radio patterns between the fingerprints being collected at the online phase and the radio map previously built during the offline phase. After mapping a single room in the whole campus, as long as the user's handheld system is able to detect one of the APs that contributed to create the radio fingerprinting of that location through its collected RSS measurements, the user will be estimated to be at that particular room. If the handheld system was positioned between two or more mapped rooms, the user would be estimated to be positioned in the room whose radio fingerprinting was more similar to the one being collected, even if he was not inside that particular room. With passage locations, we also need to map the locations through which the user passes by when moving between rooms, and thus the AtOcu client, where the location estimations are generated, can identify when the user is between rooms or inside rooms with a better accuracy since the mapped locations set becomes less sparse.

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. These entities are created when the AtOcu server receives anonymous location updates from AtOcu clients whose identity was not previously registered. The identity in question is randomly generated by each AtOcu client, by applying a hash function to a random sequence of characters and a salt based on the current time. After being created or receiving a new location update, the occupant entity sends its identifier to the Cache module with an expiration time of 70 seconds, taking into account the time period adopted by the AtOcu client between location updates.

The occupancy information stored on locations entities just reflects the current location occupancy situation and thus the Entity module creates occupancy snapshot entities that can be later consulted by the AtOcu platform's administrator or building management personnel. To automate the creation of occupancy snapshots, we used the Akka toolkit[28], available on the Play Framework, that can be used to schedule future tasks. 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. After each scheduled job is finished, the occupancy parameters of locations entities are reset, so that the next snapshot only takes into account the maximum and minimum occupancy values obtained in the last hour.

Persistence Module

On the server side architecture of the AtOcu platform, we used the Ebean ORM as the Persistence module. Ebean was integrated into the Play Framework and similarly to what we did at the AtOcu client's Persistence module, we annotated entity classes with `@Entity` and `@Table` and their attributes with `@Column` so that they are recognized, respectively, as tables and table columns. These tables are then persisted within an instance of the PostgreSQL database and their rows are manipulated through an API provided by Ebean.

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 HTML, CSS, Javascript and Scala programming languages for the following tasks:

- HTML and CSS were used to, respectively, structure and style the Web pages;
- Javascript was used to generate charts in the Web pages;

[28] Akka toolkit, http://akka.io/

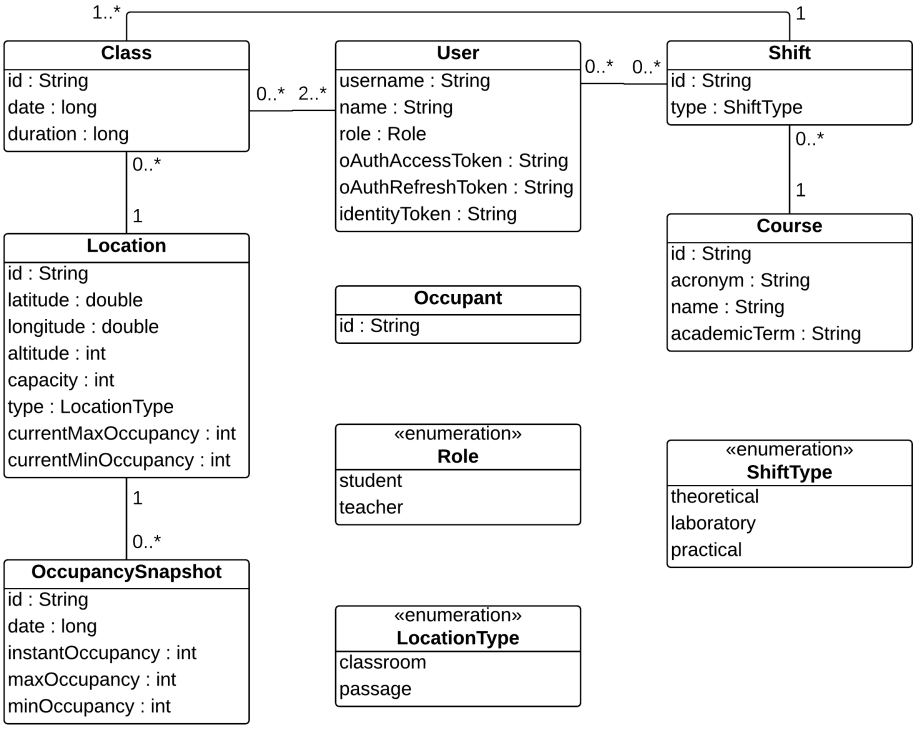


Figure 4.8: Entities managed by the AtOcu server.

- Scala is integrated in the Play Framework platform and was used to retrieve the information from the API module. When rendering a view, the template engine of the Play Framework generates HTML based on the Scala code.

The AtOcu Website comprises six pages and its site map is represented in Figure 4.9. All the pages but the login page are structured as a dashboard and thus they share a common side menu, header and footer.

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. On the other hand, if we considered that these users had IST credentials, for the purpose of this prototype we would not need to retrieve any additional information about them from the FenixEdu API.

After a successful login, the user is redirected to the overview page presented in Figure 4.10. On this page, the user can consult general numbers of the following entities:

- Registered and active users;
- Courses being monitored;
- Classes that were lectured;
- Rooms currently mapped.

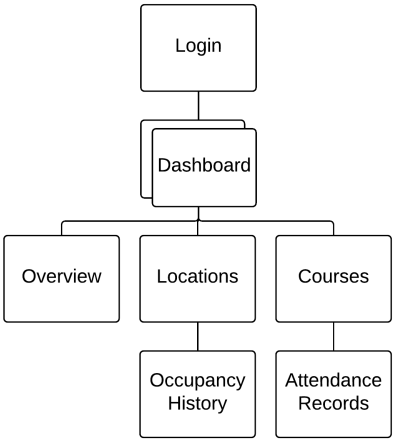


Figure 4.9: Site map of the AtOcu Website.

On the side menu, there are two additional options that give access to the locations and courses pages.

On the locations page (Figure 4.11), it is possible to consult the two types of locations mapped, rooms and locations, and their floor and the number of users currently occupying each location. In the particular case of rooms, it is also possible to consult the capacity of the room and if is currently occupied with classes.

By clicking in any of the listed locations, the user is redirected to the occupancy history page (Figure 4.12) of the selected location so that he could consult its occupancy history for a particular day. This information is presented in the form of a table or a chart.

The courses being monitored are listed on the courses page (Figure 4.13). In this page, each course is represented through its acronym, full name, its associated academic term and number of shifts.

In this case it is also possible to click in any of the listed courses to access its correspondent attendance records page (Figure 4.14). This page lists the lectured classes grouped by their respective shift and each one of these classes are represented by their date and time, location and number of attendees.

It must be noted that the users of the AtOcu website are just able to consult aggregated attendance and occupancy data collected by the AtOcu platform, and thus it is not possible to identify the targets' current or past position nor their own identity, or which specific students attended the listed lectured classes.

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

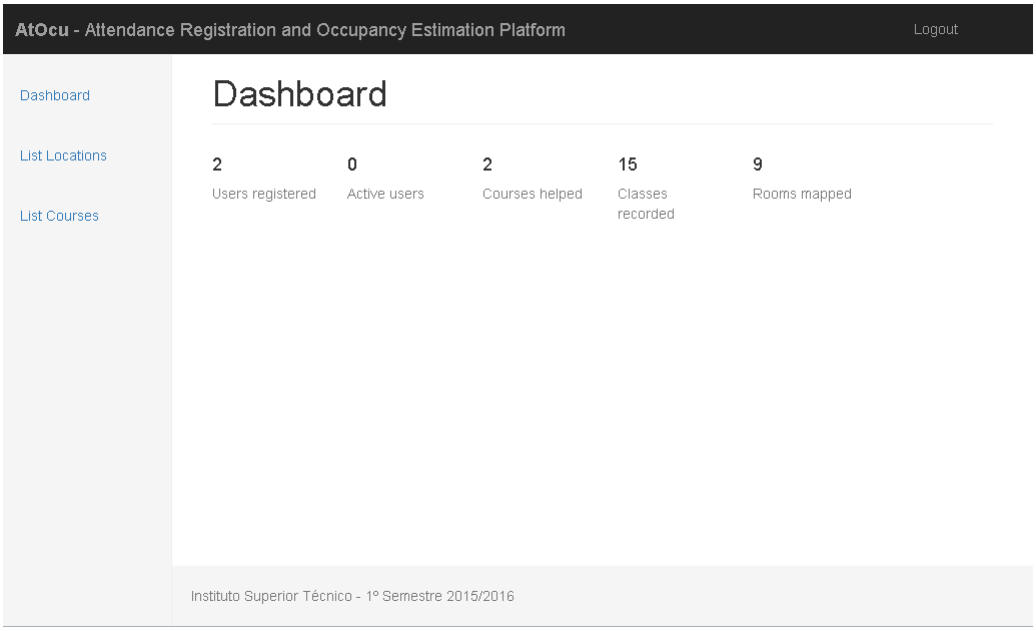


Figure 4.10: AtOcu website overview page.

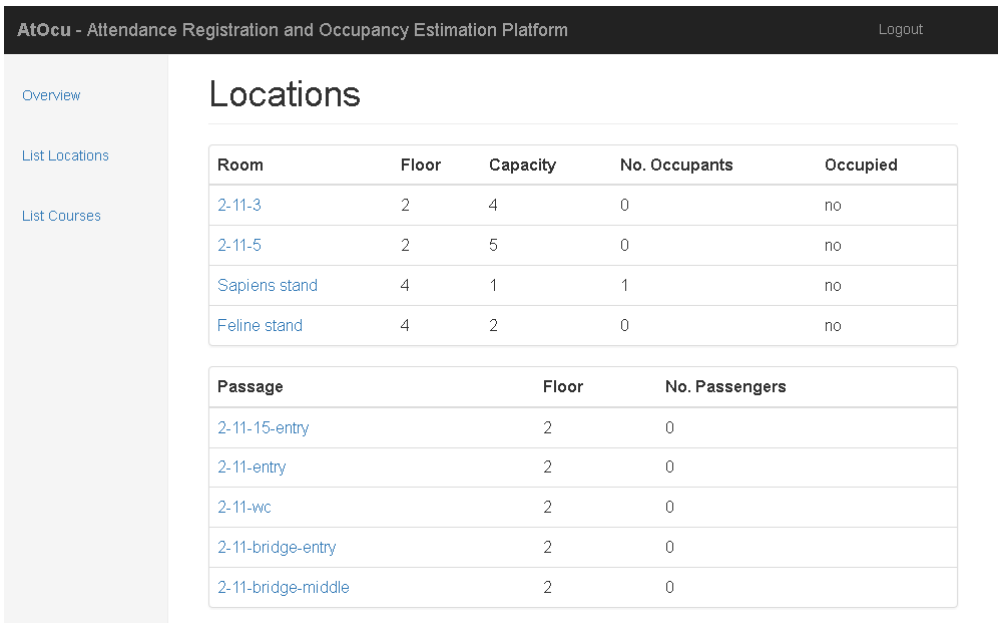


Figure 4.11: AtOcu website locations page.

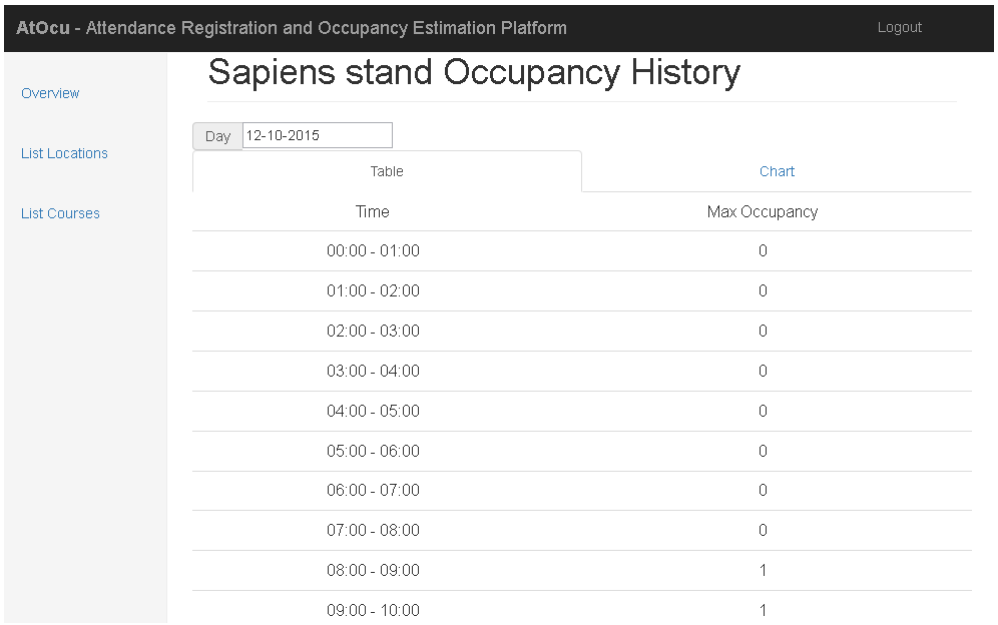


Figure 4.12: AtOcu website occupancy history page.

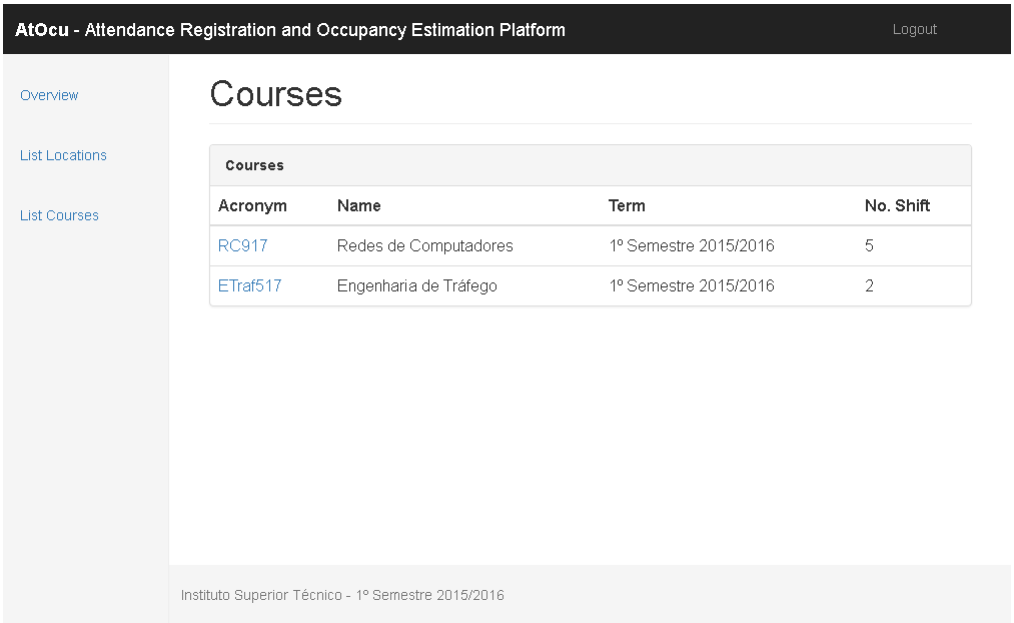


Figure 4.13: AtOcu website courses page.

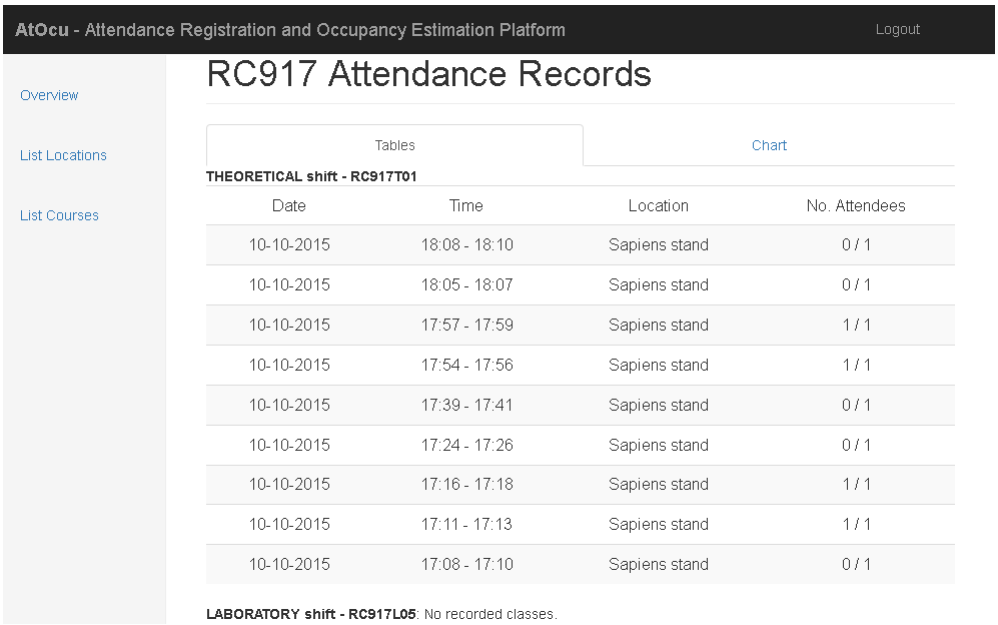


Figure 4.14: AtOcu website attendance records page.

Router component. We also created a routes file that defines the mapping between each HTTP request and the controller method responsible for its processing, so that the Router knows which action to invoke. Each one of the defined HTTP requests are listed with their HTTP method and endpoint URI, and optionally with query parameters.

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. Each one of these clients consumes different subsets of the API and the format of the response returned by the AtOcu server is also different for both. We used the JSON format for the AtOcu client and HTML for the web client.

On the other hand, the processing of HTTP requests can require different modules of the AtOcu server architecture, depending on the client that made them. The Router component analyses the routes file, decides which controller is responsible for generating a response and invokes the corresponding controller method (API module). In case the requests was previously made and there is a cache entry that satisfies it, the controller returns a response to the client by just accessing the cache (Cache module).

Otherwise, the controller invokes methods on the models (Entity module) to create or update any of the existing entities, or simply retrieves the required entities to satisfy the request. In case the request was made by the AtOcu client, the controller converts the entities obtained in the last step to the JSON format to be included in the body of the HTTP response. If, instead, the requester was the web client, the controller invokes the views to render the result of the request in HTML format so that it could include it in the body of the returned HTTP response.

The API endpoints consumed by the AtOcu client are the following:

GET /login has a code as query parameter to be used in FenixEdu SDK to obtain the OAuth 2.0 access

token and returns a redirect to a known URI. This redirect also provides an entity token and an anonymity token as query parameters.

GET /authorize returns a redirect to the authorization form of the FenixEdu IST instance.

GET /institutional returns general information about the institution, specifically its name and the current academic term. The returned response is cached.

GET /locations/mapping returns the locations represented by geographic coordinates (latitude, longitude and altitude) and a symbolic name. The returned response is cached.

GET /locations/occupancy returns the capacity and the current occupancy of all the rooms that are not currently occupied with lectures.

GET /user/{userId} returns general information about the user {userId}, namely its username, name and role (student or teacher).

GET /user/{userId}/courses returns the information about the courses (acronym, name and academic term) in which the user {userId} is currently enrolled.

GET /user/{userId}/classes returns the classes in which the user {userId} is currently enrolled.

GET /class/{classId}/attendance returns the number of registered students attending the class {classId}.

POST /user/{userId}/shift/{shiftId} enrolls the user {userId} in the shift {shiftId}.

POST /user/{userId}/class/{classId} registers the attendance of the user {userId} in the class {classId}.

POST /user/{userId}/location/{locationId} registers the detection of the user {userId} in the location {locationId}. For anonymity, the {userId} must be replaced by a random identity generated by the requester.

Whenever the AtOcu client invokes any of the endpoints previously referred, it needs to include the entity token in the query parameters. The only exception to the previous statement is the last endpoint listed, where instead of the entity token, the AtOcu client must include the anonymity token.

The API endpoints consumed by the web client used by the administrator or building management personnel are the following:

GET / returns the overview page.

GET /login returns the login page.

POST /login submits a form with username and password and returns a bad request if any of the credentials is wrong or a redirect to the overview page with a session token as a query parameter.

GET /locations returns the locations page.

GET /location/{locationId}/occupancy/history returns the occupancy history page of the location {locationId}.

GET /courses returns the courses page.

GET /course/{courseId}/classes returns the attendance records page of the course {courseId}.

4.3.4 Security

Secure connections

In order to enable 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.

We configured Nginx to just support TLS protocol version 1.2 because it is the most recent and thus secure version. This was only possible because the Android OS version we used during the development of the prototype is compatible with the chosen version of the TLS protocol. To guarantee that all requests are sent over HTTPS, we also configured Nginx to rewrite all HTTP requests to HTTPS requests.

User authentication

In the previous subsections we discussed the user login process from AtOcu client and AtOcu server individual perspectives. The complete login process is represented in Figure 4.15 and comprises the following steps:

1. The AtOcu client requests the authentication URL from the AtOcu server;
2. The AtOcu server redirects the AtOcu client to the authentication URL;
3. The FenixEdu instance returns a login web page so that the user can supply his IST credentials;
4. The user logs in successfully and authorizes the AtOcu server to access his personal information;
5. The FenixEdu instance redirects the AtOcu client to the login URL with the authorization code as a query parameter;
6. The AtOcu server requests the user's access token supplying the authorization code as a query parameter;
7. The FenixEdu instance returns the access and refresh token;
8. The AtOcu server returns the identity and anonymity token;

By using this login sequence, we assumed that the FenixEdu instance deployed at IST was a trustful entity and thus we could rely on its services to authenticate the users of the AtOcu platform.

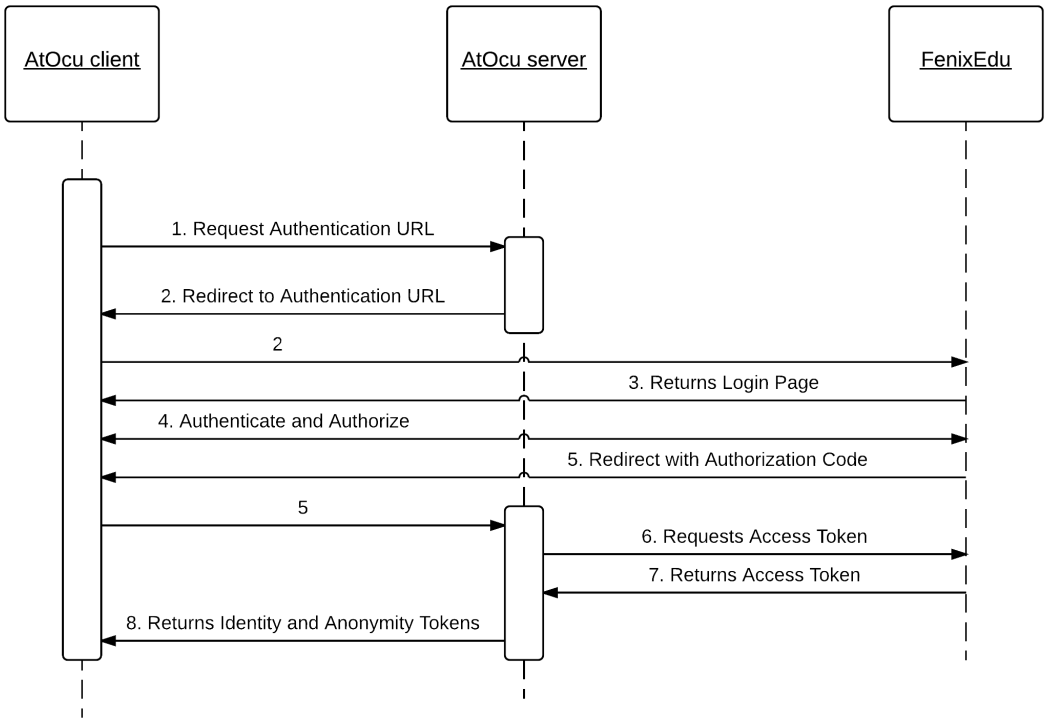


Figure 4.15: Interaction between AtOcu components and FenixEdu during user's login.

4.4 Concluding Observations

In this chapter, we described the implementation of a prototype of the AtOcu platform. The prototype was developed for the Taguspark campus of IST and took advantage of the local Academic Management System, the FenixEdu platform, in order to obtain information about courses, their schedules, enrolled students and teachers.

Considering that the Taguspark campus is covered by a WLAN based on the IEEE 802.11 standard, we chose an IPS based on the Wi-Fi technology, the SmartCampusAAU platform, to be integrated with the AtOcu prototype. The SmartCampusAAU platform enabled the AtOcu prototype to estimate locations with room-level accuracy at the client level.

Both components of the AtOcu platform, the client and the server components, were developed using open source technologies.

The AtOcu client component was developed as a mobile application for the Android OS. We took advantage of the system alarm services of this OS to schedule the estimation of the user's location at every minute and verify the user's attendance at classes. Once collected, this data is then sent to the AtOcu server and it constitutes part of the contextual information that the user of the AtOcu client can consult through a GUI.

On the other hand, the AtOcu server was developed using the Play Framework, which follows the REST architectural style. The implemented AtOcu server offers an web API in order to expose its functionality, and it is accessed by each AtOcu client through HTTPS. Apart of serving client requests,

the AtOcu server also takes occupancy snapshots at every hour and makes this information, as well as aggregate attendance information, available through a web GUI, the AtOcu website. This website can be consulted by the administrator of the AtOcu platform or building management personnel.

Chapter 5

Evaluation

During the development of the AtOcu prototype described in Chapter 4, unit tests were performed to verify the correct functionality of the individual modules implemented in AtOcu client and server components and their respective integration. On the other hand, since the AtOcu client requires the user to be logged in to use its functionalities, we created mock users that were associated with real courses that were ongoing during the current academic term, not only to test the download of schedule specific information but also to accelerate the testing of the interface module of the AtOcu client.

In this Chapter we describe the experiments that were conducted to validate the proposed solution through a set of measurable tests applied to the implemented prototype. In Section 5.1 we describe the deployment of the prototype. The experiments performed to verify the integration of the prototype and the SmartCampusAAU solution and the positioning performance of the final system are presented in Section 5.2. Section 5.3 and Section 5.4 describe, respectively, the behavior of the AtOcu server when increasing the number of requests and the analysis of the response times obtained by the AtOcu client.

5.1 Prototype Deployment

The AtOcu server was deployed in a Virtual Private Server (VPS) provided by DigitalOcean[1] where we also installed an instance of the PostgreSQL database. This VPS had 512 megabytes of dedicated Random Access Memory (RAM), 1 processor core, 20 gigabytes of disk space and it was running the Ubuntu OS, specifically the 14.04.2 version.

The AtOcu client was installed in a tablet, the Asus Nexus 7, and in a smartphone, the Samsung Galaxy Ace Style (Figure 5.1). Both of these devices were running the Android version 4.4 and are compatible with the Wi-Fi technology. Besides the AtOcu client, the SmartCampusAAU app was also installed in the tablet device.

As previously referred in Chapter 4, we used the SmartCampusAAU server that was running at Aalborg University, not only for the development of the AtOcu prototype, but also during its evaluation.

[1] DigitalOcean, https://www.digitalocean.com/

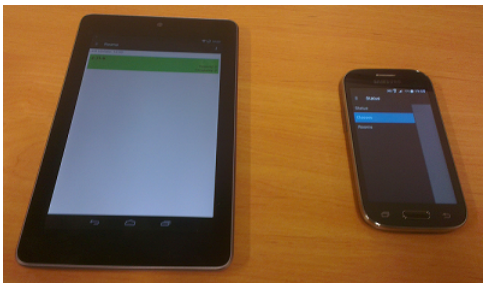


Figure 5.1: Handheld systems utilized to run the AtOcu client.

5.2 SmartCampusAAU Integration

The SmartCampusAAU IPS served as an enabling technology for the correct operation of the AtOcu platform, and for that reason, we decided to test their integration.

5.2.1 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 are particularly important because the AtOcu platform provides LBSs that depend on the correctness of the location information to automate the attendance registration and occupancy estimation, and the subsequent functionality that is dependent on these services.

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 needed to map the rooms and the passage locations that are part of their surroundings. For that purpose we used the adapted SmartCampusAAU app to assist us with the process of creating a radio map, specifically by:

1. Adding the Taguspark campus of the IST as a building and its floors where the rooms were located;
2. Collecting 50 Wi-Fi scans at a stationary point in the desired location. If the location was a room, the stationary point corresponded to its approximate geometric center of its floor area.
3. Sending the identification and capacity (ignored in case of passage locations) of the newly mapped location to the AtOcu server through its API.

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.

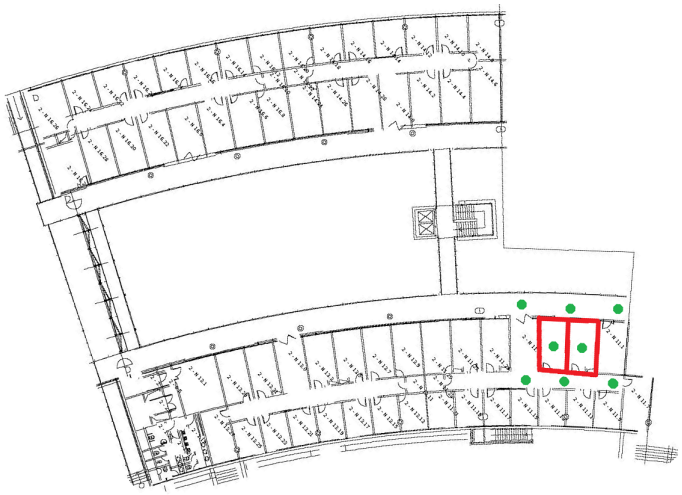
The first scenario involved two contiguous offices, office 2-11-3 and office 2-11-5, on the second floor of the IST - Taguspark main building, whose relative location and respective mapping are presented in Figure 5.2(a). These offices represent one of the smallest types of rooms used in IST for academic activities.

The second scenario involved two contiguous lecture halls, A4 and A5, on the ground floor of the IST - Taguspark main building. Their relative location and respective mapping is presented in Figure 5.2(b). Lecture halls are the largest types of rooms that can be found on campus that are used for scheduled classes.

In the first scenario, we obtained correct estimations in all tests in 50% of the defined positions for the office 2-11-3 (5.3(a)) and in 55.56% of the defined positions for the office 2-11-5 (5.3(b)). 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 (5.3(c)) and lecture hall A4 (5.3(d)).

In Figure 5.4, we provide a different representation of the data obtained by associating different colors that represent the number of successful location estimations in each tested position, considering all tests performed.

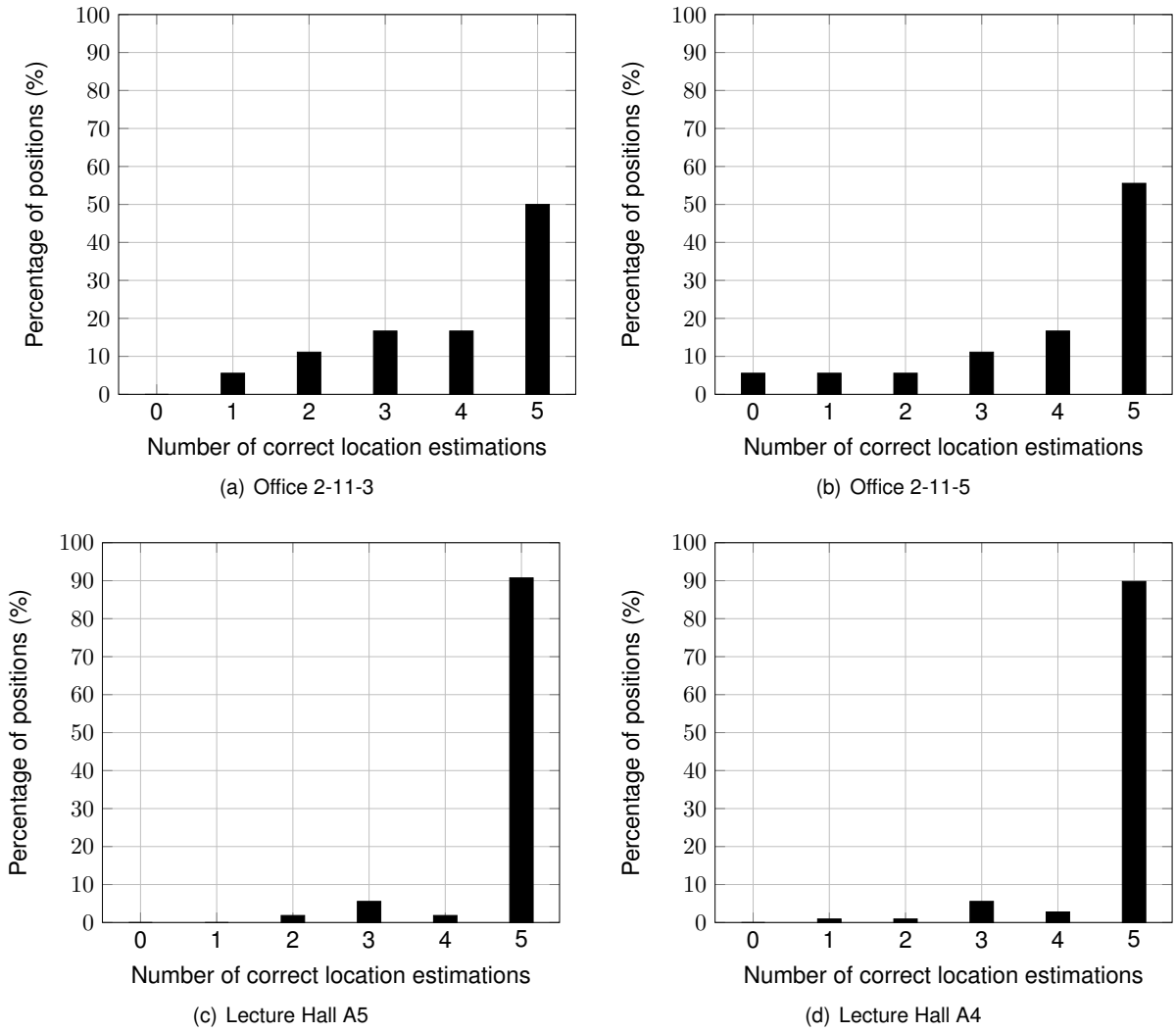
The results obtained for the positioning tests demonstrated that the performance of the system was not uniform across the whole area of the rooms tested. However, the positions in which the system performed poorly were relatively near passage points such as doors where the user is not expected to stay 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 at least one correct location estimation, we consider that the results obtained confirm that the AtOcu platform reached room-level accuracy when integrated with the SmartCampusAAU platform.



(a) Contiguous offices 2-11-3 and 2-11-5.

(b) Contiguous lecture halls A5 and A4.

Figure 5.2: Blueprint section of the rooms used for positioning tests. The rooms and mapped positions are represented, respectively, by red rectangles and green dots.



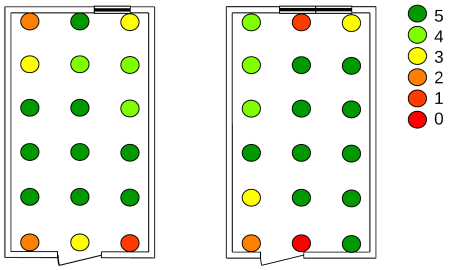
(a) Office 2-11-3

(b) Office 2-11-5

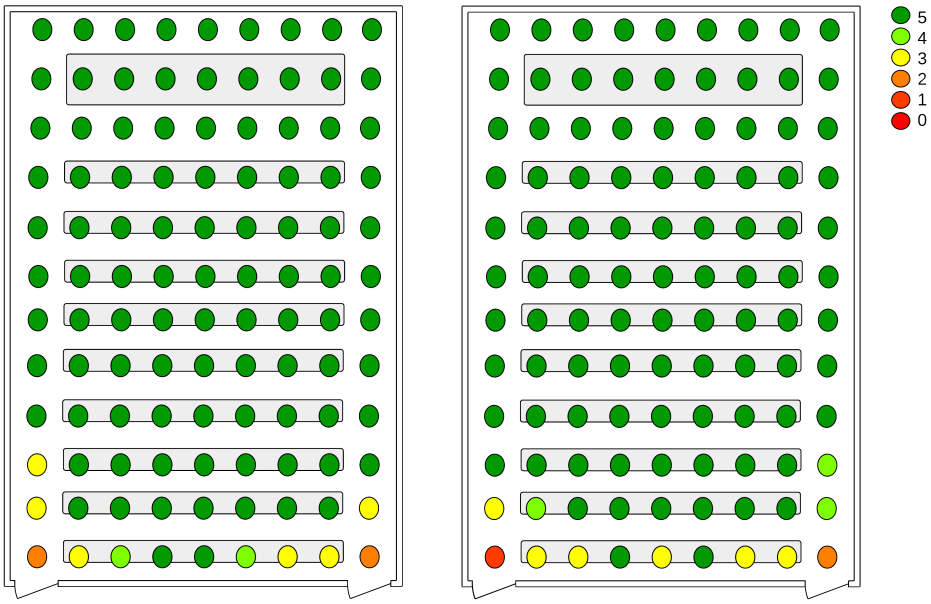
(c) Lecture Hall A5

(d) Lecture Hall A4

Figure 5.3: Results of the positioning tests.



(a) Office 2-11-3 (b) Office 2-11-5



(c) Lecture Hall A5 (d) Lecture Hall A4

Figure 5.4: Results of the positioning tests per tested position. Each position is associated with a color that represents the number of successful location estimations produced by the prototype during the positioning tests.

5.3 AtOcu server

5.3.1 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 AtOcu clients.

AtOcu clients access the AtOcu server functionality through its API. For this reason, we selected the following three API endpoints that request the AtOcu server to perform different actions:

GET /locations/mapping This endpoint returns a cached response;

POST /user/{userId}/location/{locationId} This endpoint requires the AtOcu server to write or update values from the database;

GET /locations/occupancy This endpoint returns non-cached resources and requires the AtOcu server to read values from the database and cache.

The AtOcu prototype had the Taguspark campus of IST as an implementation scenario. Considering that the entire IST has around a combined number of 12350 students and academic staff[2], and that the Taguspark campus is significantly smaller in terms of bachelor and master programmers offered (8 programmes in Taguspark campus against 48 programmes in Alameda campus), we decided to test AtOcu server to support at least half of the previously mentioned number of students and academic staff.

Since we were limited in number of physical handheld systems to run tests that involved a increasing number of requests up to that defined limit, we used the Loader[3] cloud-based load and scalability testing service.

Using this service, we run a set of tests where we specified the total number of clients to invoke the respective API endpoint over the duration of 60 seconds. In other words, if we specify a test to run with 6000 clients for 60 seconds, 100 clients will be connected at each second during the test.

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 5.5. 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

[2]https://tecnico.ulisboa.pt/en/about-IST/facts-figures/ (2015-07-30)
[3]Loader, https://loader.io/

and reached its upper limit at 7000 clients per test with an average response time of 1934 milliseconds. Nevertheless, the AtOcu server was able to respond to all requests.

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.

5.4 AtOcu client

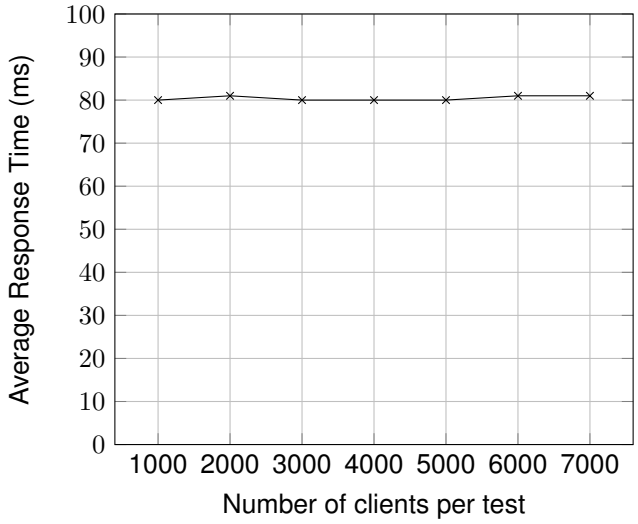
5.4.1 Response Time Tests

When testing the performance of the AtOcu server in dealing with an increasing number of requests, the endpoint `/locations/occupancy` was the only one of those tested that showed a significant increase in the average response time. This particular endpoint can be invoked regularly by the AtOcu client so that the user can access room occupancy information. Therefore, it is of practical importance that this information does not take a long time to be received and shown to the user.

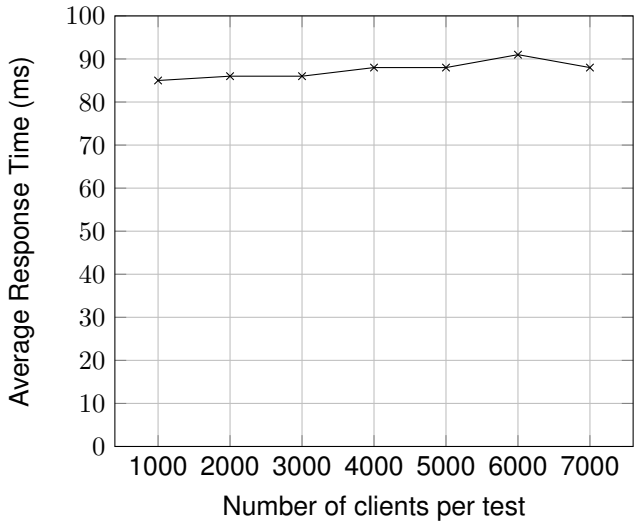
For that reason, we decided to perform a response time test targeting the previously referred endpoint by using handheld systems running the AtOcu client, to obtain results from the client perspective.

Using each handheld system, we performed 10 requests and took note of the respective response times. These requests were performed sequentially while the AtOcu server was having a load of 20 requests per second.

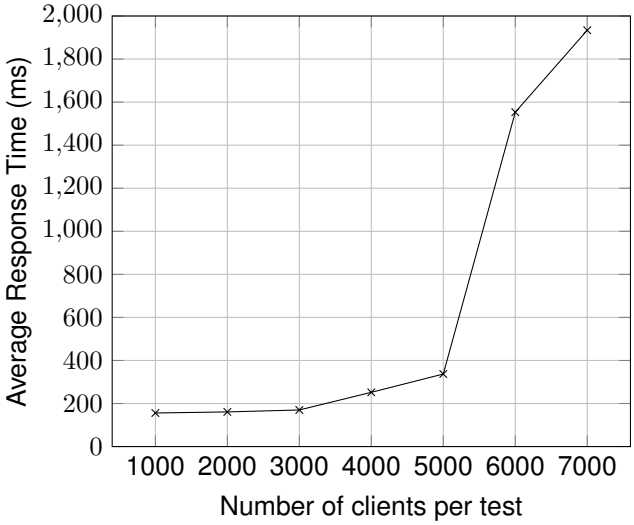
The average response times obtained by the smartphone and the tablet were, respectively, 80 and 89 milliseconds. The complete results of the response time tests that we performed with both devices are presented in Figure 5.6.



(a) /locations/mapping



(b) /user/{userId}/location/{locationId}



(c) /locations/occupancy

Figure 5.5: Results of the load tests for the AtOcu server.

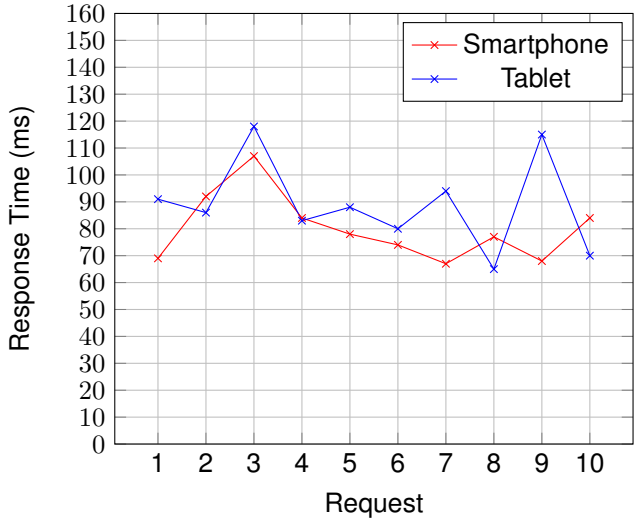


Figure 5.6: Results of the response time tests for the AtOcu client.

Chapter 6

Conclusions

This dissertation addressed the problem of collecting student attendance at lectures and estimating the number of occupants in class and study rooms. Although this problem was identified in the particular case of Instituto Superior Técnico (IST), it could also be applied in other education institutions and contexts.

To solve this problem, we proposed the AtOcu (Attendance and Occupation) platform that makes use of the most common handheld systems currently used by students and teachers and interacts with an Indoor Positioning System (IPS) to provide Location-Based Services (LBSs) that automate the student attendance registration and the room occupancy estimation.

To build a better understanding about the architectural and privacy implications of the services being proposed, we analyzed the different categories and architectures of LBSs. We also compared various IPSs with the purpose of identifying the best technologies to provide location information in the context of our problem. We found that the Radio Frequency (RF) based technologies were the most suitable for our solution, with emphasis on systems that take advantage of Wireless Fidelity (Wi-Fi).

Based on the information gathered, we designed the architecture of the AtOcu platform to be modular, extensible and IPS independent. We also took into consideration the preservation of users' anonymity. A prototype was then developed for the IST - Taguspark scenario that took advantage of its Wireless Local Area Network (WLAN) and an existing IPS, the SmartCampusAAU solution, to estimate the location of its users and consecutively estimate room occupancy information. 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. The data collected by the prototype was then aggregated and was made available through a website that could be consulted by the prototype's administrator or building management personnel.

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 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. More specifi-

cally, the prototype was found to support at least 7000 users that made an request over the duration of 1 minute, which is sufficient to handle half of the number of students and teachers at IST. On the other hand, the prototype was also successful in generating correct location estimations in the majority of positions found inside the rooms tested, which is sufficient for the correct operation of the LBSs offered, namely the attendance registration and occupancy estimation.

Considering the evaluation of the prototype, we concluded that the proposed 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.

6.1 Future Work

While developing and evaluating the AtOcu prototype, we have reasoned about improvements and new functionalities that could be applied to the proposed system in future prototypes. Some of these ideas are:

- The AtOcu platform could take advantage of Dead Reckoning (DR) techniques to infer when it is really necessary to generate new location estimations and thus save resources spent in collecting a radio fingerprinting. Currently, the AtOcu client generates location estimations at every 60 seconds when the user is not attending classes. Considering situations in which the user is staying in the same room for long periods of time, for instance while studying, the AtOcu client could use the handheld system's accelerometer, or other appropriate sensors, to detect movements of its user and act accordingly. An appropriate algorithm would need to be developed to differentiate situations wherein the user is moving within the limits of its current location from situations in which the user is leaving his last detected location. Taking advantage of the modularity of the system, this feature could be incorporated in the location module of the AtOcu client.

- The Application Programming Interface (API) of the AtOcu server could be extended to expose some of the data being collected. This feature would favor the adoption of the AtOcu platform by academic management systems and building management systems that would like to improve their services with the data offered.

- To deploy the current AtOcu prototype, it is necessary to previously use the SmartCampusAAU app to map new locations, which could be a slow process if we consider the total number of rooms that an university normally has. To counteract this fact, the AtOcu platform could be extended to automate this process by relying on crowdsourcing. Teachers could be labeled as trustful entities so that their AtOcu client could start collecting radio fingerprintings automatically after one of their classes had begun. Then, these newly collected fingerprintings would need to pass through an inspection process in which it was determined if the user was effectively where he was supposed to. The inspection process could involve the students that are supposed to be attending the class or, instead, successive radio fingerprintings could be collected across a determined period of time.

- During the development of the AtOcu platform's architecture, we assumed that the handheld system in which the AtOcu client would be running was not compromised and that the user was not going to abuse the API provided, since these concerns would fall outside the scope of this work. Although the data being collected might not have much value associated, it would still be interesting to explore and analyze methodologies to prevent the opposite of our assumptions from happening and thus contributing to improve the AtOcu platform to resist such threats. For instance, if we consider that an user is able to access the source code of the AtOcu client and extract the necessary data to invoke API endpoints, such as his own tokens, he could register his attendance to classes he is not attending or register itself multiple times as an occupant to interfere with the occupancy LBS.

Bibliography

[1] Barata, R.S.: Mobility and location in academic information systems. Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico (2013)

[2] Junglas, I.A., Watson, R.T.: Location-based services. Communications of the ACM **51**(3) (2008) 65–69

[3] Hightower, J., Borriello, G.: Location systems for ubiquitous computing. Computer **34**(8) (2001) 57–66

[4] Barkhuus, L., Dey, A.K.: Location-based services for mobile telephony: a study of users' privacy concerns. In: INTERACT. Volume 3., Citeseer (2003) 702–712

[5] Bellavista, P., Kupper, A., Helal, S.: Location-based services: Back to the future. Pervasive Computing, IEEE **7**(2) (2008) 85–89

[6] Rodas, J., Barral, V., Escudero, C.J.: Architecture for multi-technology real-time location systems. Sensors **13**(2) (2013) 2220–2253

[7] Hightower, J., Brumitt, B., Borriello, G.: The location stack: A layered model for location in ubiquitous computing. In: Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on, IEEE (2002) 22–28

[8] Kushki, A., Plataniotis, K.N., Venetsanopoulos, A.N.: WLAN Positioning Systems: Principles and Applications in Location-based Services. Cambridge University Press (2012)

[9] Schiller, J., Voisard, A.: Location-based services. Elsevier (2004)

[10] Beresford, A.R., Stajano, F.: Location privacy in pervasive computing. Pervasive Computing, IEEE **2**(1) (2003) 46–55

[11] Krumm, J.: A survey of computational location privacy. Personal and Ubiquitous Computing **13**(6) (2009) 391–399

[12] Cellular Telephone Industries Association: Best practices and guidelines for location based services (2010)

[13] Liu, H., Darabi, H., Banerjee, P., Liu, J.: Survey of wireless indoor positioning techniques and systems. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on **37**(6) (2007) 1067–1080

[14] Stojanović, D., Stojanović, N.: Indoor localization and tracking: Methods, technologies and research challenges. Facta Universitatis, Series: Automatic Control and Robotics **13**(1) (2014) 57–72

[15] Das, S.M., Koutsonikolas, D., Hu, Y.C., Peroulis, D.: Characterizing multi-way interference in wireless mesh networks. In: Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization, ACM (2006) 57–64

[16] Kjærgaard, M.B.: A taxonomy for radio location fingerprinting. In: Location-and Context-Awareness. Springer (2007) 139–156

[17] Gu, Y., Lo, A., Niemegeers, I.: A survey of indoor positioning systems for wireless personal networks. Communications Surveys & Tutorials, IEEE **11**(1) (2009) 13–32

[18] Lane, N.D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., Campbell, A.T.: A survey of mobile phone sensing. Communications Magazine, IEEE **48**(9) (2010) 140–150

[19] Sharp, I., Yu, K.: Sensor-based dead-reckoning for indoor positioning. Physical Communication (2013)

[20] Chen, Y.C., Chiang, J.R., Chu, H.h., Huang, P., Tsui, A.W.: Sensor-assisted wi-fi indoor location system for adapting to environmental dynamics. In: Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems, ACM (2005) 118–125

[21] Enge, P., Misra, P.: Special issue on global positioning system. Proceedings of the IEEE **87**(1) (1999) 3–15

[22] Society, I.C.: Ieee 802.11 - ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications (2012) (accessed at 2015-01-07).

[23] Bahl, P., Padmanabhan, V.N.: Radar: An in-building rf-based user location and tracking system. In: INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Volume 2., Ieee (2000) 775–784

[24] Hansen, R., Thomsen, B., Thomsen, L.L., Adamsen, F.S.: Smartcampusaau–an open platform enabling indoor positioning and navigation. In: Mobile Data Management (MDM), 2013 IEEE 14th International Conference on. Volume 2., IEEE (2013) 33–38

[25] Bolliger, P.: Redpin-adaptive, zero-configuration indoor localization through user collaboration. In: Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, ACM (2008) 55–60

[26] Weissman, Z.: Indoor location. White paper, Tadlys Ltd (2004)

[27] Contreras, D., Castro, M., Torre, D.S.: Performance evaluation of bluetooth low energy in indoor positioning systems. Transactions on Emerging Telecommunications Technologies (2014)

[28] Zhao, X., Xiao, Z., Markham, A., Trigoni, N., Ren, Y.: Does btle measure up against wifi? a comparison of indoor location performance. In: European Wireless 2014; 20th European Wireless Conference; Proceedings of, VDE (2014) 1–6

[29] Want, R., Hopper, A., Falcao, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems (TOIS) **10**(1) (1992) 91–102

[30] Curran, K., Furey, E., Lunney, T., Santos, J., Woods, D., McCaughey, A.: An evaluation of indoor location determination technologies. Journal of Location Based Services **5**(2) (2011) 61–78

[31] Hightower, J., Want, R., Borriello, G.: Spoton: An indoor 3d location sensing technology based on rf signal strength. UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA **1** (2000)

[32] Ni, L.M., Liu, Y., Lau, Y.C., Patil, A.P.: Landmarc: indoor location sensing using active rfid. Wireless networks **10**(6) (2004) 701–710

[33] Montaser, A., Moselhi, O.: Rfid indoor location identification for construction projects. Automation in Construction **39** (2014) 167–179

[34] Ozdenizci, B., Ok, K., Coskun, V., Aydin, M.N.: Development of an indoor navigation system using nfc technology. In: Information and Computing (ICIC), 2011 Fourth International Conference on, IEEE (2011) 11–14

[35] Society, I.C.: Ieee 802.15.4 - ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) (2011) (accessed at 2015-01-07).

[36] Sugano, M., Kawazoe, T., Ohta, Y., Murata, M.: Indoor localization system using rssi measurement of wireless sensor network based on zigbee standard. Target **538** (2006) 050

[37] Hu, X., Cheng, L., Zhang, G.: A zigbee-based localization algorithm for indoor environments. In: Computer Science and Network Technology (ICCSNT), 2011 International Conference on. Volume 3., IEEE (2011) 1776–1781

[38] Steggles, P., Gschwind, S.: The ubisense smart space platform. White paper, Ubisense (2005)

[39] Priyantha, N.B.: The cricket indoor location system. PhD thesis, Massachusetts Institute of Technology (2005)

[40] Lo, D., Mendonça, P.R., Hopper, A., et al.: Trip: A low-cost vision-based location system for ubiquitous computing. Personal and Ubiquitous Computing **6**(3) (2002) 206–219

[41] Stillman, S.T., Tanawongsuwan, R., Essa, I.A.: A system for tracking and recognizing multiple people with multiple camera. (1998)

[42] Shoewu, O., Idowu, O.: Development of attendance management system using biometrics. The Pacific Journal of Science and Technology **13**(1) (2012)

[43] Li, T.C., Wu, H.W., Wu, T.S.: The study of biometrics technology applied in attendance management system. In: Digital Manufacturing and Automation (ICDMA), 2012 Third International Conference on, IEEE (2012) 943–947

[44] Avireddy, S., Veerapandian, P., Ganapati, S., Venkat, M., Ranganathan, P., Perumal, V.: Mitsat - an automated student attendance tracking system using bluetooth and eyeos. In: Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on, IEEE (2013) 547–552

[45] Saad, T., Waqas, A., Mudassir, K., Naeem, A., Aslam, M., Ayesha, S., Martinez-Enriquez, A., Hedz, P.R.: Collaborative work in class rooms with handheld devices using bluetooth and wlan. In: Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on, IEEE (2014) 1–6

[46] Jamil, T.: Automatic attendance recording system using mobile telephone. In: Telecommunications Forum (TELFOR), 2011 19th, IEEE (2011) 1297–1299

[47] Yeop Sabri, M., Abdul Aziz, M., Shah, M., Abd Kadir, M.: Smart attendance system by using rfid. In: Applied Electromagnetics, 2007. APACE 2007. Asia-Pacific Conference on, IEEE (2007) 1–4

[48] Kassim, M., Mazlan, H., Zaini, N., Salleh, M.K.: Web-based student attendance system using rfid technology. In: Control and System Graduate Research Colloquium (ICSGRC), 2012 IEEE, IEEE (2012) 213–218

[49] Al-Barhamtoshy, H.M., Altalhi, A.H., Mashat, A.S.: Academic monitoring in classrooms using rfid technology. In: The International Conference on Digital Information Processing, E-Business and Cloud Computing (DIPECC2013), The Society of Digital Information and Wireless Communication (2013) 204–209

[50] Subpratatsavee, P., Promjun, T., Siriprom, W., Sriboon, W.: Attendance system using nfc technology and embedded camera device on mobile phone. In: Information Science and Applications (ICISA), 2014 International Conference on, IEEE (2014) 1–4

[51] Patel, U.A.: Development of a student attendance management system using rfid and face recognition: A review. **2** (2014)

[52] Fielding, R.T.: Architectural styles and the design of network-based software architectures. PhD thesis, University of California, Irvine (2000)

[53] Dierks, T.: The transport layer security (tls) protocol version 1.2. (2008)

[54] Krasner, G.E., Pope, S.T., et al.: A description of the model-view-controller user interface paradigm in the smalltalk-80 system. (1988)

[55] Gu, Y., Zhang, J., Wang, J., Gao, B., Du, J.: Research on rfid indoor localization algorithm. AICT 2014 **52** (2014) 70–75

[56] Gezici, S., Poor, H.V.: Position estimation via ultra-wide-band signals. Proceedings of the IEEE **97**(2) (2009) 386–403

[57] Yamasaki, R., Ogino, A., Tamaki, T., Uta, T., Matsuzawa, N., Kato, T.: Tdoa location system for ieee 802.11 b wlan. In: Wireless Communications and Networking Conference, 2005 IEEE. Volume 4., IEEE (2005) 2338–2343

[58] Hammadi, O.A., Hebsi, A.A., Zemerly, M.J., Ng, J.W.: Indoor localization and guidance using portable smartphones. In: Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03, IEEE Computer Society (2012) 337–341

[59] Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-camera multi-person tracking for easyliving. In: Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on, IEEE (2000) 3–10

[60] Ficco, M., Palmieri, F., Castiglione, A.: Hybrid indoor and outdoor location services for new generation mobile terminals. Personal and ubiquitous computing **18**(2) (2014) 271–285

[61] dos Santos Carvalho Leite, L.A.: Context guide context and location-awareness for an indoor mobile guide. Master's thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico (2010)

[62] Kamath, A., et al.: Indoor location-based services in the telecommunications network. Master's thesis, Aalto University, School of Electrical Engineering, Department of Communications and Networking (2014)

[63] Wang, H.T., Jia, Q.S., Song, C., Yuan, R., Guan, X.: Estimation of occupancy level in indoor environment based on heterogeneous information fusion. In: Decision and Control (CDC), 2010 49th IEEE Conference on, IEEE (2010) 5086–5091

[64] Hansen, R., Wind, R., Jensen, C.S., Thomsen, B.: Seamless indoor/outdoor positioning handover for location-based services in streamspin. In: Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on, IEEE (2009) 267–272

[65] Hansen, R., Thomsen, B.: Efficient and accurate wlan positioning with weighted graphs. In Granelli, F., Skianis, C., Chatzimisios, P., Xiao, Y., Redana, S., eds.: Mobile Lightweight Wireless Systems. Volume 13 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg (2009) 372–386