

Masters Degree in Informatics Engineering
Internship
Final Report

Mobile Tolling – A Geotracking Capable Mobile Application for Toll Fee Collection

Miguel Lopes de Almeida
mtac@student.dei.uc.pt

Advisors:

Licínio Roque, Phd

António Ricardo Pinto

Date: 27-06-2014



FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Abstract

The traditional toll collection process uses two major charging techniques: manual collection, where the vehicle is forced to stop at toll plazas, is a reliable but both time consuming and costly method, and electronic collection, where the vehicles do not need to stop at toll plazas and are identified by an On-Board Unit (OBU), which still carries production and maintenance costs and forces the vehicles without an OBU to use the manual collection method.

The necessity to improve the amount of time lost, the operational costs with manual toll operators as well as the need to implement tolling in already built roads, gave way to the creation of an automatic license plate recognition (ALPR) method so that multi-lane free flow scenarios were possible. This removed the time consumption constraint from the process but since the state of the art in ALPR still only achieves 85% reliability using OCR (Optical Character Recognition) engines and 96% reliability using Fingerprint engines, global operational costs processing the unidentified cars, which have to be manually validated, are still very high when we talk of handling millions of image recognitions per day.

The goal of this internship is not to replace the OBU, but to study a method of improving the success rate of the automatic vehicle recognition by using mobile phones geolocalization services to track the vehicle position as a mean of identifying that the vehicle passed through a certain toll point. Using the events from both the ALPR and the Mobile Tolling application, the back-office can correlate the two, improving the confidence level for the identification close to 100%, reducing the operation costs with negative and false positive identifications.

Keywords

Mobile Tolling, Electronic Toll Collection, Automatic License Plate Recognition, Vehicle Identification, Vehicle Position Tracking, Geolocalization.

Table of Contents

1.	Introduction	2
1.1.	About the Project Context.....	2
1.2.	Objectives	2
1.2.1.	Mobile Tolling Back-Office Correlation Module	3
1.2.2.	Mobile Tolling Mobile Applications	3
1.3.	Scope.....	3
2.	State of the art.....	6
2.1.	Tolling Solutions.....	6
2.1.1.	GeoToll.....	7
2.1.2.	BancPass PToll.....	7
2.1.3.	The Toll Roads.....	8
2.1.4.	PayUrToll.....	8
2.2.	Comparison	9
3.	Methodology	12
3.1.	Accenture Software Development Methodology	12
3.2.	Activity Plan according to the methodology	13
3.3.	Risk Assessment.....	16
4.	Requirements Specification	18
4.1.	Authentication.....	18
4.2.	Account Management	18
4.3.	Location Management.....	19
4.4.	Event Management.....	19
4.5.	Notifications.....	20
4.6.	Bluetooth Management.....	20
4.7.	Back-Office	20
5.	Architecture.....	22
5.1.	Project Integration	23
5.2.	Mobile Tolling Application Overview	24
5.2.1.	Model-View-Controller	25
5.2.2.	Business Logic.....	26
5.2.3.	Application Modules.....	29
5.2.4.	User Interface & User Experience (UI/UX)	31
5.2.5.	Persistence	32

5.3.	Data Model Overview	32
6.	Quality Assurance	36
6.1.	Tests Definition	36
6.1.1.	Functional Tests	36
6.1.2.	Technical Tests	37
6.1.3.	Back-Office Tests	37
6.2.	Tests Execution	38
6.3.	Test Results and Analysis	38
7.	Conclusion	44
7.1.	Results Achieved	44
7.2.	Future Work	44
7.2.1.	Risk Pending	44
7.3.	Lessons Learned	45
8.	Bibliography	46

List of Figures

Figure 1 - Accenture Structure of Services.....	2
Figure 2 - Free Flow Electronic Video Toll Collection (Source [7]).....	6
Figure 3 - V-Model Software Development Methodology (Source [4])	12
Figure 4 - Mobile Tolling: Internship Plan.....	14
Figure 5 - Mobile Tolling: System Overview.....	22
Figure 6 - Mobile Tolling Back-Office: Correlation Module Process Flow.....	23
Figure 7 - Mobile Tolling: Application Modules Overview	25
Figure 8 - Mobile Tolling: Model-View-Controller architecture adaptation.....	25
Figure 9 - Mobile Tolling: MVC Details.....	26
Figure 10 - Mobile Tolling: Location Provider change process.....	26
Figure 11 - Mobile Tolling: Gantry Crossing Detection Process	27
Figure 12 - Mobile Tolling: Gantry Map	28
Figure 13 - Mobile Tolling: Conceptual Architecture Diagram	30
Figure 14 - Mobile Tolling: Application Flow Chart.....	31
Figure 15 - Mobile Tolling: Application ER Diagram.....	33
Figure 16 - Mobile Tolling: Back-Office ER Diagram.....	34
Figure 17 - Mobile Tolling: Virtual Toll Point Crossing Event	38
Figure 18 - Mobile Tolling: All Results Standard Deviation and Mean Graph.....	39
Figure 19 - Mobile Tolling: Relevant Events.....	40
Figure 20 - Mobile Tolling: Useful Results Standard Deviation and Mean Graph.....	40
Figure 21 - Mobile Tolling: Normal Distribution Definition (Source [32])	41
Figure 22 - Mobile Tolling: Results Histogram.....	42
Figure 23 - Mobile Tolling: Results Normal Distribution.....	42
Figure 24 - Mobile Tolling: Improved Events	43

List of Tables

Table 1 - Mobile Tolling: Tolling Solutions Comparison.....	10
Table 2 - Mobile Tolling: Authentication Requirements	18
Table 3 - Mobile Tolling: Account Management Requirements.....	19
Table 4 - Mobile Tolling: Location Management Requirements.....	19
Table 5 - Mobile Tolling: Event Management Requirements	19
Table 6 - Mobile Tolling: Notifications Requirements	20
Table 7 - Mobile Tolling: Bluetooth Management Requirements.....	20
Table 8 - Mobile Tolling: Back-Office Requirements	21
Table 9 - Mobile Tolling: Business Logic Classes	29
Table 10 - Mobile Tolling: Modules Description	30
Table 11 - Mobile Tolling: Main Classes Description	32
Table 12 - Mobile Tolling: Persistence Classes	32
Table 13 - Mobile Tolling: Main Tables Description	33
Table 14 - Mobile Tolling: Functional Tests.....	37
Table 15 - Mobile Tolling: Technical Tests	37
Table 16 - Mobile Tolling: Back-Office Tests.....	38
Table 17 - Mobile Tolling: All Results Statistics	39
Table 18 - Mobile Tolling: Useful Data Statistics.....	40

Glossary

ADM	Accenture Delivery Methodology
ALPR	Automatic License Plate Recognition
CE	Collection Entity
ER	Entity Relationship
GPS	Global Positioning System
GUI	Graphical User Interface
IALPR	Intelligent Automatic License Plate Recognition
LPN	License Plate Number
MT	Mobile Tolling
OBO	Operational Back-Office
OBV	On-Board Unit
OCR	Optical Character Recognition
RFID	Radio Frequency Identification
RSE	Road-Side Equipment
TCOE	Tolling Center of Excellence
TP	Tolling Point
TVC	Tariff-based Vehicle Classification
TTL	Time To Live
UTC	Coordinated Universal Time
VTC	Video Toll Collection
VTP	Virtual Tolling Point
WS	Web Services

1. Introduction

1.1. About the Project Context

Accenture [1] is a global management consulting, technology services and outsourcing company, with approximately 281,000 people serving clients in more than 120 countries. Accenture helps organizations assess in maximizing their performance and work with them to achieve their vision, develops and implements technology to improve its clients productivity and efficiency, ultimately, Accenture enables its clients to become high-performance businesses and governments.

Accenture divides its Services Structure in three areas:

- Management Consulting;
- Technology;
- Business Process Outsourcing.

Which operate across five groups:

- Communications, Media & Technology;
- Financial Services;
- Health & Public Service;
- Products;
- Resources.

This internship is placed in the Technology area, working in the Products group. The project is integrated in Accenture's Tolling Center of Excellence (TCoE) which provided every necessary resources and conditions for the success of the work.

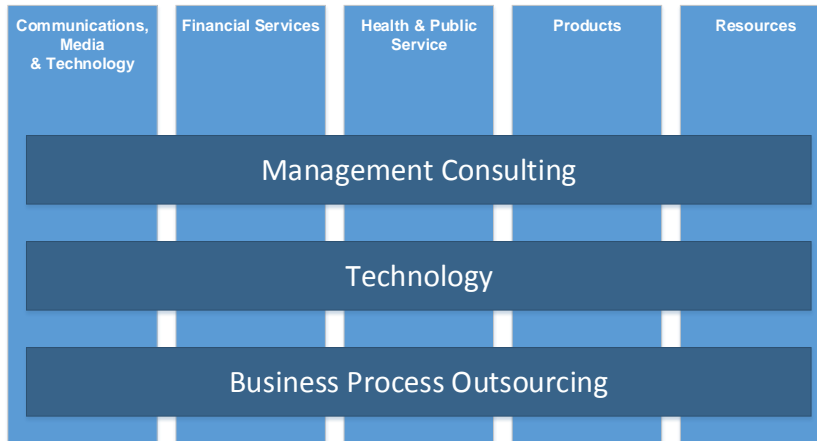


Figure 1 - Accenture Structure of Services

1.2. Objectives

From the internship point of view, the main goal is the experience in Software Engineering gained through working in a real corporate environment, facing real projects and real clients. This allows the intern to participate in all project stages and continuously improve to fulfil the project and client expectations while following rigid metrics implemented worldwide by the Accenture Delivery Methodology (ADM).

As from the project point of view, the goal is to improve the confidence level with which a customer that does not have an OBU, is automatically identified by Accenture's video toll collection (VTC) system when crossing a toll point, in order to reduce operational costs for the toll operator and the user.

This can be divided in two development goals:

- Mobile Tolling Back-Office Correlation Module
- Mobile Tolling Mobile Applications

1.2.1. Mobile Tolling Back-Office Correlation Module

This module is responsible for matching the events from the mobile application and from the automatic video toll collection engine and decide if the confidence level can be improved with this match, or not.

1.2.2. Mobile Tolling Mobile Applications

This application works by collecting information from the user mobile phone while he is traveling through a tolled highway which is identified by pre-configured virtual toll points, and sending the event to the back-office for correlation with the real event.

This report describes the Mobile Tolling system (Back-Office) and applications (Android and iOS), how it was implemented and how it integrates with the remaining video tolling infrastructure.

1.3. Scope

Nowadays, as a mean of recovering the investment, several road operators resort to road tolling. This means that the users that wish to use these roads have to use one of three available methods to pay their tolls [2].

The first and most commonly used method required the user to stop at a toll plaza at the entrance of the road, get a ticket that identifies the entrance point, and stop at a toll plaza in the exit of the road to pay the operator.

The second method, created to cut the costs in operator salaries, changed toll booths to automatic machines that accepted several paying methods.

The main issues with these two methods were the time the user lost in stopping at the toll point and the massive operational costs that the either the operator or the machine maintenance required. This led to the creation of a third method, the electronic toll collection, which places a barrier free toll point that identifies users by automatic license plate video recognition or by RFID transponders placed in the cars frequently called on-board-units (OBU).

Even with these three options available, the need for improvement still remained. The OBUs required the user to buy the hardware and also needed maintenance, and the video identification was not flawless, the chance of identifying the wrong vehicle or not identifying at all were still 15% if using Optical Character Recognition (OCR) and 4% if also using Fingerprinting [3]. These percentages, although small, translate to very high costs to the collection entities (CE) when dealing with millions of transactions per day [21].

With Mobile Tolling both these problems are addressed, by installing the application in the users smartphones to identify the toll crossing using geolocation data, and by matching that information with the one collected from the VTC equipment, the costs with OBUs are dropped and the identification success percentage significantly increases (to close, or equal, to 100%). This project will implement this application, the back-office mobile tolling correlation module and test the improvements these bring to the IALPR solution already implemented by Accenture.

2. State of the art

Free Flow Toll Collection Systems are extensively spread throughout the world and identify the road users either by detecting an RFID OBU, or by automatic license plate recognition using video analysis. The most popular and most accurate system (100%) is the OBU identification, but this requires the user to purchase an OBU and carry it in the vehicle. For those that do not have an OBU, the only solution available in barrier free roads is automatic video license plate identification.

The typical Video Toll Collection (VTC) system uses an OCR engine to detect the vehicle license plate and achieves an accuracy of more than 85%, but these 15% of undetected or wrongly detected license plates, when analysing millions of transactions per day, end up in high extra costs for the operator.

Accenture developed a new solution to improve its VTC system to an accuracy level of more than 96%, using image Fingerprinting [3], which is very close to the RFID method, but intends to improve further this accuracy to reduce additional operational costs.

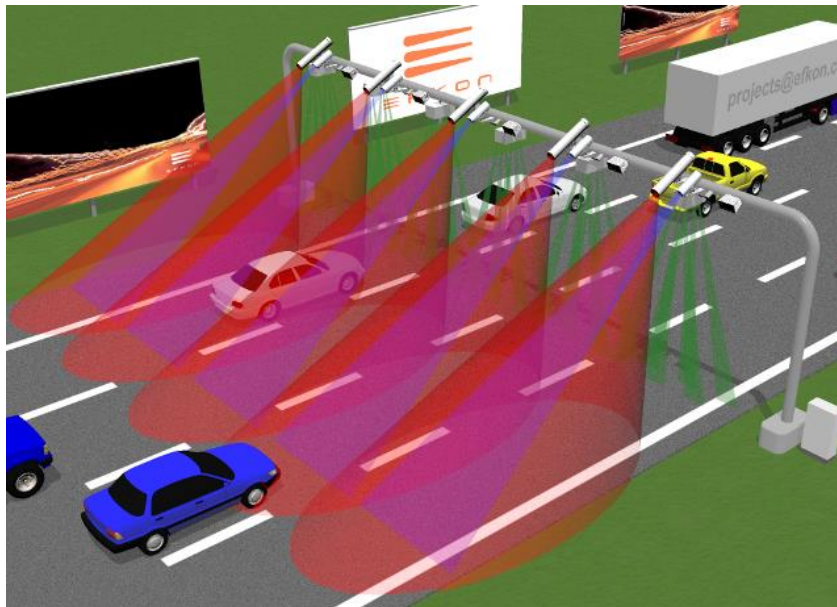


Figure 2 - Free Flow Electronic Video Toll Collection (Source [7])

The idea is to create a method that does not use OBUs, to remove the maintenance and operational costs associated with them, but still achieve the best possible accuracy when detecting the correct account and license plate.

2.1. Tolling Solutions

The electronic tolling market, at the moment, is still very focused in tolling methods that require the use of OBUs with RFID capabilities because these are the most accurate, hence the most reliable. This reliability comes at a cost, a cost to the operator that needs the infrastructure to identify the OBUs and a cost to the user that needs to buy an OBU. Because of this, many customers do not want to use this method and rely on manual toll points or automatic video identification, which brings more costs to the operator in order to maintain all these tolling options.

In this section a few applications are analysed that address the Mobile Tolling theme, attempting to, once again, change the way toll collection works, in order to simplify the customer's experience.

2.1.1. GeoToll

GeoToll [9] is an application that defines itself as an OBU in a different form factor, which uses an RFID tag and an application in the user's smartphone.



It uses geolocalization to identify when the user crosses a toll point, but to improve accuracy, it uses an RFID tag attached to the mobile phone which is activated when approaching a toll point to identify the user in the same manner a OBU does.

This application offers the following key features:

- Allow user to create account and link multiple vehicles
- Allow user to link rental cars
- Automatic toll charging from connected account
- Provides security by associating the RFID with the smartphone
- Create expenses reports
- View and print transactions

This application has the following downsides:

- Requires RFID tag to work
- Smartphone requires NFC to communicate with the tag
- Tag does not work if device fails to activate it before toll point
- Only available for Android
- App is not free

2.1.2. BancPass PToll



PToll [10][11][12] is an application by BancPass that allows its users to automatically pay tolls by providing and photographing the vehicle license plate so it can be matched with the VTC at the toll plaza, which requires toll operators to partner with them to access their LPN database. In spite of this it could be considered more of a mobile payment app rather than a tolling solution.

PToll does not use GPS or RFID tags, it relies only in the video identification and matching with the user provided license plate to correctly identify the account to bill.

This application offers the following key features:

- Works with multiple vehicles
- Convenient automated billing
- Choose payment method
- Approve payments interactively
- Choose pre- or post-transaction approvals

- No need for GPS or RFID tags

This application has the following downsides:

- Does not help identify the vehicle
- Only used as payment solution
- Only available in Austin, Texas

2.1.3. The Toll Roads

The Toll Roads [13] [14] is not an electronic tolling application, but a management app for users of the Fastrack tolling system, which uses RFID transponders to identify vehicles. This application works as an easier access point for the features available for account and payment management in the online website.



This application offers the following key features:

- Register new accounts
- Check account balance
- Make a payment
- Add or delete a vehicle (works with rentals)
- Manage transponders
- Review recent activity
- Toll Points map

This application has the following downsides:

- Only available for Fastrack users
- Does not help identify the vehicle
- Only manages the features also available in their website

2.1.4. PayUrToll



PayUrToll [23][24] is an application that aims to make toll payments easier. It seeks to decrease traffic congestions by speeding the payment process at toll plazas. According to the submitted patent, this app relies on a transponder device attached to the smartphone via the headphones jack. This transponder facilitates the communication between the toll plaza equipment and the application using sound.

This application offers the following key features:

- Track location with geographic coordinates
- Calculate tolls due, based on time and distance on toll roads
- Track in-lane offering (as states switch to pay-per-lane model)
- Save money on infrastructure costs
- Offer convenient signup and payment options through smartphone or website
- Integrate non-GPS technology options too

This application has the following downsides:

- Requires additional device to communicate with tags/toll plaza
- The “no smartphone” option relies only on VTC identification

2.2. Comparison

For this evaluation, the following metrics will be analysed:

- **Multiple Operators**
Can the application be used with different toll operators?¹
- **Toll Point detection**
Does the application detect when the vehicle crosses a toll point?
- **Detection Method**
What is the detection method used by the application?
- **Requires interaction**
Does the application run in background or needs interaction from the user to function properly?
- **Additional hardware**
Does the application need additional hardware to function properly?
- **Account Management**
Can the user view his account details and change his account data from the application?
- **Multiple Vehicles**
Can the same application be used with multiple vehicles?
- **Transactions History**
Can the user check the list of previous transactions in the application?
- **Transactions Details**
Can the user check the details of a transaction in the application?
- **Notifications**
Does the application notify the user when a toll point is approaching or when a crossing is charged?
- **Payment Management**
Can the user change his payment details in the application?
- **Automatic Payment**
Can the user provide a payment method so the transactions are automatically paid by the application when travelling?
- **Reports**
Can the user check and create expenses reports
- **Toll Points Map**
Can the user view a map with all the toll points?
- **Back-Office system**
Does the solution provide a back-office?

¹ In USA, a user that frequently crosses many states, typically needs one OBU for each state.

Metrics	GeoToll	PToll	The Toll Roads	PayUrToll	Accenture Mobile Tolling
Multiple Operators	✓	✓	✗	✓	✓
Toll Point Detection	✓	✗	✗	✓	✓
Detection Method	RFID Tag	VTC	– ²	RFID Tag Sound Device	Geolocation
Autonomous	✓	✗ ³	✗	✓	✓
No Additional Hardware	✗	✓	✗	✗	✓
Account Management	✓	✗	✓	✓	✓
Multiple Vehicles	✓	✓	✓	✓	✓
Transactions History	✓	✗	✓	✓	✓
Transactions Details	✓	✗	?	✓	✓
Notifications	✓ ⁴	?	?	✓	✓
Payment Management	✗	✓	✓	✓	✓
Automatic Payment	✓	✓	✗	✓	✓
Reports	✓	?	✗	✗	✗
Toll Points Map	✗	✓	✗	? ⁵	✗ ⁶
Back-Office	✓	?	?	✓	✓

Table 1 - Mobile Tolling: Tolling Solutions Comparison

² Not Applicable

³ Interaction is required if “requires payment authorization” is configured

⁴ Email notification also available

⁵ Provides a “My Location” tab

⁶ Does not have a map with all toll points but in transaction details the map of a particular toll point is available

Accenture Mobile Tolling Solution uses mobile phone data to track the position of the users and detect when they pass the toll points using geolocation services and pre-configured virtual toll points. This way an event can be sent containing the toll point information, and matched with the VTC event in the back-office boosting the accuracy level to 100% confidence.

There is no solution that covers all the features found, but looking at the main objective, approach accuracy levels achieved without an OBU to those obtained when using OBUs, Accenture Mobile Tolling is the only solution to that achieves it, without using any additional devices or tags, and further improves the experience by providing several features also present in the other apps, and addressing some of the downsides found on the other solutions.

3. Methodology

3.1. Accenture Software Development Methodology

In the course of this project the adopted development methodology was an extension of the traditional Waterfall methodology, the V-Model Methodology [4]. This procedure bends the waterfall process line upwards after the coding stage to better demonstrate the relationships between each stage of the development process and its associated stage of testing (see Figure 3) thus achieving the “V” shape.

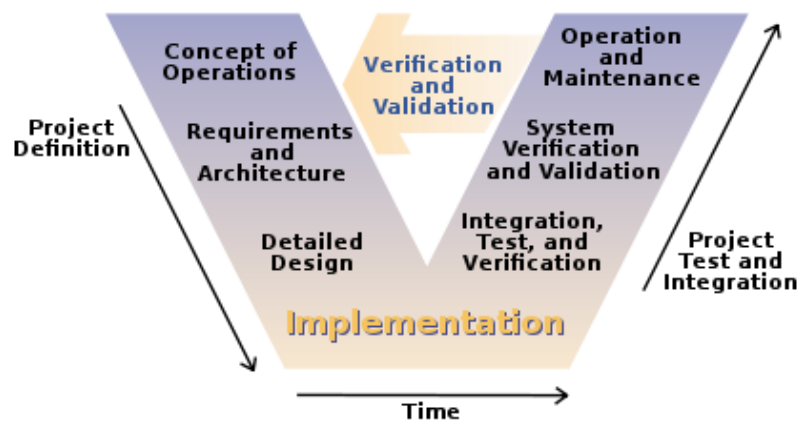


Figure 3 - V-Model Software Development Methodology (Source [4])

This methodology has the following phases:

- **Requirements Analysis**

The requirements of the project are gathered by analysing the needs of the user, and carefully defined and verified by the client so they can be used as the foundation for the system design phase. This phase explains the functionality as seen by the user, rather than determining how the software will be built.

At this time is also produced the user acceptance tests based on the requirements gathered.

- **System Design**

In this phase, system engineers analyse the requirements document to produce a software specification document where they identify how the requirements are going to be implemented and all the technical details required to achieve the proposed system. If some requirement is not feasible, the user is informed, a compromise is reached and the requirements document updated. System testing specifications are also defined at this stage.

- **Architecture Design**

At this stage is defined the high level design of the system, all the modules that make it up, their functionality, the relationships and dependencies, the necessary database tables, the architecture diagrams, and all other required technological details and produced the architecture design document.

In addition, for the later testing stages, the integration design document is also defined in this phase.

- **Module Design**

After defining the high level design of the architecture, every piece must be fragmented and detailed further at a lower level, every detail, possible errors, expected inputs and outputs, must be explained, so that the developer may follow it as a recipe while coding.

With this level of detail, the unit test guidelines and tests are also defined in the low level design phase.

- **Coding**

The implementation phase is where the developers accomplish what was defined in the previous stages, where the software is built according to the specifications.

- **Unit Testing**

In the unit testing phase, the internal logic of each individual part of the software, each function, is tested with multiple input variations to assert every possible case of execution and determine if it is fit for use. These tests are created by the programmers and can be assisted by static analysis tools to facilitate the process.

- **Integration Testing**

In this phase the separate models are tested together to verify possible errors in the interfaces and interaction between them. This usually does not require deep knowledge of the code since it is not directly checked for errors.

- **System Testing**

After integration tests are completed, the system tests compare the specified system against what was really made. It checks if the system complies with the specifications.

- **Acceptance Testing**

This is the final stage, used by the client to decide whether or not the system is ready to be accepted and launched. It verifies if all the requirements are satisfied by the finished product.

It promotes meticulous design, development, and documentation necessary to build stable software products, placing crucial importance in the project definition phase, as the building block for coding and validation phases. These building blocks are continuously being tested since the tests also start at the beginning of the project, allowing the developers to add or fix requirements as the testing is done.

During this time the intern was able to experience this methodology by participating in several meetings with developers and managers, and produce relevant documents for the definition of the project.

3.2. Activity Plan according to the methodology

The activity plan for this internship is described in the Gantt chart below in Figure 4.

It was compiled by Accenture's internship advisor, along with the project manager and prioritized according to the project needs while maintaining the integrity of the internship.

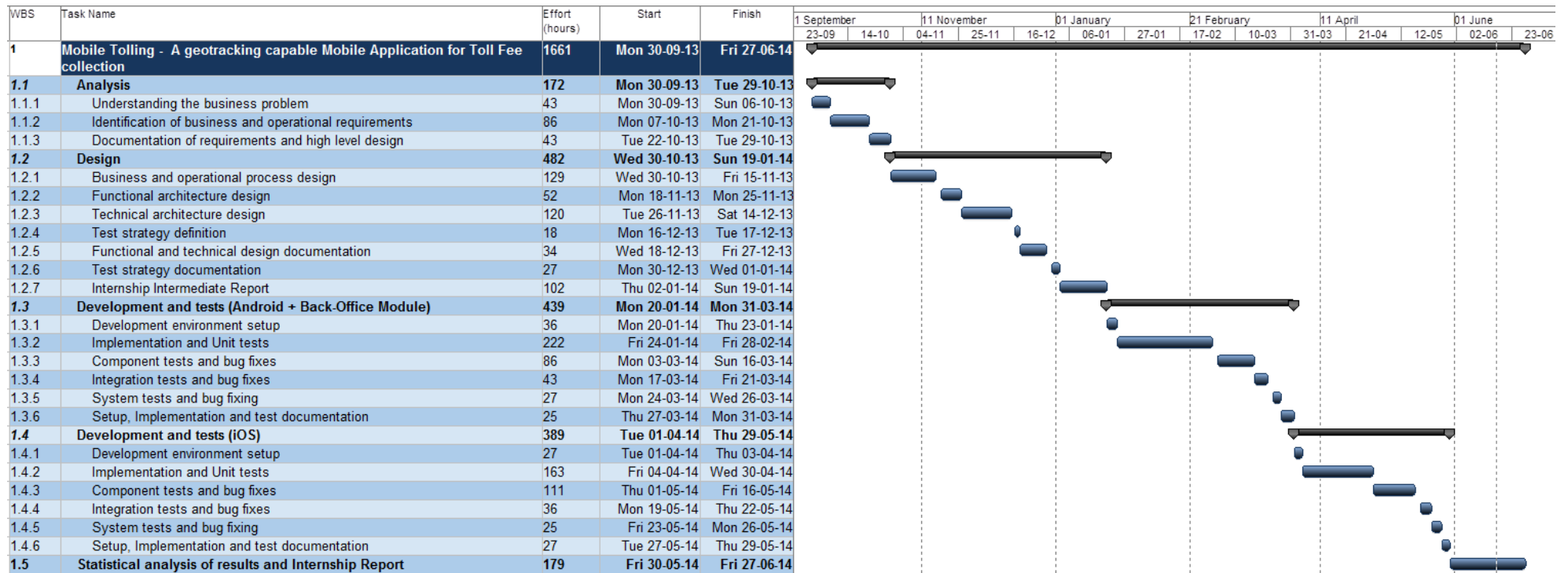


Figure 4 - Mobile Tolling: Internship Plan

The objectives are solid, fulfil project requirements and achieve acceptable results with the purpose of improving the confidence level of video tolling in order to decrease operational costs to the operator, raising the value of Accenture's tolling solution.

The requirements (described in the next chapter) were defined following the TCoE business rules and the assessed necessities of clients currently using Accenture's Tolling solution. These are stable, validated by the project manager and move the business solution towards the path it needs to follow to surpass the proposed objectives.

To accomplish this, and as described above, the proposed methodology was chosen because it allows the testing phase to be performed at the same time as the planning and building phases, which in practice saves in possible problems that are immediately detected, instead of only appearing in later stages. By verifying and validating at each stage, it improves risk management and quality control of the final product. This methodology is recommended internationally by Accenture and is already implemented and verified by the hosting project.

As the rest of TCoEs projects, this project will be inserted in the continuous code analysis framework SonarQube [22] which examines the project code searching for several common mistakes:

- Bugs and Potential Bugs
- Coding Standards Breach
- Duplications
- Lack of Unit Tests
- Bad Distribution of Complexity
- Spaghetti Design
- Not Enough or Too Many Comments

To improve further, it also uses the Findbugs tool, a program that uses static code analysis to find bugs in java code. This combination allows us to keep track of the quality of the software being developed from an early stage so it can be improved as soon as possible.

Along with the software quality, some performance metrics must be achieved to consider this project a success:

- Toll point crossing must be detected
- Crossing event generated within ten seconds⁷ of crossing the toll point
- Battery consumption must be very low, not to bother the user in his daily activities

Also, in order for the entire team to be aware off the work being done by each member at all times, to keep a record of time spent and facilitate the management of the project, the Redmine [25] platform is being used. This allows recording and managing the features being implemented and all issues found, as well as who is working in each one and automatically build the project Gant diagram to have a global vision of how long each action will take and when the project will finish.

⁷ Ten seconds is an adequate time to tolerate acceptable network latency and still allow the back-office to correlate the two events, given the short size of the message sent in the event.

3.3. Risk Assessment

In this section we discuss the risks identified for this project, as well as the mitigation plan for each of them.

Mobile application malfunction

This covers several aspects that might occur such as the operating system kills the application or the application is unable to work (detect a crossing) due to excessive operative system load. These risks result in the event not being sent without the influence of the user. In these cases the back-office requests the logs from the application to ensure that the user did not have input in the malfunction, and apply the discount anyway, if the user had no fault in the error.

Event not published in time

The application requires a network connection to send the events to the back-office, but we cannot force the user to keep the connection on all the time. If the application does not have access to a network connection because the user disconnected it, and the event does not arrive in time for the correlation to be successful, the discount will not be applied. Since the pending events are sent to the back-office when the user has a connection, if the event is detected as being late, it will be analysed by an agent in manual processing.

Crossing not detected

If a user is on a trip crossing several toll points, it is possible that the application, for some reason, misses the detection of a toll point and correlation is not done. This causes one of the events to go to manual verification. This is mitigated by a system of routes. The correlation module uses a module called trip builder that verifies the previous and next crossings as well as the missed toll point images and events, concluding that the user did or did not pass the missed toll point and apply the discount anyway if it is asserted that he did.

4. Requirements Specification

Requirements definition is of major importance to the success of a software project, because they give a good notion of what needs to be done and how long it might take to do it, they are the basis during the entire development stage.

In this section are described the features and corresponding requirements essential to the Mobile Tolling solution:

- Authentication
- Account Management
- Location Management
- Event Management
- Notifications
- Bluetooth Management
- Back-Office

In Appendix A can be seen a more detailed explanation of the requirements by defining and describing user scenarios for each feature.

Each requirement has been given a priority level, so that there is an order of importance between them. This means that, in order to be able to demo the product to possible clients and to gather test results, the requirements with higher priority (P1) need to be implemented first.

4.1. Authentication

Authentication requirements enable users to access their accounts in the application.

Id	Functional Requirement Name	Functional Requirement Description	Priority
HLR_ATH_01	User can Login into his account	The user can login using his active account credentials	P2
HLR_ATH_02	User can Logout of his account	The user can end his account session in one specific device	P2

Table 2 - Mobile Tolling: Authentication Requirements

4.2. Account Management

Account management requirements define the creation of an account to use the application, the account information that can be accessed in the application and the update procedure for this data.

Id	Functional Requirement Name	Functional Requirement Description	Priority
HLR_AM_01	User can create new Account	The user can create a new account in the System	P2

HLR_AM_02	User can update his account	The user can change some information relative to his account using the Mobile Application	P2
HLR_AM_03	User can view his account information	The user can check the information related to his account using the Mobile Application.	P2

Table 3 - Mobile Tolling: Account Management Requirements

4.3. Location Management

Location management requirements describe what is needed for the application to use geolocation services and detect when the user crosses toll points.

Id	Functional Requirement Name	Functional Requirement Description	Priority
HLR_LM_01	Mobile Application manages the GPS service	The application is able to decide when to start and when to terminate the GPS tracking for battery management	P1
HLR_LM_02	Mobile Application can manage the topology	The Application will hold the current topology configuration and will be able to receive configuration updates from the BO	P1
HLR_LM_03	Mobile Application can detect VTP crossings	The application is able to detect when it crossed a gantry, route and VTP	P1
HLR_LM_04	User can stop and start the tracking service	User can stop and start the tracking service without having to logout of the current account	P1

Table 4 - Mobile Tolling: Location Management Requirements

4.4. Event Management

Event management requirements define the features necessary to generate, publish and consult toll point crossing events.

Id	Functional Requirement Name	Functional Requirement Description	Priority
HLR_EM_01	Mobile Application is able to generate crossing events	The application will be able to locally create events every time a VTP is crossed.	P1
HLR_EM_02	Mobile Application is able to publish the generated crossing events	The application will try to publish an event right after its creation. If the publishing fails, then the app will try to publish all pending events when it detects network connection.	P1
HLR_EM_03	Event information and status will be updated into the application	The application is able to receive updates for all the published events	P1
HLR_EM_04	The user can view the detail for the last events	The application will be able to display the information related with the last events tracked by the smartphone	P1

Table 5 - Mobile Tolling: Event Management Requirements

4.5. Notifications

These are the requirements necessary for the notifications feature.

Id	Functional Requirement Name	Functional Requirement Description	Priority
HLR_NT_01	The application generates local notifications when necessary	The application will generate and display visual and sound notifications from the BO or generated in the Mobile App.	P3

Table 6 - Mobile Tolling: Notifications Requirements

4.6. Bluetooth Management

These are the requirements necessary for the Bluetooth management feature.

Id	Functional Requirement Name	Functional Requirement Description	Priority
HLR_BT_01	An account can be associated with one or more Bluetooth devices	When creating a new account in the application, the user can specify a Bluetooth device to use	P5
HLR_BT_02	Application uses Bluetooth information	if the account has configured Bluetooth devices, the application will start the GPS tracking only when one of them is connected	P5
HLR_BT_03	User can manage Bluetooth devices	The user can add, remove or edit Bluetooth devices related with his account	P5

Table 7 - Mobile Tolling: Bluetooth Management Requirements

4.7. Back-Office

These are the requirements for the back-office correlation module to manage the event matching and communication with the application.

Id	Functional Requirement Name	Functional Requirement Description	Priority
HLR_BO_01	The BO will be able to send remote notifications to a device	The BO is able to create and send remote message notifications to every device running the Service	P2
HLR_BO_02	The BO performs an automatic validation to the account	Every time the user creates a new account it will be sent to the BO and automatic validated	P2
HLR_BO_03	The BO user can search accounts	The BO operator can search for accounts registered in the system, being able to see all the details related with that account	P4
HLR_BO_04	The BO user can view the events related with one account	The BO user can view the events related with one account	P4

HLR_BO_05	The BO can correlate RSE events with App events	The BO is able to create a relation between a RSE event and an event detected by the app.	P1
HLR_BO_06	RSE events correlated with App events are not sent to IR	After Events correlation, the events are not sent to IR because the vehicle information is automatically detected	P1
HLR_BO_07	Correlated event is rated with a different price	The system is able to rate a correlated event with a different configured price	P3
HLR_BO_08	BO correlates delayed application events	If an application event is published in the BO as delayed, the system will correlate that event with the existing RSE event, even if the last is already published	P3
HLR_BO_09	Delayed application events have no discount advantage	If an application event is published in the BO as delayed, the system will not give any type of monetary discount to that event	P3
HLR_BO_10	BO correlates delayed RSE events	If an RSE event is published in the BO as delayed, the system will correlate that event with the existing application event	P3
HLR_BO_11	Delayed RSE events have discount advantage	If an RSE event is published in the BO as delayed, the system will give the full monetary discount to that event	P3
HLR_BO_12	The BO waits for the correlation for a configured period of time	The BO will wait for application and RSE events only for a configured period of time. If there's no correlation during this time, the RSE event is released and the application event is discarded.	P2
HLR_BO_13	BO inserts Mobile Tolling accounts in the database.	When the device sends a request to register account, the BO will insert the user data into tables MT_Vehicle_Info, MT_Personal_Info and MT_Account	P2
HLR_BO_14	BO inserts Mobile Tolling events in the database.	When the device sends a request to register an event the BO will insert the data in the tables MT_Event, MT_Event_Evidence and OD_Event	P1
HLR_BO_15	BO creates Mobile Tolling messages to be sent to the device	The Mobile Tolling messages are sent by the BO to the device in response to what it requests or are sent by the BO to notify the device that it changed something. There are 3 types of response messages (ACCOUNT_OK, ACCOUNT_ERROR and EVENT_UPDATE) and 2 types of notifications from the BO (ACCOUNT_UPDATE and ACCOUNT_REMOVE)	P2

Table 8 - Mobile Tolling: Back-Office Requirements

5. Architecture

The present document is divided into three sub-sections:

- **Project Integration**
Provides an overview of the integration of the mobile application in the tolling infrastructure
- **Mobile Tolling Application Overview**
Describes the details of the application architecture and implementation
- **Data Model Overview**
Describes the data module of the project

The tolling system is composed, as can be seen in Figure 5, by the operational area and the billing area. This internship is centred in the operational area, specifically in the mobile applications and the operational back-office.

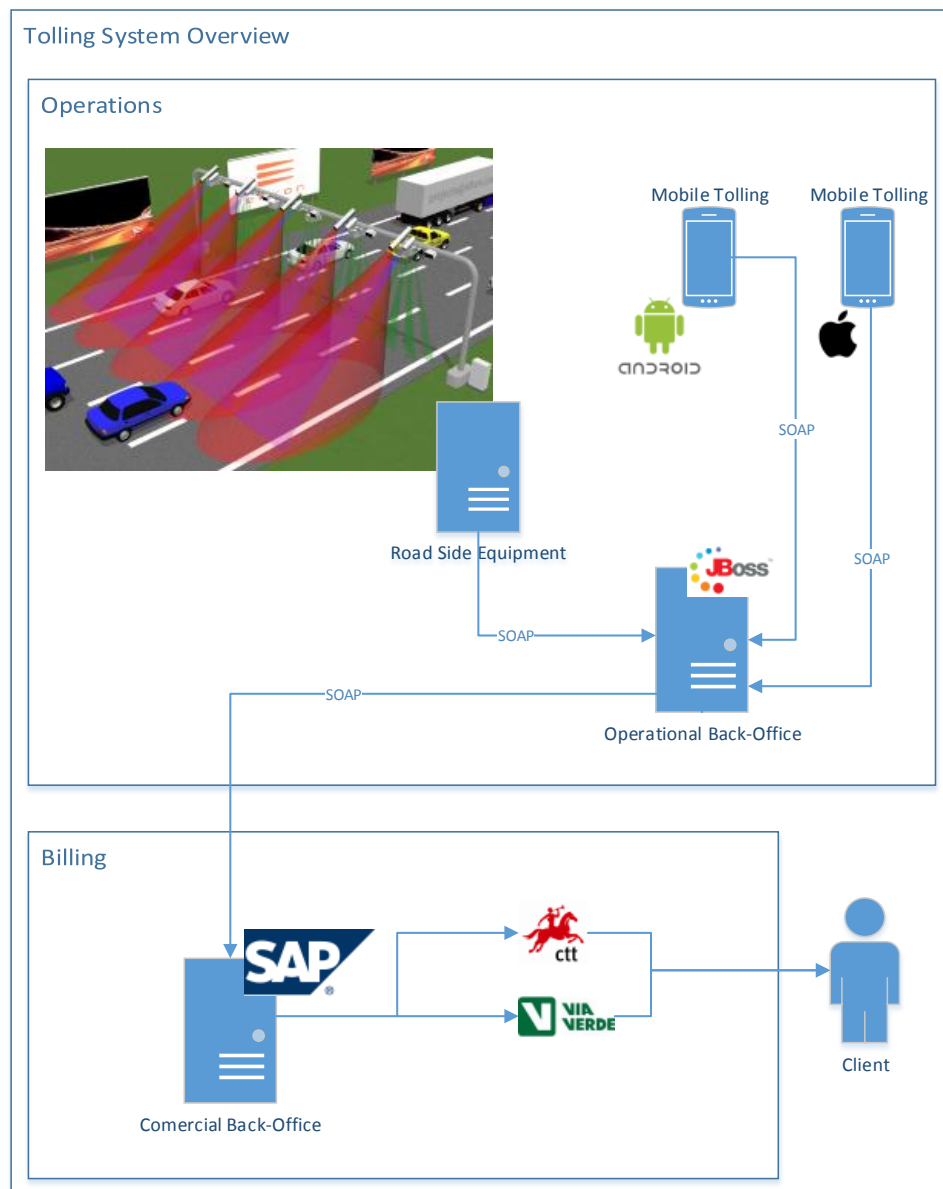


Figure 5 - Mobile Tolling: System Overview

5.1. Project Integration

The Mobile Tolling (MT) event correlation module is part of the Operational Back-Office (OBO) system and aims to improve the automatic vehicle recognition rate obtained in Video Toll Collection (VTC) by matching the events generated by the VTC with the events generated by the mobile application when the user normally passes through the toll points while using the application.

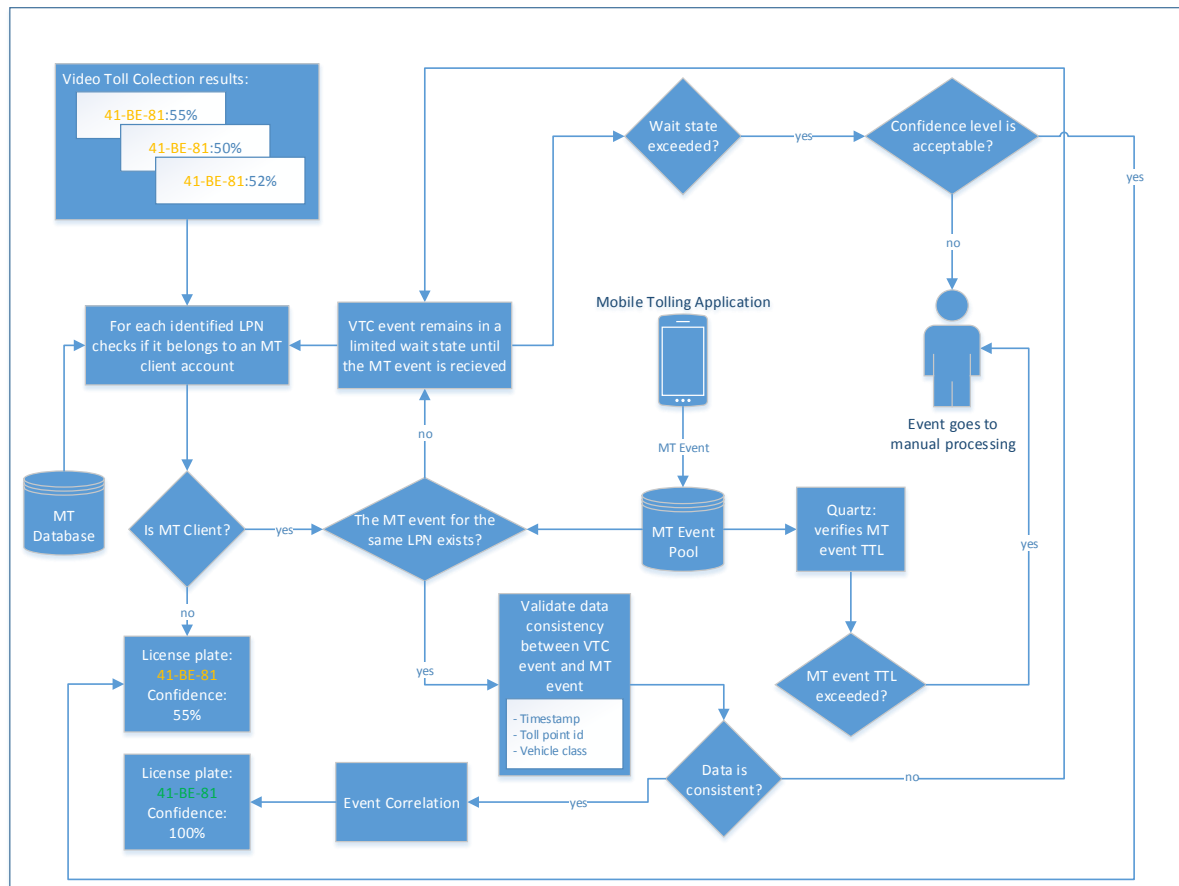


Figure 6 - Mobile Tolling Back-Office: Correlation Module Process Flow

In the normal process, when a user passes a toll point, a photo is taken and analysed by the automatic video recognition engine and a VTC event is generated.

Upon arrival of a VTC event, the BO checks if the identified Licence Plate Numbers (LPN) are MT clients, if not the VTC event does not benefit from this process and is accepted as the best result possible and either classified as final result or sent to manual validation, depending on the confidence level with which it was identified.

This is where the Mobile Tolling project intervenes. As shown in Figure 6, when the user passes a toll point, the mobile application generates an event and sends to the Back-Office (BO).

When an this event arrives in the BO it is added to an events pool configured for MT Events while waiting for the VTC events to correlate with.

So, enhancing the normal procedure with the MT event, if the LPN corresponds to an MT client, the MT Event pool is checked for events with the same LPNs. If no matching event is found, the VTC event waits a configured time for this event and after that time, if the MT event still did not arrive, the VTC event is either classified as final result or sent to manual validation, depending on the confidence level with which it was identified.

If the MT event does arrive before the configured time or is already available, the two events are sent to a rule engine (Drools [26]) for data validation. The timestamp, toll point id and vehicle class are verified, and if the events data match within a certain deviation margin, it is safe to say that the two events correspond to the same vehicle passage which raises the confidence level to 100% and finishes the processes with an improved confidence level. If the data validation fails, the events return to the respective queues and wait the configured time.

When an MT event reaches its time-to-live (TTL) and no VTC event has arrived that could be correctly correlated with it, the event is discarded as no longer usable for correlation but not deleted, instead, is sent for manual validation.

5.2. Mobile Tolling Application Overview

The Mobile Tolling application meets the following requirements: it allows the user to check his contractual information and update it, check the list of transactions and notifications and keep a better track of his tolling expenses. In the background, it tracks the user position and when he crosses a toll point, generates an event and sends it to the back-office.

In Figure 7 are shown the modules responsible for this behaviour. The Graphical User Interface is always present as long as the application is in foreground and is used to display the information to the user and to interact with him.

In order to track the user position so the crossing of toll points can be detected and an event generated, the application Virtual Gantry Recognition Engine uses the mobile phone location providers (GPS + Sensors) and a set of virtual gantries (Topology) to detect when the user is approaching a toll point, and the moment he crosses it.

This application should have a low memory footprint and battery usage and must be able to function without internet connection even if with limited functionality. Although, to benefit from the use of the application, the events require internet connection to be sent to Back-Office, within the configured time limit.

In order not to be a bother to the user, the application is intended to function properly with minimum interaction from the user, detecting and registering automatically when the user crosses a toll point.

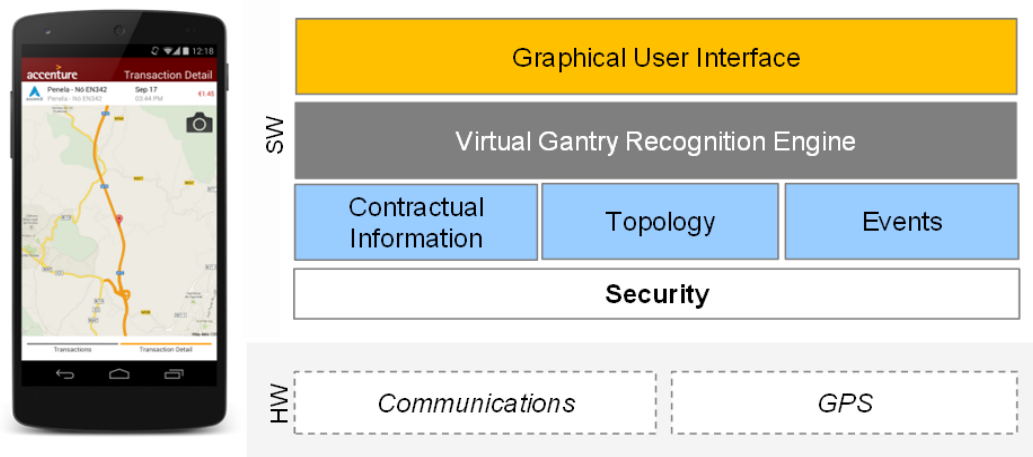


Figure 7 - Mobile Tolling: Application Modules Overview

5.2.1. Model-View-Controller

The application follows an adaptation of the MVC architectural pattern as represented in Figure 8 and Figure 9. This allows decoupling the Interface from the Business logic, keeping a modular and easily updatable and maintainable project. This adaptation removes the interaction between Model and View, leaving the Controller responsible for updating the view and model whenever necessary.

For instance, if the user were to check the list of transactions, the view requests the list of transactions to display from the controller which requests the data to the model and updates the view with the requested data. The controller also updates the model and notifies the view when a new event is received from the BO.

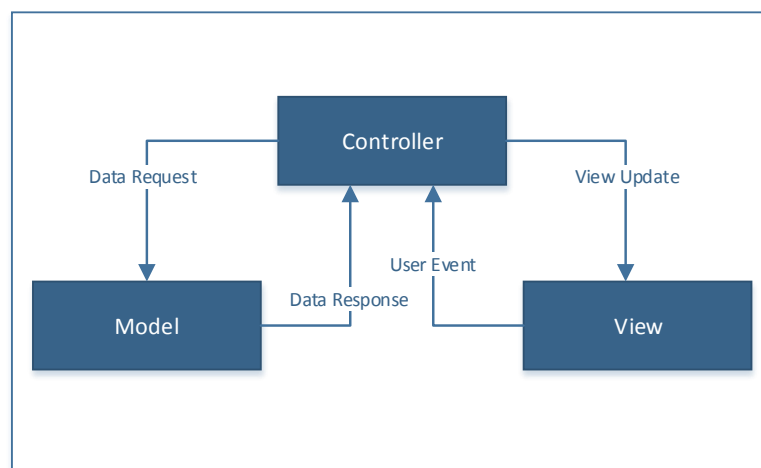


Figure 8 - Mobile Tolling: Model-View-Controller architecture adaptation

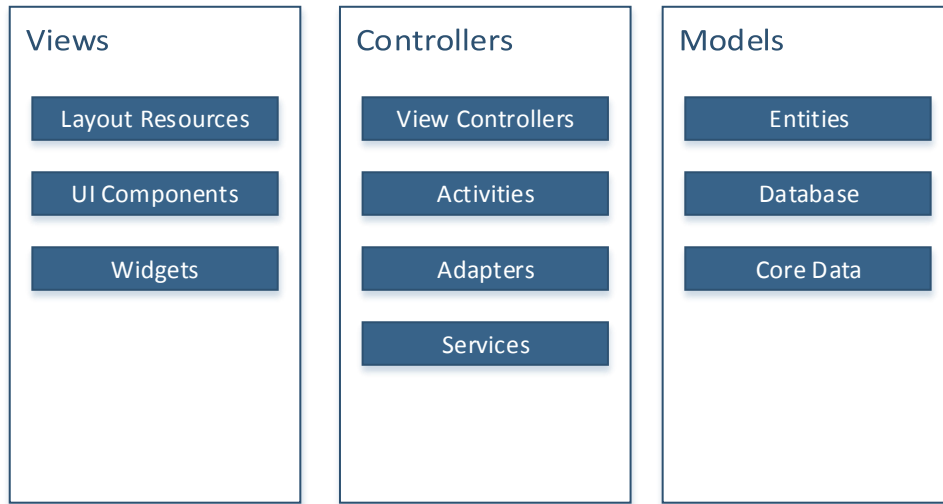


Figure 9 - Mobile Tolling: MVC Details

5.2.2. Business Logic

The main purpose of this application is to detect when a user crosses a toll point and communicate that event to the Back-Office. To accomplish this objective, in the background, the application is listening to location changes and analysing them to find if the route driven crosses any toll points.

The Location Service starts using a passive location listener [18][28] and when receives a location update evaluates the position against the virtual gantries topology to check if has entered a fence. If the location update is inside a fence, the location provider is changed to GPS, so that a more precise location is received in the following location updates. When a location update is detected outside a fence, the provider is reverted back to passive mode again.

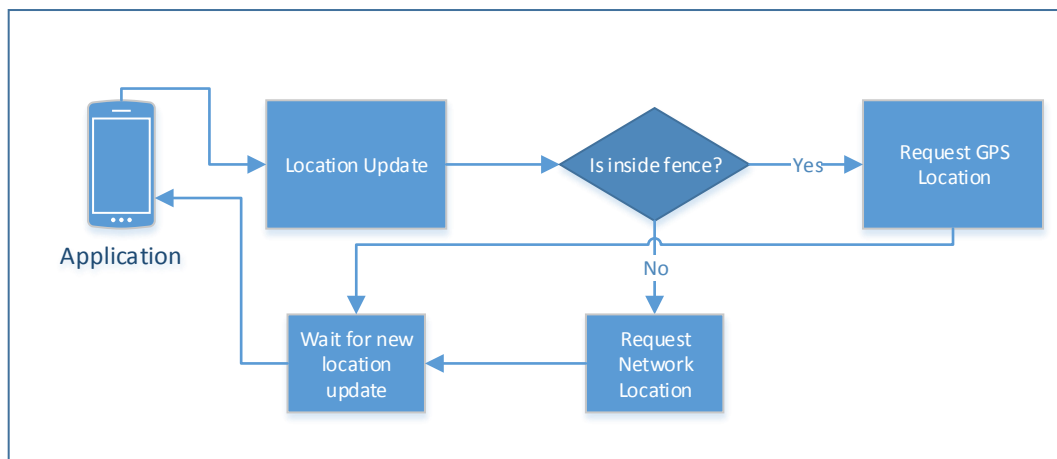


Figure 10 - Mobile Tolling: Location Provider change process

After entering the fence and changing to GPS, the location evaluation method will now try to detect when the user crosses a gantry.

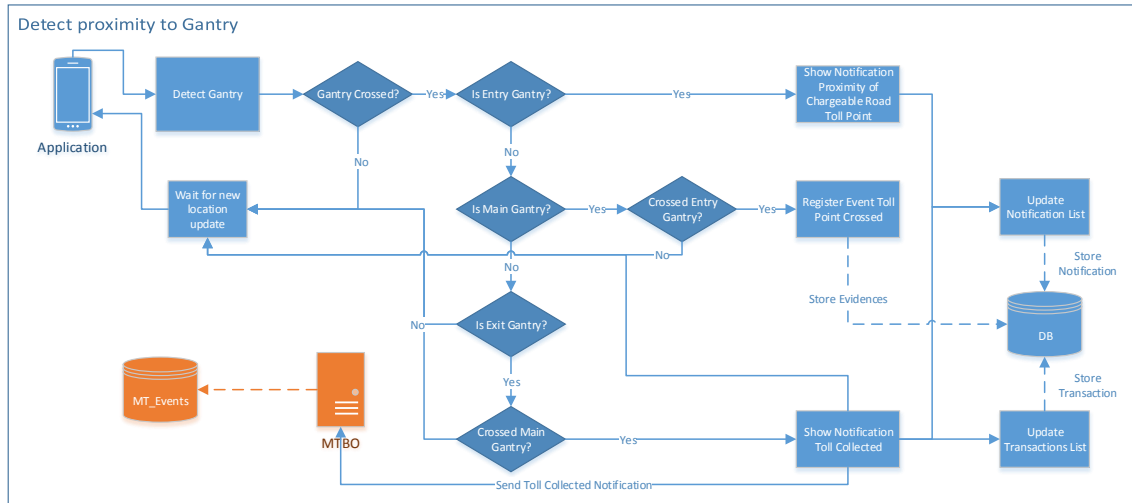


Figure 11 - Mobile Tolling: Gantry Crossing Detection Process

To detect the passage through a real toll point three [29] types of scenarios can be used:

- **Zones**
Simple polygon used to cover wide areas.
- **Corridors**
Rectangle segments placed over the road sectors to monitor.
- **Virtual Gantries**
More complex polygons composed of six points and seven line segments.

Zones are better for distance charging since they offer little information other than enter and exit zone. In this situation more information is required, such as the knowledge of direction of travel or direction change situations.

Corridors accuracy depend on how wide they are, this can bring problems since there might be an non tolled road next to the tolled road that cannot be mistakenly detected, so limits the size of the corridor, decreasing its accuracy. A major problem with corridors is that the detection is made at the entry so if that misses, the hole corridor is missed.

But, those that bring the most advantages are the virtual gantries [30], since with the high precision with which they are detected and the available segments, we can detect some error situations that could not be detected with zones or corridors. They also occupy several less memory than corridors.

In order to correctly detect that the user passed a toll point, a set of three virtual gantries are used:

- **Approaching Toll Point Virtual Gantry**
First gantry is used to notify the user of an approaching toll point so the user can avoid it if necessary
- **Toll Point Virtual Gantry**
Matches the position of the real Toll Point and records the passage time and coordinates. This is also used to notify the user.
- **Exiting Toll Point Virtual Gantry**
Last gantry is used to notify the user of the toll charge and to confirm the virtual route travelled.

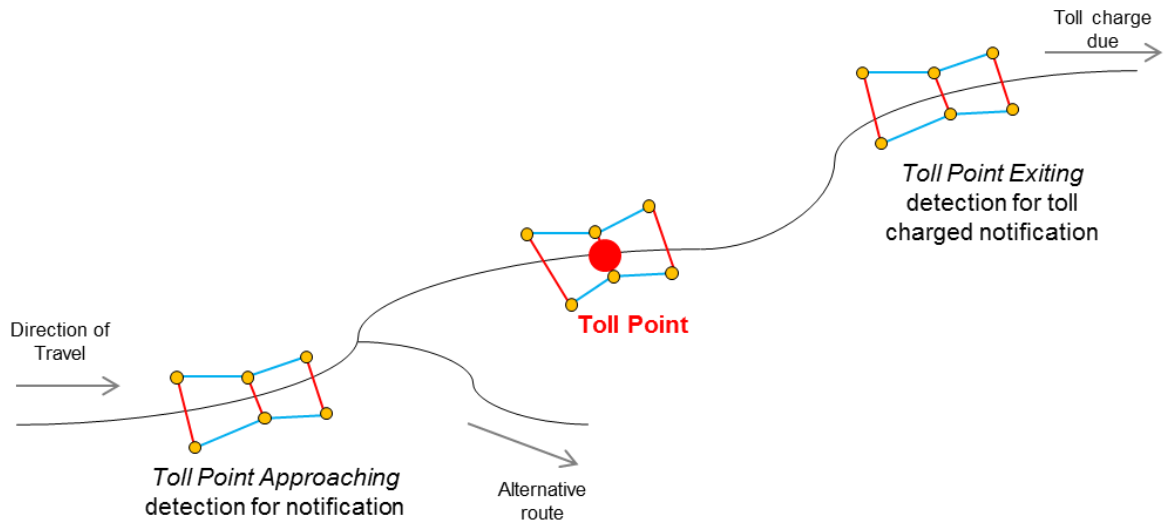


Figure 12 - Mobile Tolling: Gantry Map

When the Location Service detects that the user passed a toll point it passes the information to Manager Service which collects all the necessary information and proof to correctly identify the account and the toll point, and generates an event to send to the Back-Office, a notification to show the user and an update to the transactions table.

This event is passed to the Event Publisher Manager that tries to send the event and changes its status to “sent” or “pending”, depending on the result of the operation.

The Event Publisher Manager is also used to send pending events when a connectivity change is detected and the application becomes online.

Communication with the BO web services (WS) uses Protocol Buffers (Protobufs) [19] to easily serialize the data to send.

To receive messages from the Back-Office push notifications are used. These notifications contain messages that can have data updates or trigger actions in the application, such as a configuration update or even blocking an account from being used.

These and other business logic classes are described in Table 9 and can be seen in Figure 13.

Class	Description
LocationService	Service running in the background responsible for: <ul style="list-style-type: none"> • Receive Location Updates • Manage Location Providers • Evaluate Location Updates • Check if fence border is crossed • Check if gantry is crossed
ManagerService	Service running in the background responsible for: <ul style="list-style-type: none"> • Process fence in/out • Process gantry and route crossing

	<ul style="list-style-type: none"> • Generate Events • Generate Notifications
EventPublisherManager	Manager responsible for sending events to the Back-Office.
ConfigurationUpdater	Service responsible for updating configurations such as topology or business logic constants.
NotificationManager	Service responsible for showing notifications to the user.
RemoteNotificationsManager	Service responsible for receiving push notifications messages.
RemoteNotificationsParser	Utility class to parse the server messages.
NetworkStateReceiver	<p>Class notified by the system when a network change occurs.</p> <p>Responsible for launching the event publisher and remote notification register when the application is online after a network change.</p>
ServicesStarter	Service responsible by starting the Location Service and the Manager Service when the application starts.
WebClientAPI	API class with the methods used by all the other classes that communicate with the Back-Office.

Table 9 - Mobile Tolling: Business Logic Classes

5.2.3. Application Modules

In order to provide reusability and a clearer understanding and organization of the application code base, the implementation was organized in separate modules.

Module	Description
controllers	In this module are located all the controllers responsible by loading the resources that build the user interface (UI) and user experience (UX) of the application.
controllers.login	Module responsible for loading the UI presented to the user for the login process.
controllers.login.newaccount	Module responsible for the new account process.
controllers.main.home	In this module are the controllers responsible for loading the UI for all account related screens and for populating this UI.
controllers.main.transactions	In this module are the controllers responsible for loading the UI for all transactions related screens and for populating this UI.
controllers.main.notifications	In this module are the controllers responsible for loading the UI for all notifications related screens and for populating this UI.
common	This module encapsulates several classes which use is shared by several other modules throughout the application.
components	Here we can find the UI components that were extended from native ones to customize the UI as required for the UX desired for this application.

common.constants	Contains all the classes that define constant values used by the application.
common.utils	Provides several utility classes and methods used by all modules.
webapi	The webapi module is responsible for all the communication with the BO.
protos	The protos module contains the classes necessary for management and use of the Protobuf protocol.
businesslogic	This module contains the logic related to the detection of virtual toll point crossing.
businesslogic.tracking	This module contains the logic associated with the position and route tracking of each user.
businesslogic.topology	This module contains the logic that defines the virtual system used for tracking and detection of positions relative to the real toll points.
persistence	The persistence module is responsible for all the interaction with the SQLite database where the data is persisted by the application.
entities	All the objects used in the application are located in the entities package.

Table 10 - Mobile Tolling: Modules Description

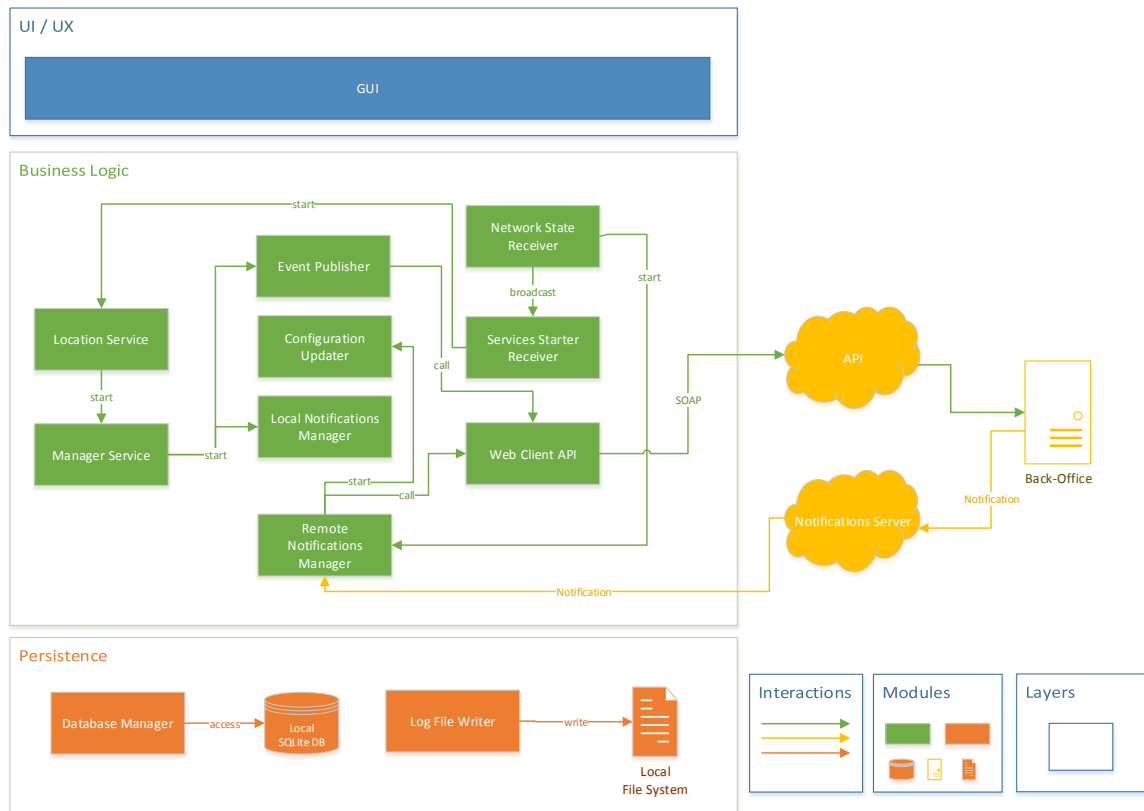


Figure 13 - Mobile Tolling: Conceptual Architecture Diagram

5.2.4. User Interface & User Experience (UI/UX)

With the purpose of providing the desired UX, the lateral navigation design pattern [16][27] was chosen to offer easy access to all information with minimum user interaction. For this the UI is divided in three main areas, Login/Registration and Home. The application flow, every screen and actions needed to access each one, are presented in Figure 14.

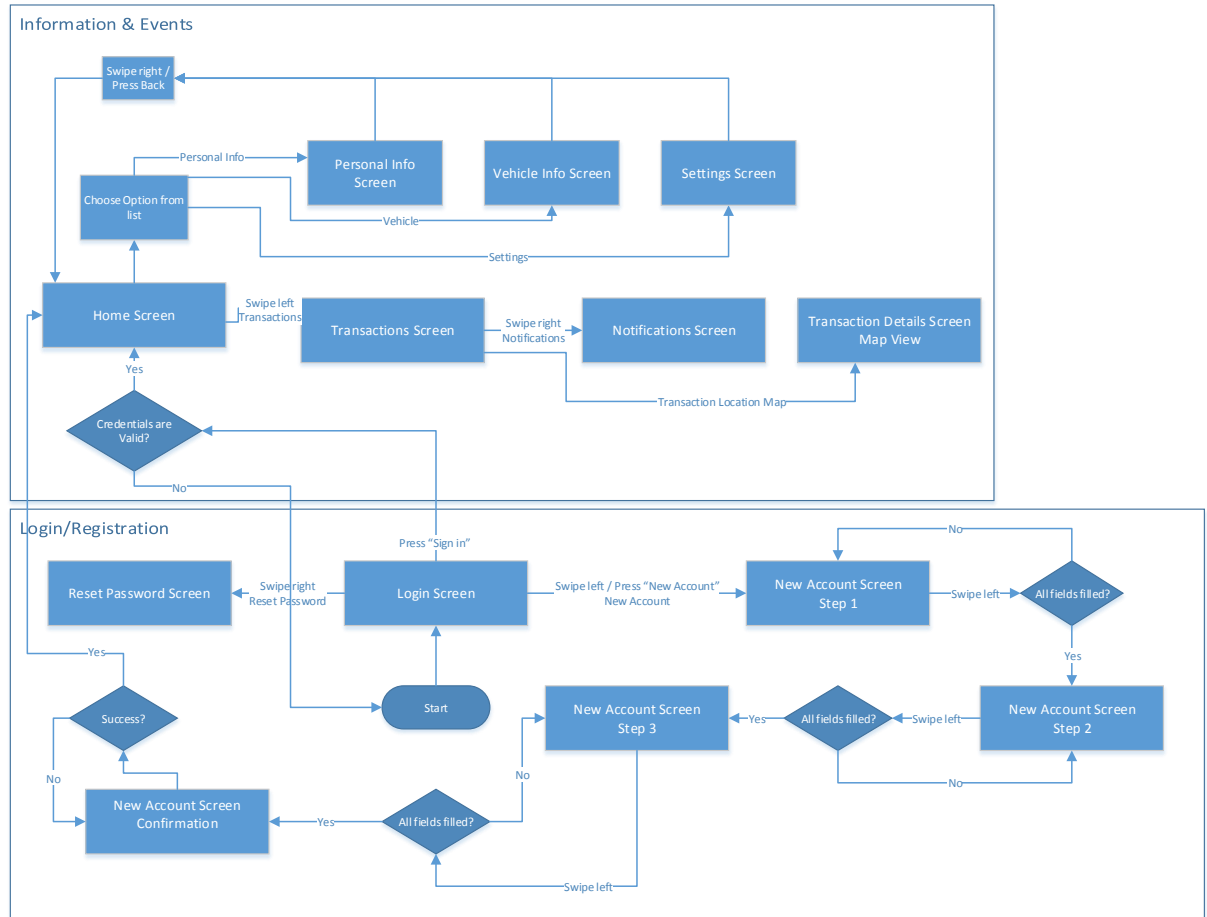


Figure 14 - Mobile Tolling: Application Flow Chart

In Table 11 is described the purpose of some of the main classes represented in Figure 14.

Class	Description
Splash Screen	This class is an Activity and is responsible for loading the first screen visible in the application, create the database baseline (in the first application use), and verify the account login status in order to decide which screen loads next, the login screen or the home screen.
Login Controller	This class has a pager widget to load the views that hold login information, and is responsible for controlling the login process, the password reset request and the new account procedure.

Home Screen Controller	This class has a pager widget to load the views that show the account summary, events and notifications. It also controls the updates of the events table and account summary.
Menu Details Controller	This class holds different views depending on the chosen menu item and controls the update actions available for each of the available menu views.

Table 11 - Mobile Tolling: Main Classes Description

5.2.5. Persistence

In order for the application to function properly either online or offline, and to be able to recover from errors, all critical data must be persisted to database. To simplify this process a central point of access was created, the Database Manager. This class has all the methods necessary to insert or retrieve data from the local SQLite database [20].

To help debug potential errors, at least during development stages, the Log File Writer class saves some important logs to database, so a track can be maintained of the application behaviour.

Class	Description
DatabaseManager	Central point to manage all database access. All classes that need to store or read something from the database use this manager.
LogFileWriter	This is a helper class to store some logs in database for debugging purposes if needed.

Table 12 - Mobile Tolling: Persistence Classes

5.3. Data Model Overview

The Mobile Tolling Application Data Model corresponds to the database view, which can be observed in Figure 15.

A brief description of the main tables can be seen in Table 13.

Table	Description
MT_Account	Account data is stored in this table. It contains the information necessary to identify the user account in each transaction.
MT_Fence	This table contains the details necessary to identify each available fence. The fences are needed to activate and deactivate the GPS in order to correctly locate the user and save battery at the same time.
MT_Gantry	The existing gantries used to identify the real toll points are stored in this table.

MT_Event	The generated events when the user passes a toll point are stored in this table.
MT_Event_Evidence	Each generated event has several evidences to prove the passage of the user through the toll point. These evidences are stored in this table.
MT_Message	This table contains the messages generated to send to the application. The MT_Message_Status table is consulted to check if there are pending messages to send.

Table 13 - Mobile Tolling: Main Tables Description

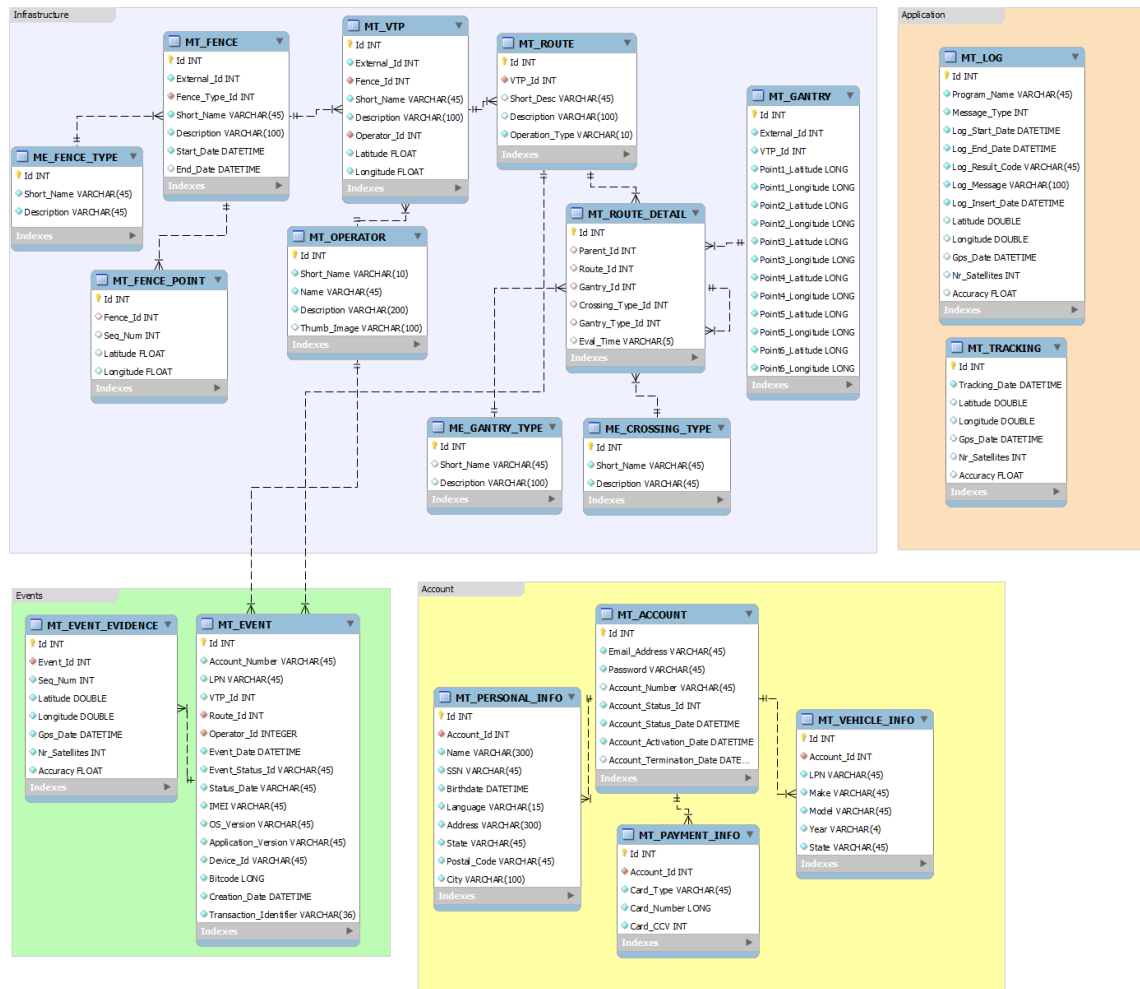


Figure 15 - Mobile Tolling: Application ER Diagram

In Appendix B is detailed the conceptual data model for the Mobile Tolling application.

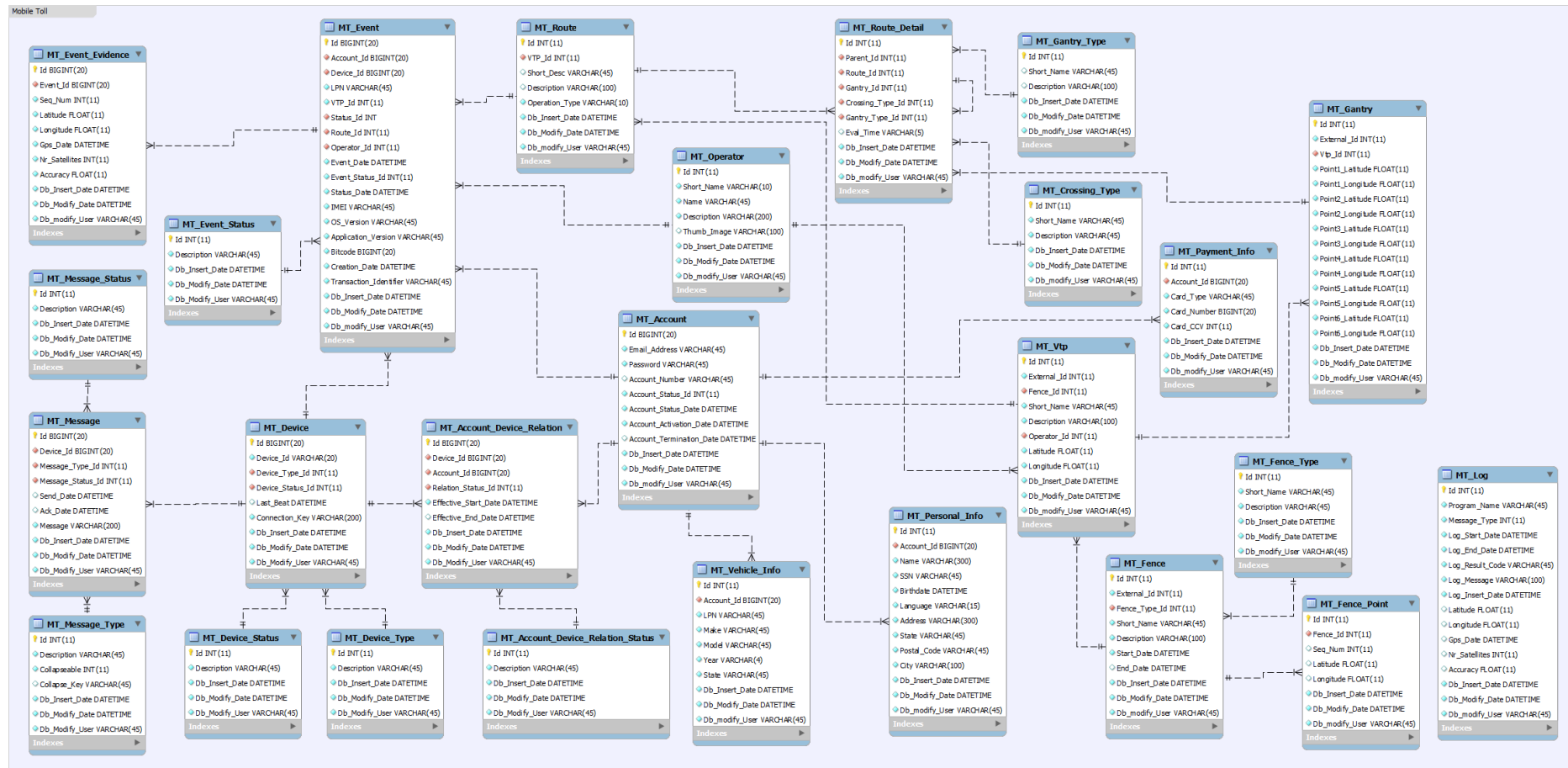


Figure 16 - Mobile Tolling: Back-Office ER Diagram

6. Quality Assurance

Quality is a major concern in a software project, therefore a good test plan is crucial to make sure we deliver a project with the best quality possible.

For this, the test plan was divided in three sections:

1. Tests Definition
2. Tests Execution
3. Results Analysis

The extensive list of tests can be checked in Appendix C, in this section only some test examples will be shown.

6.1. Tests Definition

6.1.1. Functional Tests

This set of tests checks the features proposed in the requirements to make sure all was implemented correctly and works as expected.

Test Condition	Expected Result
User starts the mobile application and authenticates successfully	User is redirected to the application home page
User is authenticated in the mobile application and is able to logout	User is redirected to the application login page
User starts the mobile application and is able to create a new account	New account is registered and the user is able to login to the application
User presses an option in the home screen list to check the information related to his account	The details page of the corresponding option is opened.
User accesses the Settings menu and disables the Location Tracker	Location tracking is disabled
User checks the list of transactions registered in the mobile application	A list of transactions is displayed to the user
User checks the list of notifications registered in the mobile application	A list of notifications is displayed to the user
Application generates a new crossing event	A new crossing event is generated and sent to back-office, transactions list and notifications list.
Application receives a notification from the server	Application parses the message and takes action

Table 14 - Mobile Tolling: Functional Tests

6.1.2. Technical Tests

This set of tests covers the technical performance of the application and the mobile phone.

Test Condition	Expected Result
Install the application	Application is installed correctly
Perform a memory analysis to the application	Application does not leave an excessive memory footprint. Memory is cleaned when application is not in use.
Perform connectivity tests : <ul style="list-style-type: none"> Start application with network connectivity; Start application without network connectivity; Lose connectivity while using the application; Regain connectivity while using the application; Change network while using the application. 	The application behaves correctly in all connectivity situations.
Perform a user interface analysis to the application (<u>Android</u> only)	The hierarchy view tree is not too deep. The android GPU overdraw has no dark red areas.
Perform a user experience analysis to the application	All actions are intuitive and easy to perform. No more than 3 clicks are needed to perform an action.

Table 15 - Mobile Tolling: Technical Tests

6.1.3. Back-Office Tests

This set of tests checks the features proposed for the back-office Mobile Tolling integration.

Test Condition	Expected Result
Register a Mobile Tolling Account	When the mobile device sends a register account request the data is inserted in the database, thus creating a user account.
Register a Mobile Tolling Event	When the mobile device sends an event data, a new Mobile Tolling Event will be registered in the database, as well as its Evidence. Additionally, an OD_Event will also be created and inserted in the database for the correlation module.
Create Mobile Tolling messages	The Mobile Tolling specific messages will be sent to the device in response to its request.

Table 16 - Mobile Tolling: Back-Office Tests

6.2. Tests Execution

To confirm the validity of the solution, the defined tests were executed using a set of toll points located in one of the highways managed by the Tolling solution developed by Accenture, the A13 highway in Portugal, so that the real toll point events could be accessed.



Figure 17 - Mobile Tolling: Virtual Toll Point Crossing Event

Figure 17 shows toll point 1337 (yellow pin) and the event when passing through it (red pin).

The results were gathered over the course of one month, from several passages through this highway, and compiled to compare the real event time sent by the road side equipment in the toll point, the virtual event time generated by the Mobile Tolling Application and the percentage of events that actually improved the confidence level of the license plate detected by the toll point cameras.

6.3. Test Results and Analysis

This section shows the tests made to the application to prove the success of the implemented solution. The results used for this analysis can be found in Appendix D.

The main objectives of these tests were:

- Prove that the mobile phone Location Service correctly detects the crossing of a toll point
- Prove that the mobile event and the real event are correctly correlated and improve the confidence level obtained through Video Toll Collection.

In order to filter the gathered results and work only with the useful events, the delta between the real event and the virtual event was calculated and from these delta times, the values calculated in table Table 17 for the variance, mean and standard deviation were used.

Variance	47666,21
Mean (ms)	345,04
Standard Deviation (ms)	218,33

Table 17 - Mobile Tolling: All Results Statistics

This way the results that are not within the standard deviation of the mean, as can be seen in Figure 18, were discarded. As can be seen in Figure 19, this resulted in 81% of the events being useful for the statistical analysis.

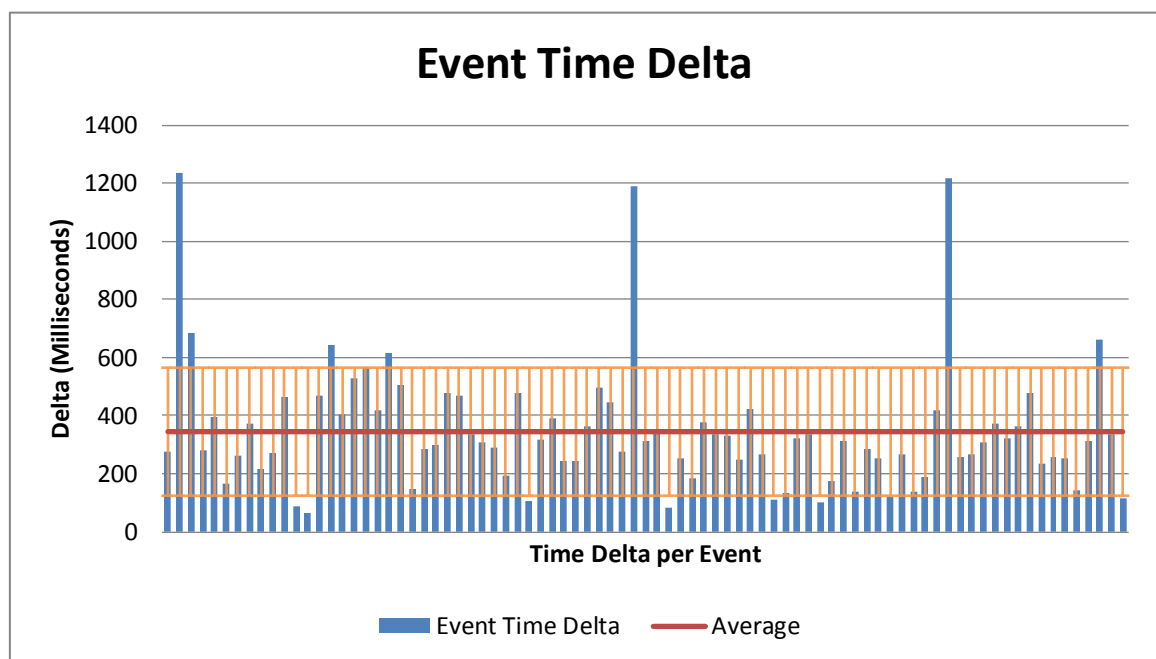


Figure 18 - Mobile Tolling: All Results Standard Deviation and Mean Graph

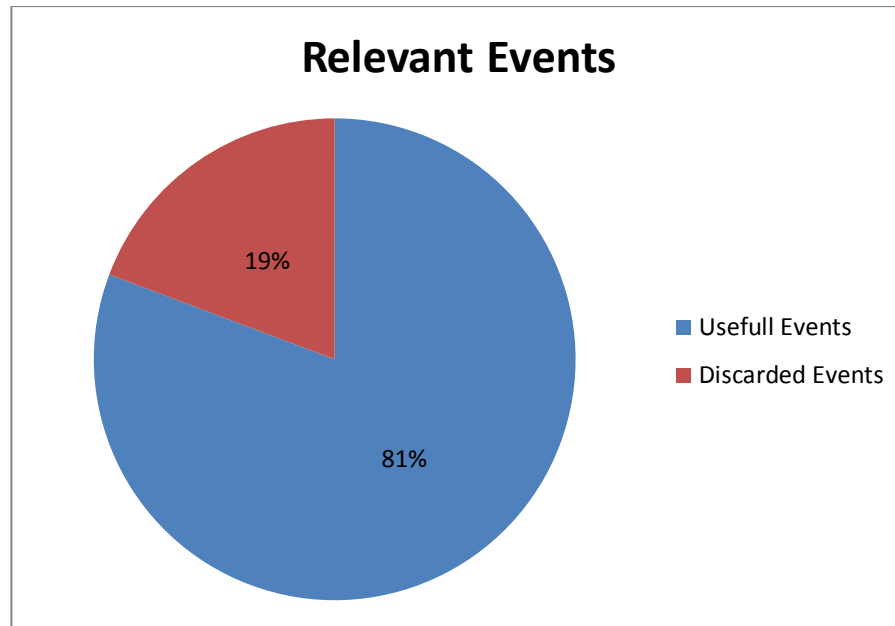


Figure 19 - Mobile Tolling: Relevant Events

For this new dataset, the same calculations were executed to find the mean and the standard deviation.

Mean (ms)	314,01
Standard Deviation (ms)	100,45

Table 18 - Mobile Tolling: Useful Data Statistics

As can be seen in Table 18, the average delta between the real event and the mobile tolling event is really small, less than half a second, which allows us to know that the location service and the virtual gantries work as expected, meaning that the configurations are accurate, achieving a good time performance result for the events generation.

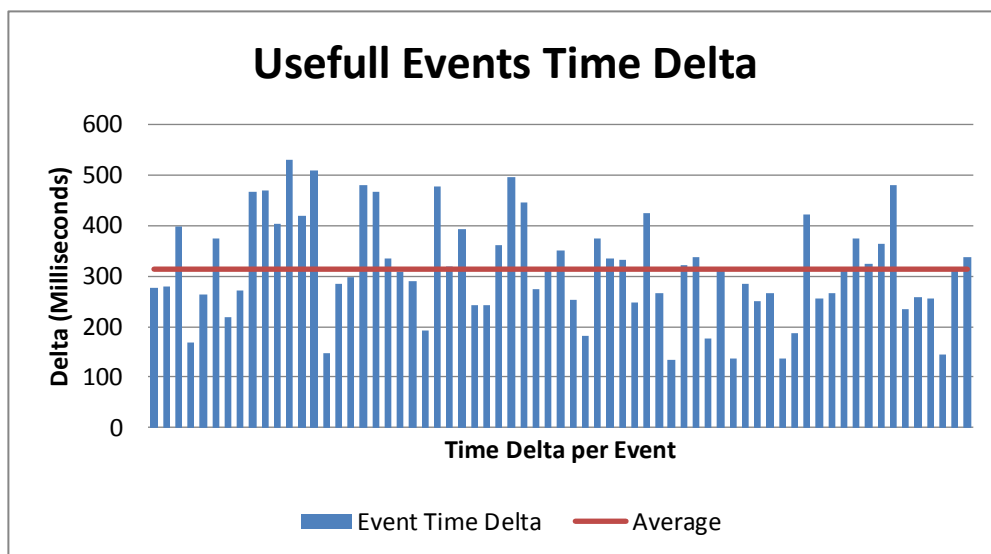


Figure 20 - Mobile Tolling: Useful Results Standard Deviation and Mean Graph

But in order to confirm that these results are indeed significant, they need to follow a normal distribution as seen in Figure 21.

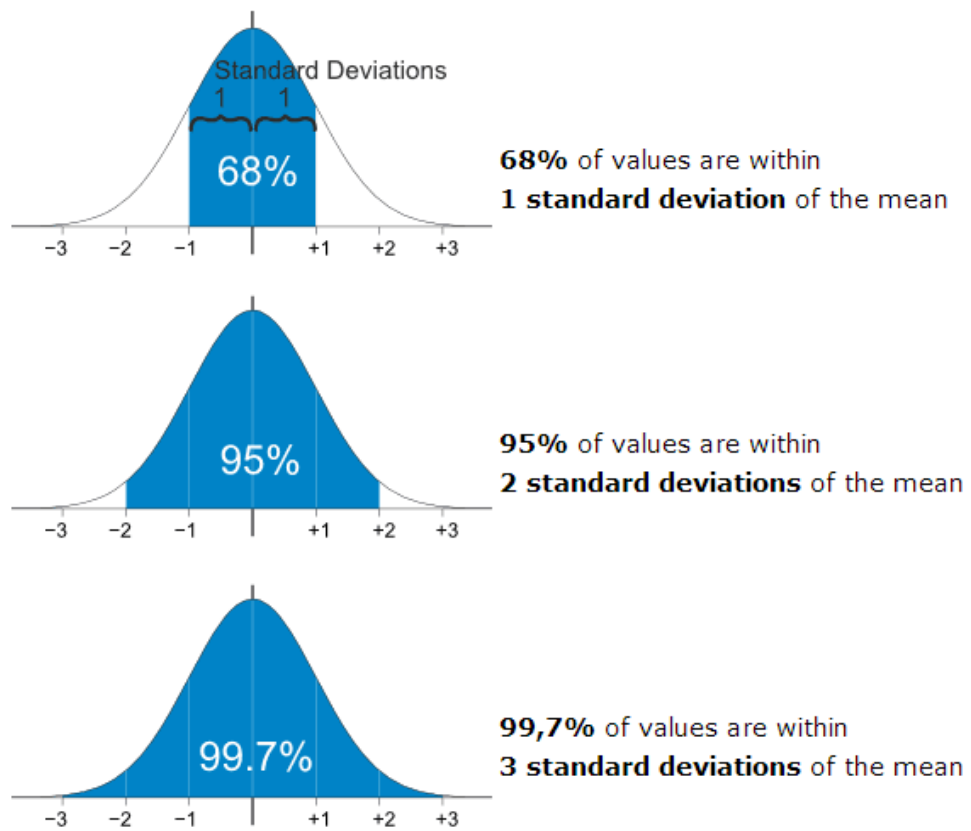


Figure 21 - Mobile Tolling: Normal Distribution Definition (Source [32])

Based on this we calculated the histogram and normal distribution for the gathered data and, as can be seen in Figure 22 and Figure 23 the results follow approximately the same shape as the normal distribution, the shape of a “bell”. Moreover we can see in Figure 23 that nearly all the results fall within two standard deviations of the mean.

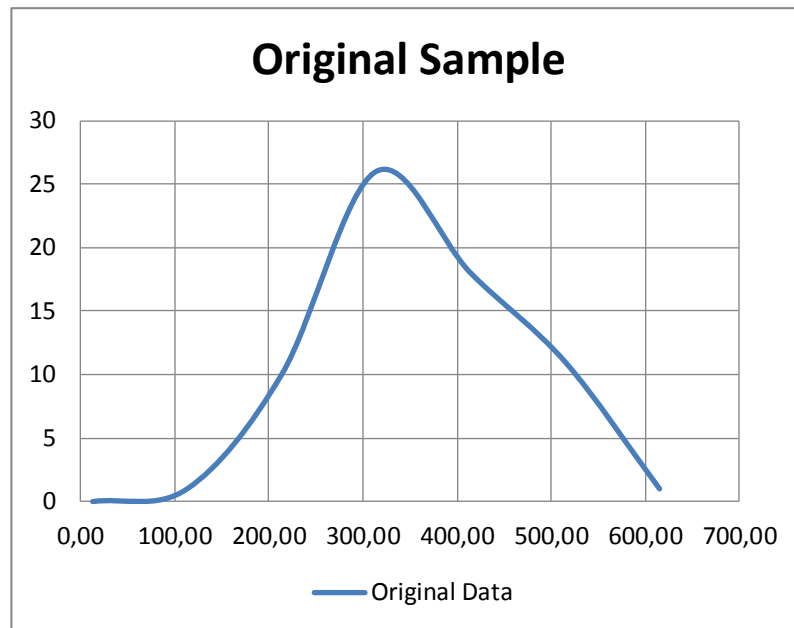


Figure 22 - Mobile Tolling: Results Histogram

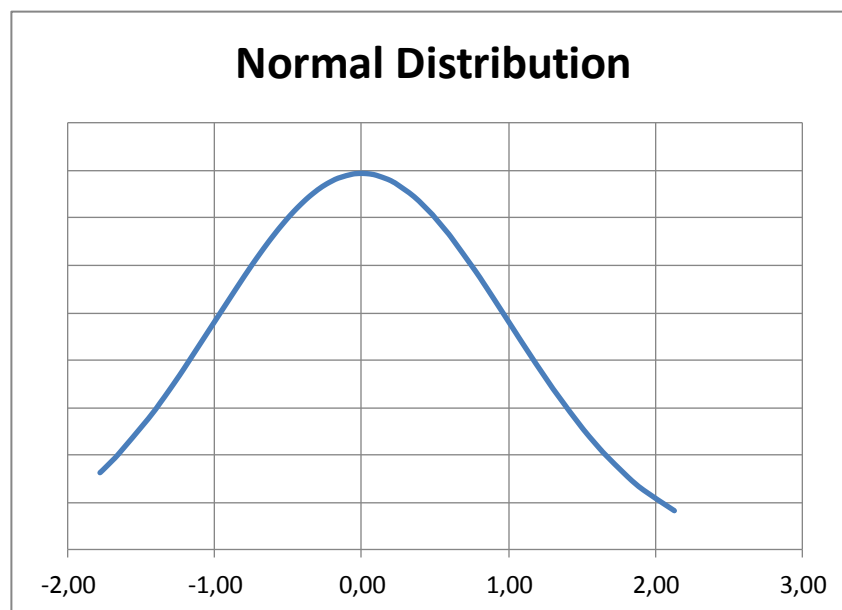


Figure 23 - Mobile Tolling: Results Normal Distribution

From the results gathered we can also see that, although the majority of the events had a very high level of confidence, 28% of the events were improved by the correlation of the real event and virtual event.

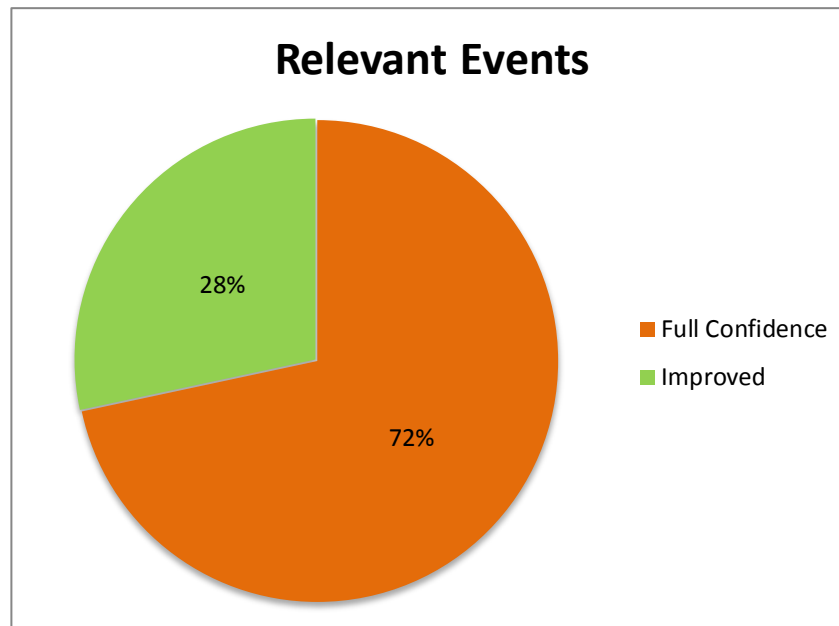


Figure 24 - Mobile Tolling: Improved Events

These tests confirmed that the detection of the events is very accurate and that it indeed helps in improving the confidence in detecting the correct license plates and client accounts.

7. Conclusion

The following chapter concludes this report by addressing some thoughts on the work done, the risks that were identified for this project, the future as well as the lessons learned during this internship.

7.1. Results Achieved

Because of the Java foundations already attained by the intern during his studies, and also because of the past developing Android applications, the contribution to the Android application started faster and more centred on the performance side of the application. In the end this application was completed and is fully functional and prepared for live demos and tests.

The Back-Office module was the smallest part to which the intern contributed, mainly to familiarize with the entire Back-Office and understand the technologies used, to be able to help the rest of the team if the necessity arose. The correlation of the real and virtual events was achieved using the Drools rule engine, and also some web service features were tuned.

With the iOS application, came the amazing opportunity of learning to code for the iOS platform. It was a very demanding task but also worthwhile, the application came out with the same functionalities as its Android counterpart, looking fantastic and working beautifully.

7.2. Future Work

This application still has a lot to grow so, in order to further improve tracking the position of the user and reduce the dependency on GPS, several other sensors can be used.

Also, the emerging tendency to bring Android and iOS devices to integrate the cars central consoles might improve the definition of Bluetooth categories or allow the introduction of this application directly in those devices.

Last but not least, due to its growth in smartphone market share, the implementation of the Mobile Tolling solution for Windows Phone devices is also an improvement to the solution.

7.2.1. Risk Pending

In this section we discuss the risks identified for this project that remain for future assessment, as well as possible mitigation plans for each of them.

User does not use the app

We know that for this solution to improve the results of video tolling, the user needs to use the application, so if the user does not wish to use the application the system resorts to the old video tolling method (without mobile tolling information). For this not to happen the operators must think of a business model to attract the users to this system, for instance, by offering discounts to those that use the application. This is considered a very important risk, but it must be resolved by toll agency by convincing the client to use the application, and benefit from it.

Several applications traveling in the same vehicle

It is possible that more than one mobile tolling user travels in the same vehicle. If both users have the application the back-office receives multiple events. This can be avoided by allowing only one account per vehicle, eliminating the events for the account that does not have this vehicle.

Another solution is the use of Bluetooth devices to detect when an application is in a vehicle, allowing only events to be sent if a Bluetooth device is paired, eliminating the applications that are not connected to a device. Since the multitude of manufacturers of Bluetooth devices not always use the correct device category, it is not possible to implement an identification system that automatically detects the device as being a vehicle and allows the mobile tolling application to function. There is no guarantee that the user is not using a mobile Bluetooth device, one that can be used outside of a vehicle. Because of this, this solution has been, for the moment, discarded.

This is a low priority risk but can be improved on nonetheless.

Even though these risks exist, it is our believe that this solution is an improvement to the available technology that accompanies the mobile evolution.

7.3. Lessons Learned

These past ten months have been very rewarding and, upon reaching the end of the internship, and looking back at all the successfully completed goals, the sense of accomplishment is remarkable.

Being able to build a product from the ground up and with it improve other company products is a very enriching experience. The work in a real corporate environment is very demanding and inspired the intern to always try harder and do better to meet the clients expectations.

With the addition of the mobile applications, Accenture now has a complete tolling solution, capable of supplying the best experience to both operator and its users, and was capable of being showcased gain potential new clients.

8. Bibliography

1. “About Accenture”, [Online]. Available: <http://www.accenture.com> [Accessed: January 2014]
2. "Electronic Toll Collection Systems", [Online]. Available: <http://mobility.tamu.edu/mip/strategies.php>. [Accessed October 2013].
3. Andriy Boychenko, “Implementing an intelligent process for Image Recognition”, Masters’ Degree in Informatics Engineering Internship Report
4. “V-Model”, [Online]. Available: [http://en.wikipedia.org/wiki/V-Model_\(software_development\)](http://en.wikipedia.org/wiki/V-Model_(software_development)) [Accessed: January 2014]
5. “MVC”, [Online]. Available: <http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> [Accessed: December 2013]
6. “Simple Example of MVC (Model View Controller) Design Pattern for Abstraction”, [Online]. Available: <http://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design> [Accessed: January 2014]
7. “Microwave DSRC Products by EFKON”, [Online]. Available: <http://www.efkon.com/en/technologies/microwave.php> [Accessed: January 2014]
8. C. Meghann and R. David, "Automated License Plate Recognition Systems," September 2012. [Online]. Available: <https://www.ncjrs.gov/pdffiles1/nij/grants/239604.pdf>. [Accessed: January 2014]
9. “GeoToll”, [Online]. Available: <http://geotoll.com/> [Accessed: October 2013]
10. “BancPass PToll”, [Online]. Available: <http://www.bancpass.com/phoneTolling.html#> [Accessed: January 2014]
11. “PToll - mobile phone tolling to be launched at CTRMA in Austin TX”, [Online]. Available: <http://tollroadsnews.com/news/ptoll---mobile-phone-tolling-to-be-launched-at-ctrma-in-austin-tx> [Accessed: January 2014]
12. “PToll Mobile Device Payment System”, [Online]. Available: <http://tollroadsnews.com/sites/default/files/PTollInfo2.pdf> [Accessed: January 2014]
13. “The Toll Roads”, [Online]. Available: <https://www.thetollroads.com/ontheroads/mobile-app.php> [Accessed: November 2013]
14. “Mobile phone app for toll account management now in use in Orange County west”, [Online]. Available: <http://tollroadsnews.com/news/mobile-phone-app-for-toll-account-management-now-in-use-in-orange-county-west> [Accessed: January 2014]
15. “Android Developers”, [Online]. Available: <http://developer.android.com/index.html> [Accessed: November 2013]
16. “Providing Descendant and Lateral Navigation”, [Online]. Available: <http://developer.android.com/training/design-navigation/descendant-lateral.html> [Accessed: November 2013]
17. “Implement Swipe Views”, [Online]. Available: <http://developer.android.com/training/implementing-navigation/lateral.html> [Accessed: November 2013]
18. “Location Services”, [Online]. Available: <http://developer.android.com/guide/topics/location/index.html> [Accessed: December 2013]
19. “Protocol Buffers”, [Online]. Available: <https://code.google.com/p/protobuf/> [Accessed: January 2014]

20. “Using Databases”, [Online]. Available:
<http://developer.android.com/guide/topics/data/data-storage.html#db> [Accessed: January 2014]
21. “Relatório de Tráfego na Rede Nacional de Autoestradas”, [Online]. Available:
<http://www.inir.pt/portal/RedeRodovi%C3%A1ria/Relat%C3%B3rios/tabid/142/language/pt-PT/Default.aspx> [Accessed: January 2014]
22. “SonarQube”, [Online]. Available: <http://www.sonarqube.org/> [Accessed: January 2014]
23. “PayUrToll”, [Online]. Available: <http://payurtoll.com/> [Accessed: May 2014]
24. “PayUrToll Patent”, [Online]. Available:
<http://www.faqs.org/patents/app/20140025444> [Accessed: May 2014]
25. “Redmine”, [Online]. Available: <http://www.redmine.org> [Accessed: May 2014]
26. “Drools”, [Online]. Available: <http://drools.jboss.org/> [Accessed: May 2014]
27. “Page View Controllers”, [Online]. Available:
https://developer.apple.com/library/ios/documentation/WindowsViews/Conceptual/ViewControllerCatalog/Chapters/PageViewControllers.html#//apple_ref/doc/uid/TP40011313-CH4-SW3 [Accessed: March 2014]
28. “Getting the User’s Location”, [Online]. Available:
<https://developer.apple.com/library/ios/documentation/userexperience/conceptual/LocationAwarenessPG/CoreLocation/CoreLocation.html> [Accessed: April 2014]
29. Juan Guillermo Jordán, “A pilot road pricing application based on GNSS”, Valencia, 16th November 2004, IV Road Pricing Seminar. [Online]. Available:
http://robotica.uv.es/RoadPricing4/docs/9_A%20pilot%20road%20pricing%20application%20based%20on%20GNSS.pdf [Accessed: June 2014]
30. Benito Úbeda, Rafael Toledo, Juan Guillermo Jordán, Soledad Montes, “A Theoretical and Practical Analysis of GNSS Based Road Pricing Systems, considering The Egnos/Sisnet Contributions”. 8-10 December 2004, ESA/ESTEC, Noordwijk, the Netherlands. [Online] Available:
http://ants.inf.um.es/ants_vehicles/files/Navitec2004_1.pdf [Accessed: June 2014]
31. “Normal Distribution”, [Online]. Available:
http://en.wikipedia.org/wiki/Normal_distribution [Accessed: June 2014]
32. “Normal Distribution”, [Online]. Available:
<http://www.mathsisfun.com/data/standard-normal-distribution.html> [Accessed: June 2014]