UC3ADF10 Application Analysis of AtB

Lynn-Mari Spencer

Abstract

The objective of this study was to examine the AtB application, developed by the public transport operator AtB in Trøndelag, to identify forensically relevant artefacts and determine significant aspects of the AtB application within a forensic context. Through the use of an emulator, an analysis of the application life cycle and user interactions has been conducted. The principal results show that personal identifiable information and favourite bus stops and bus lines are stored in plain text and are easily available. The currently examined public transportation application AtB, provides a range of forensic artefacts that, together with other information, can contribute to building timelines and user profiles in an investigative setting.

Keywords: AtB, Public Transport application, Application Life Cycle Analysis, Emulator, Android forensics, Artefacts

1 Introduction

Public transport is an integral part of societal structure, and as society is increasingly digitalised, different service providers also need to move to digital solutions. The traditional bus ticket has gone through a digitalisation, where an application is the rule rather than the exception. Most public transport companies develop and maintain their own applications, meaning that those moving between different geographical areas, whether in a country or between countries, are likely to need multiple applications.

An application supports several purposes for both the user and the transport company. For the transport operator, the application is necessary to provide relevant services and collect fares. For the passengers, the application is important to get access to information, pay the fare and be able to produce a valid ticket in case of controls.

This study explores the public transport application AtB. According to AtB (2025) the AtB application is developed by ATB, the public transport operator in Trøndelag. The application aims to provide information to travellers on departures of buses, trains, and express boats in real time. Users can plan their trips, buy tickets for themselves or others, and receive relevant or important operational messages from AtB in real time (AtB, 2025). If a user wants to buy tickets for others and send them, the recipient also needs to have the application downloaded and be logged in. Within the application, users can also lodge customer support requests or supply feedback to AtB.

The application requires Android version 7.0 or newer, the application has more than 100,000 downloads, and has received 776 reviews, adding to a total of 4.1 (AtB, 2025). According to AtB (2025) the application collects a range of data such as device ID, financial information and purchase history, personal information, application activity or application interactions, application information and performance. In terms of security, the data is not encrypted in transit, and it is not transferred over a secure connection (AtB, 2025). The PEGI for the application is set to 3 (AtB, 2025), which means that children with a smartphone can also download and use it. The application can be downloaded from AtB - Apps on Google Play (AtB, 2025).

This study reports on the forensic application analysis of the AtB application, with the aim of identifying potential forensic artefacts and examining their forensic significance.

The study is of an exploratory nature and will examine whether there are relevant forensic artefacts tied to the used functions of the application. In addition to artefacts generated through user-related actions, artefacts generated from installation and artefacts left over after uninstallation will be examined, adding to a three-step life-cycle analysis of the application.

The application is chosen as it is the solution provided by the public transport operator in Trøndelag for buying tickets for transport in the area of Trøndelag.

For the following paper, Section 2 will provide information on related work, and Section 3 will outline the methodology. Section 4 presents the results and discussion, while Section 5 contains the conclusion of the study.

2 Related Work

The area of public transport applications appears to be sparsely researched, despite the amount of personal information they typically collect and their wide use. The limited studies that do exist, however, reveal notable security and privacy issues.

One example is the work of Kitsaki et al. (2018) who examined 15 different Android applications in the categories banking, public transport and mobile network carriers through code analysis and what they refer to as a disk analysis. This analysis is defined as the examination of files created and managed by the application (Kitsaki et al., 2018). The analysis of the unnamed public transport application showed that the application provided information on bus routes, locations and favourites. They found that personal information connected to favourite routes and search history could be retrieved in plaintext from the table `SearchHistory` in the folder `databases`. Kitsaki et al. (2018) recognise the forensic value of these findings; however, they also highlight the problem it poses in terms of privacy. In their discussion, Kitsaki et al. (2018) put forward an expectation that the situation with sensitive data stored in plaintext to be resolved in the near future due to the increasing security in new devices and increased use of security technologies.

Complementing the findings of Kitsaki et al. (2018), Sanz Maroto et al. (2020) conducted focused research on the m-ticket app used by Lothian Buses in the UK. They uncovered several vulnerabilities within the application, which opened up for a range of potential incidents, including DoS attacks, information leakage and reactivation of tickets, potentially affecting the revenue of the transport operator. They also found the reset mechanisms for passwords to be poorly implemented, making it easy to brute-force user credentials and exploit the information. One of the reasons for ticket-related vulnerabilities was identified as the possibility within the application to activate and generate tickets regardless of internet connection (Sanz Maroto et al., 2020). These findings further support and highlight the importance of research into this group of applications.

The broader relevance of these findings and concerns was also reflected in the related area of airline applications. Maryam and Hussain (2023) conducted a forensic analysis of airline applications for Android. These applications store much the same type of information as public transport applications, and they serve the same purpose. In their study, Maryam and Hussain (2023) found all five examined applications to store different types of information, such as passport information, login information, membership numbers, email messages with passwords and payment information in plaintext.

The results of Maryam and Hussain (2023), together with the findings reported by Kitsaki et al. (2018) and Sanz Maroto et al. (2020) suggest that the issues related to security and privacy are part of a greater pattern spanning several types of travel-related applications. Even though the security and privacy of these applications are questionable, they do hold forensic potential.

2.1 Methodologies

Kitsaki et al. (2018) used a rooted Android phone and conducted an analysis of the files created and managed by the application under investigation. In addition, they used DB Browser for SQLite to examine the files on the phone; no further details on the procedure are provided in the article. During the analysis, they found a `SearchHistory` table within the folder `databases`, which detailed the users' search history related to buses and a table called `Favourites` storing all the users' saved bus routes. Kitsaki et al. (2018) also reported on finding the folder `cache/volley/` where data regarding user requests to the server was stored, providing information on IP addresses and data related to the information requested by the user.

To examine the m-ticket application Sanz Maroto et al. (2020) employed a combined methodology comprised of static analysis in terms of code auditing and reverse engineering, and dynamic analysis, which entailed testing the application while it was running. The dynamic analysis was also two-fold, with one part examining the application's communication with the server and a second part examining the changes caused by the application on the device itself. When examining changes to the device, Sanz Maroto et al. (2020) used a rooted Android phone and examined app-related information stored in the well-known `/data/data` folder. This folder contained the following folders: code_cache, databases, files, and shared_prefs. Sanz Maroto et al. (2020) does not report detailed findings; however, the provided figure shows that the database folder contains three databases: `corethree`, `google_app_measurement.db` and `com.microsoft.appcenter.persistence`. They do, however, report on finding the majority of ticket-related information within the `/files/.config/data.jason` folder. This finding was deemed significant as it allows for the manipulation of tickets (Sanz Maroto et al., 2020).

Maryam and Hussain (2023) used the NIST SP 800-101 Rev. 1:2014 outline detailing mobile device forensics. They examined the traces left behind on the device by the applications upon download and installation, while performing different user-related activities and when the application was uninstalled. A rooted Android device was used, a physical disk image was created for examination, and Autopsy was used for analysis. Most of the findings reported by Maryam and Hussain (2023) were located in the main directory `/data`. However, some findings relating to the download and installation were found in the main directory `/permissions` , and findings related to the booking of flights were found in the main directories `/app` and `/com`. These findings were obtained during the investigation of the *Pakistan International Airlines (PIA)* application.

Anglano et al. (2016) report that several locations are relevant to explore when conducting an application analysis, but in particular, databases located at `data/data/databases` where the application often stores user-related content and `data/data/shared_prefs` , which may contain application settings, preferences and user-related information.

According to Sanz Maroto et al. (2020) using a rooted device allows for privileged or rooted control of the device. Sandvik (2017) further details the access to allow a forensic investigator to read the unencrypted file system or raw block devices. In other words, rooting a device allows for greater access to areas such as the file system, configurations and logs, making this information more accessible to the investigator or researcher.

An alternative approach to research on Android without using real devices is the use of emulators. Anglano et al. (2016) conducted a forensic analysis of applications for Android smartphones using an emulator. —They advocate that emulators are a cost-effective way to perform application analysis, as well as allowing for greater repeatability of studies. They also argue that it is easier to replicate settings in an emulator than on a phone. Anglano et al. (2016) report validating their findings using real devices, which yielded identical results on the emulator and real devices. Almuqren et al. (2024) further adds to the argument of using emulators, as they note that it is challenging to protect the integrity of the data a device holds. This is because mobile devices continuously run, making it difficult to make a concise bitwise copy. These highlighted issues may provide support for the use of emulators in forensic application analysis, at least in a research context.

In their 2017 study, Anglano et al. argued that the first stage of application analysis is an analysis of application functionality, see Fig 1. This phase entails mapping out the different functionalities of the application in question, where the purpose is to identify functions relevant from a forensic perspective. The result of the functionality analysis provides the foundation for the following activity testing. Also in this study, Anglano et al. (2017) used an emulator and validated the results with real devices, yielding identical results (Anglano et al., 2017).

The summary of related works shows that research into applications developed for the public transport sector is limited, and the few studies that have been conducted have used rooted Android phones. The current investigation into the AtB application will, as such, add to the very limited body of research in two ways. First, the current study will employ an emulator to conduct the research and will accordingly add to the methodology of research into the area of public transport applications. Second, this study will investigate a Norwegian public transport application, which, to the best of the researchers' knowledge, has not been previously done.

3 Methodology

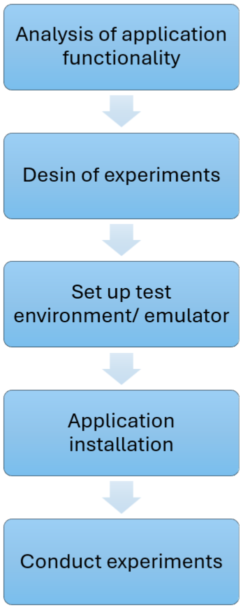

Analysis of application functionality
Desin of experiments
Set up test environment/ emulator
Application installation
Conduct experiments

Figure 1: Research Process

The testing of this application followed a stepwise and systematic approach, inspired by the methodology used by Anglano et al. (2016) and Anglano et al. (2017). The methodology is two-fold, with one process for planning and gathering data, and one process for analysing data. These are illustrated in Fig.1 and Fig. 2

3.1 Research Process

The first phase is to conduct an *"Analysis of the application functionality"*. To analyse the functionality, a manual inspection of the application is conducted, supported by the documentation from the application developer. The AtB application has a range of user functions which can be seen in Fig 3. It is not feasible to test all functions within one study; hence, functions marked with a red box are considered relevant for the design of experiments in light of a forensic context, and these are included in the experimental steps in Fig5.

The application also provides two options in terms of

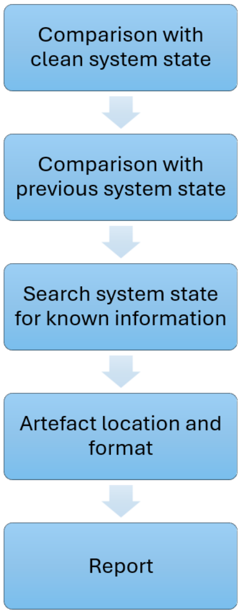

Comparison with clean system state
Comparison with previous system state
Search system state for known information
Artefact location and format
Report

Figure 2: Analysis Process

using it in a logged-in or non-logged-in mode. A functionality analysis of the two states is also conducted, and the results of this analysis are depicted in Fig 4. This analysis shows the alternatives in terms of login and no login, as well as the options available for the two user modes. Functions to be tested are marked with red boxes, and included in the experimental steps in Fig5. For a profile with login using a phone number, only the phone number is necessary and stored. Other information in the profile, such as first name, last name and email address, is optional to provide. For this study, first name, last name and email are included in the profile. Within the settings area, there are two additional options: ”emptying the search history” and ”emptying gathered data”; the functions of these are not examined. For the application to gather data, there needs to be an explicit consent to this in the form of a selection box.

The second phase entails the *”Design of experiments”*. As identified in Fig 4, the experiments are conducted after logging in to the application using the phone number and an additional external device for confirmation. Then the following activities will be conducted:

- *Favourites*: There will be a search for the bus stops City Syd and Peder Morserts veg, then these are added as favourite stops. Second, there will be a search for bus lines 1 and 16, then these will be added as favourite lines.

- *Buy tickets*: A child fare ticket will be bought and sent to another number. Second, a child's fare ticket to self will be bought. Both payments will be through Vipps.

Table 1: Test Environment

Variable	**Setting**
Device	Samsung Galaxy S23 Ultra
IMEI	863818021974440
CPU	4 Cores
RAM	4096
Display Phone	1080x1920 (DPI 480)
Android Version	9

Table 2: Application Installation Details

Property	**Value**
AtB	1.72
Downloaded from	Google Play
E-mail account used	lysp39329@google.com

The third phase is to *”Set up test environment/emulator”*. The current project utilised the LD-Player 9 emulator (as detailed in Tab 3) to create a controlled environment to conduct the testing within. The utilised settings for the virtual device are detailed in Tab 1.

The fourth phase was *”Application installation”* , settings are detailed in Tab 2.

The last phase within the research process was to conduct the experiments, which is a structured process, adhering strictly to a pre-planned execution of steps. These followed the phases outlined in Fig5,

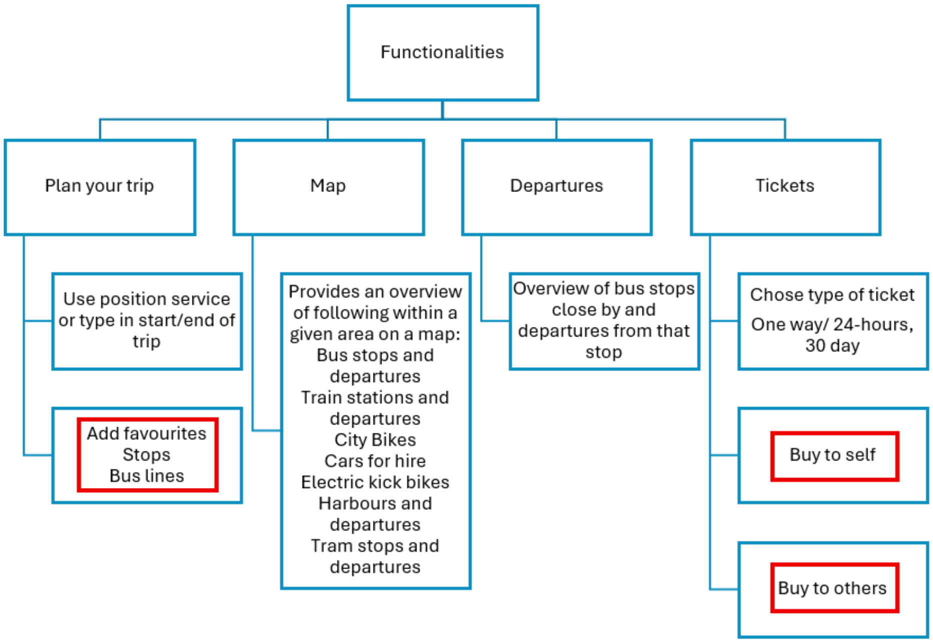

Functionalities
Plan your trip
Map
Departures
Tickets
Use position service or type in start/end of trip
Add favourites
Stops
Bus lines
Provides an overview of following within a given area on a map:
Bus stops and departures
Train stations and departures
City Bikes
Cars for hire
Electric kick bikes
Harbours and departures
Tram stops and departures
Overview of bus stops close by and departures from that stop
Chose type of ticket
One way/ 24-hours, 30 day
Buy to self
Buy to others

Figure 3: Functionalities

however, step 3 is worth taking note of. The experimental design was based on the assumption that the payment step would be transferred to an external device where two different payment solutions (Vipps and credit card) were set up. This was not the case, revealing an application dependency, where for the AtB application to work, a payment solution like Vipps had to be set up on the emulator as well, before it was possible to proceed with the original experimental setup.

3.2 Analysis Process

Once all the experiments were conducted and the data were gathered, the second part of the study, the analysis, was done. Tab 3 shows which tools and versions have been used, and the purpose of each when examining the vmdk-files. Vmdk is the file extension used for virtual machine disk image files. As illustrated in Fig 2 , the first step is to compare the vmdk-file where the application has been installed with a vmdk-file showing a clean system state. Furthermore, the analysis process involves comparing each set of experimental vmdk-files with the previous one to identify changes that the experiments have caused in the system, as well as locate relevant artefacts. As this study also examined the life cycle of the application, an examination of traces or artefacts created by the application upon installation and uninstallation is also investigated.

4 Results and Discussion

This section will first summarise the findings on the application lifecycle and the designed experiments, before the results are compared and discussed in terms of previous literature.

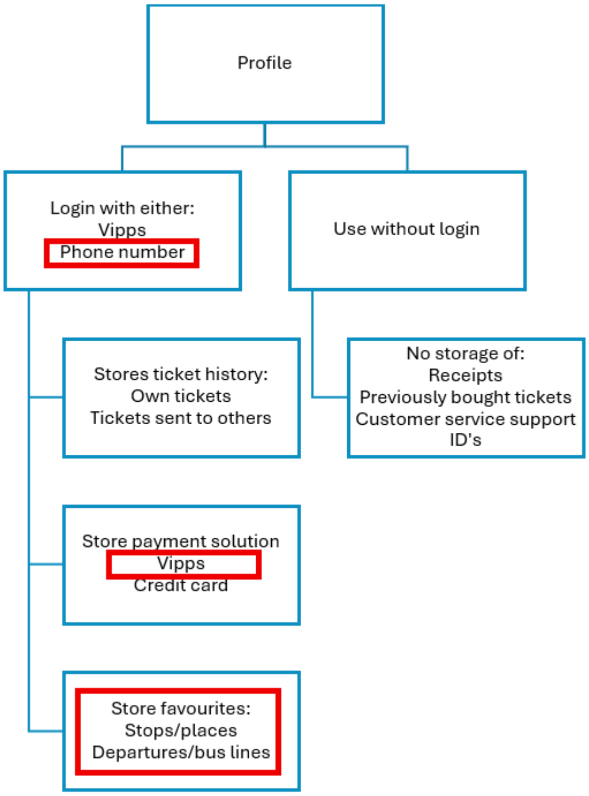


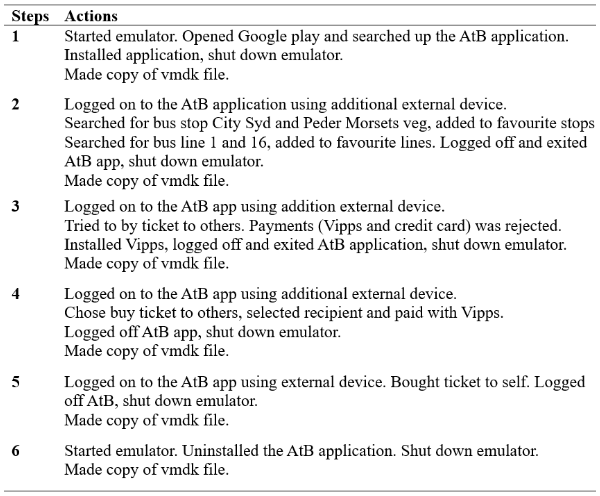
Figure 4: Login options

4.1 Results

The current study shows several forensically valuable findings at all stages of the life cycle analysis. The experimental scenarios are also yielding interesting findings.

4.1.1 Installation

Upon installation, the application creates a folder at `/app/no.mittatb.store-E5JXENHJmbRKBpd53XcQ-g==` and a folder at `/data/no.mittatb.store`.



Steps	Actions
1	Started emulator. Opened Google play and searched up the AtB application. Installed application, shut down emulator. Made copy of vmdk file.
2	Logged on to the AtB application using additional external device. Searched for bus stop City Syd and Peder Morsets veg, added to favourite stops Searched for bus line 1 and 16, added to favourite lines. Logged off and exited AtB app, shut down emulator. Made copy of vmdk file.
3	Logged on to the AtB app using addition external device. Tried to by ticket to others. Payments (Vipps and credit card) was rejected. Installed Vipps, logged off and exited AtB application, shut down emulator. Made copy of vmdk file.
4	Logged on to the AtB app using additional external device. Chose buy ticket to others, selected recipient and paid with Vipps. Logged off AtB app, shut down emulator. Made copy of vmdk file.
5	Logged on to the AtB app using external device. Bought ticket to self. Logged off AtB, shut down emulator. Made copy of vmdk file.
6	Started emulator. Uninstalled the AtB application. Shut down emulator. Made copy of vmdk file.

Figure 5: Experimental steps

In `/data/no.mittatb.store/databases` there are three databases created: `google_app_measurement_local.db`, `RKStorage` and `firestore.%5BDEFAULT%5D.atb-mobility.platform.%28default%29`. In the database `RKStorage`, table `catalystLocaStorage`, variable `REACT_QUERY_OFFLINE_CACHE` , the installed version of AtB (1.72) is found together with a timestamp of when the application was installed: 1756840136373, given in Unix Epoch Time. This translates into 02.09.2025 19:08:56, however, it does not correspond completely with the time the application was downloaded, as the time is about 6 minutes off. This database also holds an install-ID *28fdf44c-0c55-462e-b4e6-3fbf767fe00d*, which according to Firebase (2025) is used by the service provider to uniquely identify a particular installation by of an application, and gather information on the install. The install-ID is not found any other places than the `RKStorage` database.

In `/data/no.mittatb.store/files/device-id`, the device-ID is found to be *30706d0c-dcb3-4a6b-9717-9f8483a50927*, and in `/data/no.mittatb.store/files/internal-device-id`, the internal device-ID is found to be *a7633f91-ab7b-49b9-a3ae-d766f7506c*. These IDs are not found in other places; however, a second device-ID is found at

Table 3: Tools Used

Tool	Version	Purpose
LD Player	9.1.57.1	Create emaulator and conduct experimental scenarios.
FTK Imager	4.7.1.2	Primary analysis tool and used to export databases for further examination.
Autopsy	4.21.0	Used for verification of analysis findings.
Magnet Axiom Examine	9.2.0.44134	Used for verification of analysis findings.
Anaconda Powershell Prompt	1.23.12811.0	Used to write write ahead logs (wal) and .shm files to the corresponding database.
DB Browser for SQLite	3.13.1	Used to examine the content of exportet databases.

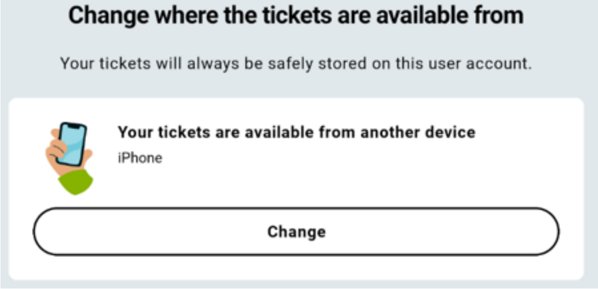
`/data/no.mittatb.store/cache/http-cache/3cdb2d5f8e3d97df9f6b824bd0e16c0f.1` and this ID is attached to the token-id used for tickets, as detailed in Tab 4.

The database `Frosting.db` contains a record of the AtB application being installed. The database `library.db` shows that the AtB application was downloaded from Google Play using the email account *lysp39329@gmail.com*. Both databases are located at `/data/com.android.vending/databases/`.

Within the area `/data/no.mittatb.store/shared_prefs`, several different tokens used for buying tickets are found; these can be seen in Tab. 4.

4.1.2 Experimental Scenarios

The first experimental scenario was to store two favourite bus stops and two favourite bus lines. In the database `RKStorage`, table `CatalystLocalStorage` key `@ATB_search_history`, search history is found to be stored in plaintext together with the distance between the device and the different searches at the time of the search. In the same database and table, but the key `stored_user_locations` favourites are stored in plaintext together with the distance between the device and the stored favourites. The application also stores the last time it was run. This can be found at `/data/no.mittatb.store/cache/last-run-info`. Upon logging on to the AtB application on the emulator the application asked if there should be a change in where the tickets are available from, see Fig 6. Despite choosing not to change, some previous ticket history was available in the `firestore.%5BDEFAULT%5D.atb-mobility.platform.%28default%29` database, table `target_documents` in the BLOB form.



Change where the tickets are available from

Your tickets will always be safely stored on this user account.

Your tickets are available from another device

iPhone

Figure 6: Ticket Availability Choice

The next experiment was to buy a ticket and send it to another user. The first attempt at buying a ticket failed as no payment options were installed at the emulator. The assumption that the payment would go through an external device was not correct, and Vipps had to be set up on the emulator. The records of the failed ticket payments are found in the `firestore.%5BDEFAULT%5D.`

Table 4: Token Names and values.

Token Name	Value	Path / Source
entur-tokens:main:tokenValidityStart	Given in Unix Epoch time 02.09.2025 19.02.01	/data/no.mittatb.store/shared_prefs/appPrefs.xml
entur-tokens:main:tokenId	Alphanumering value in the form 8-4-4-8-12 values	/data/no.mittatb.store/shared_prefs/appPrefs.xml
entur-tokens:main:tokenValidityEnd	Given in Unix Epoch Time 01.11.2025 20.02.01	/data/no.mittatb.store/shared_prefs/appPrefs.xml
entur-tokens:main:tokenCertificate	Base-64 encoded certificate	/data/no.mittatb.store/shared_prefs/appPrefs.xml

atb-mobility.platform.%28default%29 database under remote_documents and divided into three different entries in the database. These show that payments were tried through both Vipps and Netsaxept, but they were rejected or failed.

Once Vipps was set up on the emulator, it was possible to buy a ticket for another user and send it. The record of the buy and send was found in the firestore.%5BDEFAULT%5D.atb-mobility.platform.%28default%29 database under remote_documents, divided into three different entries. These entries showed that the ticket was purchased using Vipps and the Vipps token, the fare of 22 kr, the customer account used to buy the ticket *y7SQvjxuffdVv1OsOOFKquy7CBv1*, that the ticket was bought via the application, that the ticket was a sent fare contract (hence to someone else) and that the ticket was pre-activated. Lastly, the entries stored the first name, last name and email address of the paying user in plaintext. When buying a ticket for oneself, the same personal information as detailed above was found in the firestore.%5BDEFAULT%5D.atb-mobility.platform.%28default%29 database under remote_documents.

Each ticket buy was divided into three records in the table remote_documents under the variable contents where the data was stored as a BLOB. The BLOB was binary; however, the information, such as user first name, last name, email address, ticket price, and payment method, was stored in plaintext and hence easily available

4.1.3 Uninstallation

Once all the experiments were conducted, the application was uninstalled by marking the application icon and choosing 'uninstall'. Upon examination of the vmdk-file created after the uninstallation, several traces of the AtB application were found. At the location /system/graphicsstats/1756771200000/ a folder named no.mittatb.store was found, same on the location /system/graphicsstats/1761004800000/ where a folder named no.mittatb.store was found. At the location /system/package_cache/1/ the folder no.mittatb.store--E5JXENHJmbRKBpd53XcQ-g== was located. In the location /system/dropbox there are several keymasters timestamped in one of the time intervals when tickets were bought using the AtB application. The database auto_update.db contains a record of AtB. The database Frosting.db contains no records of the application, but the database gass.db located at /data.com.google.gms/databases contains a record of the AtB application together with a version code and SHA-256 value of *4cf35448fe3b98c84676d0843aaf6aa606cc248b91d9be96e021265f0c540021*.

4.2 Discussion

Like Kitsaki et al. (2018), the current study found that the public transport application under examination stores favourite locations, bus lines and search history easily accessible in plaintext. Kitsaki et al. (2018) expected the situation regarding sensitive data being stored in plaintext to change in the ”near future” due to security technologies like TEE (Trusted Execution Environment) and the increased inclusion of security technology in new devices. However, some years later, this is still a problem as illustrated by the study of Maryam and Hussain (2023) and the current study, despite the AtB application being rather newly developed, with its first release in spring 2021(AtB, 2021), but continuously developed in the following years, as previous applications were to be taken off the market in April and May 2024 (AtB, 2024).

The current study also found the application to store the distance between places searched and the device's location, as well as favourite bus stops saved and the location of the device. This was found in the database `RKStorage` , table `catalystLocalStorage` and the keys `stored_user_locations` and `@ATB_search-history`. The information was stored in *json*, hence easily retrievable and readable. When the user bought tickets, the first name, last name, and email address in plaintext together with the ticket information and payment details, such as payment method were easily accessible from the database `firestore.%5BDEFAULT%5D.`
`atb-mobility.platform.%28default%29`
under `remote_documents` stored as a BLOB. The payment information available is, however, not as detailed as the results retrieved by Maryam and Hussain (2023) when examining airline applications.

As AtB requires either Vipps or a phone number (where a unique code is sent upon every login), it can be assumed to be less prone to brute forcing attacks like the finding of Sanz Maroto et al. (2020). Despite a thorough search of the gathered data material, it was not possible to locate the user profile with the provided phone number, first name, last name and email address.

Contrary to the findings in Sanz Maroto et al. (2020) the AtB application requires an internet connection to function. The application records, down to the second (Rønning, 2025), when a passenger enters a bus, providing detailed information on location data (of both passenger and bus) and timestamps. Due to the study being conducted using an emulator, this was not possible to explore.

The experiment also highlights the importance and relevance of dependencies between the application under investigation and other applications or functions of an Android operating system. For instance, to conduct payments with AtB, there had to be a setup with either Vipps or BankId to get payments through. It was also necessary to activate location tracking to retrieve bus stops or other relevant services near the device's location. This application dependency might strengthen the forensic significance of the findings within the application in terms of the identity of the user, locations and use of the application.

The findings of the current study coincide with the findings of research into other studies on public transport applications (Kitsaki et al., 2018; Maryam & Hussain, 2023; Sanz Maroto et al., 2020), highlighting the utter importance of the principle of security by design (Kitsaki et al., 2018) , when developing applications using and gathering a range of personal identifiable information and other information, which together provide a significant image of an individual's habits.

4.2.1 Limitations

A limitation of the current study is the fact that it is conducted on an emulator. This limits the possibility of testing for true location data, for instance, as a bus is entered. The use of an emulator also strongly limited the testing of other functions, such as city bikes or electric kick bikes.

AtB clearly states that data is not encrypted in transit, and it is not transferred over a secure connection

(AtB, 2025). The design of this study, not utilising live memory forensics, might have severely impacted the possibility of finding payment details in plain text.

5 Conclusions

The current study has shown that a recently developed Norwegian application for the public transport operator AtB stores a significant amount of information regarding the use in plaintext. With access to the application, one can find the first name, last name, email address, locations, frequency of tickets bought for the user's own use, tickets sent to others, and the fare zones, hence which geographical area of Trøndelag the user is travelling within. It is also possible to see how the tickets have been paid for, whether that is Vipps or a credit card, and if the tickets are bought using the application.

It can be argued that the forensic potential of this application is significant. To log in and use the application, either a mobile phone number or a Vipps account needs to be used. Both types of information are strictly regulated and linked to a named individual. Furthermore, the use of an email address, the first name and the last name, together with location data, can provide valuable information in an investigation. It is, however, necessary to point out that this type of information is voluntary to add to the profile and can either be manipulated or left out. Cases where this information is relevant might be diverse, but the retrieved data might support or refute a person's movements through the use of location data and timestamps. This can be done as part of verifying an alibi, event reconstruction or creating a timeline. It can also be used in terms of payment or ticket fraud; it may be useful in cases related to harassment, stalking or domestic abuse, as well as accident investigations or transport safety.

The current study encountered some issues while conducting the application analysis through the emulator. Future work could replicate the current procedure, but do the testing on an actual Android device. This should allow for more realistic location data, and it would also make it possible to test for functionalities such as city bike rentals or electric kick bike rentals. In the application description, it is explicitly stated that data is not encrypted in transit (AtB, 2025), a natural next step could be to examine the application for live artefacts such as payment information and personal identifiable information during a ticket buy. Another possibility is to examine and compare the results with other similar applications in Norway, such as Ruter, EnTur or Skyss Billett.

References

Almuqren, A., Alsuwaelim, H., Hafizur Rahman, M. M., & Ibrahim, A. A. (2024). A systematic literature review on digital forensic investigation on android devices [International Conference on Machine Learning and Data Engineering (ICMLDE 2023)]. *Procedia Computer Science*, *235*, 1332–1352. https://doi.org/https://doi.org/10.1016/j.procs.2024.04.126

Anglano, C., Canonico, M., & Guazzone, M. (2016). Forensic analysis of the chatsecure instant messaging application on android smartphones. *Digital Investigation*, *19*, 44–59. https://doi.org/https://doi.org/10.1016/j.diin.2016.10.001

Anglano, C., Canonico, M., & Guazzone, M. (2017). Forensic analysis of telegram messenger on android smartphones. *Digital Investigation*, *23*, 31–49. https://doi.org/https://doi.org/10.1016/j.diin.2017.09.002

AtB. (2021, May). *Årsrapport 2021. atb - til stede i trøndelag* (Report). AtB. https://www.atb.no/getfile.php/1397238-1736252538/Rapporter/%C3%85rsrapporter/AtB%20%C3%A5rsrapport%202021.pdf

AtB. (2024). Nå trengs bare atb-appen! her er datoen mobillett-appen slutter å virke. https://www.atb.no/getfile.php/13110845-1710250757/Pressemeldinger/Pressemelding_Her%20er%20datoen%20Mobillett-appen%20slutter%20a%CC%8A%20virke.pdf

AtB. (2025). Atb-appen. https://play.google.com/store/apps/details?id=no.mittatb.store&hl=en

Firebase, G. (2025). Manage firebase installations. https://firebase.google.com/docs/projects/manage-installations?

Kitsaki, T.-I., Angelogianni, A., Ntantogian, C., & Xenakis, C. (2018). A forensic investigation of android mobile applications. *Proceedings of the 22nd Pan-Hellenic Conference on Informatics*, 58–63. https://doi.org/10.1145/3291533.3291573

Maryam, U., & Hussain, M. (2023). A study on the forensic analysis of airlines applications on android operating system. *2023 IEEE Conference on Dependable and Secure Computing (DSC)*, 1–9. https://doi.org/10.1109/DSC61021.2023.10354207

Rønning, E. S. (2025). Fikk bot for manglende bussbillett - nå slipper hun å betale. to sekunder var det som avgjorde at hun fikk bot i utgangspunktet. https://www.adressa.no/nyheter/trondheim/i/Xjxgdm/fikk-bot-for-manglende-bussbillett-naa-slipper-hun-aa-betale

Sandvik, J.-P. (2017). Mobile and embedded forensics. In *Digital forensics* (pp. 191–273). John Wiley Sons, Ltd. https://doi.org/https://doi.org/10.1002/9781119262442.ch6

Sanz Maroto, J., Liu, H., & Patras, P. (2020). On the struggle bus: A detailed security analysis of the m-tickets app. In W. Susilo, R. H. Deng, F. Guo, Y. Li & R. Intan (Eds.), *Information security* (pp. 234–252). Springer International Publishing.