July 2024

Design of a Satellite Tracking Mobile App Using Flutter and Android Studio IDE

Audrey Tahwa
University of South Florida

Scholar Commons Citation

Tahwa, Audrey, "Design of a Satellite Tracking Mobile App Using Flutter and Android Studio IDE" (2024). *USF Tampa Graduate Theses and Dissertations.*
https://digitalcommons.usf.edu/etd/10567

Design of a Satellite Tracking Mobile App Using Flutter and Android Studio IDE

by

Audrey Tahwa

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science
Department of Electrical Engineering
College of Engineering
University of South Florida

Co-Major Professor: Robert H. Bishop, Ph.D.
Co-Major Professor: Wilfrido A. Moreno, Ph.D.
Ismail Uysal, Ph.D.

Date of Approval:
July 2, 2024

Keywords: Azimuth, Drag, Eccentricity, Epoch, Perturbations

Dedication

This thesis is dedicated to God Almighty, to my friends, who have offered unwavering support and motivation throughout the master's program. I'm grateful beyond words and to all my professors who supervised.

Acknowledgments

I would like to express my sincere gratitude to Dr. Robert Bishop and Dr. Wilfrido Moreno for their supervision, assistance, and guidance throughout this research study. Their invaluable instructions and support have been pivotal in this academic journey. I would also like to extend gratitude to my family for their support and encouragement.

Table of Contents

List of Tables

List of Figures

Abstract

A pop-up free mobile application was developed utilizing one-minute intervals for position tracking. Predictions are conducted off-line after the user has connected at least once. In order to enhance data accuracy, the application includes a feature to download satellite data and update raw data to be used in prediction calculations. Additionally, users receive a notification when the application is ready for offline use.

Chapter 1: Introduction

Among the four emergency communication options identified, satellite communication stands out as critical. The first one is the widespread public communication network that was used by the general public. The second one is personal networks established primarily for social welfare or governmental purposes, often causing interference with other communication channels. The third one is satellite communication accessible only through specialized terminals due to its extensive coverage and limited system capacity for its transfer bandwidth. The fourth one is shortwave communication, which has a strong endurance component despite its poor bandwidth performance and network dependability challenges.Therefore, advancing satellite communication accessibility to the general public is critical.The aim of this thesis is to develop an application for satellite tracking that operates without internet connectivity as an initial step towards facilitating widespread access to satellite communication.

1.1 Background

The mobile sector is seeing tremendous technical breakthroughs, and the increasing accessibility of devices is transforming mobile communication and satellite tracking, profoundly impacting human society. A key advantage of mobile technology is its mobility, enabling users to interact and communication with one another, remotely access data, and conduct business from any location, at any time. Satellites play a critical role in telecommunications, supporting mobile applications such as radio and TV broadcasting, as well as connectivity with hand-held terminals, vehicles, aircraft, and ships. Additionally, satellites are instrumental in global mobile communication and weather forecasting.

1.1.1 Fundamental Methods for Satellite Tracking

Satellite tracking techniques have been developed mainly in three different fields. In the first, satellites in sub-synchronous orbits are tracked from a ground-based station. In this case, the satellite is only intermittently visible from a given point on Earth's surface and traverses the observer's sky within specified orbital periods. Even with moderately wide-beam antennas, a stationary base station would quickly lose contact resulting in degraded communication unless a certain tracking system is used.

The second scenario involves monitoring geostationary satellites from ground stations. These satellites have limited movement within a small orbital window, typically not surpassing k3. Antennas with beam widths of this scale will probably not entail active tracking and can remain fixed, albeit periodic adjustments by a human operator may be required to allow for satellite drift. However, narrower beam width antennas would still require active tracking. The third field concerns inter-satellite links, where tracking techniques are based on accuracy, weight, and reliability.These factors impose specific constraints and requirements on tracking techniques.

1.1.2 Importance of Satellite Communication and Tracking

A vast array of applications, including vital national security operations, internet access, agriculture, construction, mining, power grids, postal services, humanitarian aid, weather forecasting, and much more, rely on data from GPS satellites, which also provide data for GPS-enabled maps, thus simplifying navigation for users worldwide.

1.2 Problem Statement

Properly setting the angle and height of a satellite dish is critical to achieve optimal positioning, hence ensuring users will obtain a clear receiving signal. To ensure that users receive consistent coverage, a broadcast satellite dish is always positioned in a fixed position for household installations. However, for mobile platforms such as air, land, and sea transportation, the challenge of satellite dish alignment on a movable platform is striking. Determining the dish's bearing and Azimuth can be time-consuming, especially under unfavorable conditions.

In the case of a moving boat it would be difficult to watch TV while the boat is moving with a fixed-pointing satellite dish because the boat is constantly shifting its direction in relation to the satellite. It has always been crucial to have crystal-clear television reception on mobile platforms. For a marine vessel to access local channels, tracking satellite signals is a requirement. Ship motions, which are partly caused by waves, will make the antenna point away from the satellite, potentially disrupting the signal.

When the signal is suppressed due to a change in the environment or a physical barrier separating the antenna and satellite, a problem also occurs. As a result, the control loop is fed with inaccurate data, which causes tracking functionality to be lost.

A land-based satellite tracking antenna's position needs to be adjusted according to the ship's current location. Once the antenna is fixed to the proper orientation in a static condition, no additional adjustments are needed because the satellite is normally synchronized with Earth's motion. It is more difficult to control the tracking of a ship's antenna because the vessel is constantly traveling. The motors and drivers needed to move the satellite dish on the movable platform are also a component of the issue that requires investigation into the system's kinematics and tracking control. Therefore, choosing the right kind of motors is an essential component of the design process. An electronic motor is suited to support the load due to the weight of a satellite tracker's entire system.

In order to accurately align the satellite dish with the correct satellite, some details need to be calculated. For a satellite tracking system, two details need to be identified. The required heading for the dish will be calculated by the Azimuth orientation. Azimuth orientation determines the required heading for the dish, ensuring it is aligned parallel to a specific point on the satellite transmitting the signal. For instance, satellites A, B, or D are located at 28.2° East, corresponds to the ship's current True South position. The dish needs to be angled above the reference heading in order to align it with the satellite it is intended to face. The term used to describe this is the elevation angle. Accurate determination of elevation areas is crucial due to the curvature of the Earth[13].

The primary challenge posed by antenna positioning is directing it aligning it with the correct satellite for the required transmissions to be received. Although satellite receivers contain some information about the satellites on them, they are unable to confirm correct orientation towards the target satellite. The dish needs to be correctly positioned by the user. Moving the antenna requires the satellite trackers control system to function properly.

Several mobile applications have been developed for satellite tracking, such as Satorbit and Heavens above. However, Satorbit presents some limitations, namely, that it tends to hang on GPS location updates without progressing further. Additionally, users can only select one agency at a time to download. Moreover, data can only be downloaded from sites once every six hours. Despite updating the information, the system still indicates that it is outdated. The problem is that the International Space Station (ISS) does not show in the ISS section but it is seen on the radio satellite at the same time. Besides, the free version contains numerous pop-up advertisements.

1.3 Purpose

To design a mobile application for notifications and satellite tracking.

1.4 Objectives

1. Develop an application that allows users to search for particular satellites
2. Design an application that enables users to view downloaded satellite data offline
3. Develop an application free of pop-up advertisements

1.5 How Satellite Communication Has Made the World a Better Place

Satellites have significantly contributed to making the world a better place in various ways:

1. Communication: Satellites enable global communication, making it easier for people around the world to connect through phone calls, internet, and television. This has fostered improved collaboration, knowledge sharing, and access to information.
2. Navigation: With the ability to provide precise location data, satellite-based navigation systems like the Global Positioning System (GPS) have transformed navigation. This has enhanced transportation efficiency, reduced travel time, and improved safety in various industries.
3. Weather Forecasting: Satellites play a crucial role in monitoring and collecting data about the Earth's atmosphere, oceans, and climate. This information is essential for accurate weather forecasting, helping communities prepare for natural disasters and mitigate their impact.
4. Disaster Management: Satellites aid in disaster management by providing real-time imagery and data during emergencies. This helps authorities respond more effectively to natural disasters, such as hurricanes, earthquakes, or wildfires, improving disaster relief efforts.
5. Environmental Monitoring: Satellites contribute to monitoring environmental changes, including deforestation, pollution, and climate change. This information is valuable for policymakers and scientists working towards sustainable environmental practices.
6. Agriculture: Satellite technology assists in precision agriculture by providing data on crop health, soil conditions, and weather patterns. This helps farmers optimize crop yields, reduce resource usage, and make informed decisions.

7. Internet Connectivity: Satellites play a role in providing internet access to remote and under-served areas where traditional infrastructure is challenging to establish. This helps bridge the digital divide, offering educational and economic opportunities to more people.
8. Scientific Research: Satellites enable scientific exploration and research beyond Earth, contributing to understanding the universe, climate patterns, and space. They provide crucial data for various fields, ranging from astronomy to environmental science.
9. In summary, satellites have made a positive impact on global communication, navigation, disaster management, environmental monitoring, agriculture, internet connectivity, and scientific research, collectively contributing to an improved and interconnected world.

1.6 Materials and Methods

While conducting the research, the following procedures were be carried out:

- Utilizing an android emulator and Flutter for modeling and simulation.
- Performing predictions using the SGP4 mathematical model written in Python.
- Installing and testing the application.
- Thoroughly reviewing literature on existing satellite tracking and communication technologies and architectures.

The steps to the design and modeling process are displayed below:

1. The tracking of satellites, sending of signals, choice of technology and architecture.
2. Selecting individual components.
3. Developing satellite tracking and mobile application front-end using Flutter and Python (Django) for back-end, with ability to send notifications.

4. Using SGP4 mathematical model written in Python for predicting satellite position prediction.
5. Building android or iOS apk using Flutter.
6. Installing the application for android or iOS on a mobile phone.
7. Testing and deploying on the Play Store.

Chapter 2: Literature Review

A comprehensive review of satellite tracking and communication was completed. Firstly, software development platforms used in application development were analyzed considering mobile applications developed using these platforms and the strengths and weaknesses of these development platforms. Secondly, previous work on satellite tracking and communication systems was looked into. Finally, android applications and other methods for satellite tracking and communication were also examined [13].

2.1 Software Selection Criteria

2.1.1 Flutter

Flutter application is used in this project to develop a satellite tracking application.

2.1.2 The Historical Background of Flutter

In 2017, Google released Flutter, an open-source toolkit for developing user interface software. This toolkit enables the creation of natively compiled desktop, web, and mobile applications from a single code base. Flutter applications are created with Dart, a Google-developed programming language. Flutter is well-known for its expressive and adaptable user interface, but its hot reload feature which enables real-time code modifications has made it more well liked among developers. As of January 2022, Flutter is still developing thanks to updates and contributions from the developer community. However, its usage can be challenging at times, as some features are experimental and have not been widely tested.

---

[13] Max H Lane and Felix R Hoots. General perturbations theories derived from the 1965 lane drag theory, 1979

2.1.3 Popular Applications Developed with Flutter

1. Google's advertising platform utilizes Flutter for its mobile app, providing a consistent user experience across different platforms.
2. Alibaba a leading e-commerce platform, used Flutter to build parts of its mobile app, enhancing the app's performance and user interface.
3. Nubank a financial technology company, utilized Flutter to create a consistent and visually appealing interface for its mobile banking app.
4. The Dream11 fantasy sports platform has incorporated Flutter to develop its mobile app, offering a smooth and engaging user experience.

2.1.4 Benefits of Utilizing Flutter

There are several compelling reasons for choosing Flutter for mobile app development:

1. Single Code-base: Flutter cuts down on development time and effort by allowing developers to build a single codebase for both the iOS and Android operating systems. This guarantees compatibility with many operating systems.
2. Hot Reload: This reload feature speeds up development and debugging processes by enabling developers to modify the code in real-time and see the results right immediately.
3. Expressive UI: With Flutter's extensive collection of configurable widgets, developers can design aesthetically pleasing, highly interactive user interfaces with a native feel.
4. Performance: Flutter ensures fluid and responsive applications by using the Dart programming language and compiling it to native ARM code. This results in great performance.

5. Community and Ecosystem: Flutter boasts a thriving ecosystem of packages, plugins, and resources, which is fueled by an ever expanding and dynamic community of developers. This kind of community support might be helpful for exchanging knowledge and solving problems..

6. Versatility: Flutter is not limited to mobile app development; it can be used for building applications for web and desktop as well. This versatility allows developers to leverage their skills across different platforms.

7. Cost-Effective Development: With a single code-base and faster development cycles, Flutter can result in cost savings for businesses compared to developing separate code-bases for iOS and Android.

8. Support from Google: As an open-source framework developed by Google, Flutter benefits from continuous improvements, updates, and support from a major technology company, adding to its credibility and long-term viability.

9. Adoption by Major Companies: Many well-known companies, such as Google, Alibaba, and Nubank, have successfully adopted Flutter for their applications, showcasing its capabilities in diverse industries.

10. Growing Popularity: The popularity of Flutter continues to rise, making it a sought-after skill among developers and contributing to a robust job market for Flutter expertise.

Overall, the combination of a single code-base, hot reload, expressive UI, performance, community support, and versatility makes Flutter an attractive choice for developers and businesses seeking efficient and cross-platform mobile app development.

2.1.5 Android Studio

The integrated development environment, or IDE, is another name for the Android Studio, users can create an Android app using the features in this IDE, which functions as a workshop. They can also examine a real-time design assessment of the app in an Android Studio. Additionally, the Android Studio evaluates a code entry for mistakes and helps fixing them by making the appropriate suggestions.

2.1.6 The Historical Background of Android Studio

Android Studio, introduced in 2013, replaced Eclipse as the official IDE for Android development. Over the years, it evolved with key milestones such as integrating the Gradle build system, introducing Instant Run in 2016, supporting Kotlin in 2017, and launching Android Studio 3.0 with enhanced profiling tools. Subsequent versions brought features like Constraint Layout, improved support for Android Oreo, and integration with modern development paradigms like Jetpack Compose. The IDE continues to be a vital tool for developers, given its adaptability to the changing landscape of Android app development.

2.1.7 Benefits of Utilizing Android Studio

Developed by Google exclusively for Android app development, Android Studio is the official Integrated Development Environment (IDE) and offers several advantages:

1. Rich feature set: A strong code editor, a visual layout editor, and sophisticated debugging tools are just a few of the many tools and capabilities that Android Studio offers specifically for Android development.

2. Gradle Integration: Android Studio easily interfaces with the Gradle build system, facilitating effective management of dependencies, projects, and build configuration.

3. Support for Emulators: Android Studio includes a feature-rich emulator that lets developers test their apps across a range of Android devices with varying setups, screen sizes, and Android versions.

4. User Interface (UI) Designer: The visual layout editor in Android Studio enables developers to design and preview their app's UI in a What You See Is What You Get (WYSIWYG) manner, making it easier to create and customize user interfaces.

5. Intelligent Code Editor: To improve efficiency and lower the number of coding errors, Android Studio comes with an intelligent code editor that includes tools like code completion, code analysis, and rapid repairs.

6. Real-time Profiling and Debugging: Developers can analyze their apps performance in real-time using Android Studio's profiling tools. The IDE also provides robust debugging capabilities, including breakpoints, watches, and variable inspection.

7. Version Control Integration: Git and other widely used version control systems are supported by Android Studio, which makes it simple for developers to work together and manage projects utilizing these platforms.

8. Extensive Plugin Ecosystem: Android Studio has a large plugin and extension ecosystem that enables developers to modify and expand the IDE's features based on functionality tailored to their specific needs.

9. Official Support: Being the official IDE for Android development, Android Studio receives continuous updates and support directly from Google, ensuring compatibility with the latest Android SDKs and features.

10. Community and Documentation: A sizable and vibrant developer community supports Android Studio. Developers can find answers to common problems more easily thanks to the abundance of internet tools, tutorials, and documentation that are available.

In conclusion, developers working on Android applications prefer Android Studio because it provides a stable and feature-rich environment for Android app development.

2.1.8 Drawbacks of Utilizing Android Studio

Android does have some potential disadvantages:

1. Resource Intensive: Utilising a lot of resources, such as a strong computer with enough RAM and processing power, is necessary when using Android Studio. Slower performance on less powerful devices could result from this.

2. Steep Learning Curve: For beginners, Android Studio may have a steep learning curve, especially for those who are new to mobile app development or programming in general. The variety of tools and features may seem overwhelming at first.

3. Gradle Build Times: The Gradle build system used by Android Studio can sometimes result in longer build times, especially for larger projects. Developers may experience delays during the build and compilation processes.

4. Emulator Performance: The built-in emulator in Android Studio, while powerful, can be slow, particularly on machines lacking hardware acceleration support. Developers may find it more efficient to test on physical devices or use third-party emulators.

5. Occasional Bugs and Issues: As with any complex software, Android Studio is not immune to bugs and occasional issues. Despite regular updates and patches, users may still face challenges that require troubleshooting.

6. Limited Support for Other Platforms: While Android Studio is designed specifically for Android development, it may not be the optimal choice for projects targeting multiple platforms (iOS, web, etc.). Cross-platform development tools might be more suitable in such cases.

7. Large Disk Space Requirement: Android Studio and associated tools can consume a significant amount of disk space, especially with the installation of various SDKs, emulators, and dependencies. This can be a concern on systems with restricted capacity.
8. Dependency Management Challenges: Managing dependencies and library versions, particularly with the Gradle build system, can sometimes lead to compatibility issues and require careful configuration.
9. IntelliJ IDEA Licensing: Android Studio is built on the IntelliJ IDEA platform, and some advanced features of IntelliJ IDEA may require a separate license for developers to use them within Android Studio.
10. Updates and Changes: The frequent updates and changes in Android Studio may pose challenges for projects that need to maintain compatibility with specific versions. This can be especially relevant when using third-party libraries or plugins.

Despite these potential drawbacks, Android Studio remains the primary and recommended IDE for Android app development, and many developers find its advantages outweigh these challenges. Regular updates and improvements continue to address some of these issues over time.

2.1.9 Python (Flask) as Back-end

Web applications are developed using the widely used Flask micro web framework for Python. It offers a flexible user-friendly method for developing Python-based web applications and APIs (Application Programming Interfaces). Flask is well known for its straightforward design, which allows developers to choose the components they prefer and tailor their projects to the requirements. Uses of Python flask are:

1. Web apps: Numerous web applications, including blogs, e-commerce websites, and social media platforms, can be made with Flask. Because of its simplicity and versatility, it may be used for both modest tasks and bigger, more complex applications.

2. API development: Create RESTful APIs to help different software systems communicate with one another. Use these APIs to perform activities, integrate various services, and communicate data.

3. Prototyping: Flask's lightweight design makes it a great choice for quickly prototyping web-based ideas and concepts. Without the burden of more intricate frameworks, it enables developers to quickly produce and test their ideas.

4. Micro-services: Building micro-services is an excellent use case for Flask. Micro-services are small, independently deploy-able parts of bigger apps. Each micro-service can be built by developers using Flask to provide unique capabilities.

5. Webhooks: They work by getting information from outside sources when specific events take place. Flask endpoints can be created to listen for these events and trigger events within the application.

6. Interactive dashboards: Real-time data visualization is possible with interactive dashboards created with Flask, something which is helpful for data analysis, reporting, and monitoring purposes.

7. Educational projects: Flask is a well-liked option for teaching web development techniques in educational settings because of its simplicity and organization. Routes, templates, and front-end/back-end interactions can be swiftly learned by students.

8. Small to medium websites: Flask offers a feature-rich, lightweight substitute for larger frameworks for websites that do not need their complexity.

9. Integration with Data Science and Machine Learning: Flask makes it easier to create web interfaces for data science and machine learning models. This allows users to interact with and use the models without having to understand the underlying code.

2.1.10 Benefits of Utilizing Python

- Code readability and maintainability
- Supports multiple programming paradigms
- Extensive standard library
- High compatibility
- Simplified software development
- Multiple Open-source frameworks and tools
- Test-driven development
- Easy to read and learn

2.2 Satellites

In the context of Earth, satellites are objects that orbit celestial bodies. They can be either man-made objects, or they can be natural objects like the Moon. Man-made satellites are intentionally launched into orbit for various purposes.

1. Orbit Types:
    (a) Low Earth Orbit (LEO): Utilized for close-range communication and earth observation.
    (b) Medium Earth Orbit (MEO): Satellites that regularly use this intermediate orbit for navigation.
    (c) Geostationary Orbit (GEO): A permanent orbit that weather and communication satellites usually occupy above a specific place on Earth.

2. Functions:
    - Communication: Satellites relay signals for television, radio, internet, and telecommunications globally.
    - Earth Observation: Monitor and collect data on Earth's surface, atmosphere, and oceans.
    - Navigation: Provide precise positioning information through systems like GPS.
    - Scientific Research: Study celestial bodies, cosmic phenomena, and space exploration.
    - Military Applications: Used for reconnaissance, surveillance, and secure communication.
    - Weather Monitoring: Track and predict weather patterns.
3. Launch and Design:
    - Satellites are launched into space using rockets.
    - They can be equipped with various instruments, antennas, and sensors depending on their intended purpose.

Overall, satellites contribute significantly to telecommunications, scientific exploration, weather monitoring, and various other aspects of modern life.

2.2.1 Satellite Tracking

Satellite tracking refers to the process of monitoring the location or flight path of an orbiting object. Tracking is applied to visual evaluation, active or passive radio communication, or just following the satellite's exact position and ground track. Most satellites emit telemetry signals or use beacon receivers to transmit their ID and location to Earth. The position of the satellite in space can be determined by tracking the location of this signal.

Due to the vast distances involved, any positional changes are imperceptible to casual observers. Specialized equipment, such as the Beacon Tracking Receiver, is employed for this purpose.

All artificial objects in earth orbit must be tracked by NORAD, which does so by utilizing the Space Surveillance Network (SSN). This network consists of radar sensors to detect relatively local objects below approximately 6,000 kilometers in altitude and optical (technically, electro-optical) sensors to follow deep-space objects above 6,000 kilometers in height. These sensors are strategically dispersed around the globe to provide comprehensive coverage. While optical sensors can only determine angles (either Azimuth and elevation or right ascension and declination), a typical radar observation can provide Azimuth, elevation, range, and range rate. Every observation is time-stamped upon collection. The data is first examined by NORAD to determine whether it corresponds to the incorrect satellite or points to the track of a previously untracked satellite.

2.2.2 Simplified Perturbations Models (SPG)

The mathematical models, known as simplified perturbations models (SGP, SGP4, SDP4, SGP8, and SDP8), are used to determine the orbital state vectors of satellites and space debris with respect to the inertial coordinate system centered on Earth. Given the widespread use of these models, especially with the two-line element sets that are developed by NASA and NORAD, they are sometimes referred to as SGP4. These models predict the impact of perturbations brought caused by radiation, drag, the Earth's curvature, and the gravitational pull of other bodies, like the sun and moon. Simplified General Perturbations (SGP) models apply to near-Earth objects with orbital periods shorter than 225 minutes. Based on simplified deep space perturbations, the SDP models can be applied to objects whose orbital periods exceed 225 minutes. Simplified Deep Space Perturbations (SDP) models are relevant for objects with an orbital period greater than 225 minutes, i.e., an altitude of 5,877.5 km, assuming a circular orbit.

First released in 1988 with sample code in FORTRAN IV, the SGP4 and SDP4 models have now been refined to account for the larger number of objects in orbit. Additional orbital decay handling improvements were introduced with SGP8/SDP8.

The SGP4 model grows at a rate of 1-3 km per day and has an inaccuracy of 1 km at epoch. Consequently, NASA and NORAD frequently update this data. The original SGP model, developed by Kozai in 1959 and refined by Hilton & Kuhlman in 1966, was used by the National Space Surveillance Control Center and later by the United States Space Surveillance Network, to track objects in orbit. The inaccuracy of the SDP4 model at epoch is 10 kilometers.

SDP4 and SDP8 solely employ ”simplified drag” equations when implementing deep space. Accuracy is not a large issue in high drag satellite settings since the orbit gradually decreases and approaches a circle, meaning that the satellites do not dwell in ”deep space” for very long. SDP4 also perturbs all orbits with lunar-solar gravity and adds Earth resonance components to the geostationary and 12-hour Molniya orbits. More iterations of the model were developed and published by 2010 by the NASA Goddard Space Flight Centre and the Jet Propulsion Laboratory's Navigation and Ancillary Information Facility to support the tracking of the SeaWiFS mission and the Planetary Data System for a variety of missions, most of which were deep space missions. Libraries of current code [11, **?**] use SGP4 and SDP4 algorithms, which were integrated into a single code base in 1990, to handle the range of orbital periods that are generally referred to as SGP4 generically.

2.2.3 Differential Correction Process

The differential correction process begins with identifying the TLE set and the required orbital model (SGP4), alongside a set of time-tagged/stamped observations. The need for updates can be determined by comparing the existing two-line element set with observations. The SGP4 orbital model using the current two-line element set is used to estimate the positions and velocities of satellites. The anticipated observations for the specified sensors' observing geometry are then computed using these values. It is now possible to compute the difference between the observed and expected data. As it might anticipated, the goal is to decrease this fluctuation (the total). To help choose which elements to alter or how to lessen the difference, the transformation function will provide the following computations for the element to observation transition.

$$y = f(x) \tag{2.1}$$

y is the dependent variable and x is the independent variable, yaw the expected observation and $yp = f(xp)$ the actual observation be. The derivative of transformation at xp can be generally represented as, according to calculus,

$$\frac{X_a - X_p}{Y_a - Y_p} = \frac{X_a - X_p}{Y_a - f\left(X_p\right)} = \frac{dx}{dy} \tag{2.2}$$

Solving the element to get its "actual" value yields dx

$$X_a = X_p + \left(Y_a\right) - f\left(X_p\right)) \cdot \frac{dx}{df(x)}\bigg|\ x = X_p \tag{2.3}$$

Finding a function's root is accomplished using this basic Newton iteration. For the method to work correctly, the element's initial value must be "close" to its real value. Every time a new "actual" value is generated, the process repeats itself, substituting the anticipated value with the "actual" value until the difference between the real and predicted elements in subsequent calculations are minimal. This iterative process aims to compute the derivative of the observation with respect to the element, or the rate at which the observation changes in response to a change in the element. The transformation function processes data from a two-line element set using the SGP4 orbital model, and then transforms them again for the coordinate system and sensor observing geometry. The same approximation used earlier is then used to numerically estimate the derivative with the element's initial value and introducing a small variation to generate.

$$\frac{df(x)}{dx} = \frac{f(x+\Delta x) - f(x)}{(x+\Delta x) - x} = \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{2.4}$$

2.2.4 Calculating Range, Elevation and Azimuth

A satellite's range, elevation, and Azimuth can only be determined by knowing its position with respect to Earth, or the position of an observer on Earth with respect to the ECI coordinate frame.

The Earth is shown in this artwork as a side cutaway with North pointing upwards. Figure 2.5 shows the z coordinate for an observer at latitude, where Re is the planet's equatorial radius. To find the x and y coordinates, the value of R from Figure 2.5 should be determined. To find z and R at elevations above mean sea level, just substitute Re for $Re + h$, where h is the height above mean sea level.

$$Z = R_\phi \sin \phi \tag{2.5}$$

$$R = R_\phi \cos \phi \tag{2.6}$$

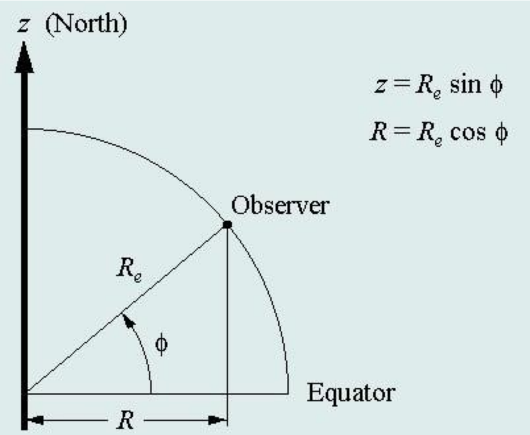


Figure 2.1: Conversion of Latitude to Earth-centered Inertial (ECI) Coordinates Taken from Detailed Description of "Two-Line Element (TLE)" Orbital Parameters by Wiley Brothers

It takes longer to calculate the x and y coordinates. The Earth revolves around the z-axis, or in the $x - y$ plane, so the x and y coordinates of a position on the surface vary with time, while the z coordinate remains constant. In contrast, the x and y coordinates are defined as a function of time by simply calculating the angle between the observer's longitude and the x-axis thanks to the vernal equinox. If the angle between the x-axis and the observer's longitude is specified as theta(tau), where is the time of interest, Figure 2.5 actually displays the values of x(tau) and y(tau). The equatorial plane and the observer's position are both cut through by the Earth, which is depicted in this image as a horizontal slice. Theta(tau) is referred to by astronomers as having a "local sidereal time" signature. The term "sidereal time" refers to time expressed in terms of the stars.

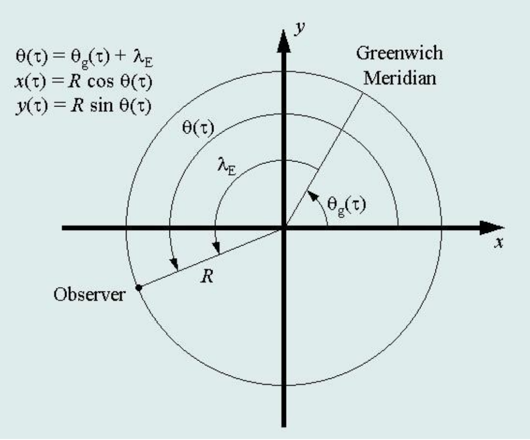


Figure 2.2: Longitude to ECI Conversion Adopted from Detailed Description of "Two-Line Element (TLE)" Orbital Parameters by Wiley Brothers.

$$\theta(\tau) = \theta_g(\tau) + \lambda E \tag{2.7}$$

$$X(\tau) = R \cos \tau \tag{2.8}$$

$$Y(\tau) = R \sin \tau \tag{2.9}$$

The Greenwich Sidereal Time (GST), $\Pi g(\tau)$, is multiplied by the observer's east longitude (λE) to give the local sidereal time. Most of the time, astronomical almanacks such as the US Naval Observatory's Astronomical Almanack cite GST (or, more precisely, GMST). $\theta_g(\Delta\tau) = \theta_g(0^h), +\omega e\Delta\tau$, where $\Delta\tau$ is the UTC time of interest and ωe is the Earth's rotation rate, is the result if GMST is known for 0h UTC hours on a specific day, or $\theta_g(0^h)$. Unfortunately, utilising this method to perform the computations requires a table of reference times. A different approach is to compute θ_g.

$$\theta_g(0^h) = 24110^s.54841 + 8640184^S.812866\tau_u + 0s.093104\tau_u^2 - 6.210 - 6\tau_u^3 \tag{2.10}$$

where du is the days of Universal Time that have passed since JD 2451545.0 (2000 January 1, 12h UT1) and $\tau_u = \frac{du}{36525}$

To determine the satellite's position in the ECI coordinate system, the following calculation is employed $[x_o, y_o, z_o]$ and $[x_s, y_s, z_s]$ if the observer is:

$$[r_x, r_y, r_z] = [x_s - x_o, y_s - y_o, z_s - z_o] \tag{2.11}$$

Rotate by an angle θ (the local sidereal time) about the z-axis (the Earth's rotation axis) and then by an angle phi (the observer's latitude) about the y-axis to approach the topo centric-horizon system. These are the new locations for r_S, r_E, and r_Z:

$$r_S = \sin\phi\cos\theta r_x + \sin\phi\sin\theta r_y - \cos\phi r_z \tag{2.12}$$

$$r_E = -\sin\theta r_x + \cos\theta r_y \tag{2.13}$$

$$r_Z = \cos\phi\cos\theta r_x + \cos\phi\sin\theta r_y + \sin\phi r_z \tag{2.14}$$

The satellite's range is:

$$r = \sqrt{r_S^2 + r_E^2 + r_Z^2} \tag{2.15}$$

also the elevation can be obtained by:

$$El = \sin^{-1}\left(\frac{r_Z}{r}\right) \tag{2.16}$$

Moreover, Azimuth is:

$$A_z = \tan^{-1}\left(\frac{r_E}{r_S}\right) \tag{2.17}$$

Instead of the usual counterclockwise movement from South in a right-handed orthogonal coordinate system, Azimuth measurement requires clockwise movement from North, indicated by the use of a negative sign. Caution should be exercised when using the Azimuth to ensure the arctangent is in the right quadrant. This vector is conveniently integrated into the ECI system. The topo centric-horizon system depicted in Figure 2.7 is the necessary place to be in order to create look angles. The x, y, and z axes in that configuration point east, south, and toward the zenith, respectively.

2.2.5 Two Line Elements

A two-line element set (TLE) is a data format that encodes the orbital elements of an Earth-orbiting object at a specific moment, known as the epoch. TLE condenses a satellite orbital characteristics into two lines of data containing 69 alphanumeric characters at most. Line 0 is followed by two lines, line 1 and line 2, which also consist of has sixty-nine characters each, but only 24 of that can be used to provide more information about the space object. Despite what its name might imply, despite its name, the TLE is made up of three lines, each of which can contain up to 69 alphanumeric characters.

Table 2.1: Comments for Line 0 Reprinted from Jean-Luc Lefebvre.

NUMBER SEQUENCE	COLUMN	CONTENT	FORMAT	COMMENT
1	0 - 24	NOAA6	AAAAAAAA AAAAAAAA AAAAAAAA	24 Alphanumeric characters to identify the chosen space object, such as NOAA6.
2	26 - 28		b.b	The length is measured in meters when the height and width are an estimated diameter is used for objects with unknown dimension.
3	30		c.c	Width is measured in meter. If the height and width are 0. Objects with unknown dim with an estimated diameter.
4	34 - 36		d.d	Height is measured in meters if the height is 0, the object with unknown dimensions with an estimated length.
5	38 - 40		e.e	Guideline for magnitude (seen from 1000km and illuminated at 50 percent).
6	42		F	The standard method for computing magnitude, where "v" stands for visual observation and "d" for dim computation.
7	44 - 46		RRR	Corresponding section is in square meters.
8	48 - 55		KM KM KM KM KM. KM KM	Apogee kilometers, decimal places altitude up to in two.
9	57 - 64		km km km km km km km	Perigee kilometers, decimal places altitude up to in two.

Table 2.2: Comments for Line 1 Reprinted from Jean-Luc Lefebvre.

NUMBER SEQUENCE	COLUMN	CONTENT	FORMAT	COMMENT
1	O1	1	1	The initial number "1" shows the line number for the TLE's first line.
2	03 - 07	11416	ggggg	NORAD catalog number.
3	08	U	X	Classification done here. "U" stands for "Unclassified" = non secret information.
4	10 - 11	84	Hh	2 Final digits before launch (1984).
4	12 - 14	123	Iii	The launch year's number (123rd)
4	15 - 17	A	AAA	"A Launch component number (important if there are numerous launches). Here, ""A"" denotes that it was the initial satellite to be launched.
5	19 - 20	86	Kk	2 last digits of the year at which this information was determined.
5	21 - 32	50.28438588	jjj-jjjjjjjj	The day and portion of a day of the calendar year on which these components were determined.
6	34 - 43	0.00000140	+/- m m m m m m m m	Half of the first time derivative of the mean motion. The value shown represents the change in the satellite's velocity.
7	45 - 52	00000-0	+/- n n n n n – n	The 2nd time derivatives of the average motion divided by one-sixth.
8	54 - 61	67960-4	+/- o o o o o o – o	SGP4 circular propagator's usage of a pseudo-ballistic factor.
9	63	0	P	Type of ephemeris used.
10	65 - 68	5,293	Q	Every time a new TLE is created for this space object, this integer increases slightly.

Table 2.3: Comments for Line 2 Reprinted from Jean-Luc Lefebvre.

NUMBER SEQUENCE	COLUMN	CONTENT	FORMAT	COMMENT
1	01	2	2	The initial number ”2” represents the second TLE’s second number.
2	03 - 07	11416	Ggggg	NORAD catalog number.
3	09 - 16	98.5105	sss.ssss	orbital inclination (degrees).
4	18 - 25	69.3305	ttt.tttt	The orbital ascending node’s right ascension.
5	27 - 33	0012788	uuuuuuu	Eccentricity (no decimal place).
6	35 - 42	63.2828	vvv.vvvv	Argument of the perigee (degrees).
7	44 - 51	296.9658	www.wwww	Mean anomaly (degrees).
8	53 - 63	14.24899292	xx.xxxxxxxx	The mean motion number (the total number of revolutions each day)).
9	64 - 68	34697	yyyyy	The number of revolutions during the time the information were established.
10	69		Z	Checksum (module10) [This data is not available right now.

With the appropriate prediction formula, it is possible to estimate the state velocity and position of any point in the past or future with a certain degree of accuracy. The TLE data format is restricted to the simplified perturbations models (SGP, SGP4, SDP4, SGP8, and SDP8). Therefore, any method that uses a TLE as input to calculate the state at a specific time must utilize one of these models. The orbital trajectories that Transient Lunar Events (TLEs) can follow in Earth’s orbit are limited. Estimating the orbital paths of space debris to be used in assessing risks, analyzing close approaches, conducting forensic investigations, and characterizing future debris events.

During the early 1960s, Max Lane created mathematical models to compute the positions of satellites using a limited amount of data sources. The inaugural publication on the topic, titled ”The Analytical Drag Theory,” was made available in 1965. The study focused on the drag effects that occur in a non-rotating, spherically symmetric environment. In 1969, K. Cranford joined them in creating an enhanced model that incorporated additional harmonic effects resulting from the interactions between the Earth, Moon, and Sun, along with various other inputs.

NASA and the military began extensively using Lane's models extensively in the late 1960s. When NORAD adopted the updated model as standard operating procedure in the early 1970s, the TLE format was eventually established. There were two types of punch cards at the time: a two-card ”transmission format” that contained only the parts that could alter and a three-card ”internal format” that required three cards to encode all of the satellite's information, including its name and other details. When updating the databases, the latter used fewer cards and created smaller decks [13].

Cranford continued developing the models, culminating in, Lane's Spacetrack Report 2, which presented the Air Force General Perturbation theory, or AFGP4. The study also described two standardized variants: Simplified General Perturbations (IGP4), which utilized a streamlined drag model, and Simplified General Perturbations (SGP4), which used a streamlined gravity model in addition to IGP4's drag model. As stated by Kelso et al. (2021).

For the majority of objects, the variations between the three models were minimal. The SGP4 model's complete FORTRAN source code was included in Spacetrack Report 3, which was published a year later and was quickly adopted as the industry and astronomical standard model [18].

Shortly after the publication of Report 3, NASA released the components necessary to create several recognizable and widely known objects. NASA published the hard copies of the transmission format data in their regular NASA Prediction Bulletins. T.S. Kelso attempted to get NASA to supply these in electronic format, but was unsuccessful. Consequently, he took it upon himself to manually transcribe the lists into text files. Subsequently, he utilized his CelesTrak bulletin board technology to disseminate these text files. The absence of the plus sign (+) on the teletype machines NASA used to run signaled a checksum issue, which was eventually linked to a glitch in the punch card era when NORAD switched computer systems from BCD to EBCDIC character sets. This was fixed when Kelso began receiving data from NORAD in 1989.

SDP4, which used of similar TLE data, laid the groundwork for further advancements in the SGP4 model and adaptations for deep space objects. Although many more intricate prediction models have been created over time, their adoption has been limited due to their reliance on additional TLE data, complicating efforts to locate the necessary components for improved modeling. The SGP series models subtly alter TLE data to optimize results, potentially leading other models which use common TLEs to forecast less accurately than SGP. SGP8/SDP8, which was created using the identical data inputs as the SGP4 model and is a relatively minor change of that model, is the latest adopted model.

2.2.6 SPG4 Model Equations

Using the NORAD mean element sets, SGP4 is able to generate predictions. Definitions for all symbols not included in the list of equations can be found in the list of symbols, which is located beneath the equations. The initial mean motion (n_o^n), and The semi major axis (a_o^n) is first obtained by the equations from the input elements.

SGP4 can make predictions using the NORAD mean element sets. The list of symbols below the equations contains definitions for all symbols that are not specified in the list of equations. (non), the initial mean motion, and The equations first retrieve the semi major axis (aon) from the input elements.

$$a_1 = \left(\frac{k_e}{n_0}\right)^{\frac{2}{3}} \tag{2.18}$$

$$\delta_1 = \frac{3}{2}\frac{k_2}{a_1^2}\frac{(3\cos^2 i_0 - 1)^{\frac{3}{2}}}{2a_1^2(1-e_0^2)} \tag{2.19}$$

$$a_o = a_1 \left(1 - \frac{1}{3}\delta_1 - \delta_1^2 - \frac{134}{81}\delta_1^3\right) \tag{2.20}$$

$$\delta_o = \frac{3}{2}\frac{k_2}{a_o^2}\frac{(3\cos^2 i_o - 1)^{\frac{3}{2}}}{(1 - e_o^2)} \tag{2.21}$$

$$n_o'' = \frac{n_o}{1 + \delta_o} \tag{2.22}$$

$$a_o'' = \frac{a_o}{1 + \delta_o} \tag{2.23}$$

The constant used in SGP4 for perigee in the range of 98 to 156 kilometers is modified to:

$$s^* = a_o^n(1 - e_o) - S + a_E \tag{2.24}$$

In the event that the perigee is smaller than 98 kilometers, s is changed to (2.25):

$$s^* = \frac{20}{XKMPER} + a_E \tag{2.25}$$

If s changes, $(q_o - s)^4 s$ value must be modified to

$$(q_o - s)^4 = [[(q_0 - s^*)^4]^{\frac{1}{2}} + s - s^*] \tag{2.26}$$

Then, using the correct values for s and $(q_o - s)^4$, calculate the constant:

$$\theta = \cos i_o \tag{2.27}$$

$$\varepsilon = \frac{1}{a_0^n - s} \tag{2.28}$$

$$\beta_o = \left(1 - e_o^2\right)^{\frac{1}{2}} \tag{2.29}$$

$$\eta = a_o'' e_o \varepsilon \tag{2.30}$$

$$C_2 = \left(q_o - s\right)^4 \varepsilon^4 n_0''(1-\eta^2)^{-\frac{7}{2}} \left[a_0'' \left(1 + \frac{3}{2}\eta^2 + 4e_o\eta + e_o\eta^3\right) + \left(\frac{3}{2}\frac{k_2\varepsilon}{1-\eta^2} - \frac{1}{2} + \frac{3}{2}\theta^2\right) + (8 + 24\eta^2 + 3\eta^4)\right] \tag{2.31}$$

$$C_1 = C_2 B^* \tag{2.32}$$

$$C_3 = \frac{\left(q_o - s\right)^4 \varepsilon^5 A_3, O\eta_o'' a_E \sin i_o}{k_2 e_o} \tag{2.33}$$

$$C_4 = 2\eta_0''(q_o - s)^4 \varepsilon^4 a_o'' \beta_o^2 (1 - \eta_2)^{-\frac{7}{2}} \left(\left[2\eta\left(1 + e_o\eta + \frac{1}{2}e_o + \frac{1}{2}\eta^3\right)\right] - \frac{2k_2\varepsilon}{a_o''(1-\eta^2)} \times \left[3(1-3\theta^2)\left(1 + \frac{3}{2}\eta^2 - 2e_o\eta - \frac{1}{2}e_o\eta^3\right) + \frac{3}{4}\right]\right) \tag{2.34}$$

$$C_5 = 2\left(q_o - s\right)^4 \varepsilon^4 a_o'' \beta_o^2 \left(1 - \eta_2\right)^{-\frac{7}{2}} \left[1 + \frac{11}{4}\eta(\eta + e_o) + e_o\eta^3\right] \tag{2.35}$$

$$D_2 = 4a_o''\varepsilon C_1^2 \tag{2.36}$$

$$D_3 = \frac{4}{3}a_o''\varepsilon^2\left(17a_o'' + s\right) C_1^4 \tag{2.37}$$

$$D_4 = \frac{2}{3}a_0''\varepsilon^3\left(221a_o'' + 31s\right) C_1^4 \tag{2.38}$$

The equations include the secular effects of gravitation and atmospheric drag:

$$M_{D_F} = M_o + \left[1 + \frac{3k_2\left(-1+3\theta^2\right)}{2a''^2_o{\beta_o}^3} + \frac{3k_2^2\left(13-78\theta^2+137\theta^4\right)}{16a''^4_o\beta_o^7}\right] n''_o\left(t-t_o\right) \tag{2.39}$$

$$\omega_{D_F} = \omega_o + \left[-\frac{3k_2\left(1-5\theta^2\right)}{2a''^2_o\beta_o^4} + \frac{3k_2^2\left(7-114\theta^2+395\theta^4\right)}{16a''_o\beta_o^8} \;+\frac{5k_4\left(3-36\theta^2+49\theta^4\right)}{4a''^4_o\beta_o^8}\right] n''_o\left(t-t_o\right) \tag{2.40}$$

$$\Omega_{D_F} = \Omega_o + \left[-\frac{3k_2\theta}{a''^2_o\beta_o^4} + \frac{3k_2^2\left(4\theta-19\theta^3\right)}{2a''^4_o\beta_o^8} + \frac{5k_4\theta(3-7\theta^2)}{2a''^4_o\beta_o^8}\right]\eta''_o(t-t_0) \tag{2.41}$$

$$\delta\omega = B^*\;C_3(\cos\omega_o)(t-t_o) \tag{2.42}$$

$$\delta M = -\frac{2}{3}\left(q_o-s\right)^4 B^*\varepsilon^4\frac{a_E}{e_o\eta}\left[\left(1+\eta\cos M_{D_F}\right)^3-\left(1+\eta\cos M_o\right)^3\right] \tag{2.43}$$

$$M_p = M_{D_F} + \delta\omega + \delta M \tag{2.44}$$

$$\omega = \omega_{D_F} - \delta\omega - \delta M \tag{2.45}$$

$$\Omega = \Omega_{D_F} - \frac{21}{2}\frac{\eta''_o k_2\theta}{a''^2_o\beta_o^2}C_1(t-t_o)^2 \tag{2.46}$$

$$e = e_o - B^*\;C_4(t-t_o) - B^*C_5\left(\sin M_p - \sin M_o\right) \tag{2.47}$$

$$a = a''_o \left[1 - C_1(t - t_o) - D_2(t - t_o)^2 - D_3(t - t_o)^3 - D_4(t - t_o)^4\right]^2 \tag{2.48}$$

$$IL = M_p + \omega + \Omega + \eta''_o \left[\frac{3}{2} C_1(t - t_o)^2 + (D_2 + 2C_1^2)(t - t_o)^3 + dfrac14 \left(3D_3 + 12C_1D_2 + 10C_1^3(t - t_o)^4 + \frac{1}{5}\left(3D_4 + 12C_1D_3 + 6D_2^2 + 30C_1^2D_2 + 15C_1^4\right)\right)(t - t_o)^5\right] \tag{2.49}$$

$$\beta = \sqrt{1 - e^2} \tag{2.50}$$

$$\eta = \frac{k_e}{a^{\frac{3}{2}}} \tag{2.51}$$

where $(t - t_0)$ is the time since the epoch. Note that when the epoch perigee height is less than 220 kilometers, the terms involving C_5, δ_ω, and δ_M are omitted, and the equations for a and IL are simplified after the C_1 term. Incorporate the long-term periodic terms.

$$a_{xN} = e \cos \omega \tag{2.52}$$

$$IL_L = \frac{A_{3,o} \sin i_o}{8k_2 a \beta^2}(e \cos \omega)\left(\frac{3 + 5\theta}{1 + \theta}\right) \tag{2.53}$$

$$a_{yNL} = \frac{A_{3,o} \sin i_o}{Ak_2 a \beta^2} \tag{2.54}$$

$$IL_T = IL + IL_L \tag{2.55}$$

$$a_{yN} = e \sin \omega + a_{yNL} \tag{2.56}$$

Kepler's equation for $(E + \omega)$ can be resolved by defining:

$$U = IL_T - \Omega \tag{2.57}$$

and using the iteration equation:

$$(E+\omega)_{i+1} = (E+\omega)_i + \Delta(E+\omega)_i \tag{2.58}$$

with

$$\Delta(E+\omega)_i = \frac{U - a_{yN}\cos(E+\omega)_i + a_{xN}\sin(E+\omega)_i - (E+\omega)_i}{-a_{yN}\sin(E+\omega)_i - a_{xN}\cos(E+\omega)_i + 1} \tag{2.59}$$

and

$$(E+\omega)_1 = U- \tag{2.60}$$

The initial quantities needed for short-term periodic calculations are determined using the following formulas:

$$e\cos E = a_{xN}\cos(E+\omega) + a_{yN}\sin(E+\omega) \tag{2.61}$$

$$e\sin E = a_{xN}\sin(E+\omega) - a_{yN}\cos(E+\omega) \tag{2.62}$$

$$eL = (a_{x2N} + a_{y2N})_2\, 1 \tag{2.63}$$

$$pL = a(1 - e_L^2) \tag{2.64}$$

$$r = a(1 - e\cos E) \tag{2.65}$$

$$r = k_e \frac{\sqrt{a}}{r} e\sin E \tag{2.66}$$

$$rf = k_e \frac{\sqrt{pL}}{r} e \tag{2.67}$$

$$\cos u = \frac{a}{r}\left[\cos(E+\omega) - a_{X_N} + \frac{a_y N(e \sin E)}{1+\sqrt{1-e_L^2}}\right] \tag{2.68}$$

$$\sin u = \frac{a}{r}\left[\sin(E+\omega) - a_{y_N} - \frac{a_{x_N}(e \sin E)}{1+\sqrt{1-e_L^2}}\right] \tag{2.69}$$

$$u = \arctan\left(\frac{\sin u}{\cos u}\right) \tag{2.70}$$

$$\Delta r = \frac{k_2}{2p_L}(1-\theta^2)\cos 2u \tag{2.71}$$

$$\Delta u = \frac{k_2}{2^2 p_L}(7\theta^2 - 1)\sin 2u \tag{2.72}$$

$$\Delta\Omega = \frac{3k_2\theta}{2p_L^2}\sin 2u \tag{2.73}$$

$$\Delta i = \frac{3k_2\theta}{2p_L^2}\sin i_o \cos 2u \tag{2.74}$$

$$\Delta r = -\frac{k_2\eta}{p_L}(1-\theta^2)\sin 2u \tag{2.75}$$

$$\Delta r\dot{r} = -\frac{k_2\eta}{pL}\left[\left(1-\theta^2\right)\cos 2u - \frac{3}{2}\left(1-3\theta^2\right)\right] \tag{2.76}$$

The osculating quantities are derived by including the short-period periodics:

$$r_k = r\left[1 - \frac{3}{2}k_2\frac{\sqrt{1-e_L^2}}{p_L^2}\left(3\theta^2 - 1\right)\right] + \Delta r \tag{2.77}$$

$$u_K = u + \Delta u \tag{2.78}$$

$$\Omega k = \Omega + \delta\Omega \tag{2.79}$$

$$i_k = i_0 + \Delta i \tag{2.80}$$

$$\dot{r}_k = \dot{r} + \Delta\dot{r} \tag{2.81}$$

$$rf_k = rf + \Delta rf \tag{2.82}$$

Subsequently, the procedure for determining unit orientation vectors is as follows:

$$U = M\sin u_k + N\cos u_k \tag{2.83}$$

$$V = M\cos u_k - N\sin u_k \tag{2.84}$$

where

$$M = \left\{\begin{matrix} M_X = -\sin\Omega_k\cos i_k \\ M_Y = \cos\Omega_k\cos i_k \\ M_Z = \sin i_k \end{matrix}\right\} \tag{2.85}$$

$$M = \left\{ \begin{array}{c} N_X = \cos \Omega_k \\ N_Y = \sin \Omega_k \\ N_Z = 0 \end{array} \right\} \tag{2.86}$$

Subsequently, the data on both the location and velocity are supplied by:

$$r = r_k U \tag{2.87}$$

and

$$\dot{r} = \dot{r}_k + \left(\dot{f}_k V\right) \tag{2.88}$$

2.2.7 Classification of Artificial Satellites

2.2.8 Low Earth Orbit (LEO) Satellites

These distances vary between 500 and 2000 kilometers from the Earth. Within this orbital path, a satellite traverses the earth at a velocity of 7.8 kilometers per second, completing one revolution in around 90 minutes. Big LEO satellites have a frequency range of 1-3 GHz, Little LEO satellites have a frequency that is below 1GHz. There are currently around 5,000 satellites in Low Earth Orbit (LEO) circling the Earth. The majority of the satellites are categorized as military, meteorological, and observational satellites. Low Earth Orbit (LEO) offers several advantages because to its proximity to Earth. These advantages include:

- Satellite imaging offers photographs with greater resolution.
- It is simpler for humans to commute between satellites and Earth that is why it is used by the International Space Station (ISS).
- Voice communication can be achieved at transmission rates of approximately 2,400 bit/s using advanced compression techniques.

- LEO provides access to this bandwidth for mobile devices with the use of omnidirectional antennas that operate at a low 1W broadcast power.
- LEO provides enhanced global coverage due to its substantially greater altitude in polar zones.
- The LEO signal exhibits greater strength compared to the GEO signal, even when both are transmitted with the same power. This is in addition to the previously noted shorter propagation time.

Nevertheless, LEO has certain disadvantages, which encompass:

- Due to its limited communications field, a large number of satellites are required to ensure adequate coverage.
- The requirement for additional protocols for satellite handover between different satellites is prompted by the short duration of visibility at a high altitude.
- The relatively brief duration of 5 to 8 years, resulting from atmospheric factors

2.2.9 Medium Earth Orbit (MEO) Satellites

These range from 5000-25000 Km from the earth. They have a rotation period of 6 hours. Path loss due to antenna beam spreading reduces beam flux density upon reaching the ground. The satellites operating from this orbit are GPS, GLONASS, and Galileo. Their altitudes are respectively 20,200 kilometers, 19,100 kilometers, and 23,222 kilometers. MEO is also utilized by communication satellites that provide coverage for the geographic regions encompassing the North and South Poles.

Benefits of the MEO orbit are:

- Fewer Satellites Required: Due to the higher altitude compared to Low Earth Orbit (LEO), fewer satellites are needed to achieve global coverage.

- Extended Communication Range: Compared to LEO satellites, due to higher altitudes.
- Difficult Propulsion Requirements: A direct flight to MEO is improbable. Needs propulsion.
- Fuel Mass Impact: Higher fuel mass causes indicates payload loss.
- Radiation Exposure: Van Allen Radiation Belts pose risks to onboard hardware.

2.2.10 Geosynchronous Orbit (GEO) Satellites

They are at least 36,000 kilometers above the earth's surface. Their rotational period is 24 hours. The utilization of this type of communications satellite in television and radio transmissions has greatly proliferated in contemporary civilization. The concept was initially proposed by science fiction author Arthur C. Clarke in 1945.

The advantages of the Geostationary Earth Orbit (GEO) are:

- Simplified tracking: Fixed antenna sites can be used by senders and receivers; no adjusting is required.
- Frequency Stability: No problems with frequency variations.
- No Handover: Due to their substantial footprint, they do not require a handover.
- Long Lifespan: The average lifespan of a GEO is approximately 15 years.

The limitations of the GEO orbit are:

- Signal Strength: Poor signal after more than 35,000 km, requiring strong transmission power.
- Coverage Limitations: Because of their permanent position above the equator, bad heights might occur in regions with latitudes above sixty degrees. Polar regions are poorly poorly covered.

- Incompatibility: Small mobile devices cannot use this type of satellite.
- High Power Requirements: Relatively high transmitter power requirements make battery-powered devices problematic.
- Limited Frequency Reuse: Due to the enormous footprint, frequency reuse is not an option.
- High Launch Costs: Launching GEO satellites is significantly expensive.
- Latency Issues: The major issue with speech and data communications, even at the speed of light, is the high latency of approximately 0.25 seconds one-way.

2.3 Mobile Applications for Satellites

2.3.1 Heavens above

Heavens above is a satellite tracking application. To date it has two versions: the free version which has the disadvantage that it has pop up adverts and the pro version which requires the user to pay and download the application in order to use it. Heavens above enables users to view satellites passing through using the Live Sky Chart feature. Additionally, it gives customers accurate up-link and down-link pass predictions for radio satellites and the International Space Station (ISS). Users' phones create the predictions, therefore after a few days, a data connection is required. Users can also get information of satellite orbits, tracks and time satellites pass through the phone.

The applications also provide the date and time of the time zone of the device.

2.3.2 Satorbit

The SatOrbit app provides real-time positions of the International Space Station (ISS) as well as 1450 other spacecraft. The application provides online and offline capabilities after acquiring orbit data and satellite data, including azimuth, elevation, latitude, longitude, altitude, and speed. Additionally, it includes a timer with customizable time intervals that can be utilized for playing, pausing, and stopping. The application presents four distinct map categories: Normal, Satellite, Terrain, and Hybrid. The program displays the solar terminal lines and the satellite footprint. The application displays the sun's position in real time. Additionally, the application offers detailed Map View and Compass View projections for satellite passes with steps up to 24, 48, and 72 hours. Additionally, users can view the sky above them using Map View and Compass View.

The homepage features an application bar and a map that displays the user's current location. The app bar features two horizontal line icons, as well as icons for settings, downloading data, and information. The Map features a plus and a minus symbol that allow users to magnify or reduce the scale. The map displays the user's location on the map using a home icon.

Pressing the three-line icon on the left opens a drawer with a list of satellites that can be selected to track.

The download data icon provides a widget to download data from sources which include celestrack, space track, Amsat and my files.

When the about or information icon is pressed the system provides an provides a widget with help, info and privacy policy

Pressing the three-line symbol on the right side will open a drawer containing a list of satellites that has been chosen to track.

When a satellite is picked, it delivers information about itself. Figure 2.11 displays the output that is produced when the right three-line icon is pressed.

By selecting one of the satellites and using the ”pass view” option, users can get detailed information about the selected spacecraft.

When one of the satellites is selected and pass view and a specific prediction is enabled, the application provides information about the satellite.

Figure 2.3: Homepage for Sat Orbit.

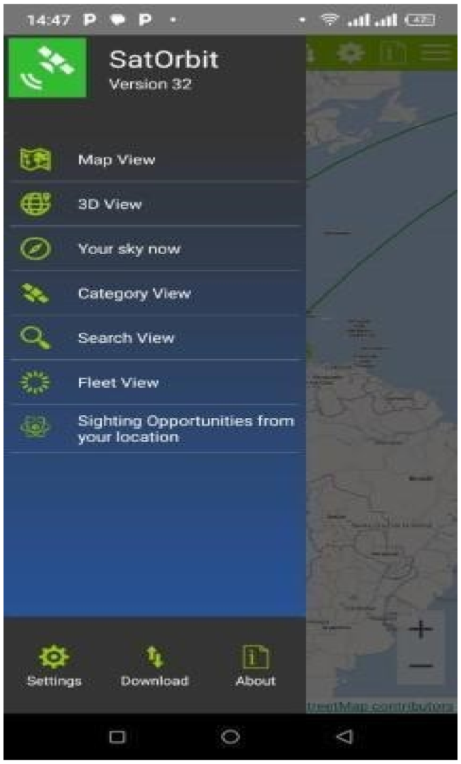


Figure 2.4: Output When the Left Three-Line Icon is Pressed.

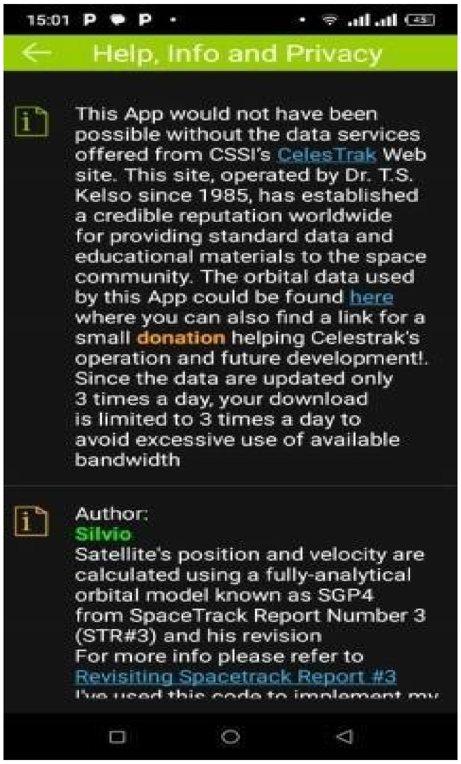
Celestrak (Recommended)
You have access to thousand satellites sorted in category - Best option balancing high computation and responsive User Interface

Space Track
Favorite List: On space-track site you can create your own favorite satellite list and give it a name - best solution if you have a Satellite observing list
You need a valid space-track.org account

Amsat Tle
Bare elements of all satellites of interest to radio amateurs.

My Tle files
You can import Tle orbital elements from a file or enter them manually.

Figure 2.5: Output When the Download Data Icon is Pressed.



This App would not have been possible without the data services offered from CSSI's CelesTrak Web site. This site, operated by Dr. T.S. Kelso since 1985, has established a credible reputation worldwide for providing standard data and educational materials to the space community. The orbital data used by this App could be found here where you can also find a link for a small **donation** helping Celestrak's operation and future development!. Since the data are updated only 3 times a day, your download is limited to 3 times a day to avoid excessive use of available bandwidth

Author:
Silvio
Satellite's position and velocity are calculated using a fully-analytical orbital model known as SGP4 from SpaceTrack Report Number 3 (STR#3) and his revision
For more info please refer to
Revisiting Spacetrack Report #3
I've used this code to implement my

Figure 2.6: Output When the Icon About or Information is Pressed.

AEROCUBE 12A - 14:48:06

Period:92 min | in Sun Light | Sat INFO

Lat	Lon	Alt
8.04N	77.64W	376.78 km
Speed	Tle Age	Time UTC
27665.07 km/h	2.449269 days	12:48:06
Azimuth	Elevation	Inclination
272.04	-53.9	51.63 deg

Figure 2.7: Output When a Satellite is Selected.

Current Satellite: AEROCUBE 12A

Location: 106 Dartford road, willowvale, Harare, Zimbabwe

Only Visible pass: OFF

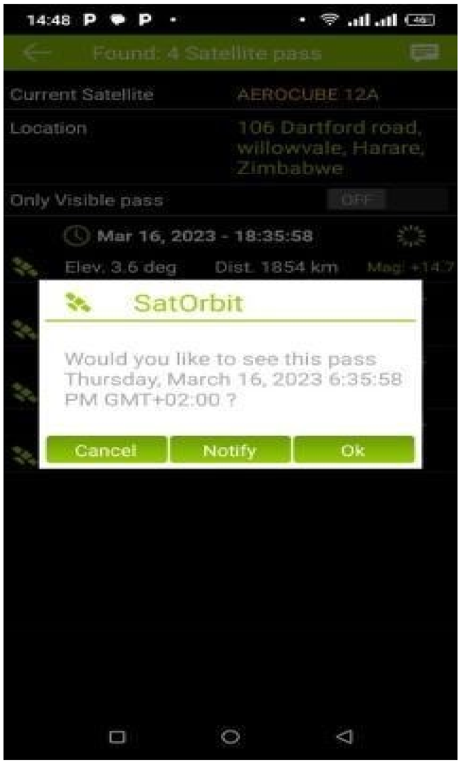
Mar 16, 2023 - 18:35:59
Elev. 3.6 deg Dist. 1854 km Mag: +14.7

Mar 16, 2023 - 20:08:59
Elev. 55.1 deg Dist. 454 km

Mar 17, 2023 - 04:20:59
Elev. 4.3 deg Dist. 1807 km

Mar 17, 2023 - 05:54:49
Elev. 71.6 deg Dist. 397 km

Figure 2.8: Output When a Satellite is Selected and Pass View is Enabled.



Current Satellite: AEROCUBE 12A
Location: 106 Dartford road, willowvale, Harare, Zimbabwe
Only Visible pass: OFF

Would you like to see this pass Thursday, March 16, 2023 6:35:58 PM GMT+02:00 ?

Figure 2.9: Output When a Satellite is Selected and Pass View is Enabled.

2.3.3 Satellite Communication Applications

2.3.4 Google Maps

The smartphone application of Google Maps serves as a prominent example of location-based service applications. By referencing the related geographic context (such as routes, POIs, landmarks, buildings, etc.), the base map assists the user in interpreting his or her location. Google Maps uses reverse geocoding to convert the location coordinates of a user on Earth into address information that is readable by humans. The navigation service is the most widely used service among the two. The navigation service and the ”Nearby” service are the two most widely used services. A user of the ”NearBy” service may want to find a ”sports bar” in close proximity to the ”Super Bowl” location, see figure 3.

The ”NearBy” button grants the user access to the application. The smartphone application will utilize geolocation to search for nearby bars and display their locations on a map. A list of these bars will be provided, with their unique attributes such as address, open/close time, customer reviews, and rating. Once the buyer is informed of this, choosing their preferred alternative will be straightforward.

Furthermore, another widely popular location-based service that is used extensively worldwide is the navigation service. The default origins for a user option are the current GPS location, a destination, and the mode of travel (such as walking, taking public transportation, driving, or bicycling). Once the user is in ”navigation” mode, the ”shortest” route from their current location to destination is displayed on the map.

The map can be rotated and zoomed in to see the locations of the streets. In addition, as more individuals start using Google Map Services, the app will have the capability to retrieve the real-time positions, navigation histories, and pathways of all users. Google Maps may acquire knowledge about people’s travel patterns across the terrain by analyzing historical user logs and other relevant data. To mitigate congestion or traffic jams, it is feasible to identify optimal routes by leveraging real-time data from traffic surveillance systems and/or public traffic reports.

2.3.5 Waze

Waze is a community-based traffic and navigation program that allows users to exchange real-time traffic data. This data helps other users make informed decisions about whether to take a detour or choose an alternative route when traveling on public roadways. The software is part of the Volunteered Geographic Information (VGI) idea, where volunteers contribute descriptions of geographic features or locations to platforms that collect and organize the information into databases. Waze application users have the ability to notify others of police checkpoints, areas with ongoing construction, congested traffic areas, and speed restrictions. In addition, they have the ability to modify routes, landmarks, petrol prices, and other data using the online map editor. They can also update information about roadways, landmarks, petrol prices, and other information via the online map editor. Since they can use the contributions of others to find the closest, cheapest gas station or choose a route with little traffic, many users are involved in the contributions. These sharing options and game features are becoming more and more common in Location-Based Services (LBS) Apps.

2.3.6 Uber

Uber (https://www.uber. com/) offers a location-based real-time carpool share service. Based on the user's location and the position of the driver, Uber wants to assist them in finding a "near-by" Uber car. In practice, these locations are determined by the drivers' mobile devices that have the Uber application installed on them. Uber utilizes efficient and adaptable carpool matching algorithms to promptly identify and notify drivers in close proximity to a passenger's selected destination. If the request is approved by an Uber driver, the user will be transported to their destination. This service dynamically correlates vast volumes of spatiotemporal data with customers and drivers in their respective contexts.

2.4 Non Mobile Phone Satellite Tracking Systems

2.4.1 Automatic Satellite Tracking System on a Movable Platform

Satellite broadcasting refers to the distribution of transmission signals or multimedia information through a satellite network. Satellite broadcasting refers to the distribution of transmission signals or multimedia information through a satellite network. The transmitted signals are typically delivered from a radio or television station to a geostationary artificial satellite via a satellite up-link before being redistributed or re-transmitted to other designated geographic regions over either open or secure channels. The local cable network's base stations, or small home satellite dishes, subsequently receive the down-links and redistribute them to their subscribers. The system was developed to function on both marine vessels and land transport in order to provide customers with real-time access to satellite television channels. The study focused on developing a simple and dependable automatic control system to constantly position the satellite receiver dish. The hardware components of this system include an accelerometer, an Arduino micro-controller, a DC motor driver, and DC motors. An accelerometer that regulates the desired Azimuth and elevation angles at a separate location can automatically control the pointing of a satellite TV dish. The location sensors that enable DC motors to rotate the reception antenna platform are magnetic compasses. An Arduino program running on the Arduino micro-controller determines the precise stopping point for the motor.

Chapter 3: Research Methodology and Design

Detailed information on research methodology and design on the mobile application.

3.1 Packages Used To Build Applications

3.1.1 Orbit

A satellite orbit is defined in terms of seven numbers. After Johann Kepler (1571–1630), this set is known is referred to as "satellite orbital elements" or simply "elements." The satellite's location inside the ellipse at a specific time is determined by first defining the ellipse and then aligning it with the earth. A satellite orbit in an ellipse with a distinct form and direction is described by the Keplerian concept. Tracking algorithms modify the Keplerian model slightly to take into account the fact that the real world is far more complex than the model suggests. These changes are referred to as perturbations. The anomalies in the earth's gravitational field—which, luckily, do not require specification—and the "pull" of the atmosphere on the satellite are known to amateur tracking programs. Drag is now an optional component of the orbit.

3.1.2 Epoch

At a certain point in time, a set of orbital elements records a satellite's path around the sun. Epoch is just a number that indicates the exact time that a picture was taken.

---

[13] Max H Lane and Felix R Hoots. General perturbations theories derived from the 1965 lane drag theory, 1979

[11] Felix R Hoots and Ronald L Roehrich. Models for propagation of NORAD element sets. Office of Astrodynamics, 1980

3.1.3 Inclination

Equatorial orbits are those with an inclination of less than 0 degrees, meaning the satellite stays almost directly above the equator. On the other hand, polar orbits have have an inclination close to 90 degrees since the satellite crosses both the North and South Poles. The line of nodes is the point where the orbital plane and the equatorial plane cross. This concept will be discussed in greater detail shortly.

The node line consists of two passes. The ascending node is where the satellite crosses the equator, moving from south to north. Conversely, the descending node is, where the satellite crosses the equator from north to south. By convention, the ascending node is the reference point used. The right ascension/declination coordinate system is an astronomical system that does not revolve around the Earth. When right ascension is set to zero, an angle is defined as the angle measured in the equatorial plane from a point in the sky. Right ascension is the term used to describe this angle. The vernal equinox is the name given to this moment by astronomers. The angle from the vernal equinox to the ascending node, measured at the center of the earth, is known as the "right ascension of ascending node" (RAAN). The RAAN of the satellite's orbit is the angle, as measured at the Earth's center, between the location where the Sun's orbit emerges past the equator and the location where the satellite's orbit intersects the equator.

3.1.4 Argument of Periapsis

The satellite's closest point to Earth is known as perigee, which is also occasionally referred to as periapsis or perifocus. Conversely, the satellite's position with regard to Earth is called its apogee, which is sometimes called its apoapsis or apifocus. The term line-of-apsides (the ellipse's major axis) refers to the line that connects the apogee and perigee. The plural of apsis is apsides. The earth's center is where the line-of-apsides crosses. The argument of perigee is the name given to the angle formed by these two lines. More precisely, the angle from the ascending node to perigee (measured at the earth's center) is the argument of perigee. Two additional angles are created anywhere two lines converge.

3.1.5 Eccentricity

Eccentricity gives details on the "shape" of the ellipse. The zeroth energy is a circle that is the ellipse. When eccentricity is approaches 1, the ellipse becomes more elongated and narrow. Conic sections include straight lines, parabolas, hyperbolas and, ellipses (including circles). The Keplerian orbit is a type of conic section. The other orbital configurations are normally not employed by satellites, at least not intentionally, and tracking software is typically not designed to handle them.

3.1.6 Mean Motion

Kepler's third law of orbital motion accurately relates the satellite's speed to its distance from the earth. Satellites in close proximity to Earth orbit quickly, while those with a longer orbit move slowly. This suggests that determining the satellite's velocity or distance from the earth can yield the same outcome! Satellites in circular orbit travel at a constant pace. Easy. Just saying that speed gets the work done. Satellites in non-circular orbits (e.g., eccentricity greater than 0) travel faster while close to the earth and slower when far from it. The standard practice is to average the speed. Although the phrase "average speed" can be employed, this statistic is referred to by astronomers as "mean motion". The units used to express mean motion are typically revolutions per day. "Orbit period" can sometimes refer to an orbital element other than mean motion. Period is the reciprocal of mean motion. For example, a satellite with a mean motion of two revolutions per day has a period of 12 hours. The semi-major axis (SMA), which may occasionally be replaced with Mean Motion (SM), is half the length of the orbit ellipse (measured the long way), and it is related to mean motion by a simple equation.

3.1.7 Mean Anomaly

Astronomers use the term ”anomaly” to refer to an angle. The mean anomaly is an angle that moves uniformly in time from 0 to 360 degrees in one orbit. Therefore, it is 0 degrees at perigee and 180 degrees at apogee.

3.1.8 Drag

Satellites experience drag due to the Earth’s atmosphere, causing them to gradually descend and accelerate as they spiral down. The only information conveyed by the Drag orbital element is the rate at which drag or related events are changing the Mean Motion. Drag is equal to half of the rate of change of mean motion over time.

3.1.9 Revolution Number at Epoch

The tracking application uses this information to determine how many orbits the satellite has completed between launch and the specified ”Epoch”. This value, referred to as epochrev, helps the tracking application display the correct revolution number.

3.1.10 Attitude

A measurement of a satellite’s orientation in orbit is its attitude. Bahn coordinates are applicable only to spin-stabilized spacecraft, which keep their antennas pointed in a fixed direction, maintaining a steady inertial orientation. This method is also known as Bahn Coordinates. The two angles that make up the Bahn Coordinates are known as Bahn Latitude and Bahn Longitude. Similar to how geographic latitude and longitude indicate a direction from the center of the earth, these two integers denote a direction in a spherical coordinate system. However, in this case, the major axis is parallel to the path taken by the satellite to reach the planet’s center during perigee.

Keys features:

- Analysis of Two-Line Elements (TLE).
- Application of SGP4, SDP4, SGP8, and SDP8 algorithms.
- Satellite propagation in outer space.
- TLE (Two-Line Element)parsing.
- Keplerian Elements.
- Implementation of SGP4, SDP4, SGP8 and SDP8.
- Satellite propagation.
- Calculation of Rise and Set times for Satellites.
- Calculation of look angle from the observer's location on the planet.
- Earth WGS72 and WGS84 and generic planet.
- Calculation of the path followed by an object on the Earth's surface, known as the ground track.

3.1.11 SGP4/SDP4

The SGP4/SDP4 program employs NORAD methodologies to accurately identify satellites and compute their orbital velocity. The methods were obtained from the NORAD article "Space Track Report No. 3", which was issued in December 1980. The SGP4/SDP4 package includes the orbital algorithms SGP4 for objects that are in close proximity to Earth and SDP4 for those that are in deep space. The satellite tracking sector frequently utilizes these methods with the existing NORAD two-line element data, resulting in exceptionally precise outcomes.

3.2 Retrofit

Retrofit is a code generator that changes types by utilizing source generation to create Dio clients. It drew inspiration from Chopper and Retrofit. The ability to obtain or provide data to the server using an API relies on the server's connectivity. The REST API is extensively utilized in Android and iOS applications, with Retrofit being the favored framework for establishing the API. Retrofit is distinguished among the other libraries for getting the API due to its user-friendly interface and capability to build code that is concise and well-organized.

3.2.1 Json

The creation of JSON serialization and deserialization code uses JSON annotation. JSON Serialization translates JSON data into a model class.

3.2.2 Dio

Dio serves as the HTTP client and manages connections. It is user-friendly and supports Interceptors, Global configuration, Form Data, File downloading, etc. Dio is very easy to use. Dio manages various edge situations and simplifies networking. Dio provides the security of a complex error-handling approach while managing multiple active network requests. Furthermore, it reduces the boilerplate code required to monitor the progress of any file upload using the HTTP package.

3.2.3 Flutter_Map Plugin

The Flutter_Map plugin is utilized to graphically represent the trajectory taken by a satellite.

3.2.4 Python SGP4

The Python SGP4 mathematical model is utilized to predict the trajectory of a satellite based on its name and Two-Line Elements (TLEs).

1. Python Flask creates endpoints for the sgp4 prediction mode1
2. The Flutter_local_notifications plugin is used to send notifications to user when TLE data download is complete
3. The Flutter_Map plugin is used to plot path followed by a satellite

3.2.5 Block Diagram Utilizing Widgets

The widgets used in the project include main navigation bar form screens form widgets, form results widgets, download widget and satellite model widget. The widget-based block diagram below illustrates the architecture of the application. The back-end is coded in Python, while the front-end in Flutter.

3.2.6 Main Widget

Google Maps package which gets information on the user's to retrieve information about the user's position. The main function initializes the navigation bar widget.

3.2.7 The Navigation Bar Widget

The navigation bar widget incorporates a drawer package that encompasses a category view, a search view, a screen for exiting the satellite passes, and containers for data downloads. The Forms widget requires the inclusion of the name field, line one field, and line two field. The code sample includes visual representations.

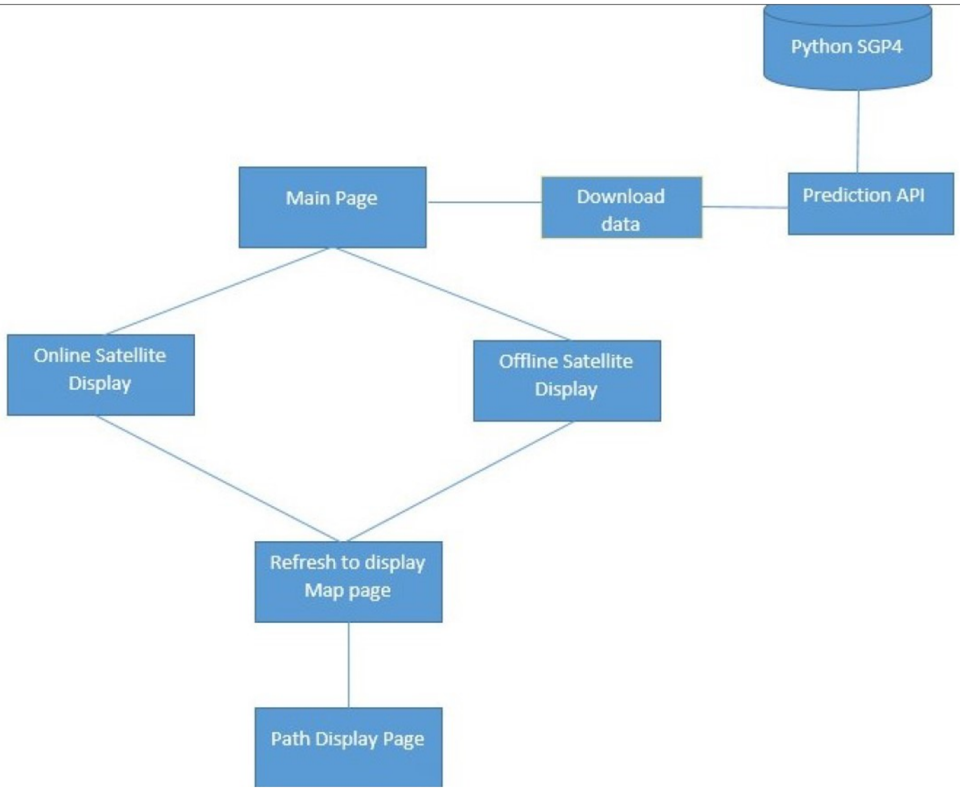
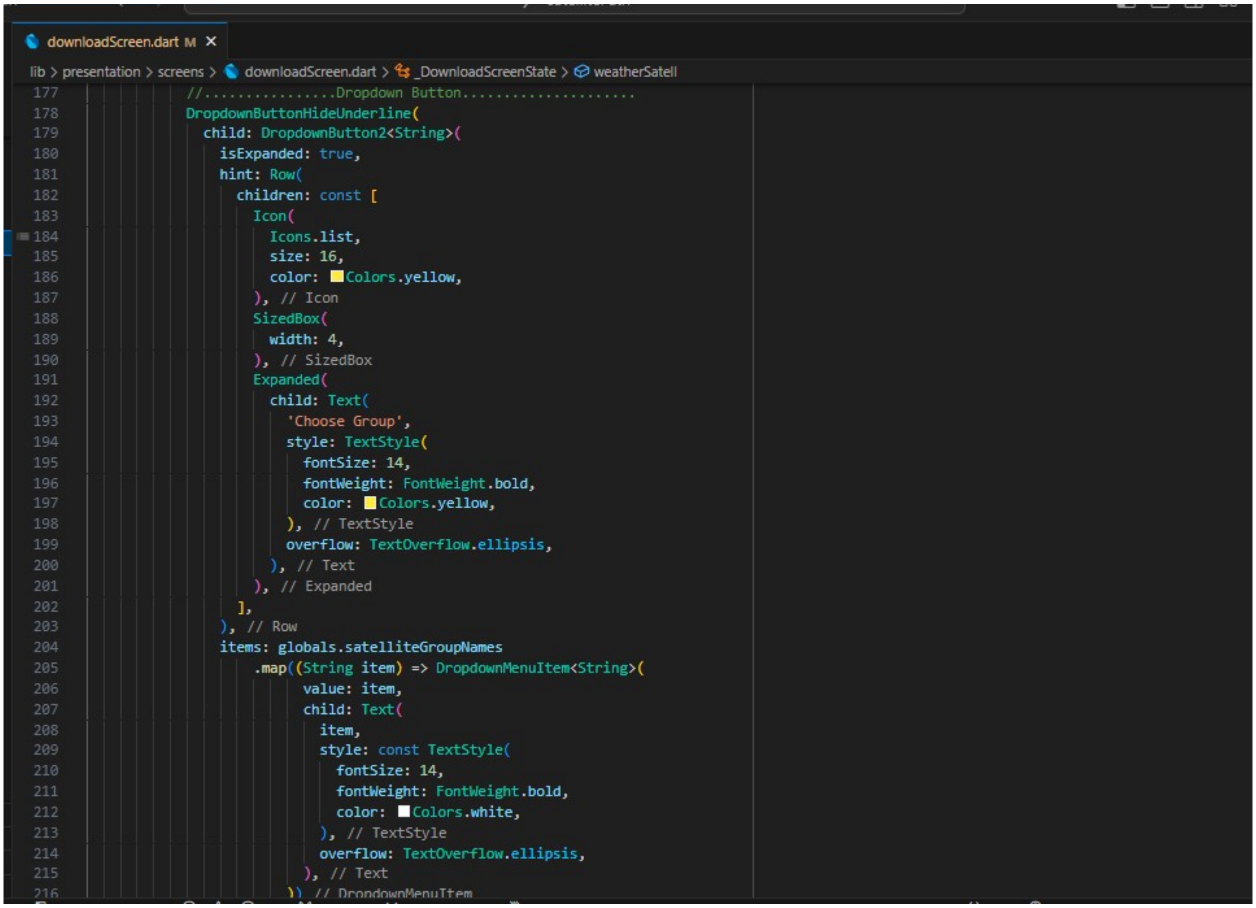


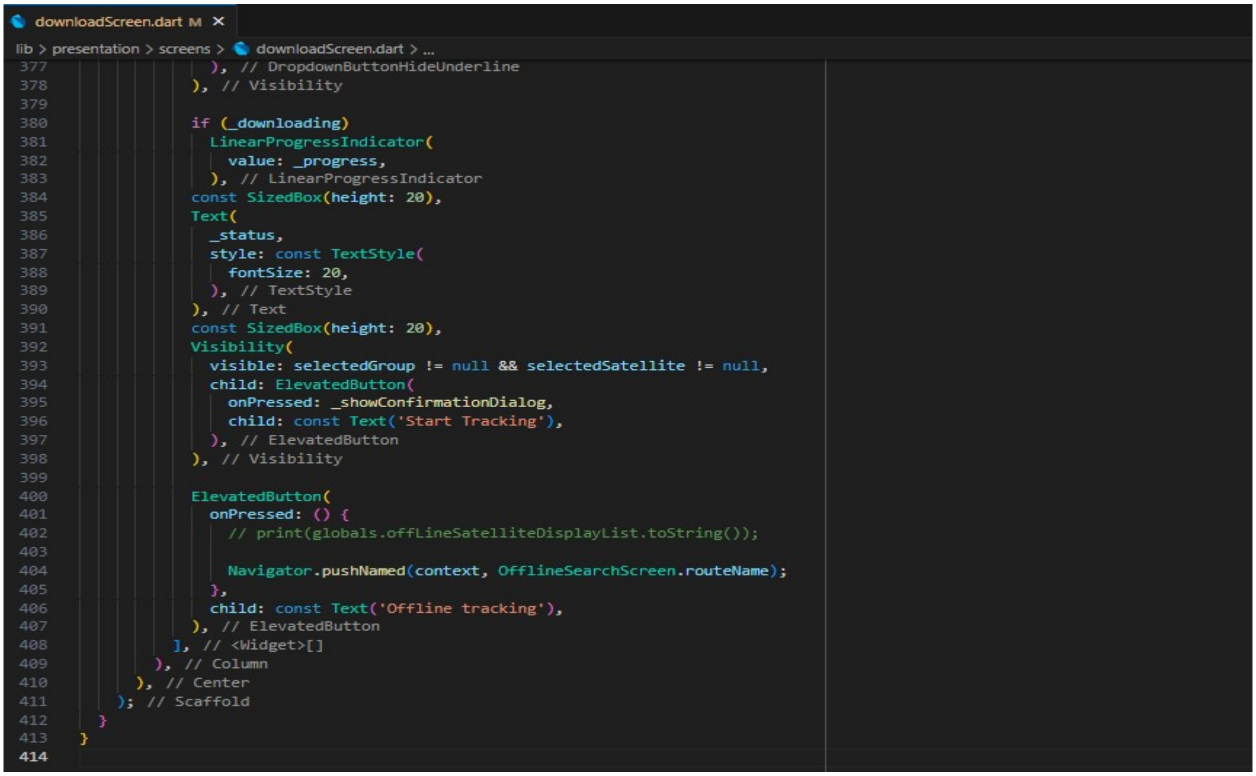
Figure 3.1: System Block Diagram Using Widgets



```dart
//...............Dropdown Button....................
DropdownButtonHideUnderline(
  child: DropdownButton2<String>(
    isExpanded: true,
    hint: Row(
      children: const [
        Icon(
          Icons.list,
          size: 16,
          color: Colors.yellow,
        ), // Icon
        SizedBox(
          width: 4,
        ), // SizedBox
        Expanded(
          child: Text(
            'Choose Group',
            style: TextStyle(
              fontSize: 14,
              fontWeight: FontWeight.bold,
              color: Colors.yellow,
            ), // TextStyle
            overflow: TextOverflow.ellipsis,
          ), // Text
        ), // Expanded
      ],
    ), // Row
    items: globals.satelliteGroupNames
        .map((String item) => DropdownMenuItem<String>(
              value: item,
              child: Text(
                item,
                style: const TextStyle(
                  fontSize: 14,
                  fontWeight: FontWeight.bold,
                  color: Colors.white,
                ), // TextStyle
                overflow: TextOverflow.ellipsis,
              ), // Text
            )) // DropdownMenuItem
```

Figure 3.2: Code Snippet for Selecting Satellite and its Category

The code snippet includes buttons that allow the user to download satellite names and TLEs, as well as move to the map panel.

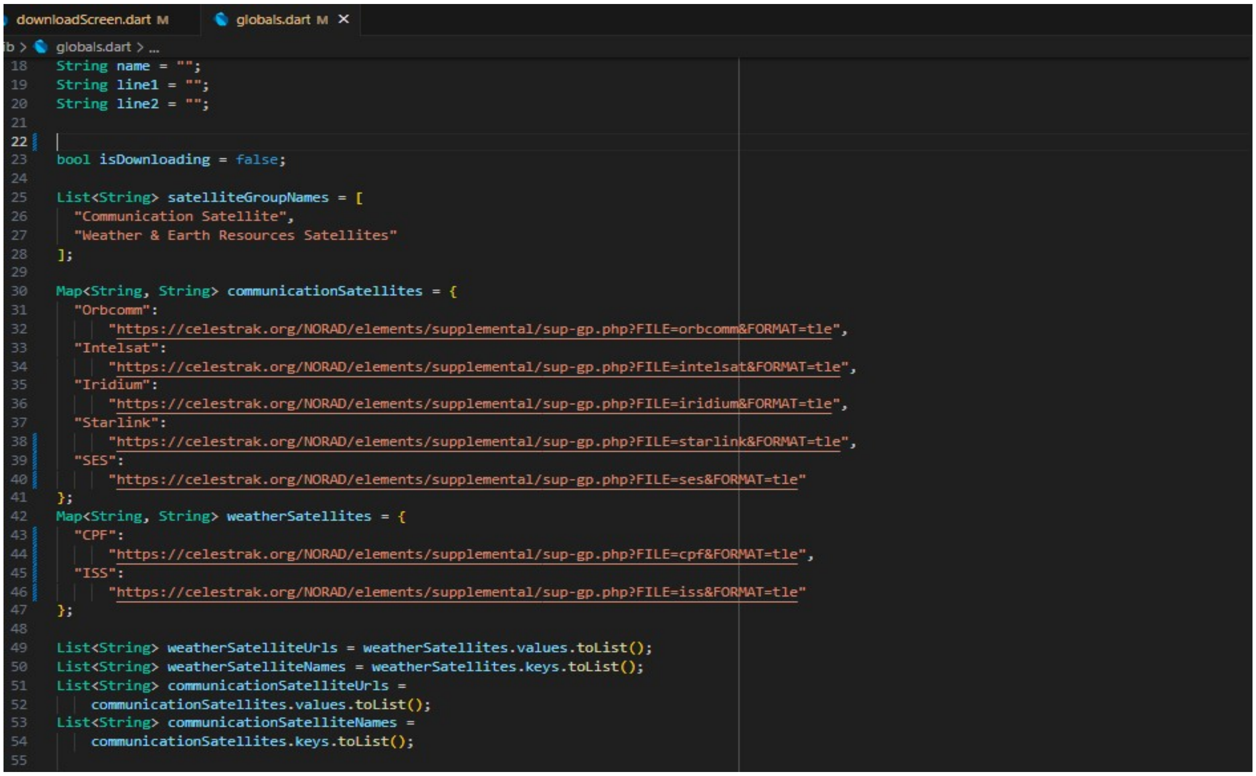


```
              ), // DropdownButtonHideUnderline
            ), // Visibility

            if (_downloading)
              LinearProgressIndicator(
                value: _progress,
              ), // LinearProgressIndicator
            const SizedBox(height: 20),
            Text(
              _status,
              style: const TextStyle(
                fontSize: 20,
              ), // TextStyle
            ), // Text
            const SizedBox(height: 20),
            Visibility(
              visible: selectedGroup != null && selectedSatellite != null,
              child: ElevatedButton(
                onPressed: _showConfirmationDialog,
                child: const Text('Start Tracking'),
              ), // ElevatedButton
            ), // Visibility

            ElevatedButton(
              onPressed: () {
                // print(globals.offLineSatelliteDisplayList.toString());

                Navigator.pushNamed(context, OfflineSearchScreen.routeName);
              },
              child: const Text('Offline tracking'),
            ), // ElevatedButton
          ], // <Widget>[]
        ), // Column
      ), // Center
    ); // Scaffold
  }
}

```

Figure 3.3: Code Snippet for Tracking

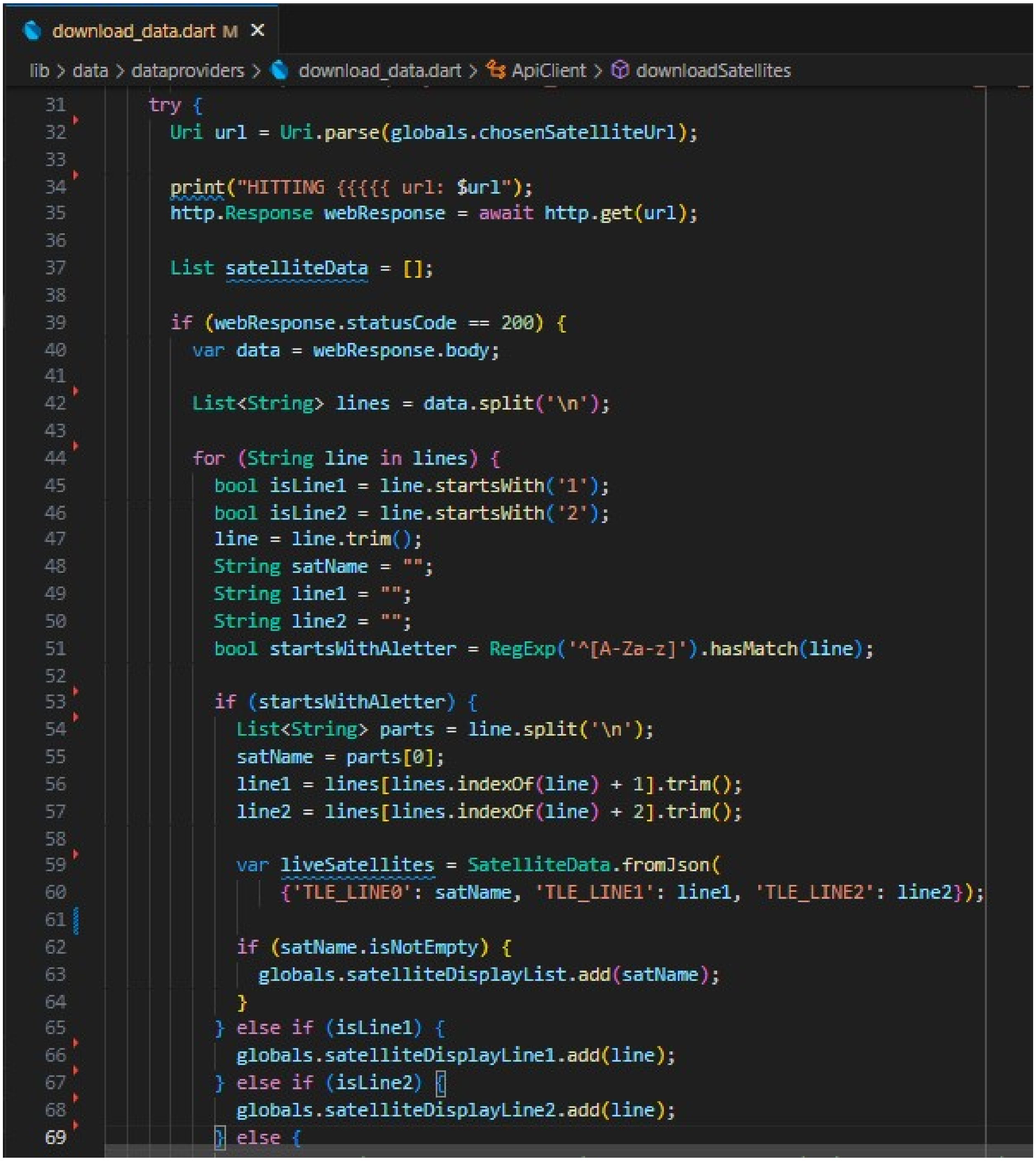
3.2.8 Access The Data Widget

The widget includes hyperlinks that let websites to fetch raw data from the Celestrak server using the Flutter http package.



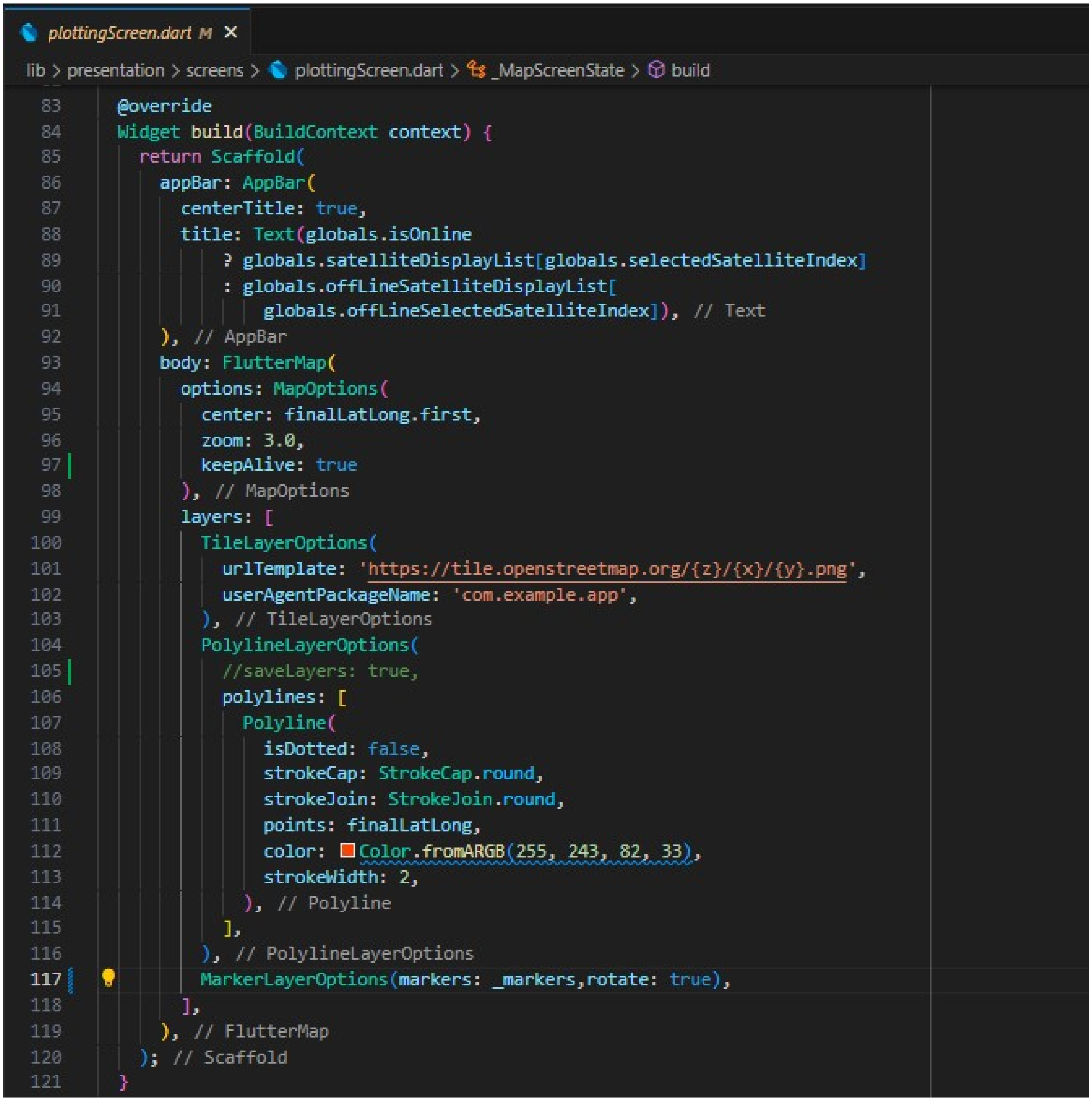
```dart
String name = "";
String line1 = "";
String line2 = "";



bool isDownloading = false;

List<String> satelliteGroupNames = [
  "Communication Satellite",
  "Weather & Earth Resources Satellites"
];

Map<String, String> communicationSatellites = {
  "Orbcomm":
      "https://celestrak.org/NORAD/elements/supplemental/sup-gp.php?FILE=orbcomm&FORMAT=tle",
  "Intelsat":
      "https://celestrak.org/NORAD/elements/supplemental/sup-gp.php?FILE=intelsat&FORMAT=tle",
  "Iridium":
      "https://celestrak.org/NORAD/elements/supplemental/sup-gp.php?FILE=iridium&FORMAT=tle",
  "Starlink":
      "https://celestrak.org/NORAD/elements/supplemental/sup-gp.php?FILE=starlink&FORMAT=tle",
  "SES":
      "https://celestrak.org/NORAD/elements/supplemental/sup-gp.php?FILE=ses&FORMAT=tle"
};
Map<String, String> weatherSatellites = {
  "CPF":
      "https://celestrak.org/NORAD/elements/supplemental/sup-gp.php?FILE=cpf&FORMAT=tle",
  "ISS":
      "https://celestrak.org/NORAD/elements/supplemental/sup-gp.php?FILE=iss&FORMAT=tle"
};

List<String> weatherSatelliteUrls = weatherSatellites.values.toList();
List<String> weatherSatelliteNames = weatherSatellites.keys.toList();
List<String> communicationSatelliteUrls =
    communicationSatellites.values.toList();
List<String> communicationSatelliteNames =
    communicationSatellites.keys.toList();

```

Figure 3.4: Snippet Tile Code for the TLE Links to CelesTrak Website



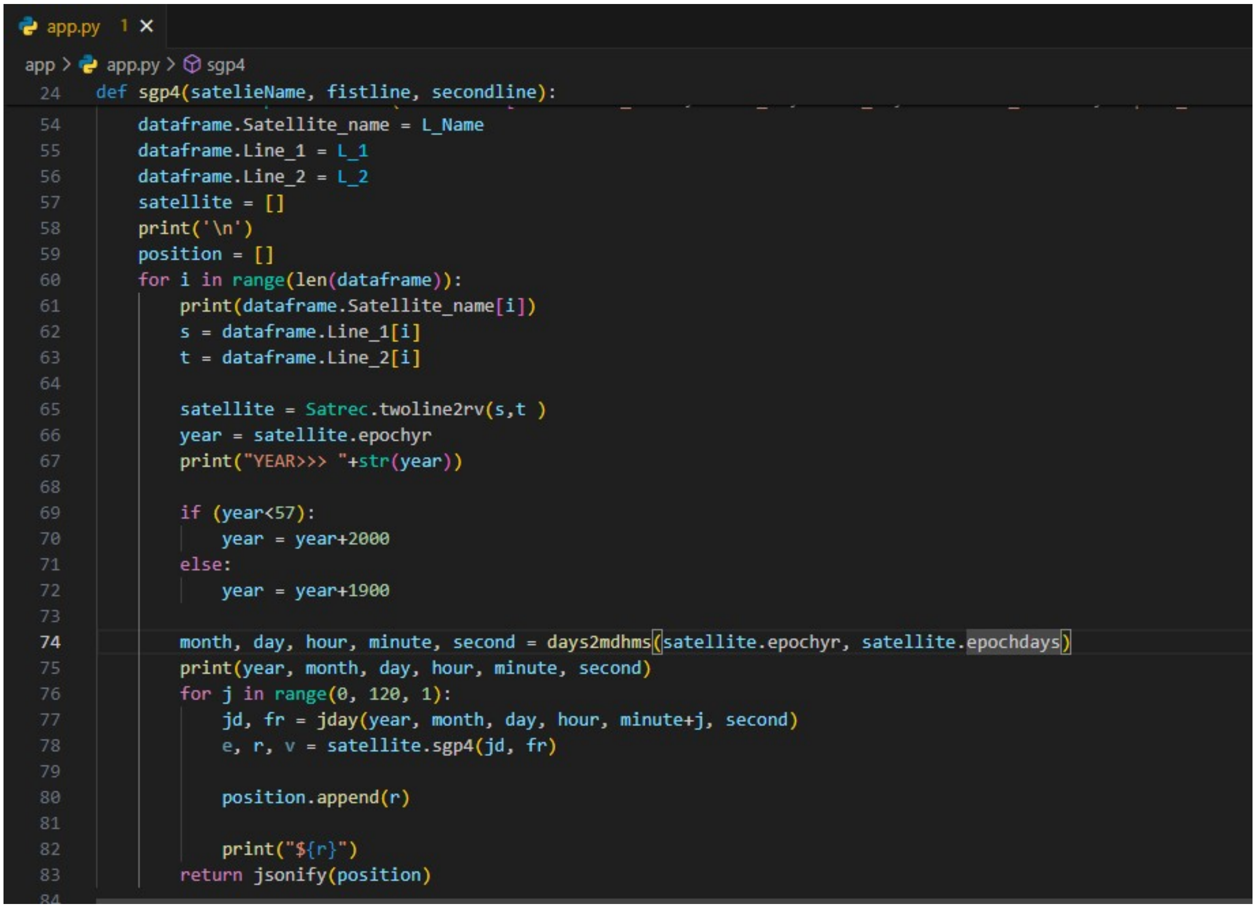
```
    try {
      Uri url = Uri.parse(globals.chosenSatelliteUrl);

      print("HITTING {{{{{ url: $url");
      http.Response webResponse = await http.get(url);

      List satelliteData = [];

      if (webResponse.statusCode == 200) {
        var data = webResponse.body;

        List<String> lines = data.split('\n');

        for (String line in lines) {
          bool isLine1 = line.startsWith('1');
          bool isLine2 = line.startsWith('2');
          line = line.trim();
          String satName = "";
          String line1 = "";
          String line2 = "";
          bool startsWithAletter = RegExp('^[A-Za-z]').hasMatch(line);

          if (startsWithAletter) {
            List<String> parts = line.split('\n');
            satName = parts[0];
            line1 = lines[lines.indexOf(line) + 1].trim();
            line2 = lines[lines.indexOf(line) + 2].trim();

            var liveSatellites = SatelliteData.fromJson(
                {'TLE_LINE0': satName, 'TLE_LINE1': line1, 'TLE_LINE2': line2});

            if (satName.isNotEmpty) {
              globals.satelliteDisplayList.add(satName);
            }
          } else if (isLine1) {
            globals.satelliteDisplayLine1.add(line);
          } else if (isLine2) {
            globals.satelliteDisplayLine2.add(line);
          } else {
```

Figure 3.5: Snippet Code for the Download Data Widget.



```dart
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        title: Text(globals.isOnline
            ? globals.satelliteDisplayList[globals.selectedSatelliteIndex]
            : globals.offLineSatelliteDisplayList[
                globals.offLineSelectedSatelliteIndex]), // Text
      ), // AppBar
      body: FlutterMap(
        options: MapOptions(
          center: finalLatLong.first,
          zoom: 3.0,
          keepAlive: true
        ), // MapOptions
        layers: [
          TileLayerOptions(
            urlTemplate: 'https://tile.openstreetmap.org/{z}/{x}/{y}.png',
            userAgentPackageName: 'com.example.app',
          ), // TileLayerOptions
          PolylineLayerOptions(
            //saveLayers: true,
            polylines: [
              Polyline(
                isDotted: false,
                strokeCap: StrokeCap.round,
                strokeJoin: StrokeJoin.round,
                points: finalLatLong,
                color: Color.fromARGB(255, 243, 82, 33),
                strokeWidth: 2,
              ), // Polyline
            ],
          ), // PolylineLayerOptions
          MarkerLayerOptions(markers: _markers,rotate: true),
        ],
      ), // FlutterMap
    ); // Scaffold
  }
```

Figure 3.6: Embedded Map in Application



```python
def sgp4(satelieName, fistline, secondline):
    dataframe.Satellite_name = L_Name
    dataframe.Line_1 = L_1
    dataframe.Line_2 = L_2
    satellite = []
    print('\n')
    position = []
    for i in range(len(dataframe)):
        print(dataframe.Satellite_name[i])
        s = dataframe.Line_1[i]
        t = dataframe.Line_2[i]

        satellite = Satrec.twoline2rv(s,t )
        year = satellite.epochyr
        print("YEAR>>> "+str(year))

        if (year<57):
            year = year+2000
        else:
            year = year+1900

        month, day, hour, minute, second = days2mdhms(satellite.epochyr, satellite.epochdays)
        print(year, month, day, hour, minute, second)
        for j in range(0, 120, 1):
            jd, fr = jday(year, month, day, hour, minute+j, second)
            e, r, v = satellite.sgp4(jd, fr)

            position.append(r)

            print("${r}")
        return jsonify(position)
```

Figure 3.7: Python Code Snippet for Implementing the SGP4 Prediction Model.

Chapter 4: Results And Analysis

4.1 Results

The application allows a user to choose a satellite group and download data. A list of satellites downloaded from CelesTrak website is displayed. The user can search through the list and choose a satellite to predict the path. Clicking the refresh button navigates to the map page showing the two hours prediction path on a map. Throughout the whole app navigation process, there are no advertisements or any other popups. The app sends an in app notification to the user when the app is ready for off-line use.

The map visualization on the application differs from the two-dimensional map displayed on the website, resulting in variations in the plotting of paths. The map displayed on the website exhibits a seamless and uninterrupted representation. The straight line on the mobile app map corresponds to a complete orbit made by the satellite.

4.1.1 Overview of System Features and Functions

The portal provides features such as offline tracking, options to download data, and the ability to choose satellites. Figure 4.1 displays the screenshot.

4.1.2 Collection Of Satellite Data

The CelesTrak data retrieval API displays satellite names upon clicking the download button. The Python SGP4 API is utilized to predict the trajectory of the satellite in the background. Furthermore, the expected data is kept for offline utilization. Once the application is ready for offline use, a notification will be displayed.

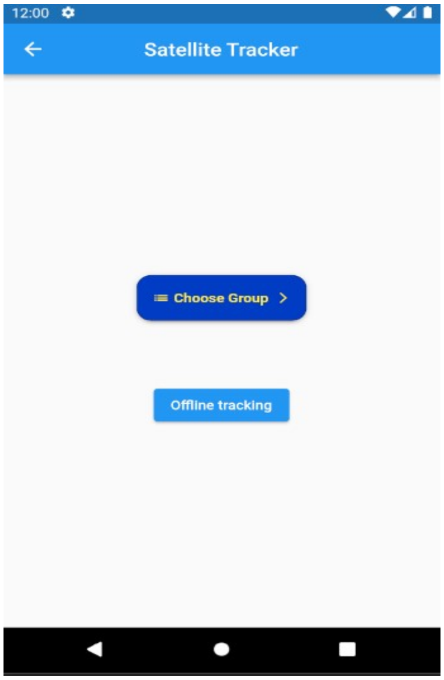


Figure 4.1: Homepage Screen.

4.1.3 A Picture of the Page that Loads the Results

A map shows the route taken by the satellite. The satellite's longitude and latitude can be seen by the user by clicking on a position.

4.1.4 Getting Data From Satellites

The maps shown in Figure 4.7 below for Iridium 106 are pretty much the same on both the Orbtrak website and the mobile app. The changes are because the apps and websites give different time predictions. The app predicts for two hours by taking satellite positions every minute, while the Orbtrak website predicts for five days.

The paths that Iridium 124 is expected to take in figure 4.8 below are pretty much the same on both the Orbtrak website and the mobile app. The changes are because the apps and websites give different time predictions. The app predicts for two hours by taking satellite positions every minute, while the Orbtrak website predicts for five days.

Figure 4.2: Select Options Page

See Figure 4.9 for the ORBCOMM FM110 travel predictions. They are pretty much the same on both the mobile app and the Orbtrak website. The changes are because the apps and websites give different time predictions. The app predicts for two hours by taking satellite positions every minute, while the Orbtrak website predicts for five days.

However, the website's path doesn't show any paths, while the app's path is a straight line. This is because the GALAXY 11 satellite stays in the same place in relation to the earth. This is also true for the INTELSAT 5 (IS-5) satellite shown in Figure 4.10.

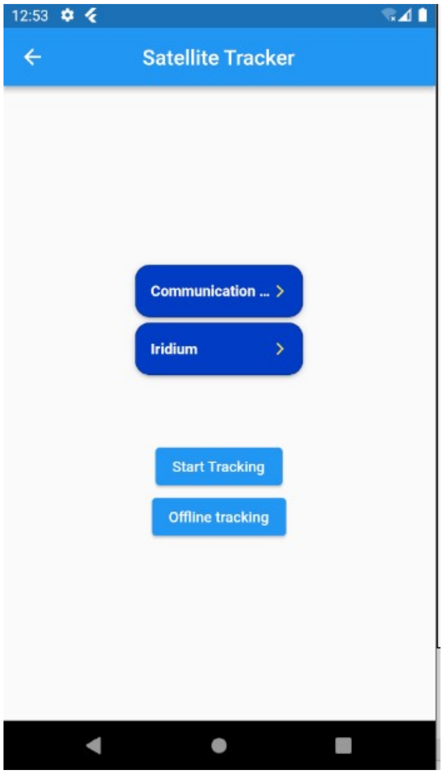
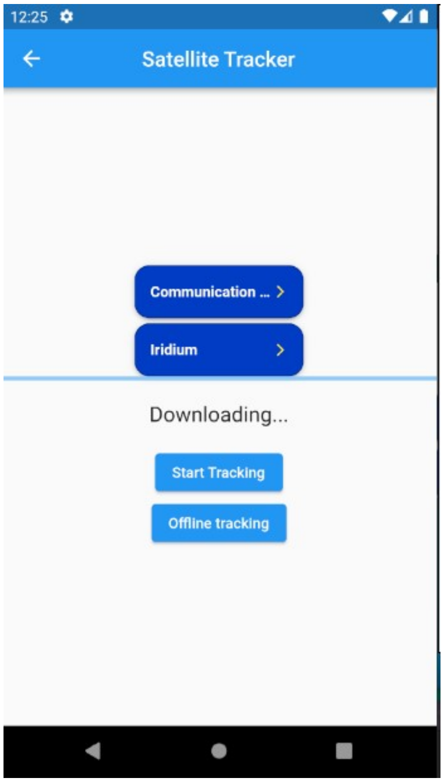


Figure 4.3: Data Collection Page Screenshot.



Downloading...

Figure 4.4: Loading Widget in Results Page.

Figure 4.5: Results Page 1.

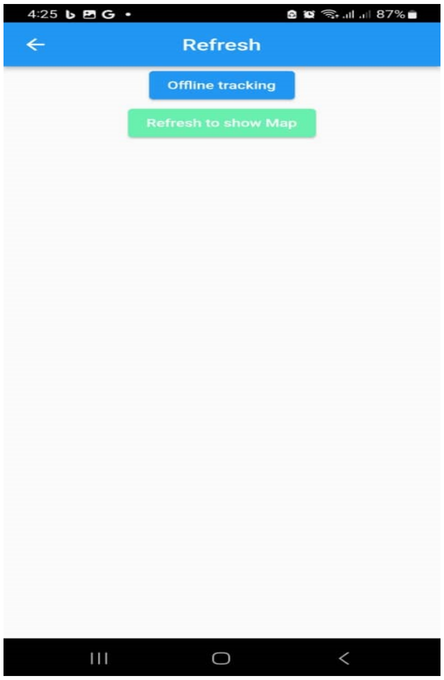
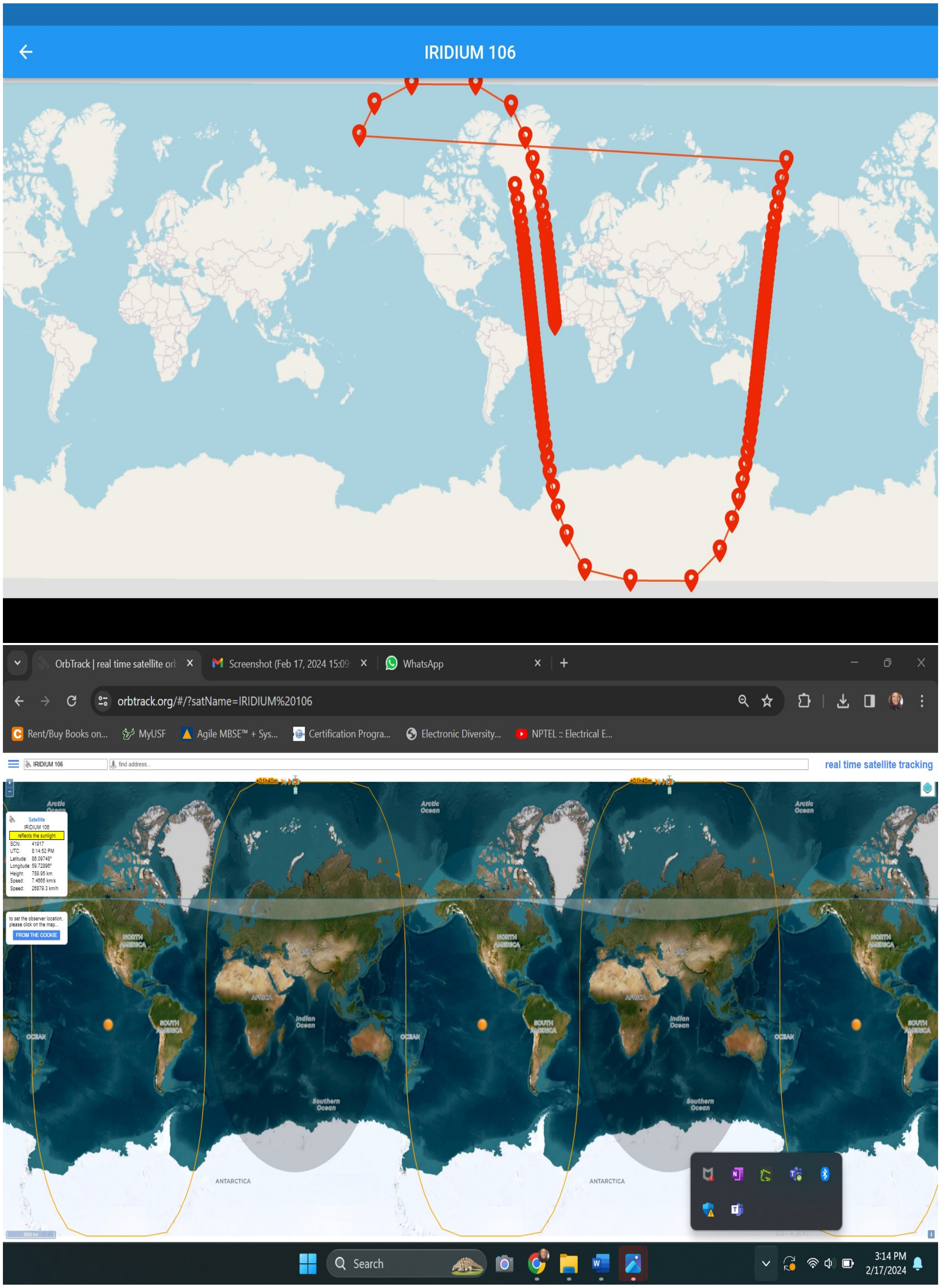


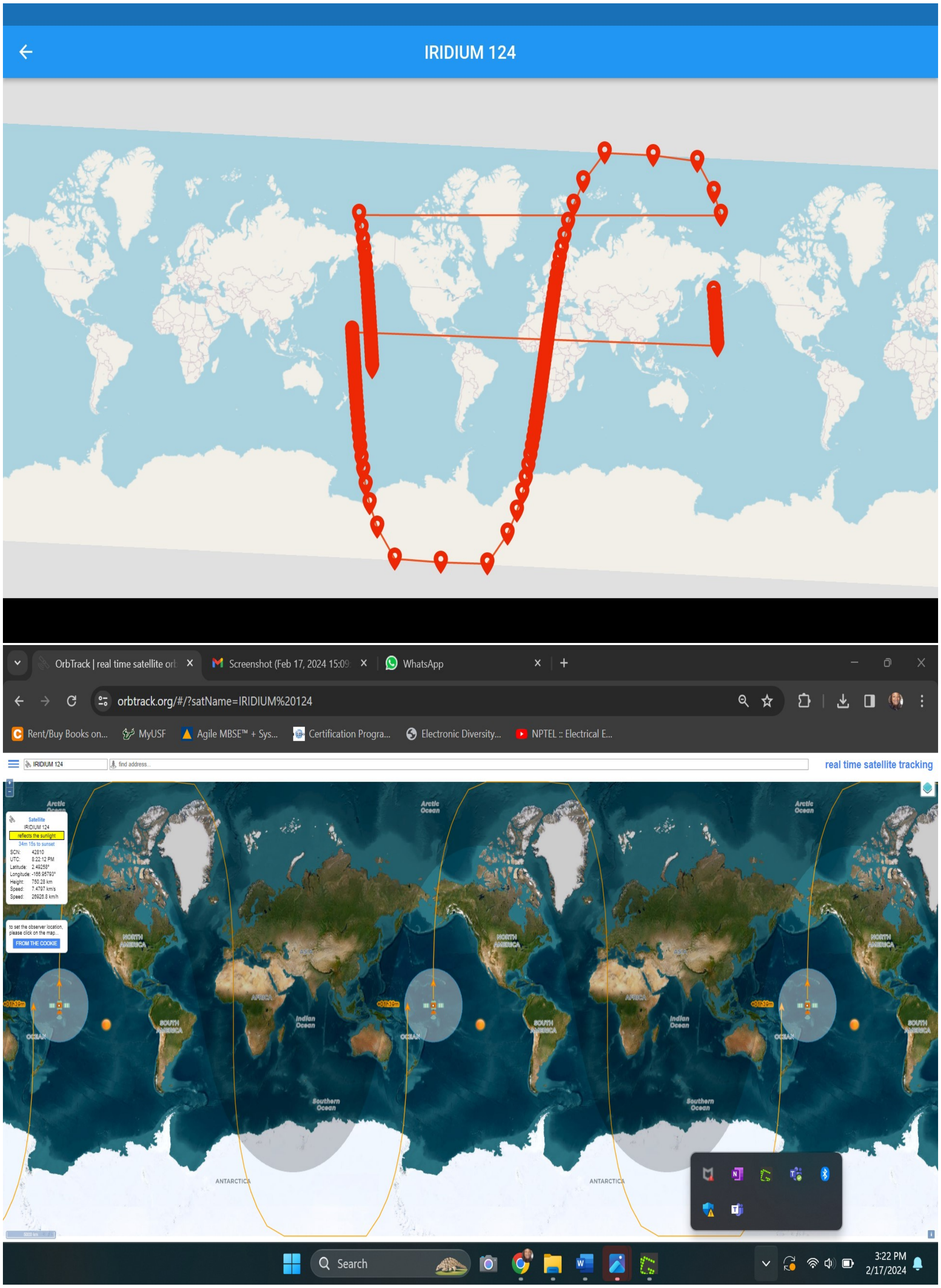
Figure 4.6: Results Page 2.



Satellite
IRIDIUM 106
reflects the sunlight
SCN: 41917
UTC: 8:14:52 PM
Latitude: 86.09748°
Longitude: 59.72896°
Height: 789.95 km
Speed: 7.4665 km/s
Speed: 26879.3 km/h

to set the observer location, please click on the map...

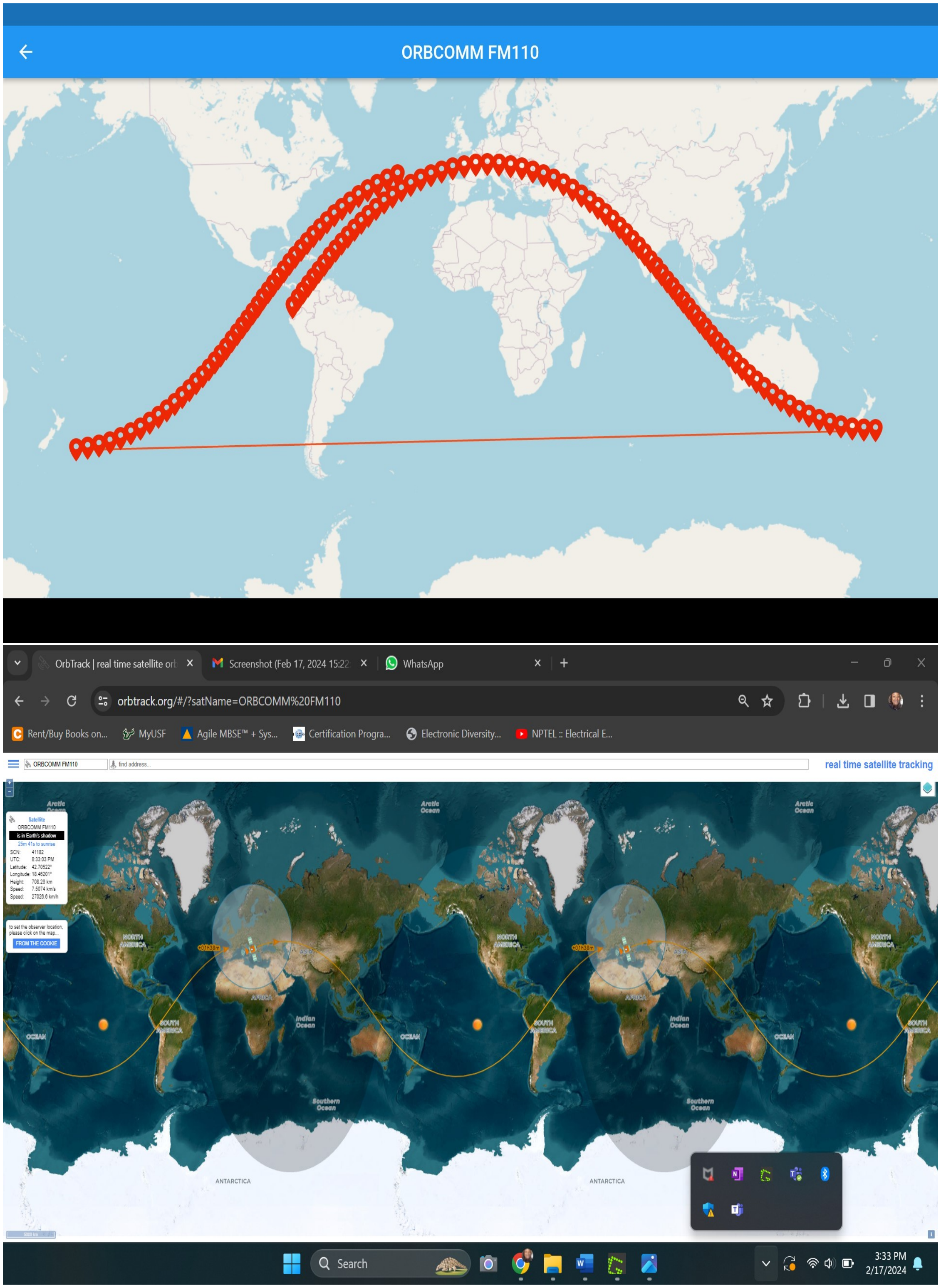
Figure 4.7: Comparison of Application's and Orbtrack Website's Iridium 106 Satellite Predicted Path



Satellite
IRIDIUM 124
reflects the sunlight
34m 15s to sunset
SCN: 42810
UTC: 8:22:12 PM
Latitude: 2.49258°
Longitude: -166.95793°
Height: 750.28 km
Speed: 7.4797 km/s
Speed: 26926.8 km/h

to set the observer location, please click on the map...

Figure 4.8: Comparison of the Iridium 124 Satellite's Predicted Path On the Orbtrack Website and the Application

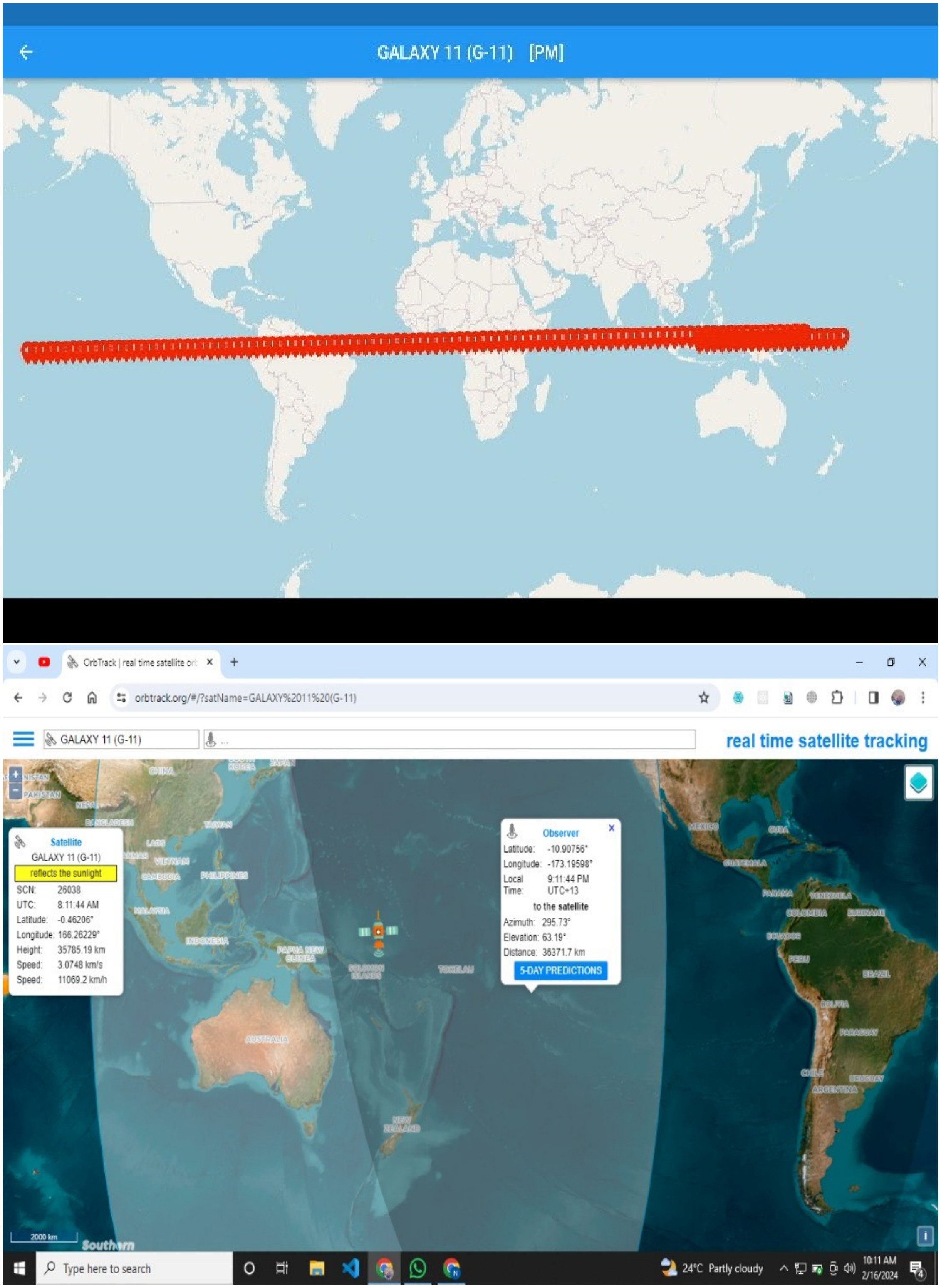


Satellite
ORBCOMM FM110
is in Earth's shadow
25m 41s to sunrise
SCN: 41182
UTC: 8:33:03 PM
Latitude: 42.70522°
Longitude: 18.46201°
Height: 708.26 km
Speed: 7.5074 km/s
Speed: 27026.6 km/h

to set the observer location, please click on the map...

FROM THE COOKIE

Figure 4.9: The ORBCOMM FM110 Satellite's Predicted Path Can Be Found 0n the Orbtrak Website and the Mobile Application



Satellite

GALAXY 11 (G-11)

reflects the sunlight

SCN: 26038
UTC: 8:11:44 AM
Latitude: -0.46206°
Longitude: 166.26229°
Height: 35785.19 km
Speed: 3.0748 km/s
Speed: 11069.2 km/h

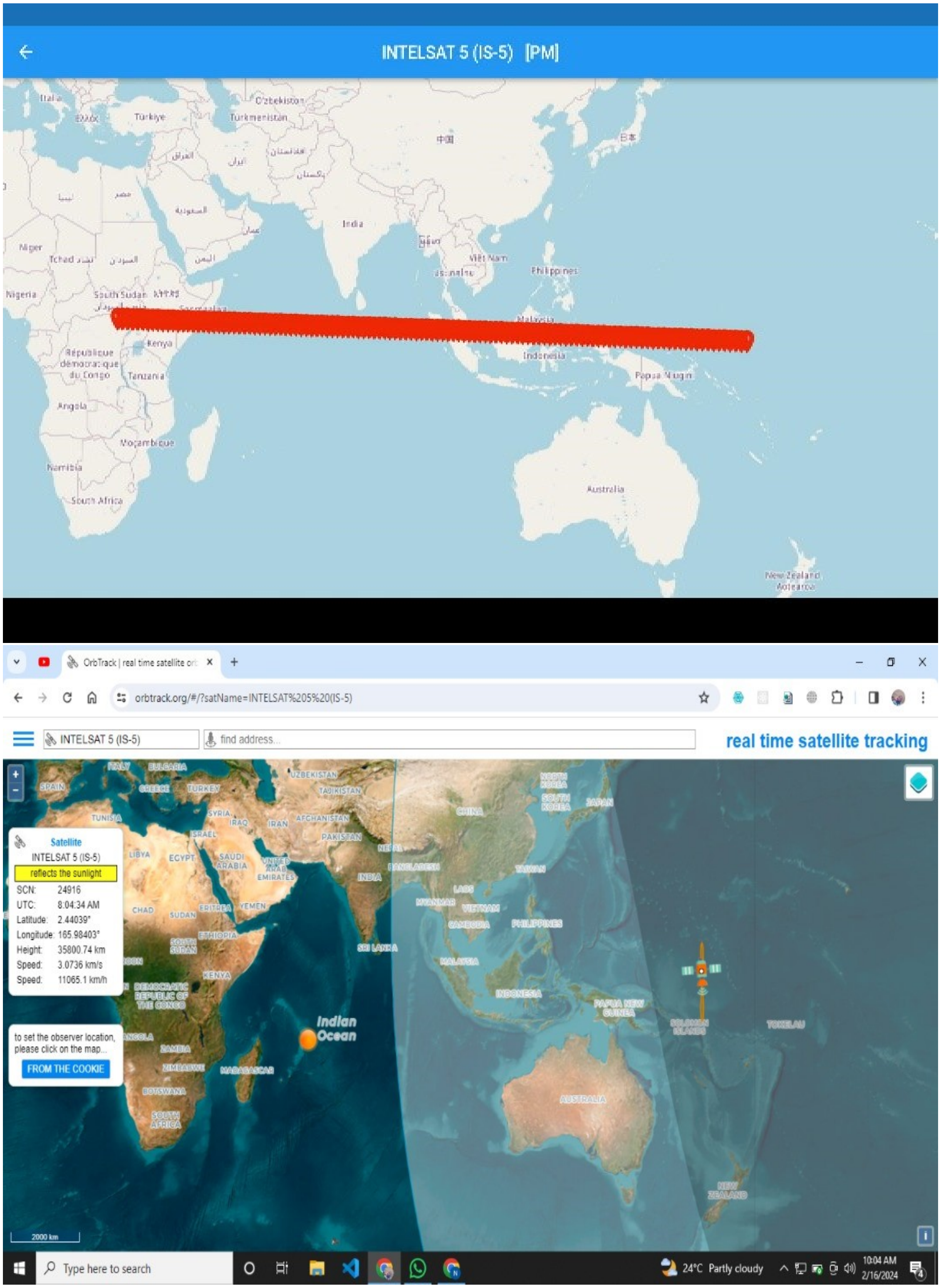
Observer

Latitude: -10.90756°
Longitude: -173.19598°
Local Time: 9:11:44 PM UTC+13

to the satellite

Azimuth: 295.73°
Elevation: 63.19°
Distance: 36371.7 km

Figure 4.10: The GALAXY 11 (G-11) Satellite's Predicted Path Can Be Seen On the Orbtrak Website and Mobile Application.

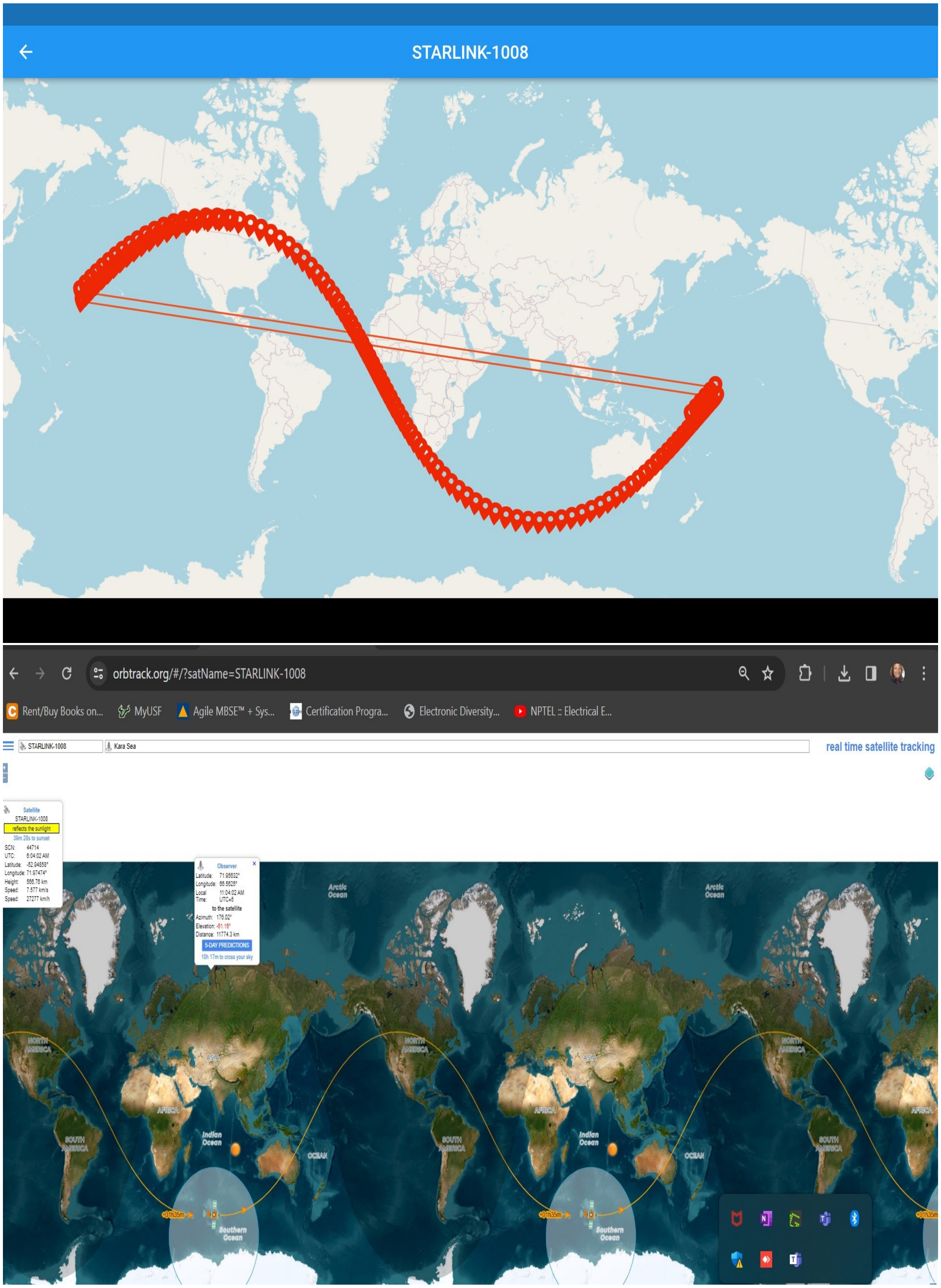


Satellite
INTELSAT 5 (IS-5)
reflects the sunlight

SCN:	24916
UTC:	8:04:34 AM
Latitude:	2.44039°
Longitude:	165.98403°
Height:	35800.74 km
Speed:	3.0736 km/s
Speed:	11065.1 km/h

to set the observer location, please click on the map...

Figure 4.11: The Predicted Path of the INTELSAT 5 (IS-5) Satellite for the Orbtrak Website and Mobile Application



Satellite
STARLINK-1008
reflects the sunlight
39m 28s to sunset
SCN: 44714
UTC: 6:04:02 AM
Latitude: -52.94858°
Longitude: 71.97474°
Height: 566.76 km
Speed: 7.577 km/s
Speed: 27277 km/h

Observer
Latitude: 71.95632°
Longitude: 66.5625°
Local Time: 11:04:02 AM UTC+5
to the satellite
Azimuth: 176.02°
Elevation: -61.16°
Distance: 11774.3 km
5-DAY PREDICTIONS
10h 17m to cross your sky

Figure 4.12: The STARLINK-1008 Seen On the Orbtrak Website and the Mobile Application

Chapter 5: Conclusions and Recommendations

5.1 Conclusions

In this work, a thorough feasibility study and a proposal for tracking satellites offline implementation were conducted. This project successfully achieved most of its objectives, demonstrating the application's capability to track satellites offline. However, satellite data is highly sensitive data that is not readily available on the internet for security reasons, and the SSN, which is the network that collects orbital information used to determine TLEs, is managed by NASA. There is no way to confirm if Celestrak TLEs are current and relevant. A comparison with Orbtrak, a website predicting satellite paths up to five days in advance, revealed discrepancies where some satellite paths were either undetermined or not listed on the website. Nevertheless, the application has effectively shown the latitude and, longitude of the satellite when a specific satellite was selected, thus validating the functionality of the SGP4 model for predicting satellite paths in a standalone offline application. However, the application does not allow satellite communication, which was one of the objectives of the thesis.

5.2 Recommendations and Future Work

Future versions of the application could provide more information on the screen such as the start and end points, timing information as well as a 3D model of the flight path.

References

[1] CelesTrak: "Orbital Coordinate Systems, Part I".

[2] Download now-android studio and sdk tools.

[3] Flutter documentation.

[4] Getting Started with Android Development.

[5] Migrating to null safety.

[6] SatOrbit - Apps on Google Play.

[7] An update on Eclipse Android Developer Tools.

[8] Ron Amadeo. Google's Dart language on Android aims for Java-free, 120 FPS apps, 2015.

[9] Timothy Carrico, John Carrico, Lisa Policastri, and Mike Loucks. Investigating orbital debris events using numerical methods with full force model orbit propagation. *Adv. Astronaut. Sci*, 130(PART 1):407–426, 2008.

[10] Dandan Fan, Benjin Li, and Ming Chen. Design of global emergency mobile communication system based on tdrss. In *2016 6th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 322–325. IEEE, 2016.

[11] Felix R Hoots and Ronald L Roehrich. *Models for propagation of NORAD element sets.* Office of Astrodynamics, 1980.

[12] Ted Kelso. Two-line element set check sum controversy.

[13] Max H Lane and Felix R Hoots. General perturbations theories derived from the 1965 lane drag theory, 1979.

[14] Max Herbert Lane and Kenneth Harold Cranford. *An improved analytical drag theory for the artificial satellite problem*. American Institute of Aeronautics and Astronautics, 1969.

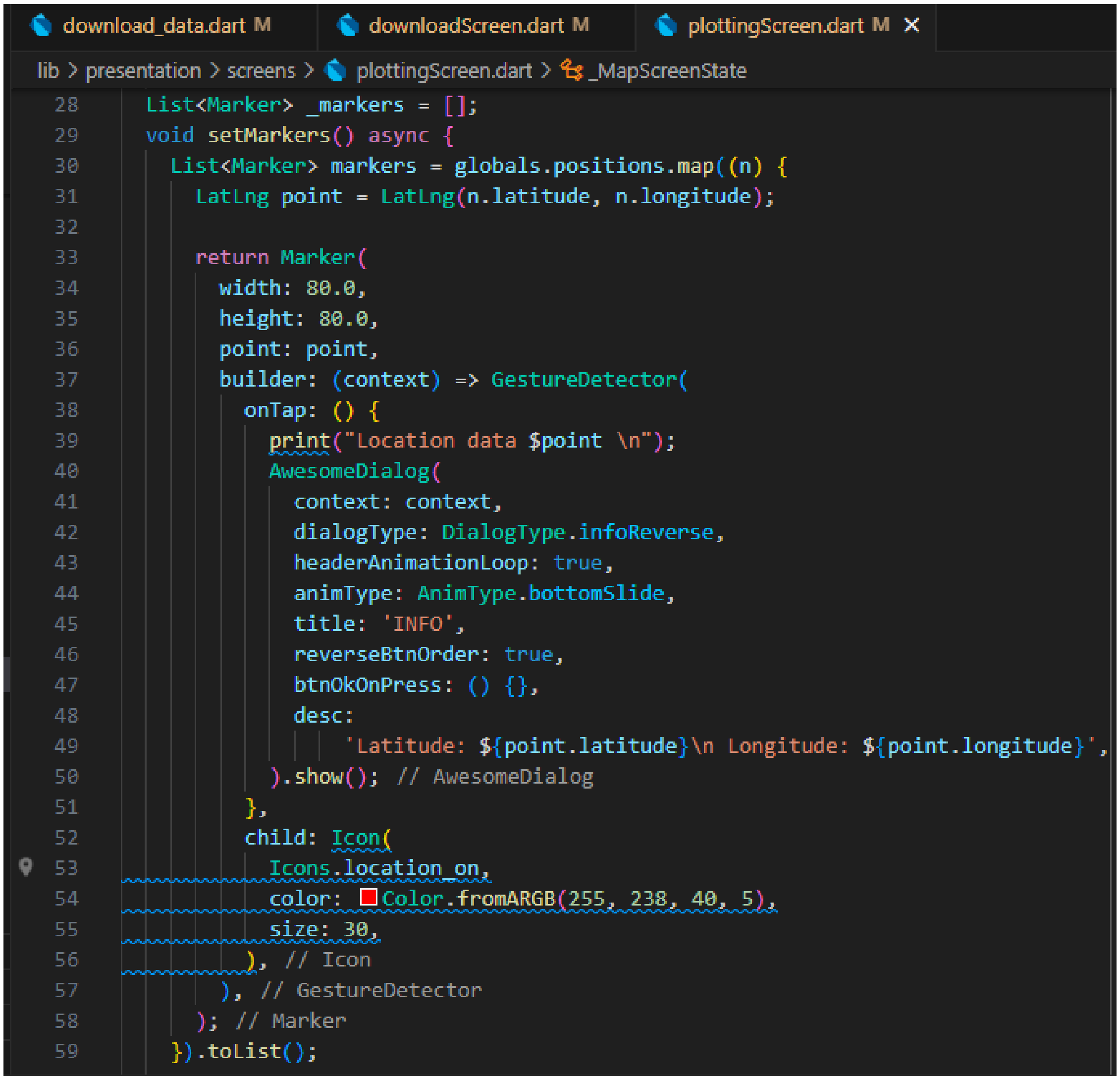
[15] June Lu, Chang Lu, Chun-Sheng Yu, and James E Yao. Exploring factors associated with wireless internet via mobile technology acceptance in mainland china. *Communications of the IIMA*, 3(1):9, 2003.

[16] Keerti Naregal, Anand Gudunavar, MC Ruiz, JA Mateo, H Macia, JJ Pardo, T Olivares, Mabel Mary Joy, Markus Becker, Wolfgang Mueller, et al. 2012 18th annual international conference on advanced computing and communications (adcom).

[17] David Pauleen, John Campbell, Brian Harmer, and Ali Intezari. Making sense of mobile technology: The integration of work and private life. *Sage Open*, 5(2):2158244015583859, 2015.

[18] Frank van Diggelen, Charlie Abraham, Javier de Salas, and Randy Silva. Gnss inside mobile phones: Gps, glonass, qzss, and sbas in a single chip. *Inside GNSS*, 6(2):50, 2011.

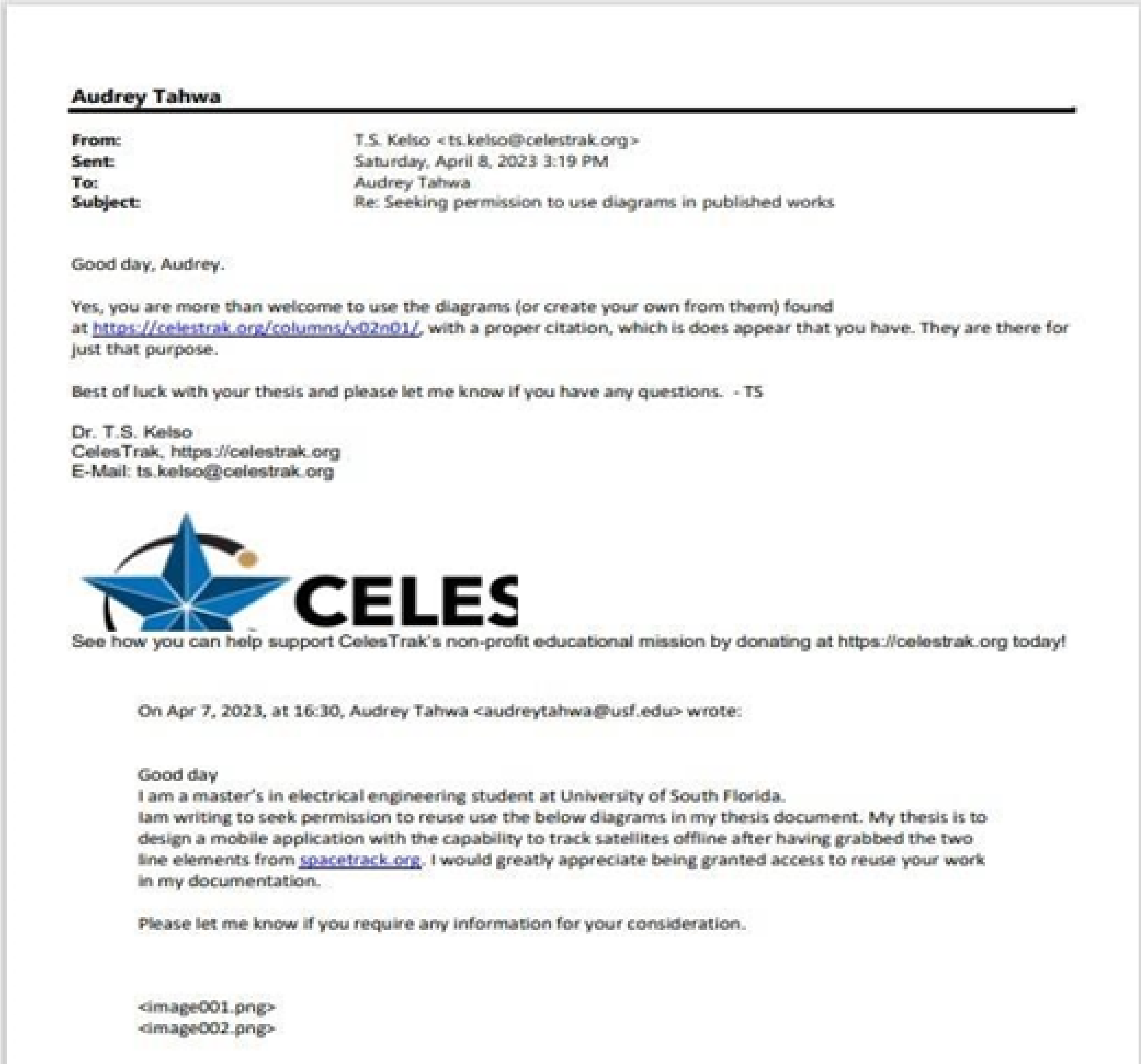
Appendix A: Snippet of Code



```dart
  List<Marker> _markers = [];
  void setMarkers() async {
    List<Marker> markers = globals.positions.map((n) {
      LatLng point = LatLng(n.latitude, n.longitude);

      return Marker(
        width: 80.0,
        height: 80.0,
        point: point,
        builder: (context) => GestureDetector(
          onTap: () {
            print("Location data $point \n");
            AwesomeDialog(
              context: context,
              dialogType: DialogType.infoReverse,
              headerAnimationLoop: true,
              animType: AnimType.bottomSlide,
              title: 'INFO',
              reverseBtnOrder: true,
              btnOkOnPress: () {},
              desc:
                  'Latitude: ${point.latitude}\n Longitude: ${point.longitude}',
            ).show(); // AwesomeDialog
          },
          child: Icon(
            Icons.location_on,
            color: Color.fromARGB(255, 238, 40, 5),
            size: 30,
          ), // Icon
        ), // GestureDetector
      ); // Marker
    }).toList();
```

Appendix B: Copyright Permissions

Below is the permission for Chapter 3 research methodology and source of Two Line Elements used in the application.



Audrey Tahwa

From: T.S. Kelso <ts.kelso@celestrak.org>
Sent: Saturday, April 8, 2023 3:19 PM
To: Audrey Tahwa
Subject: Re: Seeking permission to use diagrams in published works

Good day, Audrey.

Yes, you are more than welcome to use the diagrams (or create your own from them) found at https://celestrak.org/columns/v02n01/, with a proper citation, which is does appear that you have. They are there for just that purpose.

Best of luck with your thesis and please let me know if you have any questions. - TS

Dr. T.S. Kelso
CelesTrak, https://celestrak.org
E-Mail: ts.kelso@celestrak.org

See how you can help support CelesTrak's non-profit educational mission by donating at https://celestrak.org today!

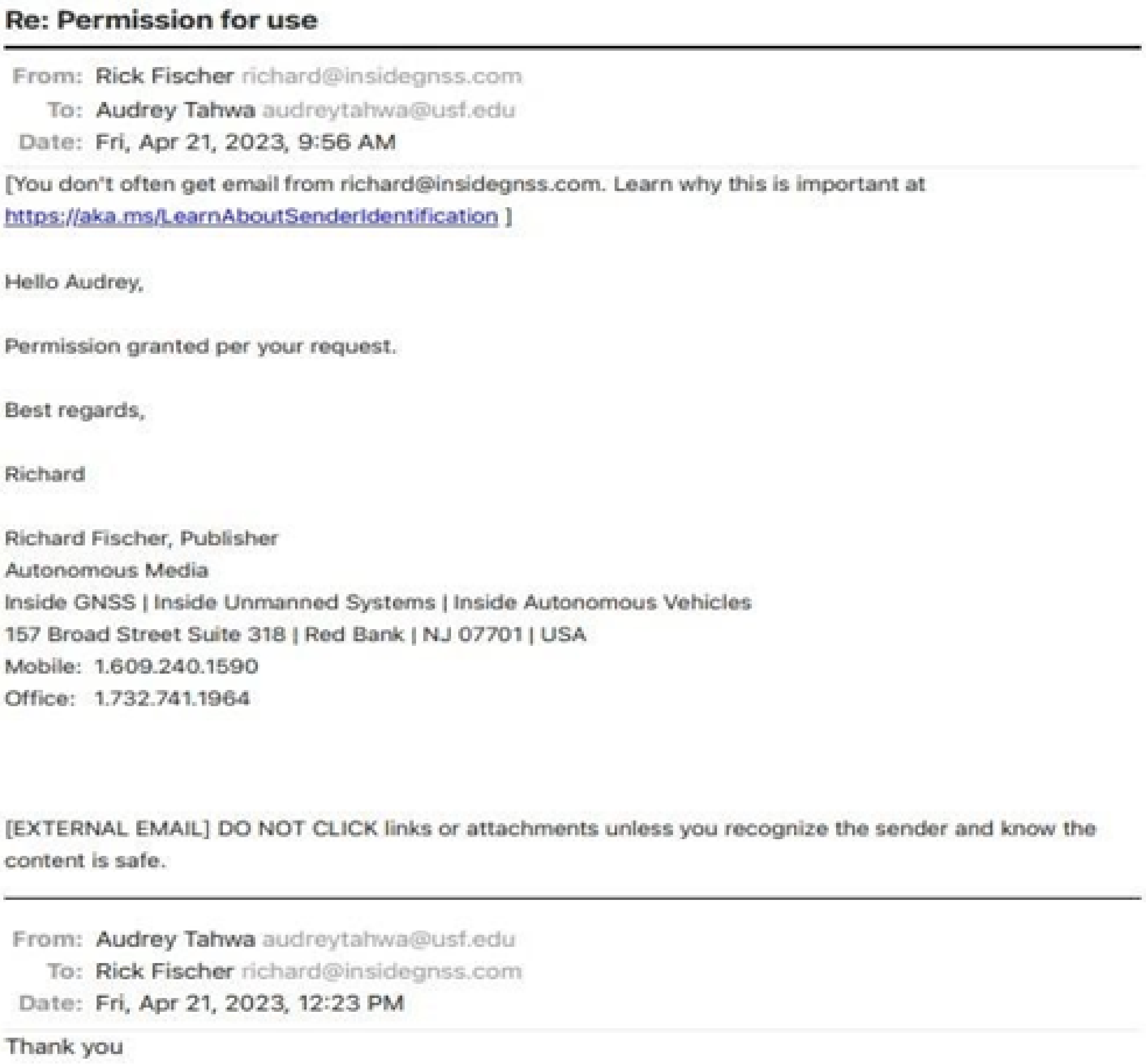
On Apr 7, 2023, at 16:30, Audrey Tahwa <audreytahwa@usf.edu> wrote:

Good day
I am a master's in electrical engineering student at University of South Florida.
Iam writing to seek permission to reuse use the below diagrams in my thesis document. My thesis is to design a mobile application with the capability to track satellites offline after having grabbed the two line elements from spacetrack.org. I would greatly appreciate being granted access to reuse your work in my documentation.

Please let me know if you require any information for your consideration.

<image001.png>
<image002.png>

Below is the permission for tables describing two line elements.



Re: Permission for use

From: Rick Fischer richard@insidegnss.com
To: Audrey Tahwa audreytahwa@usf.edu
Date: Fri, Apr 21, 2023, 9:56 AM

[You don't often get email from richard@insidegnss.com. Learn why this is important at https://aka.ms/LearnAboutSenderIdentification]

Hello Audrey,

Permission granted per your request.

Best regards,

Richard

Richard Fischer, Publisher
Autonomous Media
Inside GNSS | Inside Unmanned Systems | Inside Autonomous Vehicles
157 Broad Street Suite 318 | Red Bank | NJ 07701 | USA
Mobile: 1.609.240.1590
Office: 1.732.741.1964

[EXTERNAL EMAIL] DO NOT CLICK links or attachments unless you recognize the sender and know the content is safe.

From: Audrey Tahwa audreytahwa@usf.edu
To: Rick Fischer richard@insidegnss.com
Date: Fri, Apr 21, 2023, 12:23 PM

Thank you

Below is three pictures for permission for text in Chapter 3 describing satellites and all parameters its orbital information.

7/30/24, 1:23 AM AJER

e-ISSN No : 2320-0847
p-ISSN No : 2320-0936

Welcome to AJER

Published monthly, online, open-access and having double-blind peer reviewed, **American journal of Engineering Research (AJER)** is an emerging academic journal in the field of Engineering and Technology which deals with all facets of the field of Technology and Engineering. This journal motive and aim is to create awareness, re-shaping the knowledge already created and challenge the existing theories related to the field of Academic Research in any discipline in Technology and Engineering. **American journal of Engineering Research (AJER)** has a primary aim to publish novel research being conducted and carried out in the domain of Engineering and Technology as a whole. It invites engineering, professors, researchers, professionals, academicians and research scholars to submit their novel and conjectural ideas in the domain of Engineering and Technology in the shape of research articles, book reviews, case studies, review articles and personal opinions that can benefit the engineering and technology researchers in general and society as a whole.

As earlier stated, there is no special emphasis on just one topic but emphasis is given on all the aspects that can be covered in the domain of Engineering and Technology, so we welcome all the stakeholders of businesses, engineering, scientists and society as a whole to come forward with their new ideas and researches and submit us their articles from anywhere in the world. The name **American journal of Engineering Research (AJER)** does not just restrict our audience from any specific country but we are of view that research and knowledge is boundary less and keeping in view that perspective, we would welcome engineers, researchers, resource person, students, social scientists and critiques from all over the world.

For the ease of authors, we hereby are listing few main topics that can be considered in the scope of the journal.

- Chemical Technology
- Computer Science
- Software Engineering
- Construction
- Electrical and Nuclear Engineering
- Environmental Engineering
- Environmental Technology
- General and Civil Engineering
- Hydraulic Engineering
- Industrial Engineering

American journal of Engineering Research (AJER) is a peer reviewed, bimonthly official international journal allowing access to abstracts and
Read More....

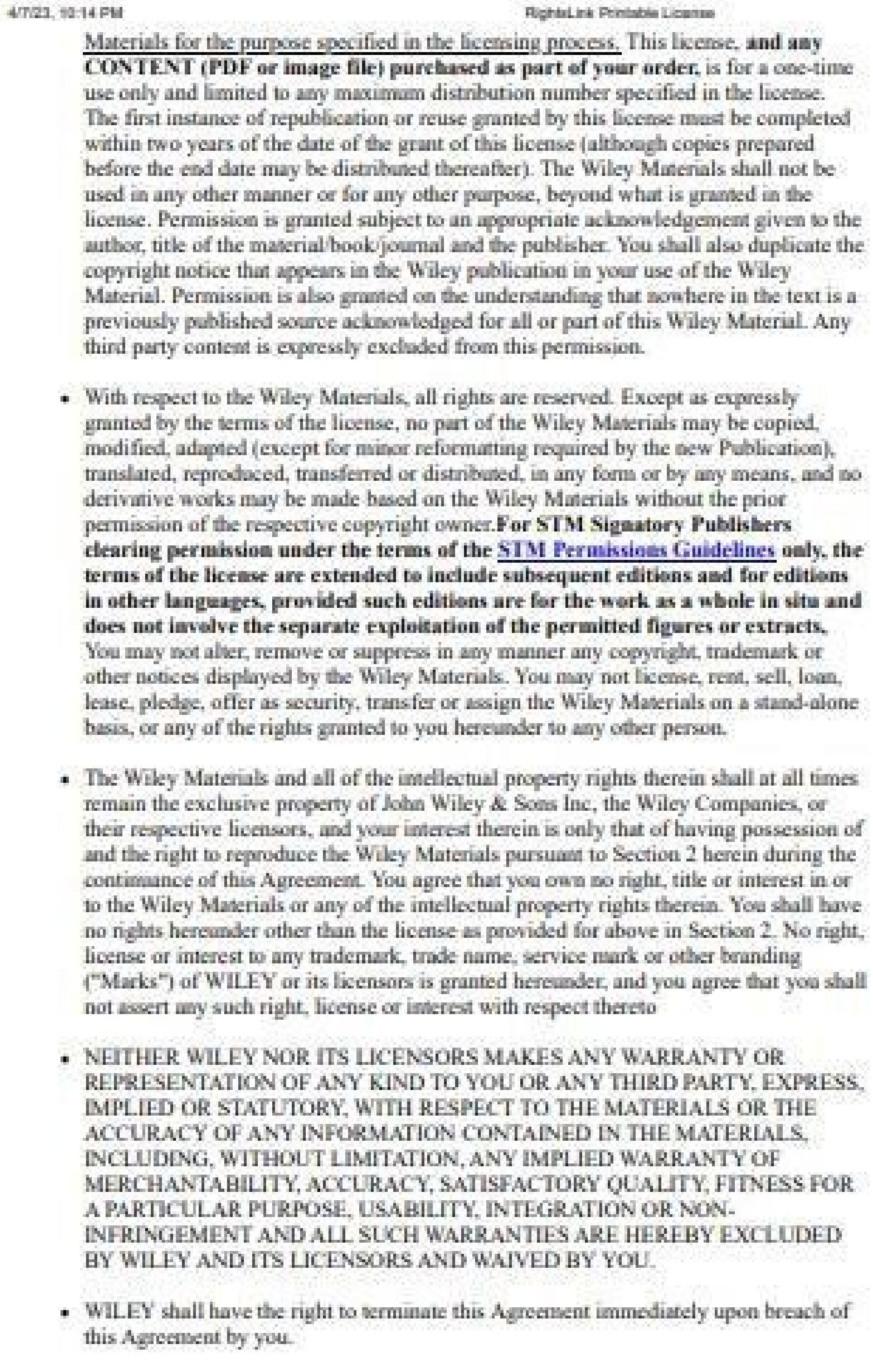
Email/SMS alert system is one of the key features and gives update information about the submitted manuscript.
Read More....

American journal of Engineering Research (AJER) is a peer reviewed, bimonthly official international journal allowing access to abstracts and
Read More....

American journal of Engineering Research (AJER) is a peer reviewed, bimonthly official international journal allowing access to abstracts and
Read More....

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License Copyright © 2012 AJER All rights reserved

https://www.ajer.org/about_us.html 1/1



Materials for the purpose specified in the licensing process. This license, **and any CONTENT (PDF or image file) purchased as part of your order**, is for a one-time use only and limited to any maximum distribution number specified in the license. The first instance of republication or reuse granted by this license must be completed within two years of the date of the grant of this license (although copies prepared before the end date may be distributed thereafter). The Wiley Materials shall not be used in any other manner or for any other purpose, beyond what is granted in the license. Permission is granted subject to an appropriate acknowledgement given to the author, title of the material/book/journal and the publisher. You shall also duplicate the copyright notice that appears in the Wiley publication in your use of the Wiley Material. Permission is also granted on the understanding that nowhere in the text is a previously published source acknowledged for all or part of this Wiley Material. Any third party content is expressly excluded from this permission.

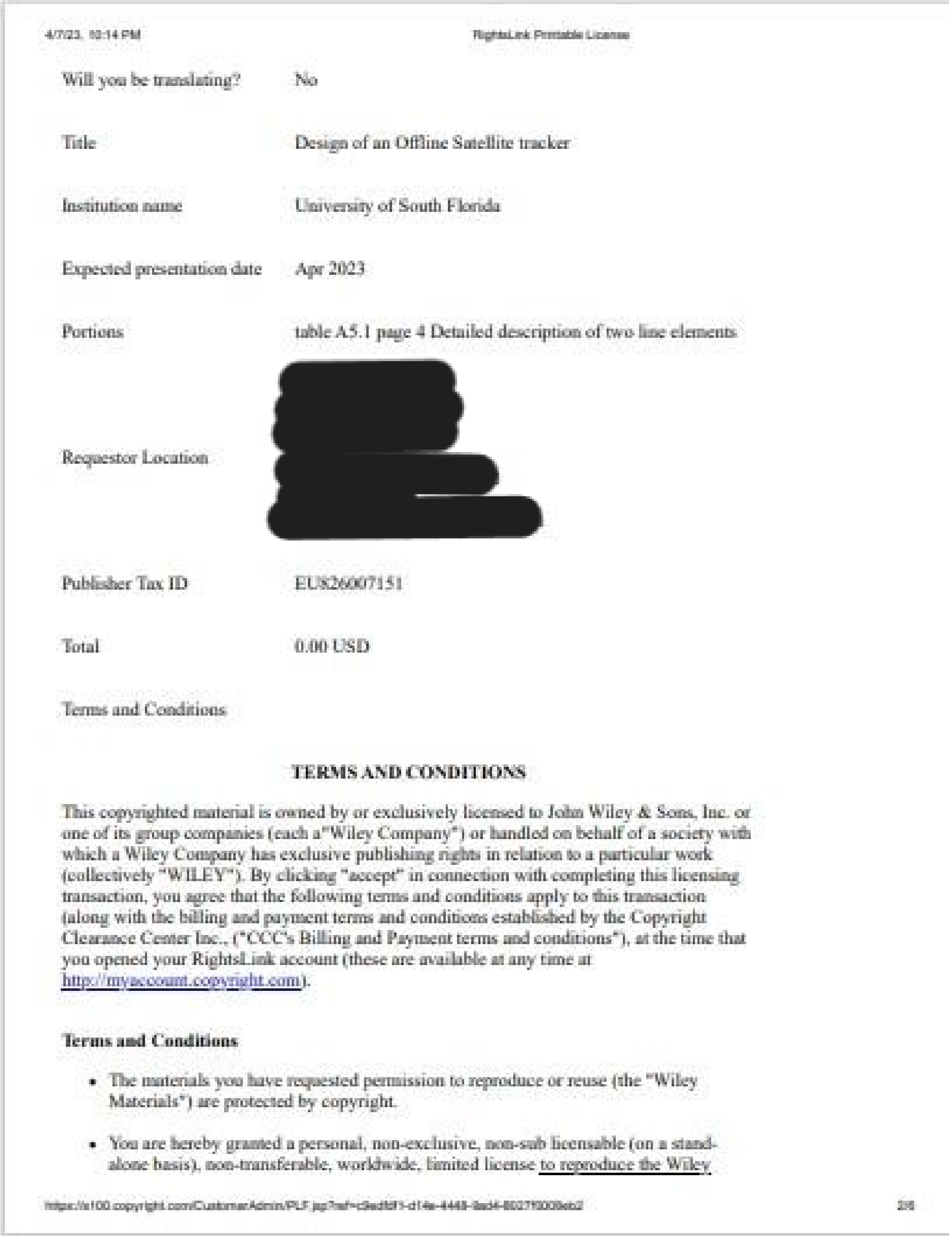
- With respect to the Wiley Materials, all rights are reserved. Except as expressly granted by the terms of the license, no part of the Wiley Materials may be copied, modified, adapted (except for minor reformatting required by the new Publication), translated, reproduced, transferred or distributed, in any form or by any means, and no derivative works may be made based on the Wiley Materials without the prior permission of the respective copyright owner.**For STM Signatory Publishers clearing permission under the terms of the STM Permissions Guidelines only, the terms of the license are extended to include subsequent editions and for editions in other languages, provided such editions are for the work as a whole in situ and does not involve the separate exploitation of the permitted figures or extracts,** You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley Materials. You may not license, rent, sell, loan, lease, pledge, offer as security, transfer or assign the Wiley Materials on a stand-alone basis, or any of the rights granted to you hereunder to any other person.

- The Wiley Materials and all of the intellectual property rights therein shall at all times remain the exclusive property of John Wiley & Sons Inc, the Wiley Companies, or their respective licensors, and your interest therein is only that of having possession of and the right to reproduce the Wiley Materials pursuant to Section 2 herein during the continuance of this Agreement. You agree that you own no right, title or interest in or to the Wiley Materials or any of the intellectual property rights therein. You shall have no rights hereunder other than the license as provided for above in Section 2. No right, license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto

- NEITHER WILEY NOR ITS LICENSORS MAKES ANY WARRANTY OR REPRESENTATION OF ANY KIND TO YOU OR ANY THIRD PARTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THE MATERIALS OR THE ACCURACY OF ANY INFORMATION CONTAINED IN THE MATERIALS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, USABILITY, INTEGRATION OR NON-INFRINGEMENT AND ALL SUCH WARRANTIES ARE HEREBY EXCLUDED BY WILEY AND ITS LICENSORS AND WAIVED BY YOU.

- WILEY shall have the right to terminate this Agreement immediately upon breach of this Agreement by you.

Below are three pictures for permission for figures and accompanying text on Latitude and latitude to ECI conversion in Chapter 2.



4/7/23, 10:14 PM RightsLink Printable License

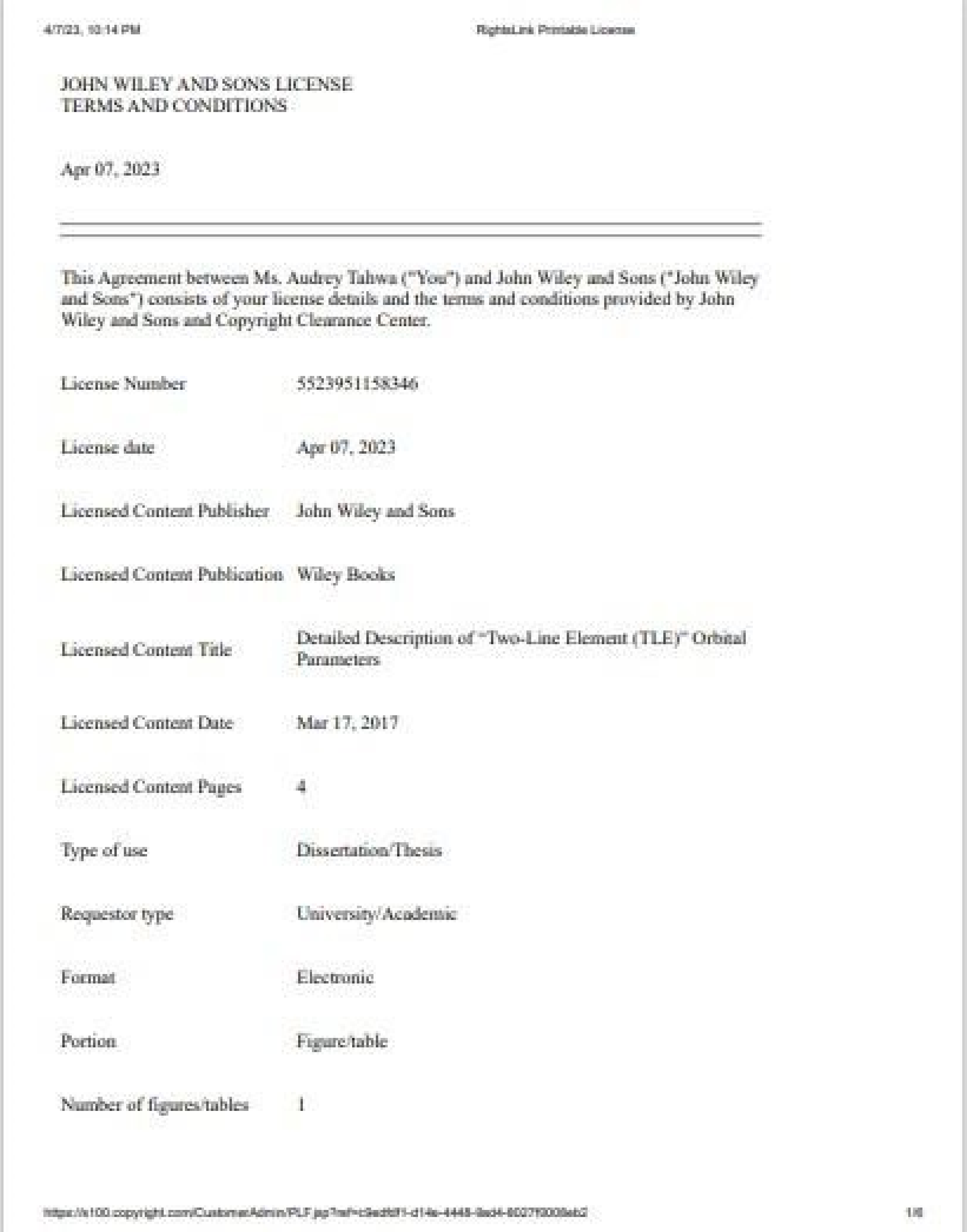
Will you be translating?	No
Title	Design of an Offline Satellite tracker
Institution name	University of South Florida
Expected presentation date	Apr 2023
Portions	table A5.1 page 4 Detailed description of two line elements
Requestor Location	[redacted]
Publisher Tax ID	EU826007151
Total	0.00 USD
Terms and Conditions	

TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a"Wiley Company") or handled on behalf of a society with which a Wiley Company has exclusive publishing rights in relation to a particular work (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your RightsLink account (these are available at any time at http://myaccount.copyright.com).

Terms and Conditions

- The materials you have requested permission to reproduce or reuse (the "Wiley Materials") are protected by copyright.
- You are hereby granted a personal, non-exclusive, non-sub licensable (on a stand-alone basis), non-transferable, worldwide, limited license to reproduce the Wiley

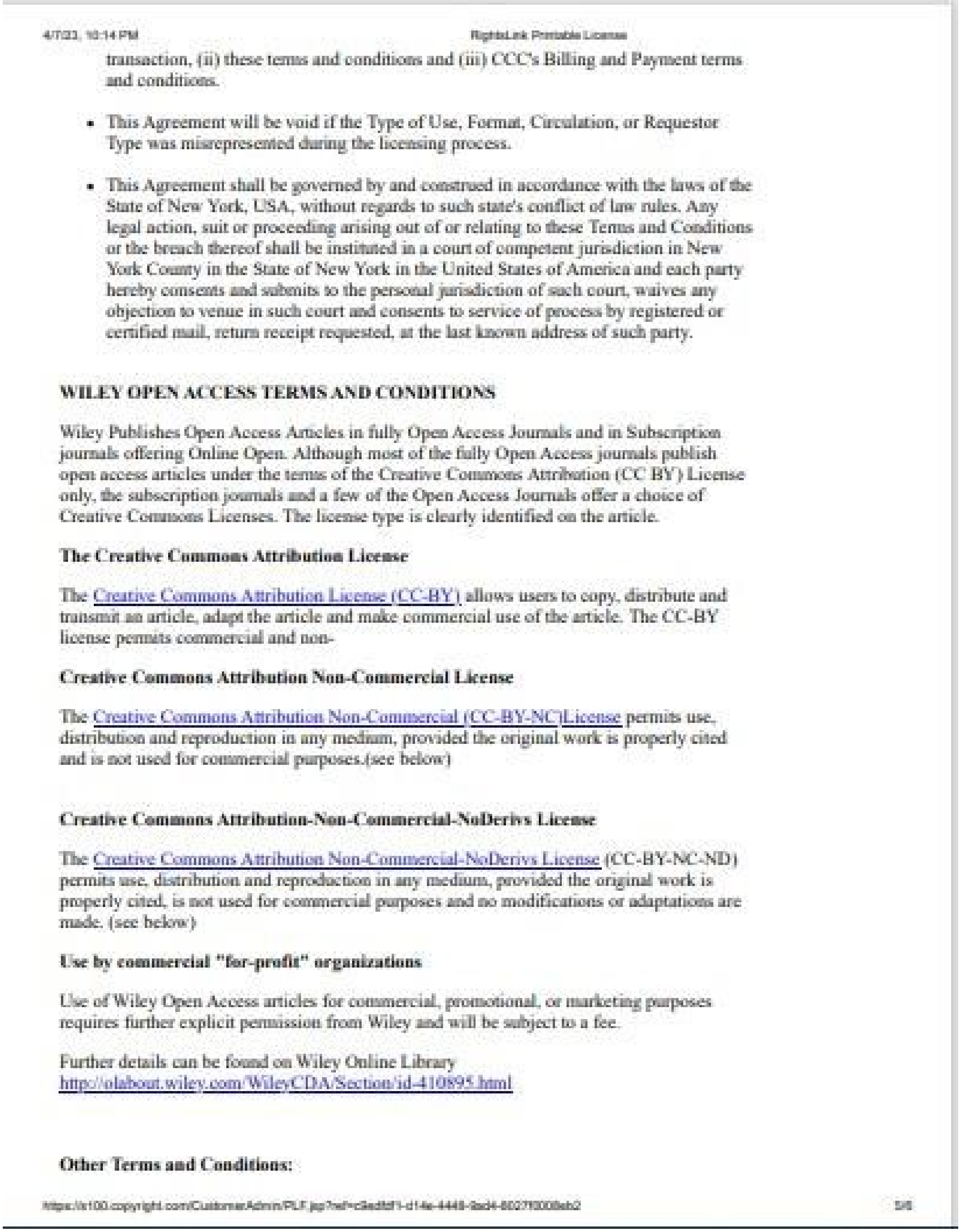


JOHN WILEY AND SONS LICENSE
TERMS AND CONDITIONS

Apr 07, 2023

This Agreement between Ms. Audrey Tahwa ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

License Number	5523951158346
License date	Apr 07, 2023
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	Wiley Books
Licensed Content Title	Detailed Description of "Two-Line Element (TLE)" Orbital Parameters
Licensed Content Date	Mar 17, 2017
Licensed Content Pages	4
Type of use	Dissertation/Thesis
Requestor type	University/Academic
Format	Electronic
Portion	Figure/table
Number of figures/tables	1



transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

- This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process.
- This Agreement shall be governed by and construed in accordance with the laws of the State of New York, USA, without regards to such state's conflict of law rules. Any legal action, suit or proceeding arising out of or relating to these Terms and Conditions or the breach thereof shall be instituted in a court of competent jurisdiction in New York County in the State of New York in the United States of America and each party hereby consents and submits to the personal jurisdiction of such court, waives any objection to venue in such court and consents to service of process by registered or certified mail, return receipt requested, at the last known address of such party.

WILEY OPEN ACCESS TERMS AND CONDITIONS

Wiley Publishes Open Access Articles in fully Open Access Journals and in Subscription journals offering Online Open. Although most of the fully Open Access journals publish open access articles under the terms of the Creative Commons Attribution (CC BY) License only, the subscription journals and a few of the Open Access Journals offer a choice of Creative Commons Licenses. The license type is clearly identified on the article.

The Creative Commons Attribution License

The Creative Commons Attribution License (CC-BY) allows users to copy, distribute and transmit an article, adapt the article and make commercial use of the article. The CC-BY license permits commercial and non-

Creative Commons Attribution Non-Commercial License

The Creative Commons Attribution Non-Commercial (CC-BY-NC)License permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.(see below)

Creative Commons Attribution-Non-Commercial-NoDerivs License

The Creative Commons Attribution Non-Commercial-NoDerivs License (CC-BY-NC-ND) permits use, distribution and reproduction in any medium, provided the original work is properly cited, is not used for commercial purposes and no modifications or adaptations are made. (see below)

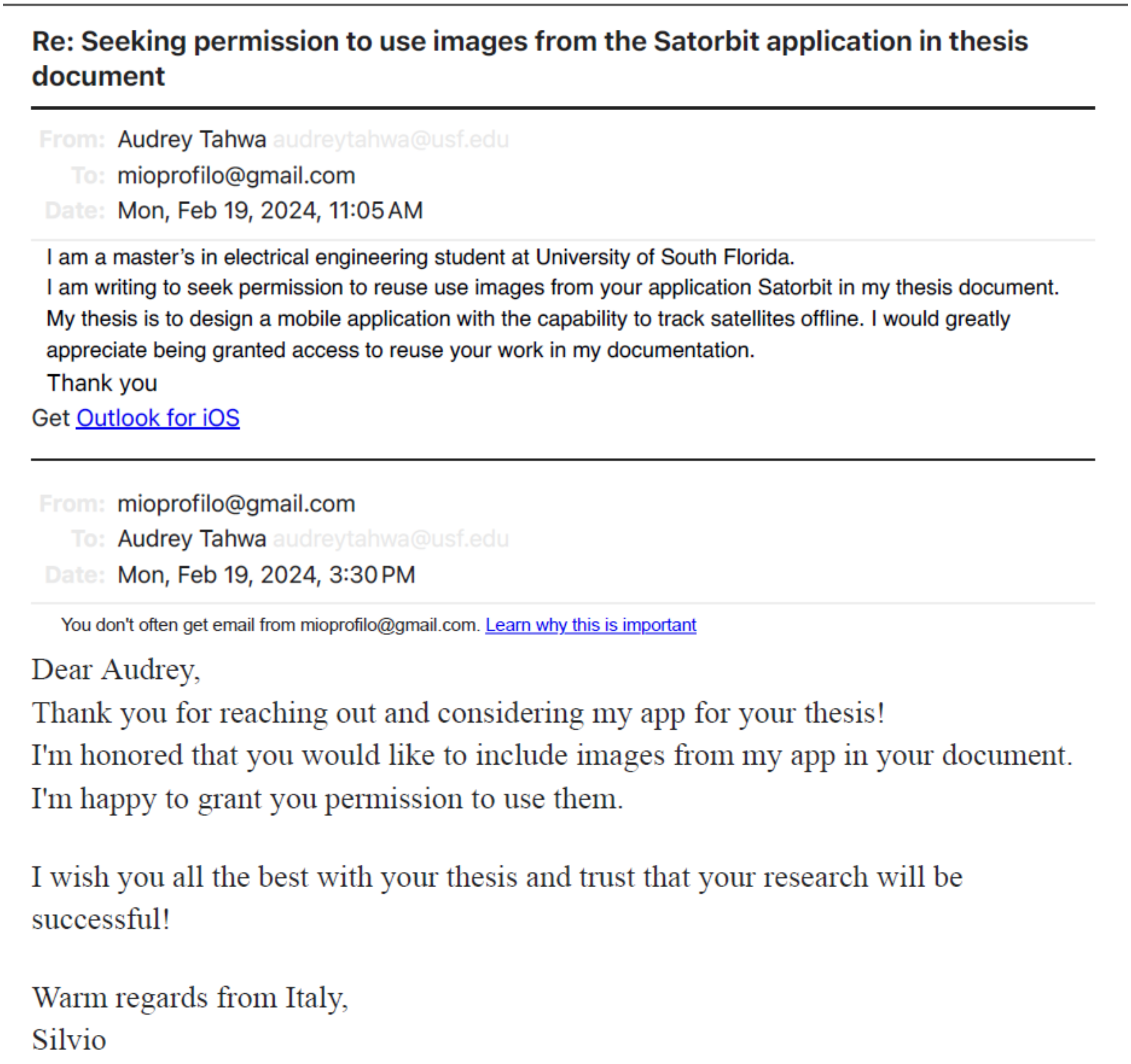
Use by commercial "for-profit" organizations

Use of Wiley Open Access articles for commercial, promotional, or marketing purposes requires further explicit permission from Wiley and will be subject to a fee.

Further details can be found on Wiley Online Library http://olabout.wiley.com/WileyCDA/Section/id-410895.html

Other Terms and Conditions:

Below is the permission for use Satorbit app images in chapter 3.



Re: Seeking permission to use images from the Satorbit application in thesis document

From: Audrey Tahwa audreytahwa@usf.edu
To: mioprofilo@gmail.com
Date: Mon, Feb 19, 2024, 11:05 AM

I am a master's in electrical engineering student at University of South Florida.
I am writing to seek permission to reuse use images from your application Satorbit in my thesis document.
My thesis is to design a mobile application with the capability to track satellites offline. I would greatly appreciate being granted access to reuse your work in my documentation.
Thank you
Get Outlook for iOS

From: mioprofilo@gmail.com
To: Audrey Tahwa audreytahwa@usf.edu
Date: Mon, Feb 19, 2024, 3:30 PM

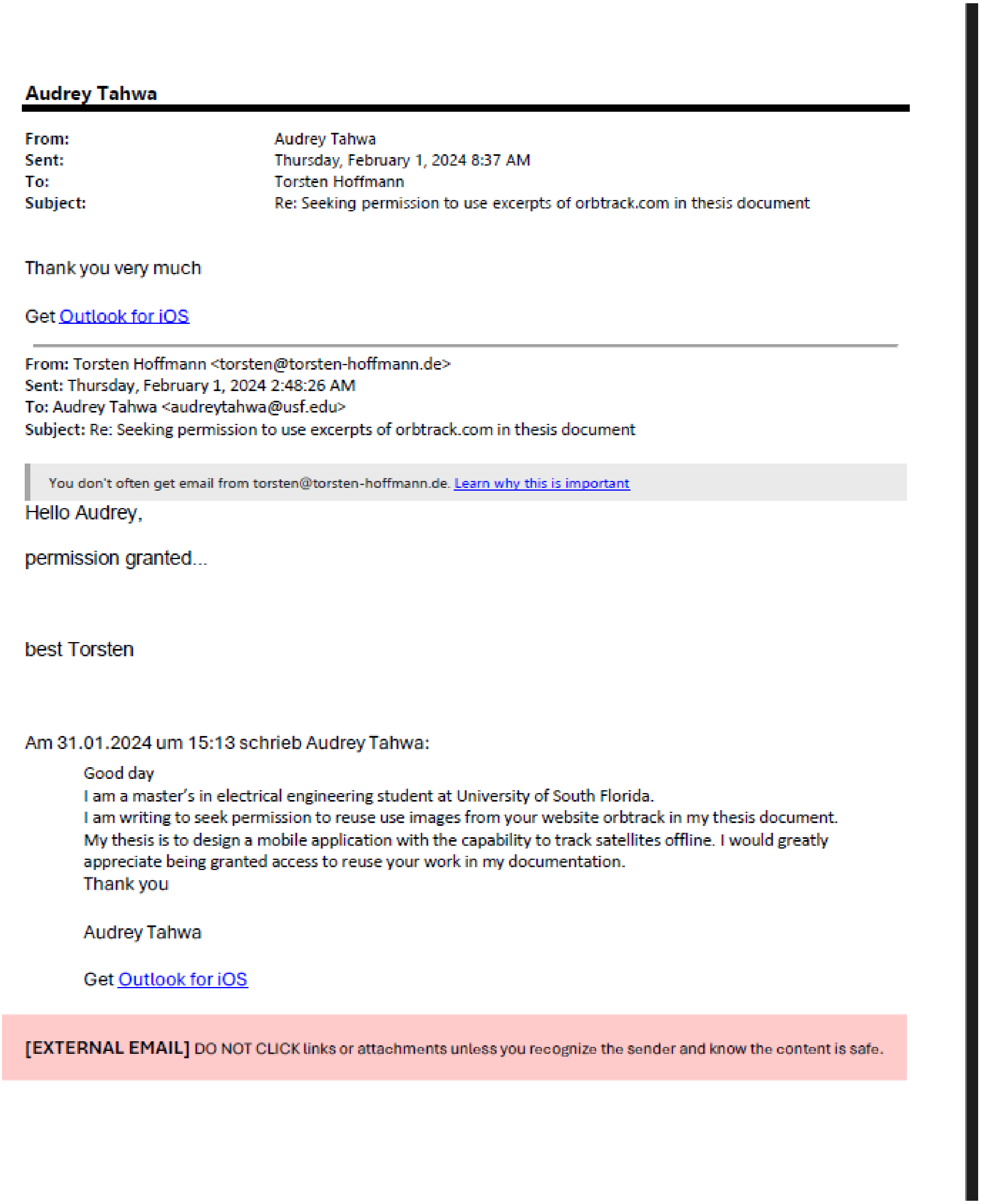
You don't often get email from mioprofilo@gmail.com. Learn why this is important

Dear Audrey,
Thank you for reaching out and considering my app for your thesis!
I'm honored that you would like to include images from my app in your document.
I'm happy to grant you permission to use them.

I wish you all the best with your thesis and trust that your research will be successful!

Warm regards from Italy,
Silvio

Below is the permission for use Obtrack website app images in chapter 4 when comparing results and analysis



Audrey Tahwa

From:	Audrey Tahwa
Sent:	Thursday, February 1, 2024 8:37 AM
To:	Torsten Hoffmann
Subject:	Re: Seeking permission to use excerpts of orbtrack.com in thesis document

Thank you very much

Get [Outlook for iOS]()

From: Torsten Hoffmann <torsten@torsten-hoffmann.de>
Sent: Thursday, February 1, 2024 2:48:26 AM
To: Audrey Tahwa <audreytahwa@usf.edu>
Subject: Re: Seeking permission to use excerpts of orbtrack.com in thesis document

You don't often get email from torsten@torsten-hoffmann.de. [Learn why this is important]()

Hello Audrey,

permission granted...

best Torsten

Am 31.01.2024 um 15:13 schrieb Audrey Tahwa:

> Good day
> I am a master's in electrical engineering student at University of South Florida.
> I am writing to seek permission to reuse use images from your website orbtrack in my thesis document.
> My thesis is to design a mobile application with the capability to track satellites offline. I would greatly appreciate being granted access to reuse your work in my documentation.
> Thank you
>
> Audrey Tahwa
>
> Get [Outlook for iOS]()

[EXTERNAL EMAIL] DO NOT CLICK links or attachments unless you recognize the sender and know the content is safe.

Appendix C: Glossary

Complete list of symbols used in the Satellite perturbation models:

- e_o - the "mean" eccentricity at epoch
- i_o - the "mean" inclination at epoch
- M_o - the "mean" anomaly at epoch
- w_o - the "mean" argument of perigee at epoch
- $omega_o$ - the "mean" longitude of ascending node at epoch
- B^* - the SGP4 type drag coefficient
- k_e - $(GM) * \frac{1}{2}$ where G is Newton's universal gravitational constant and M is the mass of the earth
- a_E - the equatorial radius of the earth
- J_2 - the second gravitational zonal harmonic of the earth
- J_3 - the third gravitational zonal harmonic of the earth
- J_4 - the fourth gravitational zonal harmonic of the earth
- $(t - t_o)$ - time since epoch
- $k_2 - \frac{1}{2} J_2 a_E^2$
- $k_4 - \frac{3}{8} J_4 a_E^4$

- $A_{3,0} - J_3 a_E^3$
- q_o - parameter for the SGP4 density function
- n_o - the SGP type ”mean” motion at epoch
- s - parameter for the SGP4 density function

About the Author

Audrey Tahwa is a holder of a Bachelor of Technology in Electronic Engineering from Harare Institute of Technology working towards a Master of Science in the Department of Electrical Engineering. She is also a technology enthusiast.