**Stakeholders:**

**1. Primary Mobile Device Users: General users seeking to secure their devices.**

**2. Corporate and Enterprise Users : Protecting organization-wide devices.**

**3. Mobile Device and App Developers : Ensuring apps meet security standards, Certifying app security before launch.**

4. Educational and Research Institutions Cybersecurity Programs: Training students with hands-on malware detection.

5. Legal and Compliance Teams: Ensuring compliance with data privacy regulations.

**User Story:**

The **Malicious App Scanner** application is envisioned as a comprehensive mobile security solution, developed to protect mobile devices from malicious applications, malware, and privacy risks. This app addresses two primary groups of stakeholders: **Primary Mobile Device Users**, who are general users aiming to secure their personal devices against potential threats, and **Corporate and Enterprise Users**, who require advanced solutions to safeguard multiple organization-owned devices, ensuring the protection of sensitive corporate data and compliance with regulatory standards.

#### **1. Primary Mobile Device Users**

* **General Users**: Everyday users who may not have extensive knowledge of cybersecurity seek a reliable, straightforward way to secure their mobile devices. These users often lack the technical expertise to detect malicious behaviors on their own but are concerned about the safety of their personal data, privacy, and device integrity. For these users, the app is designed as an intuitive, user-friendly tool to help them keep their devices safe without the need for complex settings.
* **User-Friendly Features**: The app’s user interface prioritizes simplicity, offering a **one-click scan** feature for a quick security check that thoroughly examines all installed apps. After each scan, users receive a clear report, highlighting any detected risks and providing easy-to-follow recommendations to take immediate action if needed. The app also includes an **app permission and intent analysis** feature, which allows users to view detailed information about each app’s permissions, revealing potential risks such as access to location, contacts, microphone, or other sensitive data. Through this, users can control their device’s privacy without needing to interpret complex technical data.
* **Customized Security Controls**: For those looking to streamline security, the app offers **app whitelisting and blacklisting options**. Users can mark trusted apps, which are automatically considered safe and ignored during routine scans, focusing attention on unknown or newly downloaded apps that might pose risks.

#### **2. Corporate and Enterprise Users**

* **Organizational Device Security Needs**: Corporate and enterprise clients, who often distribute mobile devices to employees for work purposes, require a scalable security solution that is easy to manage across numerous devices. This app provides tools specifically tailored to the security needs of companies, allowing administrators to monitor, assess, and secure each device in real time. By offering an **organization-wide scheduled scan feature**, corporate users can schedule regular scans across all employee devices to ensure continuous security, actively identifying and mitigating any potential threats.
* **Compliance and Policy Enforcement**: The scanner is designed to help companies adhere to data privacy and security regulations like GDPR and CCPA by continuously monitoring app permissions and detecting excessive access requests, a critical feature for industries that handle sensitive client or company information. Through **policy-aligned app whitelisting and blacklisting**, administrators can maintain control over which apps employees can access, ensuring company devices are only running secure, trusted applications. The app’s integrated **audit and reporting system** logs all detected threats, permission requests, and device actions, providing a historical view that can support compliance audits, risk assessments, and forensic analysis if necessary.
* **Data Privacy and Threat Management**: For corporate users, device security goes beyond threat detection to include proactive protection. The app empowers companies to enforce data privacy policies by providing real-time updates on new apps installed across devices and issuing **immediate notifications** when any app attempts to access sensitive data. The app is also designed to support **centralized device management**, allowing corporate security teams to configure and enforce security policies remotely. The goal is to maintain a unified security standard across all devices in the organization, minimizing data breach risks and providing peace of mind to both users and stakeholders.

#### **3. Mobile Device and App Developers**

* **Security Standards Compliance**: App developers require tools to evaluate and ensure their applications meet industry security standards. By scanning their apps through this tool, they can identify potential vulnerabilities, assess permissions usage, and ensure compliance with security best practices before release.
* **Pre-Launch Security Certification**: Developers can test their applications for malware, overreaching permissions, and suspicious intents. The **permissions and intent analysis** feature provides insights into data requests, helping developers adjust permissions to align with user expectations and privacy needs.
* **Malware Reporting**: The app generates a **pre-launch security report** detailing any potential risks or vulnerabilities, serving as a valuable tool for development teams focused on releasing secure applications. This report allows them to resolve issues early and achieve a higher level of trust with users and platform providers.

### **Key Features Supporting All Users**

1. **Malicious App Detection**: The core functionality of the app, powered by a machine-learning model trained to identify malicious and benign apps, is designed to provide accurate threat detection without impacting device performance. The scanner analyzes apps for harmful code, suspicious permissions, and abnormal behavior.
2. **Permission & Intent Analysis**: For both individual users and organizations, this feature allows full transparency into each app’s permissions, helping users understand which apps access sensitive information. Users can make informed decisions about which permissions to allow or deny, thus retaining control over their personal and corporate data.
3. **One-Click Scan and Scheduled Scans**: The app provides a single-click option for an instant, full scan and also supports automated scheduled scans (daily, weekly, monthly), ensuring continuous monitoring and up-to-date protection.
4. **Centralized Whitelisting and Blacklisting**: This feature lets users mark apps as trusted or risky based on past scans and policy requirements. Enterprises can enforce a company-wide app policy by managing which apps employees can access on their devices.
5. **Comprehensive Threat Reports**: The app generates detailed reports for detected threats, offering insights into the type of risk, affected permissions, and recommended actions. For corporate users, these reports serve as valuable documentation to support internal security reviews and compliance efforts.

### **Conclusion**

The **Malicious App Scanner** provides a powerful security solution that caters to the specific needs of **Primary Mobile Device Users**, **Corporate and Enterprise Users**, and **Mobile Device and App Developers**. It empowers general users to secure their devices and data, supports organizations in enforcing compliance and protecting corporate data, and assists developers in building safer, more secure apps. Designed with these diverse needs in mind, this app ensures a secure, user-friendly experience that prioritizes device protection, regulatory compliance, and development transparency, benefiting both individual users and large-scale organizations alike.

The stakeholders of our project are the android users who use different android devices. Here they want to have the application file scanned whether they are safe or not. There are many apps on our devices which may steal our information through different ways. So for the security of our information, we want to have our devices safe from these threats.

For a malicious app scanner application, users have specific needs for maintaining their device security, each targeting different functionalities to protect against potentially harmful apps. **General users** are primarily concerned with having a straightforward way to scan all installed apps to detect and remove any that may be malicious. They look for a **one-click scan feature** that initiates a complete, on-demand check to promptly assess the safety of their devices. These users also seek **permission monitoring** capabilities to understand which apps are accessing sensitive data like contacts, location, or camera, helping them identify and manage unnecessary or risky permissions. In addition to ad-hoc scans, security-conscious users want the ability to schedule scans at intervals (daily, weekly) so that the app can automatically check for newly installed or updated apps, allowing them to maintain a consistently safe device environment.

Advanced features cater to users who wish to control scan behavior and notifications further. A **whitelist feature** is crucial for users who regularly use trusted apps, enabling them to mark certain apps as safe and reduce scan time by focusing only on unvetted or new apps. For proactive security, users want **real-time threat notifications** when any app is flagged as suspicious, allowing them to take immediate action to remove or investigate the app in question.

Beyond basic scanning, users appreciate access to **detailed threat reports** that outline the type of threat, risk level, and recommended actions for any identified risks, empowering them to make informed decisions about their app usage. Security analysts or technically savvy users benefit from **advanced scanning options**, including deeper insights into permissions and intent analysis, to identify behaviors that might not initially appear malicious but indicate potential misuse of data or functions. Meanwhile, administrators maintaining the app itself require the ability to update the app’s internal **malware signature database** regularly, ensuring the scanner remains effective against the latest threats.

### **Software Requirements Specification (SRS) Story for Malicious App Detection Software**

The **Malicious App Detection Software** is envisioned as a robust security solution designed to protect mobile and enterprise environments from malicious applications, malware, and privacy risks. Developed for multiple user segments, it supports **Primary Mobile Device Users** who want to secure their devices from potential threats, **Corporate and Enterprise Users** who need advanced tools to manage security across multiple organization-owned devices, and **Mobile Device and App Developers** who aim to meet security standards before launching applications.

#### **1. Primary Mobile Device Users**: For general users who may not have in-depth cybersecurity knowledge, this software offers an accessible and reliable way to safeguard personal devices. This group is primarily concerned with protecting personal data and privacy while ensuring their devices remain free from malware. The software offers an **intuitive interface** that simplifies security for non-technical users, allowing them to keep devices secure without needing advanced technical expertise.

* **User-Friendly Features**: With a **one-click scan function** for quick and comprehensive security checks, users can easily scan installed applications and receive detailed report. The software’s **permission and intent analysis** feature reveals the access permissions of each app, helping users make informed decisions about privacy risks, such as access to contacts, location, and other sensitive data.
* **Customized Security Controls**: Users can control the software’s focus through **app whitelisting and blacklisting** options. This allows them to designate trusted apps to be ignored during scans, while unknown or new apps receive prioritized attention.

#### **2. Corporate and Enterprise Users**

* **Organizational Device Security Needs**: Designed for organizations that distribute mobile devices to employees, this software provides a scalable solution to monitor and secure multiple devices. Companies can conduct regular, scheduled scans on all devices, ensuring proactive threat detection and centralized device security.
* **Compliance and Policy Enforcement**: To support industries that manage sensitive data, this software helps organizations enforce compliance with regulations such as GDPR and CCPA. Through **automated permission monitoring** and **centralized whitelisting and blacklisting**, companies maintain control over app permissions, allowing only secure, approved applications to run on their devices. An **audit and reporting system** provides detailed logs on detected threats, permission requests, and security actions, which are essential for compliance documentation, risk assessments, and forensic analysis.
* **Real-Time Threat Management**: Beyond detection, this software enables companies to act on security risks quickly. It issues **immediate notifications** when suspicious activity occurs, and the centralized management feature allows administrators to configure and enforce security protocols remotely across all company devices, reducing the risk of data breaches.

#### **3. Mobile Device and App Developers**

* **Security Standards Compliance**: Developers can scan their applications to ensure compliance with security standards, identifying and addressing potential vulnerabilities and permission issues before release.
* **Pre-Launch Security Certification**: By analyzing permissions and detecting suspicious intents, the software helps developers prepare their applications for secure launches, adjusting permissions to meet privacy expectations.

### **Key Features Supporting All Stakeholders**

1. **Malicious App Detection**: At the core of the software is a machine-learning-driven detection model that accurately identifies threats without slowing down device performance.
2. **Permission & Intent Analysis**: This feature offers insights into each app’s permissions, empowering users and administrators to make informed privacy and security decisions.
3. **One-Click Scan and Scheduled Scans**: Provides a quick scan option and allows users to automate regular scans, ensuring continuous monitoring.
4. **Centralized Whitelisting and Blacklisting**: Users can designate apps as safe or risky, while organizations can enforce policies organization-wide.
5. **Detailed Threat Reports**: Reports include detected threats, permissions flagged as risky, and recommended actions, supporting individual and corporate users in compliance and internal reviews.

### **Conclusion**

The **Malicious App Detection Software** is a comprehensive security solution that caters to **Primary Mobile Device Users**, **Corporate and Enterprise Users**, and **Mobile Device and App Developers**. By addressing specific security needs, this software enables general users to protect their data, helps organizations enforce compliance and secure sensitive information, and provides developers with tools to enhance application security. With user-friendly interfaces, detailed reporting, and centralized management capabilities, it supports both individual and enterprise-level protection, ensuring broad and effective security coverage.

**Features:**

1. Malicious App Detection (Both)
2. APK Extraction (Both)
3. Permission & Intent Viewer (App Developer)
4. User-Dedicated Database (Both)
5. User Dashboard (Both)
6. Overall Detailed Report (App Developer)

**Malicious App Detection:**

Benign App Data Collection:

* Extract .apk file.
* Analyze permissions and intents, storing them in a database. (App Developer)

Malicious App Data Collection:

* Source data from security research platforms.
* Extract and filter relevant data based on similarities with benign apps.
* Combine malicious and benign data in one database .

Model Training:

* Train a machine learning model to classify apps as malicious or benign.
* Update the database with new data.

**Exciting Features:**

* Multiple Scan Options: Full scan, quick scan. (Both)
* Custom App Management: Whitelist trusted apps, blacklist risky ones, and block known malicious apps using a predefined list. (Primary Mobile User)

**User Story :**

The **Droid Scanner: App Security Management System** is designed as a robust software solution, safeguarding mobile devices from potential threats and ensuring high-security standards for app development.

This system serves two primary stakeholder groups: **Primary Mobile Device Users**, who prioritize device security, and **Mobile App Developers**, who need tools to ensure their apps are free from vulnerabilities and comply with privacy standards.

The software provides flexible scan options, offering a **Full Scan** : A comprehensive scan of all the installed apps and a **Quick Scan** : scan a particular App on command. All the forms ensure security monitoring.

For Primary Mobile Device Users, the system operates as a personal security *tool*. Users initiate the setup by accessing their dedicated User Dashboard, which displays the user profile, previous scans history, and black & white applists. The system's Malicious App Detection feature performs in-depth scans to identify harmful apps by analyzing APK files. The APK Extraction feature allows users to save or examine APK files. With the Custom App Management feature, users can whitelist trusted apps to avoid redundant checks and blacklist risky apps. All scans and their outcomes are stored securely in a User Database, which enables easy access to past reports, scan summaries, helping users stay informed about potential risks and ensure data privacy.

For Mobile App Developers, the system serves as an essential security and compliance tool to evaluate and improve app safety before launch. Developers access a dashboard that enables them to view scan reports. The system’s APK Extraction and Permissions tools let developers save and review copies of their apps, making sure permissions like location or camera are only used when necessary. The Permission Viewer shows every permission request in detail, helping developers easily choose which to keep or remove for better user privacy. After each scan, the system generates a comprehensive report detailing **app details, permissions used, intents, security status, feature analysis and API calls , helping developers align their app with industry standards**. Additionally, each developer has access to a secure User Database where all detected issues, past scans, and security records are stored, ensuring a traceable history for continuous improvement and compliance.

Through the App Security Management System, both primary users and developers receive vital, tailored software for device protection and app security. The platform’s streamlined dashboards, customizable scan options, in-depth reports, and centralized database empower users to maintain data integrity and developers to release trustworthy, compliant applications that prioritize user security and privacy.

**Report:**

1. Report Header

Title: "APK Analysis Report"

App Name and Version: Extracted from the APK metadata

User Details: Username, if applicable (for personalized databases)

Date of Analysis: Automatically generated date and time of the report

2. Summary

Classification Result: "Malicious" or "Benign"

// Overall Risk Level: "High", "Medium", or "Low" (based on classification score or probability)

// Quick Recommendations: Brief advice based on the risk level (e.g., "It is recommended to uninstall this app due to high-risk indicators")

3. Feature Analysis

Permissions Analysis:

List of permissions requested by the app, categorized into high-risk (e.g., camera, SMS, location) and standard permissions.

Highlight any potentially dangerous permissions.

API Calls:

Highlight API calls related to data access, messaging, or system modification.

Intents:

List of intents, especially those with security implications.

Categorize intents by purpose (e.g., communication, media access) and note any that may raise security concerns.

4. Detailed Analysis Results (Optional, for more technical users)

Classification Probability: Show the model's confidence level in classifying the app as malicious or benign.

Feature Contribution: Display the top features (permissions, API calls) that contributed most to the classification result, providing insights into why the app is flagged as risky.

5. Conclusion

Final Recommendation: Summarize whether to trust or avoid the app based on the analysis.

6. Appendix/Raw Data (Optional)

Detailed list of all features extracted from the APK file for further technical inspection.

This outline keeps the report user-friendly while offering technical insights, especially in sections like Feature Analysis and Detailed Analysis Results, if needed. You can format each section in HTML or PDF as per your requirement.

**QFD:**

QFD, or Quality Function Deployment, is a structured approach used in product development and project management. It helps ensure that customer needs and expectations are translated into specific product or service features. Essentially, it's a way to bridge the gap between what customers want and what a company delivers. QFD involves capturing customer requirements, prioritizing them, and then aligning internal processes to meet those requirements efficiently. It's like a roadmap for turning customer desires into tangible results.

**Normal:**

1. User will have a dashboard where they can see all the options of access
2. User Profile edit option
3. APK analysis
4. Apk scan history

**Expected:**

1. APK file can be uploaded
2. Malicious apps would get detected

**Excited:**

1. Multiple Scan Options: Full scan for scanning all apps in the device, quick scan for a single or fewer apps.
2. Custom App Management: Whitelist for trusted apps, blacklist for risky ones, and this would be done by the choice of the users.

DashBoard:

User login

Box

Whitelist blacklist

Previous Scans

Sign up - As General User or Corporate App developer → Verification through email , and activated through the sent mail

Login - email, pass

1. Profile (Button)
2. Name
3. Dashboard
   1. Profile
   2. Lists
   3. History
4. Logout

Scan Button

* Full Scan / Multiple Scan
* Single App Scan
  + Upload an apk
  + Connect your device (Use debugging mode)
  + On the device, go to Settings > About <device>.
  + Tap the Build number seven times to make Settings > Developer options available.
  + Then enable the USB Debugging option
  + Connect the mobile device to the laptop with a USB cable
  + On the mobile device, tap Always allow from this computer
  + General user: Only short report of benign/malicious
  + Developer: Detailed report with the result
  + APK icon… download apk
  + Enlist whitelist/blacklist

**Use Case Diagram:**

# **Use Case Diagram**

A Use Case describes the system behavior under various conditions as the

system responds to a request from one of its stakeholders. In fact, a use case diagram is a kind of visualization of the system where an end-user has an idea of a specific feature. It simply describes a story using corresponding actors who perform important roles in the story and make the story understandable for the users.

The first step in writing a Use Case is to define the set of “actors” that will be involved in the story. Actors are the different people or systems that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as system operators. They procedure some information or consume some information. Every user has one or more goals when using the system.

## **Primary Actor**

Primary actors interact directly to achieve the required system function and

derive the intended benefit from the system. They work directly with the

software. They produce some information and consume some information too.

## **Secondary Actor**

Secondary actors support the system so that primary actors can do their work. They either produce or consume information.

Here is the use case diagram to observe the non-technical view of the

System.

**Actors:**

· Mobile Device Users

· Mobile App Developers

· System

Primary Actor:

· Mobile Device Users

· App developers

· System

Secondary Actor:

Modules:

1.Authentication

2.Dashboard

3.Scan Module

3.Apk Management

4.Custom App Management

5.Report Generation

6.Data Storage and Security

· Usage history management

Level 0:

Name: ASMS

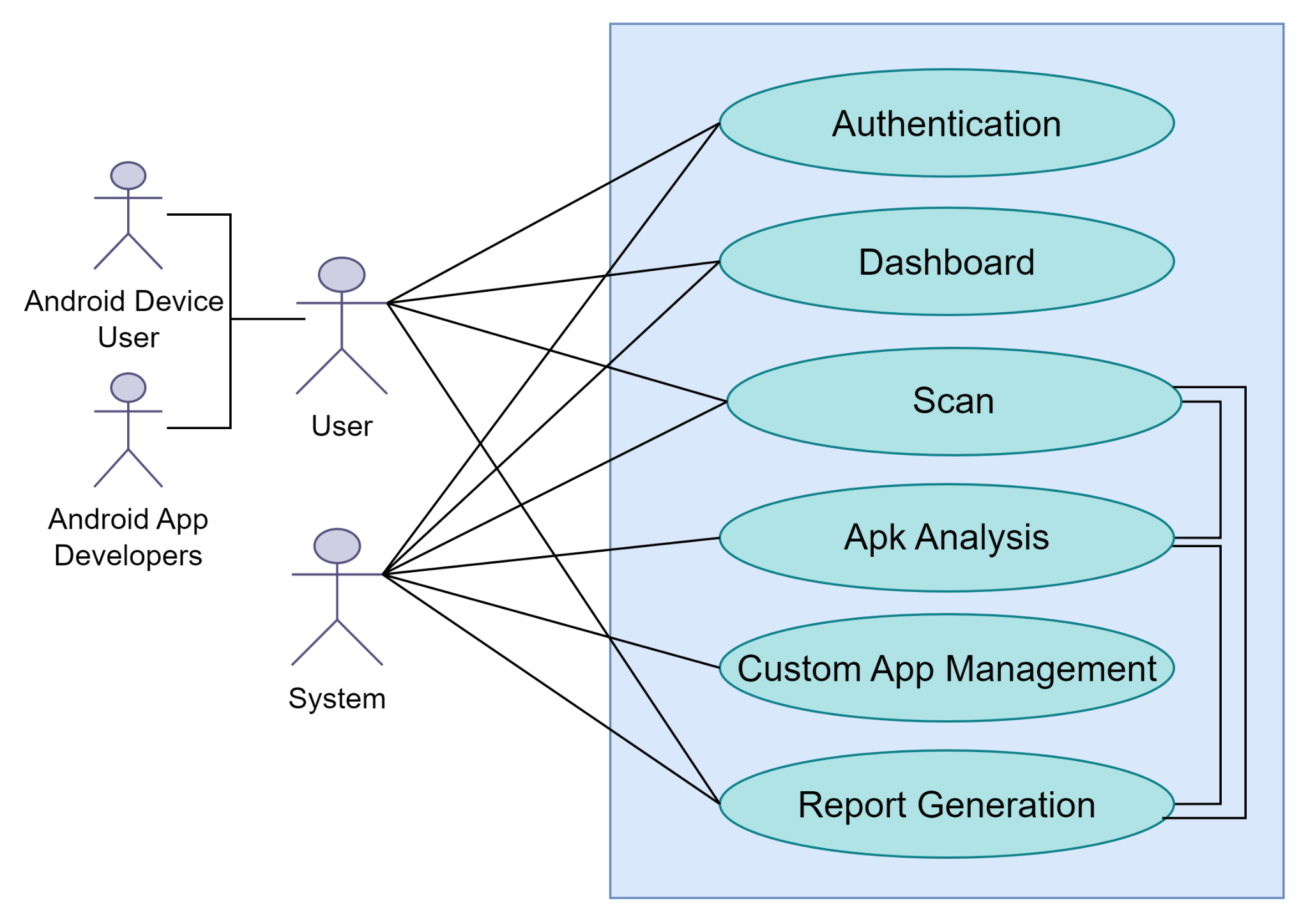
Primary Actor: Android device users, Android app developers, System



Level 1:

Name: ASMS(Detailed)

Primary Actor: Android device users, Android app developers, System



Level 1.1:

Name: Authentication

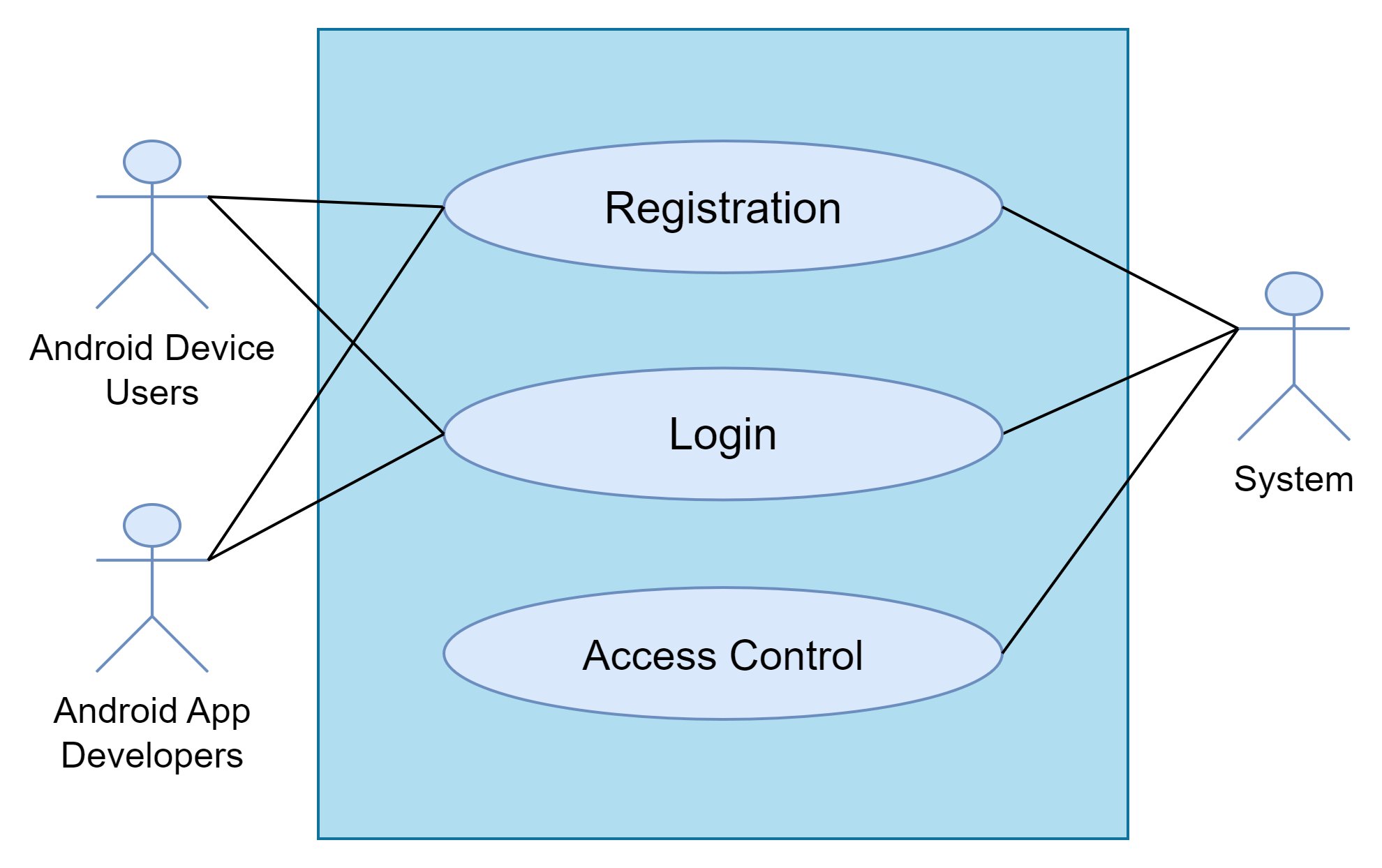
Primary Actor: Android device users, Android app developers

Secondary Actor: System

· Registration

· Login

· Access Control



Level 1.2:

Name: Dashboard

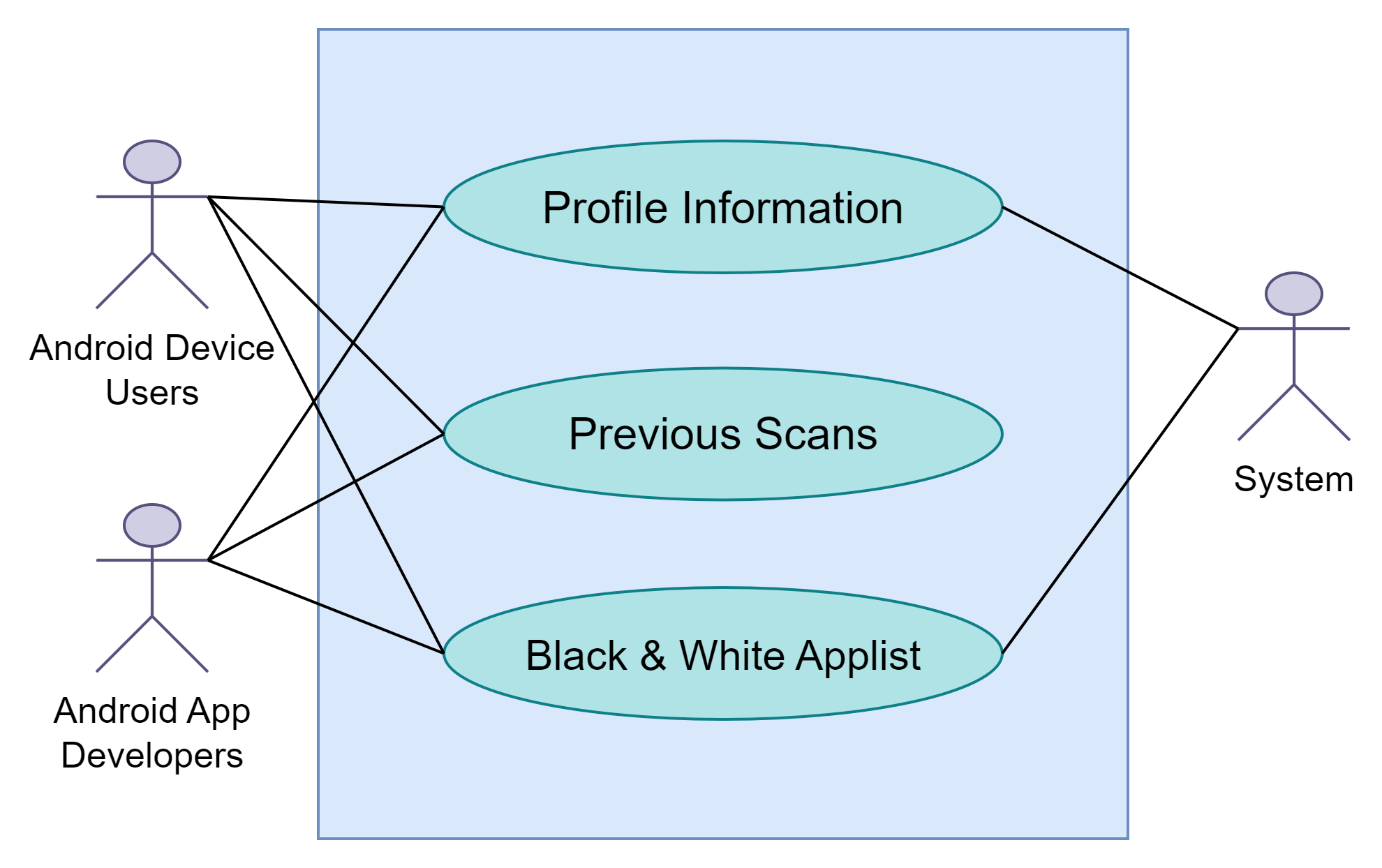
Primary Actor: Android device users, Android app developers

Secondary Actor: System

· Profile Information

· Previous Scans

· Black & White Applists



Level 1.3

Name: Scan

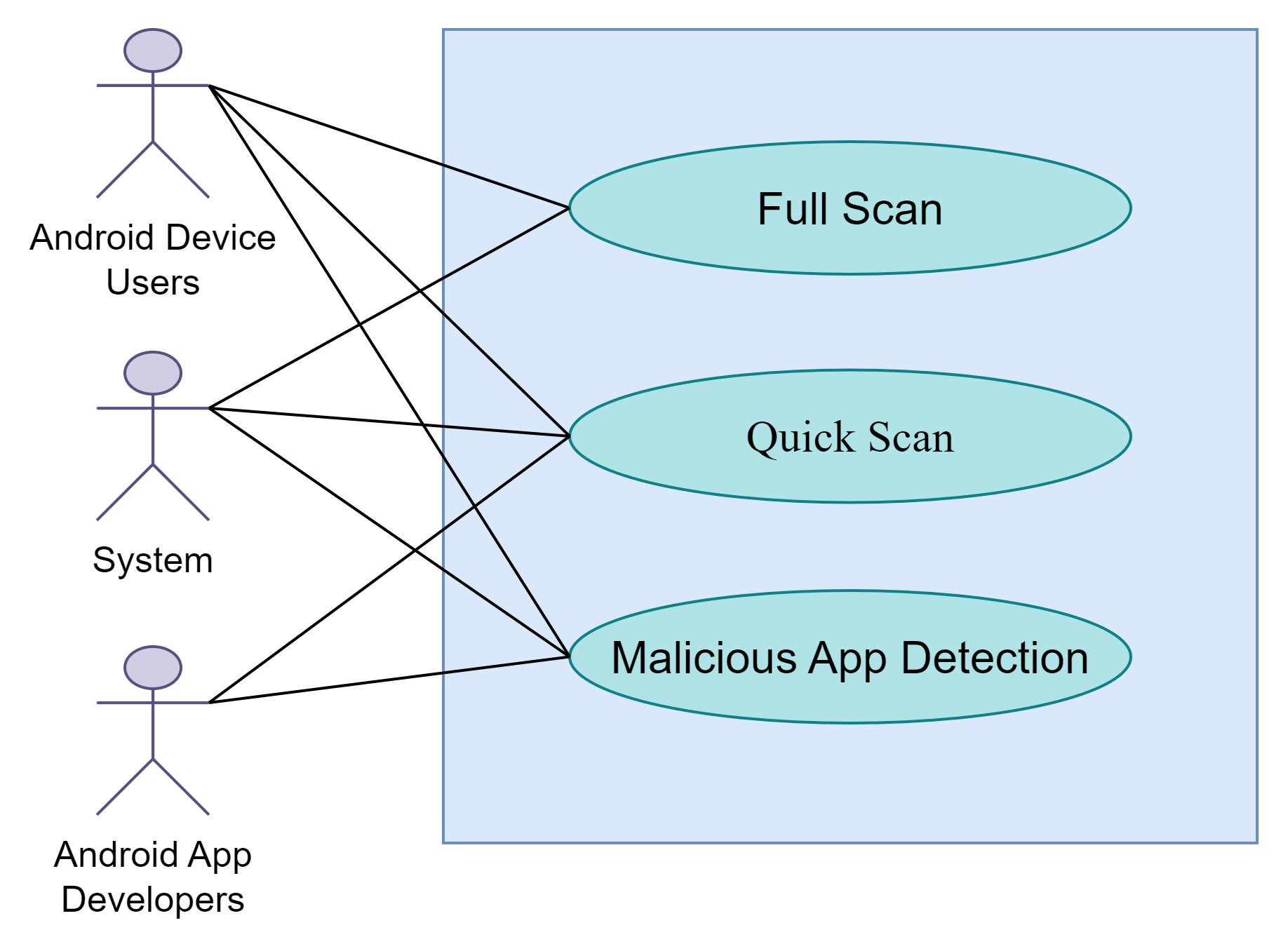
Primary Actor: System

Secondary Actor: Android device users, Android app developers

· Full Scan

· Quick Scan

· Malicious App Detection



Level 1.4:

Name: Apk Analysis

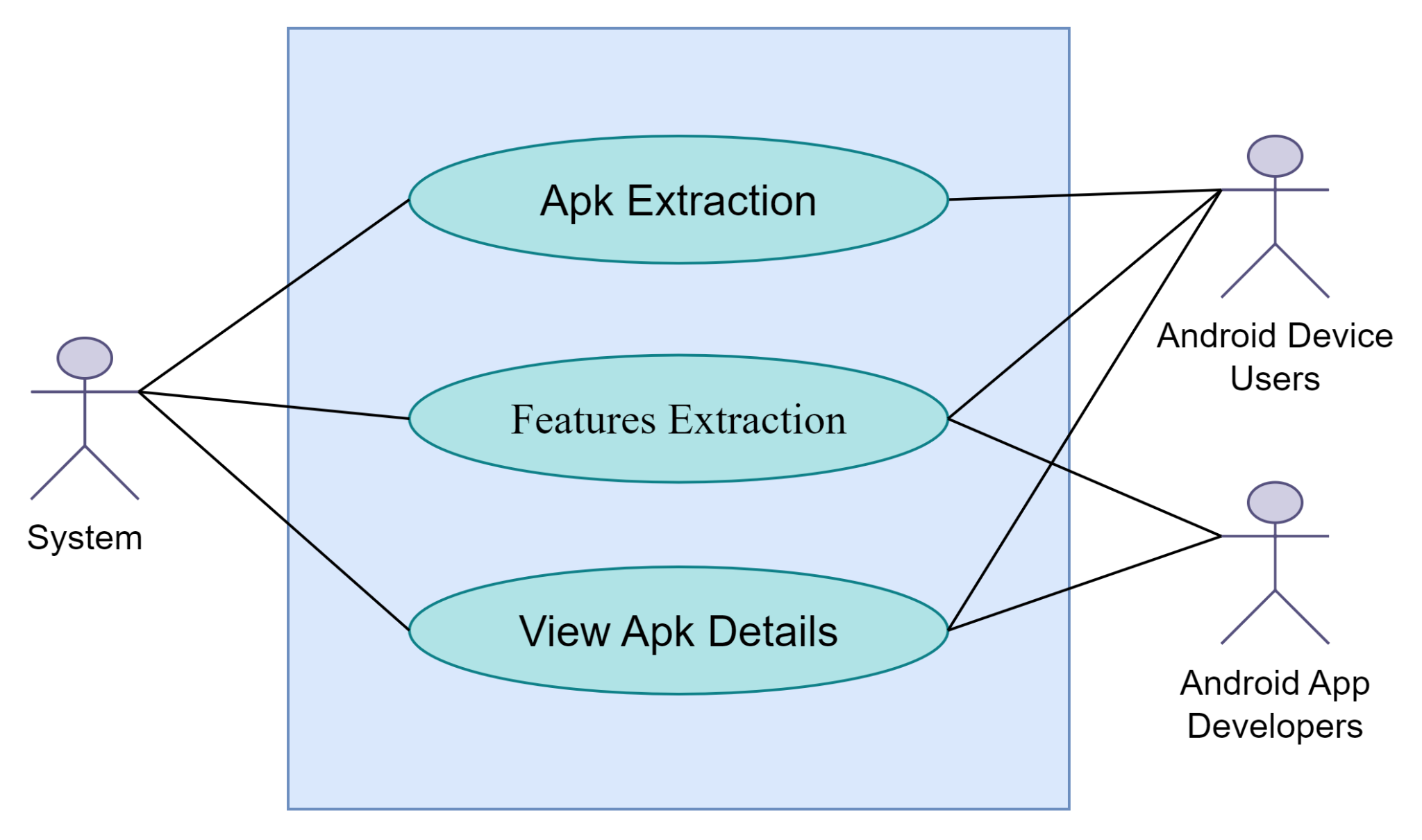
Primary Actor: System

Secondary Actor: Android device users, Android app developers

· Apk Extraction

· Features Extraction

· View Apk details



Level 1.5

Name: Custom App Management

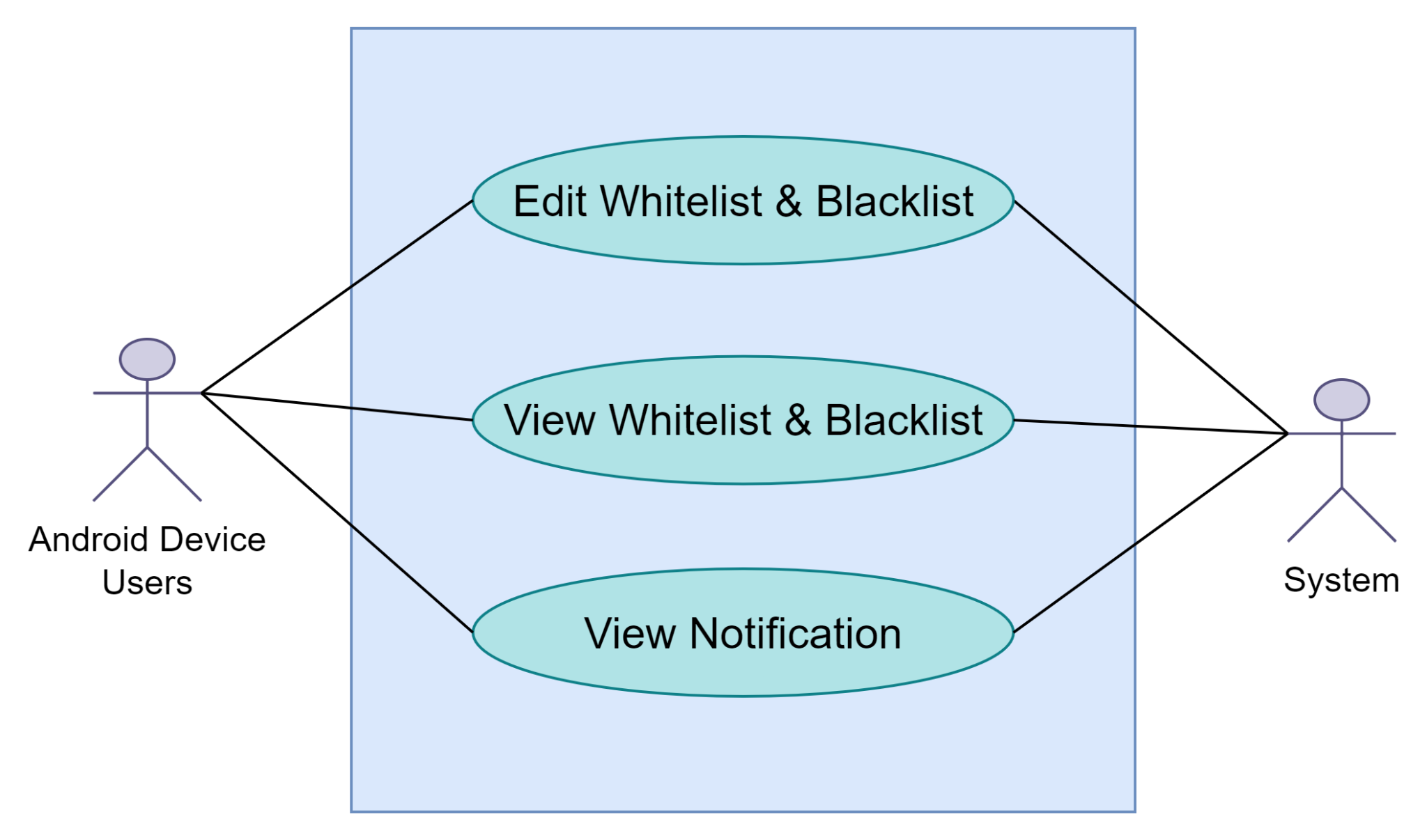
Primary Actor: Android device users

Secondary Actor: System

· Edit whitelist & blacklist

· Show whitelist & blacklist

· Show Notification(when detected any blacklisted app)



Level 1.6

Name: Report Generation

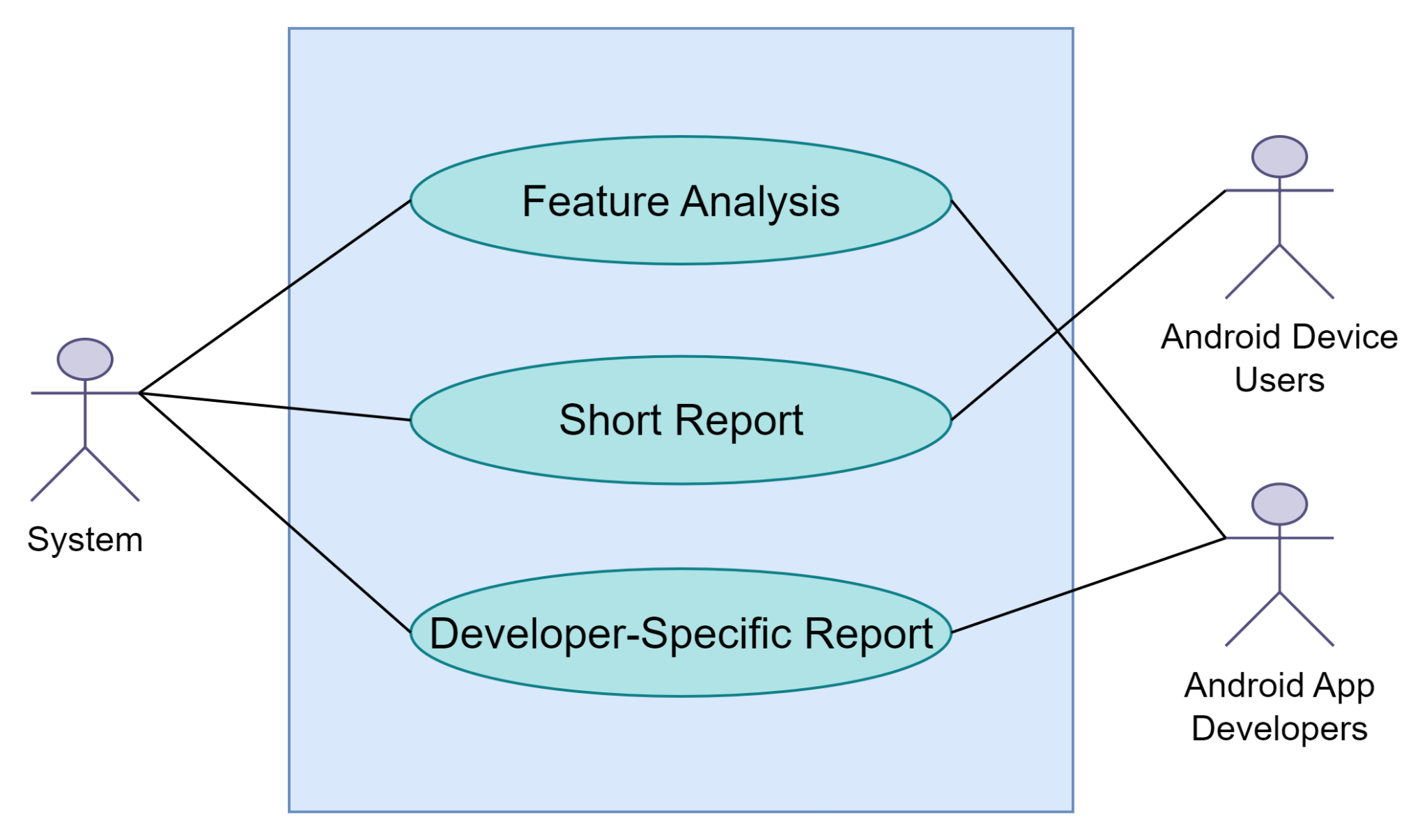
Primary Actor: System

Secondary Actor: Android device users, Android app developers

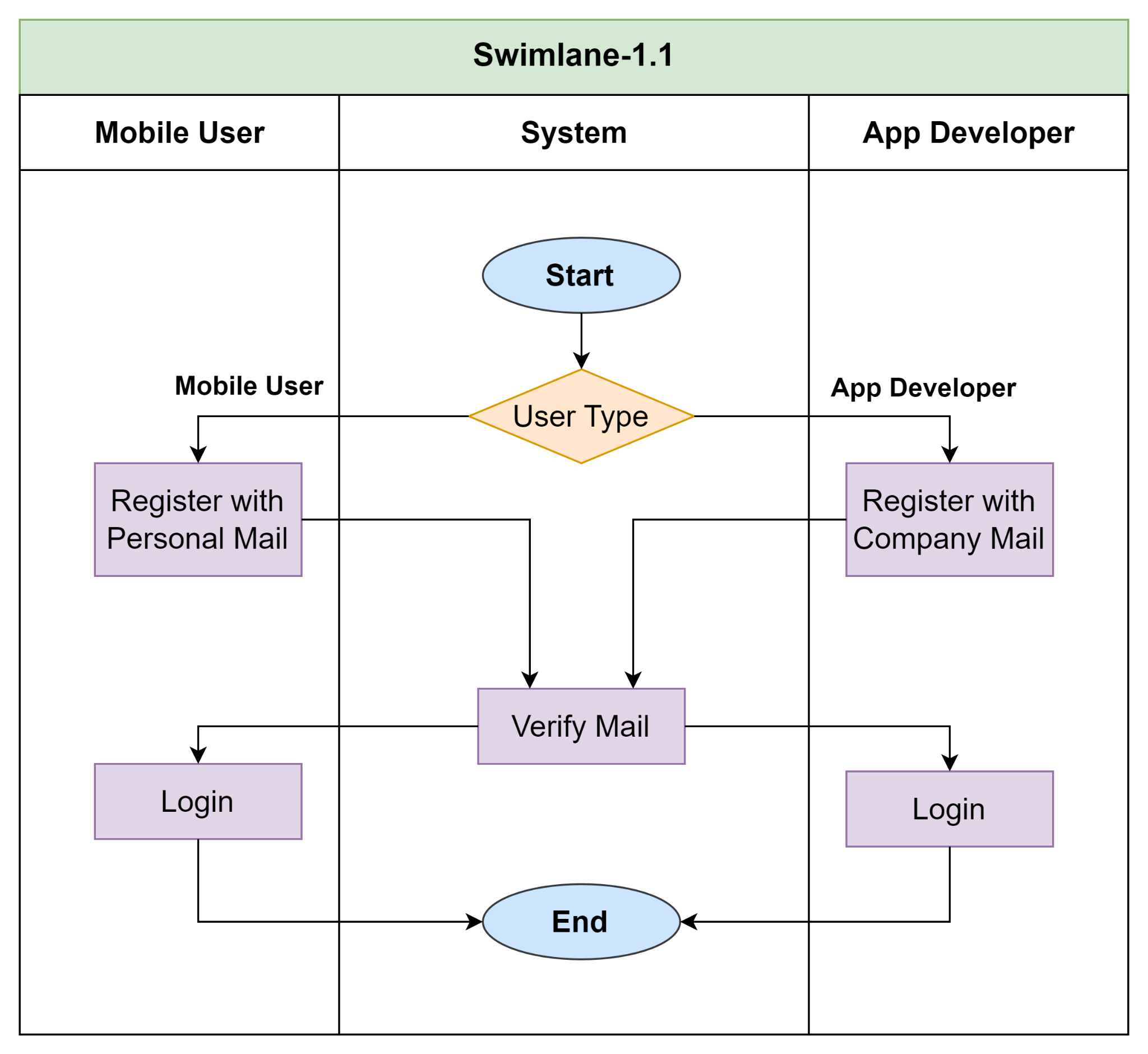
· Feature Analysis

· Short Report

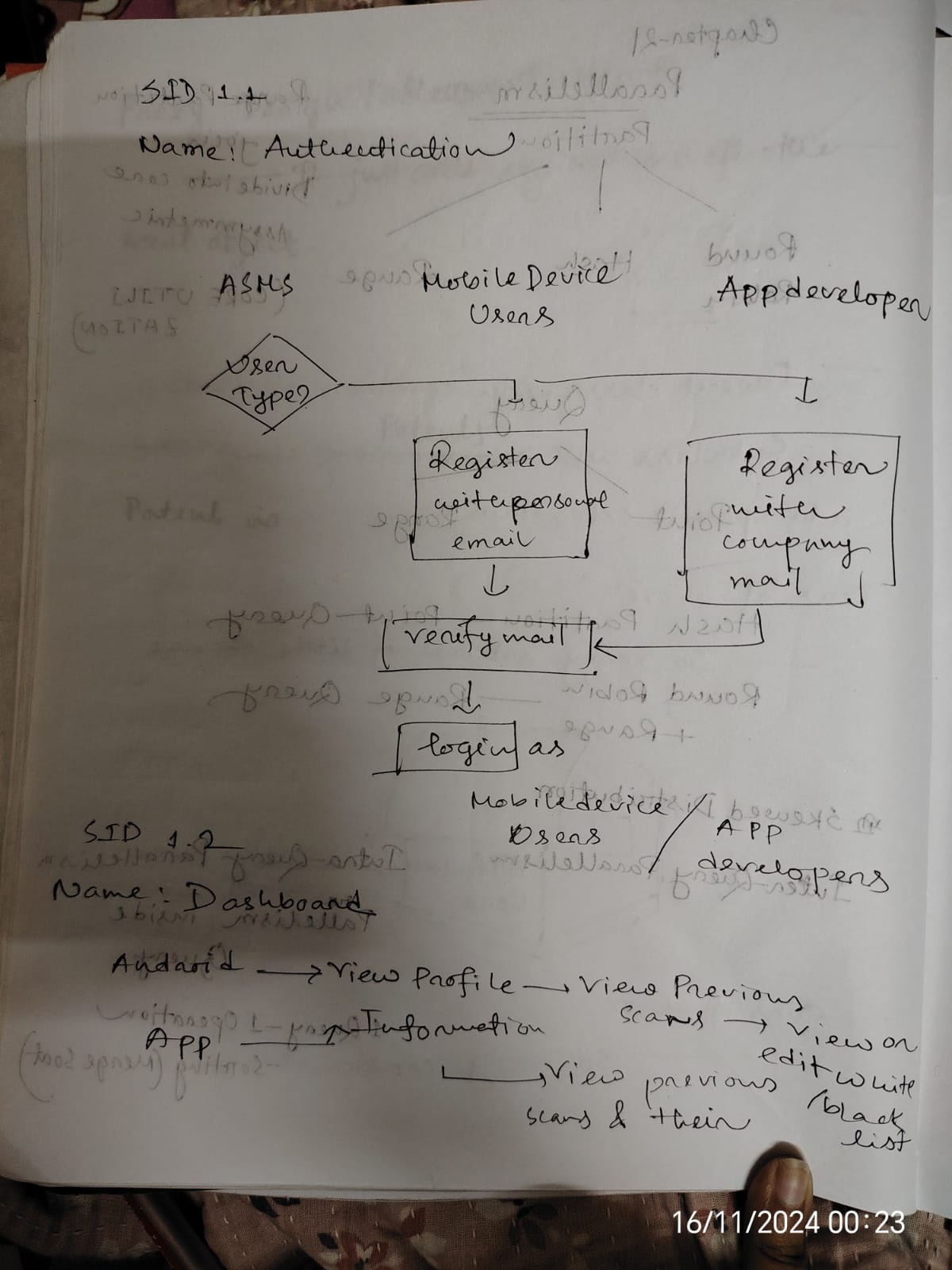
· Developer-Specific Report



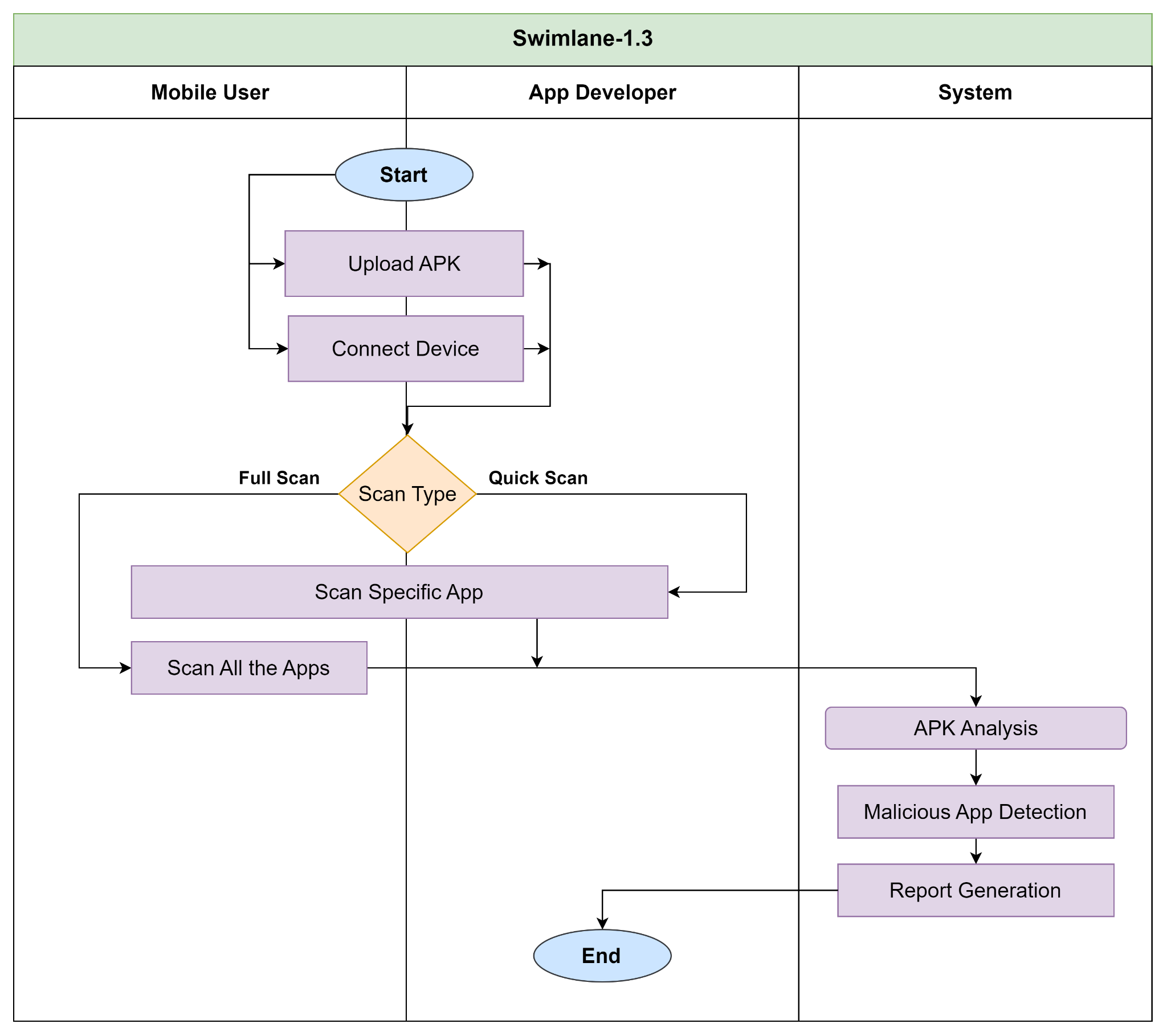
**Swimlane-1.1: Authentication**



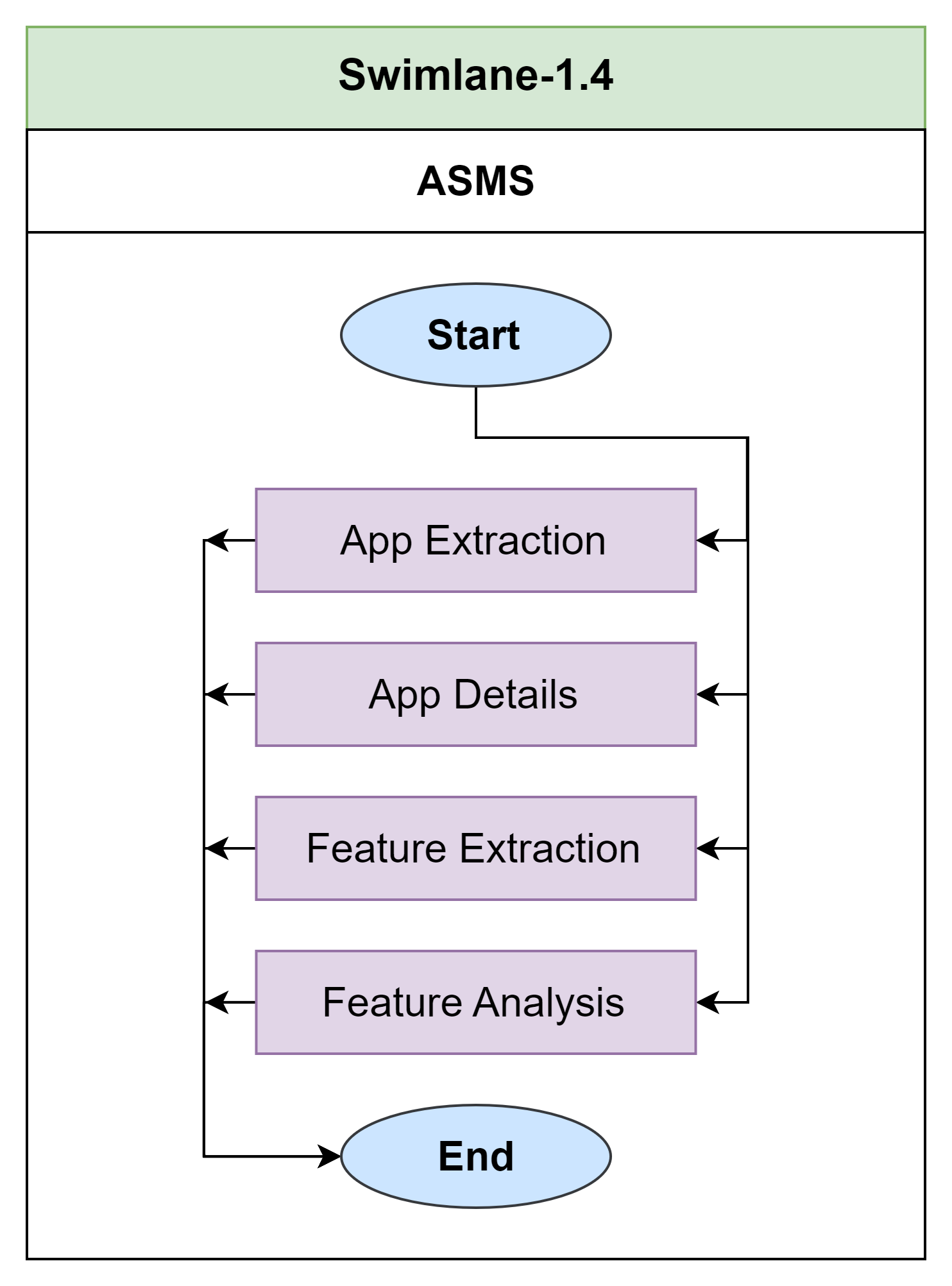
**Swimlane-1.2:Dashboard**

****

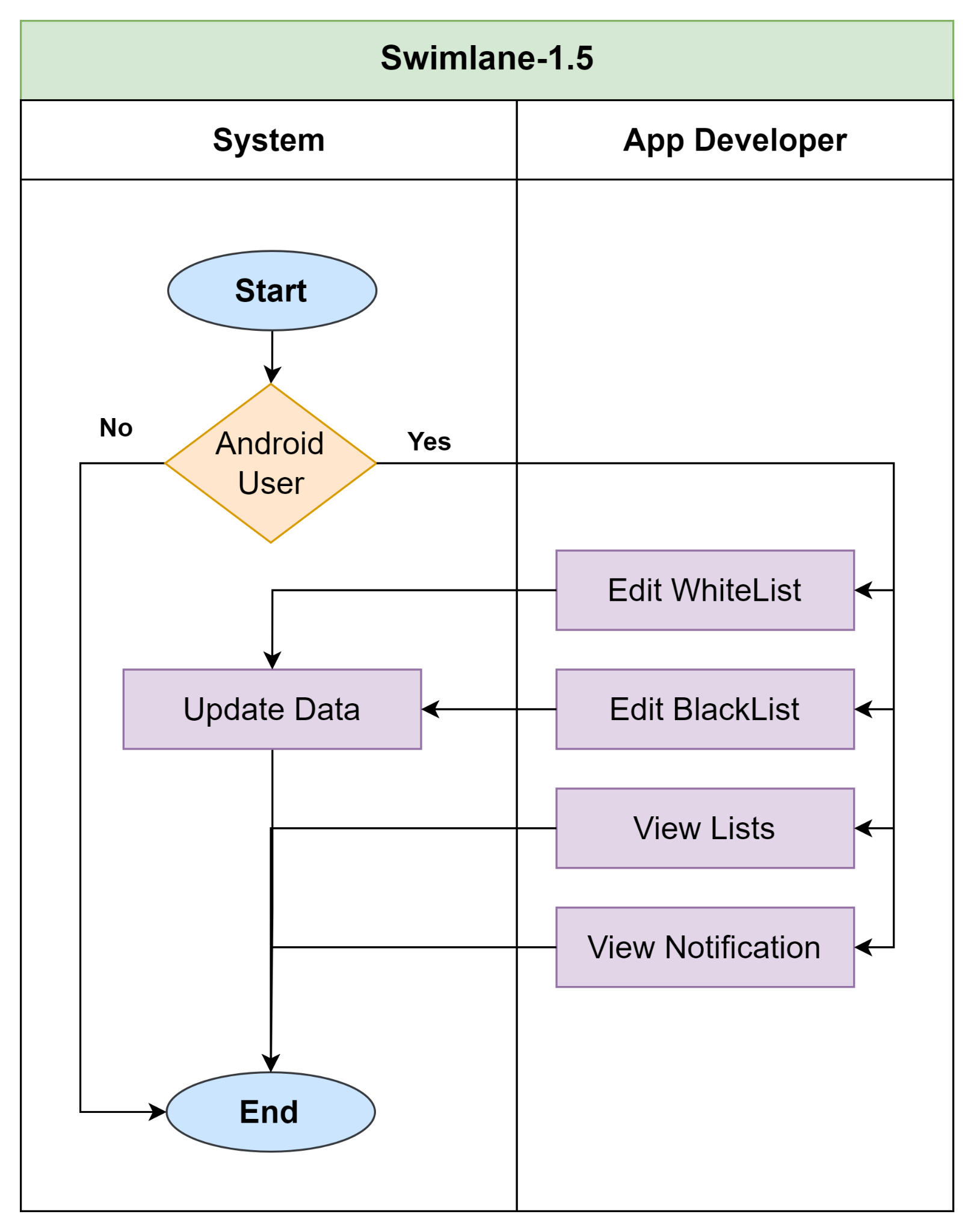
**Swimlane-1.3: Scan**

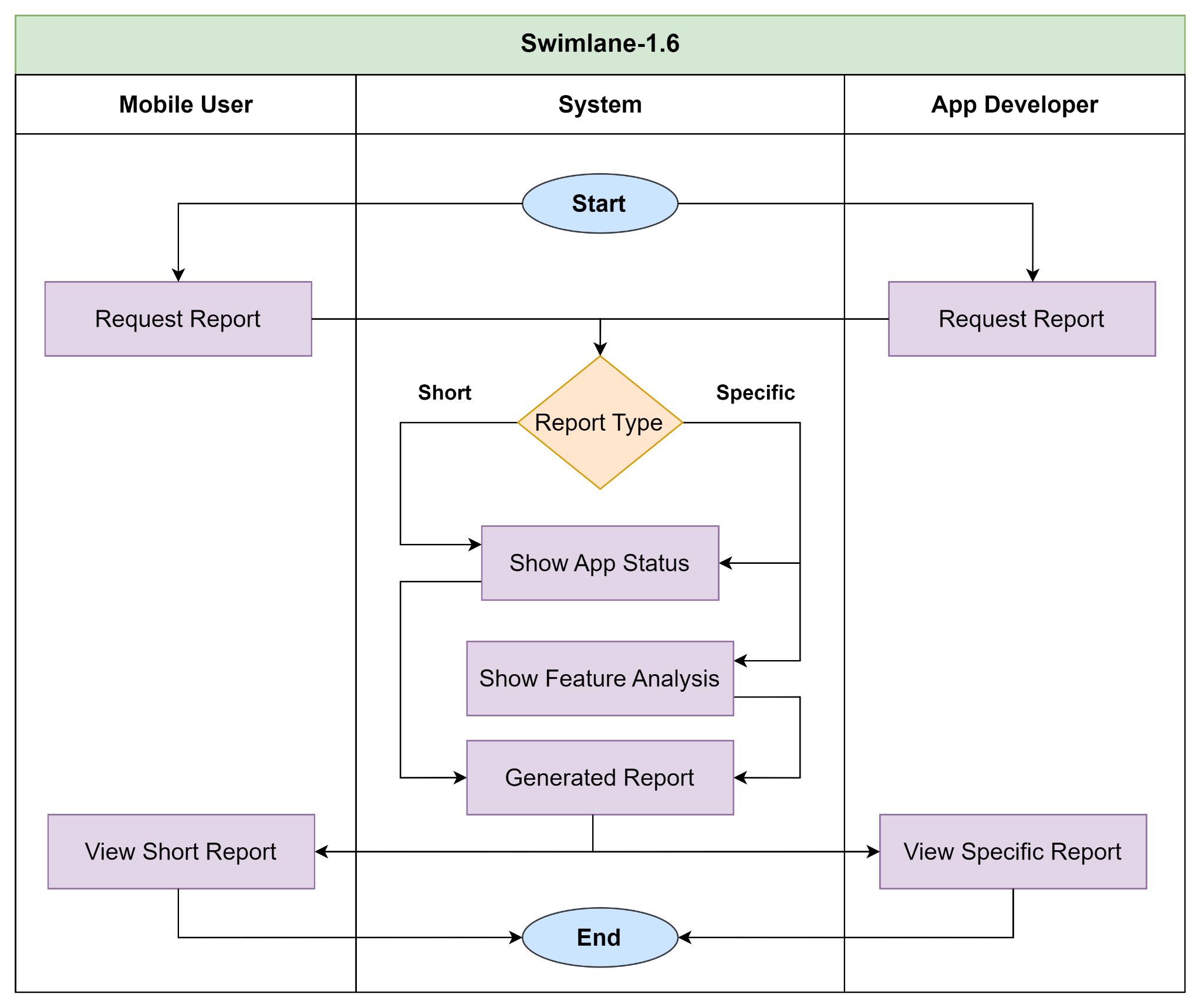
****

**Swimlane-1.4:**Apk Analysis

****

**Swimlane-1.5:** Custom App Management

****

**Swimlane-1.6:**

**Data Based Modeling:**

| **SL** | **Noun** | **Problem/Solution Space** | **Attributes** |
| --- | --- | --- | --- |
| 1 | Droid Scanner | S |  |
| 2 | App | S | 64, 9, 35, 51, 59, 62 |
| 3 | Security | S |  |
| 4 | Management System | S |  |
| 5 | Software | S |  |
| 6 | Solution | S |  |
| 7 | Mobile Device | S |  |
| 8 | Threat | P |  |
| 9 | Name | S |  |
| 10 | App Development | S |  |
| 11 | User | S | 64, 9, 28, 57, 27 |
| 12 | Groups | S |  |
| 13 | Primary Mobile Device Users | S |  |
| 14 | Device Security | S |  |
| 15 | Mobile App Developers | S |  |
| 16 | Tools | S |  |
| 17 | Vulnerabilities | P |  |
| 18 | Privacy | S |  |
| 19 | Options | S |  |
| 20 | Full Scan | S |  |
| 21 | Quick Scan | S |  |
| 22 | Command | S |  |
| 23 | Forms | S |  |
| 24 | Security Monitoring | S |  |
| 25 | Tool | S |  |
| 26 | Setup | S |  |
| 27 | User Dashboard | S |  |
| 28 | Profile | S |  |
| 29 | Scans History | S |  |
| 30 | Blacklist | S |  |
| 31 | Whitelist | S |  |
| 32 | Applist | S |  |
| 33 | Malicious App Detection | S |  |
| 34 | Feature | S |  |
| 35 | APK Files | S |  |
| 36 | APK Extraction | S |  |
| 37 | Custom App Management | S |  |
| 38 | Notification | S | 64, 11, 76, 74, 62 |
| 39 | Risky Apps | S |  |
| 40 | Scan | S | 64, 74, 2, 50, |
| 41 | Outcome | S |  |
| 42 | User Database | S |  |
| 43 | Reports | S | 64, 40, 58, 51, 69, 59, 60, 61 |
| 44 | Scan Summaries | S |  |
| 45 | Data Privacy | P |  |
| 46 | Compliance | P |  |
| 47 | Safety | P |  |
| 48 | Launch | S |  |
| 49 | Dashboard | S | 64, 11, 28, 29, 30, 31 |
| 50 | Scan Reports | S |  |
| 51 | Permissions | S |  |
| 52 | Location | S |  |
| 53 | Camera | S |  |
| 54 | Permission Viewer | S |  |
| 55 | Request | S |  |
| 56 | Privacy | P |  |
| 57 | Role | S |  |
| 58 | App Details | S |  |
| 59 | Intents | S |  |
| 60 | Feature Analysis | S |  |
| 61 | API Calls | S |  |
| 62 | Status | S |  |
| 63 | Industry Standards | S |  |
| 64 | ID | S |  |
| 65 | Security Records | P |  |
| 66 | App History | P |  |
| 67 | Improvement | P |  |
| 68 | Platform | S |  |
| 69 | Security Status | S |  |
| 70 | Database | S |  |
| 71 | Integrity | S |  |
| 72 | Applications | S |  |
| 73 | Protection | P |  |
| 74 | Date | S |  |
| 75 | Result | S |  |
| 76 | Message | S |  |

**Final Data Objects:**

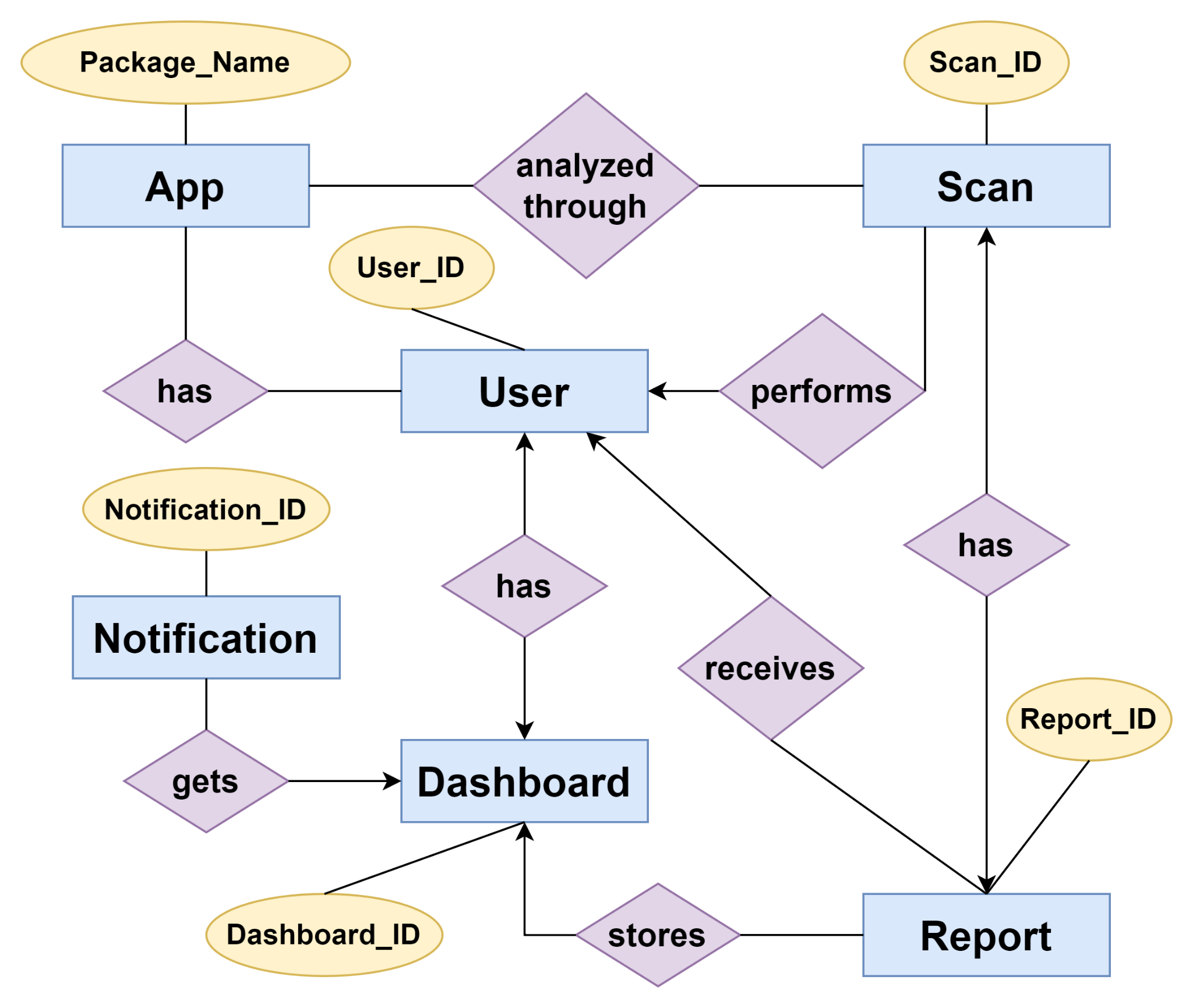
### **Data Objects**

1. **User**
   * Attributes: User\_ID, Name, Role (Primary User/Developer), Profile
2. **Dashboard**
   * Attributes: Dashboard\_ID, User\_ID (FK), Blacklist, Whitelist, Scan History, Report\_ID(FK)
3. **App**
   * Attributes: Package\_Name, Name, APK\_File, Permissions, Intent, Status (Malicious/Benign), API\_Calls
4. **Scan**
   * Attributes: Scan\_ID, Type (Full/Quick), Date, App\_ID (FK), User\_ID (FK), Scan\_Results
5. **Report**
   * Attributes: Report\_ID, Scan\_ID (FK), App Details, Permissions Used, Intent Analysis, Security Status
6. **Notification**
   * Attributes: Notification\_ID, User\_ID (FK), Message, Date, Status (Read/Unread)

**Relationship between Objects:**

| **Data Object** | **Relationship** | **Related Data Object** |
| --- | --- | --- |
| Scan | has | Report |
| User | performs | Scan |
| Dashboard | gets | Notification |
| User | receives | Report |
| User | has | App |
| App | analyzed through | Scan |
| User | has | Dashboard |
| Dashboard | stores | Report |

**Entity Relation ( ER ) Diagram:**

**+**

**Schema Diagram:**

| **Data Object** | **Attribute** | **Type** | **Size** |
| --- | --- | --- | --- |
| User | User\_ID  Name  Role  Profile | Varchar  Varchar  Varchar  Varchar | 100  100  50  100 |
| Dashboard | Dashboard\_ID User\_ID (FK) Report\_ID(FK)  Blacklist  Whitelist  Scan History | Varchar  Varchar  Varchar  Text  Text  Text | 100  100  100 |
| App | Package\_Name  Name  Status  APK\_File Permissions  Intent  API\_Calls | Varchar  Varchar  Varchar  Blob  Text  Text  Text | 100  100  50 |
| Scan | Scan\_ID  Type  App\_ID (FK) User\_ID (FK) Scan\_Results  Date | Varchar  Varchar  Varchar  Varchar  Text  DateTime | 100  50  100  100 |
| Report | Report\_ID  Scan\_ID (FK)  Security Status  App Details Permissions Used Intent Analysis | Varchar Varchar  Varchar  Varchar  Varchar  Varchar | 100  100  50  100  100  100 |
| Notification | Notification\_ID  User\_ID (FK) Message  Date  Status | Varchar Varchar  Varchar  DateTime  Varchar | 100  100  100  100 |

**-**

**Class Based Diagram:**

Noun Identification:

General Classification:

1. External Entities

2. Things (information about the system)

3. Occurrences or Events

4. Roles (people who interact with the system)

5. Organizational units(Groups or Events)

6. Places

7. Structures(Related class of object)

| **SL** | **Noun** | **General Classification** |
| --- | --- | --- |
| 1 | Droid Scanner | 1, 2 |
| 2 | App | 2, 7 |
| 3 | Security | 2, 3 |
| 4 | Management System | 2, 7 |
| 5 | Software | 2, 7 |
| 6 | Solution | 2, 7 |
| 7 | Mobile Device | 2, 6 |
| 8 | Threat | 2, 3 |
| 9 | Standards | 2, 7 |
| 10 | App Development | 2, 3 |
| 11 | Stakeholder | 1, 4 |
| 12 | Groups | 5 |
| 13 | Android Device User | 1, 4, 5 |
| 14 | Device Security | 2 |
| 15 | Android App Developers | 1, 4, 5 |
| 16 | Tools | 2, 7 |
| 17 | Vulnerabilities | 2, 3 |
| 18 | Privacy | 2 |
| 19 | Options | 2 |
| 20 | Full Scan | 2, 3 |
| 21 | Quick Scan | 2, 3 |
| 22 | Command | 2, 3 |
| 23 | Forms | 2, 7 |
| 24 | Security Monitoring | 2, 3 |
| 25 | Tool | 2, 7 |
| 26 | Setup | 2, 7 |
| 27 | User | 1, 2, 3, 7 |
| 28 | Profile | 2 |
| 29 | Scans History | 2 |
| 30 | Blacklist | 2 |
| 31 | Whitelist | 2 |
| 32 | Applist | 2, 3 |
| 33 | Malicious App Detection | 2 |
| 34 | Feature | 2, 3 |
| 35 | APK Files | 2 |
| 36 | APK Extraction | 2, 7 |
| 37 | Custom App Management | 2, 7 |
| 38 | Checks | 2, 3 |
| 39 | Risky Apps | 2 |
| 40 | Scan | 2, 3, 7 |
| 41 | Outcome | 2, 7 |
| 42 | User Database | 2, 7 |
| 43 | Reports | 2, 3, 7 |
| 44 | Scan Summaries | 2 |
| 45 | Data Privacy | 2 |
| 46 | Compliance | 2 |
| 47 | Safety | 2, 3 |
| 48 | Launch | 2 |
| 49 | Dashboard | 1, 2, 7 |
| 50 | Scan Reports | 2, 7 |
| 51 | Permissions | 6, 7 |
| 52 | Location | 6, 7 |
| 53 | Camera | 2, 7 |
| 54 | Permission Viewer | 2 |
| 55 | Request | 2 |
| 56 | Privacy | 2 |
| 57 | Report | 2, 7 |
| 58 | App Details | 2, 3 |
| 59 | Intents | 2 |
| 60 | Feature Analysis | 2 |
| 61 | API Calls | 2 |
| 62 | Status | 2 |
| 63 | Industry Standards | 2 |
| 64 | Issue | 2 |
| 65 | Security Records | 2 |
| 66 | Traceable History | 2, 7 |
| 67 | Improvement | 2, 7 |
| 68 | Platform | 2, 3 |
| 69 | Notification | 2, 3, 7 |
| 70 | Database | 2, 3 |
| 71 | Integrity | 2, 3 |
| 72 | Applications | 3 |
| 73 | Protection | 6, 7 |

**Class Based Diagram:**

Noun Identification:

General Classification:

1. External Entities

2. Things (information about the system)

3. Occurrences or Events

4. Roles (people who interact with the system)

5. Organizational units(Groups or Events)

6. Places

7. Structures(Related class of object)

**Analysis:**

For all lists of Noun general classification is performed.If a noun is an essential entity or fill-up 3 or more classification criteria then it is considered as a potential class.Here potential classes are:

1. User
2. Android Device User
3. Android App Developer
4. Scan
5. Report
6. Notification
7. Dashboard