applied sciences

MDPI

Article

MZAP—Mobile Application for Basketball Match Tracking and Digitalization of Endgame Reports

Predrag Pecev [1,*] and Branko Markoski [2]

[1] College of Applied Studies-Sirmium, Zmaj Jovina 29, 22000 Sremska Mitrovica, Serbia
[2] Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia; markoni@uns.ac.rs
* Correspondence: pecev@vs-sirmium.sm.ac.rs

Abstract

This paper presents MZAP, a mobile application designed to digitalize basketball match tracking and generate secure, searchable endgame reports. Used by the Basketball League of Serbia, MZAP creates tamper-proof digitally signed records stored as password-protected PDFs with unique UUIDs, digital signatures, and QR codes. Each report is accompanied by a JSON file containing match data, enabling efficient validation through hashed checksums and facilitating data extraction and searchability. The system supports both online and offline modes, bilingual interfaces, mobile and tablet use, and includes features such as WiFi-based monitoring, physical printing, and various sharing options. The solution aims to reduce officials' working time and increase data accuracy by minimizing human error through structural and UI-level validation methods and real-time monitoring by multiple observers during games. As part of the MZAP software suite, MZAP Converter is under development to support the digitization of legacy paper-based reports using custom CRNN neural networks to optically recognize and digitize historical paper-based reports, bringing them to the same standard as newly created digital ones. The paper also reflects on the broader impact of digital transformation within the Basketball League of Serbia.

Keywords: basketball; endgame; report; digitization; digitalization; match tracking; android application

check for updates

Academic Editor: Christos Bouras

Received: 10 May 2025
Revised: 14 June 2025
Accepted: 16 June 2025
Published: 30 June 2025

Citation: Pecev, P.; Markoski, B. MZAP—Mobile Application for Basketball Match Tracking and Digitalization of Endgame Reports. *Appl. Sci.* **2025**, *15*, 7339. https://doi.org/10.3390/app15137339

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Digitization converts physical information into a digital form, while digitalization uses digital tools to improve business processes. The MZAP application exemplifies digitalization by streamlining endgame reporting in the Basketball League of Serbia. In contrast, the MZAP Converter, still in beta, represents digitization by using neural networks to digitize old paper reports. Digital transformation is the next stage, where widespread digital adoption reshapes entire businesses or industries. When most workflows are digitized and digitalized, an organization is considered digitally transformed.

Document digitization is a global trend across many fields, driven by the vulnerability of paper documents to damage and forgery. A major issue with non-digitized documents is the difficulty of searching and extracting data, often requiring manual effort. Once digitized, it is essential that documents are searchable within a database and allow for efficient data extraction and use.

The purpose of the research presented in this paper stems from the problems that the Basketball League of Serbia have been encountering over the past decade, which are

endgame record storing and on-demand data extraction. To address these issues, a solution that handles the archiving of past and future endgame reports must be created.

MZAP (an abbreviation that stands for "Mobilni Zapisnik" in Serbian, meaning "Mobile Match Report" in English) is an Android application that solves the problem of archiving future endgame reports. It is intended to handle endgame reports for the Basketball League of Serbia. However, with slight modifications, it can manage endgame reports for other leagues as well. The core of the MZAP application is essentially a basketball game tracking system, similar to the one used by FIBA [1], with the addition of generating digitally signed, password-protected endgame reports that include a QR code. This application defines structures and methods for handling both future and past endgame reports.

To address the large archive of paper-based endgame reports in the Basketball League of Serbia, the MZAP Converter application is being developed as part of the broader MZAP software suite. Currently in beta and operating through a command-line interface, MZAP Converter leverages a Convolutional Recurrent Neural Network (CRNN) trained on a custom dataset composed of scanned historical reports. The OCR model is tailored to the layout and structure of league-specific forms, and early testing has shown promising performance, though formal evaluation and optimization are ongoing. While not flawless, this approach is significantly faster and more efficient than the manual transcription of paper-based match endgame reports into digital form. Unlike off-the-shelf OCR solutions, a custom model was developed to ensure independence from third-party software. The design, implementation, and performance evaluation of the MZAP Converter and its CRNN neural network will be the subject of a follow-up paper, to be published once the software exits beta and is deemed stable for systematic testing and benchmarking. The application outputs a JSON file containing structured match data for indexing, search, and data retrieval, and a generated PDF endgame match report file based on the mentioned match data. Both outputs are later stored on the MZAP server. This will enable comprehensive digitization and archiving of both historical and future match reports.

Since the application runs on mobile phones and tablets, it addresses several technical challenges commonly faced by basketball officials, such as the unavailability of laptops, complex setup procedures at game venues, and limited internet connectivity. By overcoming these issues and reducing working time and error rates, the application significantly enhances the user experience for basketball officials.

The research presented in this paper has several goals:

- Create an efficient solution with distinctive features for the digitization and digitalization of endgame match reports for the Basketball League of Serbia, which relies as little as possible on other solutions, packages, and frameworks.
- To deliver a more efficient and error-resistant workflow, along with a new or enhanced user experience for basketball officials.
- Reduce an official's working time by 30% by providing a workflow that is fast, intuitive, and resistant to errors, thereby decreasing the possibility of errors.
- Prove that digital match reporting improves acquired data accuracy. Data accuracy is increased through application data validation methods during data entry on both the structural and UI levels, and visual control methods where digital reports are exposed to various officials who can observe and pinpoint an error during data entry, which occurs during a basketball game.
- Digitalization brings various benefits to the Basketball League of Serbia that will emerge during and after the first season of using the software.

This research addresses the need for a customized mobile solution that enables the digitalization of basketball endgame match records and generates searchable, digitally signed, and password-protected documents, thus providing a high level of security. The

proposed system also offers the potential for template-based report configuration on mobile devices, supporting adaptability across different leagues and possibly other sports. In doing so, it addresses a notable gap in the existing literature. Furthermore, this study aims to demonstrate that such solutions can enhance user experience, reduce working time, and minimize the likelihood of errors, thus increasing data accuracy. To the best of our knowledge, there are currently no domain-specific solutions that combine real-time digitalization of structured game data (via the MZAP Android application) with the digitization of historical paper-based reports (via the in-development MZAP Converter) in a unified and workflow-integrated manner, as proposed in our approach.

2. Related Work

Numerous high-quality applications offer basketball match tracking, statistics, and analysis, including FIBA LIVESTATS [2], Statastic Basketball Tracker [3], Basketball Stats Assistant [4], Easy Stats for Basketball [5], Breakthrough Stats [6], iScore Basketball Scorekeeper [7] and Basketball Digital Scoresheet [8]. Among these, Basketball Digital Scoresheet [8] most closely aligns with MZAP, as both focus on generating digital endgame reports, though they differ in features and system integration. Most of these solutions are mobile apps for Android or iOS, emphasizing portability and ease of use. FIBA Digital Scoresheet [1] is an official FIBA Windows desktop application designed to replace paper scoresheets, offering functionality comparable to both MZAP and Basketball Digital Scoresheet [8]. However, unlike the mobile-focused MZAP and Basketball Digital Scoresheet [8], its desktop-based implementation limits portability, making it less suited for on-the-go match tracking and reporting.

Compared to MZAP, the listed applications that are more directed towards statistics and analysis offer several features and detailed reports that MZAP does not. However, MZAP is not intended to be a mobile application for tracking basketball matches and providing statistics, but rather an application designed to generate digitally signed endgame reports for the Basketball League of Serbia.

Riza et al. [9], in their study, compared paper-based scoring using traditional scoresheets with application-based scoring, and concluded that application-based scorers (i.e., match tracking applications) can effectively replace manual, paper-based methods.

Given that the Basketball League of Serbia's endgame reports include a section for tracking score changes by period, MZAP was developed as a straightforward basketball score-tracking application. While it can be extended with statistical features and integrated with other systems, its primary purpose remains the generation of digitally signed endgame reports.

With the addition of the previously mentioned MZAP Converter software, the MZAP software package can be considered a unique solution due to its synergy. If further upgraded, the digitalized data acquired using the MZAP application could be used for statistical analysis and scouting using artificial intelligence, as shown in the work by Ivanković et al. [10]. Chomątek and Sierakowska [11] presented a solution that utilizes neural networks to extract data from paper-based match endgame reports (scoresheets), which is conceptually similar to the MZAP Converter.

The works of Pitić et al. [12] and Mijajlović et al. [13,14] explore various dimensions of digitalization in Serbia, highlighting both its advantages and associated challenges. Mijajlović et al. [13] examine several national digitization initiatives and their current progress, while [14] presents a project focused on digitizing scientific journals—a technically comparable effort to the MZAP Converter, which aims to digitize historical endgame reports.

Zaušková et al. [15] present the current state and predict the future of digitization as part of Industry 4.0, offering insights that are especially relevant considering sports like

basketball and football (soccer) are industries in their own right. Xiao et al. [16] discuss digitalization in sports broadly, while Santomier et al. [17] focus on digitalization within the NBA. The studies presented in these two papers share some similarities with the work discussed in this paper.

Poulsen [18] argues that digital transformation offers major opportunities for sports organizations, especially through new digital business models and increased value creation, with growing tech investments expected to enhance the fan and consumer experience. Mamedov [19] highlights digital transformation as a key driver of business improvement, focusing on its impact in sports like basketball and football. Their work emphasizes the role of digital tools in expanding opportunities, especially during challenges like the COVID-19 crisis. The piece also explores how digitalization enhances fan engagement, resource access, and operational efficiency across all levels of the sports industry.

In the context of Android application development, the MVP (Model-View-Presenter) architectural pattern was selected for the development of the MZAP application. This decision was based on the comparative analysis conducted by Vijaywargi and Boddapati [20], who examined MVP, MVVM (Model-View-View-Model), and MVI (Model-View-Intent) patterns, as well as the analysis of MVVM and MVP architectures by Zakaria and Nuryana [21]. The MVP pattern was deemed most suitable for the requirements of the Basketball League of Serbia.

The work of Li et al. [22] provides an overview of Android fragments, their life cycles, and the data flow between them, while Mohamed A. El-Zawawy [23] concentrates on the first two aspects. These studies offered valuable insights into optimizing the use of fragment architecture in Android development.

The foundations of the Agile software development methodology are detailed in the work of Waja et al. [24], while its impact on project outcomes is concisely explained in the paper by Dipendra Ghimire and Stuart Charters [25].

Book chapters by Weyde Lin [26] and by Shubhani Aggarwal and Neeraj Kumar [27] explore the structure and implementation of digital signatures. Additionally, the practical applications of digital signatures are discussed by Antonio Maña and Sonia Matamoros [28], as well as by Afrianto et al. [29]. Linfeng Jiang [30] examines various PDF protection mechanisms, whereas Singh et al. [31] focus specifically on password-protected documents. According to Sarkar et al. [32], the AES-256 encryption standard is most commonly employed for securing PDF documents with passwords.

Cryptographically secure pseudo-random number generators, which are used in password generation, are detailed in the work of Elena Almaraz Luengo and Javier Román Villaizán [33]. Among the leading algorithms for password hashing is Argon2, as presented by Biryukov et al. [34], while BLAKE3 has been shown to perform exceptionally well in generating cryptographic file hashes, as suggested by Mane et al. [35].

This work resembles a synthesis of Basketball Digital Scoresheet software [8] with the data handling approach proposed by Chomątek and Sierakowska [11]. In response to the requirements of the Basketball League of Serbia, the development of MZAP commenced in June 2022.

3. MZAP Android Application—Architecture and System Design

MZAP is a modular Android application that was developed using a fragment-based [22,23] MVP (Model-View-Presenter) architecture [20,21] in the Java programming language, within the Android Studio Giraffe integrated development environment (IDE). The Agile software development model [24,25] was selected for the development of the MZAP application due to its iterative approach, which, in addition to planned development, allows for continuous testing and feedback from basketball officials.

The application targets devices running Android 15 (SDK 34), with a minimum compatibility requirement of Android 10 (SDK 29). In addition to standard Android libraries, several third-party libraries were integrated to support the required functionalities.

To facilitate the storage of match-related data, a set of class entities was implemented to encapsulate all essential elements of the match report. These data are serialized in JSON format and stored as a string within the application's SharedPreferences. They can be exported to external storage (e.g., an SD card) or shared via other applications that support data transfer. To abstract and manage interactions with SharedPreferences, a custom utility class named MatchRecordManager was developed. For JSON serialization and deserialization, the Gson library [36] was used.

Java UUID Generator (JUG) library [37] was utilized to ensure the generation of globally unique identifiers for each match. In cases where match data are retrieved from a RESTful API server, match records are instantiated based on the received data, with their unique identifiers derived accordingly. To facilitate communication with external RESTful services, the Apache Commons Lang library [38] was incorporated.

The generation of PDF files was accomplished by using the native Android PdfDocument library. To provide encryption capabilities for the generated PDF files and to provide access to PDF metadata for PDF report updates and versioning, an Android-compatible port of Apache PDFBox, known as PdfBox-Android [39] was integrated. Additionally, QR codes embedded within the PDF document (for purposes such as verification or indexing) are generated using the ZXing Core library [40].

The following paragraphs will present the structure of the MZAP Android application. Each package contains various components that are organized based on their specific purpose and functionality. Currently, the application is divided into 12 packages, as detailed below:

- adapters: This package includes classes (adapters) designed to display lists of objects within the Spinner control.
- components: This package contains components (controls) specifically developed for the MZAP application.
- dialogs: This package encompasses all dialogs available within the MZAP application.
- entity: This package contains the classes required to facilitate the data necessary for match reports.
- entity_online: This package includes classes that handle the match data downloaded from the RESTful API server.
- filters: This package contains classes used to filter user input.
- fragments: This package comprises all fragments available within the MZAP application.
- interfaces: This package includes all custom interfaces implemented within the MZAP application.
- network: This package contains classes responsible for facilitating communication with the RESTful API server.
- utility: This package provides a collection of custom utility classes used throughout the application.
- viewmodel: This package contains ViewModel wrapper classes for certain entities from the entity package.
- wifi: This package includes wrapper classes and interfaces for the simplified sending and receiving of data over a WiFi network.

The activities within the MZAP application are derived from a custom base class, BaseActivity, which extends the standard Activity class. Each activity serves a specific purpose, as outlined below:

- BaseActivity: A custom base class for all activities, inheriting from the standard Activity class.
- MZAPCaptureSignatureActivity: An activity used to capture the signatures of basketball officials assigned to the tracked match.
- MZAPViewSignatureActivity: An activity that enables users to preview and delete the captured signatures of basketball officials assigned to the tracked match.
- MZAPViewImageActivity: An activity that previews the current match report.
- MZAPMainActivity: The main activity designed for both mobile phones and tablets. It contains a single FrameLayout that displays various application fragments.
- MZAPMainPanelActivity: An activity intended solely for tablets, containing two side-by-side FrameLayouts that load key fragments for tracking a basketball match: TeamsFragmentPanel and CurrentScoreFragmentPanel.
- MZAPWifiPreviewActivity: An activity that allows users to preview the match report from a designated server.

The current state of the application comprises a total of 15,605 lines of Java code, distributed across 106 source files, and an additional 8258 lines of XML markup within 71 XML configuration files. These figures specifically account for the primary application logic and interface definitions.

It is important to note that lines pertaining to auxiliary components, such as build scripts, support files, font assets (e.g., TTF files), and other non-core resources, have been deliberately excluded from the presented statistics to maintain a focused representation of the functional and structural codebase.

System Design

This chapter presents the data flow, concurrency management, and error handling mechanisms implemented within the MZAP application. For each of these aspects, a corresponding flowchart is provided to visually support the discussion. Additionally, a system integration diagram illustrates the architectural relationships between the MZAP Android application, the MZAP Server, and the MZAP Converter (currently under development), as components of the overall MZAP software suite.

The data flow model of the MZAP application, illustrated in Figure 1, is centered around a singleton object, the MZAP Match Record, which encapsulates multiple entities designed to store basketball match data. This object is coupled with the MZAP application's SharedPreferences, which acts as in-memory storage. Data flow between the aforementioned entities is bidirectional.

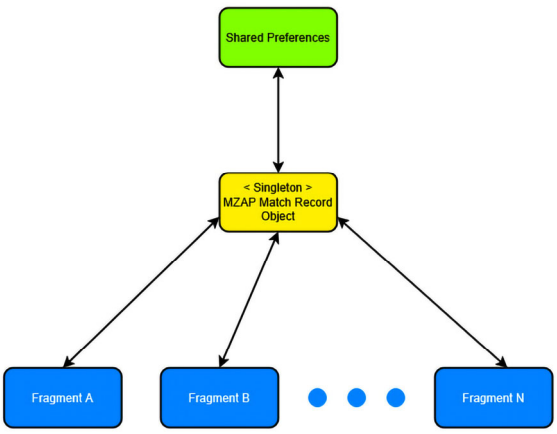


Figure 1. MZAP data flowchart.

Using a custom SharedPreferences manager, data are saved to the SharedPreferences storage every 10 s via an application-level thread, without compromising the user experience. Read and write operations are synchronized using a mutex mechanism (the synchronized keyword).

As various application fragments handle distinct segments of basketball match data, to each fragment that manages data, references to the particular object within the MZAP Match Record singleton are sent. Updates to the singleton MZAP Match Record are propagated to SharedPreferences via the previously described thread. Changes in fragment properties are reflected in the Match Record, and conversely, fragments are refreshed with data from the Match Record upon initialization or redraw.

Concurrency is managed using the ExecutorService. All processes related to network communication, complex object initialization, report generation, and similar tasks are executed as background tasks that can be canceled. Before each background task begins, a cancelable progress dialog is displayed to the user. If the task does not complete within the predefined maximum execution time, a timeout occurs. If the task finishes on time, its results are posted to the main thread and, provided the associated activity or fragment is still active, the user interface is updated accordingly. This concurrency flowchart is depicted in Figure 2.

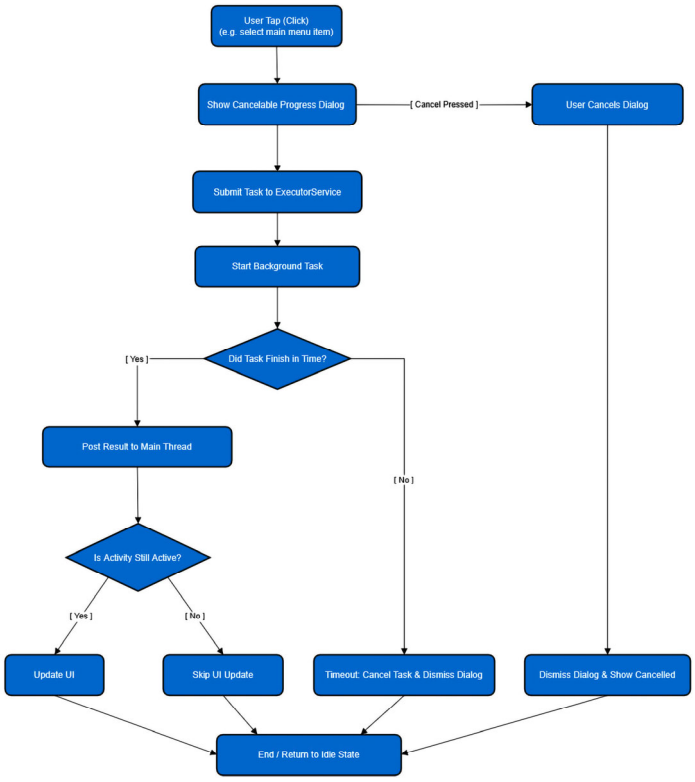


Figure 2. MZAP concurrency flowchart.

Error handling is implemented via an application-level thread managed by the ExecutorService. The error handler is initialized in the BaseActivity class, from which all other activities inherit. This thread waits for exceptions to occur, classifies each error, wraps it in a custom error object, logs the stack trace to a file, and notifies the user. This process is shown in Figure 3. When a user submits an error report, the corresponding log file containing the stack trace is also sent. This facilitates easier error reproduction, diagnostics, and error resolution.

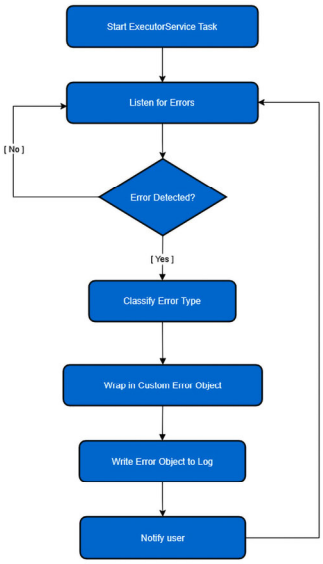


Figure 3. MZAP error propagation flowchart.

The MZAP software suite integration diagram is shown in Figure 4. Although developed using different technologies and programming languages, both the MZAP Android application (Java) and the MZAP Converter (Python v3.13.5) generate locked, encrypted, and digitally signed match endgame reports from a unified JSON Match Record Structure. While these applications do not interact directly, they both communicate with the MZAP server. By submitting a match UUID, each application requests credentials necessary for locking, encrypting, and signing the generated PDF documents. As previously stated, the MZAP Converter currently features only a command-line interface, through which each request is handled using a predefined set of commands. Once a PDF document has been created, both solutions transmit the generated PDF along with the corresponding JSON match record to the MZAP server for validation and archival.

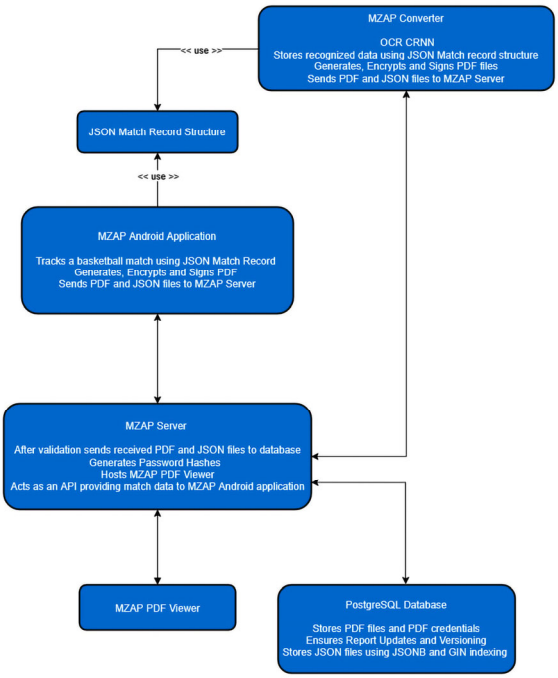


Figure 4. MZAP integration diagram.

Given that both solutions currently handle the generation, locking, encryption, and digital signing of PDF files, future development plans include migrating these functionalities to the MZAP server or creating a dedicated server to manage these processes.

The MZAP server functions as an API, providing match data to the MZAP Android application, generating password hashes required for PDF locking, encryption, and signing, and, following validation, storing the received PDF and JSON files in a database. Additionally, the server hosts a custom-built MZAP PDF Viewer.

The PostgreSQL database server stores the generated PDF match endgame reports along with the associated JSON match records. The JSON match records are stored using the JSONB data type and are indexed using a GIN index. PDF credentials are also maintained within the database, along with metadata necessary to support report updates and versioning of immutable PDF documents.

4. MZAP Android Application—Appearance and Usage

This chapter, along with its subsections, provides a detailed overview of the appearance and functionality of the MZAP Android application. To offer a comprehensive understanding of the features provided by the MZAP application, the discussion will begin with an examination of the application's Main Menu. Subsequently, each fragment of the application will be analyzed, with a focus on its appearance and specific functionalities.

4.1. Main Menu

The Main Menu is organized into four primary sections: Actions/Operations, Record, WiFi, and Application, as depicted in Figure 5. In Figure 5, the sections are presented side by side for clarity, whereas within the application, they are arranged vertically, one after the other. Each section contains specific menu items, the functionality of which will be discussed in the subsequent paragraphs.

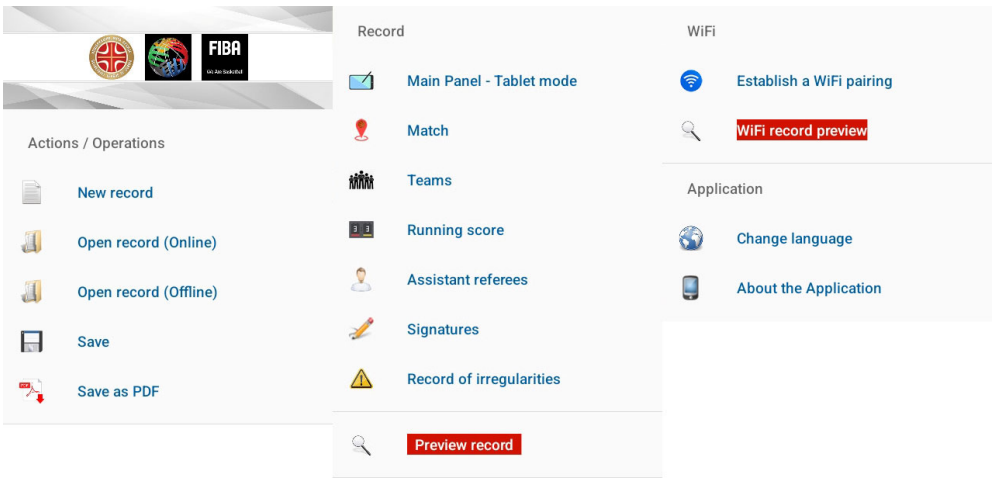


Figure 5. MZAP Main Menu.

The Actions/Operations section includes the following menu items:

- New Record: Creates a blank match record and discards the currently loaded record.
- Open Record (Online): Opens a match record to be played from an online database, discarding the currently loaded record.
- Open Record (Offline): Opens a match record from local storage and discards the currently loaded record.
- Save: Saves the currently loaded match record as a *.zap file in local storage and generates a match report in the form of a *.jpeg file.

- Save as PDF: Saves the currently loaded match record as a digitally signed PDF file to local storage and opens a share dialog for further actions.

The Records section includes the following menu items:

- Main Panel—Tablet Mode: Loads the MZAPMainPanelActivity, which is used to track a basketball match when the MZAP application is running on a tablet.
- Match: Loads the MatchBasicInfoFragment, which displays basic information about the basketball match.
- Teams: Loads the TeamsFragment, which presents basic information about the two teams (labeled Team A and Team B) whose match is being tracked.
- Running Score: Loads the CurrentScoreFragment, which is used to record three currently available in-game events: score (for one, two, or three points), the end of a period (quarter), and the end of the match.
- Assistant Referees: Loads the RecordersFragment, which contains data on the assistant referees and other basketball officials assigned to the match.
- Signatures: Loads the SignaturesFragment, which is used to collect or clear signatures from the basketball officials assigned to the match.
- Record of Irregularities: Loads the IrregularitiesRecordFragment, which allows the user to record any irregularities that occurred during the match.
- Preview Record: Generates the match report in the form of two *.jpeg files and previews them in the MZAPViewImageActivity.

The WiFi section includes two menu items designed to track, in real-time, whether basketball officials are properly monitoring the basketball match:

- Establish a WiFi Connection: When connected to a WiFi network with internet access, this option generates a five-digit random number to serve as a sharing key and begins transmitting the match report to a designated server every 3 s.
- WiFi Record Preview: When connected to a WiFi network with internet access, this option prompts the user to enter the sharing key. Upon entering the key, the MZAP-WiFiPreviewActivity is activated, displaying the match report by pulling data from the designated server. In this manner, an unlimited number of clients can simultaneously track a basketball match.

The Application section includes the following menu items:

- Change Language: Opens a dialog through which the application's language can be changed.
- About the Application: Displays the MZAP About dialog, which contains basic information about the application.

4.2. Match Fragment

The Match Fragment displays essential information about the basketball match, including the names of the teams playing, the name of the competition, the match number, the city where the match is being held, the date of the match, the start and end times, and the names of the referees assigned to officiate the match. Figure 6 illustrates the layout of the Match Fragment.

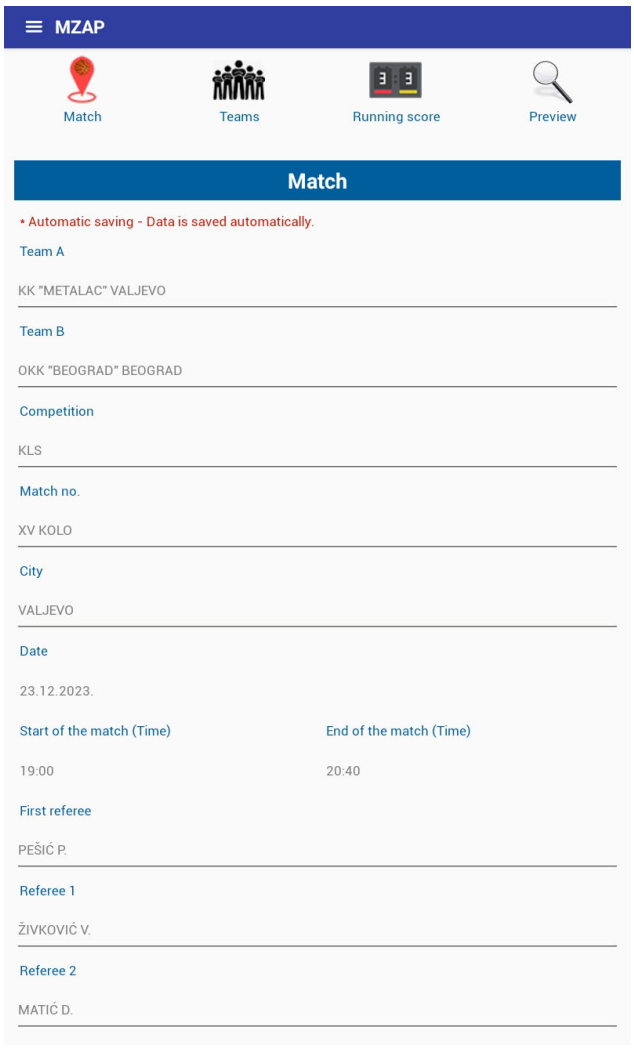


Figure 6. Match Fragment. The * annotations are the functional element of the application.

4.3. Teams Fragment

The Teams Fragment contains detailed information about the two teams playing the match, labeled as Team A and Team B, which are presented in two separate tabs with identical data fields. The data fields for each team include the following: the team name, a list of timeouts indicating the period and minute when each timeout was called, a list of team fouls per period (automatically populated based on individual player fouls), details of any challenges indicating the period and minute of occurrence, a list of players with highlighted parameters for each (to be explained in the subsequent paragraphs), and the names of the head coach and assistant coach, accompanied by five boxes for tracking the fouls each coach could incur during the game.

The players and their highlighted parameters are presented in a table format, with the following header columns: LNO (License Number), Players (Player names), NO (Number/Jersey), IG (In Game/Entered the Game), and F1–F5 (Fouls 1 to 5). When a user selects a player from the list, a Player Dialog is displayed. Similarly, selecting a foul box for any of the coaches triggers the Fouls Dialog. The functionality of the Player and Fouls dialogs will be further explained in subsequent chapters.

Each team has the option to add a player and to sort players in descending order based on the player list. All of the features described above are illustrated in Figure 7.

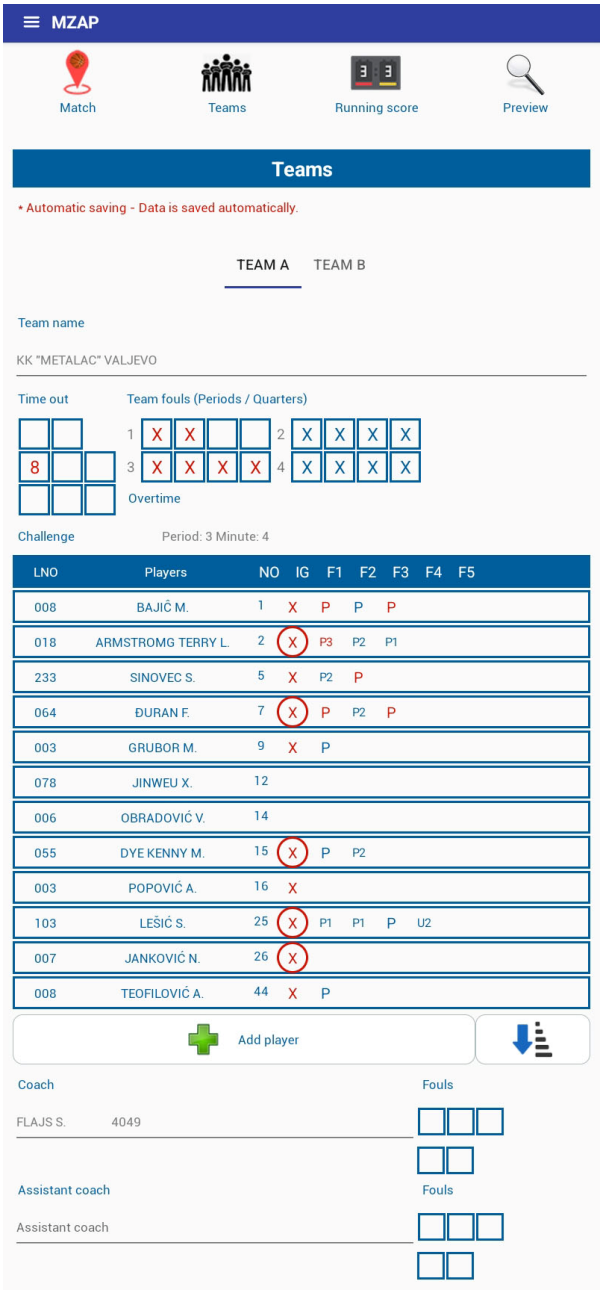


Figure 7. Teams Fragment. The * annotations are the functional element of the application.

Player and Fouls Dialogs

The Player Dialog, shown in Figure 8, displays detailed information about a player. This dialog allows for the addition, editing, or deletion of player data. When deleting a player, several confirmation dialogs appear, especially if the player has entered the game and scored points. The data fields for each player include the following: license number, captain status, surname and first name, player number (jersey), roster status, entry status (indicating the period in which the player entered the game), starting player status, and a set of boxes to track the player's fouls.

The Fouls Dialog, shown in Figure 9, provides information about a foul, specifically the type of foul and the period in which it occurred. The type of foul is selected by choosing the corresponding box labeled with the specific foul type. Upon selection, the box of the chosen foul turns red. This dialog is used to add, edit, or delete fouls committed by either a player or a coach.

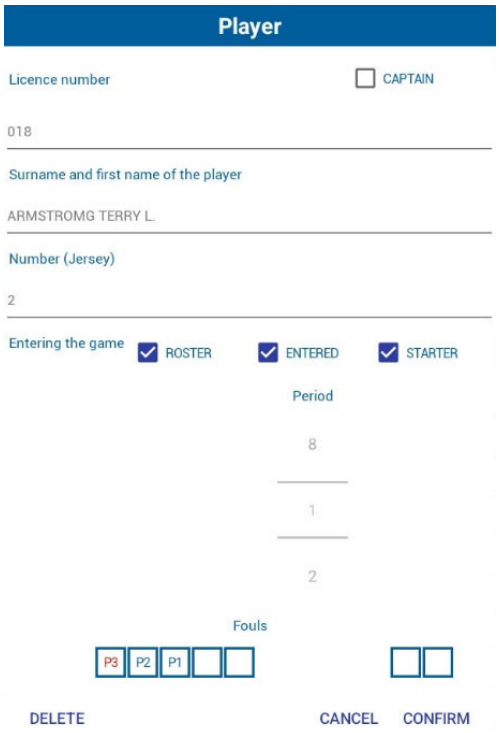


Figure 8. MZAP Player Dialog.

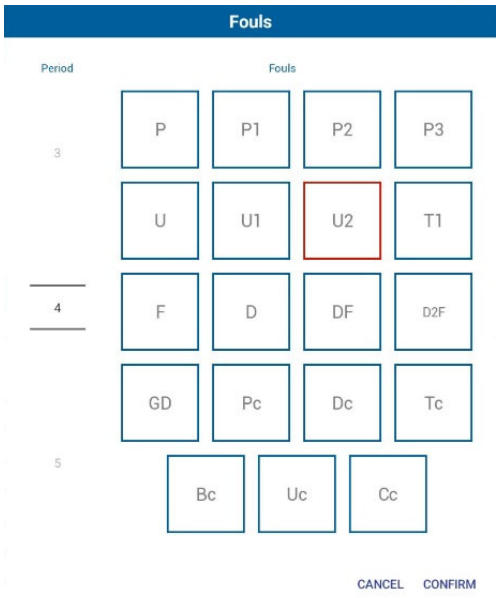


Figure 9. MZAP Fouls Dialog.

4.4. Running Score Fragment

The Running Score Fragment, shown in Figure 10, displays the overall running score as well as the score by periods. This fragment is also used to track the current period of the game and monitor the events occurring during the match, including scoring (for one, two, or three points), the end of a period (quarter), and the conclusion of the match.

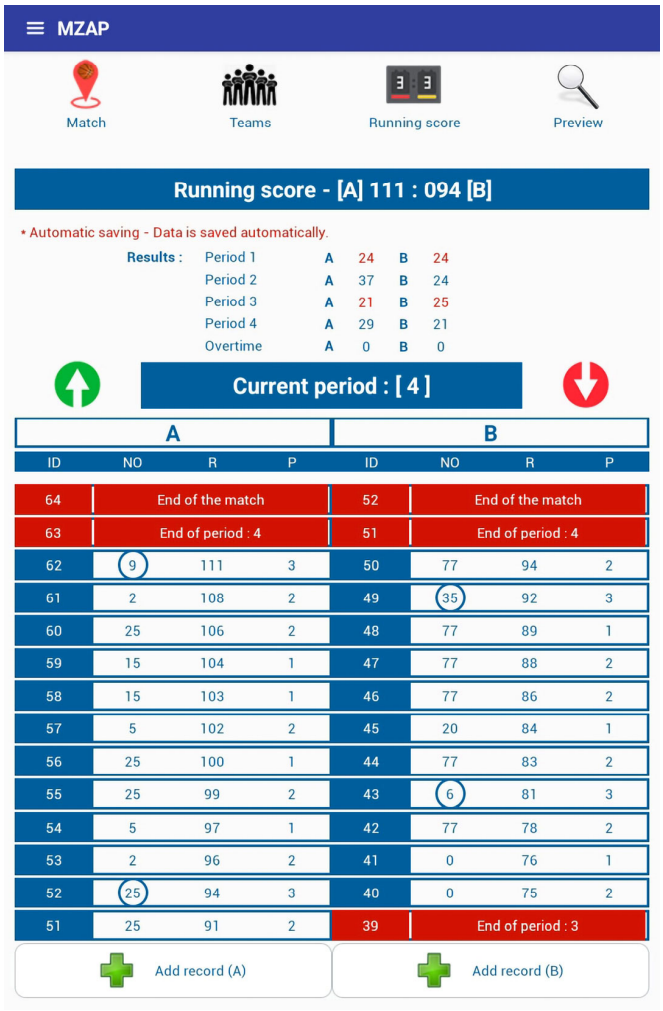


Figure 10. Running Score Fragment. The * annotations are the functional element of the application.

The event tables for Team A and Team B record points scored during the game and also capture two special events: the end of a period and the end of the match. Each entry includes the following fields: ID (event number), NO (player number/jersey), R (result), and P (points scored, from one to three). Special events are visually marked with red boxes indicating period end, overtime, or match conclusion.

Event recording is performed through the Records Dialog, which can be accessed by selecting the Add Record (A) button (for recording events for Team A) or the Add Record (B) button (for recording events for Team B), or by selecting any recorded event from the event tables of either team. This dialog allows users to add, edit, or delete the aforementioned game events. As shown in Figures 11 and 12, the Records Dialog consists of two tabs: Points and Match. The Points tab, depicted in Figure 11, contains a list of players from Team A or Team B, along with a list of point-scoring actions available for each player. The Match tab, shown in Figure 12, includes two buttons that, when selected, indicate the end of a period or the conclusion of the match.

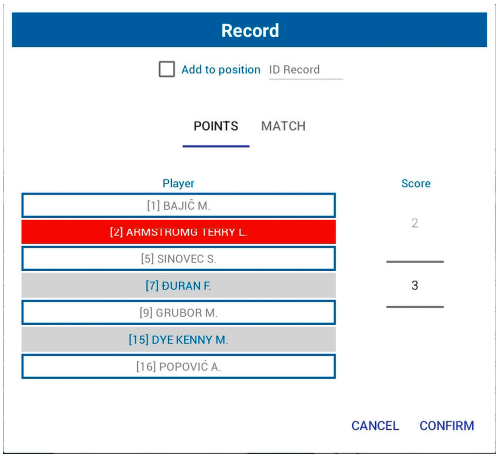


Figure 11. MZAP Record/Points Dialog.

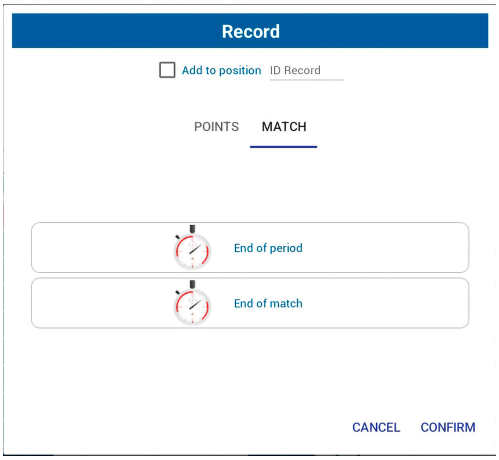


Figure 12. MZAP Record/Match Dialog.

4.5. Assistant Referees

The Assistant Referees Fragment, shown in Figure 13, provides information about the assistant referees. It includes four fields: Recorder, Assistant Recorder, Timer, and Timer 24″.

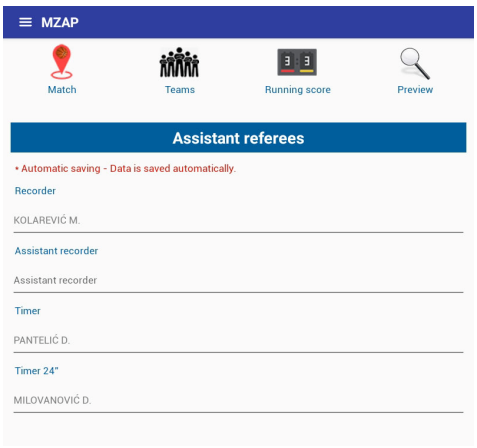


Figure 13. Assistant Referees Fragment. The * annotations are the functional element of the application.

4.6. Signatures

The Signatures Fragment (Figure 14) displays information about all officials assigned to a basketball match and provides functionality for capturing their digital signatures. A pencil icon and a red "No Signature" label indicate unsigned reports, while a magnifying

glass icon appears once a signature is captured. Tapping the pencil icon launches the MZAPCaptureSignatureActivity, enabling the official to sign the report.

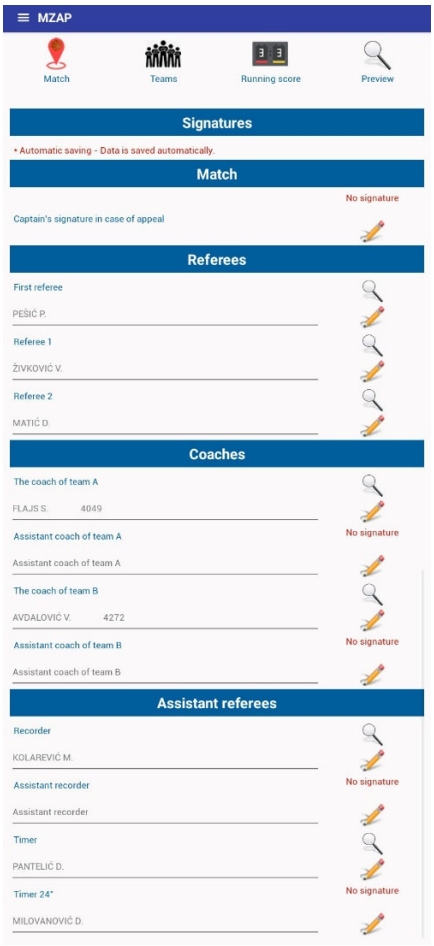


Figure 14. Signatures. The * annotations are the functional element of the application.

The MZAPCaptureSignatureActivity (Figure 15) provides a canvas for drawing signatures, along with Save and Cancel buttons to store or discard the input. Once a signature is saved, selecting the magnifying glass icon opens the MZAPViewSignatureActivity (Figure 16), which displays the signature using an ImageView and includes Undo and Close buttons for clearing the signature and exiting the view.

Signatures are stored as a Base64-encoded string within a JSON file that contains all the match data.



Figure 15. Depositing signature.

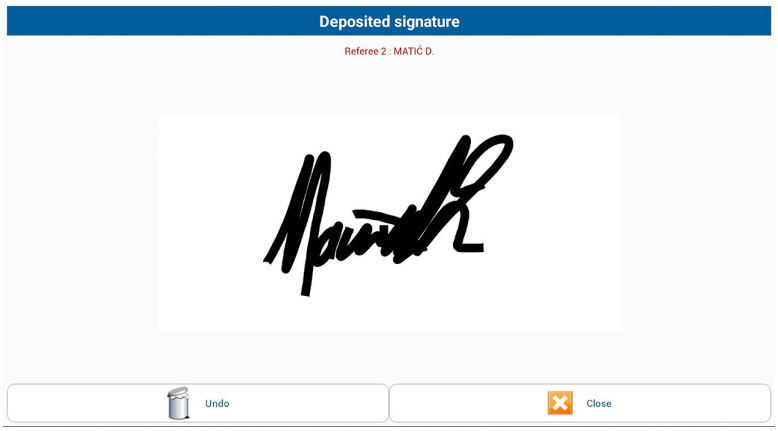


Figure 16. Deposited signature.

4.7. Record of Irregularities

The Record of Irregularities Fragment, shown in Figure 17, is used to document any irregularities that occurred during the tracked game and cannot be represented by fouls. These irregularities are reported in the form of text, which is stored in the Description of Irregularity field. The record of irregularities is considered valid if it has been signed by any of the referees or a delegate. Therefore, fields for depositing signatures, similar to those in the Signatures Fragment, are included.

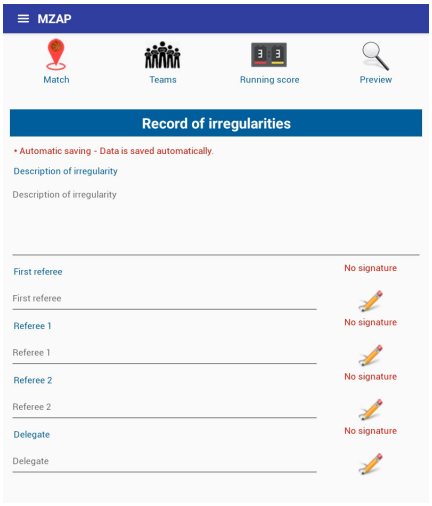


Figure 17. Record of irregularities. The * annotations are the functional element of the application.

This report is printed on the reverse side of the match report if any irregularities are documented. If no irregularities are reported, the reverse side of the report remains empty, and the generated PDF consists of only one page.

4.8. Main Panel—Tablet Mode

The Main Panel Activity, shown in Figure 18, is designed exclusively for tablets and serves as the most frequently used activity in the MZAP application. It consists of two side-by-side Frame Layouts, each loading the Teams and Running Score fragments. As a result, it provides all the essential features necessary for seamless tracking of a basketball match.

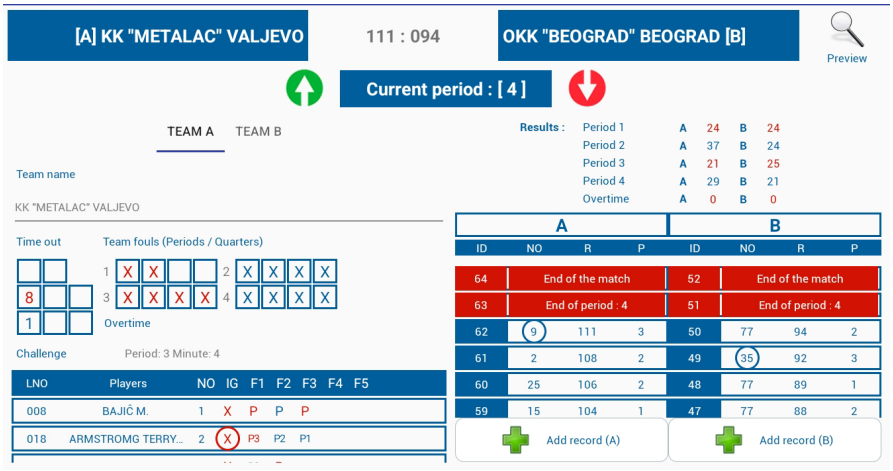


Figure 18. Main Panel—Tablet mode.

4.9. Loading Match Records (Online and Offline)

During use, MZAP automatically saves match record data to the application's Shared Preferences and allows exporting it as a .zap file to internal storage or an SD card via the Save option in the Main menu. Each record is assigned a unique UUID, which is used in the default file naming convention, though naming based on data from the Match Fragment is also supported.

When the user selects the Open Record (Offline) menu item, the Open Record Dialog (Figure 19) appears. This dialog acts as a custom file browser, showing only files with the .zap extension. Upon selecting a match record and confirming the dialog, the current match record in the application's Shared Preferences is replaced by the selected record, which is then loaded and displayed.

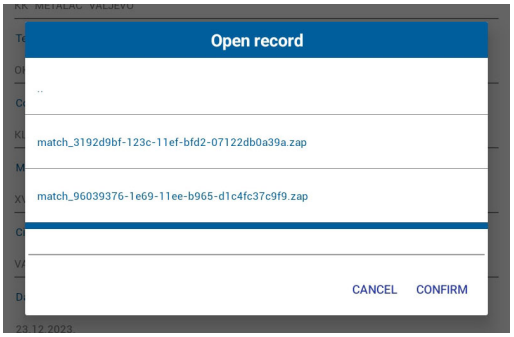


Figure 19. Open Record Dialog (offline).

When the user selects the Open Record (Online) menu item, the Open Record (Online) Dialog, shown in Figure 20, is displayed. This dialog features two spinners: one to display available leagues and another to display the basketball matches associated with the selected league.

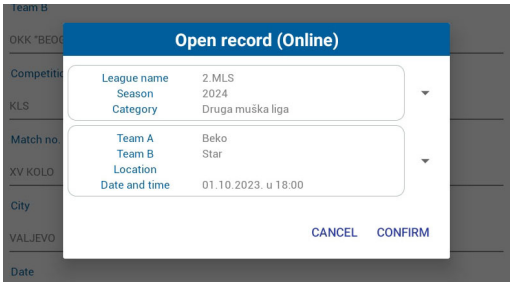


Figure 20. Open Record Dialog (online).

First, a list of leagues is loaded into the league spinner. Upon selecting a league, the list of matches associated with that league is displayed in the matches spinner below. If a user selects a match from this spinner and confirms the dialog, the current match record is discarded, and a new match record is generated based on information retrieved from the RESTful API server. It is important to note that this option does not load a complete match record from the online source; instead, it loads all available information regarding the match, players, referees, and other officials to prepare the MZAP application for match tracking. If the match record is not loaded from an online source, it must be manually created, which is time-consuming. Therefore, loading match data from an online source is the preferred option when using the MZAP application.

Data synchronization and consistency checks are performed when match reports are transmitted to the RESTful API server. There are two types of match records: those initiated on the MZAP Android application and those initiated on the RESTful API server. The primary distinction between these types lies in which application is responsible for generating the identifiers associated with match report entities.

When a match record is created on the MZAP Android application, both the match UUID and player identifiers are generated locally. These locally generated identifiers must be validated for inconsistencies upon transmission to the RESTful API server. Based on the match endgame data, a corresponding match record is created on the server. In this process, the server adopts the match UUID provided by the MZAP Android application and, using the received identifiers, attempts to resolve references to players, referees, coaches, and other officials. This ensures that the data is accurately stored in a relational database and that a valid copy of the match report is maintained through the insertion of scoreboard data.

Conversely, when a match record is retrieved from the RESTful API server, the MZAP Android application adopts the predefined match UUID and player identifiers to maintain consistency during subsequent data transmission back to the server. When this type of match record is received by the RESTful API server, it undergoes the same validation procedures as those applied to records initially created on the Android application.

At present, integration with existing league management systems is not prioritized. However, should such a requirement arise, integration would be implemented through designated RESTful API endpoints.

5. MZAP Android Application—Match Report Preview and Digital Signing

The first subchapter of this chapter examines the structural composition of the generated match endgame report and outlines the technologies employed in its development. Furthermore, it provides a detailed analysis of the information embedded within the QR code that accompanies each report.

The security mechanisms implemented to safeguard the integrity and confidentiality of the match endgame report are addressed in the second subchapter. Specifically, it explores the application of digital signatures and the implementation of password-protected encryption measures.

5.1. Match Report Preview

To preview a match report, an image containing the tracked match data, QR code, and timestamp is rendered on the canvas, saved to local storage, and then displayed using the MZAPViewImageActivity. Depending on the selected language of the application, different template bases are used, as illustrated in Figure 21.

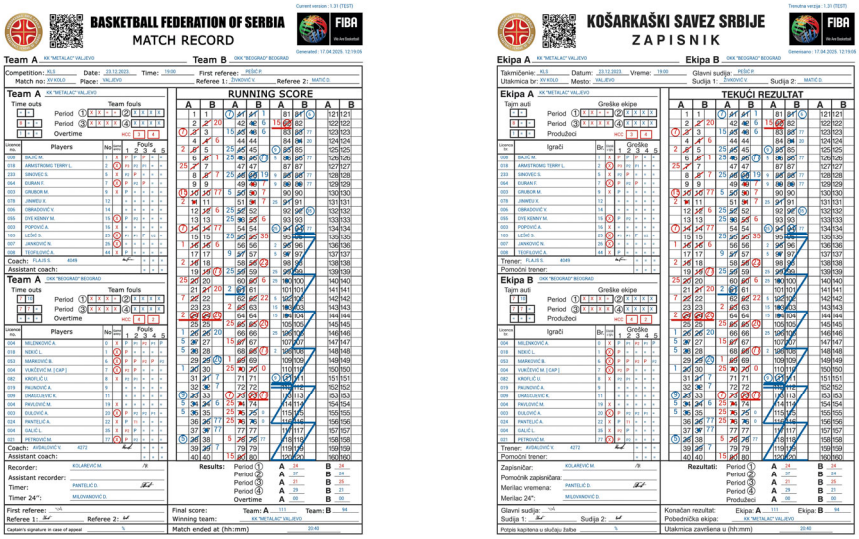


Figure 21. Match report preview in English (**left**) and in Serbian (**right**).

The QR code embedded in the match report contains a JSON object with three fields:

- Match UUID: This identifier can be utilized for various purposes, such as accessing detailed match statistics on the internet.
- Timestamp: This indicates when the match report was generated.
- Language Indicator: This field specifies the language of the generated report.

As outlined in previous chapters, the UUID is generated using the Java UUID Generator (JUG) library [37], while the QR code is created using the ZXing Core library [40]. The PDF file generation and its subsequent locking and encryption are accomplished using the standard Android PdfDocument and PdfBox-Android [39] libraries, respectively. To create the PDF match report, methods that generate a preview of the match report are used to render an image of the match report on the canvas of the first page. If any match irregularities are reported, a second page is created, and the details of the irregularities, along with the officials' names and signatures, are displayed on its canvas.

5.2. Security Mechanisms for PDF Match Endgame Reports

When a PDF match endgame report is generated, a request is sent to the MZAP server to provide the necessary data to secure it with a set of passwords and digitally sign the document. Once the report is digitally signed and password-protected, specific values that will be used to validate the integrity and authenticity of the generated document are returned to the MZAP server. The following chapters describe the MZAP security policies in detail.

5.2.1. Password Protection of a PDF File

Password generation is handled exclusively on the MZAP server, with each PDF being assigned two unique passwords. Prior to PDF generation, the match UUID is transmitted to the server. Both owner and user passwords are then generated using a cryptographically secure pseudo-random number generator (CSPRNG) [33], resulting in a 10-character string drawn from a custom ASCII character set. The passwords must include at least one uppercase letter, one digit, and one special character.

Subsequently, the generated passwords are hashed using the Argon2 algorithm [34], producing a 95-byte (127-character Base64) hash. A random 10-character salt (generated using the same method as the passwords) is used in the hashing process. The resulting password hashes are stored in the MZAP server's database along with the associated UUID.

The passwords for digital signing are transmitted alongside the password for encrypting the PDF file. Upon receiving the specified data, the generated PDF file is locked using a hashed owner's password (as shown in Figure 22), encrypted with AES-256 encryption algorithm [32] using a hashed user's password and then digitally signed.

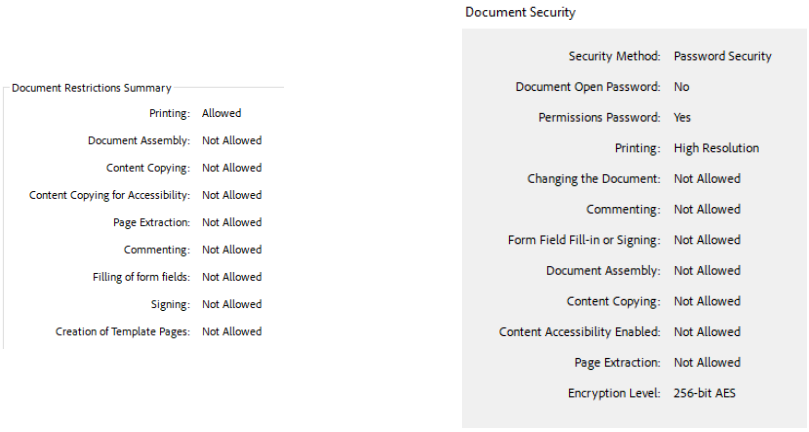


Figure 22. Match report PDF document properties.

5.2.2. Applying Digital Signatures to PDF Files

Digital signing of the document is performed using a PKCS#12 KeyStore, which is generated locally and embedded within the MZAP Android application. While the generated digital signatures use SHA-256 with RSA and are cryptographically valid and verifiable, the use of a self-signed X.509 certificate means they are not inherently trusted by standard verification mechanisms, as they lack endorsement from a recognized Certificate Authority (CA). However, as the certificate is self-signed, standard revocation mechanisms such as CRL or OCSP are not applicable. Certificate trust distribution and rotation are managed via server-side validation mechanisms and enforced through application updates.

In the event of certificate compromise, implemented revocation strategy relies on server-side blacklisting of the compromised certificate's unique identifier and mandatory application updates to replace the affected KeyStore. While this approach lacks real-time revocation capabilities provided by CA-backed PKI systems, it is sufficient within our narrowly scoped trust model. Despite its limitations (particularly the distribution and bundling of a PKCS#12 KeyStore) this approach was adopted since it suits the needs of the Basketball League of Serbia of internal use.

Future development plans aim to enhance security by integrating digital signatures with a Time Stamping Authority (TSA). Additionally, the process of digitally signing and password-protecting the PDF documents will be migrated entirely to the MZAP server. This will enable full utilization of certificate-based security mechanisms, leveraging CA-issued certificates typically delivered via USB tokens.

5.2.3. Trust Model and Verification

As stated earlier, upon receiving the hashed passwords, the MZAP application generates the PDF file and locks and encrypts it using the provided passwords. The PDF is then signed and a PDF file size in bytes is calculated followed by computation of a 64-character BLAKE3 [35] cryptographic hash of the entire PDF content. The accompanying JSON (*.mzap) file, that is alongside PDF file sent to the MZAP server, is similarly processed to obtain its file size and BLAKE3 cryptographic hash. All of these data are transmitted back to the MZAP server and associated with the previously stored password hash.

PDF file generation occurs only after the match report is finalized and no further modifications are expected. If a user wishes to review the report for inconsistencies, a

"Preview Report" option is available. In cases where the user needs to amend the report, the MZAP administrator must be notified to invalidate the previously stored data, thereby allowing the user to regenerate and re-sign the PDF.

The act of generating the PDF file signifies that the user has finalized the report and is requesting the document to be locked, encrypted, and digitally signed for archival purposes. Only PDF files that can be decrypted using the user's hashed password, unlocked using owner's hashed password, match the expected file size in bytes, and resolve to the predetermined cryptographic hash are accepted into the digital archive. These are the essential prerequisites for a file to be deemed valid, and they collectively serve to verify that the PDF has not been tampered with. Mentioned validation steps are performed on the MZAP server, a backend system developed using the .NET Core framework and the C# programming language. Since PDF files are locked, encrypted and digitally signed, a custom MZAP viewer was made to enable efficient document preview.

Server-side PDF certification validation using PDFsharp and BouncyCastle libraries involves several key steps. First, the PDF document containing a single hidden signature is opened using the appropriate password. Next, the internal structure of the PDF is carefully parsed to locate the embedded digital signature. In this case, the signature exists as an annotation object rather than a visible form field. For this signature, PDFsharp helps extract the raw cryptographic blob and the crucial byte ranges, which BouncyCastle then uses to compute a hash of the document's signed content. This computed hash is then compared against the hash embedded within the signature's cryptographic data to verify document integrity.

If the password is compromised, the PDF file could be altered. However, since the PDF consists solely of generated images, modifications require image editing software. Such edits typically result in changes to both the file size and content hash. Even if the file size remains unchanged, the content hash would differ, thereby flagging the file as altered. JSON files are validated similarly, excluding the hashed password requirement, as they are not password-protected.

The primary function of the QR code embedded in each PDF report is to store the match UUID, facilitating easier identification and lookup. To initiate report validation, the UUID is extracted from the QR code, and the tamper detection procedure is carried out using the associated hashed password, file size, and cryptographic hash.

5.2.4. Report Updates and Version Control

MZAP endgame reports are inherently immutable; that is, they are never deleted from the system. When a revision or update is required, the existing report is marked as invalid on the MZAP server rather than being modified or overwritten. In such cases, the current record is flagged as invalid by an administrator, and a new report is subsequently generated, locked, encrypted, digitally signed, and uploaded to the MZAP server for archival. During this process, new passwords are generated, and the entire signing and uploading procedure is repeated to ensure data integrity and authenticity.

For each updated report, a corresponding changelog is created and linked to the UUID of the match to which the report pertains. This enables robust traceability and supports auditability.

Each report is annotated with several metadata fields embedded as custom properties within the PDF document. These include the version number of the MZAP application used to generate the report, a timestamp indicating the exact time of generation, a server-managed document version number that is incremented with each update, and the UUID of the associated match. These fields, shown in Figure 23, collectively support version control, provenance tracking, and data consistency across the archival system.

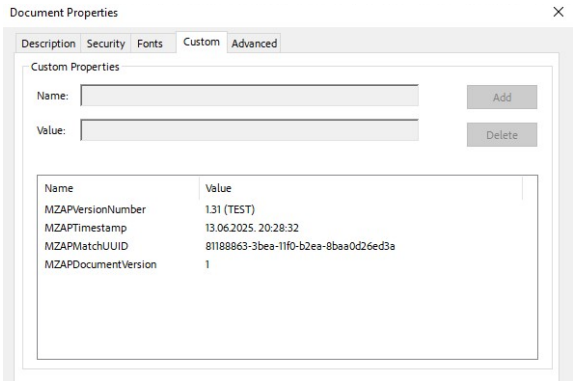


Figure 23. PDF custom properties (metadata) used for version control.

5.2.5. Threat Modeling and Asset Protection

Based on the current architecture and implementation of the MZAP Android application, several security threats have been identified, and corresponding mitigation mechanisms have been developed. The primary assets targeted for protection are the generated PDF files and the PKCS#12 KeyStore, which contains cryptographic keys essential for ensuring data integrity and non-repudiation. Table 1 presents the threat model for the MZAP Android application, outlining potential risks, impacted components, and implemented countermeasures.

Table 1. MZAP Android application threat model.

Threat	Description	Mitigation
PDF Tampering	Attacker modifies signed PDF	Modification invalidates signature
Key Extraction via Reverse Engineering	Attacker decompiles APK to extract PCKS#12	Obfuscated app, use server-side validation and signing
Impersonation (Fake Signature)	Attacker copies KeyStore to sign fake PDFs	Server-side checks for checksums and PDF passwords
Certificate Trust Failure	PDF readers show warnings for self-signed certificates	Educate users to accept certificates internally and use custom MZAP viewer
Brute Force on PDF Password	Weak passwords guessed to view content	Enforced strong passwords and AES-256 encryption

5.2.6. Legal Compliance and Data Protection

Since the application is currently used exclusively within Serbia and pertains specifically to the Basketball League of Serbia, the primary regulatory framework governing personal data processing is the Law on Personal Data Protection of the Republic of Serbia. Digital signatures are employed to ensure data integrity, thus protecting personal data from unauthorized alterations and to provide non-repudiation for audit purposes.

To mitigate the risk of private key exposure, the application implements in-app security measures. Trust in the digital signatures is strictly confined to the League's internal systems, which are explicitly configured to recognize and accept the League's self-signed certificate. This approach satisfies the accountability and trust requirements under Serbian data protection law.

If the application is expanded for use outside Serbia, it will be necessary to ensure compliance with the General Data Protection Regulation (GDPR) as part of the internationalization process.

6. MZAP Android Application—Adoption and the Effects of Digitalization

The results presented in this chapter were obtained through a survey administered to basketball officials. Of the 293 officials contacted, 271 completed the survey, corresponding to an adoption rate of 92.5%. This figure aligns with the number of MZAP application installations recorded on the tracking server, as the application is not distributed via Google Play and is instead installed manually.

Officials who did not participate in the survey stated several reasons for their non-adoption of the MZAP application: a preference for traditional paper-based match records, limited use of mobile devices, opposition to digitalization, or a refusal to disclose their rationale. A subset of officials expressed reluctance toward the system due to their familiarity with the paper-based process and a lack of trust in the new digital platform. Although these individuals have adopted the application, their mistrust has led them to work in teams wherein one official records data on paper while the other uses the mobile device, allowing both systems to run in parallel.

6.1. Adoption Rate and System Usability Scale

The adoption rate among officials aged 20 to 40 was 96.26%, with a 95% confidence interval (CI) ranging from 93.72% to 98.80%. For officials aged 40 and above, the adoption rate was 82.28%, with a 95% CI of 73.86% to 90.70%. These results, calculated from data presented in Table 2, show that younger officials (20 to 40 years old) were significantly more likely to adopt the MZAP app compared to older officials (40+ years). A chi-square test [41] confirmed this difference was statistically significant with following values: χ^2 (1, n = 293) = 14.29, p = 0.0002.

Table 2. User age groups and adoption of the MZAP application.

Age Group	Adopted	Not Adopted	Total
20–40	206	8	214
40+	65	14	79
Total	271	22	293

This disparity is expected, as younger officials are generally more accustomed to the use of digital technologies. Nevertheless, older officials have demonstrated commendable effort in adapting to the MZAP application's functionalities and workflow.

To evaluate the usability of the basketball endgame reporting application, we conducted a structured assessment using the System Usability Scale (SUS) as described and shown in papers by James R. Lewis [42] and Ozzi Suria [43], a widely recognized instrument for the rapid and reliable evaluation of system usability.

The usability test involved five distinct user groups: coaches (n = 27), delegates (n = 48), referees (n = 47), statisticians (n = 94), and other basketball officials (n = 55), each representing varying levels of technological proficiency. Following a training session on the use of the MZAP application, each participant was asked to complete an endgame report for a full basketball match. Immediately upon task completion, participants filled out the SUS questionnaire, which comprises 10 statements rated on a 5-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree).

The SUS scores ranged from 62.5 to 90.1, with a mean score of 79.12 and a standard deviation (SD) of 7.34, aggregated across all user groups, thus falling into the "Good" category. Table 3 presents the SUS scores disaggregated by user group.

Table 3. SUS scores for user groups.

User Group	SUS Score	SD (Standard Deviation)
Coaches	85.45	1.97
Delegates	86.40	2.92
Referees	77.06	2.80
Statisticians	81.62	2.15
Other	67.21	4.54

The reported SUS scores fall within the "Marginal," "Good," and "Excellent" usability categories. The Other group, which primarily includes recorders, timekeepers, and other supporting basketball officials, reported usability ratings within the Marginal range. In contrast, all other user groups rated the usability of the MZAP application as Good or Excellent.

6.2. Working Time Reduction and Decrease in Error Rates

Working time reduction results were obtained through a survey in which officials were asked to rate how much the use of the MZAP application has reduced their working time. A scale was created ranging from 0% to 50% in 5% increments, with intermediate options included to represent percentages that fall between two main steps. These intermediate options were denoted with a plus sign next to the main step—e.g., "5+%" represents a percentage greater than 5% but less than 10%. The survey results regarding the reduction in working time are presented in Table 4.

Survey results indicate that the MZAP application has led to a reduction in basketball officials' working time, highlighting the over 40% reduction in the majority of cases while significantly surpassing the anticipated 30% reduction in 87.82% of cases. Table 4 shows that the majority of basketball officials (169 officials, or 62.36% of those surveyed) reported a working time reduction of 40% or more. When responses indicating a reduction between 30% and 40% are also included, the total number of officials who experienced a minimum 30% decrease in working time rises to 238, or 87.82%. This surpasses the initial goal of achieving at least a 30% reduction. Notably, all officials reported a reduction in working time, with only two mentioning the smallest reduction of 10%.

A primary contributor to this efficiency gain is the application's ability to facilitate real-time error detection and correction. During games, multiple officials can monitor the input of one official and immediately identify discrepancies. Previously, substantial time was allocated for checking the report and post-game revisions, and scanning and uploading finalized match reports. With the implementation of the MZAP application, all data are instantly available to online viewers upon finalization of the match report.

Survey results also indicate that, due to improved control measures, the likelihood of an official making an error when using the MZAP application is now estimated between 3% and 7%, compared to an estimated 12% to 21% error rate when using the manual (paper scoresheet) system. This improvement is attributed to the constant display of the match report on a large screen, allowing multiple officials to monitor and verify the data being entered. Nonetheless, occasional input errors persist, often due to the rapid pace of in-game events.

Table 4. Decrease in working time survey results.

Working Time Reduction	Number of Officials	Percentage
40+%	107	39.48%
40%	62	22.88%
35+%	21	7.75%
35%	27	9.96%
30+%	13	4.8%
30%	8	2.95%
25+%	5	1.85%
25%	10	3.69%
20+%	7	2.58%
20%	9	3.32%
10%	2	0.74%
Total	271	100%

6.3. Increase in Website Traffic

The digitalization of match endgame reports has significantly improved their accessibility, resulting in an approximate 11% increase in traffic to the live score website and a corresponding rise in advertising revenue. Moreover, the searchable format of digitalized reports has proven beneficial to both sports statisticians and stakeholders in the betting industry.

The increase in website traffic was calculated based on server logs of the Basketball League of Serbia from the first six months of the 2024/2025 season. The total number of unique visitors for each month was aggregated and compared to the corresponding period of the previous season, revealing an increase of 11.02% $\approx$ 11%. Month-by-month comparisons revealed that the percentage increase in traffic, rounded up to 2 decimal places, varied between 8% and 12.69%. However, if the monthly increase was averaged, it amounts to 11.12%, which closely aligns with the previously stated figure of 11% (11.02%). Comparative Basketball League of Serbia's website traffic data are shown in Table 5.

Table 5. Comparative Basketball League of Serbia's website traffic data.

Month	Season 2023/2024	Season 2024/2025	Increased Traffic
October	37,142	41,854	12.69%
November	31,202	34,946	12%
December	32,202	36,388	13%
January	39,106	42,234	8%
February	36,645	40,310	10%
Match	37,254	41,352	11%
Total	213,551	237,084	
	Traffic increase: 11.02% $\approx$ 11%		Average: 11.12%

Given that an increase in traffic was observed consistently in each month when compared to the same month of the prior season, it is reasonable to project that this upward trend will continue for the remainder of the current season. Furthermore, it is anticipated that the traffic levels in the upcoming season will remain within a similar range as the current season or experience a slight decline.

7. MZAP Android Application—User Feedback, Failure Cases, and Error Taxonomy

Given that the Agile software development methodology was employed, a continuous testing process was integrated into the development cycle. A test group consisting of eight individuals was engaged throughout development and to this day remains active. These users provided continuous feedback through detailed email correspondence, addressing various aspects of the application, such as usability, visual clarity, and the interpretation of user interface (UI) elements. Feedback was also obtained from other basketball officials after MZAP application training sessions.

The development process was guided by the regulatory requirements set forth by the International Basketball Federation (FIBA) and the Basketball League of Serbia. As such, while most user suggestions were implemented to improve application stability, usability, and interface clarity, feedback that conflicted with these governing rules was not incorporated.

Initially, the MZAP application was designed primarily for use on mobile phones, with a focus on small-screen optimization and menu-based fragment navigation. However, user feedback indicated a strong preference for a layout more suitable for larger displays. Specifically, users expressed dissatisfaction with the need for frequent fragment switching and requested a unified interface that would consolidate key components, such as the Teams and Running Score sections. In response, a tablet-optimized mode was developed and has since become the primary interface used for match tracking and data entry.

The majority of observed failure cases were attributed to unanticipated edge cases in user interaction and unhandled input scenarios. These issues were not related to object or list initialization, which was consistently handled correctly throughout the development process. Instead, the failures primarily stemmed from insufficient coverage of atypical usage patterns and incomplete input validation, or, simply put, minor bugs in code. A similar failure case is described in the work of Wong et al. [44].

To better understand the nature of user errors presented in the work of Palanque et al. [45] and form an error taxonomy, we have conducted an analysis of the previously mentioned user feedback. Based on this analysis, the errors that we encountered are categorized into three primary groups:

- Input Errors—These involved the selection of incorrect options within the application, such as selecting the wrong player, foul type, point value, etc.
- Omission Errors—These referred to instances where users failed to record an event entirely. Users most commonly omitted the end-of-quarter and end-of-game events. Additionally, users frequently neglected to update a player's game entry status, which is a prerequisite for that player to appear on the list of active participants eligible to score. Failure to perform this step often led to subsequent input errors, as the intended player would not be available for selection during scoring actions.
- Navigation Confusion—These errors stemmed from difficulties in navigating the user interface, particularly before the introduction of the tablet-optimized mode, which significantly improved usability.

Due to this, taxonomy design improvements that were aimed at reducing the frequency and impact of the mentioned errors were devised and implemented. Before the introduction of the tablet-optimized mode, input errors accounted for 34% of all recorded issues, followed by omission errors at 25%, and navigation-related errors at 41%. After the deployment of the tablet mode, a significant reduction in navigation-related errors was observed, with this category decreasing to just 4% of total errors. Conversely, input errors increased to 62%, becoming the most prevalent category, while omission errors rose to 34%.

8. MZAP Android Application—Scalability, Selected Benchmarks and Application Performance

The selected performance benchmarks for the MZAP Android application are designed with mid-tier devices in mind and include the following criteria: App Launch Time (target: warm start under 1.5 s, cold start under 2.5 s), Local Data Storage Performance (target: read/write operations under 200 ms), Battery Usage (target: no more than 2–3% battery drain per 10 min of active usage), Memory Usage (target: less than 300 MB of RAM), and UI Responsiveness (target: 60 FPS and above).

Currently, the majority of officials are running the MZAP application on the Lenovo Tab M9, which is classified as an entry-level (budget-tier) device. Although the application demonstrates satisfactory performance on stated hardware when evaluated against benchmarks intended for mid-tier devices, there is no guarantee that the application scales effectively across a diverse range of device classes.

To address this, future testing of the MZAP application will involve a broader set of devices under real-world usage conditions, with ongoing evaluation based on the aforementioned benchmarks.

9. Conclusions

The MZAP Android application, as part of the broader MZAP software package, offers a solution that focuses on the digitalization of basketball endgame match reports, while simultaneously providing a high-level match tracking functionality. As previously discussed, the primary objective of the MZAP Android application is not to function as a comprehensive match tracking or statistical analysis tool; however, it facilitates the collection of reliable match tracking data, which can be utilized to generate statistical reports. Due to real-time Wi-Fi monitoring by multiple observers during games, match tracking is considered highly reliable, and the likelihood of an official making an error is significantly reduced. As the statisticians' team records a broader range of match data compared to the endgame match report team, an initiative has been proposed to develop an extension that checks for inconsistencies between the data collected by these two teams in real time, thereby enhancing the overall reliability of both collected data and the software itself.

Given that online data availability is increasingly regarded as a fundamental requirement in the current era, a key limitation of the MZAP software is its reliance on local data storage, with data being transmitted online only after the conclusion of a match. However, if real-time online access to MZAP data becomes necessary, this limitation can be addressed by implementing a dedicated thread that periodically transmits match data to a designated web API endpoint.

The application is designed for both flexibility and scalability, allowing it to support various basketball leagues with minimal alterations within the source code, leading to a new customizable solution. Theoretically, the MZAP application could be generalized to track matches and generate reports of other sports. In that case, application logic and rules of the game, along with UI, would lie in templates that are called upon during compilation or runtime.

In addition to its focus on digitalization, the MZAP Android application stands out from other similar solutions by incorporating a feature for capturing the signatures of basketball officials. This feature adds an additional layer of accountability and serves as a preventive measure against data tampering. By requiring officials to sign the endgame match report, they are personally guaranteeing the validity of the recorded data. Should the data be found to be inaccurate, officials who have signed the report may be held accountable and face appropriate consequences. Furthermore, the security measures applied to the PDF

report ensure the integrity of the document by preventing unauthorized modifications and providing mechanisms to detect any alterations.

The digitalization of endgame match reports has led to a measurable reduction in the working time of basketball officials, exceeding expectations. Given that 87.82% of officials reported at least a 30% reduction in working time (a threshold we set as our goal), it can be concluded that the MZAP application has had a significant positive impact on workflow efficiency. The subgroup reporting a reduction of 40% or more (62.36% of surveyed officials) represents the most favorable outcomes, and it serves as an indicator of the application's potential as an effective time-saving tool.

The error rate dropped by more than half when transitioning from a manual scoresheet system to using the MZAP application. This significant reduction suggests that implementing digital systems like MZAP can greatly minimize human error in official tasks, enhancing both accuracy and efficiency.

Additionally, digitalization has resulted in an estimated 11% increase in traffic to the live score website, accompanied by a corresponding growth in advertising revenue. While these improvements are immediately observable, it is anticipated that further benefits associated with digitalization will emerge over time.

While the MZAP Android application reduces officials' working time and improves data accuracy by minimizing human error through built-in validation and real-time monitoring, its System Usability Scale (SUS) scores range from "Marginal" to "Excellent". The "Other" group of users (recorders, timekeepers, and support officials) rated usability as "Marginal", whereas all other user groups rated it as "Good" or "Excellent."

Future objectives for the MZAP software package include:

- The continued development and integration of the MZAP Converter software, which was briefly introduced in this paper.
- Examine if CRNN models can outperform standard off-the-shelf OCR solutions in domain-specific digitalization.
- The creation of a configuration module to enable the customization of the endgame match report format within the MZAP Android application, making it adaptable to any basketball league without source code customizations. This would be done through the introduction of templates, a generic event system, and dynamic UI binding to enable easier integration between the data model and the user interface. Plugin architecture is also being considered for this feature.
- Optional integration of the LiveStats module to enhance the MZAP Android application through the implementation of a dedicated thread that periodically transmits match data to a designated web API endpoint.
- The potential addition of user accounts that will enhance security by introducing an additional layer of access control. Administrators will be granted the authority to revoke user accounts in the event of any type of violation.
- UX (User eXperience) optimization by constantly improving the user interface in order to further reduce the possibility of user input errors.
- Integration with a TSA server and migration of password protection and digital signing of PDF documents to the MZAP server to fully utilize CA-issued certificates, typically delivered as USB tokens.
- Addition of support for interface languages beyond the currently supported Serbian and English. Language support would be implemented through application updates or downloadable language files from the MZAP server's central repository.

In addition to the presented findings and contributions, this research integrates aspects of both a case study and a proof of concept.

Author Contributions: Conceptualization, P.P. and B.M.; methodology, B.M.; software, P.P.; validation, P.P. and B.M.; formal analysis, P.P. and B.M.; resources, B.M.; data curation, B.M.; writing—original draft preparation, P.P.; writing—review and editing, P.P. and B.M.; visualization, P.P.; supervision, P.P. and B.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We would like to express our sincere gratitude to the Basketball League of Serbia for their wonderful support and cooperation.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. FIBA. *Digital Scoresheet*; FIBA: Mies, Switzerland, 2024. Available online: https://about.fiba.basketball/en/services/data-and-video-solutions/digital-scoresheet (accessed on 21 April 2025).
2. FIBA. *FIBA LIVESTATS*; FIBA: Mies, Switzerland, 2025. Available online: https://www.fibaorganizer.com/ (accessed on 21 April 2025).
3. Statastic. Statastic Basketball Tracker. Available online: https://www.statastic.info/ (accessed on 21 April 2025).
4. Basketball Stats Assistant. 2024. Available online: https://www.basketstatsapp.com/ (accessed on 23 April 2025).
5. Easy Stats. Easy Stats for Basketball. Available online: https://www.easystatsapp.com/ (accessed on 23 April 2025).
6. Breaktrough Basketball. Breakthrough Stats App. Available online: https://www.breakthroughbasketball.com/apps/stats/ (accessed on 21 April 2025).
7. iScore Sports. iScore Basketball Scorekeeper. Available online: http://iscoresports.com/basketball/ (accessed on 21 April 2025).
8. yukitaka13. Basketball Digital Scoresheet. GooglePlay. Available online: https://play.google.com/store/apps/details?id=com.yukitaka13.basketballScoresheet (accessed on 23 April 2025).
9. Riza, A.R.; Azandi, F.; Ilham, Z.; Sagala, R.S.; Siregar, F.S. Effectiveness of App-Based Basketball Scorers for Basketball Games. *Kinestetik J. Ilm. Pendidik. Jasm.* **2022**, *6*, 724–730. [CrossRef]
10. Ivanković, Z.; Racković, M.; Markoski, B.; Radosav, D.; Ivković, M. Appliance of Neural Networks in Basketball Scouting. *Acta Polytech. Hung.* **2010**, *7*, 167–180.
11. Chomątek, Ł.; Sierakowska, K. Automation of Basketball Match Data Management. *Information* **2021**, *12*, 461. [CrossRef]
12. Pitić, G.; Savić, N.; Verbić, S. Digital transformation and Serbia. *Econ. Enterp.* **2018**, *66*, 107–119. [CrossRef]
13. Mijajlović, Ž.; Ognjanović, Z. A survey of certain digitization projects in serbia. *Rev. Natl. Cent. Digit.* **2004**, *4*, 52–61.
14. Mijajlović, Ž.; Ognjanović, Z.; Pejović, A. Digitization of Scientific Journals in Serbia. In Proceedings of the ELPUB2007 Conference on Electronic Publishing, Vienna, Austria, 13–15 June 2007.
15. Zaušková, A.; Kusá, A.; Kubovics, M.; Ščepková, S.; Urmínová, M. Current state and prediction of the future of digitization as a part of industry 4.0. *Serbian J. Manag.* **2022**, *17*, 111–123. [CrossRef]
16. Xiao, X.; Hedman, J.; Ter Chian Tan, F.; Tan, C.-W.; Lim, E.T.K. Sports Digitalization: An Overview and A Research Agenda. In Proceedings of the Thirty Eighth International Conference on Information Systems, Seoul, Republic of Korea, 10–13 December 2017; pp. 1–21.
17. Santomier, J.; Dolles, H.; Kunz, R. The National Basketball Association's (NBA) Digital Transformation: An Explanatory Case Study. *Qual. Sport* **2023**, *11*, 63–80. [CrossRef]
18. Poulsen, N.J.L. *Digital Transformation in Sports: How Digital Transformation Influence Value Creation*; Copenhagen Busineess School: Copenhagen, Denmark, 2021.
19. Mamedov, R.G. Digital transformation of the sports industry: Trends and business practice. *Ekon. Upr. Probl. Resheniya* **2022**, *6/1*, 66–74. [CrossRef]
20. Vijaywargi, A.; Boddapati, U.K. Architectural Patterns in Android Development: Comparing MVP, MVVM, and MVI. *Int. J. Res. Appl. Sci. Eng. Technol.* **2024**, *12*, 4611–4616. [CrossRef]
21. Zakaria, A.H.; Nuryana, I.K.K.D. Android Software MVVM and MVP Architecture Analysis with iTourism App Case Study. *J. Inform. Comput. Sci. JINACS* **2023**, *4*, 351–357.
22. Li, Y.; Ouyang, J.; Mao, B.; Ma, K.; Guo, S. Data Flow Analysis on Android Platform with Fragment Lifecycle Modeling and Callbacks. *EAI Endorsed Trans. Secur. Saf.* **2017**, *4*, 1–11. [CrossRef]

23. El-Zawawy, M.A. Modeling Android Fragments and Activities Life Cycles Coordination. In Proceedings of the Computational Science and Its Applications-International Conference on Computational Science and Its Applications-ICCSA 2017, Trieste, Italy, 3–6 July 2017; pp. 620–634. [CrossRef]
24. Waja, G.; Shah, J.; Nanavati, P. Agile software development. *Int. J. Eng. Appl. Sci. Technol.* **2021**, *5*, 73–78. [CrossRef]
25. Ghimire, D.; Charters, S. The Impact of Agile Development Practices on Project Outcomes. *Software* **2022**, *1*, 265–275. [CrossRef]
26. Lin, W. Digital Signature. In *Trends in Data Protection and Encryption Technologies*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 77–81.
27. Aggarwal, S.; Kumar, N. Digital signatures. In *The Blockchain Technology for Secure and Smart Applications Across Industry Verticals*; Academic Press: New York, NY, USA, 2021; Volume 121, pp. 95–107.
28. Maña, A.; Matamoros, S. Practical Mobile Digital Signatures. In *E-Commerce and Web Technologies (EC-Web 2002)*; Elsevier: Amsterdam, The Netherlands, 2002; pp. 224–233.
29. Afrianto, I.; Heryandi, A.; Finadhita, A.; Atin, S. Design Of E-Document System With Digital Signature Using User Centered Design Method. In Proceedings of the Conference SENATIK STT, Yogyakarta, Indonesia, 10 December 2019; pp. 345–355. [CrossRef]
30. Jiang, L. A research on PDF protection mechanisms. In Proceedings of the 5th International Conference on Computing and Data Science, Jaipur, India, 23–25 November 2023; pp. 159–165. [CrossRef]
31. Singh, A.; Tiwari, V.; Naidu, A.S.; Tentu, A.N.; Raju, K.S.; Saxena, A. Analysis of Password Protected Documents Using Statistical Approaches on High Performance Computing. In Proceedings of the Sixth International Conference on Microelectronics, Electromagnetics and Telecommunications (ICMEET 2021)-Advances in Micro-Electronics, Embedded Systems and IoT, Bhubaneswar, Odisha, India, 27–28 August 2022; Volume 1, pp. 533–545. [CrossRef]
32. Sarkar, B.; Saha, A.; Dutta, D.; Sarkar, G.D.; Karmakar, K. A Survey on the Advanced Encryption Standard (AES): A Pillar of Modern Cryptography. *Int. J. Comput. Sci. Mob. Comput.* **2024**, *13*, 68–87. [CrossRef]
33. Luengo, E.A.; Villaizán, J.R. Cryptographically Secured Pseudo-Random Number Generators: Analysis and Testing with NIST Statistical Test Suite. *Mathematics* **2023**, *11*, 4812. [CrossRef]
34. Biryukov, A.; Dinu, D.; Khovratovich, D. Argon2: New Generation of Memory-Hard Functions for Password Hashing and Other Applications. In Proceedings of the 2016 IEEE European Symposium on Security and Privacy (EuroS&P), Saarbruecken, Germany, 21–24 March 2016; pp. 292–302. [CrossRef]
35. Mane, O.; Koravi, S.; Tarale, H.; Renukdas, A. Using Blake3 hash value as a key. *Int. Res. J. Mod. Eng. Technol. Sci.* **2025**, *7*, 8431–8437.
36. Google. gson. Available online: https://github.com/google/gson (accessed on 23 April 2025).
37. Saloranta, T. Java UUID Generator. Available online: https://github.com/cowtowncoder/java-uuid-generator (accessed on 23 April 2025).
38. Apache. Apache Commons Lang. Available online: https://commons.apache.org/proper/commons-lang/ (accessed on 23 April 2025).
39. Roush, R. PdfBox-Android. Available online: https://github.com/TomRoush/PdfBox-Android (accessed on 23 April 2025).
40. ZXing. ZXing. Available online: https://github.com/zxing/zxing (accessed on 21 April 2025).
41. Franke, T.M.; Ho, T.; Christie, C.A. The Chi-Square Test: Often Used and More Often Misinterpreted. *Am. J. Eval.* **2012**, *33*, 448–458. [CrossRef]
42. Lewis, J.R. The System Usability Scale: Past, Present, and Future. *Int. J. Hum.-Comput. Interact.* **2018**, *34*, 577–590. [CrossRef]
43. Suria, O. A Statistical Analysis of System Usability Scale (SUS) Evaluations in Online Learning Platform. *J. Inf. Syst. Inform.* **2024**, *6*, 992–1007. [CrossRef]
44. Wong, W.E.; Li, X.; Laplante, P.A. Be more familiar with our enemies and pave the way forward: A review of the roles bugs played in software failures. *J. Syst. Softw.* **2017**, *133*, 68–94. [CrossRef]
45. Palanque, P.; Cockburn, A.; Gutwin, C. A Classification of Faults Covering the Human-Computer Interaction Loop. In Proceedings of the 39th International Conference, SAFECOMP 2020, Lisbon, Portugal, 16–18 September 2020; pp. 434–448. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.