University of Sheffield

Ice Hockey Individual Performance Analysis Tool - Tablet Application

Radu-Mihai Niculae

Supervisor: Dr. Ramsay Taylor

A report submitted in fulfilment of the requirements
for the degree of BSc in Computer Science

in the

Department of Computer Science

May 10, 2023

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Radu-Mihai Niculae

Signature: RMN

Date: May 10, 2023

Acknowledgements

Firstly, I would like to express my gratitude to my supervisor, Dr. Ramsay Taylor, for his invaluable guidance and constant feedback throughout this project.

Secondly, I would like to dedicate this work to my family, who provided unconditional support, patience, and understanding during my moments of complaint and self-pity throughout this journey. Thank you Toni, Magda, Andrei, Ioana and Vlad.

Abstract

Given the fast-paced and continuous nature of ice hockey, the assessment of players' performance, its meaningful and intuitive representation, and the prediction of future performances pose considerable challenges. A majority of players lack access to effective tools that address these concerns. The situation is further compounded by the lack of data that can assist in data analysis, machine learning, and statistical evaluations. This project seeks to mitigate these limitations by developing a software tool that enables the capture of essential in-game actions and the conversion of this data into valuable insights, which are then presented in a comprehensive and well-structured report and can also be used as input samples for predictive models. The report's clear and straightforward format allows coaches, players, and scouts to easily interpret the information, facilitating the assessment of the player's performance. By integrating a variety of visual elements and employing an SVM algorithm, the tablet application aims to streamline the evaluation and prediction processes for players' performance. Consequently, this innovation contributes to the enhancement of the sport's competitiveness and efficacy, offering a practical solution to the complex challenges faced by the ice hockey community.

”We need to have the strength and power of a football player, the stamina of a marathon runner, and the concentration of a brain surgeon. But we need to put all this together while moving at high speeds on a cold and slippery surface while 5 other guys use clubs to try and kill us. Oh yeah, did I mention that this whole time we’re standing on blades 1/8 of an inch thick”

Brendan Shanahan - Retired NHL Player

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Ice hockey, a breathtaking, well-admired game, fascinates numerous individuals across the globe with its distinctive fusion of speed, talent, and aggressiveness. Players must demonstrate outstanding mental and physical capabilities[19], as the sport requires not only agility and strength but also critical thinking and team coordination. The intense dynamics of the game make it one of the most entertaining sports to watch or play, having millions of followers around the world[39].
The ongoing technological boom has gradually transformed the world of sports[21], revolutionizing various aspects of training, performance analysis, and fan engagement. However, the adoption of sophisticated performance tracking and analysis tools remains primarily limited to well-funded professional leagues and elite athletes who are a very small portion of the **1,563,749** registered hockey players[26], mainly due to the high costs associated with these services. This limitation results in a considerable gap in the availability of comprehensive and affordable performance analysis tools for amateur, semi-professional, and budget-conscious professional players and teams. The situation is further compounded by the **lack of data** that can assist in data analysis, machine learning, and statistical evaluations.

Mobile/Tablet Applications in Hockey Industry

In recent years, mobile and tablet applications have revolutionized the hockey industry, particularly for players, coaches, and scouts. The NHL has adopted tablet use on the bench[16], allowing real-time analysis and facilitating better in-game decision-making. Coaching and training are enhanced by apps like CoachThem[10] and Hockey Coach Vision[4], which provide resources such as video analysis, practice planning, and drill libraries, streamlining collaboration between coaches and players.

Player development benefits from apps like HockeyTracker[9] and iSnipe[6], helping players track performance, set goals, and analyze strengths and weaknesses, with personalized training programs for skill improvement. Wearable technology integration through apps like Catapult Sports[2] and Firstbeat[3] offers insights into player performance, health, and safety, enabling informed decisions on lineups and training.

Overall, mobile and tablet applications have positively impacted the hockey industry by enhancing coaching and training methods, aiding player development, integrating wearable technology, and supporting officials and scouts, making the sport more efficient and accessible.

1.2 Aims and objectives

The primary goal of the project is to create a tablet application that allows the user to capture the essential actions of a hockey player, giving the possibility to create a database that can further be used to train predictive machine learning models, and generate visually appealing reports, presenting player performance metrics in a clear and informative manner. This application supports the hockey community by helping players identify strengths and weaknesses, assisting coaches in making informed decisions on training and game strategies, and enabling scouts to evaluate potential talent more effectively.

1.3 Project plan and structure

This document contains 6 more chapters, each of them focusing on a specific aspect of the project. The following list offers a concise overview of the content covered in each chapter:

Chapter 2 looks into the current state of data science in ice hockey, presents the GUI design principles of a tablet application, resume the mathematical concepts of SVM models and researches the key aspects of a similar product on the market.

Chapter 3 presents the requirements analysis of the application, defining the user stories and subsequently deducing the key features of the application from them. Moreover, it underlines the validation process of the project.

Chapter 4 builds on the previous chapter, clearly stating the design of each component of the application, and offering a detailed overview of the system's architecture. Alongside this, it includes mock-ups of the most important components to provide a visual representation of their logic.

Chapter 5 demonstrates how the most important features were implemented, along with information regarding the development tools and frameworks used. It also details the testing methods mentioned in Chapter 3.

Chapter 6 discusses the final product, presenting the achieved features, alongside the results of the tests. This chapter also debates the limitations and potential areas of expansion within the project.

Chapter 7 presents a comprehensive synthesis of the project, encapsulating the principal components and providing a concluding assessment of the project's accomplishments.

Chapter 2

Literature Survey

2.1 Data Science in Ice Hockey

2.1.1 Ice Hockey Overview

Ice hockey is a game played on ice between two teams of players who each have a curved stick with which they try to put a puck (= a small, hard disc) into the other team's goal. Each team usually has 20 players, forming 4 distinct lines, with only one line playing on the ice at any given time.

The game requires a combination of skills like speed, physical strength, agility, and team coordination, making it one of the most entertaining sports to watch[39]. With shots surpassing 100 mph and players skating at impressive speeds reaching up to 40 mph[22], the game offers thrilling and fast-paced action.

In recent years, the use of data science in ice hockey has gained traction, as teams employ sophisticated analytics and machine learning methods to secure a competitive edge.

2.1.2 Relevant Metrics

The National Hockey League (NHL) is a professional ice hockey league established in Montréal, Québec, in 1917. Widely recognized as the world's premier hockey league, it currently contains 31 franchises, with 7 in Canada and 24 in the United States[32]. As a significant sports entity in North America, the NHL is known for its exceptional competitive level, drawing a vast audience and producing an impressive annual revenue of $3.3 billion, with $200 million originating from television advertising alone[23].

Apart from its considerable influence in the sports sector, the NHL maintains an exhaustive database that serves as the most comprehensive source of individual and team performance data accessible (NHL STATS). Covering every game played in the league from 1917 to the present, this database rigorously documents an array of individual and team-related metrics. For a more in-depth exploration of this abundant information, data analysts and enthusiasts can access a wider range of supplementary statistics via the NHL's API. While the database provides an extensive selection of both individual and team statistics, this project will particularly focus on individual performance metrics.

Table 2.1 provides an overview of the most commonly utilized stats for characterizing and forecasting player performance[30][31], along with their respective descriptions.

Stat	Description
G	Goals (total number of goals scored)
A	Assists (total number of assists made)
PTS	Points (total points, calculated as the sum of goals and assists)
S	Shots (total number of shots taken)
+/-:	Plus/Minus (the difference between goals scored by the player's team and the opposing team while the player is on the ice)
TOI	Time on Ice (total time spent on the ice during play)
FOW	Faceoff Wins (total number of faceoffs won by the player)
CF	Corsi For (the total shot attempts, including shots on goal, missed shots, and blocked shots, taken by a team while a player is on the ice)
FF	Fenwick For (similar to Corsi For, but excludes blocked shots)
BS	Blocked Shots (total number of shots blocked by the player)
MS	Missed Shots (total number of shots taken by the player that missed the net)
TO	Turnovers (total number of times the player lost possession of the puck to the opposing team)

Table 2.1: Common stats

2.2 Graphical User Interface

Graphical User Interface (GUI) is the user interface that facilitates the interaction between humans and computers through different elements (icons, menus, buttons). In this section, the focus will be on the design of the GUI, especially for tablets, considering the reasons presented in Section 3.2.2.

2.2.1 GUI Design for Tablets

Accessibility

In the paper "Enabling comfortable thumb interaction in tablet computers: a Windows 8 case study"[35] the challenges and importance of designing user interfaces for thumb interaction on tablet computers are discussed. The two-handed grip is often necessary in situations where external support is unavailable, leaving the thumbs as the primary means of interaction. However, due to limited reach, a significant portion of the screen becomes inaccessible. Additionally, there is a need to ensure comfortable thumb interactions to avoid issues like "blackberry thumb," a term coined by Joyce (2005) to describe thumb pain associated with thumb-based keyboards.

In the results section of the paper, Figure 2.1 and Figure 2.2 are presented, displaying heat maps for tablets with corner and side grips, respectively.

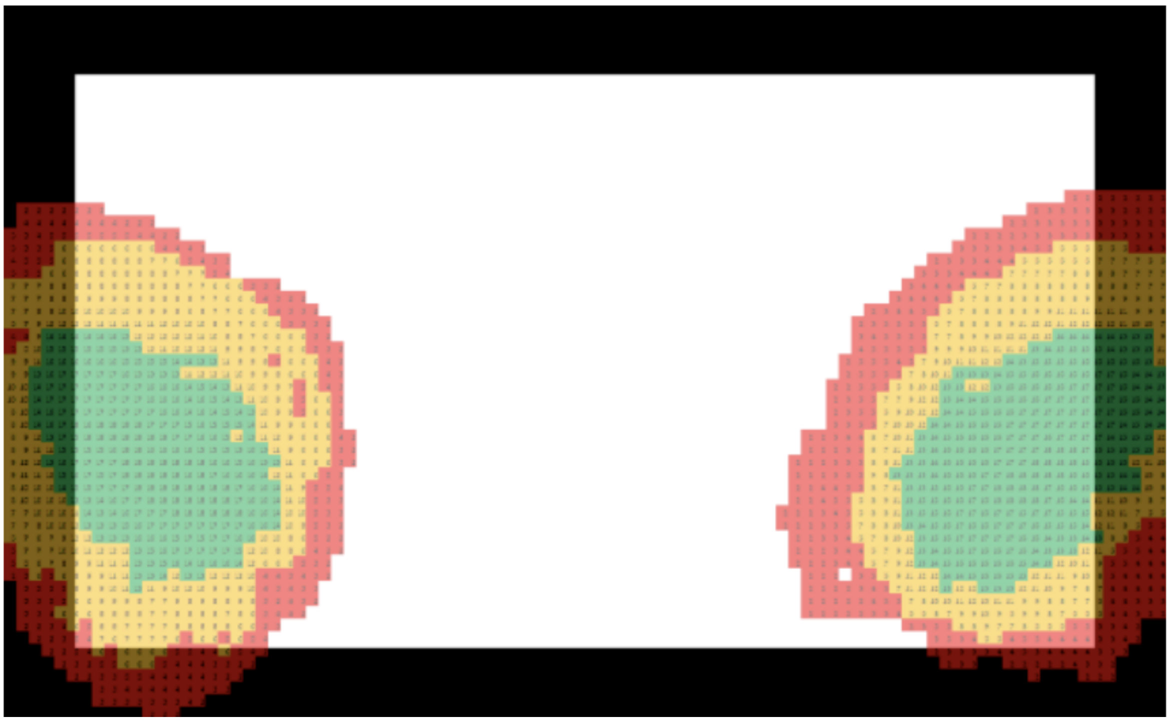

Figure 2.1: Corner Grip heat map that represents levels of comfort and accessibility for thumb interaction (excluding large males): green representing the most comfortable and easily reachable regions, yellow for areas with mixed accessibility, and red for unreachable areas - Source: "Enabling comfortable thumb interaction in tablet computers: a Windows 8 case study"

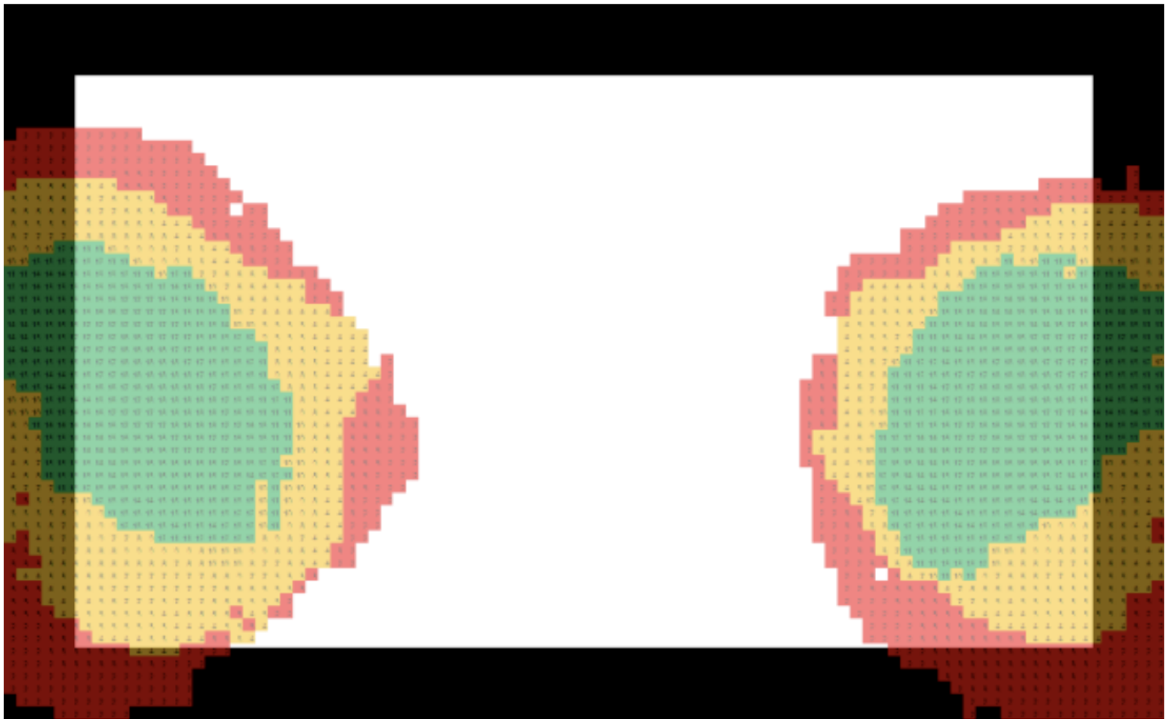

Figure 2.2: Side Grip heat map that represents levels of comfort and accessibility for thumb interaction (excluding large males): green representing the most comfortable and easily reachable regions, yellow for areas with mixed accessibility, and red for unreachable areas - Source: "Enabling comfortable thumb interaction in tablet computers: a Windows 8 case

These heat maps illustrate the areas of an 11.6" tablet screen that are most easily accessible and comfortable for thumb interaction. The reach regions would be similar for other devices that employ a comparable grip style and tablet thickness. The different colors in the mentioned figures represent levels of comfort and accessibility for thumb interaction: green for the most comfortable and easily reachable regions, yellow for areas with mixed accessibility, and red for unreachable areas. These visualizations highlight the need to position GUI elements, especially those that will be most frequently used, in the green regions to ensure an optimal user experience.

Information Architecture and Classification

Based on the book "Information Architecture for the World Wide Web: Designing Large-Scale Web Sites"[34] information architecture is a crucial aspect of GUI design, focusing on the organization, structure, and labelling of content within digital environments. Classification, an essential component of information architecture, involves grouping similar elements to enhance interface usability and navigability. This organization helps users easily locate information and reduces cognitive load, ultimately improving the overall user experience.

The following basic guidelines should be considered for effectively implementing classification[37][12]:

- Group-related elements to improve findability and reduce cognitive load.
- Use clear, concise labels for categories that accurately reflect the grouped content.
- Take user expectations and mental models into account to create intuitive and easily navigable structures.

By considering these guidelines, the design of the application will be accessible, intuitive, and efficient, leading to an enhanced user experience.

Button Size

In the research paper "Touch Screen User Interfaces for Older Adults: Button Size and Spacing"[29] conducted by Zhao Xia Jin, Tom Plocher and Liana Kiff, it was demonstrated that the average time of reaction decreased as the size of the button increased. However, there were no differences when the buttons were bigger than 16.51 mm^2.

In 1996, the EIA (European Interoperability Architecture) suggested that the minimum touch-sensitive button size should be 19.05 mm^2[40].

2.3 Machine Learning

Machine learning is a branch of Computer Science that deals with the development of **learning algorithms** that create models using the training data provided[43]. This process in which the model is tuned is called the "training phase", and there are three main approaches to it, based on the type of feedback that the system is receiving during this process[14]:

- Supervised Learning - Training data is already labelled with the correct output.
- Unsupervised Learning - Training data only contains the input, no output is provided.
- Semi-supervised Leaning - Training data contains both labelled and unlabelled examples.

In the majority of learning algorithms, input samples are represented by converting their features (the various attributes or characteristics of a given sample) into vectors. These vectors can be translated as points in a multidimensional space, with the goal of the model being to find the optimal way to separate or classify the different classes or categories within the data.

To achieve this, the model learns to create a line, plane, or a more complex shape depending on the algorithm and the data's complexity, called decision boundary. This decision boundary is designed to maximize the separation between classes (distance between the two closest samples of different classes), allowing the model to correctly classify new, unseen samples based on their feature vectors.

In practice, this can be used in numerous industries, such as pharmaceutical[20], agriculture[42], finance[15], cybersecurity[11], sport[25], and many more[7].

2.3.1 Machine Learning in Ice Hockey

Machine learning has significantly broadened the scope of analytics applications in hockey[24]. By processing large amounts of data, machine learning algorithms can detect patterns and insights that might be missed through conventional analysis. This has resulted in advanced techniques for player evaluation[30], game strategy optimization[38], and outcome forecasting within the sport[13][23].

Rob Vollman's book, "Hockey Abstract Presents... Stat Shot: The Ultimate Guide to Hockey Analytics"[41], is inspired by Bill James' groundbreaking work in baseball analytics. James published his influential book "The Bill James Historical Baseball Abstract"[28] in 1987, and since then, is widely regarded as the inventor of sabermetrics, the empirical analysis of baseball statistics. Vollman's book highlights the major changes that occurred in the summer of 2014, often referred to as the "Summer of Analytics" in the elite hockey world. During this time, NHL teams began hiring analysts and adopting data-driven decision-making, integrating machine learning techniques into their operations. This shift underscored the growing appreciation for the contributions of analytics and machine learning to hockey, allowing teams to gain a competitive advantage and make well-informed decisions regarding player performance and team strategies.

Moreover, machine learning has been employed in areas beyond team management, such as **betting and scouting**. Betting agencies now utilize machine learning algorithms to more accurately predict consumers' habits[17] and calculate odds[27]. This not only increases the precision of their predictions but also elevates the overall betting experience for users. In scouting, machine learning can streamline the identification of promising talent[30] by analyzing player performance data, making the process more efficient and effective.

In summary, the incorporation of machine learning into ice hockey has revolutionized various aspects of the sport, from team management and strategy to betting and scouting.

In this project, the focus will be on the Supervised Learning approach, with particular explanations given on the Support Vector Machine (SVM) technique since this algorithm will be integrated into my project as proof of concept (see Section 3.4).

2.3.2 Support vector machines (SVM)

This section will delve into the foundational mathematics that underpins Support Vector Machines (SVM) and the use of the kernel function, first described by Vladimir Vapnik and Corina Cortes in 1995[18].

The fundamental principle of this method is to calculate the optimal decision boundary based on a few vectors in the dataset that lay close to the decision hyperplane (the data points that are the most difficult to classify). These vectors are known as Supporting Vectors, giving the method its name. This approach distinguishes SVM from neural networks, where all vectors are considered in the decision-making process, rather than focusing solely on the extreme cases within each class.

$$Let \qquad w_0 \cdot z + b_0 = 0 \tag{2.1}$$

be the formula of the hyperplane, defined by vector w_0, where b_0 is the biased term (if set to 0, the hyperplane passes through the origin).

The **optimal hyperplane**

$$w_0 \cdot x + b_0 = 0 \tag{2.2}$$

is defined by the l-dimensional vector w_0 that satisfies

$$y_i(w \cdot x_i + b) \geq 1, \qquad i = 1, ..., l \; and \; y_i \in \{-1, 1\}, \tag{2.3}$$

where y_i are the labels of the classes and x_i are the components of the vectors in the dataset.

Finding the optimal hyperplane is equivalent to maximising the distance

$$\rho(w, b) = \underset{\{x:y=1\}}{min} \frac{x \cdot w}{|w|} - \underset{\{x:y=-1\}}{max} \frac{x \cdot w}{|w|}, \tag{2.4}$$

subject to the constraints (2.3).

This can be rewritten as

$$\rho(w_0, b_0) = \frac{2}{|w_0|} = \frac{2}{\sqrt{w_0 \cdot w_0}}, \tag{2.5}$$

which is equivalent to minimising $w \cdot w$ under constraints (2.3).

It is possible to rewrite w_0 as a weighted sum of vectors in the data set

$$w_0 = \sum_{i=1}^{l} y_i \alpha_i^0 x_i, \tag{2.6}$$

where $\alpha_i^0 \geq 0$ are the weights of this linear combination. Thus, the problem of finding w_0 can be rewritten in terms of finding Λ_0^T defined as

$$\Lambda_0^T = (\alpha_1^0, ..., \alpha_l^0). \tag{2.7}$$

It can be shown (see Appendix A.1 from Vapnik and Cortes), that minimising (2.5) is equivalent to maximising $W(\Lambda)$, defined as

$$W(\Lambda) = \Lambda^T 1 - \frac{1}{2} \Lambda^T D \Lambda \tag{2.8}$$

with respect to $\Lambda^T = (\alpha_1, ..., \alpha_l)$, under the constraints

$$\Lambda \geq 0, \tag{2.9}$$

and

$$\Lambda^T Y = 0, \tag{2.10}$$

where D is a symmetric $l \times l$ matrix with elements

$$D_{ij} = y_i y_j x_i \cdot x_j, \qquad i, j = 1, ..., l. \tag{2.11}$$

This method offers a significant advantage because the optimization problem is defined in terms of the inner product of vectors. This property is beneficial as it enables the use of a kernel function K to compute the inner products between vectors in the non-linearly mapped space $\phi(x)$, as one would do when the vectors were not linearly separable in the original space. In this case, the matrix D becomes simply

$$D_{ij} = y_i y_j K(x_i, x_j), \qquad i, j = 1, ..., l. \tag{2.12}$$

This "kernel trick" avoids the explicit calculation of $\phi(x_i) \cdot \phi(x_j)$ that can be computationally demanding and replaces it with $K(x_i, x_j)$.

The kernel function K must satisfy the condition that it represents a convolution of an inner product in the non-linearly mapped space. Common choices of kernel functions include Radial Basis Function (RBF), Polynomial Function, and Sigmoid Function[36].

2.4 Similar services

Upon conducting a literature review and market analysis to identify a service that is similar to the intended application being developed, several platforms were discovered. Among these, the most analogous platform is inStat, a company that was acquired by Hudl[8], a sports performance analysis company. InStat offers comprehensive reports on athlete performance across various sports, making it a vital resource for coaches, teams, and individual athletes to break down their performance and maintain a competitive edge. Two other noteworthy services include Catapult Sports[2], and Sportlogiq[5], each providing valuable analytics and insights within the sports performance analysis domain. Nonetheless, the **report service feature** of InStat exhibits the most similarities to the proposed application, making it the most pertinent platform for further scrutiny and comparison. Investigating the current market environment underlines the potential worth and practicality of the report service aspect of the software tool being developed, accentuating the necessity for ongoing examination and innovation within this domain.

2.4.1 User Experience

Ease of use

InStat's sports analytics service offers impressive ease of use, catering to coaches, analysts, and players. The platform is accessible on various devices such as smartphones, tablets, and laptops, ensuring that users can review crucial data and insights from virtually any location. This compatibility across devices is ideal for professionals who are constantly playing away games.

The intuitive design and layout of the platform streamline the user experience, making it easy to navigate and understand. InStat has made significant efforts to ensure that their sports analytics service is user-friendly for individuals with varying levels of technical expertise. Clear and concise tutorials allow users to quickly learn the system and begin analyzing data, while customer support is available for users who may encounter difficulties or have questions about the platform.

Output

The output provided by InStat's sports analytics service is both comprehensive and customizable. Users have the ability to access a comprehensive variety of metrics, graphs, and reports, adapted to their particular needs. The platform's advanced analytical capabilities allow an in-depth exploration of the data, uncovering patterns and insights that may contribute significantly to making informed strategic decisions.

Furthermore, InStat's integration of machine learning techniques allows for the generation of predictive models, offering valuable foresight into future game outcomes and player performance. Figure 2.3 showcases a sample report, highlighting the depth and variety of information available to users in a visual representation, helping them better understand the types of data and insights they can expect from the platform.

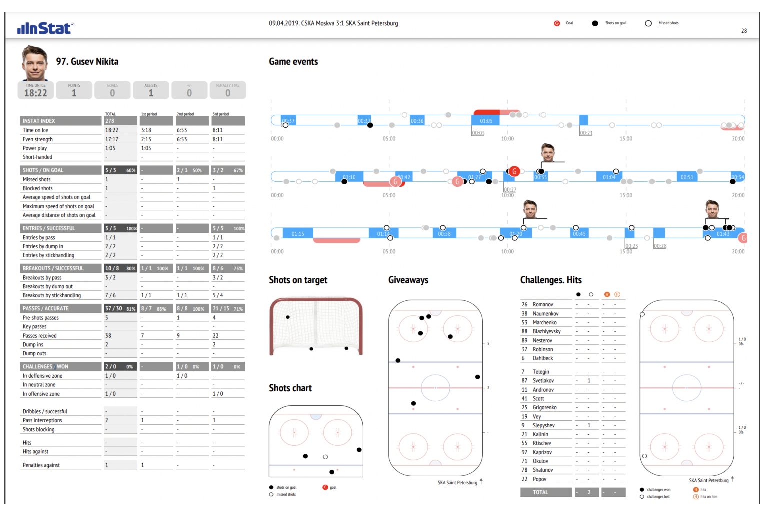

Figure 2.3: InStat Report 2022 - Source: InStat Website 2022

This combination of in-depth data analysis and predictive capabilities makes InStat's sports analytics service a powerful tool for sports professionals seeking a competitive edge.

2.4.2 Criticism

Though InStat has numerous advantages, there are certain aspects that could be improved. As illustrated in Figure 2.4, which displays a screenshot of their price plans for the report feature, the pricing structure might be restrictive for smaller organizations or individual users with limited financial resources.

Figure 2.4: InStat Report Pricing - Source: InStat Website 2022

With the "FULL" version of the service priced at 50 euros and the "PRO" version of their report feature costing 100 euros per game, as of November 2022, a service offering lower quality but at a reduced cost or even for free could be more suitable for certain players, including those who are amateur, semi-pro, or professional athletes with budget constraints. It is worth noting that the pricing has undergone updates since InStat's transition to Hudl, and currently, a price list is not available on their website.

Another limitation is the platform's reliance on video recordings to perform its analysis. In cases where games are **not video recorded**, InStat is unable to provide any analysis or insights. This issue could be addressed by incorporating alternative data sources or allowing users to input data manually, expanding the platform's applicability to a broader range of games and situations.

Additionally, while the platform provides an extensive range of metrics and analyses, some users may find the sheer volume of data overwhelming. Incorporating additional customization options could help users more efficiently navigate and interpret the information available to them.

In conclusion, InStat's sports analytics service delivers a comprehensive, portable, and accessible solution for sports professionals seeking data-driven insights. Although there are areas for improvement, such as addressing the dependency on video recordings and pricing strategies, the platform's overall user experience, advanced analytics capabilities, and machine learning integration make it a valuable tool for those looking to gain a competitive edge in the world of sports analytics.

Chapter 3

Requirements and analysis

This section establishes a collection of requirements for the individual performance analysis application.
The essential features for this application's implementation have been inspired by specific criteria outlined in Section 2.4.2. The project aims to fill the gap in the hockey community by providing players outside of major leagues with access to detailed statistics and performance reports while generating data samples suitable for machine learning purposes. To do so, it is required to create an Android tablet application that allows the users to input the action of a player and generate a report that presents his/her game performance, along with a database containing all the captured stats.
Figure 3.1 presents the user's stories based on which the features of the application will be deduced.

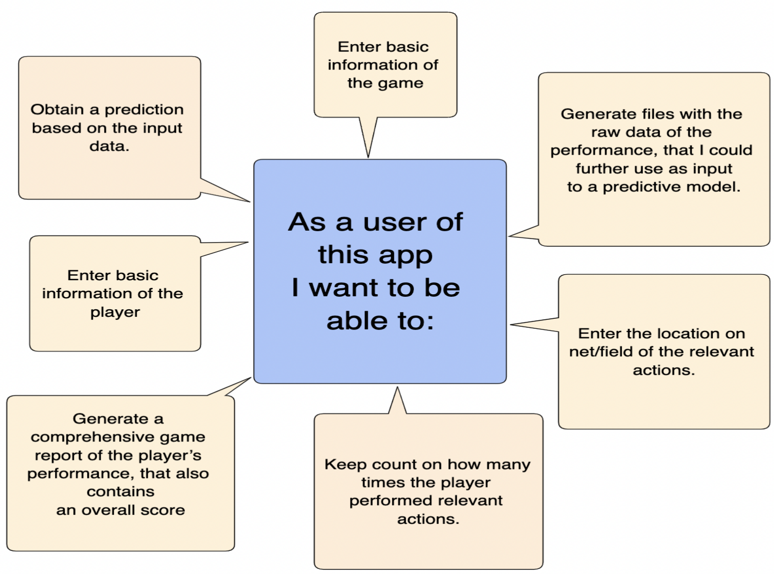

Figure 3.1: User's stories

3.1 Application's Features

From the user's stories presented in Figure 3.1, features presented in Table 3.1 could be extracted and **all of them** are expected to be implemented:

Feature	Description	Priority
Player Information Input	Allow users to enter basic information of the player such as name and position.	MUST
Game Information Input	Enable users to enter essential game information, including game date, final score, and name of the teams.	MUST
Action Tracking	Provide an intuitive interface for users to easily track and count the occurrence of specific actions performed by the player during the game, such as goals scored, assists, hits, possessions and more.	MUST
Location Mapping	Allow users to input the location of each relevant action on a digital representation of the ice rink and net, helping to visualize patterns and areas of strength or improvement.	SHOULD
Comprehensive Game Report Generation	Automatically generate a detailed game report that includes a breakdown of the player's performance, highlights specific actions, and calculates an overall score based on predefined criteria.	MUST
Performance Prediction	Utilize machine learning with the provided information and predict a certain event (e.g., scoring a point in the next game)	SHOULD
Raw Data Export	Offer an easy-to-use export feature that allows users to download raw data of the player's performance in CSV format for further analysis and integration with predictive models or other analytics tools.	MUST

Table 3.1: Application's features

Considering the features mentioned in Table 3.1, the following aspects of the application need to be discussed.

3.2 Data Capturing

3.2.1 Recorded Stats

Apart from the basic information about the player and the game (Name, Position on ice, Date, Location, Team for and Team against), the individual stats that will be recorded are: Goals, Assists, Shots, Time on ice, Number of Shots, Possessions, Faceoffs, Hits, Fights, and Penalties. These stats are considered to be the most relevant when assessing a player's performance, based on the literature review in Section 2.1.2 and I believe that would properly encapsulate his/her overall contribution to the game.

Each of the mentioned metrics will be split into sub-stats (based on the period number) to present a more detailed view of the player's individual performance throughout the game.

For the complete list of these period-based stats, refer to Section 4.3.2.

For the predictive component of the project, additional stats related to the player's past performance will be required, these metrics and their relevance towards the player's form will be presented in Section 4.2.2.

By gathering these statistics, the application will be better at assessing the player's performance and the data samples will be more detailed, enabling users to make informed decisions based on the presented information and the machine learning algorithms to train more precise models.

3.2.2 GUI Design

To ensure that the application is user-friendly and efficient, it is essential to adhere to the design principles outlined in Section 2.2.1. These principles include minimizing the number of clicks required for user input, organizing buttons logically, ensuring easy access, and optimizing button size.

- **Minimizing Clicks**: The application should allow users to input player actions in 4 or fewer clicks. This streamlined approach will enhance the user experience by reducing the time and effort spent inputting data.
- **Logical Organization**: Buttons should be grouped by the type of action to facilitate quick and intuitive data capturing. This organization will allow users to easily locate the desired action and minimize confusion.
- **Button Placement**: Actions that occur more frequently, such as changes of possessions and shots, should be located near the edges of the screen for easy access. This strategic positioning will help to improve the overall efficiency of the application. Conversely, actions that are seldom used can be located in the middle of the screen, as they will not be accessed as frequently.
- **Button Size**: To minimize the risk of misclicking and enhance user experience, the ideal button size should be approximately 3 cm^2, which is considerably larger than the values mentioned in Section 2.2.1 (19 mm^2). By employing larger buttons, users can reliably select their desired actions and enjoy a more aesthetically pleasing interface.

By incorporating these design principles, the application will offer a seamless user experience that facilitates quick and accurate input of player actions. This will ultimately contribute to a more efficient and enjoyable application for tracking and recording hockey data.

Given that the application is designed to enable users to input player information while watching a game, either live or through a replay, it should be **optimized for use on tablets** rather than smartphones or desktop computers. Tablets offer a larger screen size, which is especially beneficial when the interface is primarily comprised of buttons. This larger display ensures that button presses are optimized, making it easier for users to input data more accurately compared to smartphones.

In addition to the screen size advantage, tablets are also lightweight and portable, which is ideal for users who need to enter data while attending live events or watching replays remotely. Desktop computers, conversely, lack portability, making them a less suitable choice for this purpose.

3.3 Output

In regard to the output, the application should be useful for players, scouts and coaches by giving a better insight into a player's performance (by describing past performance based on a detailed report), and for users interested in establishing databases of information. Therefore, the output will consist of two types of files, PDF and CSV. The requirements of each will be described below.

3.3.1 PDF Report

Given that inSTAT, a highly recognizable game analysis service offers an extensive report encompassing key aspects of a player's game performance, such as stats related to shots, passes, zone entries, breakouts, and challenges, as well as diagrams illustrating the location of various events (see Section 4.3.1), the report produced by our system should also incorporate some of these components.

To guarantee that the report is efficient and informative, it should exhibit the following attributes:

- Well-organized: The report should be structured into separate sections, with each segment representing a distinct aspect of the game. This arrangement will enable effortless understanding for the reader.

- Comprehensive: The report should include all relevant metrics, delivering a thorough overview of the player's performance. This thoroughness will aid users in making informed decisions based on the recorded data.

- Insightful and comprehensible: The report should be organized in a manner that allows even a non-expert reader to easily grasp the presented stats and deduce conclusions about the player's performance. To accomplish this, the report should employ succinct explanations and representative visualizations.

By integrating these components and characteristics, the report will serve as a valuable resource for users to evaluate a player's performance in an approachable manner, ultimately augmenting the overall efficacy of the game analysis process.

3.3.2 CSV Files

Considering that one of the main goals of the project is to create an application that captures data that can further be used as input for training predictive models, the statistics should be saved in an accessible manner. Therefore, the CSV component will be composed of two files that will save all the captured data. To increase the ease of use, one of the two files will contain the stats, while the other will only store the type of each event, its location and the timestamp of the game when it occurred so users can chronologically understand the events. In this way, a future database could be easily compiled from these files.

3.4 Machine Learning Model

The objective is to create a simple yet effective machine learning model that demonstrates the usefulness of the statistics captured by the application. The model should be designed as a straightforward Support Vector Machine (SVM) to anticipate whether a specific event, like a player scoring a point (goal or assist), will occur in their forthcoming match. The selection of SVM as the type of algorithm is attributed to its **extensive usage and simplicity** of implementation, as mentioned in Section 2.3.2.

By showcasing the model's capacity to train and do relatively accurate predictions (see Section 6.1.3) using the same type of datasets gathered by the application, it will confirm the worth of the data obtained, therefore demonstrating that one of the project's main goals was achieved.

Concentrating on forecasting whether a player will score a point in their next game, the model will not only emphasize the significance of the collected data but also demonstrate its practical applications for teams and coaching staff.

Employing a simple SVM model allows for an efficient and easily interpretable approach to prediction. Instead of forecasting probabilities, this model will provide a binary outcome in the PDF report, stating whether the player will score a point or not. This predictive capability can inform decision-making processes, such as game strategies and player selections, ultimately enhancing overall team performance.

By demonstrating the effectiveness of a common and user-friendly model like SVM, it will become evident that the data captured by the application holds significant value and can be utilized for more complex usage in the future.

3.5 Analysis

The analysis phase is crucial for the success of the project and is divided into two vital components: the application and the model. This segmentation allows for a comprehensive evaluation of both aspects, ensuring that they perform effectively and achieve the desired outcomes. By testing each component, potential issues can be identified and addressed, therefore contributing to the overall success and effectiveness of the project.

3.5.1 Application

To be able to consider that the goals of the project have been achieved, the reliability and effectiveness of the Android application must be validated. Consequently, a systematic testing process will be carried out (see Section 5.2 for the implementation of the testing methods and Section 6.1 for the results). The next elements of the project will be tested:

Interface

The first stage of testing will focus on evaluating the application's interface to verify that it is user-friendly, responsive, and functional across the targeted Android devices and screen resolutions.

Functionality

The second stage of testing will target the core functionality of the application, which involves recording the game events for an ice hockey player.

3.5.2 Machine Learning Model

To validate the machine learning model, a comprehensive evaluation will be conducted using various performance metrics - accuracy, recall, precision, and F1 score (these metrics were chosen because they offer a complete overview of the performance, and they are the most popular ones[33]). Alongside these metrics, a confusion matrix will be created to provide a visual representation of the chosen model's performance (see Section 6.1.3). The test data used for this evaluation will be randomly selected from the initial dataset.

This assessment will determine the model's effectiveness in predicting specific events related to the targeted player's performance, such as scoring a point (goal or assist) in their next game. By doing so, it will prove the potential practical applications of the captured data for teams and coaching staff, ultimately contributing to informed decision-making processes and improving overall team performance.

Chapter 4

Design

4.1 GUI Design

In accordance with Section 3.1, for efficiency and simplicity, Figure 4.1 presents the proposed application flow:

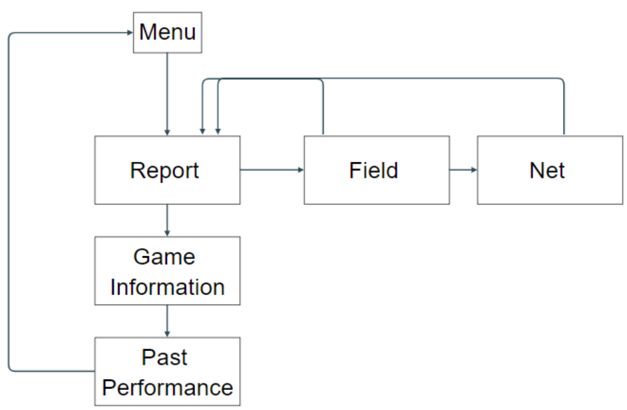

Figure 4.1: Application flow

As stated in Section 3.2.2, the targeted devices are tablets, therefore, the design and buttons' location will be made to suit the Android Studio's "Tablet" reference device (1280 x 800 dp, xxhdpi) which states "This reference device uses the EXPANDED width size class, which represents **97% of Android tablets** in landscape orientation".

Activities Mock Design

The design and organisation of the buttons in Figure 4.2 will be proposed for the Report activity, while the final result of the implementation can be seen in Figure 5.5.

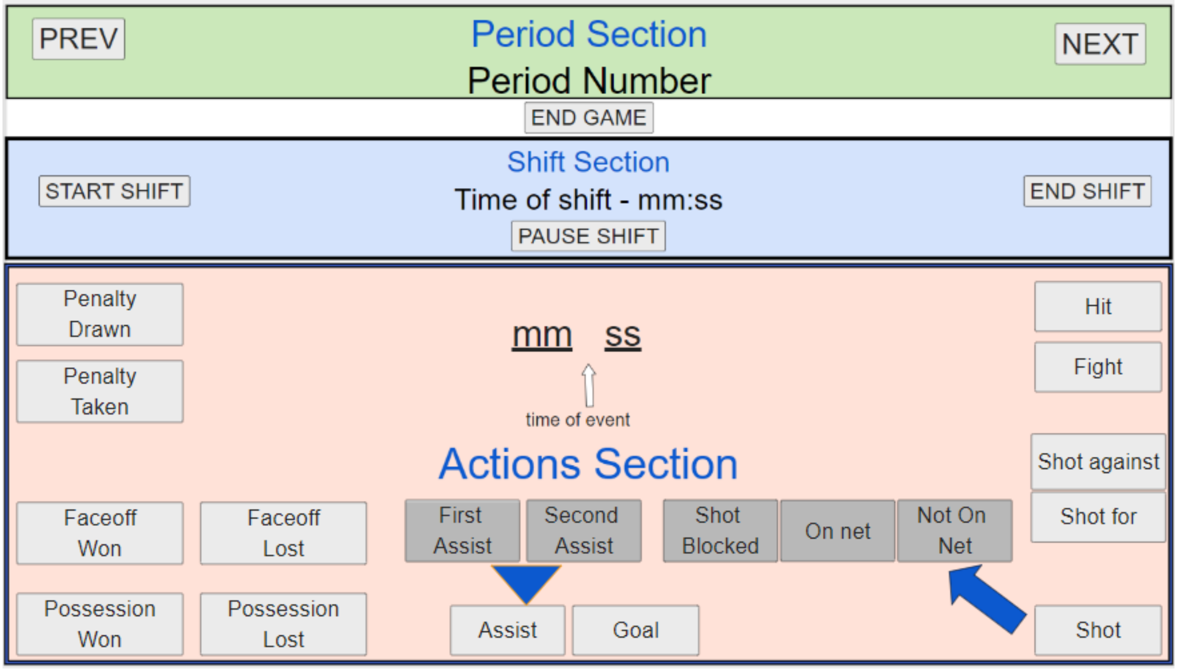

Figure 4.2: Report Activity Mock

The screen is divided into three prominent sections, each representing different aspects of the game: periods, shifts, and actions during shifts. The decision to position buttons for **more frequent actions**, such as possessions and shots, along the edge of the screen was made to improve accessibility and ease of use, complying with the recommendations in Section 2.2.1. The sections are ordered in a horizontal hierarchy, also sorted by their frequency and relevance. By arranging the buttons in this manner, users can easily access and interact with the interface, resulting in a more efficient and user-friendly experience that aligns with the natural flow of the game.

The Field and Net activities will be simple pictures of an ice rink and a hockey net, on which the user will tap to indicate the location of the events so there is no need to implement a mock design for them.

Information and Past Performance activities will be simple forms that will capture general data of the player and game (name, teams, position, differential, score, location and date) and past performance, respectively. Figure 4.3 represents a mock of the Game Information activity and Figure 4.4 the mock for Past Performance activity.

Figure 4.3: Game Information Activity Mock

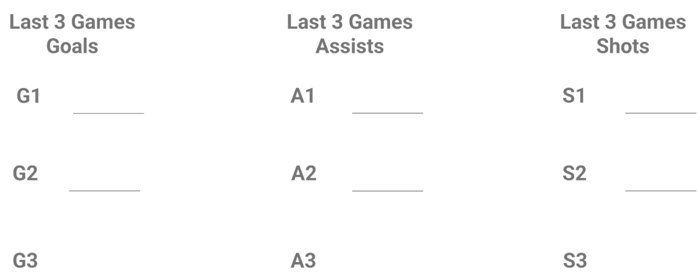

Figure 4.4: Past Performance Activity Mock

The color schemes will be automatically generated by Android Studio in order to guarantee the best contrast and font legibility.

The size of each button will be set according to the text that indicates its functionality, but not less than 19.05 mm^2 (as recommended in Section 2.2.1).

4.2 Machine Learning Model

4.2.1 Training Data

The training data will be scraped from different websites that display the complete game-logs of the NHL players. Before training the models, the parameters will be scaled and split into two categories: training and testing. Considering how difficult it is to predict individual or team performances in hockey, the class that will be predicted will indicate whether the player will be able to obtain a point (goal or assist) in his next game, as stated in Section 3.4.

4.2.2 Feature Selection

The selection of appropriate features is crucial in building an accurate support vector machine (SVM) model. The chosen features incorporate conventional statistics, focusing on various aspects of a player's performance in recent games. I have selected 17 features that are representative for reflecting the player's recent form and give the best insights into whether he/she will be able to get a point in the next game. Table 4.1 includes each feature that is relevant to the player's scoring, play-making ability and role within the team, or team's performance.

Stat	Description
G	Goals (total number of goals scored)
A	Assists (total number of assists made)
PTS	Points (total points, calculated as the sum of goals and assists)
S	Shots (total number of shots taken)
+/-:	Plus/Minus (the difference between goals scored by the player's team and the opposing team while the player is on the ice)
TOI	Time on Ice (total time spent on the ice during play)
GF	Goals For (total number of goals by the player's team)
GA	Goals Against (total number of goals by the opposite team)
G1	Goals in last game
G2	Goals in second to last game
G3	Goals in third to last game
A1	Assists in last game
A2	Assists in second to last game
A3	Assists in third last game
S1	Shots in the last game
S2	Shots in second to last game
S3	Shots in third to last game

Table 4.1: Model's Features

4.2.3 Hyperparameters

The final configuration of the hyperparameters for the SVM model will be determined through an optimization process, which involves testing various combinations to achieve the best model performance (refer to Table 6.2). The hyperparameters under consideration for tuning are:

- Kernel
- Regularization Parameter (C)
- Gamma
- Class Weight

4.3 Output

4.3.1 PDF Report

As mentioned in Section 3.3.1 the PDF report will be inspired from the InStat's one (Figure 2.3). It will contain eight different sections, each presenting different aspects of the game. Figure 4.5 is a mock design for the report, that presents each section.

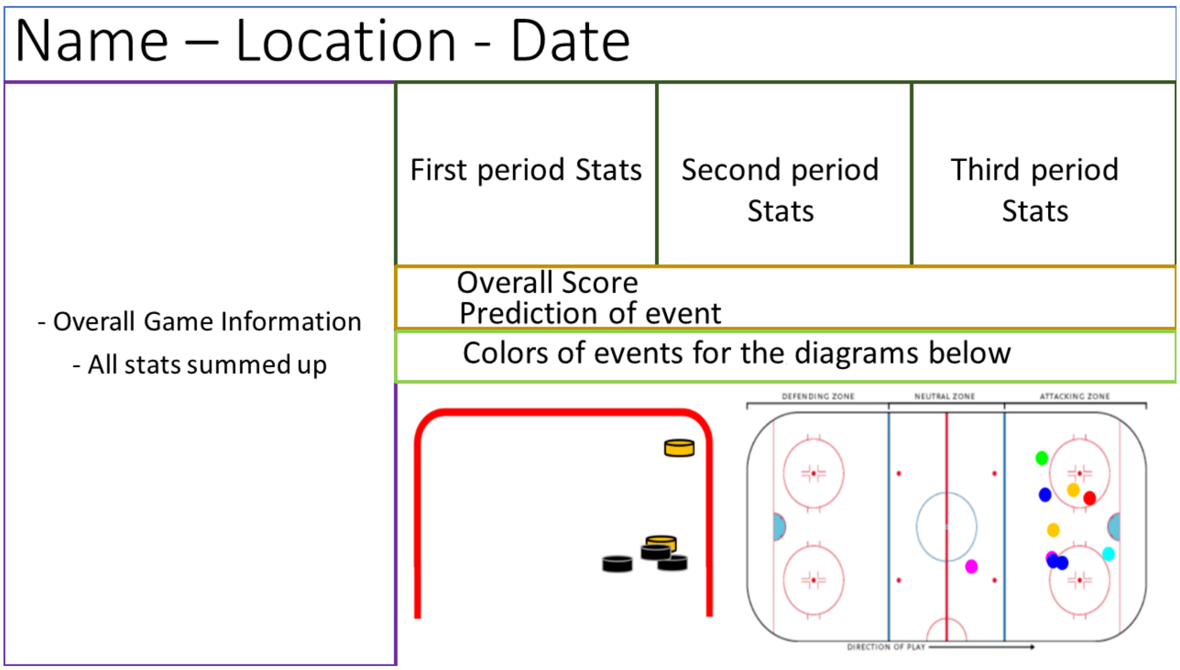

Figure 4.5: Report Mock Design

To understand the exact information that will be contained in each section refer to Figure 5.6.

4.3.2 CSV Files

In total, the first CSV file that is being generated will contain 70 different parameters (columns). Considering Section 3.2.1 the next parameters will be recorded:

General information of the player:

- Name, Position.

General information of the game:

- Date, Team For, Team Against, Location.

For each period: 1, 2, 3 and OT:

- Goals, Assists (A1 + A2), Shots on goal, Time on ice, Shift average, Possessions won, Possessions lost, Number of shifts.

Parameters for Player Game Score metric:

- Goals, A1, A2, Shots on goal, Blocked shots, Penalties drawn, Penalties taken, Faceoffs won, Faceoffs lost, Corsi for, Corsi against, Goals for, Goals against.

Other individual performance stats:

- Shots not on goal, Total time on ice, Total possessions won, Total possessions lost, Total number of shifts, Total shift average, Hits, Fights, +/- Differential.

For each of the last three games:

- Goals, Assists, Shots.

Model prediction:

- Predicted class

The second CSV file will contain information about each action that is location related and will have 5 columns that are self-explanatory: Location Type (Net or Field), Event Type, X Coord, Y Coord, Time.

Chapter 5

Implementation and testing

In this chapter, the main components of implementing the application will be presented, along with the testing methods. The results will be presented in Chapter 6.

5.1 Implementation

Features

As stated in Section 3.1, **all of the features** included in Table 3.1 were implemented.

Development Environment and Tools

The Android app was developed using Android Studio, which is "the official Integrated Development Environment (IDE) for Android app development"[1]. Java was chosen as the programming language due to its compatibility with Android development and its extensive support for building complex applications. Additional libraries and dependencies, such as iTextPDF, Aspose.Cells and Apache POI for generating the PDF and CSV files were integrated into the project as needed. The model was trained in Python.

GUI Design

The project consists of 6 activities and 3 helper classes that ensure the proper functionality of the activities. The first activity will be discussed separately since it represents the most important aspect of the application:

Report Activity activity which is the core of the application consists of a user interface with buttons, labels (for the period number and shift length) and "EditTexts" elements that were conditioned to accept only numbers (for the time of events), designed for easy data input. The layout and design of the buttons were established in Section 4.1 (Figure 4.2), ensuring a user-friendly and intuitive experience. Each button in this activity is equipped with an action listener that serves one or more of three purposes: modifying the appropriate data fields to record relevant statistics, initiating other activities that allow users to input more detailed information about specific actions, such as locations on the ice rink or enabling/disabling other buttons for subsequent actions. Each of the figures below exemplifies one of the three purposes mentioned earlier, with Figure 5.1 corresponding to the first purpose mentioned, Figure 5.2 representing the second purpose, and Figure 5.3 illustrating the third one.

```
raduniculae123
onNetBtn.setOnClickListener(new View.OnClickListener() {
    raduniculae123
    @SuppressLint("SetTextI18n")
    @Override
    public void onClick(View view) {
        sog++;
        if (period == 1) {
            firstPeriod[2]++;
        } else if (period == 2) {
            secondPeriod[2]++;
        } else if (period == 3) {
            thirdPeriod[2]++;
        } else if (period == 4) {
            otPeriod[2]++;
        }
```

Figure 5.1: Action Listener that modifies data

```
raduniculae123 *
endGameBtn.setOnClickListener(new View.OnClickListener() {
    raduniculae123 *
    @Override
    public void onClick(View view) {
        Intent intent = new Intent( packageContext: ReportActivity.this, GameInfoActivity.class);
        startActivityForResult(intent, THIRD_ACTIVITY_REQUEST_CODE);
    }
});
```

Figure 5.2: Action Listener that initiates activity

```java
/*------------------------------------ASSISTS-----------
raduniculae123
assistBtn.setOnClickListener(new View.OnClickListener() {
    raduniculae123
    @SuppressLint("SetTextI18n")
    @Override
    public void onClick(View view) {
        firstAssistBtn.setVisibility(View.VISIBLE);
        secondAssistBtn.setVisibility(View.VISIBLE);
        penaltyDrawnBtn.setEnabled(false);
        penaltyTakenBtn.setEnabled(false);
        faceOffWonBtn.setEnabled(false);
        faceOffLostBtn.setEnabled(false);
        shotBtn.setEnabled(false);
        shotForBtn.setEnabled(false);
        shotAgainstBtn.setEnabled(false);
        possessionLostBtn.setEnabled(false);
        possessionWonBtn.setEnabled(false);
        assistBtn.setEnabled(false);
        hitBtn.setEnabled(false);
        fightBtn.setEnabled(false);
        goalBtn.setEnabled(false);
```

Figure 5.3: Action Listener that disables buttons

The method that handles the data produced from other activities is ”onActivityResult”, which processes the provided information based on the requestCode, resultCode, and Intent data, updating the app’s state accordingly. Those activities are being called by the ”startActivityForResult” method, which allows the ”ReportActivity” to run in the background (the static variables are not re-instantiated). Figure 5.4 showcases a segment of the ”onActivityResult” method, which, depending on the requestCode and resultCode arguments, launches a new activity (in this instance, it launches the ”NetActivity” and passes a parameter indicating whether the shot resulted in a goal or not).

```
raduniculae123
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == FIRST_ACTIVITY_REQUEST_CODE) {
        if (resultCode == RESULT_OK) {
            String eventType = data.getStringExtra( name: "eventType");
            float eventX = Float.parseFloat(data.getStringExtra( name: "eventX"));
            float eventY = Float.parseFloat(data.getStringExtra( name: "eventY"));
            // GOAL
            if (eventType.equals("0")) {
                fieldEvents.add(new Triplet(eventType, eventX, eventY, minutes, seconds));
                Intent myIntent = new Intent( packageContext: ReportActivity.this, NetActivity.class);
                myIntent.putExtra( name: "goal",  value: "0");
                startActivityForResult(myIntent, REQUEST_CODE_2);
                // SHOT ON NET
            } else if (eventType.equals("1")) {
                fieldEvents.add(new Triplet(eventType, eventX, eventY, minutes, seconds));
                Intent myIntent = new Intent( packageContext: ReportActivity.this, NetActivity.class);
                myIntent.putExtra( name: "goal",  value: "1");
                startActivityForResult(myIntent, REQUEST_CODE_2);
            }
```

Figure 5.4: Segment of "onActivityResult" method

Also, this class contains the methods in charge of the creation of the PDF report and the CSV, since all the data fields of the performance are captured inside it. This activity is using class "ModelHelper" to integrate the model into the interface.
The rest of the activities and their helper classes (where needed) will be described in Table 5.1.

Name	**Helper Class**	**Description**
HockeyFieldActivity	HockeyFieldView	Allows the user to record the location on the rink for the relevant actions
NetActivity	NetView	Allows the user to record the position on net of the shots taken
GameInfoActivity	-	Allows the user to insert general data of the game and player
PastPerformance	-	Allows the user to insert data regarding the last games (only used as input data for the predictive model)
MenuActivity	-	Entry point to the application and indicates the user that the report has been generated.

Table 5.1: Application's Activities

In this document, a screenshot displaying the final implementation of the design for each activity can be found in the APPENDIX A section. The Report activity is specifically highlighted in Figure 5.5, while the remaining activities are documented with corresponding figures in the Appendix.

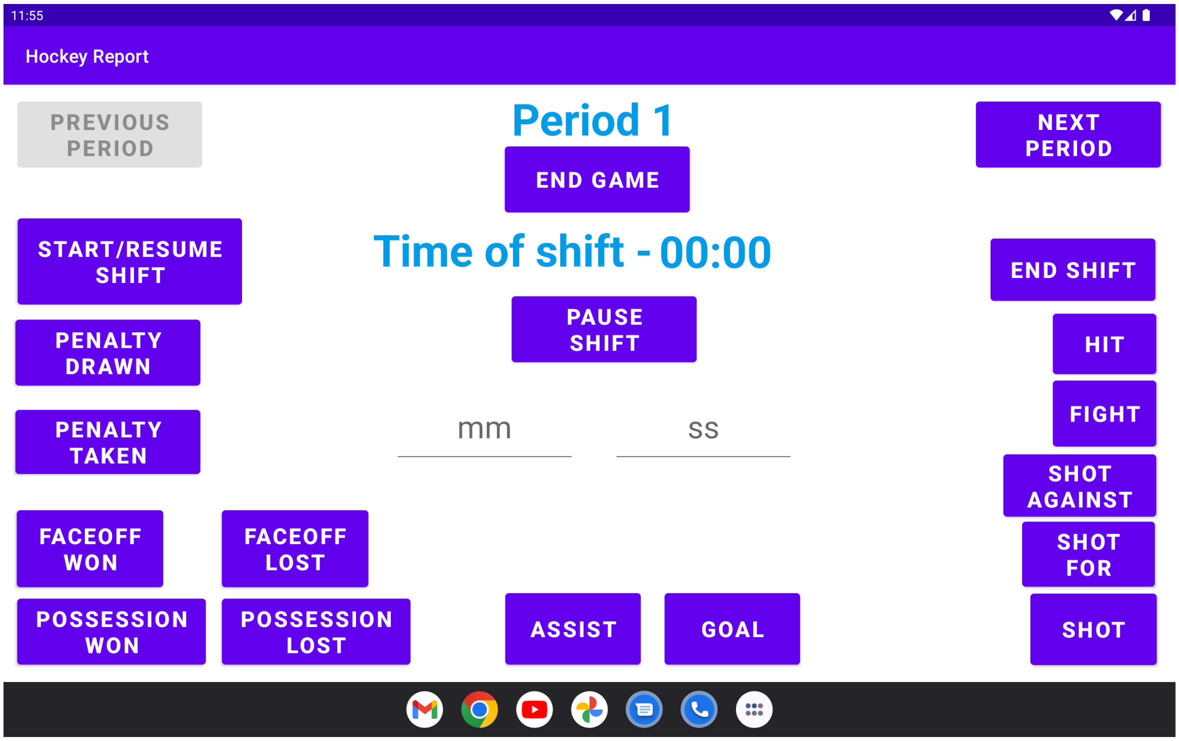

Figure 5.5: Report Activity

It can be observed that the actual implementation of the interface is in accordance with the GUI recommendations and design principles highlighted in Section 2.2.1.

PDF Report

The actual implementation of the report was carried out based on the mock design showcased in Figure 4.5, as illustrated in Figure 5.6.

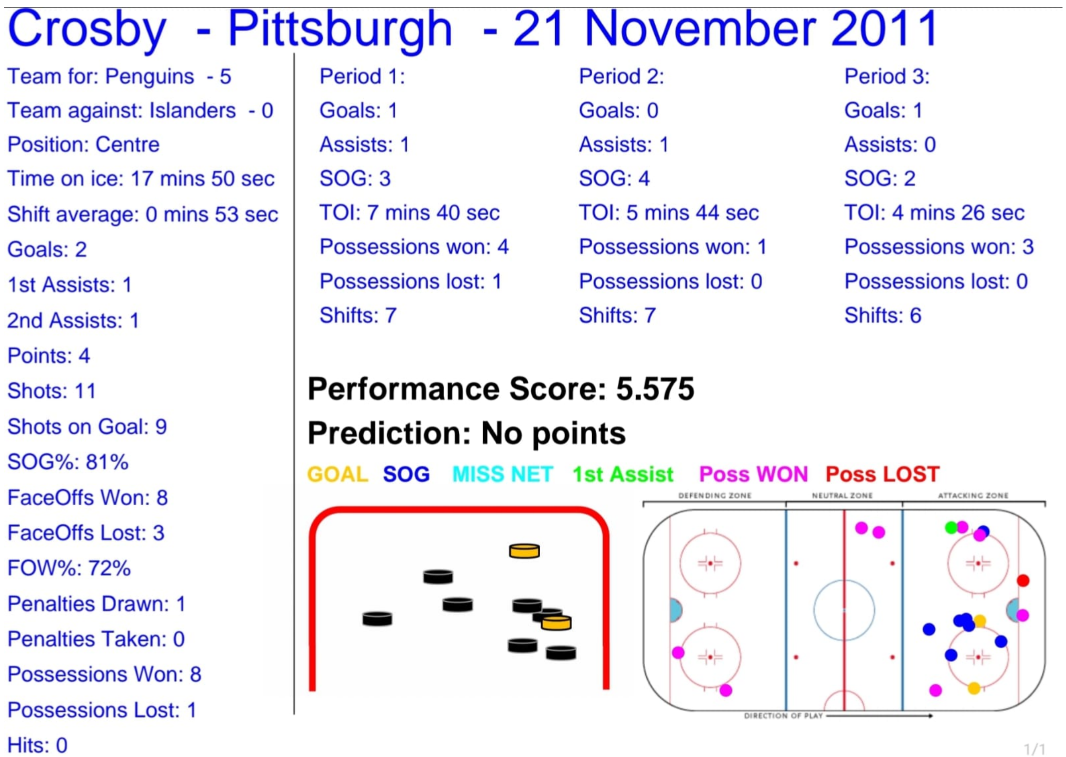

Figure 5.6: Game Report

The final report includes 45 statistics and one prediction, providing a comprehensive overview of the game. The data within this report was generated from a real NHL game played by Sidney Crosby, showcasing the practicality of the app. It demonstrates how anyone with basic knowledge can effectively understand a player's performance by simply reviewing the report. This highlights the app's potential value and its ability to provide users with insights into player performance.

Upon completion, the PDF report is saved in the "Downloads" directory of the user's tablet. The PDF report, along with the two CSV files, can be identified by their unique names, which consist of the player's name and the date of the played game, making it easier for the user to distinguish different reports. This naming convention assumes that a player will not play more than one game within a 24-hour time frame.

Figure 5.7 provides a screenshot that indicates the location in the device's memory and the name of the files, proving the ease of further use, like exporting.

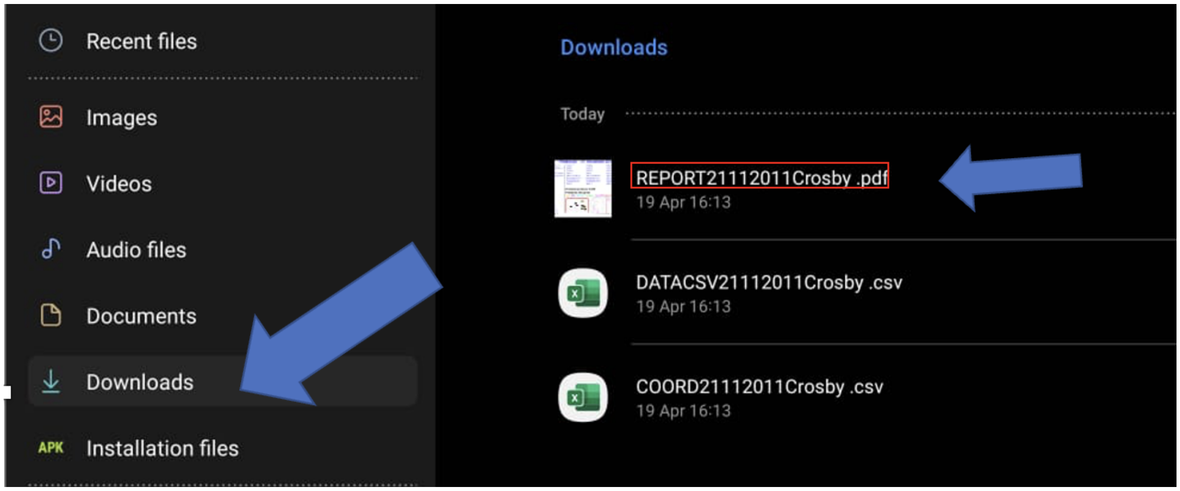

Figure 5.7: Location in memory

CSV files

The first CSV file contains all 70 parameters (columns), explained in Section 4.3.2, Figure 5.8 presents a snippet of this file (with only 14 parameters), compiled for the simulation represented in Figure 5.6.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Date	Team For	Team Against	Position	Location	Goals For	Goals Against	p1g	p1a	p1sog	p1toi	p1shfavg	p1posw
2	Crosby	21/11/2011	Penguins	Islanders	Centre	Pittsburgh	5	0	1	1	3	460	65	4

Figure 5.8: Snippet of first CSV file

The details of each location-related action will be stored in the **second CSV file**. The values of each column are self-explanatory, except for the "Event Type" column, where each integer value represents a specific action, as follows:

- For Net Activity: 0 - Goal, 1 - Shot on Net
- For Field Activity: 0 - Goal, 1 - Shot on Net, 2 - Shot NOT on Net, 3 - First Assist, 4 - Possession Won, 5 - Possession Lost

Additionally, a screenshot showcasing the CSV file rendered in Microsoft Excel is showcased in Figure 5.9

Field/Net	Event Type	X Coord	Y Coord	Time
Net	0	2161	1067	01:20
Net	0	2031	457	30:40:00
Net	1	2111	340	30:40:00
Net	1	1992	407	45:40:00
Net	1	1716	333	45:40:00
Field	0	776.7273	196.72728	01:20
Field	3	700	129.09091	02:20
Field	5	816.36365	214.18182	02:20
Field	4	724.36365	341.45456	25:30:00
Field	4	528.7273	359.27274	25:40:00
Field	0	728	281.81818	30:40:00
Field	2	862.1818	332.36365	30:40:00
Field	1	749.4546	351.47803	30:40:00
Field	1	727.6364	347.63638	45:40:00
Field	1	708	206.9091	45:40:00

Figure 5.9: Second CSV file - location related

Machine Learning Model

As mentioned at the beginning of this chapter, the training phase and testing phases of the model were done in Python. Figure 5.10, provides the code responsible for training the model that was then exported into the Android project.

```python
# Separate input features (first 17 columns) and target variable (last column)
X = data.iloc[:, :17]
y = data.iloc[:, -1]
# Repeat experiment 20 times
for i in range(20):
    # Use StratifiedShuffleSplit to create balanced train and test sets
    sss = StratifiedShuffleSplit(n_splits=1, test_size=0.2, random_state=42 + i)
    for train_index, test_index in sss.split(X, y):
        X_train, X_test = X.iloc[train_index], X.iloc[test_index]
        y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    # Normalize input features
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

    # Apply SMOTE to balance the dataset
    smote = SMOTE(random_state=42)
    X_train_resampled, y_train_resampled = smote.fit_resample(X_train_scaled, y_train)

    # Train SVM model with balanced class weights
    class_weights = {"PNGA": 1, "PGA": 1.5}  # Adjust the weight for the first class
    svm_model = SVC(kernel='rbf', C=1, gamma='scale', class_weight=class_weights,
    probability=True)
    svm_model.fit(X_train_resampled, y_train_resampled)

    # Make predictions on the test set
    y_pred = svm_model.predict(X_test_scaled)
```

Figure 5.10: SVM model training

Coding Practices

The coding methodology employed in this project strictly adhered to good coding practices. This was achieved by strategically dividing the project into distinct classes, as demonstrated in Table 5.1, which facilitated a well-organized and modular structure. Moreover, the implementation of sensible comments and appropriately named variables, evidenced by Figure 5.10, ensured enhanced readability and maintainability of the code. These practices collectively contributed to a robust and efficient programming project.

5.2 Testing

As stated in Section 3.5, the project will be subject to the following testing methods.

5.2.1 Interface Testing

The following aspects will be considered during interface testing:

Button responsiveness

Ensuring that all buttons within the application respond accurately to user inputs and trigger the intended actions. The testing process involved the following steps:

- Identification of all buttons in the application, including navigation buttons, action buttons, and any other interactive elements.

- Execution of a series of tests to simulate user interactions, such as taps, long presses, and swipes, to verify that each button responds as expected.

Button placement

Testing button placement involves evaluating the location, size, and arrangement of buttons within the application to ensure that they are easily accessible, visually appealing, and contribute to a positive user experience. The steps involved in testing button placement include:

- Review of design guidelines: Begin by reviewing the design guidelines and best practices for the targeted platform to ensure that the button placement adheres to the recommended standards - comply with Section 3.2.2.

- Assessment of button hierarchy: Evaluate the arrangement of buttons based on their importance and frequency of use. Ensure that primary actions are prominently placed and easily accessible, while secondary actions are placed accordingly - comply with Section 3.2.2.

- Accessibility evaluation: Test the button placement with respect to accessibility, ensuring that they are available and are not obstructed by other interface elements. Check if buttons that are invisible by default (”On Net”, ”Not On Net”, ”Blocked”, ”First Assist”, ”Second Assist”) are visible when required (after pressing ”Shot” or ”Assist” buttons). Check if buttons that are disabled based on the state of the application are becoming enabled when needed - ”Previous Period” if the period number is larger than 1, ”Next Period” if the period is not ”OT”, location-related buttons if the time of the event is not complying with the period number.

Activity transitions

Testing activity transitions involves evaluating the smoothness, responsiveness, and visual appeal of screen transitions within the application, ensuring that they contribute to a seamless and positive user experience. The steps involved in testing activity transitions include:

- Identify all transitions: List all activity transitions within the application and check if the behaviour is the intended one - same as in Figure 4.1.
- Evaluate responsiveness: Assess the responsiveness of the transitions by measuring the time it takes for the new activity to load and become interactive. Transitions should be smooth and quick, with minimal lag or delays.

Compatibility

Testing the application on the targeted Android devices and screen resolutions to ensure consistency in appearance and functionality.

- The application will be run on one emulator integrated into Android Studio and one physical tablet: Pixel C and Galaxy Tab S7.

5.2.2 Functionality Testing

The following aspects will be considered during functionality testing:

Data accuracy and persistence

Ensuring that the application accurately records game events according to user input and properly stores the data for later access.

- Verify that the number of button presses for each specific action is correctly represented in the CSV and PDF files and that the data containers are successfully exported to these formats. This will allow for effective analysis and review of the game data.
- Verify if after generating a report, the values of each variable are reset.

Error handling

Testing the application's ability to handle potential errors, such as invalid inputs or interruptions during the recording process, without compromising the data integrity or user experience.

- Verify the ability to only introduce valid timestamps for each event (timestamp should correlate to the period).
- Verify the ability to only set positive values to all stats except for the "+/- Differential" metric.
- Verify if exiting the application and re-opening it or pressing the "Back" button affects the state of the variables.

5.2.3 Model Testing

The model was tested using the validation set, which was created by splitting the initial data set into 80% for training and 20% for validation. To validate the model, the validation set was inputted, and the model's predictions were compared to the actual values. As mentioned in Section 3.5.2, four key metrics were used to assess the model's performance: accuracy, recall, precision, and F1 score. Additionally, a confusion matrix heat map was generated to compare the number of actual classes to the predicted ones. This entire process was repeated 20 times to obtain the average metrics of the model, and the **best-performing model was exported** into the application. The code responsible for the model validation can be seen in Figure 5.11.

```python
        # Evaluate model performance
        accuracy = accuracy_score(y_test, y_pred)
        f1 = f1_score(y_test, y_pred, average='weighted')
        precision = precision_score(y_test, y_pred, average='weighted')
        recall = recall_score(y_test, y_pred, average='weighted')

        # Append the results to the lists
        precision_list.append(precision)
        recall_list.append(recall)
        accuracy_list.append(accuracy)
        f1_list.append(f1)

        # Update the best model if the current accuracy is higher
        if accuracy > best_accuracy:
            best_accuracy = accuracy
            best_svm_model = svm_model

    # Calculate average statistics
    avg_accuracy = np.mean(accuracy_list)
    avg_f1 = np.mean(f1_list)
    avg_precision = np.mean(precision_list)
    avg_recall = np.mean(recall_list)
    # Use the best model to make predictions on the test set
    y_pred_best = best_svm_model.predict(X_test_scaled)
    best_f1 = f1_score(y_test, y_pred_best, average='weighted')

    # Print average and best statistics
    print(f"Average Accuracy: {avg_accuracy:.2f}")
    print(f"Average F1-score: {avg_f1:.2f}")
    print(f"Average Precision: {avg_precision:.2f}")
    print(f"Average Recall: {avg_recall:.2f}")
    print(f"Best Accuracy: {best_accuracy:.2f}")
    print(f"Best F1-score: {best_f1:.2f}")
```

Figure 5.11: Code of Model Validation

Chapter 6

Results and discussion

6.1 Results

The results of the tests defined in Section 5.2.1 were manually achieved, and the results of each component are described below.

6.1.1 Interface Testing Results

Button responsiveness

The test results demonstrated that the button responsiveness was successfully implemented across various devices and resolutions. All buttons within the application responded accurately to user inputs and triggered the intended actions without any delays or inconsistencies.

Button placement

The test results show that the button placement, following the design mock-up in Figure 4.2 and implemented as in Figure 5.5, has successfully created a user-friendly and intuitive interface. Buttons were thoughtfully positioned to facilitate easy access and reduced effort for users when interacting with the app.

The arrangement of buttons, based on their **importance and frequency of use**, was found to be effective in ensuring that primary actions were prominently placed and easily accessible, while secondary actions were positioned accordingly. Accessibility evaluation demonstrated that the button placement provided adequate access without being obstructed by other interface elements.

Buttons that were initially invisible by default (such as "On Net", "Not On Net", "Blocked", "First Assist", and "Second Assist") were confirmed to be visible when required (after pressing "Shot" or "Assist" buttons). Moreover, buttons that were disabled based on the state of the application were found to become enabled when needed: "Previous Period" if the period number was larger than 1, "Next Period" if the period was not "OT", and location-related buttons if the time of the event did not comply with the period number.

Activity transition

During the testing process, all activity transitions were identified and their behaviour was confirmed to be consistent with the intended design, as depicted in Figure 4.1. Responsiveness evaluation revealed that the transitions were smooth and quick, with minimal lag or delays. The maximum time for transitions involving intense background processes, such as generating output files, **did not exceed 1.5 seconds**, indicating efficient performance.

Compatibility

Compatibility tests were performed on the targeted Android devices to ensure consistency in appearance and functionality. The application was run on an emulator integrated within Android Studio (Pixel C) and a physical tablet (Galaxy Tab S7).

The tests confirmed that the application's appearance and functionality were **consistent** across both devices. A screenshot of the Report activity on the Galaxy Tab S7 device can be seen in Figure 6.1, while Figure 5.5 showcased the application running on the Pixel C emulator.

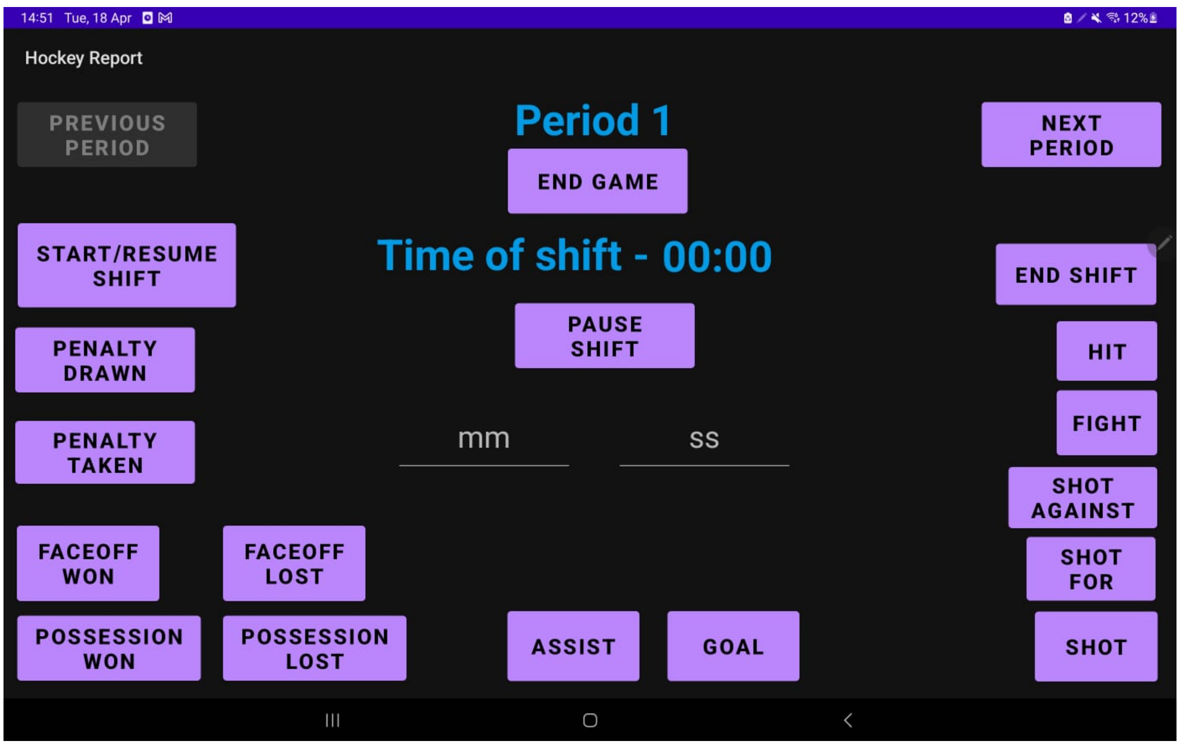

Figure 6.1: Report Activity on physical device

6.1.2 Functionality Testing Results

Data accuracy and persistence

The tests conducted to ensure data accuracy and persistence confirmed that the application accurately records game events according to user input and properly stores the data for later access.

During the testing process, it was verified that the number of button presses for each specific action was correctly represented in the CSV and PDF files and that the data containers were successfully exported to these formats. This enables effective analysis and review of the game data.

Additionally, the tests confirmed that after generating a report, the values of each variable were reset, ensuring a clean slate for new data collection and preventing any interference with subsequent game data recording.

Error handling

Error handling tests were conducted to assess the application's ability to handle potential errors, such as invalid inputs or interruptions during the recording process, without compromising data integrity or user experience.

The tests confirmed the following:

- The application only allowed the introduction of valid timestamps for each event, ensuring that timestamps correlate to the appropriate period. For example, if a timestamp of minute 25 was entered during the first period, which ranges from minute 0 to minute 20, the application displayed an invalid time notification (Figure 6.2) and disabled the buttons for actions that require a timestamp (Figure 6.3) until a valid time was entered.

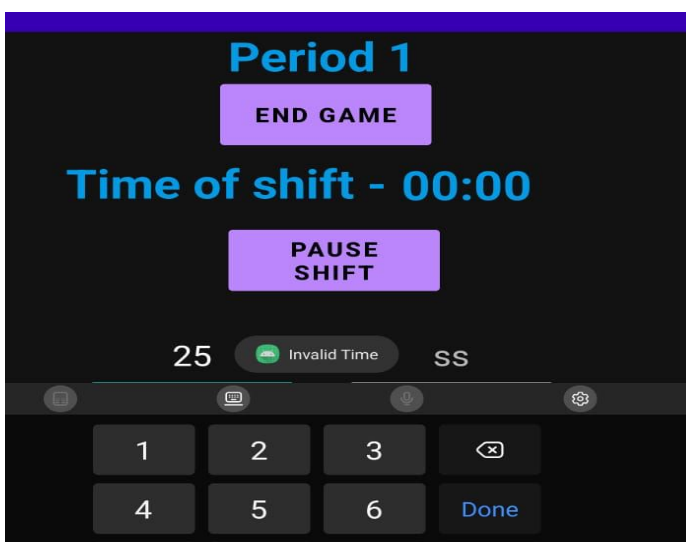

Figure 6.2: Invalid Time Notification

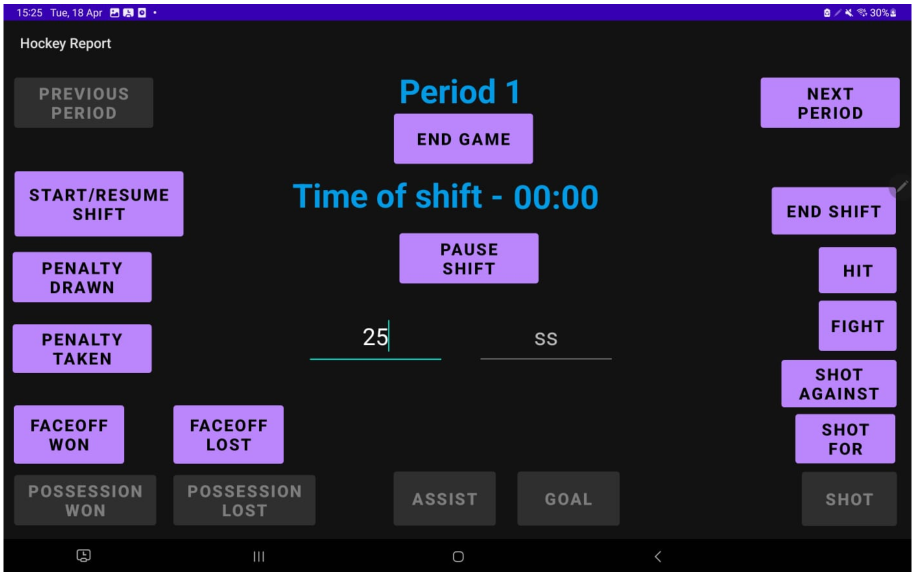

Figure 6.3: Disabled Button - Invalid Time

- The application only allowed the setting of positive values for all stats, except for the "+/- Differential" metric.
- Exiting the application and re-opening it, or pressing the "Back" button, did not affect the state of the variables, ensuring data integrity and consistency in the user experience.

Conclusion: Interface Testing Results

The comprehensive testing conducted for the application has demonstrated its success in meeting all requirements and providing a seamless, robust, and reliable user experience.

The button placement tests verified that the strategic positioning of frequently used buttons effectively improved the overall user experience, ensuring easy access and reduced effort for users when interacting with the app.

The activity transitions tests showcased the smoothness, responsiveness, and visual appeal of screen transitions within the application, contributing to a positive and seamless user experience.

Compatibility tests confirmed that the application is consistent in appearance and functionality across targeted Android devices, resulting in a uniform and reliable user experience.

Data accuracy and persistence tests demonstrated that the application accurately records game events according to user input and properly stores the data for later access, ensuring a reliable and efficient data recording and storage process.

Finally, error handling tests established that the application effectively manages potential errors, maintaining a seamless user experience while ensuring data integrity.

In summary, the successful completion of all tests reaffirms the application's robustness and reliability, providing users with an enjoyable experience across various aspects of the app.

Table 6.1 presents the tests conducted and their respective results, further emphasizing the application's success in meeting all requirements.

Test	Result
Buttons responsiveness	Pass
Buttons placement	Pass
Activity transitions	Pass
Compatibility	Pass
Data accuracy and persistence	Pass
Errors handling	Pass

Table 6.1: Interface Results

6.1.3 Model Testing Results

During the validation process, 12 different model configurations were tested by iteratively changing the hyperparameters, such as the kernel function, C, gamma, and class weight. Due to the imbalanced dataset, the best class weight was found to be "PNGA": 1, "PGA": 1.5, which helped improve the model's performance. After the first 9 configurations, it became evident that the radial basis function (RBF) kernel, along with the mentioned class weights, yielded the best results. Consequently, three more configurations were tested using the RBF kernel and the optimal class weights, while varying the C parameter.

It is worth noting that the alternative option for the gamma parameter ("auto") produced extremely poor results that were not worth including in the comparison table. Based on the validation results presented in Table 6.2, the best model of the highlighted configuration (*) was chosen to be exported into the application, as that demonstrated the highest metrics overall, achieving **0.61 accuracy and 0.64 F1 score**.

KERNEL	C	gamma	class_weight	iterations	avg accuracy	avg f1	avg recall	avg precision
rbf	1	scale	balanced	20	0.53	0.54	0.53	0.55
rbf*	1	scale	"PNGA": 1, "PGA": 1.5	20	0.6	0.54	0.6	0.56
rbf	1	scale	"PNGA": 1, "PGA": 2	20	0.61	0.5	0.61	0.56
poly	1	scale	balanced	20	0.5	0.5	0.5	0.55
poly	1	scale	"PNGA": 1, "PGA": 1.5	20	0.59	0.54	0.59	0.54
poly	1	scale	"PNGA": 1, "PGA": 2	20	0.6	0.51	0.6	0.54
sigmoid	1	scale	balanced	20	0.5	0.5	0.5	0.52
sigmoid	1	scale	"PNGA": 1, "PGA": 1.5	20	0.53	0.52	0.53	0.52
sigmoid	1	scale	"PNGA": 1, "PGA": 2	20	0.55	0.53	0.55	0.53
rbf	1.2	scale	"PNGA": 1, "PGA": 1.5	20	0.59	0.55	0.59	0.56
rbf	0.8	scale	"PNGA": 1, "PGA": 1.5	20	0.6	0.54	0.6	0.56
rbf	1.4	scale	"PNGA": 1, "PGA": 1.5	20	0.59	0.55	0.59	0.55

Table 6.2: Model Results - *this model was integrated in the application

For that specific model, a confusion matrix has been generated and can be viewed in Figure 6.4.

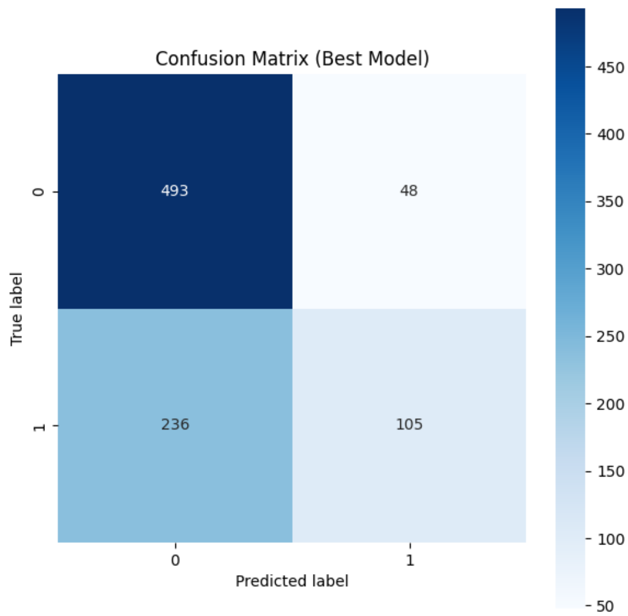

Figure 6.4: Confusion Matrix of best model

To have a better understanding of the input and the output of the algorithm, Table 6.3 illustrates the first input sample of the testing dataset and Figure 6.5 shows the actual label of the example, along with the probabilities of each class. In this case, the model **correctly** predicts that the player in question will score at least one point (goal or assist) the next game he plays.

Stat	G	A	PTS	S	+/-:	TOI (seconds)	GF	GA	G1	G2	G3	A1	A2	A3	S1	S2	S3
Value	1	1	2	5	-1	1219	6	1	0	0	0	1	1	1	2	3	3

Table 6.3: Input sample of the testing dataset

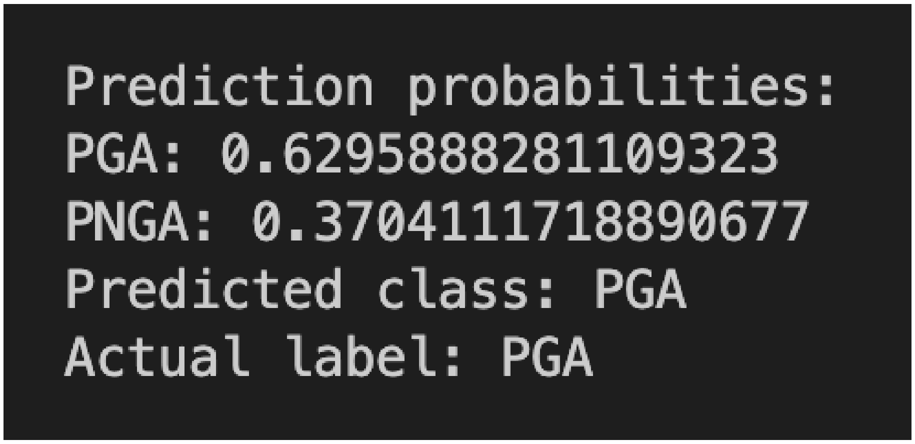

Figure 6.5: Sample results - correctly labelled

It is essential to acknowledge that the model is relatively simple, consisting of only 17 easily obtainable input parameters, and that hockey is an inherently unpredictable sport. The primary objective of the model from the outset was to showcase how the data collected by the interface could be employed to train more sophisticated models in the future.
The successful validation of the model, combined with the selection of the top-performing configuration, underscores the potential of the gathered data and the utility of the model when incorporated into the application as a part of the PDF report.

6.2 Discussion

6.2.1 Goals achieved

The project successfully achieved its main goals, which were to develop a reliable and effective Android application for capturing data regarding ice hockey players' actions and generating insightful reports to benefit coaches, scouts, and players. The importance and relevance of the data were also demonstrated by integrating it as input parameters to a predictive model that was incorporated in the report.

The interface test results showed that the application was user-friendly, responsive, and consistent across targeted Android devices and screen resolutions.

The functionality test results demonstrated that the core features of the application, such as recording game events for an ice hockey player and generating comprehensive performance reports, were working as intended.

Table 6.4 presents a list of the main requirements of the project and how they were achieved after implementation.

Requirement	Implementation
Accumulate a comprehensive range of statistical data to provide a thorough analysis of performance outcomes.	The first CSV file captures 70 different parameters, including the ones mentioned in the Literature Review, Section 2.1.2 – showcased in Figure 5.8.
Minimize the number of Clicks: The application should allow users to input player actions in 4 or fewer clicks.	The action that requires the most clicks is the "Shot on net", which can be captured in 4 clicks – showcased in Figure A.6.
Logical Organization: Buttons should be grouped by the type of action to facilitate quick and intuitive data capturing.	The organization was presented in the Design chapter (Section 4.1), according to the guidelines presented in the Literature Review, Section 2.2.1 (Information Architecture and Classification) – showcased in Figure 4.2
Button Placement: Actions that occur more frequently, such as changes of possessions and shots, should be located near the edges of the screen for easy access.	In accordance with the results section of [35] and presented in Figure 2.1 and Figure 2.2 (heatmaps of accessibility for different types of tablet grips), the positions of the most frequently used buttons were implemented in the GUI as in Figure 5.5.
Button size: To minimize the risk of misclicking and enhance user experience, the size of the buttons should be approximately 3 cm^2.	In accordance with the EIA's recommendation[40] presented in Section 2.2.1 (Button Size), the size of the implemented buttons is approximately 3 cm^2, as in Figure 5.5.
Output: The generated PDF report should be well-organized, comprehensive and insightful.	Comparably to the InStat's report (Figure 2.3), presented in Section 2.4 (Similar services), the generated PDF report encapsulates the overall productivity of the player in the recorded game, separating different aspects of his performance into different, easy to understand categories. Figure 4.5 presents the logical organization of the report, while Figure 5.6 demonstrates the final implementation.
Output: The CSV files should allow users to input the generated data to a predictive model.	The concept of inputting the generated data into a predictive model was demonstrated by integrating into the interface a SVM model that takes as input, data captured by the application. This is showcased as a part of the PDF report in Figure 5.6.

Table 6.4: Overview of the main requirements and how they were achieved

6.2.2 Limitations and future works

Despite the project's success, some limitations and areas for future improvements were identified:

- Feature expansion: The current application focuses on recording basic game events and generating performance reports based on manually given input by the user. Future enhancements could include incorporating video analysis to automatically generate input that would provide richer insights into the players' performance.
- Integration with other platforms: The application could be further developed to allow integration with other platforms, such as team management systems or sports analytics software, to provide a more comprehensive solution for ice hockey teams.
- Customization options: Future versions of the application could include customization options for users, allowing them to tailor the application to their specific needs, such as

tracking different stats, using customized event labels, or predicting performance based on the opponent's strength points or weaknesses.

Chapter 7

Conclusion

This project aimed to bridge the gap in the ice hockey community regarding the limited availability of databases for training machine learning predictive models. Consequently, an Android tablet application was developed, facilitating the real-time recording of hockey players' performance data during games.

The GUI design, a critical aspect of the project, was meticulously crafted to optimize user experience by adhering to guidelines derived from the Literature Review chapter. This careful attention to detail, encompassing the strategic positioning and sizing of interface buttons, greatly contributed to the application's overall usability.

The application's dual output comprises a PDF report and two CSV files. The report, inspired by an existing service detailed in the second chapter, presents the player's overall contribution to the game. The CSV files, totalling more than 70 parameters, act as a database for potential use as input samples in predictive models.

As outlined in the Results chapter, the application's interface, functionality, and predictive model (which was integrated into the application to prove the project's concept) exhibited reliability. Each feature mentioned in the Requirements chapter was implemented following the guidelines and specifications provided in the Design chapter. This approach resulted in an application that performed as expected in all aspects.

In conclusion, I consider this project a success, as it provides a practical and reliable tool for capturing ice hockey performance data. Its potential to significantly impact budget-constrained teams and players will ultimately contribute to the development of the ice hockey community, making it a helpful resource for those involved in the sport.

Appendix A

Appendix

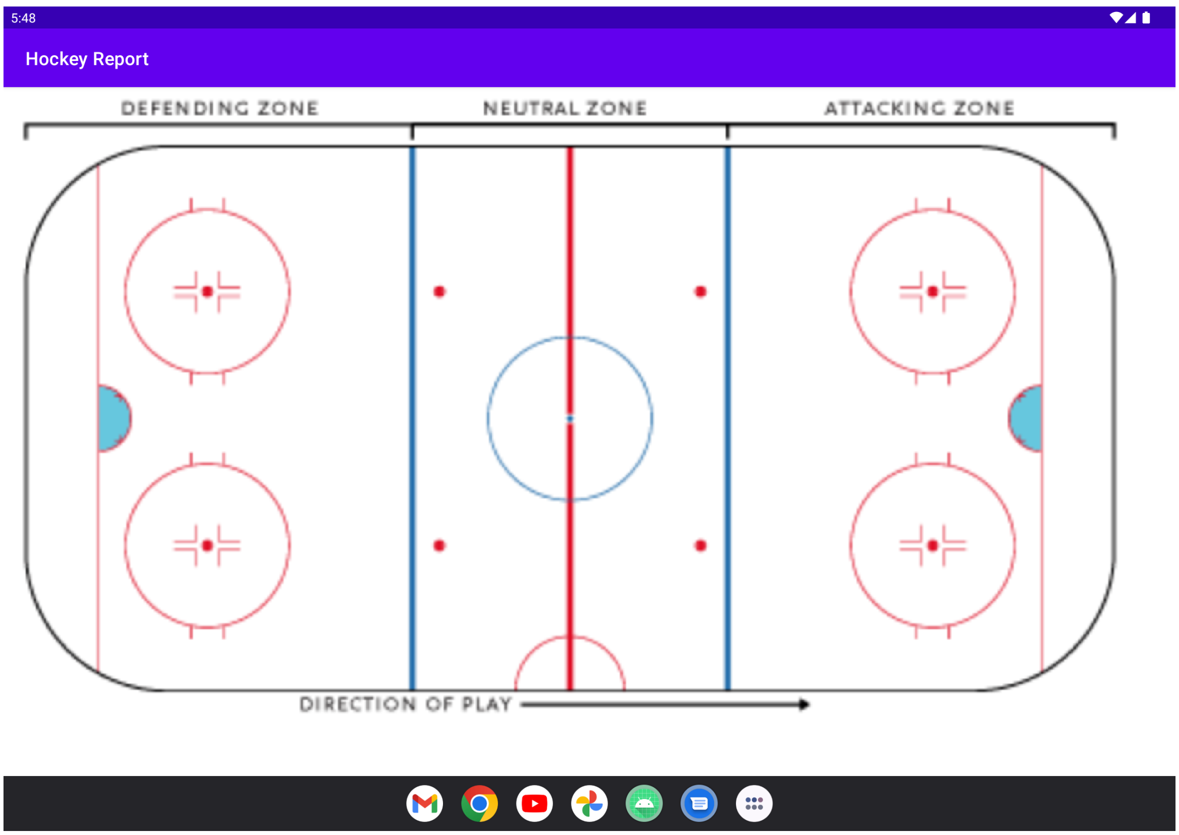

Figure A.1: Field Activity

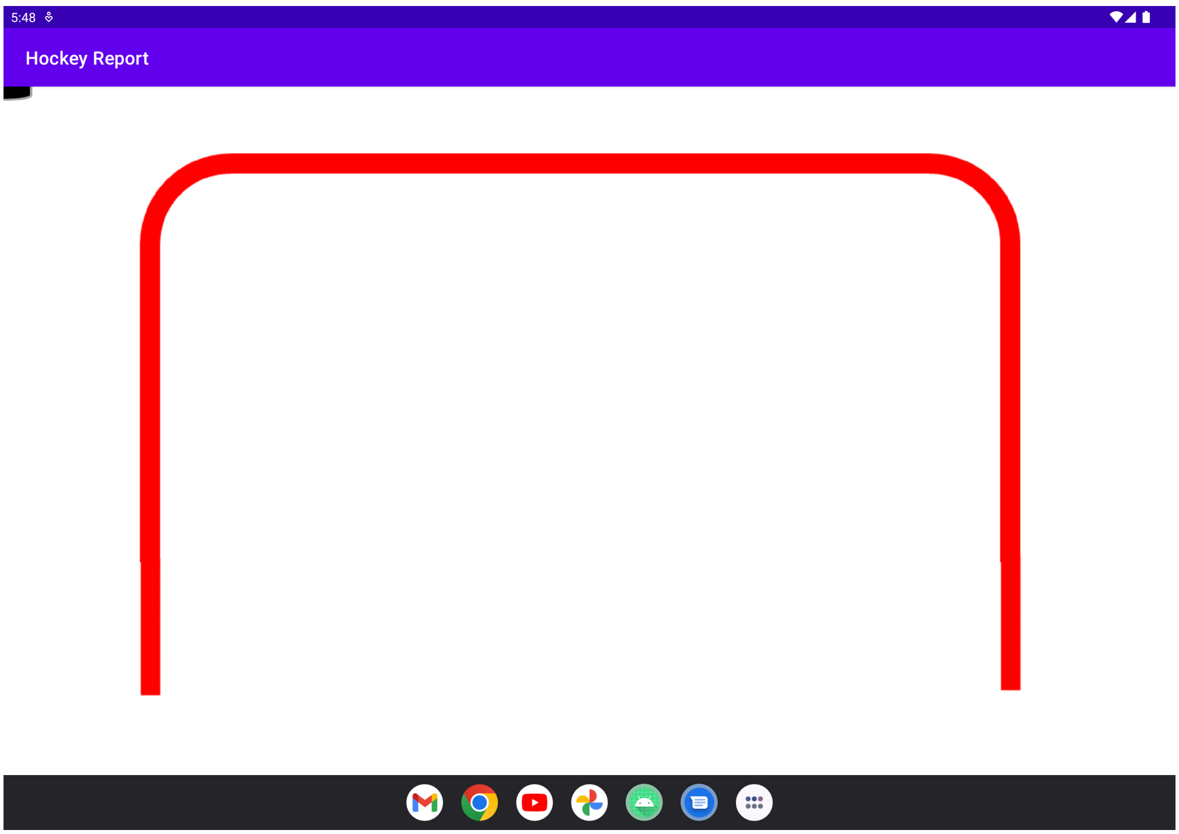

Figure A.2: Net Activity

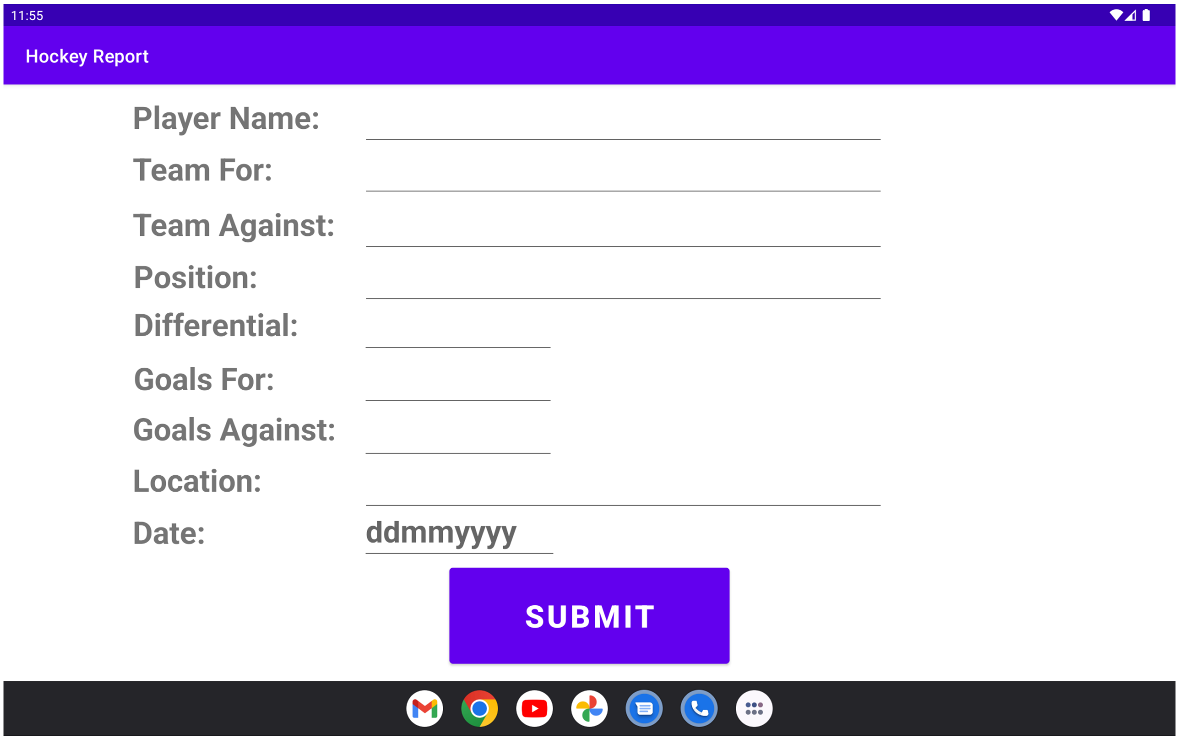

Figure A.3: GameInfo Activity

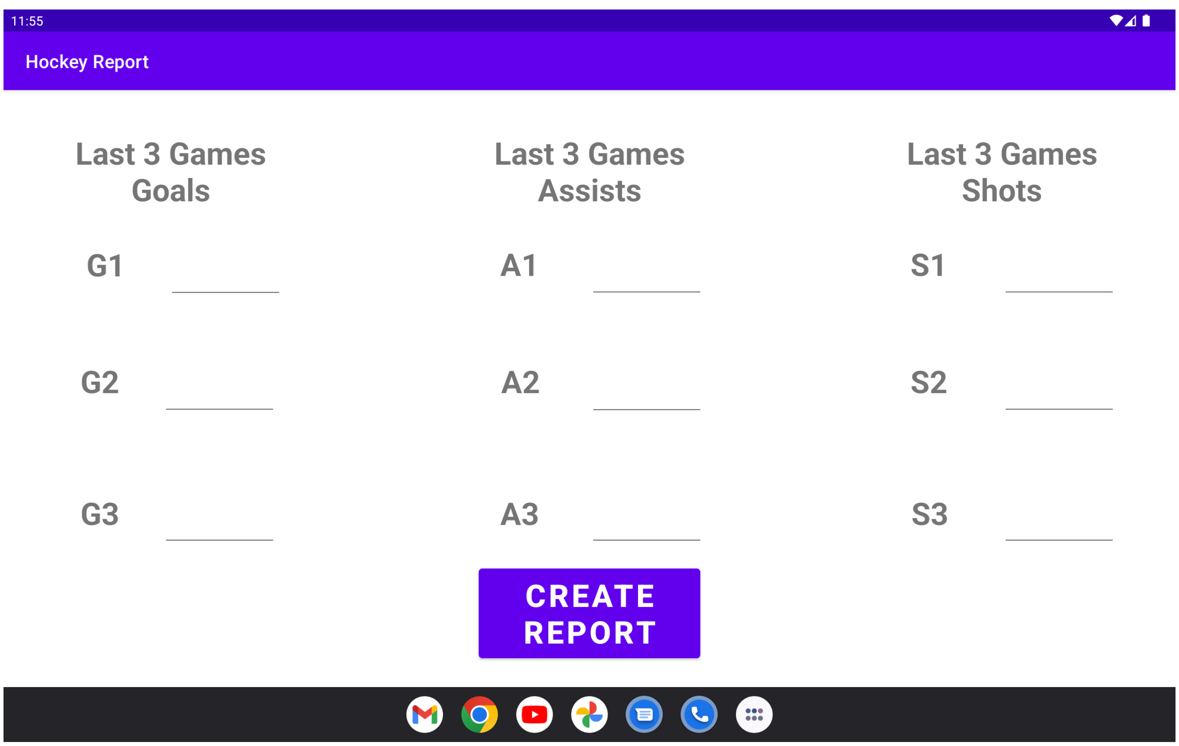

Figure A.4: PastPerformance Activity

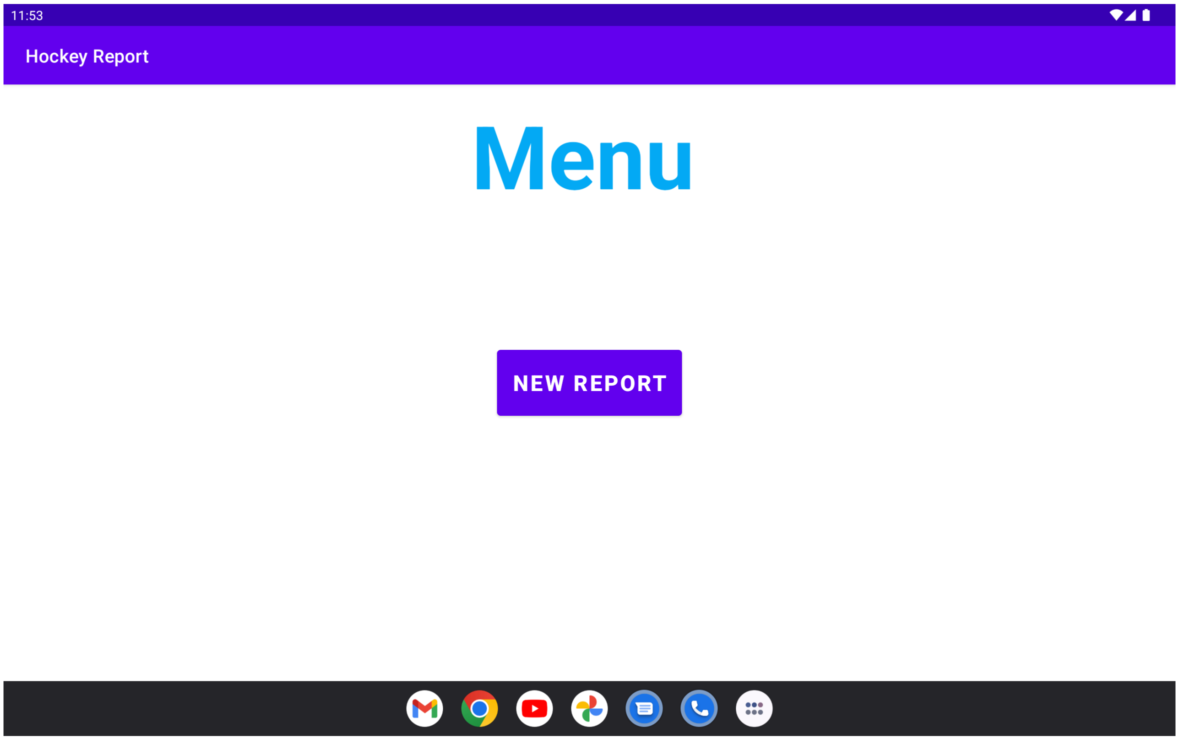

Figure A.5: Menu Activity

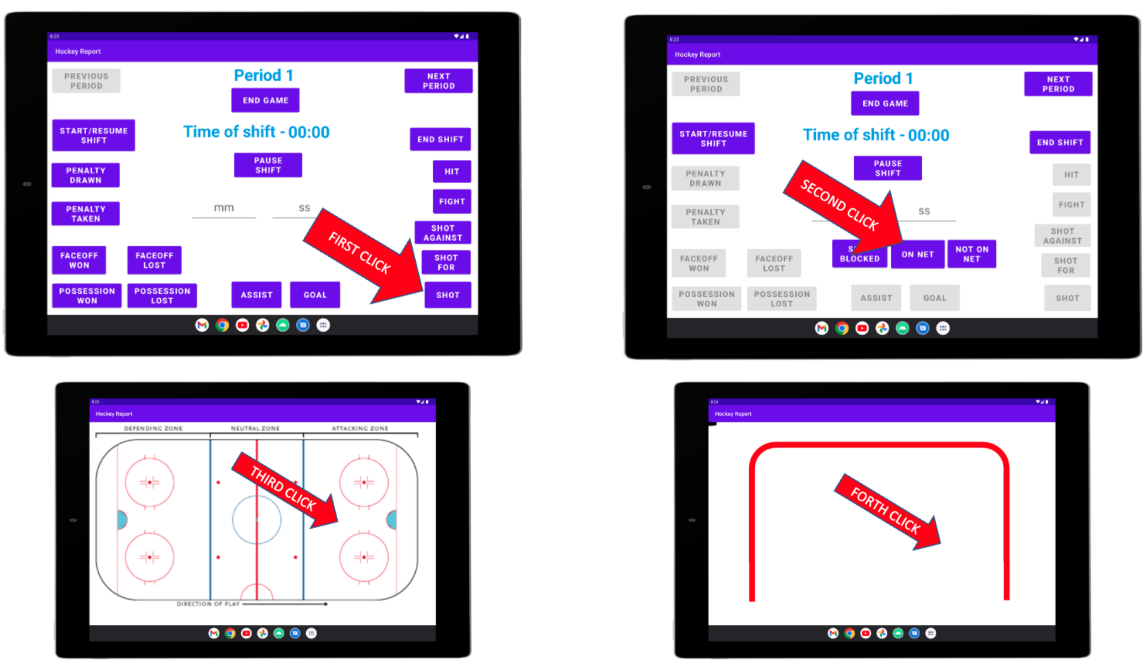

Figure A.6: Shot on net Action - 4 is the maximum number of clicks to record

Bibliography

[1] Android studio. https://developer.android.com/studio. Accessed on 08.05.2023.

[2] Catapult - ice hockey. https://www.catapultsports.com/sports/ice-hockey. Accessed on 25.04.2023.

[3] Firstbeat sports for ice hockey. https://content.firstbeat.com/firstbeat-sports-for-ice-hockey. Accessed on 25.04.2023.

[4] Hockey coach vision. https://hockeycoachvision.com/. Accessed on 25.04.2023.

[5] Sportlogiq - ice hockey. https://sportlogiq.com/en/hockey. Accessed on 19.04.2023.

[6] isnipe hockey shooting trainer. https://play.google.com/store/apps/details?id=com.howtohockey.isnipe, Apr 2014. Accessed on 25.04.2023.

[7] *Applications of Machine Learning 2021*. International Society for Optical Engineering, 2021.

[8] Hudl acquires sports performance analysis provider instat. https://www.hudl.com/blog/hudl-instat, Aug 2022. Accessed on 19.04.2023.

[9] Hockeytracker. https://apps.apple.com/us/app/hockeytracker/id1297971760, Jan 2023. Accessed on 25.04.2023.

[10] Take your coaching to the next level. https://coachthem.com/, 2023. Accessed on 25.04.2023.

[11] Abdiyeva-Aliyeva, G., Aliyev, J., and Sadigov, U. Application of classification algorithms of machine learning in cybersecurity. *Procedia computer science 215* (2022), 909–919.

[12] Aela Editorial. Information architecture: How to organize ui content. https://aelaschool.com/en/userexperience/information-architecture-organize-\ui-content/, July 2022.

[13] Amazon Staff. Nhl unveils first ai-powered stat to predict face-off winner before puck is dropped. https://www.aboutamazon.

`com/news/aws/nhl-unveils-first-ai-powered-stat-to-predict-\face-off-winner-before-puck-is-dropped`, Mar. 2022. Accessed: 28.04.2023.

[14] Ayodele, T. O. Types of machine learning algorithms. *New advances in machine learning 3* (2010), 19–48.

[15] Aziz, S., Dowling, M., Hammami, H., and Piepenbrink, A. Machine learning in finance: A topic modeling approach. *European financial management : the journal of the European Financial Management Association 28*, 3 (2022), 744–770.

[16] BURNS, M. J. Top four sports technology storylines to follow during the 2016 nhl season. `https://www.si.com/media/2016/10/19/sports-technology-2016-nhl-season-\tablets-puck-tracking`, Oct. 2016. Accessed: 16.04.2023.

[17] Busby, M. The impact of artificial intelligence on sports betting. `https://www.analyticsinsight.net/the-impact-of-artificial-intelligence-on-sports-betting/`, Feb. 2023. Accessed: 28.04.2023.

[18] Cortes, C., and Vapnik, V. Support-vector networks. *Machine learning 20* (1995), 273–297.

[19] Cox, M. H., Miles, D. S., Verde, T. J., and Rhodes, E. C. Applied physiology of ice hockey. *Sports Med 19*, 3 (Mar. 1995), 184–201.

[20] Dasgupta, N. Top use cases for machine learning in pharma. `https://www.rxdatascience.com/blog/top-use-cases-for-machine-learning-in-pharma`, April 2020. Accessed on 25.04.2023.

[21] Dyer, B. The controversy of sports technology: a systematic review. *SpringerPlus 4*, 1 (Sep 2015).

[22] FloHockey Staff. How fast can a hockey player skate? `https://www.flohockey.tv/articles/7949700-how-fast-can-a-hockey-player-skate`, 2022.

[23] Gu, W., Foster, K., Shang, J., and Wei, L. A game-predicting expert system using big data and machine learning. *Expert Systems with Applications 130* (2019), 293–305.

[24] Han, M. Ai-powered hockey analytics: A game changer. `https://medium.com/syncedreview/ai-powered-hockey-analytics-a-game-changer-\8534b2e263aa`, Apr. 2018. Accessed: 28.04.2023.

[25] Horvat, T., and Job, J. The use of machine learning in sport outcome prediction: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 10*, 5 (2020), e1380.

[26] IIHF. Survey of players. `https://www.iihf.com/en/static/5324/survey-of-players`. Accessed: 23.11.2022.

[27] IndustryTrends. Revealed: how bookies use ai to keep gamblers hooked. `https://www.theguardian.com/technology/2018/apr/30/bookies-using-ai-to-keep-\gamblers-hooked-insiders-say`, Apr. 2018. Accessed: 28.04.2023.

[28] James, B. *The Bill James historical baseball abstract.* Villard Books, 1987.

[29] Jin, Z., Plocher, T., and Kiff, L. Touch screen user interfaces for older adults: Button size and spacing. vol. 4554, pp. 933–941.

[30] Lehmus Persson, T., Kozlica, H., Carlsson, N., and Lambrix, P. Prediction of tiers in the ranking of ice hockey players. In *Machine Learning and Data Mining for Sports Analytics* (Cham, 2020), U. Brefeld, J. Davis, J. Van Haaren, and A. Zimmermann, Eds., Springer International Publishing, pp. 89–100.

[31] Macdonald, B. An expected goals model for evaluating nhl teams and players. In *Proceedings of the 2012 MIT Sloan Sports Analytics Conference* (2012).

[32] Marsh, J. H. National hockey league (nhl). *The Canadian Encyclopedia*.

[33] Minaee, S. 20 popular machine learning metrics. part 1: Classification regression evaluation metrics. `https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-\evaluation-metrics-1ca3e282a2ce`, Oct 2019. Accessed on 08.05.2023.

[34] Morville, P., and Rosenfeld, L. *Information architecture for the World Wide Web: Designing large-scale web sites.* " O'Reilly Media, Inc.", 2006.

[35] Odell, D., and Chandrasekaran, V. Enabling comfortable thumb interaction in tablet computers: a windows 8 case study. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (2012), vol. 56, SAGE Publications Sage CA: Los Angeles, CA, pp. 1907–1911.

[36] Patle, A., and Chouhan, D. S. Svm kernel functions for classification. In *2013 International Conference on Advances in Technology and Engineering (ICATE)* (2013), pp. 1–9.

[37] Sens, R. Ux design 101: Information architecture — structuring organizing content. `https://blog.prototypr.io/ux-design-101-information-architecture-structuring-organizing-\content-967833278b41`, Oct. 2019. Accessed: 30.04.2023.

[38] Shearin, Z., and Pazour, J. A. Nhl's point system: Fairness optimization and game play statistical analysis. In *IIE Annual Conference. Proceedings* (2018), Institute of Industrial and Systems Engineers (IISE), pp. 188–193.

[39] SportyTellEditors. Top-10 most popular sports in the world 2023. `https://sportytell.com/sports/most-popular-sports-world`. Accessed: 17.04.2023.

[40] Technologies, M. Resource guide for accessibility: Design of consumer electronics, 1996.

[41] Vollman, R., Awad, T., and Fyffe, I. *Hockey Abstract Presents... Stat Shot: The Ultimate Guide to Hockey Analytics.* Ecw Press, 2016.

[42] Yoosefzadeh-Najafabadi, M., Earl, H. J., Tulpan, D., Sulik, J., and Eskandari, M. Application of machine learning algorithms in plant breeding: Predicting yield from hyperspectral reflectance in soybean. *Front Plant Sci 11* (Jan. 2021), 624273.

[43] Zhou, Z.-H. *Machine learning.* Springer Nature, 2021.