



# **AICCSA - AI tool for Collecting and Classifying Sports Applications**

University of Oulu  
Information Processing Science  
Master's Thesis  
Juhani Karjalainen  
2026

## Abstract

Mobile applications for sports and, Health & Fitness form a vast and rapidly growing ecosystem, where comprehensive research requires access to extensive application datasets. However, collecting and categorizing such data for Sport Human-Computer Interaction research using manual methods is a highly labour-intensive and time-consuming process. To address this challenge, this thesis adopts a Design Science Research methodology to develop and evaluate an AI-based software tool called AI tool for Collecting and Classifying Sports Applications (AICCSA) that automates the collection and categorization of mobile application data. Recent advances in generative artificial intelligence have enabled the development of tools capable of automating repetitive tasks in areas such as data processing, programming, and knowledge work. The proposed tool utilizes publicly available APIs for data collection and leverages OpenAI's GPT-5-mini model to classify applications by sport type, purpose, and primary user group. The performance of the proposed artifact was assessed by comparing its classification outcomes with those produced by a human evaluator. The results demonstrate a high level of potential for using AI-based software tools to support Sport HCI research by enabling efficient collection and classification of large application datasets.

### *Keywords*

Sports HCI, Large Language Models, Artificial Intelligence

### *Supervisor*

Doctoral Researcher Marko Moilanen  
PhD, University lecturer Leena Arhipainen

*Conference paper to be submitted (by 7<sup>th</sup> of April 2026)*

Karjalainen, J. (2026). *AICCSA – AI tool for Collecting and Classifying Sports Applications*. To be submitted to the 43<sup>rd</sup> Annual Symposium of Computer Science (TKTP 2026). Turku, Finland 8-9, June, 2026

## Foreword

This thesis originated from a research traineeship at the INTERACT research group at the University of Oulu in autumn 2025. I would like to thank the INTERACT group for the grant and for the opportunity to work on such an interesting topic. During this research, I learned a great deal about the development of AI tools, and working on an interesting subject was truly a joy.

I would like to warmly thank my supervisor Marko Moilanen for his wisdom, dedication of time, and invaluable support from the traineeship through to the ideation, writing, and completion of this thesis. I would also like to thank my supervisor Leena Arhippainen for her encouragement, guidance, and constructive feedback. Furthermore, I wish to thank Mikko Rajanen for reviewing this thesis.

I am grateful to my parents for their lifelong support. Finally, I would like to thank my wife for her endless love and support, without which the completion of this thesis would not have been possible.

Oulu, 9.2.2026

Juhani Karjalainen

## Declaration of Generative AI in the Thesis

The content of this thesis was initially written by the author itself, after which the language was checked and refined using generative AI tools (ChatGPT-5, Gemini 3). GitHub Copilot and OpenAI Codex coding agents were used in the development of the artifact, and ChatGPT-5 was used for brainstorming the architecture of the artifact's code.

# Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
DFS	Daily Fantasy Sports
DSR	Design Science Research
DSRP	Design Science Research Process
FEI	Fédération Equestre Internationale
FRI	Format-Restricting Instructions
GPT	Generative Pre-trained Transformer
GenAI	Generative Artificial Intelligence
HCI	Human-Computer Interaction
IDE	Integrated Development Environment
IP	Intellectual Property
IS	Information Systems
ISO	International Organization for Standardization
IT	Information Technology
JSON	JavaScript Object Notation
JSONL	JavaScript Object Notation Lines
LLM	Large Language Model
MLM	Masked Language Modelling
MMLU	Massive Multitask Language Understanding
MoE	Mixture of Experts
NL	Natural Language
NLP	Natural Language Processing
NTP	Next Token Prediction
OBD	On-Board Diagnostics

OPT	Open Pre-trained Transformer
PGA	Professional Golfers' Association
SVC	Support Vector Classification
SWE-bench	Software Engineering Benchmark
XML	Extensible Markup Language

# Contents

Abstract .....	2
Foreword .....	3
Declaration of Generative AI in the Thesis.....	4
Abbreviations .....	5
Contents .....	7
1. Introduction .....	8
1.1 Background and motivation.....	8
1.2 Research questions and methods .....	9
1.3 Limitations .....	9
1.4 Structure of the thesis .....	10
2. Background .....	11
2.1 Sports applications .....	11
2.1.1 Definitions and scope.....	11
2.1.2 User groups and use contexts.....	11
2.1.3 Purposes .....	12
2.2 Health & Fitness applications .....	14
2.3 Artificial Intelligence and generative AI .....	15
2.3.1 Overview of AI and generative AI technologies.....	15
2.3.2 Large Language Models .....	16
2.3.3 Prompt engineering .....	17
2.3.4 Coding agents .....	19
2.4 Automation in application categorization.....	21
3. Research methodology .....	22
3.1 Design science research .....	22
3.1.1 Problem identification .....	23
3.1.2 Definition of objectives .....	23
3.1.3 Design and development .....	23
3.1.4 Demonstration .....	24
3.1.5 Evaluation.....	24
3.1.6 Communication .....	24
3.2 Literature review .....	24
4. Development of the AI tool.....	26
4.1 Problem description .....	26
4.2 Data collection phase .....	26
4.3 Data classifying phase .....	29
5. Evaluation.....	35
5.1 Evaluation criteria.....	35
5.2 Functional suitability .....	35
5.2.1 Functional completeness .....	36
5.2.2 Functional correctness .....	36
5.3 Performance efficiency .....	39
5.4 Interaction capability .....	40
6. Discussion .....	42
6.1 Answers to research questions.....	42
6.2 Contributions .....	43
6.3 Limitations and future research .....	46
7. Conclusion.....	48
References .....	49

# 1. Introduction

Sports and Health & Fitness applications are widely available and commonly used by the general public to track their hobbies and physical activity, as well as by professional athletes to improve performance. According to AppBrain (2025) there are 74 248 applications in Health & Fitness category and 35 737 applications in sports category. Due to the vast number of applications in the market, collecting a sufficient number of applications for research purposes can be a very time-consuming process for the researcher to do manually. Two examples of research on applications and the associated data gathering processes are briefly presented below.

For their research Kebede et al. (2018) manually collected 6018 applications related to physical activity by conducting searches in the Google Play Store using 25 specific search terms. The resulting applications were then manually screened by two reviewers to identify relevant physical activity and fitness applications where the language was either English or German. Finally, the dataset was further reviewed to identify and remove duplicate entries and “lite” versions of applications. Finally, the descriptions of 1216 applications were screened according to prior defined inclusion and exclusion criteria.

In some cases, the applications need to be further examined and categorized into different groups. Chembakottu et al., (2023) obtained 2621 sports applications from the Google Play store through manual search. The applications were then manually verified for eligibility and applications not relevant for the study, like games, were removed. From the final set of 2058 apps, a random sample of 326 applications was manually categorized by sport type (football, golf etc.), functionality (training, betting etc.) and the type of analytics used in the apps.

The process in the cases like the ones described above can take lot of human work and time, so a tool that automates the collection and categorization of applications would save a lot of time, result in a broader data set, and enable the classification across the entire dataset, instead of using sampling.

## 1.1 Background and motivation

The work presented in this thesis originates from a research traineeship conducted at the University of Oulu during the autumn of 2025. The traineeship was part of ongoing research within the INTERACT research unit in the field of Sport Human-Computer Interaction (HCI) examining the ecosystem of sports applications and the user groups engaged with it. The primary task of the traineeship was to develop a conceptual map of sports applications and to investigate related academic research. Achieving this required the collection of large-scale dataset of Sports and Health & Fitness applications, followed by their categorization into sport types, purposes, and user groups. The category of Health & Fitness was included due to their substantial functional overlap with sports applications, particularly in areas such as activity and workout tracking. Given the large dataset required to obtain a representative overview of the application market, an automated solution was necessary to manage both data collection and classification. Following the data collection phase, which produced a dataset of over 53 000 mobile applications, it was estimated that manually reviewing and categorizing the applications would require approximately half a year of full-time work. This challenge motivated the development of the AI-based tool presented in this thesis, which completed the

classification task in approximately three and a half hours. The dataset collected during the traineeship forms the resulting conceptual map of sports applications.

## 1.2 Research questions and methods

A major challenge in research on sports applications, and mobile applications more broadly, is the substantial effort required to collect large-scale application data, screen applications for eligibility, and categorize vast numbers of applications into meaningful categories. Manual approaches are highly time-consuming and limit the size of datasets that can be analyzed. An automated approach offers a strong potential to significantly reduce researcher's workload and allow access to broader and more representative application datasets beyond what manual collection methods can reasonably achieve.

Typically, when doing research on sports applications the application data is collected by using search terms like "football", "golf", etc. in either Google Play store or Apple App store, and screen the results (Chembakottu et al., 2023a; Kebede et al., 2018). This is a slow process, and the search results will have lots of applications that are not relevant for the study, like arcade games. When the developers add the application to the store, they are required to add a primary genre for the application. Some of the sports themed arcade games are put into "sports" category, even though they would be a better fit in a "games sports" category. When searching for applications in a sports category, whether it is done manually or using APIs (Application Programming Interface) like apple-store-scraper or play-store-scraper, the results need to be screened, and arcade games and other irrelevant applications removed.

For this research, design science research methodology (Peppers et al. 2007) is utilized to investigate the use of an AI (Artificial Intelligence) tool for collecting sports application data, assessing the relevance of applications to the study, and classifying the applications into sport types, user groups and purposes.

The study will have a literature review (Baumeister & Leary, 1997; Snyder, 2019) section which will be conducted by searching scientific articles from Scopus and Google Scholar databases and using research tools like Keenious, Elicit and Connected Papers. The research methods are described in more detail in chapter 3.

This study aims to answer the following research questions:

**RQ1:** What types of Sports and Health & Fitness applications can be found on the market?

**RQ2:** How reliably can an AI software tool screen the eligibility and categorize Sports and Health & Fitness applications into sport types, purposes and user groups?

## 1.3 Limitations

This research will focus only on applications available on Sports, and Health & Fitness categories in Google Play Store and Apple App Store. The application data is gathered using google-play-scraper and apple-store-scraper APIs. Apple-store-scraper returns a maximum of 200 applications per request and google-play-scraper returns a maximum of 250 applications per request for search terms and 500 applications per request from top lists. These limitations will affect the number of applications that can be collected within a reasonable time. The AI classifier will use the descriptions uploaded to the application page in the app store. Sometimes these descriptions are very limited, and this will affect

the classification. For cost efficiency reasons the AI classifier will use GPT-5-mini model, which is not as advanced as the more high-end end models like GPT-5 or GPT-5-pro. The applications will be categorized into 23 different purposes which might not be exhaustive.

## 1.4 Structure of the thesis

The thesis is organized into seven chapters. Chapter 1 introduces the topic and sets the context for the study. Chapter 2 presents a review of related work, presenting prior research on Sport and Health & Fitness applications. This chapter provides the foundational taxonomies on which the AI tool's categorization is based on and also reviews relevant literature on large language models, coding agents, and prompt engineering. Chapter 3 introduces the design science research, design science research process and literature review methods used in the thesis, followed by chapter 4 which presents the design and development of the AI tool. Chapter 5 presents the evaluation of the performance of the proposed AI tool and Chapter 6 discusses the results in relation to research questions, outlines the contributions of the study to the field, goes through limitations of the study, and proposes suggestions for future research. Finally, chapter 7 concludes the thesis by bringing together the key insights and findings.

## 2. Background

This section provides the theoretical background for the study by reviewing prior research on sports applications, Health & Fitness applications, and artificial intelligence, particularly Large Language Models (LLM), and coding agents. The literature review starts with an examination of sports applications including definitions, user groups and use contexts followed by an introduction to Health & Fitness applications, their main functionalities and taxonomy. Next chapter discusses the artificial intelligence technologies relevant to this study, mainly LLMs and AI-assisted coding agents. Last part of the background section introduces a case where software tool was developed for application categorization.

### 2.1 Sports applications

This subsection introduces the key concepts related to sports mobile applications, including their definitions, the user groups and contexts, as well as their different core purposes.

#### 2.1.1 Definitions and scope

Research on sports applications can be approached from two main perspectives: SportsHCI, which focuses on the human aspects of interaction (Moilanen et al., 2024; Mueller & Young, 2018), and an information systems-focused viewpoint. (Haffner et al., 2025; Xiao et al., 2017)

Tubek and Duplaga (2018) classify sports applications into two major categories: applications focused on tracking and analysing users' physical activity and those that deliver instructional resources aimed at supporting individual training. The capabilities of sports applications have grown substantially with improvements in smartphone technology. Shaw et al. (2021) highlight that modern smartphones can capture a wide range of physiological and biometrical data without any additional equipment. Using built-in components such as microphones, cameras, light sensors, accelerometers, gyroscopes, inclinometers, and magnetometers, these applications can generate detailed measurements directly from the phone itself.

#### 2.1.2 User groups and use contexts

Haffner et al. (2025) proposed a definition for four stakeholder groups to map the actors most affected by sports digitalization across diverse sporting contexts. These stakeholder groups include competitors, support staff, supporters, and governing entities. Their stakeholder model builds on Goebeler et al. (2021) classification which distinguishes between non-competitive and competitive actors within the sports domain. Non-competitive actors refer to individuals who facilitate competitive activities, such as judges, referees and other officials. Competitive actors are individual athletes and teams participating in competitions, as well as supporting personnel such as coaches and team management, and the spectators who engage with and follow the competition.

Haffner et al. (2025) referred to different user groups as stakeholders and proposed a four-tier classification (competitors, support staff, supporters and governing entities) that

moves beyond the traditional competitive/non-competitive distinction. This broader categorization offers terminology that better reflects the range of actors discussed in the IS (Information Systems) literature on sports, not only those involved in competitive settings. The stakeholder groups differ across three dimensions: physical proximity to the contest, competition influence, and group size.

Proximity refers to the physical distance of each stakeholder from the competition site. Competitors, such as athletes and teams, are closest, as they directly perform the sport and without them, there would not be a contest. Support staff (e.g., coaches, physiotherapists etc.) and governing entities (e.g., officials, judges) also operate close to the competition area as they assist and regulate the play. Supporters, who might be physically attending the venue or following online, have the least physical closeness to the actual contest.

Proximity and influence are tightly connected: Competitors, as active participants, have the greatest influence on the competition outcome. Support staff contribute indirectly by enhancing the performance of competitors through support. The impact of supporters is typically limited to motivational support and emotional atmosphere. Governing entities are neutral and unbiased actors so the influence on the competition outcome is indirect. Group size refers to the number of individuals within each stakeholder category. Typically, the farther a stakeholder group is from the competitive action and the less direct influence it has, the larger the size of the stakeholder group tends to be. The stakeholder groups and the three dimensions are presented in the table 1 below.

**Table 1.** Stakeholders in sports context (adapted from Haffner et al., 2025)

Stakeholder	Physical proximity	Competition influence	Group size	Examples
<b>Competitors</b>	Highest  Actively participate	Most influential:  Directly affect	Smallest	Athletes, teams
<b>Support staff</b>	High  Provide assistance	Influential:  support performance	Medium	Coaches, medical staff, team management, analysts, physios
<b>Governing entities</b>	High  rule over competition	Influential:  Rules, officiating	Medium	Referees, judges, federations, leagues
<b>Supporters</b>	Lowest (on-site or remote)	Least influential:  Emotional or tangible support	Largest	Fans, spectators

### 2.1.3 Purposes

To characterize sports applications Chembakottu et al. (2023a) conducted a manual assessment of a statistically representative sample of 2058 sports applications collected for their study. Their objective was to construct a three-dimensional categorization

framework based on type of sport, functionality, and type of analytics, providing a structured way to describe the diverse roles and capabilities of sports-related mobile applications. This work complements earlier work by Tubek and Duplaga (2018) who analyzed the most popular applications supporting physical activity and identified the key features in them. Their findings indicated that reminders about training sessions or goal-oriented notifications were the most common feature, appearing in 88% of the applications studied. Other frequently observed functionalities were voice instructions, workout diaries, statistical summaries, and social sharing options. Together, these studies show that while some applications primarily focus on monitoring and analyzing user performance, others place greater emphasis on providing instructional content to guide independent training.

Chembakottu et al. (2023a) determined the type of sport based on the primary sport that each application targets (e.g. tennis, cricket). Applications that did not focus on a single specific sport were labelled as *general*. If the application was unrelated to sports altogether, for example general news application, it was classified as *NA*.

Functionality refers to the primary purpose of the application, such as training, league management or betting. Applications with many functionalities were labelled based on the main functionality. The aim of (Chembakottu et al., 2023b) was to analyse how topics cluster within functionally similar applications, to highlight the themes that are most dominant among competing applications. To support this, the core features of each application were emphasised so that functionality-specific characteristics could be clearly distinguished. By keeping functionalities separate rather than merging them, the researchers were able to reduce analytical noise and ensure a more focused functionality-based analysis.

Chembakottu et al. (2023b) categorize type of analytics used in the application. The analytics type is extracted from the application description text and categorized to predictive or statistical and to unknown when the type can't be inferred and NA if the application does not use any analytics.

As a result of the manual analysis, Chembakottu et al. (2023b) derived a total of 13 different main functionalities for the sports applications described below:

**Betting tips:** The functionality delivers betting advice, including suggestions and predictions of forthcoming and live sporting events

**Training:** This encompasses training programs designed for various sports. *Nike Training Club* is an example of a training application as it offers at-home training and mindfulness routines.

**Live updates:** This feature delivers instant updates to the user device of chosen events with push notifications. *BeSoccer – Soccer Live Score* includes coverage of over 10 000 competitions and delivers alerts on goals, starting lineups, news etc.

**Streaming:** This functionality provides live video streaming of sports events.

**Tools:** The feature utilizes on-device and wearable sensors to capture activity metrics that support athletes. For example, various golf applications like *YamaTrack Mobile*, provide exact distances from tee to green and from the current point of the player to green.

**Tracking:** This functionality collects activity data and turns it into guidance so athletes can monitor and analyse their training. As an example, *myCloudFitness* records improvements and offers tailored tips.

**Betting:** This feature enables betting on sports contests. For instance, *SuperDraft – General Book: Free to Play for Prizes* covers a wide range of sports like tennis, football and basketball and users can place bets on these various sports competitions.

**News:** This feature delivers coverage from journalists on sports. *Onefootball - Soccer News, Scores & Stats* is an example of an application providing news coverage, scores and stats for major competitions worldwide. like Premier League and La Liga.

**Social Network:** This functionality supports social networking with individuals interested in same sports. *TennisPAL* is an application which links tennis enthusiasts around the globe to have conversations about tennis.

**League Management:** The feature provides league management tools and enables users to organize their own leagues follow existing leagues. As an example, *Nizampur Premier League* is an app that lets users track local cricket leagues.

**Radio:** Provides audio streams of sports events. *Pro Baseball Audio* for example supplies scores, schedules and streaming of games live in local radio stations.

**Team Management:** Provides a tool for sports team managers to plan and organize team activities. For example, Soccer Tactics Board supports managers map tactics and monitor development.

**Health Tips:** The feature offers tailored advice on health and injuries for particular sport. For example, Health Diet Foods fitness Help gives medical guidance like diet tips aligned with fitness programmes.

## 2.2 Health & Fitness applications

Mobile health applications are increasingly used as tools for behaviour change. Health & Fitness apps monitor and shape health-related behaviours (Pinem et al., 2024). These applications allow users to track calorie intake and consumption, monitor blood pressure and heart rate, track physical activity and sleeping habits (Buntoro & Kosala, 2019). In addition to these health-focused tools within mHealth taxonomy a distinction is often made for wellness applications which provide support for maintaining healthy habits. Rather than targeting specific disease prevention, these apps aim to foster overall healthy lifestyles (Olla & Shimskey, 2015).

Health & Fitness is the seventh-largest app category on the google Play store, with 74 676 mobile applications (AppBrain, 2025). While both Google Play Store and Apple App Store rely on single, broad Health & Fitness category without more granular labeling, the academic literature offers multiple subcategories that allow for a more detailed classification. These subcategories are introduced below.

A study by Villasana et al. (2020) examined the functionalities of nutrition and physical activity mobile applications available on the Google Play Store. 82 mobile applications related to health, nutrition and physical activity were analysed, and as a result they proposed two taxonomies for the applications, one based on the goals of the mobile applications, and one for the features of the applications.

In the reviewed mobile applications, the functionalities were mapped into six groups: diet, anthropometric measures, social functions, physical activity, medical parameters, and vital signs. Based on the functionalities identified across the analysed applications, they can be grouped into four main categories: Education, Diet and Nutrition, Physical activity, and Health. “Diet and Nutrition” applications primarily focus on promoting healthy eating habits and this was the category with most applications (52%) on the analysed data set. The second most represented category was “Health”, making up 25% of the applications. Applications classified under “Health” address all the themes considered in the study including diet/nutrition, exercise and medical content. The third-most applications were in the “Physical activity” category, comprising 12% of the sample. The “Physical activity” group includes applications for measuring and tracking parameters of physical activity. The “Education” category had the least share, comprising 11% of the applications in the study. Applications classified as “Education” deliver learning content, using concepts derived from published literature or knowledge shared by individuals.

In their article “Toward Automated Categorization of Mobile Health and Fitness Applications,” Xu et al. (2014) introduced an automated categorization framework for health, fitness, and medical apps. They collected 1430 apps from Google Play Store and 62 286 applications from Apple App store which were further categorized into 11 sub-categories they came up after consulting experts of the field and examining the descriptions of the applications. These sub-categories were exercise and fitness, healthy eating and weight loss, reference, women’s health, reminders and alerts, symptom checker, mental health, sexual health, pet health, personal health record, and disease monitoring. “Exercise and fitness” and “healthy eating and weight loss” categories by Xu et al. (2014) are similar to “Diet and Nutrition” and “Physical activity” categories by Villasana et al. (2020), but the “Health” category is further split into more distinct sub-categories like “Mental health” and “Sexual health” in the categorization by Xu et al. (2014).

## 2.3 Artificial Intelligence and generative AI

This subchapter presents an overview of artificial intelligence and generative AI technologies, with a particular emphasis on Large Language Models and AI-assisted programming tools.

### 2.3.1 Overview of AI and generative AI technologies

Generative AI (GenAI) refers to a rapidly developing technology of artificial intelligence built on machine learning models that can produce diverse types of content (Veera & Satya, 2024). Recent advances in artificial intelligence technology have enabled a shift from data-driven discriminative tasks to more creative capabilities. From simple user prompts, GenAI can produce original and realistic outputs, such as text, images and software code across diverse fields using deep generative models (Banh & Strobel, 2023).

Recent advances in AI technology have contributed to the growing popularity of GenAI tools. This trend is reflected in recent statistics from Tilastokeskus (2025), which show a rapid increase in the use of GenAI among the Finnish population. The proportion of citizens aged 16-89 who had used GenAI within the past three months nearly doubled from 23% to 41% in one year. GenAI usage increased across all age groups. The most common use cases were information search, text creation and modification, and creation of images and videos. In 2025, information search was the most frequent use of GenAI

as more than half of 16-34-year-olds and 33% of the overall population reported using it for this purpose. Recent advances in AI technology have contributed to the growing popularity of GenAI tools. This trend is reflected in recent statistics from Tilastokeskus (2025), which show a rapid increase in the use of GenAI among the Finnish population. The proportion of citizens aged 16-89 who had used GenAI within the past three months nearly doubled from 23% to 41% in one year. GenAI usage increased across all age groups. The most common use cases were information search, text creation and modification, and creation of images and videos

According to the Tilastokeskus report (2025), the second most common use case was text creation and enhancement, with more than one-quarter of the population using GenAI for these tasks. Younger groups (16-34-year-old) were the most active users (42%). Additionally, 11% of 25-34-year-olds had used GenAI for image and video creation, while coding remained less common, with only 6% engaging in such activities.

Among 16-24-year-olds, the most common context of use was studies (36%). For people aged 25-54, nearly 40% reported using GenAI in their work. However, in most cases, GenAI use was related to something other than studies or work (Tilastokeskus, 2025).

### 2.3.2 Large Language Models

Research on large language models has increased substantially over the past few years. In 2019, when LLMs were only beginning to attract academic attention, just 15 publications addressed this topic. Significant progress in LLM algorithms took place during the COVID-19 pandemic, when the widespread shift toward online work environments supported faster development and broader integration of AI technologies. Since then, academic interest has grown rapidly, rising from 359 papers published in 2022 to 5092 in 2024. This growth is likely driven by the public release of ChatGPT in late 2022, the emerge of other chat-based LLMs, and their expanding integration in diverse research domains, including medicine and law for instance. Moreover, increased integration of LLMs into research infrastructures and workflows has played a key role in driving the expansion of research dedicated to these technologies. (Cotfas et al., 2025)

Built on natural language processing and Large Language Models, text-based models, especially conversational chatbots have significantly transformed the AI field since the introduction of ChatGPT. These models provide wide range of capabilities including summarization, writing assistance, programming support, language translation and sentiment analysis (Gozalo-Brizuela & Garrido-Merchán, 2023).

In addition to improvements in performance, the cost of using LLMs has decreased dramatically in recent years. For instance, the price of querying a model with GPT-3.5-level performance, MMLU (Massive Multitask Language Understanding) score 64,8, has dropped from 20\$ per million tokens in November 2022 to just 0,07\$ per million tokens by October 2024, a more than 280-fold reduction. Across different tasks, LLM inference prices have fallen between 9 to 900 times per year. These rapidly declining costs have made LLMs increasingly accessible (Maslej et al., 2025).

#### *Core architecture and model types*

The evolution of LLMs has progressed from pre-1990s rule-based and statistical approaches to deep neural methods, eventually leading to the emergence of transformer-

based architectures in the late 2010s. These models now form the foundation of modern LLMs due to their scalability, parallelism, and ability to capture long-range dependencies (Mahmoud Sajjadi Mohammadabadi et al., 2025). Transformer architectures have become central in modern Natural Language Processing (NLP) due to their self-attention mechanism which allow models to efficiently process large texts and capture complex linguistic patterns that are essential in tasks such as translation and text summarization. Models like BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer) and BLOOM, are built on variations of the transformer framework, which leverages encoder-decoder structures to handle input sequences efficiently and produce coherent outputs (Kalra & Sharma, 2025). The variants of transformer architectures used in LLMs include encoder-decoder, causal decoder, prefix decoder, and MoE (Mixture-of-Experts) architecture (Naveed et al., 2025).

### *Decoder-only models*

Decoder-only models play an important role in modern LLM architectures and they are particularly efficient in speech-to-text translation tasks, where they have been shown to outperform encoder-decoder architectures (Huang et al., 2024). Decoder-only models including the GPT series (GPT-2, GPT-3, GPT-4), PaLM, Llama, and DeepSeek-V3 use only the decoder blocks of the Transformer architecture. Their core mechanism is a left-to-right generation process in which text is outputted sequentially, with each new token influenced only by the ones generated previously. This is achieved through masked self-attention, which restricts the model so that when computing a token in location  $I$ , the model receives information only from tokens before  $I$  (Mahmoud Sajjadi Mohammadabadi et al., 2025).

### *Training LLMs*

During pre-training, the model undergoes self-supervised learning on massive collections of unlabeled text to acquire general language structure and world knowledge. This single large-scale training step, often driven by Next Token Prediction (NTP) or Masked Language Modelling (MLM), serves as the core foundation of today's LLMs (Mahmoud Sajjadi Mohammadabadi et al., 2025).

## 2.3.3 Prompt engineering

In the context of Large Language Models, a prompt can be understood as a structured set of instructions that directs how the LLM interprets a task, structures its reasoning and formats its response. The prompt sets the conversational frame where the model should operate by giving rules, defining the context, signaling what information should be prioritized, and what form the generated output should take. These instructions shape the conversation, influencing all subsequent exchanges. The systematic development of these instructions is called prompt engineering. Prompt engineering aims to “program” the LLM to enhance its performance for a specific task (White et al., 2023). Thakur (2024) notes that prompting is an essential mechanism as it enables users to make full use of LLM's versatility without the need for extensive training or relying on specialised training datasets. Thakur (2024) describes the prompting as a form of natural language programming, giving users the ability to guide, influence and control the model's behaviour. In practice, prompt functions like a query where wording and semantics directly shape the model's output. Prompting further provides a way to test and evaluate

LLM's strengths and weaknesses, helping users to explore the model's capabilities and find out new behaviours that emerge through interaction.

The amount of different prompting techniques is huge. Schulhoff et al., (2025) identified 58 text-based prompting techniques categorized into six major groups: Zero-Shot, Few-Shot, Thought Generation, Ensembling, Self-Criticism, and Decomposition.

Earlier work by Zheng et al. (2023) suggested that role prompting enhance the performance of LLMs. Their experiments across three open-source models Llama-2, OPT (Open Pre-trained Transformer), and Flan demonstrated that assigning social roles into system prompts guided them to more precise outputs across 2457 questions and 162 roles. These findings indicated that giving the model a social role within prompt provide additional context which helps the model to generate more accurate answers. However, when the same authors Zheng et al. (2024) later conducted a new experiment for larger set of nine more advanced open-source LLMs such as Llama-3, Qwen2.5 and Mistral, the findings were different. In the extended study with comparable set of 2410 questions and the same set of roles, they found that persona prompts did not reliably improve the model accuracy. Instead, the findings showed that adding personas could even slightly reduce the accuracy of LLMs when compared to neutral prompts without roles. While the researchers observed that selecting the optimal role for each question sometimes led to better results, this effect varied considerably. The newer study showed that while assigning a role to a prompt may occasionally lead to better results, but its impact on newer LLMs is uncertain making role selection a challenging task.

The inconsistent effects of role prompting demonstrated by Zheng et al. (2024) showed that the performance of LLMs sometimes improved but could also be reduced, resulting in lower accuracy. Luz De Araujo et al. (2025) shed light on why this unpredictability occurs. Their work shows that current LLMs are extremely sensitive to minor, irrelevant details, such as persona's name or favourite colour. When such details are added to the persona description, they introduce noise that disrupts the model's reasoning. This helps explain why the social roles used by Zheng et al. (2024) produced unpredictable outcomes. Luz De Araujo et al. (2025) reached a different conclusion when the personas were aligned with the specific task and designed to reflect domain-relevant expertise: in these cases, the model accuracy either improved or at least maintained performance rather than harming it. They tested nine state-of-the-art LLMs, including models like Llama-2 and Qwen2.5, using three criteria: *expertise advantage* (experts should surpass baselines), *robustness* (irrelevant details should correspond to better results), and *fidelity* (higher qualifications should correspond to better results). While expert personas generally met these expectations to a modest degree, the models still showed sensitivity to task-irrelevant attributes, which could lead to significant decreases in output quality.

Extending the studies of the instability of persona prompting Bai et al. (2025) argued that the observed benefits of social roles likely arise from keyword priming instead of genuine expert activation. They introduced the idea of "Domain priming", which simply states the nature of the task, such as informing that "this is a mathematic question", without assigning an expert role. Their experiments evaluated both persona prompting and domain priming across mathematics, psychology, and law using four LLMs (Gemini 2.5 Flash, GPT-4.1, Qwen3 32B, and Llama 3.1 8B). The researchers compared baseline prompts, domain priming, and three types of personas, generic, historical, and modern, and conducted cross-domain tests where personas from one field (e.g. mathematics) were applied to other domains (e.g. law). These cross-domain results showed that mismatched personas often performed similarly to the properly matched ones calling into question the assumption that personas activate genuine domain expertise. They also examined negated

personas where the model was instructed that it was not an expert and the results show that negated personas performed on par or exceeded the positive expert personas. This outcome questions the role-playing hypothesis supported by earlier studies and suggest that domain-specific keywords, rather than personas, are the source of performance gains. Overall, Bai et al. (2025) found that domain priming delivered more reliable and consistent improvements, showing an average gain of 3.8% across all domains, reasoning modes, and models. Both human designed and model-optimized priming approaches outperformed persona strategies, indicating that supplying relevant domain information yield more stable benefits than adopting expert roles.

The impact of format constraints in LLM outputs has been examined by Tam et al. (2024), who explored how structured output generation affects LLM performance. Structured generation, especially JSON (JavaScript Object Notation) and XML (Extensible Markup Language), are widely used in industrial applications to make parsing workflows more reliable. The work of Tam et al. (2024) compared four types of instruction styles: Format-Restricting Instructions (FRI), Constrained decoding with JSON-mode, Natural Language to Format, and standard Natural Language (NL). FRI directs the model to follow a JSON/XML schema through instructions, whereas JSON-mode is a build-in, stricter technique in mainstream models that enforces valid JSON output. They evaluated these formats on both reasoning and classification tasks across several LLMs, including gpt-3.5-turbo, Claude-3-Haiku, Gemini-1.5.Flash, Llama-3-8B-Instruct, and Gemma-2-9B-Instruct. Their results showed that stricter format constraints, especially JSON-mode, generally led to greater performance declines in reasoning tasks, whereas looser format constraints improved reasoning quality. In contrast, for classification tasks, JSON-mode outperformed the other methods. Tam et al. (2024) hypothesize that format restrictions reduce errors in classification tasks by limiting the model’s output space but reduce reasoning performance by preventing the model from producing intermediate “chain-of-thought” steps that typically enhance LLMs reasoning quality. Overall, their findings highlight the importance of balancing structured output needs against reasoning capabilities when designing prompts for LLM applications.

### 2.3.4 Coding agents

Generative AI tools in the coding and software development aim to streamline and automate the creation of program code. The defining criterion for this category of AI tools is that the final output is code. These tools cover several subtypes that convert text prompts into code, websites, software or mobile applications. Less common applications offer designs-to-code or text-to-RPA (Robotic Process Automation), and code to translator functionalities. Since their introduction, tools like GitHub Copilot and ChatGPT have significantly helped developers. By using natural language commands, developers can receive assistance with coding, creating websites and help automate routine tasks like documentation (Gozalo-Brizuela & Garrido-Merchán, 2023). In addition, AI-assisted development tools increasingly provide intelligent in-IDE (Integrated Development Environment) code suggestions and inline autocompletion, further accelerating developer workflows (Zheyuan Cui et al., 2025).

The capabilities of coding tools have advanced rapidly: on the SWE-bench (Software Engineering Benchmark) in 2023, the best AI systems were able to solve only 4,4% of coding problems, but this figure increased to 71,7% by 2024 with the top tool, OpenAI’s o3 model. This rapid increase in performance will create a need for more challenging coding benchmarks from AI researchers (Maslej et al., 2025). Beyond their function as automating repetitive tasks and code creation, AI development agents are increasingly

taking the role of a “collaborative partner” or “virtual coworker” that can meaningfully participate in the development process. Developers are also using them for higher level reasoning activities in the early stages of the development process. These tasks include brainstorming system architecture, examining potential design paths, and formulating developments strategies before writing any code. This transition reflects an integration of AI tools into shaping early-stage decision and problem solving rather than just assisting with code and completions (Banh et al., 2025).

Developed in collaboration with GitHub and OpenAI, GitHub Copilot is trained with extensive public GitHub repository code. This large training data enables it to learn real-world practical coding conventions, patterns, and language-specific techniques used in real software development. GitHub Copilot works within standard development environments, providing intelligent context-aware autocompletion. While developers type code or comments, it interprets surrounding context and provides relevant snippets, documentation, and suggestions. Copilot can fill in code that would normally be typed manually or suggest solutions that would require manual online searching by the developer. This speeds up the development process and can also enhance the code quality by suggesting options that developers might not know about. However, as with any LLM-based tool, Copilot is not error-free. If its suggestions are accepted without careful evaluation, it may lead to incorrect code or reduced overall code quality (Zheyuan Cui et al., 2025). Banh et al., (2025) further highlight this challenge by noting that AI development tools offer efficiency gains by their ability to deliver “contextual knowledge”, significantly reducing the effort required to browse external sources such as Stack Overflow. While this direct access to relevant information reduces search time, these gains come with a price of what the authors call as “underestimate overhead”. Because GenAI tools can hallucinate, they sometimes produce code that seems right but is functionally flawed. This leads to the increase of developers’ time spend on refining the prompts and carefully validating the proposed solutions which can offset or even exceed the initial productivity benefits.

GitHub Copilot shows strong potential in real software development environments. In a twelve-week empirical study by Gonçalves & Gonçalves, (2025), software engineers used Copilot as a pair-programming assistant and evaluated it across the SPACE-M dimensions: satisfaction, performance, activity, communication and collaboration, efficiency and flow, and monetization. The activity dimension was further extended to include specific use cases such as code generation, documentation, adding comments and explaining the code, bug fixing, testing, and optimization. Overall, developers reported particularly high satisfaction with Copilot’s ability to improve efficiency and flow which shows that it accelerated repetitive tasks, reduced the need for online searches, and lowered cognitive load. On a classification scale from 0 to 4, all dimensions, except testing (2) and bug identification (2,8) scored 3 or higher after twelve weeks of use. These findings suggest that Copilot is a promising tool for pair programming, offering meaningful productivity gains. However, further improvements in unit testing support and addressing potential security concerns are still needed.

The findings of Gonçalves & Gonçalves (2025) are further supported by the study of Zheyuan Cui et al. (2025), which also demonstrates Copilot’s effectiveness as a development tool in real-world settings. In their multi-month field experiment conducted across three large companies Microsoft, Accenture, and a Fortune 100 electronics manufacturer, 4,867 software developers used Copilot as part of their daily workflow. The primary outcome of interest was the number of completed pull requests, complemented by secondary measures: number of commits, code quality, number of builds, and build success rate. The results indicate substantial productivity gains:

developers using Copilot produced 26.08% more pull requests, 13.55% more commits, and 38.38% more builds compared to those not using the tool. The benefits were especially pronounced for junior developers, who experienced improvements of 21% to 40%, whereas senior developers saw more modest increases of 7% to 16%. The qualitative findings of Banh et al., (2025) offer insight into this difference by introducing the concept of “developer empowerment”. According to their findings, the junior developers use AI assistants not just for coding, but also for interpreting and explaining legacy code and clarifying new technical concepts as they work. In this sense, the AI-tool acts as an on-demand senior mentor offering explanation and guidance that would otherwise require the time of senior colleague. This mentoring aspect likely contributes to the reason why junior developers appear to benefit more from AI assistance compared to their senior counterparts.

Despite these gains, adoption levels varied considerably. Between 42% and 75% of treated developers adopted Copilot during the study period, while 30%–40% did not try it at all across the participating companies. This suggests that although Copilot offers clear productivity advantages, factors such as developer preference, familiarity with AI tools, or organizational practices may influence adoption rates. While individual developer preferences influence adoption, Banh et al., (2025) identify data privacy and intellectual property (IP) considerations as major deciding factors for adopting AI tools at the organizational level. Companies are worried about sharing proprietary source code with external AI services as they fear that it could be leaked or used in model training. As a result, companies have implemented strict policies banning externally hosted AI tools, even when such tools are shown to offer efficiency gains.

## 2.4 Automation in application categorization

Xu et al. (2014) presented a framework for automated categorization of health, fitness and medical applications in their article “Toward Automated Categorization of Mobile Health and Fitness Applications”. In the study, 1430 Android and 62 286 iOS applications were collected from app stores using a web crawler developed for the purpose. From the dataset, they manually classified 1399 applications to total of 11 categories to use as a reference when evaluating the automated classifier. Google Play and Apple App Store have only medical and health & fitness categories for health-related applications, so Xu et al. (2014) added 11 new sub-categories. These sub-categories were exercise and fitness, healthy eating and weight loss, reference, women’s health, reminders and alerts, symptom checker, mental health, sexual health, pet health, personal health record, and disease monitoring.

For categorizing the applications into new sub-categories, Xu et al. (2014) used natural language processing and machine learning technologies. In the first stage of categorization, the description texts available on the apps stores was processed by doing text segmentation, stop word elimination, illegal English word elimination, word stemming, TF-IDF extraction, and feature selection. The processed description text was then used to classify the applications using three different classification algorithms: linear support vector classifier, NearestCentroid classifier, and Naïve Bayes classifier. linear Support Vector Classification (SVC) performed the best resulting in f1 score of 0.88.

### 3. Research methodology

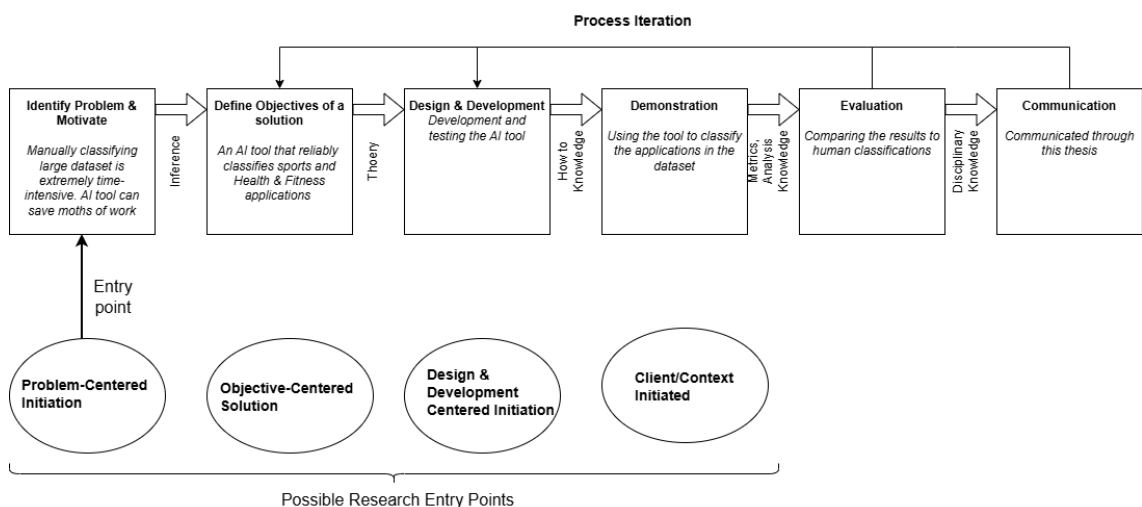
The following chapter provides an overview of the research methodology applied in this thesis. The study follows the Design Science Research (DSR) methodology together with the Design Science Research Process (DSRP) model as its guiding framework. The key principles of the methodology and how the DSRP model was applied in this research are introduced in the section below.

#### 3.1 Design science research

Design Science Research methodology by Hevner et al., (2004) is a problem-solving paradigm that focuses on creation and evaluation of innovative IT (Information Technology) artifacts intended to solve organizational problems. The research must address issues faced that are unsolved. The evaluation of the artifact is at the core of the methodology. DSR requires demonstration of artifacts utility, quality and efficiency with evaluation methods and clear research contributions in a form of artifact itself or from advances in e.g. methods or models. DSR process is an iterative search for effective solutions that involves alternative design solutions and refinements across building and evaluation cycles. Clear communication of results is vital in DSR. Research outcomes must be presented in way that is understandable to both technical experts interested in technical details as well as to managerial audiences who are concerned with how the solution addresses organizational needs.

This study adopted the Design Science Research Process model presented by Peffers et al. (2007). This framework provides a model for the production and presentation of Information Systems research centred artifact creation. The DSRP model is consistent with the design science research methodology by Hevner et al., (2004) further synthesising elements from it into a cohesive process.

Design Science Research Process has six steps which the research followed: (1) Identify problem & Motivate, (2) Define objectives of a solution, (3) Design & Development, (4) Demonstration, (5) Evaluation, and (6) Communication. Figure 1 illustrates the stages of the DSRP model in this research.



**Figure 1.** DSRP model (adapted from Peffers et al. 2007) as implemented in this study

### 3.1.1 Problem identification

Identifying problem and motivating involved clearly defining the challenges with gathering and classifying large volumes of sports and Health & Fitness applications, as well as demonstrating the need for the proposed artifact. This work was commissioned as a part of a research project conducted by the University of Oulu's INTERACT research group, which required a comprehensive dataset for further analysis. For research purposes, it was essential to collect and analyse as many as possible applications from both categories. According to AppBrain (2025) statistics, the Google Play Store alone includes 74 248 applications in the Health & Fitness and 35 737 in the Sports category. Manually searching, screening and collecting a large quantity of application data requires a substantial human effort (Chembakottu et al., 2023a; Kebede et al., 2018). For research purposes, the applications further needed to be classified according to user groups, intended purposes and sport types. Given that the initial dataset contained approximately 50 000 applications, manually reviewing them would have been extremely time-consuming. To ensure that a sufficiently large and representative dataset could have been collected and categorised within a reasonable time frame, automation was essential.

### 3.1.2 Definition of objectives

Definition of objectives of the solution was derived directly from the problem formulation. The primary goal was to develop a software tool capable of efficiently collecting a large number of Sports and Health & Fitness applications and organizing them into a structured format, like an excel sheet, for further analysis. Beyond data collection, the tool needs to achieve high accuracy in classifying applications into predefined categories including user groups, purposes of use, and sport types, as well as identifying irrelevant applications, such as arcade games, in the dataset. In essence, the objective was to create an automated tool that could as closely as possible replicate the quality and correctness of manual human classification while significantly reducing the amount of time and effort required. Achieving these objectives was crucial for enabling large-scale analysis of the Sports and Health & Fitness applications ecosystem within a reasonable time frame.

### 3.1.3 Design and development

Design and development phase focused on constructing the actual artifact that would address the identified problem. The tool was implemented using the Visual Studio Code IDE, combining JavaScript for the data-gathering component and Python for the AI-driven classification module. To streamline and accelerate the development process, GitHub Copilot and OpenAI Codex AI-assisted coding agents were used to enable more efficient code generation and problem-solving during implementation, and ChatGPT-5 was used for brainstorming the structure and functions for the application. For the application connection functionality, the artifact relied on web scraping libraries google-play-scraper and app-store-scraper to extract data from the Google Play Store and Apple App Store. The classification component of the tool was built on OpenAI's GPT-5-mini-API (Application Programming Interface), which was used for categorization and identifying irrelevant applications from the dataset.

### 3.1.4 Demonstration

In the demonstration phase, the functionality of the artifact's classification module was assessed using a set of previously gathered application descriptions, which the tool used to carry out the classifications. This demonstration illustrated how the tool processes application information, assigns each entry into relevant user groups, purposes, and sport types, and filters out irrelevant applications.

### 3.1.5 Evaluation

The evaluation step involved assessing the performance of the AI tool by comparing its classifications with those done manually by human evaluator using the same dataset of applications. This comparison made it possible to determine how accurately and consistently the artifact performed the classification tasks and to assess how close it was to human-level decision making.

### 3.1.6 Communication

Finally, the communication of both the identified problem and the developed artifact is accomplished through this thesis, which documents the full design science research process in detail. In addition, the results and insight generated by this work are part of a future scientific publication submitted to an Information Systems conference.

The DSRP model provides four possible entry points depending on the chosen approach. This study followed a problem-centred approach which began at the first entry point: problem-centred initiation. The work was initiated when we recognized that manually collecting and classifying a large dataset of applications would require significant amount of time and effort. Consequently, the design and development of the AI artifact were directly driven by the need to overcome this challenge.

## 3.2 Literature review

The background section of the thesis was conducted following the literature review methodology. Literature review is a technique for theory building and it connects and interprets wide range of studies spanning different topics (Baumeister & Leary, 1997). Literature review is used to describe prior research to map and assess the research landscape to motivate the aim of the study and justify the research questions (Snyder, 2019). The background section of this thesis was developed through a review of scientific research papers and websites related to the topic. Most of the research papers were identified through searches in the Scopus database using the Scopus AI tool, which generates search terms from natural language queries. Further sources were identified by examining the references of selected papers and by utilizing academic discovery tools including Google Scholar, Keenious, Elicit and Connected papers. Abstracts were reviewed to determine relevance, after which suitable articles were downloaded, organized in the Mendeley Reference Manager, and skimmed to gain an overview of their key contributions.

Initial searches in the Scopus database were conducted using keywords such as "Sports Applications", "Large Language Models", "LLM", and "msport\*". However, many of the papers used in this thesis were identified through natural language searches enabled

by the Scopus AI feature. This feature transforms natural language queries into optimized keyword sets used for database searching. Examples of search queries used in Scopus AI include “Prompt engineering LLMs”, Adoption of GitHub Copilot in software development”, and “Generative artificial intelligence technologies”. Similar functionality was provided by the Elicit research tool, which was used to search topics such as “Taxonomy of Health and Fitness applications”. Additionally, Keenious was employed to identify related studies by analysing pasted text from existing scientific articles and retrieving thematically similar publications. Connected papers is a research tool that produces a visual network of related scientific publications based on a submitted article. It was utilized to discover articles addressing similar topics, as well as publications that cited or were cited by the submitted work.

## 4. Development of the AI tool

This section presents the problem identification and the development of the artefact designed to address the defined problem. The artefact was developed in response to a challenge identified during a research traineeship at the INTERACT research unit at the University of Oulu. The artifact consists of two modules: a data gathering module, and the data classifying module. Development of these modules is presented in the following section.

### 4.1 Problem description

The objective of the traineeship was to produce a comprehensive conceptual map of sports applications and the related research landscape. The position involved identifying and mapping sports applications used by field practitioners—drawing on sources such as Google Play and the App Store—as well as reviewing academic research and systematically collecting and organizing data.

There are total of 113 011 applications in the Sports and Health & Fitness categories on the Google Play Store only (AppBrain.com, 2025). Comparable statistics for the Apple App Store are not publicly available. To produce a comprehensive conceptual map of sports applications, a sufficiently large dataset of relevant applications is required. Given the vast number of applications available, manually collecting an adequate sample of applications for research purposes would be a highly time-consuming task for the researcher. In addition to data collection, the applications would also need to be further classified into user groups, purposes, and sport types, which would significantly increase the overall time and effort beyond collection alone. Automated scraping from the Google Play Store and Apple App Store resulted in a data set of 53 221 applications. Manually reviewing and classifying each of these applications would have taken approximately six months.

The study by Xu et al. (2014) on automated categorization is very similar to our research. However, there are few differences. They used natural language processing and machine learning technologies for the categorization and web crawler for collecting the data, whereas our study uses OpenAI's artificial intelligence tool GPT-5-mini and google-play-scraper and apple-store-scraper. This study is focusing also on sports and Health & Fitness applications, not medical applications. This research will further categorize the applications based on user groups and purposes. The study by Xu et al. (2014) was conducted in 2013 and the number of applications in the Google Play Health & Fitness category has increased from total of 49 084 applications in Health & Fitness and medical categories to 72 248 only in Health & Fitness. (AppBrain, 2025). New technologies that can be utilized in automated categorization of mobile applications such as AI have emerged since 2013 and the suitability is investigated in this research.

### 4.2 Data collection phase

Initially, the data collection started by manually searching Google Play Store and Apple App Store using sports relate keywords like “Golf”, “Hockey” etc. This is a slow process, and the website only returns a limited number of applications without an option to filter out arcade games and other applications not relevant for study purposes. To support large-scale data gathering, the collection of applications was carried out using an automated,

programmatic approach based on web scraping technologies. In particular, the *apple-store-scrapers* and *play-store-scrapers* APIs were used to extract application information from the Apple App Store and Google Play Store. Both APIs offer several methods for retrieving application data, allowing developers to choose the most suitable approach for their needs. The artifact made use of the *list* and *search* methods for data extraction. The *list* method enables the retrieval of applications from predefined store collections. In the Google Play Store, these collections include “Top Free”, “Top Paid”, and “Top Grossing”, while Apple App Store offers additional lists such as “New iOS”, and “Top Free iPad”, and others resulting in a total of 13 collections. All available collections were queried with the genre parameter set to “SPORTS” and “HEALTH\_AND\_FITNESS” from Google Play Store and 6 lists for Apple App Store. The APIs have limits for the number of results returned, allowing up to 200 applications per request from the Apple App Store and up to 500 from the Google Play Store. The *list* method supports additional parameters beyond genre selection, including country and language settings. Using these options, the artifact scraped applications lists from 30 different countries worldwide across both the Google Play Store and Apple App Store expanding the geographic coverage of the collected dataset.

The second approach used for scraping application data was the *search* method. The *search* method retrieves applications based on specified search terms. Like the *list* method, it supports parameters such as country and language in addition to the search term. To capture a wide range of Sports and Health & Fitness themes, the artifact employed 190 different search terms. These search queries were applied to the same 30 countries as the list-based collection. The APIs limit the number of results returned by the search method to a maximum of 200 applications from the Apple App Store and 250 from the Google Play Store. Furthermore, only applications with a primary genre of “SPORTS” or “HEALTH\_AND\_FITNESS” were included in the dataset.

Because applications may appear across several top lists, countries, and be retrieved through multiple search queries, the *search* and *list* methods produced overlapping results. To address this, the artifact identified duplicate applications using their unique *appId* retrieved from the APIs and excluded duplicate entries. The APIs returned application data in JSON format, which was then converted into CSV files to allow easier analysis in spreadsheet software such as Microsoft Excel. The JSON output from the Google Play Store API consisted of 56 key-value pairs describing various application attributes, such as *genre*, *appId*, *title*, *description*, and *score*. In contrast, the Apple App Store API returned a JSON file containing 33 key-value pairs, 17 of which overlapped with those from the Google Play Store, including fields like *appId*, *title*, and *score*.

As the study by Zheyuan Cui et al. (2025) indicates, GitHub Copilot coding assistant has shown gains productivity especially among junior developers. Therefore, coding agents were used in programming the AI tool. Along with GitHub Copilot, OpenAI’s Codex tool was also briefly used while developing the AI tool, and ChatGPT-5 was used for brainstorming and ideating the structure of code for the AI tool. An overview of the application data collection process is shown in Figure 2.

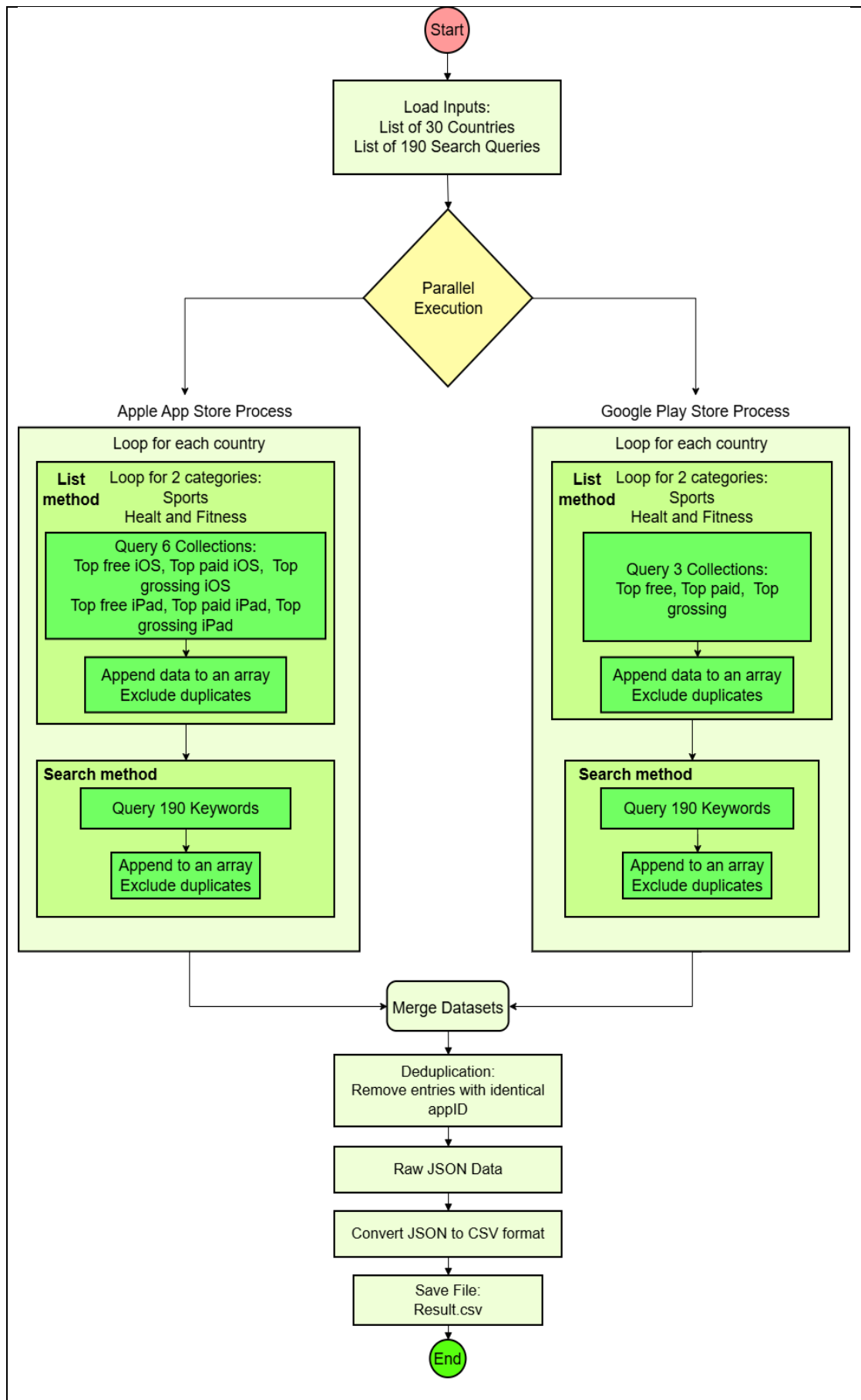


Figure 2. Application data gathering process of the artifact

Challenges and solutions:

**Data Corruption:** The application descriptions often contained long text blocks with line breaks, which caused parsing errors and led to corrupted CSV columns. To address this issue, a script was developed to repair the column leakage problem. However, this approach resulted in truncated descriptions texts, leaving the initial dataset with incomplete application descriptions. This issue was later solved during the classification phase by re-collecting only the description data for each application based on their unique *appId* values.

**API inefficiencies:** The initial scraping strategy involved collecting application data from 118 countries across the globe using 424 distinct search queries. Executing such a large-scale scraping would have required more than a week to complete. The APIs implement throttling mechanisms that limit the number of requests made per second to prevent excessive load on the app store websites. Exceeding these limits may result in temporary IP bans, further slowing the process. To overcome these practical limitations, the number of countries included in the scrape was reduced to 30, focusing on the largest markets, and the number of search queries was limited to 190. With these adjustments, the final data collection process was completed in approximately 50 hours resulting in a raw dataset of 53 331 applications.

### 4.3 Data classifying phase

The data scraping module resulted in a dataset of 53 331 mobile applications. While the collected API data included a wide range of metadata, it did not provide information on targeted user groups (athlete, supporter, support staff, governing entity), specific sport types (such as golf or tennis), or the main purpose of the application (for example training, tracking, or live scores). Manually assigning these properties to over 50 000 applications would have taken an estimated six months. Therefore, an automated solution for the classification task was required.

As numerous LLM-based artificial intelligence services are currently available, an AI-supported solution was considered for automating the classification process. As the price of using LLMs has decreased significantly, employing such models to address the classification task became a viable option. Given the scale of the dataset, which consists of nearly 50 000 applications, cost efficiency was a key consideration.

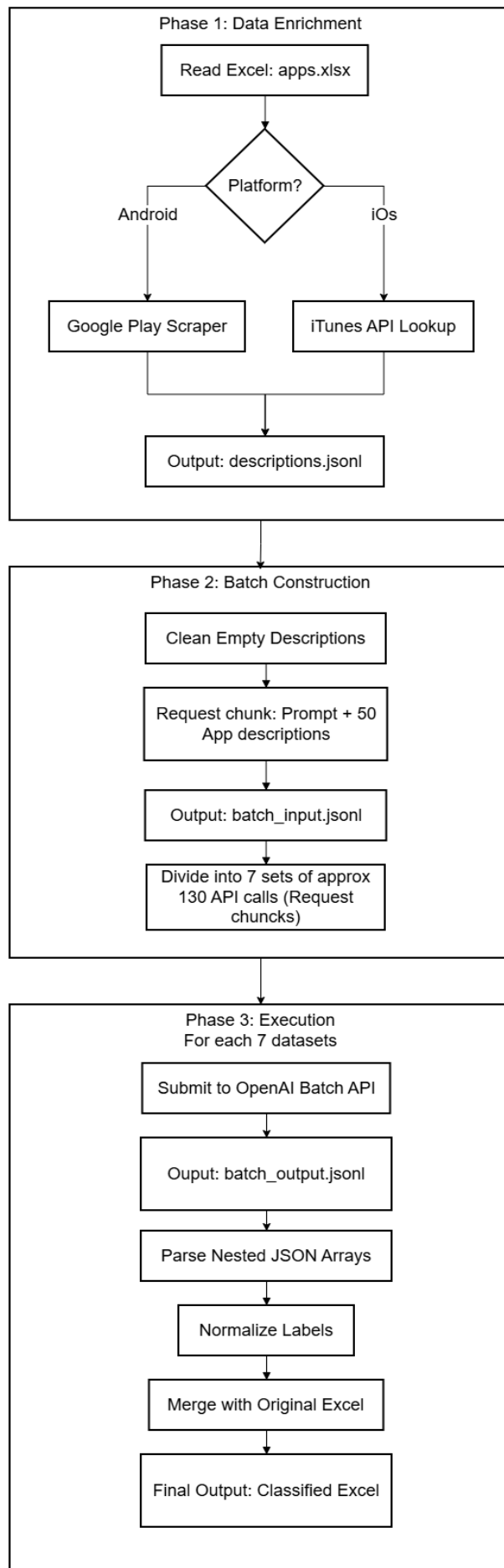
Different AI APIs were examined and compared to determine which option was most suitable for the requirements of this study. According to information provided by OpenAI (2026) the GPT-5 mini-API is a cost-effective implementation of GPT-5 optimized for clearly scoped tasks and precise prompting. The pricing for GPT-5-mini-API is USD 0.25 per million input tokens and USD 2.0 per million output tokens. This model was chosen for the classifications task due to its reasonable balance between cost and performance.

The AI tool was built upon existing classification frameworks, including the tracking and training categories introduced by Tubek & Duplaga (2018). While these categories provide a useful foundation, they alone are not sufficient to describe the full scope of available applications. To address this limitation, the AI tool employs a more granular categorization scheme that better reflects the diversity of application purposes.

The classification component of the artifact was developed as a modular Python application. GitHub Copilot Coding Agent and OpenAI's Codex coding tools were used

to speed up the coding process. Initially, the GPT-5-mini-API was integrated into the artifact's codebase. The API was first tested using individual application descriptions and prompting the model to perform classification tasks. Prompt refinement was conducted incrementally, testing the model's reasoning behavior with one description at a time. Once the model produced consistently accurate classifications for individual descriptions, the accuracy of the API for classifying the applications was further tested using larger sets of application descriptions.

The classification module is organized into three distinct phases: Data preparation, AI processing, and results integration. Because the scraping phase introduced data corruption in the application description fields, the descriptions had to be re-collected prior to performing the classification tasks with the AI. The data preparation process begins by reading the CSV file containing the collected application data and extracting application IDs along with platform information into a new list. The application IDs are then passed to either the google-play-scraper or the Apple iTunes Lookup API to retrieve detailed application descriptions. This step produces a JSONL (JavaScript Object Notation Lines) file containing application IDs, platform identifiers, and application description texts. In the next step, applications with missing description fields are filtered out to prevent redundant API calls to the GPT-5-mini model. For cost efficiency reasons, the classification requests are submitted through a batch API endpoint. The batch API handles the requests within a 24h timeframe and reduces costs by 50% compared to standard synchronous API requests. Since the study did not require immediate responses, the batch API approach was sufficient for the classification task. The JSONL file was then divided into smaller chunks that were ready to be submitted to the batch API.



**Figure 3.** AI classification process

The AI classifier module is built on OpenAI’s GPT-5 mini model and leverages the batch API, which completes processing within 24 hours. To minimize input token consumption, classification requests are sent in batches that consist of a single system prompt with detailed instructions and a set of 50 application descriptions. This approach improves cost efficiency by sending the long instructional prompt only once for each batch of descriptions. The full dataset was divided into seven batches, each consisting of roughly 130 API calls. The complete classification task required total of 868 API calls and processed 19 406 038 tokens. The total cost of the classification was USD 8,49, of which USD 1,92 was spent on input tokens, USD 0,05 on cached input tokens, and USD 6,52 on output tokens.

### **Prompt engineering**

The system prompt was developed iteratively by testing its performance on smaller samples of around 200 applications and evaluating the resulting classifications by comparing them with human classification performed on the same application set. Based on these comparisons, the prompt was adjusted after each iteration until the results reached an acceptable level of accuracy. The final version of the prompt assigned the model the role of “*Specialist in classifying Sports, and Health & Fitness applications*” and restricted the model’s output for each application description to a JSON object containing only the following predefined fields:

- id (string, as input)
- Platform (string, as input)
- Not\_relevant (Boolean true/false)
- Purpose (string)
- Sport\_type (string)
- Athlete (Boolean true/false)
- Supporter (Boolean true/false)
- Support\_staff (Boolean true/false)
- Governing\_entity (Boolean true/false)

The final version of the system prompt contains approximately 200 lines of text, organized into multiple sections. Each section provides detailed guidance on specific aspects of the classification process, including output formatting, global rules, user group classification, purpose and sport type assignment, and the application of tags for Health & Fitness applications. The technique used in this study was a combination of Role prompting and Style prompting, which are sub-types of Zero-prompting.

The system prompt starts by briefly defining the AI’s role as a *specialist in classifying Sports and Health & Fitness applications*, clarifies the type of input the model will receive. A structured prompting technique was employed in which the input was formatted as a JSON structure and the model was explicitly guided to restrict the output to a JSON object to simplify parsing when constructing the final table. The impact of such format constraints has been examined by Tam et al. (2024), who explored how structured output generation affects LLM performance.

Next, the global rules section provides AI with a set of exclusion criteria to identify applications that should be labeled as *not\_relevant*. The dataset contained several types of applications that were considered irrelevant for the purpose of this study, including arcade-style games, generic web browsers, web stores, and gambling games without sports betting features. The prompt also provides guidance for handling edge cases, such as excluding automotive diagnostic or tuning tools while including telemetry and lap-

timing applications. Wellness applications that provide only generic motivational quotes, as well as dating and social networking applications, were also excluded. Similarly, digital sports games without any real-world training plans or performance tracking features were excluded. Furthermore, the prompt contains detailed instructions for assigning the appropriate user group to each application, supported by illustrative examples that demonstrate how various situations should be interpreted during the user group classification process.

For the assignment of user groups for the applications, the system prompt relied on the stakeholder groups introduced by Haffner et al. (2025). These stakeholder groups are used as originally defined, providing a structured and consistent framework for categorizing the intended users of each application. Athletes are defined as individuals who actively engage in sports, including both exercisers and competitive players. Support staff include coaches, physiotherapists, parents, and team administrations who contribute through logistical, training, or organizational support. Governing entities are limited to authoritative roles such as referees, judges, and tournament directors, who are responsible for scheduling, rules, and official scoring. Supporters are defined as fans, spectators, and sponsors. For fantasy sports, betting, and prediction applications, the user group is assigned as supporter by default. In cases where an application serves both athletes and spectators, both user group flags are set to true, capturing cases such as live update applications that are relevant to both groups.

The system prompt then introduces a section that defines the taxonomy for the primary purpose of each application. This study adopted the purpose categories proposed by Chembakottu et al. (2023a) as a baseline for assigning the main purpose to each application in the AI tool. In addition to these pre-existing categories, ten additional purpose tags were introduced to better cover the wide range of Sports and Health & Fitness application functionalities. Furthermore, an *unknown* tag was added to the AI tools repertoire to account for applications for which the main purpose could not be clearly identified based on the available information. In this section of the system prompt, the AI tool is instructed to select exactly one category from a list of 24 predefined labels such as *Betting*, *Booking*, *Training*, *Live\_scores*, and *Team\_management*.

To ensure consistent classification, the prompt provides detailed disambiguation logic for handling common overlaps between purposes. A frequent challenge involved distinguishing between betting, predictions and fantasy sports applications. To address this, the AI tool was instructed to label an application as *Betting* only when real money wagering was involved, *Predictions* when participation was free-to-play, and *Fantasy Sports* to season-long league formats with team management features, even when real money was involved.

Another recurring classification challenge concerned distinguishing between *Tracking*, *Training*, and *Tools*, as these categories often share overlapping features. The AI tool was guided to classify an application as *Tracking* only when the main emphasis was on longitudinal data, such as personal logs, historical records, or progression metrics. The *Training* label was reserved for applications that focus on providing structured instructions, exercises, or drills intended to support skill improvement. Finally, the *Tools* category was applied to applications that function primarily as real-time utilities, including range finders, timers, maps and shot tracers.

Following the general-purpose classification guidelines, the system prompt introduces a list of 18 *domain rules*. These domain rules were introduced to systematically address classification difficulties that repeatedly occurred when determining an application's

primary purpose. Each rule offers precise instructions that guide the AI tool on how to handle such cases consistently. One example concerns school athletics administration applications, which are particularly common in the dataset. For these applications, the prompt guides the AI tool to assign *Team management* as the application's purpose, setting both *Athlete* and *Support staff* to true. This reflects their typical use by both students, or athletes and coaching staff, for activities such as team communication, scheduling, and enrollment in matches and training sessions.

The final section of the prompt focuses on how the sport type should be assigned to each application. The prompt includes a predefined list of commonly used tags that serve as a default option when determining the sport type. If none of the predefined tags accurately describe an application, the AI is permitted to generate a new sport-type tag. The tag *Various* is applied when an application covers multiple distinct sports or leagues, or when the specific sport is not clearly defined. When the sport type cannot be identified at all, the *UNKNOWN* tag is applied. In the case of motorsports applications, the AI is guided to set a tag of a single series or discipline if the app is dedicated to that purpose; otherwise, the general *Motorsports* tag is applied. The AI tool utilized a dedicated set of tags to classify non-sport health contexts within Health & Fitness applications, as they may not correspond to any specific sport and can instead relate to general well-being, menstrual cycle monitoring, or nutrition. These tags were primarily derived from the categories introduced by Villasana et al. (2020) and Xu et al. (2014), with few supplementary additions to better capture specific health-related contexts. If the primary content involves workouts, training plans, or tutorials, tags such as *Fitness\_gym* or specific modalities like *Yoga* or *Pilates* are assigned. The tool is explicitly instructed not to invent new tags for Health & Fitness applications, and to use *UNKNOWN* or *General\_Health* when appropriate. rather than applying the sport type tags used for sports-focused apps.

The system prompt concludes with a validation stage. In this section, the AI tool is instructed to maintain the original order of the input data and to output a strictly formatted JSON object only. The prompt explicitly prohibits any additional keys, comments, or textual output outside the defined JSON structure, ensuring that the output can be reliably parsed without further processing.

## 5. Evaluation

This section outlines the evaluation of the AI tool. It first presents the ISO 25010 software product quality model which provides the framework for assessing the artifact and subsequently describes the evaluation of the artifact against the selected attributes.

### 5.1 Evaluation criteria

The evaluation of the AI tool is based on the ISO/IEC 25010 product quality model, which serves as a comprehensive framework for assessing software product quality. The model provides guidance on which quality attributes should be examined when evaluating a software product. Quality refers to how well the system satisfies stakeholder needs and delivers value. These stakeholder needs, such as functionality, performance, compatibility, and reliability are systematically reflected in the quality model, which organizes product quality into a hierarchy of nine characteristics and their sub-characteristics shown in figure 4 below (International Organization for Standardization [ISO], 2023).

SOFTWARE PRODUCT QUALITY								
Functional suitability	Performance efficiency	Compatibility	Interaction capability	Reliability	Security	Maintainability	Flexibility	Safety
Functional completeness	Time behaviour	Co-existence	Appropriateness recognizability	Faultlessness	Confidentiality	Modularity	Adaptability	Operational constraint
Functional correctness	Resource utilization	Interoperability	Learnability	Availability	Integrity	Reusability	Scalability	Risk identification
Functional appropriateness	Capacity		Operability	Fault tolerance	Non- repudiation	Analysability	Installability	Fail safe
			User error protection	Recoverability	Accountability	Modifiability	Replaceability	Hazard warning
			User engagement		Authenticity	Testability		Safe integration
			Inclusivity		Resistance			
			User assistance					
			Self- descriptiveness					

**Figure 4.** The product quality model (adapted from ISO, 2023)

### 5.2 Functional suitability

The quality characteristic most relevant to the evaluation of the AI tool is Functional suitability. According to (International Organization for Standardization, 2023) the functional suitability

*“Represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions.” (International Organization for Standardization [ISO], 2023).*

The functional suitability consists of the following sub-categories: Functional completeness, functional correctness, and functional appropriateness.

## 5.2.1 Functional completeness

Functional completeness represents the

*“Degree to which the set of functions covers all the specified tasks and intended user objectives.”(International Organization for Standardization [ISO], 2023)*

The assessment of functional completeness was therefore conducted by comparing the implemented functionality of the AI tool against its stated requirements.

The objective of the traineeship was to create a comprehensive conceptual map of sports and Health & Fitness applications. An AI tool was designed and developed specifically to support this task and to align with the overall goals of the traineeship. The position focused on identifying and mapping sports applications used by field practitioners, using platforms such as Google Play Store and the Apple App Store as primary data sources. In order to ensure a representative dataset, it was necessary to collect as many relevant applications as possible. Once collected, each application was classified into categories such as purpose, sport type and intended user group. Consequently, the AI tool was required to support the collection of large volumes of applications, potentially tens of thousands, and perform accurate and reliable classifications.

The AI tool successfully collected and categorized a total of 53 331 individual applications from Google Play and Apple App Store. This represents a significant increase in scale when compared to previous studies, such as Chembakottu et al. (2023a) who analyzed 2621 applications, and Kebede et al. (2018) who examined 6018 applications. The substantially larger dataset obtained in this study demonstrates that the AI tool effectively fulfilled the requirement of gathering a broad set of applications and showed a good level of functional completeness as well as functional appropriateness.

## 5.2.2 Functional correctness

Functional correctness is defined as the

*“Degree to which a product or system provides accurate results when used by intended users.”(International Organization for Standardization [ISO], 2023)*

To assess this quality characteristic, the accuracy of the AI tool was measured by comparing its classification results with those produced by a human evaluator. The AI tool performed classification across seven parameters; relevance, purpose, sport type, and four user type indicators set as either true or false (athlete, supporter, support staff, and governing entity). For evaluation purposes, a single application classification was deemed correct only when all seven parameters exactly matched the corresponding human classification.

Duplicate entries referring to the same application across different platforms (iOS and Android) were excluded from the test dataset. These duplicates were removed because their classification results were effectively identical and retaining them would not have contributed meaningful additional insights to the evaluation process. Some applications were no longer available in the app stores at the time of evaluation and therefore could not be compared. As a result, these applications were excluded from the final analysis.

The final evaluation dataset consisted of 145 applications, which were used for assessing the performance of the AI tool. One application was not classified by the AI tool, potentially due to the application description being temporarily unavailable on the source website at the time of data retrieval. In the seventh and final iteration, the system prompt produced a complete match with human classification for 133 out of 144 applications. Beyond assessing exact matches with human classification, the accuracy of the AI tool was further analyzed across each of the seven classification parameters individually against the corresponding human-assigned classifications. The number of correct and incorrect classifications for each classification parameter as well as the percentage of correct classification are represented in table 2 below.

**Table 2.** Results of the Classification Evaluation

	Relevance	Purpose	Sport type	Athlete	Supporter	Support staff	Governing entity
Correct	141	131	136	135	135	134	137
Incorrect	3	6	1	2	2	3	0
Correct %	97,92	95,62	99,27	98,54	98,54	97,81	100

When evaluating the relevance of applications to the study, the AI tool produced the same assessments as the human evaluator for 141 out of 144 applications. Within the test dataset, five applications were marked as *not relevant* by the human classifier, of which the AI tool correctly identified four. However, the AI tool classified two sports card scanner applications, LUDEX Sports Card Scanner and CollX: Sports Card Scanner, as not relevant, whereas the human evaluator considered them relevant. These applications allow users to scan sports cards, create digital collections, and monitor market values. These features position the applications within the *fan engagement* purpose category, making them relevant for sports application research. In the system prompt, the AI tool was explicitly guided to classify OBD (On-Board Diagnostics) gauge applications as not relevant. Despite this guidance, it failed to label the application *DashCommand – OBD-II Gauges*, as not relevant, in contrast to the assessment made by the human evaluator.

The classification of application purposes resulted in the greatest number of inconsistencies between the AI tool and the human evaluator. Out of 137 applications for which a purpose was assigned, 131 classifications aligned with the human evaluation, while seven applications were categorized under a different purpose. The following list presents the specific applications where the classification errors occurred:

Playmetrics – This is a team management application designed to track player attendance, manage game schedules and scores, facilitate club-wide communications, and more. The AI tool labeled it as a *League management* application. However, the application is not intended for organizing entire leagues; it is designed for individual teams and is primarily used by team managers and players, making *Team management* the more accurate classification.

MaxPreps – This application provides updates on high school sports teams, including news, schedules, statistics, and more. The AI tool mistakenly classified it as a *Team management* application. However, the application is not designed to organize or manage teams. Instead, the main functionality of MaxPreps is to deliver timely updates and

information about high school sports teams, making *Live updates* or *News* a more accurate classification.

Chalkboard DFS & Social Picks – This is a betting platform where real money is involved. The AI tool incorrectly classified it as a *Fantasy sports* application. The application is not intended for fantasy team management, but its focus is on real money betting. The inclusion of the phrase “chalkboard daily fantasy” in the app description, combined with the description’s short length, likely caused the AI tool to misinterpret the applications’ true purpose.

PGA TOUR – This application provides real-time leaderboards, scorecards, tee times and additional information on PGA (Professional Golfers’ Association) TOUR events. It also offers access to live video and audio coverage of these events. The AI tool mistakenly classified it under the category *streaming*. While streaming is one feature of the application, its primary purpose is to provide up-to-date information and real-time updates of PGA tour events. Therefore, a more accurate classification for this application would be *live updates*.

PractiScore Competitor – This application serves competitors in shooting contests by enabling them to compare match results and analyze the performance of both themselves and other participants. The AI tool classified it as *tracking*, but this does not fully capture its primary purpose. The main function of the application is to provide match results, making *live updates* a more suitable label.

FEI EquiTests 2 – Eventing - This application provides equestrians with an opportunity to familiarize themselves with FEI (Fédération Equestre Internationale) Eventing dressage tests. It is primarily used by riders to prepare for upcoming tests. The AI tool classified it under *tracking*, but this is not accurate. Because the application serves as a practice tool rather than tracking performance metrics, it should be classified as *tool*.

When assigning the sport types to the applications, the AI tool and human evaluator showed a high level of agreement, with only one mismatch out of 137 applications. This discrepancy occurred in the classification of the application QZ – qdomyos-zwift, which is designed to connect a exercise bikes and treadmills to a smartphone. The AI tool categorized the sport type as various a classification that is partially justified since the application supports multiple endurance-based training machines. However, the system prompt includes a specific label for *endurance sports*, which offers a more accurate and specific description of its sport type.

The categorization of user groups involved labeling four predefined user groups as either true or false. When evaluating the *athlete* user group, there were two discrepancies between the AI tool and the human evaluator out of 137 applications. The first mismatch involved the application CheerCast, which offers event-related information specifically for cheerleaders. The AI tool incorrectly labeled the *athlete* user group as *false*, even though the application is intended for athletes who participate in cheerleading events. The very short description text provided for the application likely contributed to the AI tool’s ability to make an accurate classification. The second discrepancy occurred with the application Assistant Coach Volleyball, which is clearly designed for volleyball coaches, rather than players. In this case, the AI tool incorrectly labeled the *athlete* user group as *true*, which is incorrect.

Assigning true or false values to the *supporter* user group resulted in two mismatches out of 137 classifications. One of these mismatches involved the previously discussed

CheerCast application. While CheerCast is designed for cheerleading athletes rather than supporters, the AI tool incorrectly labeled the *supporter* user group as true. The second discrepancy occurred with the application Rank One, which explicitly states in its description that it serves school users, parents, students and fans by providing access to school athletic information. Despite this clear indication, the AI tool incorrectly set the *supporter* user group to false.

Labeling the *support staff* user group as either true or false resulted in four mismatches between the AI tool and the human evaluator. One of these discrepancies occurred with the application VolleyWrite Buddy, which is intended for scorekeepers to track and record scores in volleyball matches. Scorekeepers are part of the *governing entity* user group, not the *support staff* category. However, the AI tool incorrectly set the *support staff* label to true for this application. This classification is inaccurate, as support staff include roles such as coaches, physiotherapists, and other team personnel not individuals responsible for official scorekeeping. Another incorrect classification related to *support staff* user group involved the application Softball Stats Tracker Pro. This application is designed to track softball statistics including batting, catching and fielding performance. The description text clearly indicates that the application is designed for coaches. Despite this, the AI tool incorrectly set the *support staff* label to false. Since coaches clearly fall within the *support staff* category, this classification is inaccurate. The final application that was incorrectly classified under *support staff* user group was Swipe Play Clock, an Apple Watch application that provides game clock functionality for American football officials. Since officials are part of the *governing entity* user group, this application should not be associated with *support staff*. Nevertheless, the AI tool incorrectly assigned a true value to the *support staff* label, leading to a clear misclassification.

The AI tool's assignments for the *governing entity* user group were fully aligned with those of the human evaluator. Out of the applications included in the test set, seven were classified as targeting governing entities by the human evaluator, and the AI tool successfully identified each of these cases. Additionally, it accurately marked the governing entity user group as false for the remaining applications. This result reflects both the reliability of the classification and the relatively limited presence of governing entities as a primary user group in the dataset.

The AI tool generated classifications that closely aligned with the human evaluator across seven distinct parameters, achieving agreement rates ranging from 95 to 100 percent. These results indicate a high level of functional correctness, suggesting that the tool can classify applications accurately and consistently in most cases.

### 5.3 Performance efficiency

Another relevant quality characteristic is performance efficiency, especially time behavior and resource utilization. Performance efficiency represents the

*“Degree to which a product performs its functions within specified time and throughput parameters and is efficient in the use of resources under specific conditions.” (International Organization for Standardization [ISO], 2023)*

The time behavior represents the

*”Degree to which the response time and throughput rates of a product or system, when performing its functions, meet requirements.”  
(International Organization for Standardization [ISO], 2023)*

The data collection phase took approximately 50 hours to complete and resulted in the retrieval of more than 50 000 applications. The APIs used during this phase enforce throttling mechanisms that limit the number of requests per second to prevent excessive load on the app store websites. As a result, the data collection process cannot be significantly accelerated when relying on these APIs. Exceeding the imposed limits may lead to temporary IP bans, which would further slowdown the process. Although a completion time of 50 hours may not be considered fast, it represents only a fraction of the time that would have been required to manually search for and record applications from Google Play and the Apple App store.

The classification phase utilizes the batch API to reduce costs, with each batch request designed to be completed within a 24-hour time window. In practice, each batch was processed in approximately 30 minutes. A total of seven batches were required due to API limitations on the maximum size of data that can be submitted in a single batch. As a result, the full classification of the application dataset was completed in roughly three and half hours. In comparison, it was estimated that performing the same classification manually would have taken approximately six months of work.

The Resource utilization represents the

*“Degree to which the amounts and types of resources used by a product or system, when performing its functions, meet requirements.”  
(International Organization for Standardization [ISO], 2023)*

The completion of the classification for the whole dataset of over 53 000 mobile applications required a total of 868 API calls and processed 19 406 038 tokens. The total cost of the classification was USD 8,49. When comparing the automated collection and classification process to a fully manual approach, the results demonstrate a good level of performance efficiency in terms of time behavior and resource utilization.

## 5.4 Interaction capability

Interaction capability is defined as the

*“Capability of a product to be interacted with by specified users to exchange information between a user and a system via the user interface to complete the intended task.” (International Organization for Standardization [ISO], 2023)*

This characteristic is composed of multiple sub-characteristics including Learnability, User assistance, and Operability that determine the overall usability of the system. Learnability represents the

*“Degree to which the functions of a product or system can be learnt to be used by specific users within a specified amount of time.”  
(International Organization for Standardization [ISO], 2023)*

Because the AI tool does not offer a user interface, learning to use the system requires time and technical expertise. In particular, the need for programming knowledge required for direct interaction with the system code which also negatively affects another sub-characteristic User assistance which refers to the

*“Degree to which a product can be used by people with the widest range of characteristics and capabilities to achieve specific goals in a specific context of use.” (International Organization for Standardization [ISO], 2023)*

Furthermore, Operability represents the

*“Degree to which a product or system has attributes that make it easy to operate and control.” (ISO, 2023)*

The AI tool lacks such attributes, as it is controlled solely through code modification. Overall, the Interaction Capability of the AI tool is relatively low, as its effective use is limited to users with sufficient programming expertise.

## 6. Discussion

In this thesis, an AI tool was developed to automate the collection and classification of sports and Health & Fitness applications. The goal of this work was to explore the potential of utilizing LLM technologies to accelerate and support the data organization process within Sport HCI research.

### 6.1 Answers to research questions

This study sought to address the research question “*What types of sports and Health & Fitness applications can be found on the market?*”. After cleaning the dataset by removing irrelevant and non-existing applications, a total of 46 862 applications remained, comprising of 21 668 sports applications and 25 194 Health & Fitness applications. The AI tool identified a total of 200 sport types across the sports application’s dataset. Excluding the various category, the most common sport types were football (3 268 applications), golf (1 848), and padel (1 164). Within the Fitness & Health category, fitness and gym applications dominated with 9 327 applications, followed by general health (2 147), and mental wellbeing (1 928).

Another objective of this study was to address the research question “*How reliably can an AI software tool screen the eligibility and categorize sports and Health & Fitness applications into sport types, purposes and user groups?*”. As outlined in section 5, the AI tool successfully screened application eligibility and categorized sports and Health & Fitness applications across these dimensions. When evaluating application relevance, the AI tool produced the same assessments as the human evaluator for 141 out of 144 applications resulting in a high similarity rate of 97,92%. Evaluating the purpose of the applications proved to be the most challenging task for the AI tool, resulting in a similarity rate of 95,62% compared to the human evaluator. The assignment of purpose is partly subjective by nature and the applications with very brief description texts were particularly difficult for the AI tool to classify accurately across the defined dimensions. Assigning user groups to applications resulted in accuracy rates ranging from 97,81% to 100%, indicating a high level of reliability for this task. Similarly, the assignment of sport types was handled effectively, with the AI tool reaching an accuracy rate of 99,27% in comparison to human evaluator.

The AI tool was further evaluated in terms of its interaction capability. As the AI tool does not include a user interface, its interaction capability is limited, since effective use requires a substantial learning effort and familiarity with programming principles. Consequently, further development efforts should focus on implementing a user interface that would support more efficient and user-friendly searching and data collection.

This study extends the research on automated mobile application categorization introduced by Xu et al. (2014) by leveraging recent LLM technologies for the classification task. In their study, Xu et al. (2014) employed natural language processing and machine learning methods, which required preprocessing of application description texts. This included text segmentation, stop word elimination, removal of illegal English words, word stemming, TF-IDF extraction, and feature selection. The processed application descriptions were then classified using three algorithms: linear support vector classifier, NearestCentroid classifier, and Naïve Bayes classifier. In contrast, the use of GPT-5-mini-API in this study significantly simplifies the classification process.

Programming effort is primarily required for setting up the API and collecting the application data, while the classification performance can be refined through prompt engineering using natural language.

This study adopted a prompting technique that explicitly constrained the model's output to a JSON object to facilitate reliable parsing during the construction of the final dataset. Prior work by Tam et al. (2024) has shown that such output constraints can negatively affect reasoning tasks performance, but that classification tasks benefit from strict formatting, particularly when using JSON-mode, a build-in feature provided by mainstream language models.

## 6.2 Contributions

Wobbrock and Kientz (2016) have presented seven research contribution types in HCI and the contributions made in this thesis are reported based on them. The seven contribution types are Empirical Research, Artifact Methodological, Theoretical, Dataset, Survey, and Opinion contributions. This thesis makes an artifact contribution to the field by introducing an AI-based tool that supports Sport HCI research through the automated collection and organization of mobile application data. The tool enables researchers to efficiently compile extensive datasets of Sports and Health & Fitness mobile applications, significantly reducing the manual effort and expenses associated with large-scale data gathering. In addition to data collection, the AI tool is capable of organizing Sports and Health & Fitness mobile applications into meaningful categories such as purpose, sport type, and main user group. The proposed AI tool can perform this classification with a high level of accuracy comparable to human evaluators with a fraction of cost and thus streamlines the research process and enhances the scalability of Sports HCI research.

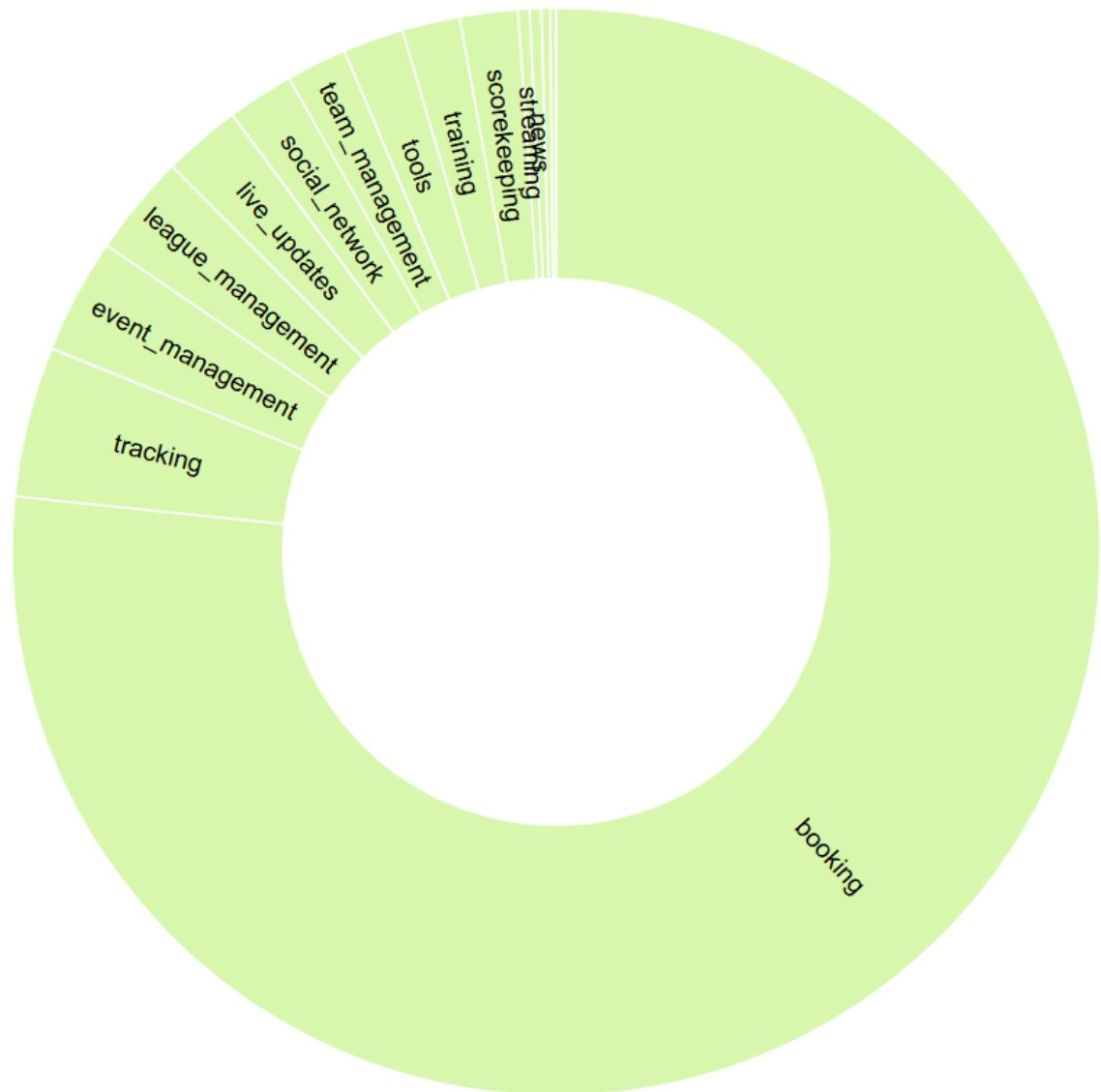
Beyond the tool itself, the study makes a dataset contribution as it delivers a large-scale dataset consisting of more than 46 000 Sports and Health & Fitness-related mobile applications, systematically organized according to sport types, purposes, and intended user groups. This dataset offers a valuable resource for Sport HCI research, enabling researchers to analyse and better understand the ecosystem of Sports and Health & Fitness applications. The dataset shed a light on how the different sport types, purposes and user groups are presented in the Sports and Health & Fitness applications field. Figure 5 below illustrates the distribution of sport types within both categories.



**Figure 5.** Conceptual Map of Sports and Health & Fitness Applications (CC BY 4.0 Juhani Karjalainen)

Beyond serving as a conceptual map of the Sports and Health & Fitness application ecosystem and its classification into sport types, the dataset facilitates more in-depth analysis. In particular, the dataset supports the examination of how different sport types are distributed within the Sports category (Figure 6) and reveals the proportion of primary application purposes associated with each sport type. This is illustrated in Figure 7, which presents the sport type panel and the distribution of application purposes within that category.





**Figure 7.** Conceptual Map of Sports and Health & Fitness Applications. Division of application purposes in padel applications (CC BY 4.0 Juhani Karjalainen)

This study offers a methodological contribution by proposing a new approach for the collection and processing of large-scale mobile application datasets. This approach is particularly beneficial for researcher working in this area, especially within the Sport HCI field, as it offers a relatively fast and cost-efficient alternative for manual work. The proposed artifact managed to do a task that would have taken approximately half a year to do manually with a fraction of a cost. This thesis extends prior research on artificial intelligence and large language model technologies by assessing their suitability for systematic research purposes. The results demonstrate that these technologies can significantly support and streamline research workflows in the mobile application research domain.

### 6.3 Limitations and future research

The scope of this research was restricted only to applications available on Sports, and Health & Fitness categories in Google Play Store and Apple App Store. Future research could explore the suitability of similar AI-driven tools in collecting and organizing mobile application data across other domains beyond Sports and Health & Fitness, such as

Education, Lifestyle, and Business. Extending this approach to additional categories would support the generalization of AI-driven tools for large-scale application data collection and organization in the broader mobile application ecosystem. Application data collection part was performed using the google-play-scraper and apple-store-scraper APIs. Consequently, the overall operation of the application data collection phase of the AI tool relies heavily on the availability and proper functioning of these publicly accessible APIs.

Due to API-related constraints, such as rate limits and restrictions on the maximum number of applications returned per request, the number of applications that could be collected within a reasonable time frame was limited. According to AppBrain.com (2025) there are a total of 78 025 applications in Health & Fitness category and 37 462 applications in the Sports category on the Google Play Store alone, resulting in 115 487 potentially relevant applications. In this study, the AI tool collected 53 221 applications from both Google Play Store and Apple App Store. The total number of applications could be increased in future work by leveraging API functionality that retrieves similar applications based on a given app ID or adding more search terms and countries to search from. Future work could also explore alternative automated data collection methods beyond these publicly available APIs to reduce the AI tool's dependency on these services.

The AI classifier utilized the OpenAI's GPT-5-mini model, which is less advanced compared to higher-tier models such as GPT-5.2 or GPT-5.2 pro, while at the same time being more expensive than smaller, less capable models such as GPT-5 nano, or GPT-4o-mini. Future research could compare different options of large language model technology APIs across various providers to offer insights into performance differences, cost-accuracy trade-offs, and optimal selection for large-scale application classification. Future research could further investigate the differences in classification performance between different prompting techniques. As an example, the JSON-mode and Format-Restricting Instructions could be compared to better understand how classification accuracy differs between them in large-scale application categorization tasks. This research is serving as the basis for a publication to a conference in the Human Computer Interaction domain.

## 7. Conclusion

In this thesis, a software tool utilizing generative artificial intelligence technology was developed and evaluated for its suitability in supporting Sports HCI research by automating the collection and classification of mobile application data. The process of manually gathering and categorizing large volumes of application data is highly time-consuming and labor-intensive. By automating these tasks, the proposed tool has the potential to improve research efficiency in the Sports HCI domain.

This research followed the Design Science Research methodology to investigate the use of an AI-based software tool for collecting Sports and Health & Fitness application data, assessing the relevance of the applications to the study, and classifying them into sport types, purposes and user groups.

The development of the artifact consisted of two distinct phases: data collection and data categorization. The data collection component relied on publicly available APIs and produced a dataset of over 50 000 applications. Automated classification was performed using OpenAI's GPT-5-mini-API. The evaluation of the AI tool was based on selected attributes of ISO 25010 software product quality model. The AI tool demonstrated a high level of correctness in classifying Sports and Health & Fitness applications according to their purpose, sport type, and primary user groups. When benchmarked against human-generated classifications, the AI tool's results aligned with human evaluations in 95,62% to 100% of cases across the assessed parameters indicating strong reliability. Following automated classification and the removal of irrelevant applications identified by the AI tool, the final dataset consisted of 46 862 Sports, and Health & Fitness mobile applications ready for Sport HCI research indicating good level of functional completeness. The data collection phase required approximately 50 hours, while the classification phase took about three and a half hours. The total cost of the classification using GPT-5-mini-API was USD 8,49. Compared to manual approach, which was estimated to require approximately half a year of work, the AI tool demonstrated a high level of performance efficiency in terms of time behavior and resource utilization. These findings suggest that Sport HCI research can benefit from the use of GenAI tools for collecting and classifying large-scale application data.

This thesis made an artifact contribution to Sports HCI research through the development of an AI-based software tool called AI tool for Collecting and Classifying Sports Applications (AICCSA) and contributed to existing LLM research by assessing the applicability of such technologies in a research context. The scope of the study was limited to Sports and Health & Fitness applications, and the data gathering process relied on publicly available APIs. Application classification was performed using the GPT-5-mini model, which is not among the most advanced models currently on the market. Future research could explore alternative data collecting methods, compare the performance of different LLMs for similar classification tasks, and examine how various prompting techniques influence model performance.

## References

- AppBrain.com. (2025). *AppBrain.com*. <https://www.appbrain.com/stats/android-market-app-categories>
- Bai, X., Holtzman, A., & Tan, C. (2025). “*You are a brilliant mathematician*” Does Not Make LLMs Act Like One.
- Banh, L., Holldack, F., & Strobel, G. (2025). Copiloting the future: How generative AI transforms Software Engineering. *Information and Software Technology*, 183. <https://doi.org/10.1016/j.infsof.2025.107751>
- Banh, L., & Strobel, G. (2023). Generative artificial intelligence. *Electronic Markets*, 33(1). <https://doi.org/10.1007/s12525-023-00680-1>
- Baumeister, R. F., & Leary, M. R. (1997). Writing Narrative Literature Reviews. In *Review of General Psychology* (Vol. 1, Number 3).
- Buntoro, I. K., & Kosala, R. (2019). *Experimentation of Gamification for Health and Fitness Mobile Application*.
- Chembakottu, B., Li, H., & Khomh, F. (2023a). A large-scale exploratory study of android sports apps in the google play store. *Information and Software Technology*, 164, 107321. <https://doi.org/10.1016/J.INFSOF.2023.107321>
- Chembakottu, B., Li, H., & Khomh, F. (2023b). *A Large-Scale Exploratory Study of Android Sports Apps in the Google Play Store*. <https://doi.org/10.1016/j.infsof.2023.107321>
- Cotfas, L. A., Sandu, A., Delcea, C., Diaconu, P., Frasinianu, C., & Stanescu, A. (2025). From Transformers to ChatGPT: An Analysis of Large Language Models Research. *IEEE Access*, 13, 146889–146931. <https://doi.org/10.1109/ACCESS.2025.3600739>
- Goebeler, L., Standaert, W., & Xiao, X. (2021). *Hybrid Sport Configurations: The Intertwining of the Physical and the Digital*. <https://hdl.handle.net/10125/71328>
- Gonçalves, C. A., & Gonçalves, C. T. (2025). Assessment on the Effectiveness of GitHub Copilot as a Code Assistance Tool: An Empirical Study. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 14969 LNAI, 27–38. [https://doi.org/10.1007/978-3-031-73503-5\\_3](https://doi.org/10.1007/978-3-031-73503-5_3)
- Gozalo-Brizuela, R., & Garrido-Merchán, E. C. (2023). *A survey of Generative AI Applications*. <http://arxiv.org/abs/2306.02781>
- Haffner, L., Oshri, I., & Kotlarsky, J. (2025). Directions for future IS research on sports digitalisation: A stakeholder perspective. In *Journal of Strategic Information Systems* (Vol. 34, Number 2). Elsevier B.V. <https://doi.org/10.1016/j.jsis.2025.101905>
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). DESIGN SCIENCE IN INFORMATION SYSTEMS RESEARCH 1. In *Design Science in IS Research MIS Quarterly* (Vol. 28, Number 1).

- Huang, C.-W., Lu, H., Gong, H., Inaguma, H., Kulikov, I., Mavlyutov, R., & Popuri, S. (2024). *Investigating Decoder-only Large Language Models for Speech-to-text Translation*. <http://arxiv.org/abs/2407.03169>
- International Organization for Standardization. (2023). *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model ISO/IEC 25010*.
- Kalra, M., & Sharma, V. (2025). Impact of Low Rank Adaptation with Parameter Efficient Fine-Tuning Techniques on Transformer Models. *2025 6th International Conference for Emerging Technology, INCET 2025*. <https://doi.org/10.1109/INCET64471.2025.11139894>
- Kebede, M., Steenbock, B., Helmer, S. M., Sill, J., Möllers, T., & Pischke, C. R. (2018). Identifying evidence-informed physical activity apps: Content analysis. *JMIR MHealth and UHealth*, 6(12). <https://doi.org/10.2196/10314>
- Luz De Araujo, P. H., Röttger, P., Hovy, D., & Roth, B. (2025). *Principled Personas: Defining and Measuring the Intended Effects of Persona Prompting on Task Performance*. <https://github.com/peluz/principled->
- Mahmoud Sajjadi Mohammadabadi, S., Cem Kara, B., Eyupoglu, C., Uzay, C., Serkan Tosun, M., & Karaku, O. (2025). *A Survey of Large Language Models: Evolution, Architectures, Adaptation, Benchmarking, Applications, Challenges, and Societal Implications*. <https://doi.org/10.3390/electronics>
- Maslej, N., Fattorini, L., Perrault, R., Gil, Y., Parli, V., Kariuki, N., Capstick, E., Reuel, A., Brynjolfsson, E., Etchemendy, J., Ligett, K., Lyons, T., Manyika, J., Carlos Niebles, J., Shoham, Y., Wald, R., Hamrah, A., Santarlasci, L., Betts Lotufo, J., ... Oak, S. (2025). *Artificial Intelligence Index Report 2025*.
- Moilanen, M. I. M. E., Iivari, N., Arhippainen, L., & Lanamäki, A. (2024). Guidelines for Disc Golf Applications and Design Principles for SportsHCI: A Human-Centered Approach. *ACM International Conference Proceeding Series*, 396–411. <https://doi.org/10.1145/3701571.3701584>
- Mueller, F., & Young, D. (2018). 10 Lenses to design sports-HCI. *Foundations and Trends in Human-Computer Interaction*, 12(3), 172–237. <https://doi.org/10.1561/11000000076>
- Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2025). A Comprehensive Overview of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 16(5), 1–72. <https://doi.org/10.1145/3744746>
- Olla, P., & Shimskey, C. (2015). mHealth taxonomy: a literature survey of mobile health applications. *Health and Technology*, 4(4), 299–308. <https://doi.org/10.1007/s12553-014-0093-8>
- OpenAI. (2026, January). *Models | OpenAI API*. <https://platform.openai.com/docs/models>
- Peppers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). *THE DESIGN SCIENCE RESEARCH PROCESS: A MODEL FOR*

*PRODUCING AND PRESENTING INFORMATION SYSTEMS RESEARCH*  
Corresponding Author.

- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Pinem, A. A., Sensuse, D. I., Suryono, R. R., Kautsarina, K., & Hidayanto, A. N. (2024). Mobile fitness application quality analysis: KANO model satisfaction or dissatisfaction. *Journal of Infrastructure, Policy and Development*, 8(8). <https://doi.org/10.24294/jipd.v8i8.5960>
- Schulhoff, Sander, Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y., Gupta, A., Han, H., Schulhoff, Sevien, Dulepet, P. S., Vidyadhara, S., Ki, D., Agrawal, S., Pham, C., Kroiz, G., Li, F., Tao, H., Srivastava, A., ... Resnik, P. (2025). *The Prompt Report: A Systematic Survey of Prompt Engineering Techniques*. <http://arxiv.org/abs/2406.06608>
- Shaw, M. P., Satchell, L. P., Thompson, S., Harper, E. T., Balsalobre-Fernández, C., & Peart, D. J. (2021). Smartphone and tablet software apps to collect data in sport and exercise settings: Cross-sectional international survey. *JMIR MHealth and UHealth*, 9(5). <https://doi.org/10.2196/21763>
- Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104, 333–339. <https://doi.org/10.1016/j.jbusres.2019.07.039>
- Tam, Z. R., Wu, C.-K., Tsai, Y.-L., Lin, C.-Y., Lee, H., & Chen, Y.-N. (2024). *Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models*. <http://arxiv.org/abs/2408.02442>
- Thakur, A. (2024). *The Art of Prompting: Unleashing the Power of Large Language Models*.
- Tilastokeskus. (2025, November 12). *Generatiivista tekoälyä käyttäneiden osuus nousi 41 prosenttiin | Tilastokeskus*. <https://stat.fi/julkaisu/cmh32zpp6711z07w6yfukiiqd>
- Tubek, A., & Duplaga, M. (2018). *The assessment of functionalities of mobile applications supporting physical activity*.
- Veera, C. B., & Satya, V. (2024). *Generative AI: Evolution and its Future*. [www.ijfmr.com](http://www.ijfmr.com)
- Villasana, M. V., Pires, I. M., Sá, J., Garcia, N. M., Zdravevski, E., Chorbev, I., Lameski, P., & Flórez-Revuelta, F. (2020). Mobile Applications for the Promotion and Support of Healthy Nutrition and Physical Activity Habits: A Systematic Review, Extraction of Features and Taxonomy Proposal. *The Open Bioinformatics Journal*, 12(1), 50–71. <https://doi.org/10.2174/1875036201912010050>
- White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. <http://arxiv.org/abs/2302.11382>

- Wobbrock, J. O., & Kientz, J. A. (2016). Research Contributions in Human-Computer Interaction. *Interactions*, 2015-April, 38–44. <https://doi.org/10.1145/2702123.2702608>
- Xiao, X., Hedman, J., Ter Chian Tan, F., Tan, C.-W., & Clemmensen, T. (2017). *Association for Information Systems AIS Electronic Library (AISeL) Sports Digitalization: A Review and A Research Agenda* (Number 6). <http://aisel.aisnet.org/icis2017><http://aisel.aisnet.org/icis2017/General/Presentations/6>
- Xu, Q., Ibrahim, G., Archer, N., & Zheng, R. (2014). Toward automated categorization of mobile health and fitness applications. *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2014-August*(August), 49–54. <https://doi.org/10.1145/2633651.2633658>
- Zheng, M., Pei, J., Logeswaran, L., Lee, M., & Jurgens, D. (2023). *Is “A Helpful Assistant” the Best Role for Large Language Models? A Systematic Evaluation of Social Roles in System Prompts*. <http://arxiv.org/abs/2311.10054>
- Zheng, M., Pei, J., Logeswaran, L., Lee, M., & Jurgens, D. (2024). *when a helpful assistant is not really helpful personas in system prompts do not improve performance of LLMs*.
- Zheyuan Cui, K., Demirer, M., Jaffe, S., Musolff, L., Peng, S., Salz, T., Coppney, P., Du, W., Gao, Y., Redford, L., Salva, R. J., Schocke, D. A., Silver, A., Tai, A.-J., Tetrick, D., & Wilcox Mert Demirer, J. (2025). *The Effects of Generative AI on High-Skilled Work: Evidence from Three Field Experiments with Software Developers \**.