DEVELOPMENT AND EVALUATION OF A FRAMEWORK TO SUPPORT EXPERIMENTAL RESEARCH ON DATA ANALYTICS

Master Thesis
of

MAX DARMSTADT

September 5, 2023

Matriculation Number
1820000

Submitted at the Chair of Enterprise Systems
University of Mannheim

Reviewer: Prof. Dr. Hartmut Höhle
Supervisor: Jan Schilpp

Contents

List of Figures

List of Tables

Listings

List of Abbreviations

AI	Artificial Intelligence
BI	Business Intelligence
CSV	Comma-Separated Values
DSR	Design Science Research
IDE	Integrated Development Environment
ID	Identification
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IS	Information Systems
IT	Informationstechnologie
JVM	Java Virtual Machine
ML	Machine Learning
UI	User Interface

Abstract

In recent years, data analytics has gained increasing significance across all industries due to the unprecedented availability of vast amounts of data. Companies can leverage data analytics to extract valuable insights into consumer behavior, market trends, and internal operations. Consequently, data analytics has emerged as a critical tool for companies striving to maintain a competitive edge in today's rapidly evolving business environment. However, despite its pivotal role in business, there has been a noticeable lack of research in the area of behavioral research in data analytics, particularly when it comes to experimental studies. Specifically, a deficiency exists in research that employs experimental methodologies.

Simultaneously, existing applications designed to facilitate experiment execution exhibit numerous drawbacks and flaws. To address these challenges, the Design Science Research (DSR) approach is adopted to conceptualize an Android application aimed at streamlining the implementation of experiments, thereby enhancing the research process in the area of behavioral research within data analytics. To achieve this objective: (1) a comprehensive review of prior research on data analytics and its methodological procedures was conducted, resulting in the identification of requirements for the application; (2) an artifact designed to improve the research process in the field of behavioral research within data analytics based on these requirements was conceptualized; and (3) said artifact was validated through the exemplary implementation of a study in data analytics using the developed application.

The resulting artifact empowers researchers to conduct experiments in the field of data analytics more efficiently and effectively. Thus, this work not only contributes to the current state of research but also equips future researchers with a powerful tool to generate new knowledge and consolidate existing knowledge in the field of data analytics.

1 Introduction

1.1 Background and Motivation

The introduction and widespread adoption of new technologies that utilize the collection of huge data sources, like data analytics or machine learning, has already been proven to be disruptive for all industries. The amount of data generated globally is increasing rapidly (Seagate, 2018), and the pressure to effectively utilize these data volumes to gain a competitive advantage is rising. This new trend, often referred to as “big data”, after the fact that never-before-seen amounts of data are generated and available for processing, enables completely new business areas. This is reinforced, among other things, by the fact that many companies already view their data as a primary business asset (Redman, 2008). Simultaneously, the emergence of big data promises to completely reshape the decision-making process of traditional businesses through the adoption of data analytics. Although sales in the area of big data have risen significantly over the past years (BIS Research, 2018, Bitkom, 2018), and businesses already view big data as an important information technology trend (Bitkom, 2017), a lot of organizations struggle to effectively utilize their data. Some 84% of industry-leading companies in the United States and around the world were already investing in big data analytics in 2019, according to their own statements, which only underlines the importance of data analytics for decision-making in the economy (Statista, 2019). This is also reinforced by the market for big data analytics worldwide expected to more than double in size in the next 6 years (Statista, 2022).

In their article, Amankwah-Amoah and Adomako study the influence of big data usage on business failure. They conclude that the mere possession of big data as an asset has no positive effect on an organization (Amankwah-Amoah and Adomako, 2019) and that, in order to prevent business failure, big data must be used effectively (Amankwah-Amoah and Adomako, 2019). Conducting research to resolve the underlying factors hindering the efficient utilization of data analytics in this particular context holds crucial significance. However, although decision-making in data analytics has recently attracted scholars’ attention (L. Chen et al., 2022), there is still a lack of research on non-technical aspects hindering the utilization of data analytics.

Section 3 of this thesis uses the information-value-chain to examine the state of research in data analytics. Specifically, a literature review in the field of Data Analytics is conducted. The results of this literature review reveal a lack of research on non-technical aspects of the information-value-chain, mainly in areas like decision-making and behavioral research. These results have been confirmed by studies in the past focusing on related fields (Trieu, 2017), which might indicate persistent issues. This becomes more apparent as new technologies, which overlap with data analytics like Machine Learning (ML) and Artificial Intelligence (AI), become more widespread. These "black-box"[1] technologies being already an alternative to human decision-making (Krakowski et al., 2023b). Moreover, other studies confirm that aspects like company culture, business models, and the overall commitment and strategy of organizations have a significant impact on the effectiveness of data analytics (Holsapple et al., 2014).

This lack of research on non-technical aspects could become a significant issue in the future, specifically for the decision-making process, as firms' top priorities increasingly focus on big data analytics for strategic decision-making (Ghasemaghaei, 2019). Furthermore, it is indicated in section 3 that the current state of research specifically lacks a variety of experiments conducted to confirm the validity of frameworks and hypotheses. Experiments are a particularly important tool for investigating causal relationships in research (Gniewosz, 2011). In addition, other means of collecting information, like inquest questionnaires, which are probably the most frequently used form of obtaining information (Mummendey and Grau, 2014) in quantitative research, are not always the most suitable method. The behavior of people, for example, which comprises the literature gap found, can be better assessed by means of observational studies or experiments (Gniewosz, 2011). While prior research has conducted behavioral research of data analytics with surveys and case studies, little or no attention has been paid to the verification of research results and hypotheses through experiments. This general lack of experimental research in certain areas connects the applied business problem of better utilizing data analytics in organizations to the theoretical problem of a lack of experimental research generally found in data analytics (referring to the results of the literature search in section 3). Improving the experimental research process in the field of data analytics

[1] Technologies whose exact internal structure is unknown.

could significantly enhance the future state of knowledge on this topic, while also allowing organizations to succeed in the fast-paced economic environment of the digital age.

1.2 Objective and Expected Contribution

The objective of this thesis is, therefore, to improve the experimental research process in the field of behavioral research in data analytics. This is done through the following three objectives: (1) the review of prior research on data analytics and its methodological procedure, (2) the development of an artifact that improves the research process in the field of behavioral research in data analytics, and (3) the validation of the artifact through the exemplary realization of a study in said field utilizing said artifact. In order to accomplish the creation of this artifact, the Design Science Research (DSR) methodology is used, which contains the six steps: *Identification of the Problem*, *Definition of Objectives for a Solution*, *Design and Development of Artifacts*, *Demonstration of the Artifact*, *Evaluation of the Solution*, and *Communication* (Peffers et al., 2006, Dresch et al., 2015). Based on these steps, the corresponding artifact is conceptualized. For this purpose, a literature review in the field of data analytics is first conducted to identify research gaps and underlying problems in the area of data analytics. Subsequently, a further literature review and other sources are used to establish requirements that the conceptualized artifact must meet to solve these research gaps and underlying problems. This is accomplished by identifying literature in the field of data analytics that uses experiments or whose research object would, in principle, have permitted the use of experiments. These insights are then used to conceptualize, analyze, and validate requirements for the final artifact, utilizing the *Requirement Engineering* approach (Sommerville, 2011, Alain Abran, James W. Moore, 2004). These requirements are then used to design and develop an artifact. Subsequently, the resulting artifact is demonstrated and assessed by implementing a real experimental study in data analytics as an example to evaluate the artifact and its benefits for the experimental research process. This exemplary implementation is also used to validate the aforementioned requirements. The last step of the DSR framework, which focuses on communicating the results to its stakeholders, is ensured by this thesis itself. The practical contribution of this thesis to research is twofold. On the one hand, an artifact is created that accelerates research in the field of data analytics through the improvement of the research process. On the other hand, meta-knowledge about the research process itself is created, which not only improves the conduct of research through said artifact but can also be used in off-topic areas beyond the use cases of this thesis.

2 Theoretical Foundations

2.1 Design Science Research Methodology

In order to accomplish the goal of this thesis, which is to create an artifact for the improvement of the research process in the field of data analytics, the DSR approach is briefly discussed and introduced. In fundamental terms, design science is a research approach that aims to develop and validate science-based design knowledge and guide research towards problem solving (Hevner et al., 2004, Dresch et al., 2015). The goal of DSR is to gain prescriptive knowledge about the composition of various artifacts, including software, methods, models, and concepts. This particular design knowledge facilitates a systematic and scientific approach to the design of future projects. The design process and its practical implementation generate design-oriented knowledge, enriching the existing knowledge in DSR (Hevner et al., 2004). Thus, the result of design science, especially in Information Systems (IS), is the creation of an effective Informationstechnologie (IT) artifact that addresses a certain problem (Hevner et al., 2004), making DSR a suitable approach for conceptualizing an artifact in the field of data analytics. Nevertheless, the exact activities of the design science model may differ from author to author to some extent (Fulcher and Hills, 1996). This thesis aligns itself with the phases and steps outlined by Peffers et al. in their 2006 article titled "The design science research process: A model for producing and presenting information systems research." In their article, Peffers et al. analyze literature that implements design science to create a generally accepted process for research in IS (Peffers et al., 2006). As a result of their work, they describe the design science research approach using the six steps: *Identification of the Problem*, *Definition of Objectives for a Solution*, *Design and Development of Artifacts*, *Demonstration of the Artifact*, *Evaluation of the Solution*, and *Communication* (Peffers et al., 2006). These six steps, as previously described, are used to conceptualize an artifact in this thesis.

2.2 Data Analytics

The term “data analytics” originated in the early 2000s and describes an interdisciplinary field that combines areas such as statistics, machine learning, pattern recognition, system theory, operations research, and artificial intelligence (Runkler, 2020). It can be generally defined “[...] as the application of computer systems to the analysis of large data sets for the support of decisions” (Runkler, 2020). This definition showcases the broadness of the topic, as most computer systems process some amount of data and thus theoretically allow for some kind of decision-making. Due to this broad definition, data analytics can cover slightly different subject areas depending on the context in which it is discussed. In this thesis, data analytics refers to the processing of large amounts of data, also referred to as “big data”, through mathematical procedures or ML methods with the goal of creating new knowledge. Subsequently, processes that merely prepare or show data are not considered data analytics, but only processes that process data in such a way that new knowledge can be derived from it. This distinction is made to differentiate data analytics from traditional data processing areas like Business Intelligence (BI). The goal of data analytics, as discussed in this thesis, is to retrieve some kind of previously unknown knowledge from a set of data. This process can be generally described using the “information value chain” model. In their research, Abbasi et al. analyze this model in the context of big data in an effort to create an inclusive research agenda for big data in information system research (Abbasi et al., 2016).

2.3 Information Value Chain

The information value chain (Figure 1) is a set of phases that define the transformation of raw data into information and eventually into knowledge. “Data” describes raw facts without any structuring. Once organized, the processed data represents “Information”. This “Information” is then used to find patterns and draw conclusions. At this point, the information becomes knowledge (Fayyad et al., 1996a, Fayyad et al., 1996b). This knowledge is then used to make “Decisions” and take corresponding “Actions” (Sharma et al., 2014).

Each phase of the information value chain also includes a different set of technologies and methodologies. For example, the “Data” phase contains technologies and actions regarding the basic storage of data, such as database systems or data warehouses (Abbasi et al., 2016). The conventional version of this information value chain represents an approach that generally explains the processing of data. The main steps of this information value chain are also applicable to big data (Abbasi et al., 2016). This general structure of processing data is also supported by literature from the data analytics field (Runkler, 2020). In addition, the information value chain includes the further phases “Decisions” and “Actions”, which deal with the influence of the processed data. These phases reflect the impact of data analytics, as data analytics is primarily a technology for the decision-making process (Runkler, 2020). For this reason, the information value chain is a suitable model to structure different phases in the processing of data in the context of data analytics.

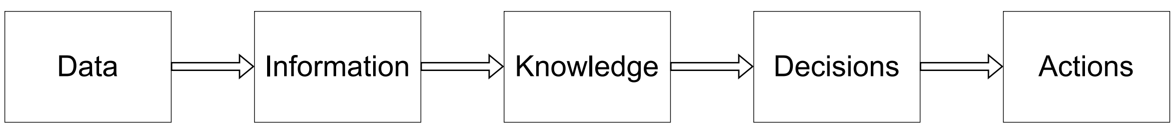


Figure 1: Information Value Chain

3 Identification of the Problem

This part of the thesis defines the problem to be solved by the DSR approach. For this purpose, a literature review is conducted to find potential gaps in the field of data analytics and the methods used in this field. Subsequently, applications and tools intended to support the research process are examined.

To identify constraints in the research on data analytics, a literature search is conducted. The main objective is to analyze existing literature to find research gaps, particularities, and interrelationships between the literature. This is supposed to provide insights into the current state of research and determine which part of the research process on data analytics still has room for improvement. Consequently, relevant literature was identified and reviewed. Afterward, the identified literature was categorized and analyzed. Initially, it was assumed that the topic of data analytics lies in both the field of information systems and business (Abbasi et al., 2016, Levina and Vaast, 2005). For this reason, the literature search was primarily conducted in literature databases that focused on these topics. The literature search was conducted using a keyword search.

To ensure the quality of the identified literature, initially, only publications from certain journals were considered. These journals include the *Senior Scholars' Basket of Journals* and the *UT Dallas Top 100 Business School Research Rankings*. The former includes journals in the area of information systems, and the latter includes journals in the area of business administration. A full list of keywords, databases, and journals used is included in appendix A. Furthermore, only peer-reviewed articles were taken into account. This was done to ensure the quality of the found publications and to additionally exclude book reviews, editorials, and opinion statements. Moreover, other 'non-scholarly' texts or publications that did not meet scientific requirements were also not considered in the search. Secondly, the abstracts of the particular articles were inspected to narrow the search further. Consequently, literature that did not match the topic of data analytics was excluded from the search. The literature found in the search was then used for a backward and forward search. During a backward search, all cited sources of an article are examined, and during a forward search, all the literature that cites the original article is examined (Webster and Watson, 2002). The backward search was conducted using Google Scholar. In addition to this, articles from other journals were, in a second step, reviewed and included as well if they met the scientific requirements, were officially published, and were relevant to the topic. This process yielded 35 research publications. The results were then assigned to different phases of the aforementioned information value chain, representing their content best. This was done to find literature gaps in the general process of data processing. Additionally, the identified literature was categorized by its research methodology to find patterns and similarities in the literature. By mapping the literature found to the information value chain, parts of the data processing process that are over- or under-represented may become visible. From this, conclusions can be drawn about the current state of research. Furthermore, the categories "Overspanning" and "Other" were introduced in addition to the phases of the information value chain to represent literature that either fits multiple phases of the information value chain or none. Using this method leads to the results shown in the "First Search" column of table 1.

Information Value Chain	First Search	Additional Search
Data	4	
Information	3	
Knowledge	21	
Decisions	4	0
Actions	0	0
Overspanning	0	3
Other	3	
Total	35	3

Table 1: Results Assigned to the Information Value Chain

Table 1 shows an overabundance of literature that was assigned to the "Knowledge" phase of the information value chain. The significantly fewer entries for the other phases indicate less research on these subparts of the information value chain. However, it cannot be concluded that this underrepresentation is due to the fact that these phases are less relevant in the context of data analytics. For this, more literature would have to exist, confirming that these areas are less important in the field of data analytics. The underrepresentation of the phases "Data" and "Information" could also be explained by the fact that these phases are more technology-driven and, therefore, less researched in a behavioral data analytics context. In fact, the corresponding literature, which was assigned to these phases, mainly consists of publications researching the technical possibilities and applications of data. Their primary research objective does not directly involve broader topics for companies or the application of data analytics. Nonetheless, in total, seven individual publications could be found that fit into these two phases. In addition, these two phases ("Data" and "Information") will be considered as one in the further thesis since the literature assigned to these phases is thematically very closely related. Only four publications were assigned to the "Decisions" phase, and none to the "Actions" phase. These results, in particular, call into question whether the topic of behavioral research in data analytics has been extensively researched. The reason for this is the fact that data analytics is primarily a decision support method (Runkler, 2020). Therefore, an overabundance of literature delineating the decision-making process of data analytics should likely exist. This is compounded by the fact that no literature could be found that addressed overarching issues, as no overarching theories could exist for an insufficiently studied topic. In order to ensure that the ratio of the literature found is based on the research state and not on the keyword search being biased in any way, a second literature search was conducted, focused on finding more literature that could be assigned to the “Decisions” or “Actions” phase. This was only done for these phases as these two are most relevant in the context of data analytics and because, in total, the least literature could be assigned to them (viewing “Data” and “Information” together). This second keyword search was conducted with the goal of finding more literature that could be assigned to the phases “Decisions” and “Actions”. Therefore, a new set of keywords, including “Decision”, “Decision Making”, and “Action”, were added to the existing set of keywords. Furthermore, the abstracts were examined with an emphasis on the aforementioned goal. The results of this second keyword search are represented in the “Additional Search” column of table 1. A total of three additional publications were identified using this second search. These three publications were all assigned to the “Overspanning” category. Consequently, no additional literature that could be assigned to the phases “Decisions” or “Actions” could be identified. This further indicates that the topic of boundaries in data analytics is not extensively researched.

To further analyze the literature and potentially draw further conclusions, the found literature was also categorized regarding the research methods used. This categorization is presented in table 2.

Research approach	Method	Number
Qualitative (22)	Case Study	13
	Interviews	4
	Experiments	2
	Observation	3
Quantitative (16)	Survey	12
	Data Analysis	6

Table 2: Research Approach Used in the Literature

The distribution presented in table 2 shows significant discrepancies in the number of research methods used. Methods such as case studies and surveys are used more frequently than average, in contrast to other methods. The least frequently used methodology is the experiment. This fact further indicates an insufficient exploration of the field of behavioral research in data analytics, as experiments are the most suitable method for researching the behavior of people (Gniewosz, 2011). In this context, experiments are a particularly important tool for investigating causal relationships in research (Gniewosz, 2011). The lack of experimental research seems to have not only created a blind spot in the current state of knowledge but also a permanent limitation for new research, which lacks existing knowledge as a foundation for new science. Although current knowledge suggests that both behavioral and experimental research should be an integral part of data analytic research, the literature review of this section was able to show that both areas are extremely underrepresented in literature and research. This circumstance is further underlined by the fact that resources already exist to support experimental research in the field of behavioral science (Columbia Experimental Laboratory for Social Sciences, 2023), but none of the articles analyzed utilized them. These resources are discussed briefly below to highlight potential problems in their use. The following applications are resources recommended for experiments by the Columbia Experimental Laboratory for Social Sciences (2023). The selection of resources analyzed in the course of this thesis is limited to applications that allow the implementation of interactive experiments. Resources or applications that are only suitable for conducting surveys or passive experiments were neglected. A small example experiment was implemented with each of the applications in order to find out the advantages and disadvantages of the respective resource and to identify potential challenges with them in the context of data analytics.

z-Tree Zurich Toolbox for Ready-made Economic Experiments: The software component z-Tree was developed as a toolbox for conducting economic experiments. The technical setup of z-Tree usually consists of a client/server architecture, where the experiment conductor runs a server and the test subjects participate via a client. In general, the program requires a Windows operating system, making it compatible with around 28.59% of devices (statcounter, 2023). Simultaneously, however, this architecture ties the experiment setup exclusively to desktop computers. Furthermore, the possibilities to use graphical interfaces in experiments are extremely limited. The same is true for computer interfaces that have been developed after the 1990s. Thus, the experiments implemented with z-Tree are mostly limited to mouse and keyboard inputs, while also limiting the amount of metadata that can be collected. These limitations are especially critical for the data analytics field, which is driven by extremely recent developments. Adapting z-Tree to newer experiments can thus involve very high programming effort (of Zurich, 2023, Fischbacher, 2006, D. Chen et al., 2016). Z-Tree is also a proprietary software which can be licensed for free, but it is not an open-source software (Fischbacher, 2006). In conclusion, z-Tree thus represents an outdated and extremely limiting software component, which is not suitable for experimental research in the area of behavioral research or data analytics.

oTree An open-source platform for laboratory, online, and field experiments: oTree is a publicly accessible, open-source, and web-driven software solution developed for the purpose of facilitating the implementation of experiments and based on the Python programming language. It removes many of the restrictions compared to the older z-Tree software component, such as limited user interfaces, a Windows constraint, limited extensibility, and more. One of the main intentions of oTree is to facilitate the execution of field experiments. By default, oTree supports simple interactive experiments. More complex experiments have to be implemented in Python, requiring programming expertise. Experiments implemented in oTree are conducted exclusively via a client/server architecture, where the experiment performer has to set up a server and the experiment participants then participate in the experiment via any device using a browser. This has great advantages, especially for field experiments where participants can be recruited via a customized web link. Nevertheless, this also requires a lot of technical programming expertise to set up additional hardware like a server. Depending on the complexity of the experiment, the work that oTree saves the developer is therefore rather limited. At the same time, simple experiments still require a certain amount of technical know-how, making oTree unsuitable for non-technically proficient individuals. In conclusion, oTree is, in principle, a solid solution for conducting experiments, but due to its high level of technical expertise and other limitations (D. Chen et al., 2016), it seems to be rather unsuitable as a basis for conducting experiments in the field of behavioral research in data analytics.

LIONESS Lab A free web-based platform for conducting interactive experiments online: LIONESS Lab constitutes a cost-free, web-based platform designed for the facilitation of interactive online experiments. It is developed by the Centre for Decision Research and Experimental Economics (University of Nottingham, UK) and the Chair of Economic Theory (University of Passau, Germany). Thus, Lioness Lab is not only a proprietary software solution but also requires special access for its use even though it is free of charge. At the same time, the publishers of LIONESS LAB repeatedly emphasize that their solution hardly requires any programming knowledge to perform experiments (Giamattei et al., 2020). A circumstance that can complicate the use of complex custom coding for complex experiments. In addition, this paradigm removes much of the complexity of the software from the coding level to the customizing level. A problematic circumstance since this limits the extensibility and basically does not reduce the perceived complexity (Chou and Chang, 2008b). In conclusion, LIONESS Lab is a platform that stands out due to a variety of functions, but unfortunately is not open source. At the same time, the extensibility of the platform is limited, which makes it a good choice for suitable experiments, but unfortunately unusable for more complex experiments like behavioral research would require (Giamattei et al., 2020).

In summary, none of the presented applications could be used for experimental behavioral research in data analytics without significant drawbacks. Therefore, it can be concluded that one of the possible reasons for the lack of experimental behavioral research in data analytics is the lack of a suitable application or framework.

4 Definition of Objectives for a solution

On the basis of the results of the previous section, the identified problems to be solved by the artifact are a gap in behavioral research in the field of data analytics and a gap in the number of experiments conducted in this field. At the same time, applications that are designed to be used to implement experiments have large capability gaps. For this reason, the artifact is designed to be implemented in the form of a software application that enables experimental research in the field of behavioral research in data analytics. In order to define an objective for a possible solution in the form of a software application, requirements must be engineered (Seacord et al., 2003). Requirements can be classified according to ISO/IEC 25000, respectively, the quality model from ISO/IEC 25010, as quality criteria for software and systems (ISO/IEC 25010, 2011). The Institute of Electrical and Electronics Engineers (IEEE) defines requirements as:

> ***"(1) A condition or capability needed by a user to solve a problem or achieve an objective. (2) A condition or capability that must be met or possessed by***

a system or system component to satisfy a contract, standard, specification, or other formally imposed documents. (3) A documented representation of a condition or capability as in (1) or (2).” (IEEE, 1990)

According to these definitions, requirements can be generally defined as properties that need to be met in order to achieve an objective. In order to engineer and provide a certain quality, these requirements are established using the *Requirement Engineering* approach for the analysis and evaluation of requirements (Alain Abran, James W. Moore, 2004, Sommerville, 2011). This approach has been shown to clearly contribute to software project successes in the past (Hofmann and Lehner, 2001) and is, therefore, a suitable approach to define the objectives for a solution. The exact individual phases and steps of the *Requirement Engineering* approach can vary from source to source and use case to use case. In general, however, all steps fall into one of three main categories: *Requirements Elicitation*, *Requirements Specification*, and *Requirements Validation* (Alain Abran, James W. Moore, 2004, Sommerville, 2011, Fernandes et al., 2009). In the first step, possible requirements and use cases are collected via a variety of sources like analyses, surveys, literature, or interviews (Sommerville, 2011). This thesis utilizes a literature review in order to discover requirements. In the next step, the requirements are then specified and categorized, with an important distinction being the difference between functional and non-functional requirements. In the *Requirements Validation* step, the elicited requirements are then tested for their validity. This phase emphasizes the reviewing of the requirements to find out whether these requirements are actually representative of the desired artifact (Sommerville, 2011). This is accomplished through Validity, Consistency, Completeness, Realism, and Verifiability checks in conjunction with prototyping and testing the requirements (Sommerville, 2011).

4.1 Requirements Elicitation

This phase gathers information in order to discover possible requirements for the final artifact. These requirements are discovered through a literature review of studies that have utilized experiments in the field of behavioral research in data analytics or that could have allowed for the usage of experimental research. Moreover, other sources and the findings from the already analyzed applications for conducting experiments are also taken into account in identifying the requirements.

4.1.1 Studies in Data Analytics - A Literature Review

In order to establish further requirements, a second literature review is conducted, which focuses on articles and studies in the field of behavioral research in data analytics. The goal of this second literature review is to understand commonalities and challenges of studies and especially experiments conducted in the area of data analytics, in order to establish requirements for the creation of the artifact. Due to the assumption that data analytics lies in both the fields of information systems and business administration, in addition to some others, the same databases as for the literature review in section 3 were used. By using the same databases as in section 3, it is also ensured that the requirements are elicited based on the same general selection of literature as the original problem was identified with. The approach to the literature search, established in section 3, was also used for this literature review, ensuring a thorough examination of relevant research while avoiding unnecessary alterations to the process of finding literature. Therefore, to maintain the integrity of the identified literature and to avoid repetition, a process was followed as described in the previous literature review (section 3). Specifically, publications from journals were selectively considered, including those listed in the *Senior Scholars' Basket of Journals* for Information Systems and the *UT Dallas Top 100 Business School Research Rankings* for Business Administration. As for the previous literature review, in order to ensure the quality of the publications, only peer-reviewed articles were considered, while book reviews, editorials, and opinion pieces were excluded. In addition, "non-scientific" texts or publications that did not meet scientific criteria were excluded from the search. The research was further refined by carefully reviewing article abstracts, which ensured that the selected literature remained relevant to the topic of experimental research in data analytics. The abstracts were reviewed not only for the use of experiments in the study but also for research designs that could have allowed hypothesis testing by experiment. This was done to counteract the effect of selection bias. Selection bias generally describes the effects of making assumptions based on a sample size that does not represent the full population (Heckman, 2010). A simple example of selection bias would be calculating the average disposable income of families based on the annual tax bill. This experimental design would reduce the total population to taxpaying families and thus lead to potentially grossly inaccurate results, since families living below the taxable threshold would not be included in the study. For the same principle, studies in which no experiment was performed are included in the literature review. The goal of this thesis is to improve the process of experimental research in the field of data analytics in general. Considering only studies that already perform experiments would not be representative of the full field and would therefore be prone to selection bias. It could be, for example, that certain circumstances, possibly the lack of an appropriate application, make it difficult to perform experiments. This fact would be completely lost if only studies that already perform experiments were considered. An adequate example of an article that falls into this category is Sebastian Krakowski, Johannes Luger, and Sebastian Raisch's 2022 article "Artificial intelligence and the changing sources of competitive advantage," in which they research how AI changes the competitive advantage by being substitutes for humans in managerial tasks and decision making. For this purpose, they are examining data from chess tournaments that have already been held. However, the same research question could have been answered by conducting experiments with chess players instead of using historical data from tournaments. Remarkably, the authors seem to come to a similar conclusion, stating that further research in this area should be conducted through experiments (Krakowski et al., 2023a). Furthermore, it should be noted that the focus of this literature review was not to outline the current state of research but to identify as many appropriate articles as possible. For this reason, fewer search terms were used, and articles from different subareas were generally admitted, as long as they are located in the larger context of data analytics or decision making. For this reason, a backward and forward search was also omitted. The search terms used for this literature review are "Data Analytics", "Decision Making", and "Big Data". With these terms, the search process should be kept as broad as possible. The search was further limited to articles written in the English language. The abstracts of all articles were then analyzed. In this way, a total of 19,955 articles were considered by inspecting their titles and, if applicable, their abstracts. A full list of databases used and identified articles in their corresponding database can be found in appendix B.

This process initially yielded 46 research publications, which were supplemented by

Research Method	Total
Case Study	16
Interview	3
Experiment	10
Observation	3
Survey	17
Data Analysis	7
Quantitative	34
Qualitative	32

Table 3: Research Methods for Literature Search on Requirements

articles from the previous literature review corresponding to the criteria. This resulted in a total number of 56 articles. The results were then classified according to the research method they utilize and whether a qualitative or quantitative approach was used. The results of this classification are presented in table 3. An important note at this point is that the number of articles using experiments does not contradict the gap identified in section 3, since in this literature search specifically filtered for research articles that use experiments. Thereafter, the exact experimental setup of the articles was analyzed in order to discover requirements for their individual experiments and therefore for the artifact which is the subject of this thesis.

Requirement	Proper	Extra	Requirement	Proper	Extra
Reusable	11	0	Participant Data	24	23
Interoperability	1	0	Displaying Information	22	23
Meta Data Collecting	8	0	Different Groups	14	2
Post-Interview	2	0	Additional Logic	12	0
Time-Flexibility	9	0	Evaluation of Data	27	22
Multi-Source	1	0	Participant Input	21	25
Vizualize Final Data	2	0	Real-Time Exchange	3	0
Proactive System	2	0	Distant Communication	18	0
Pre-Loading Data	1	0	Selecting Data	2	0
Monitoring of Study	1	0	Simplicity	1	0
Debrefing Info	1	0			

Table 4: Requirements Uncatagorized

The requirements derived from this are included in table 4. As already mentioned, not only the requirements of the performed experiments were included, but also the requirements of setups that would have been suitable for an experiment. The column “Proper” indicates whether a requirement was actually mentioned directly in the article, and the column “Extra” indicates whether this requirement would have had to be used theoretically if the study had conducted an experiment. As already explained, this should ensure that not only requirements that fit already performed experiments are discovered, but also to counteract selection bias.

4.1.2 Further Relevant Reference Resources

In addition to the literature review, this section includes other sources that can contribute to the derivation of requirements. These sources and requirements mainly refer to external influencing factors or specifications. Only requirements that have not yet been established by the previous methods are included.

In their book on Empirical Educational Research, Gniewosz et al. (2011) describe, among other things, the structure and empirical theory in relation to experiments. An important part of conducting experiments, according to the authors, is to educate the subjects about the experiment and its benefits. Rather than explaining the design of the experiment, this involves explaining the actual benefits and goals of the experiment to the participants after it has been conducted (Gniewosz, 2011). Furthermore, confounding variables have to be eliminated for the effective execution of experiments. Although some of these measures must be implemented on a case-by-case basis by the individual experimental setup itself, the artifact is intended to assist in this process when possible. Especially measures regarding the allocation of groups represent an important part of this (Gniewosz, 2011) and could be supported by an artifact. For this reason, the requirements "Random or targeted assignment of test subjects" are included in the list of requirements. The creation of control groups via different participant groups is already included as a requirement from several sources. In addition, findings from the analysis of the alternative applications for conducting experiments from section 3 are used to establish requirements. These include the use of complex and up-to-date user interfaces, deployment on non-Windows platforms, the open-source approach, and the possibility to add custom program code for the experiments based on the analysis of z-Tree and oTree. Further requirements based on the analysis of LIONESS Lab include the openness of the platform, the creation of different groups regarding the test subjects, and the possibility to use unrestricted coding.

4.2 Requirements Specification

After the requirements have been discovered and roughly outlined in the previous section, they are concretely specified, organized, and classified in this section. For this purpose, the requirements were specified concretely with explanations and categorized based on their task. At the same time, the requirements were classified into functional and non-functional requirements. A functional requirement describes a function that a system or system component must be able to perform (IEEE, 1990). An example of a functional requirement would be the calculation of a price tag in euros and in dollars. Non-functional characteristics describe, on the other hand, the behavior of a system (Seacord et al., 2003) and go beyond functional characteristics. Thus, functional requirements describe what a system must be able to do, and non-functional requirements describe how this should be done. Non-functional requirements also often describe the quality of the individual functions and can influence several other requirements (Balzert, 2011). An example of non-functional properties would be that the conversion from euros to dollars must be performed in “a few seconds”. Table 5 contains the functional, and table 6 the non-functional requirements. The requirements are composed of the requirements discovered in sections 4.1.1 and 4.1.2 and are included together in the tables. Requirements derived from multiple sources and methods are also just listed once in the table.

Requirement	Description
Information	
F1.1 Displaying Information	Information must be able to be displayed
F1.2 Debrefing Info	Debrifing Information must be able to be displayed
Data Collecting	
F2.1 Participant Data	Basic data about the participants must be able to be collected
F2.2 Meta-Data	Meta data must be collected
F2.3 Post-Interview	It must be possible to collect data after the experiment
Pre-Loading	
F3.1 Pre-Loading Data	Data must be able to be pre-loaded
F3.2 Selecting Data	Data must be able to be pre-selected and deleted
Experiment Setup	
F4.1 Additional Logic	Custom logic/program code can be executed within the artifact
F4.2 Participant Input	The artifact enables user input
F4.3 Proactive System	The artifact can pro-actively propmt a user action
Groups	
F5.1 Different Groups	The artifact must allow the division of participants into different groups
F5.2 Communication of Groups	The different groups must be able to interact with each other
F5.3 Targeted Assignment	Groups of test subjects must be able to be created based on certain attributes like confounding variables.
F5.4 Random Assignment	Groups of test subjects must be able to be created based on random assignment

Table 5: Functional Requirements Structured

Requirement	Description
Time-space non-reliance	
N1.1 Distand Communication	The artifact can be used regardless of the location of the participant
N1.2 Time-Flexibility	The artifact can be used independently of a given period of time
Data Postprocessing	
N2.1 Evaluation of Data	The data can be retrieved in a suitable way for further processing and evaluation
N2.2 Vizualize Final Data	The data can be retrieved in a suitable way for further visualization
Simplicity	
N3.1 Simplicity	The artifact is simple to use
Reusable and Interoperable	
N4.1 Reusable	Experiments with the artifact are easy to re-do
N4.2 Interoperability	The artifact is interoperable
N4.3 Openness of Platform	The artifact is open to changes and enhancements
Monitoring	
N5.1 Monitoring of Study	The study conducted with the artifact can be monitored
Pre-Loading	
N6.1 Multi-Source	Data from multiple-sources must be pre-loaded
Advanced User Interface	
N7.1 Advanced User Interface	The Artifact enables the usage of modern user interface components

Table 6: Non-Functional Requirements Structured

4.3 Requirements Validation

In this section, the previously established requirements are validated. The criteria validity, consistency, completeness, realism, and verifiability from the *Requirement Engineering* approach are used for this purpose. The validity criterion indicates whether the requirements imposed actually correspond to the intended functions. Since the requirements were drawn up by taking into account applications that have already been developed and studies that have already been carried out, it can be assumed that the requirements correspond more precisely to the functions that are actually required than if they had been drawn up by any stakeholders. In addition, the requirements were established using a clearly defined scientific process and by adding literature from the field, which should also increase validity. Some of the requirements derived from analyzing existing applications are also in conformity with requirements from subsection 4.1.1 and 4.1.2, indicating that the requirement search covers the problem to be solved very well. For these reasons, it is assumed with a high probability that the validity of the requirements is guaranteed. Moreover, the established requirements in table 5 and 6 do not contradict each other, which means that the consistency criterion is also met. The completeness criterion, which describes whether the overall scope of the functions is covered by the requirements, is difficult to confirm. The reason for this is the open nature of the artifact, which should enable the improved execution of arbitrary experiments. Thus, the functional scope of the artifact is theoretically endless. Nevertheless, this criterion can be confirmed considering requirement F4.1 ensures that additional logic and program code can be implemented within the artifact, which means that theoretically an infinite number of further functions can be implemented by the person performing the experiment (as far as these requirements can be implemented by a Turing-Complete programming language). As a result, the completeness criterion is also considered to be fulfilled. The criterion realism can also be confirmed. No requirement indicates that it could not be implemented technically, as will be shown in section 5. The last criterion to confirm the validity of the requirements is the verifiability, which describes whether the individual requirements are formulated in a way that they can be tested. This is an important point since the artifact must be tested for all requirements following the DSR approach. In order to be able to formally confirm the criterion, test cases are therefore designed which check the artifact against the corresponding requirements. This approach is also supported by the scientific literature for the validation of requirements and is later used in the DSR approach to confirm the requirements (Sommerville, 2011).

ID	Test Description	Requirement
T1	A welcome and goodbye message is displayed	F1.1, F1.2
T2	Participants are prompted to input their age at the beginning and prompted to input how the liked the experiment at the end	F2.1, F2.3, F4.3, F4.2
T3	The information about how long the experiment took is collected	F2.2
T4	The sex and the weight of the participant is pre-loaded into the experiment from different files. The sex of the participant is deleted.	F3.1, F3.2, N6.1
T5	A chess game is added as custom logic	F4.1, F4.2, F4.3
T6	Two groups are created, one of the groups is particularly chosen the other one randomly selected	F5.1, F5.3, F5.4
T7	A chess turn is played by both parties not using the same device	F4.2, F5.2, N1.1, N1.2
T8	The results of the experiment are retrieved and displayed in third party software	F2.2, N2.1, N2.2, N5.1
T9	The experiment is redone a second time and another experimental setup is implemented	N4.1
T10	The experiment is conducted on different devices	N4.2
T11	During the experiment the current state of the chess board is exported to the conducter of the experiment	N5.1

The non-functional requirements of “Simplicity”, “Openness of Platform”, and an “Advanced User Interface” cannot be precisely tested by a test case due to their subjectivity. One way to counteract this would be to formulate the requirements more precisely. For example, the requirement “Advanced User Interface” could be replaced with “A User Interface Developed in 2023”. This would satisfy the verifiability criterion but would no longer satisfy the validity criterion, since this requirement is actually fairly subjective, and the test might not capture the initial intention of the requirement. The original intention of the requirement is to have an advanced user interface that stays up to date with current technologies. The same holds true for all three of these non-functional requirements. They are partly subjective requirements that cannot be clearly covered by test cases. Nevertheless, these requirements represent important insights and demands for the developed artifact. For this reason, the three requirements “Simplicity”, “Openness of Platform”, and “Advanced User Interface” are included as valid requirements, but no test cases can be included for them. Nevertheless, they are taken into account in the development of the artifact and finally evaluated as best as possible. Regarding the other requirements, by successfully setting up tests for them, the verifiability criterion can thus also be confirmed. Thus, the established requirements meet the criteria of validity, consistency, completeness, realism, and verifiability and are thus valid requirements for the artifact.

5 Design and Development of the Artifacts

The artifact, intended to improve experimental research in behavioral data analytics, is to be implemented as a software component or application. In this step, a suitable architecture is developed based on the established requirements. To achieve this, the application is first fundamentally conceptualized. This involves analyzing possible technologies and the corresponding processes. The architecture is then implemented in practice based on this initial conceptualization.

5.1 Process Conceptualization

To effectively implement the requirements for the application, the processes described by the requirements and validated by the test cases are represented in the following swim lane diagrams. The goal is to illustrate the individual processes in a technology-independent manner. The reason for this is that the individual processes are initially presented in a generalized form, irrespective of technological restrictions or limitations. Swim lane diagrams depict processes by showing business activities in relation to each other and how they are associated with each other (Caudle, 2009). A significant portion of the final functionalities can be divided into three different categories: requirements related to functionalities and data pertaining to the experiment itself, the individual participants, and the interaction between these participants and the application through the User Interface (UI). The three lanes, “Experiment Data”, “Participant Data”, and “User”, represent these three categories and data sources. The “User” lane shows the activities performed by the test subjects, the “Experimental Data” lane shows all activities associated with the experimental setup, and the “Participant Data” lane shows all data and activities related to the test subjects.

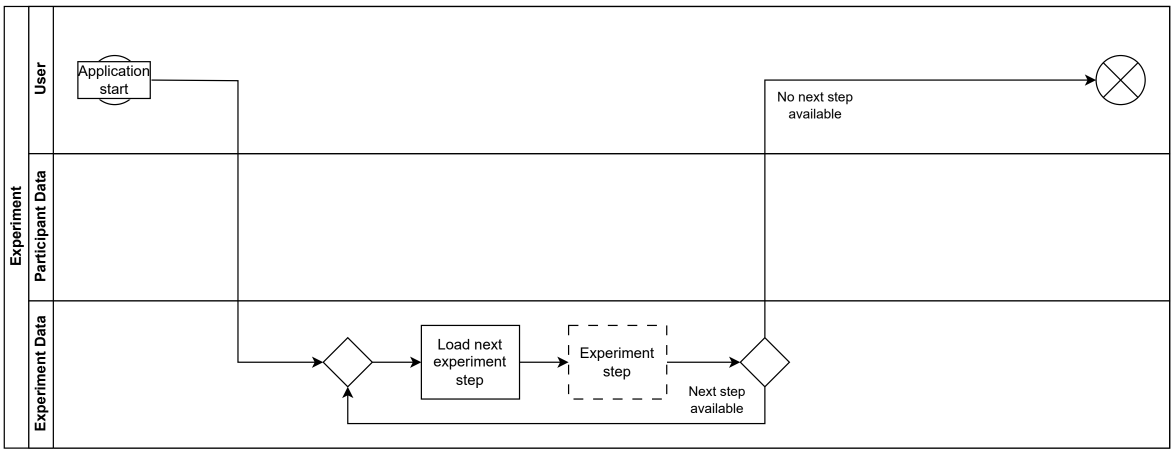


Figure 2: Experiment - Swim Lane

Figure 2 shows the basic structure of any experiment. The whole experiment is divided into different experiment steps, which are executed one after the other until the experiment is completed. The application is started first, and then the first step of the experiment is executed. This could be, for example, an information message for the participants. After that, the next step is executed if another step is available. If no further step is available, the experiment ends. This simple abstraction of an experiment concentrates on the essentials and thus allows for the most flexible and adaptable construction kit for any experiment. According to the requirements, a certain number of standard “experiment steps” are developed for the artifact. These “standard steps” represent functionalities that are common to every imaginable experiment. In addition to this, the possibility should exist to create custom “experimental steps” in order to extend and customize the experiment. The following five “standard experiment steps” based on the functional requirements are illustrated using swim lane diagrams. The dashed box “Experimental Step” in Figure 2 therefore represents one of the “standard steps” from Figures 3, 4, 5, 6, and 7. For this reason, the mentioned graphics also begin and end in the “Experimental Repository” lane.

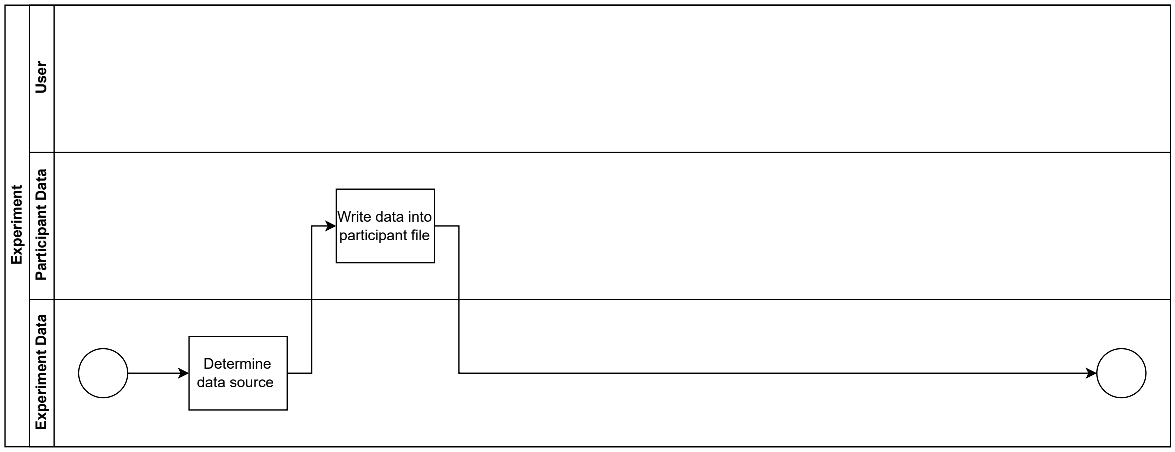


Figure 3: Data Input Step - Swim Lane

Figure 3 represents the process step of reading data from one or more sources. The assumption is that this data must first be processed or standardized before it can be meaningfully assigned to the participants and used within the experiment.

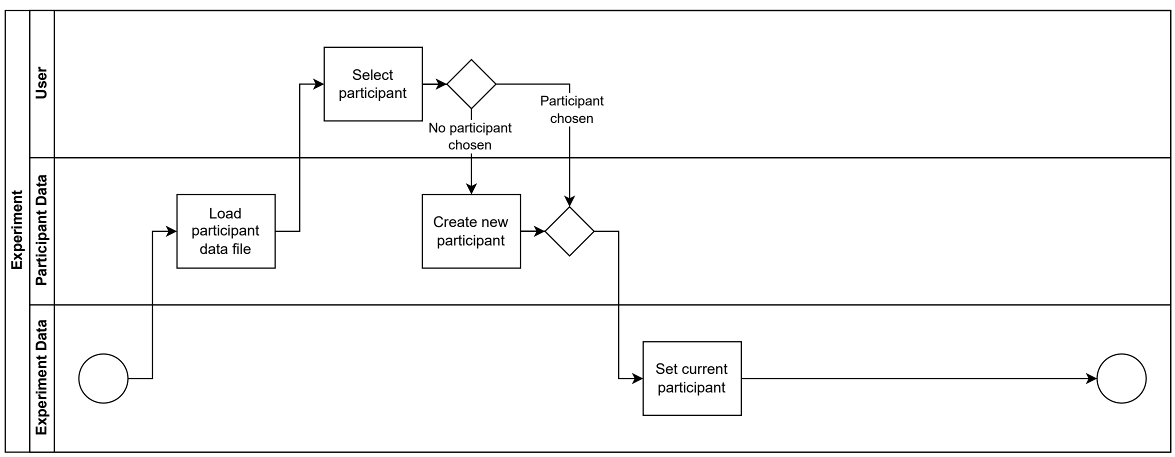


Figure 4: Choose Test Subject Step - Swim Lane

Figure 4 enables the mapping between the test subject and their Identification (ID), which will be used for the rest of the experiment.

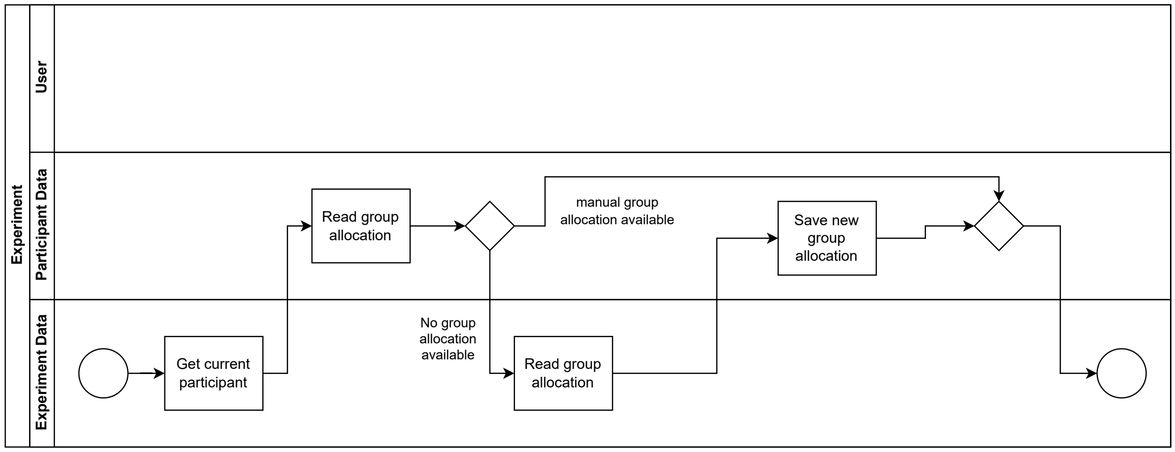


Figure 5: Group Allocation Step - Swim Lane

Figure 5 represents the process of assigning individual participants to groups. Participants can be divided either according to a predefined group, arbitrarily, or according to certain criteria.

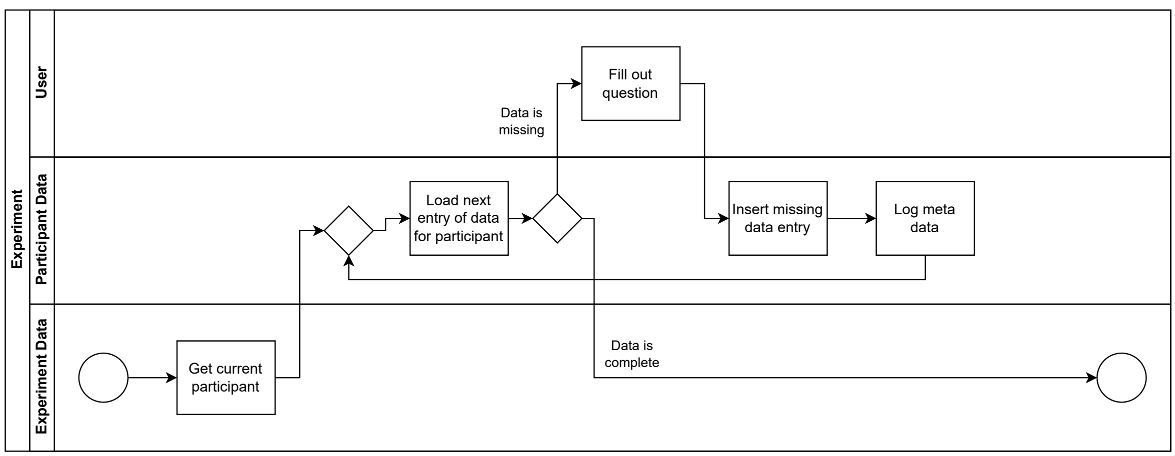


Figure 6: Questionair Step - Swim Lane

Figure 6 is used to complete missing information or data about the participants. For this purpose, test subjects are asked questions based on missing data entries corresponding to their ID. This process is conceptualized similarly to the general experiment setup, so that any number of questions or no questions at all are displayed for completion, depending on the number of missing data points about a test subject. An example would be a participant whose age is missing in a dataset. An input field is automatically displayed for the participant to fill in the missing information about themselves. If all data for a test subject is complete, nothing is displayed.

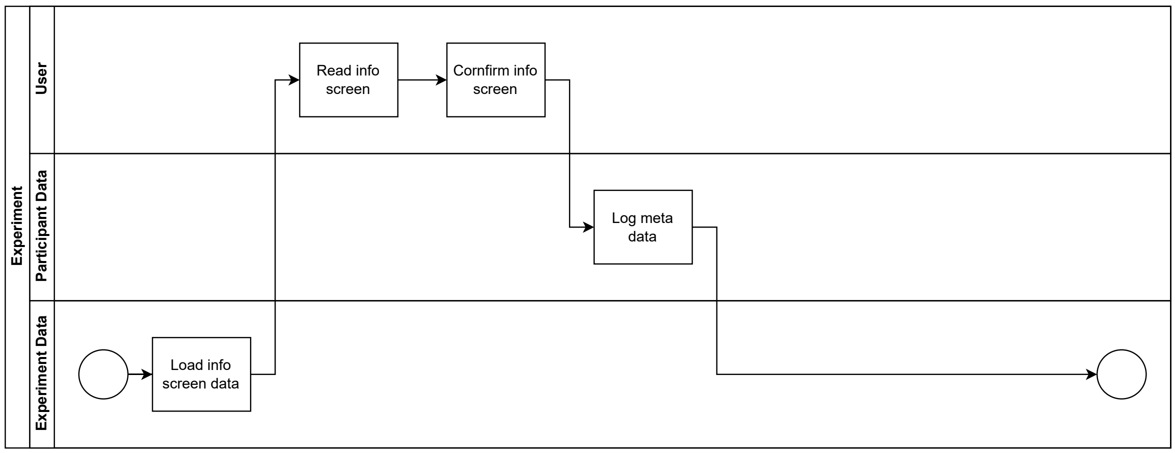


Figure 7: Info Screen Step - Swim Lane

Figure 7 displays information to the test subjects and can be used both as a means of notification during an experiment or just as a representation of information.

5.2 Technology Selection

In order to select appropriate technologies for the implementation of the application, the established requirements must be considered. It can be assumed that all functional requirements can be implemented by any modern programming language that is Turing complete. The non-functional requirements N2.1 (Evaluation of Data), N2.2 (Visualize Final Data), N6.1 (Multi-Source) are represented by the availability of various interfaces. Furthermore, the two non-functional requirements N1.2 (Time-Flexibility) and N5.1 (Monitoring of Study) are not relevant to the technology selection as these refer to the way the application is implemented and not its technological nature. This leaves requirements N1.1 (Distant Communication), N3.1 (Simplicity), N4.1 (Reusable), N4.2 (Interoperability), N4.3 (Openness of Platform), N7.1 (Advanced User Interface), and the aforementioned availability of interfaces as requirements for selecting a suitable technology. In the following sections, technologies and tools are presented that are intended to meet these requirements.

5.2.1 Android and Android Studio

Android is an open-source operating system for mobile devices, which was first announced by Google in 2013. To date, Android has achieved a market share of over 90% in the mobile sector and is the most used operating system overall, being used in almost every second device (statcounter, 2023, E. Richter, 2019). The standard development environment to develop Android applications is the Integrated Development Environment (IDE) Android Studio, which supports a wide range of developer tools and functionalities. As an open-source project and, due to its high distribution on various devices, Android suits the N4.3 “Openness of Platform” and N4.2 “Interoperability” requirement (E. Richter, 2019). Android applications are programs designed for touch inputs, specifically for mobile devices. However, as a widely used open-source operating system, Android also supports other input options, advanced network capabilities, and a variety of interfaces and extensions (E. Richter, 2019). Thus, it can be assumed that the interface and N1.1 “Distant Communication” requirements can be fulfilled through the usage of Android. The two programming languages that can be used to develop these Android applications are Java and Kotlin. Android applications can be tested either directly on an Android device or on a variety of virtual devices integrated into Android Studio, which further facilitates the development of said applications (E. Richter, 2019). As a mobile operating system, one of Android’s main focuses is the UI as an input function, in addition to network functionalities and a variety of interfaces. Android also has a wide range of design guidelines, interface functionalities, and is updated at regular intervals (Statista, 2023, E. Richter, 2019). Due to these regular updates, the widespread use of Android, the focus on UI-intensive use cases, and the broad functionalities like network capabilities and support for different interfaces, it can be assumed that an application developed in Android fulfills the aforementioned requirements.

5.2.2 Java

Since 2009, Java has been part of the product portfolio of Oracle Corporation. Java is an object-oriented programming language, making it a universally applicable and robust programming language (Ullenboom, 2017). Unlike many other programming languages, one of the special features of Java is its platform independence. Most programming languages use a compiler or interpreter to translate program code into bytecode, which varies depending on the hardware and can only be executed on the appropriate processors. Java avoids this limitation by first having a compiler translate the Java program code into bytecode, which is then executed via an interpreter in a virtual environment called Java Virtual Machine (JVM). In this way, Java code can theoretically be executed on any system (Ullenboom, 2017). This makes Java not only a programming language but also a runtime system, which is made clear by the naming of the Java Platform by Oracle. The Java Platform supports, besides Java itself, the execution of some other programming languages, such as Kotlin (kotlinlang.org, 2023). Due to this fact, Java is especially suitable for the implementation of the artifact based on the requirements N4.1 (Reusable) and N4.2 (Interoperability). Java also supports a variety of programming concepts through standard libraries. These include data structures, string processing, date/time processing, graphical interfaces, input/output, network operations, threads, and more (Ullenboom, 2017), which ensures the N1.1 (Distant Communication) requirement and the availability of interfaces. The Java runtime environment also enables fast code execution and comes with various utilities such as a garbage collector and output name handlers. The syntax of Java is generally considered to be easy to understand and beginner-friendly (Ullenboom, 2017), fulfilling the requirement N3.1 (Simplicity). In addition to technical aspects, Java is also open-source, extremely widespread, popular, and a variety of literature is available for it (Ullenboom, 2017), which generally indicates openness of the platform (Requirement N4.3 Openness of Platform) and convenience in developing code (N3.1 Simplicity). Disadvantages of Java, which are also mentioned for the sake of completeness, mainly relate to specific platform-dependent use cases. Since Java was developed as a general-purpose programming language and is platform-independent, it is difficult to access hardware or drivers directly (Ullenboom, 2017). However, these limitations are specific and therefore irrelevant in the context of this thesis.

5.3 System Architecture Development

Overall, it can be summarized that the non-functional requirements, which consider properties of the system, are completely fulfilled by using the Android operating system in combination with the Java programming language. In general, most of the requirements could already be met by both Android and Java alone, so the combination of the two provides a solid foundation for meeting the requirements. However, as already noted in section 4.2, the fulfillment of the non-functional requirements is by no means a binary state, but can be partly of a subjective nature, which is why no test cases could be set up for some requirements. Taking the above arguments into account, and especially the extremely wide distribution of the Android operating system, the technical combination of Android and Java for the implementation of the artifact is considered an appropriate choice. For this reason, the artifact is implemented in the form of a mobile Android application. The Android API 24 (Android 7.0, Nougat) is used for this implementation. This Android version was chosen because it has all the interfaces and functionalities needed for the development of the artifact, and an application developed on this version runs on about 95.4% of all Android devices. At the same time, the application can easily be ported to a new version if newer features are needed for the implementation of an experiment (Google, 2023a).

The recommended architecture for an Android application consists of three layers: the UI layer, the domain layer, and the data layer. The UI layer displays application data and the application itself to the user. The domain layer is optional. It is used for abstraction and structuring of the data layer and is recommended above all if the application is to represent complex business cases or the application must be designed to be reusable (Google, 2023a). Due to the requirements N4.1 (Reusable), a domain layer is therefore used in the conceptualized architecture. Classes in this layer are usually called use cases or interactions and always represent a single functionality. For example, the output of the time could be a functionality that must be used by several components. This functionality would then be represented by a *GetTimeUseCase* class. The data layer contains the data and business logic of the application (Google, 2023a). This layer defines to what extent data is processed, modified, or stored. It is also divided into two parts: the repositories and the data sources (Google, 2023a). The repositories are responsible for exposing the data to the rest of the application, centralizing changes to the data, and resolving conflicts between data sources. A repository can contain zero or multiple data sources. The repositories also contain the business logic and abstract the data sources from the rest of the application. Each data source is represented by one data source class, which is the link between the system for data operations and the application. Sources for these data sources could be a file, a network, or a local database (Google, 2023a). In the following, the concrete implementation of the architecture of the artifact is presented. The technical components are divided into the UI layer, the domain layer, and the data layer.

5.3.1 Data Layer

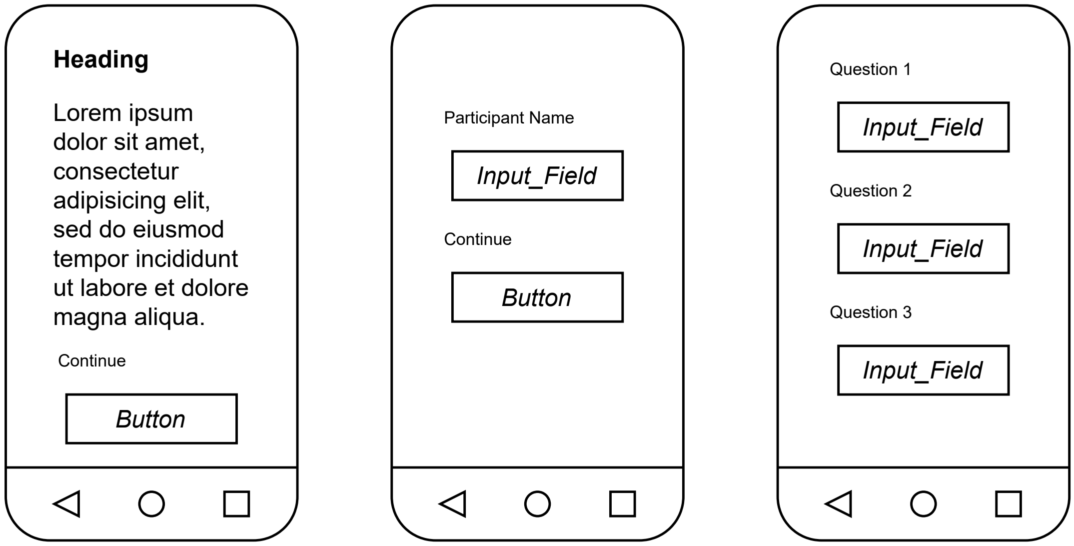
The individual data sources are masked from the rest of the application by the so-called repositories. In principle, however, a large number of different data sources can be used. Among others, local files, database systems, or network storage. The number of data sources can vary from none to any number. The individual data sources depend on the respective experiment. For this reason, a simple local Comma-Separated Values (CSV) file is used as a placeholder for different data sources. This is done for the sake of simplicity; in an actual experiment, this placeholder file can then be replaced by any other combination of data sources. The data layer as a whole corresponds to the *Participant Data* and *Experiment Data* lanes from the previously defined processes in the swim lane diagrams 2, 3, 4, 5, 6, and 7. These lanes contain the business logic for elements associated with the experiment itself and the test subjects, respectively. For this reason, and in order to clearly abstract the standard data sources of the artifact from potential data sources that get implemented within experiments, two repositories called *ParticipantRepository* and *ExperimentRepository* are implemented. The Android development scheme for the application specifies that all processing and unification of data should take place exclusively in the data layer. For this reason, all processes and activities involving data sources must be implemented in this layer. This concerns only the unification and standardization of data sources. Logic and the modification of data are performed in the domain layer. For this reason, the “Data Input Step” illustrated in swim lane figure 3, which includes the reading and standardization of data, is not implemented as a single process step of an experiment but in the course of the development of the data layer. Thus, this step is the only one which must be implemented in the course of the development of the experiment. In the actual technical implementation, the data of the experiment and the participants are passed on to the repositories above it and the domain layer in the form of so-called entity classes. The corresponding data and properties are contained as attributes in these entity classes and can be modified via getter and setter methods. The persistence of the data is ensured by the singleton principle, which is applied to the respective repositories.

5.3.2 Domain Layer

The domain layer is an optional layer that encapsulates complex business logic or logic that needs to be reused frequently (Google, 2023a). Since it is not known to what extent the individual elements will be used in an experiment, and a special focus is placed on the reusability of the individual components, this optional layer is implemented in the artifact. Further advantages resulting from the use of a domain layer are the avoidance of duplicated code, the improved readability of the architecture, an improvement in the testability of the application, and the avoidance of large classes by splitting the tasks (Google, 2023a). The domain layer classes are accessed in the same way as repositories of the data layer are accessed by the UI. A simple unrelated example of a domain layer class would be to request the addresses of the best authors of the year. In this example, there would be two data sources with repositories, one for authors and their addresses and another one containing the best-selling books of the year. A domain layer class would hide access to these two repositories and the complex logic of determining the addresses of the best authors of the year from the rest of the application. To keep the classes of the domain layer simple, it is advised that each class should contain only a single functionality and should not contain mutable data (Google, 2023a). These individual functions are also called use cases. Use cases (Domain Layer classes) can call each other and can be hierarchically dependent on each other within the domain layer as needed. Since one use case is supposed to represent one function at a time, a domain layer class (or use case) is created for each function or action that appears in one of the process swim lane diagrams in the *Experiment Data* and *Participant Data* lanes from section 5.1. Entire processes (swim lanes) that do not contain user input are also illustrated by these domain layer use case classes. This case applies to the process in swim lane diagram 5, which is why it is modeled as a use case in the Domain Layer. A detailed view of the derived use cases is included in appendix C. For the implementation of the individual use cases, a Java class is created for each case, which contains the corresponding logic, and the use case calls corresponding interfaces through methods.

5.3.3 User Interface Layer

The concept of a graphical UI is implemented in Android via so-called activities, which represent special classes that contain a temporary data state and the UI elements the user interacts with. While the start of an application in regular Java applications takes place via a *main()* method, an Android application initiates code via these activities. The Android developer documentation describes an activity as "[...] entry point for an application's interaction with the user" (Google, 2023a). The required UI elements of the application are generated in these activities. An activity corresponds to a single screen of an application. An application can contain several different UI screens and thus several different activities. Activities can also call other activities and navigate to them as desired. An application is therefore a sequence of different activities that represent different screens. Usually, an activity serves as the entry point to the application. This activity, also called the main activity, is the first screen that is shown when the application is started. Although activities together form a complete application, activities are only superficially connected to each other. Basically, each activity is a replaceable, self-contained component that can be called in any order. This reusability and separation make activities the perfect foundation to implement the individual "experiment steps" that contain user interactions described in section 5.1. Based on this fact and the fact that an activity corresponds to a UI, a separate activity is designed for each process step from section 5.1 that contains user interactions. The three experiment steps, which include user inputs, are the *Choose Test Subject*, *Questionnaire*, and *Info Screen* steps, depicted in Figures 4, 6, and 7. Figure 8 depicts these three use cases as a UI prototype. Figure 8b represents the process step of choosing a test subject depicted in figure 4. Figure 8c represents the process step of letting the test subject answer questions depicted in Figure 6. Figure 8a represents the process step of showing information to the test subject depicted in Figure 7. The three UI sketches show the basic division of UI elements to which the actual implementation of the activities is oriented. Furthermore, other custom steps from an experiment that contain user interactions would also be implemented using activities. In this way, the reusability and separation of the individual experiment steps are guaranteed. The individual activities only hold a temporary dataset of information that is displayed or required in the UI. As already explained, all other business logic is outsourced to the domain layer. The domain layer can



(a) Info Screen (b) Choose Test Subject (c) Questionair

Figure 8: User Interface Prototype of Artifact

be called and consumed by all activities at any time.

After the basic UI has been designed, the navigation between the individual steps must be conceptualized and implemented. Generally, the sequence of screen calls is defined by the logic within each activity. This enables each activity to call every other activity at any given moment. The processes for which the designed screens in Figure 8 are intended are all non-time-sensitive, which is why they all contain a “continue” button to navigate to the next screen. In principle, however, it would also be possible to navigate to the next screen based on a time limit or other logic. The exact order of the screens is not stored in the code of the respective activities, as is usual for Android applications, but in the data of the experiment. As conceptualized in swim lane diagram 2, this should increase the reusability of the application and reduce the time required to set up an experiment. This is implemented by storing the activity class names and the desired order in the experiment data. A click on the button calls the GetNextExperimentStepUseCase use case, which contains the next activity to be opened as a string. This will then initiate and open a corresponding activity class. The code for this activity call is included in appendix C. The navigation to experiment steps that do not contain any user interaction and are therefore not bound to any activity, but are in the domain layer, takes place directly before the call of the next activity. An example of this would be an activity calling one or more domain layer steps and then the next activity when the continue button is pressed.

5.4 Consolidated System Architectural Summary

In summarized form, the resulting architecture is shown in Figure 9 with all layers and how they relate to each other. The green-colored use cases and data sources represent elements that relate exclusively to data and processes concerning the participants, the blue-colored use cases and data sources represent elements that relate exclusively to the experiment, and the cyan-colored parts represent elements that deal with both participant and experiment data and processes.

Overall, the technical implementation of the artifact can be summarized as follows: The processes that the artifact needs to support are represented by the processes depicted using swim lane diagrams in section 5.1. The "User" lane of these diagrams represents the interaction of the user with the artifact. Each activity that contains UI interactions is, therefore, represented by an Android activity. The individual events in the "User" lane are represented by Android UI elements. Each process that does not contain any user interactions is represented by coding within a domain layer use case, with the exception of the "Data Input Process" (Swim Lane Figure 3), which exclusively handles the preparation of a data source and is, therefore, represented by a repository within the data layer. The events within the "Experiment" and "Participant" lanes of the "standard experiment steps" are represented by a single use case and thus form the domain layer. The individual use cases link either directly to the data repository or to other use cases. The actual data resides in various data sources and is abstracted from the rest of the application by the data layer repositories "ExperimentRepository" and "ParticipantRepository". The complete architecture, including the UI, domain, and data layer, is represented in Figure 9.

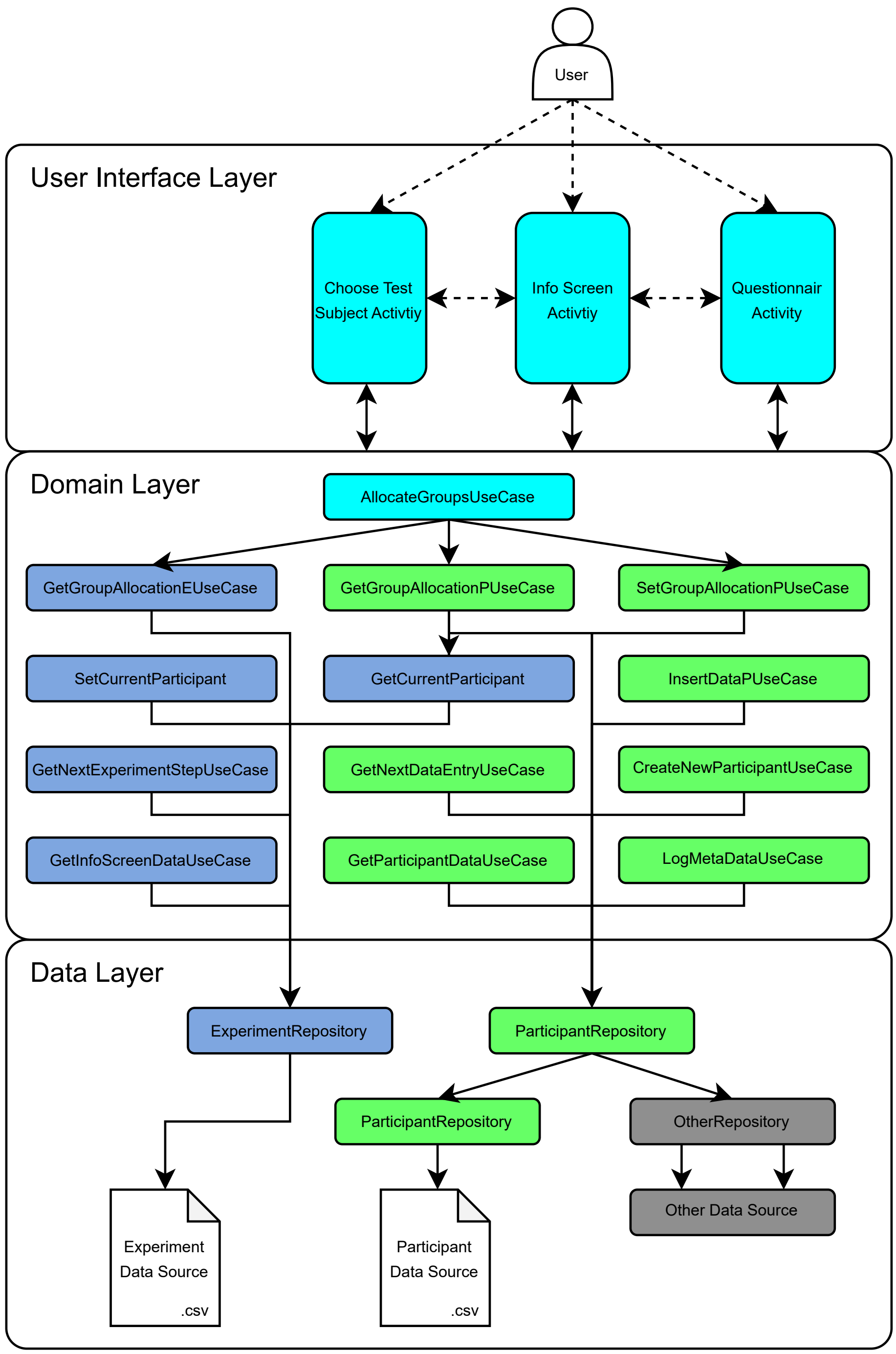
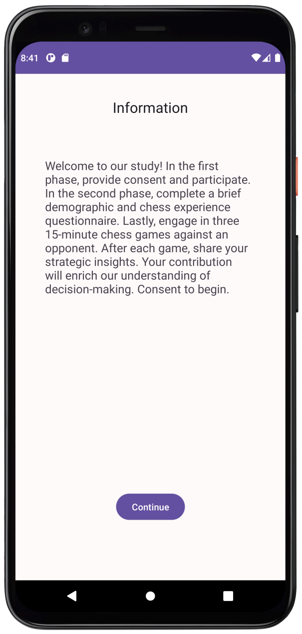


Figure 9: Complete Architecture of the Artifact

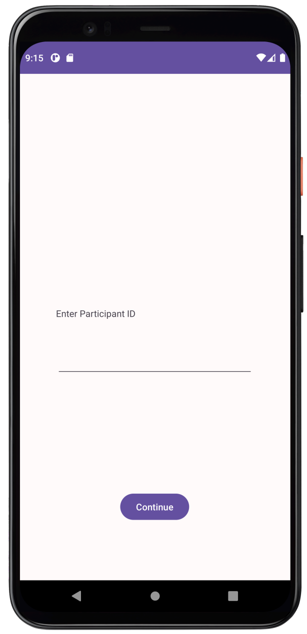
6 Demonstration of the Artifact

In order to validate the conceptualized artifact and demonstrate its functionality, an experimental setup from a real study in the field of data analytics is exemplarily implemented with the conceptualized artifact. For this purpose, the study of S. Krakowski et al. (2022) is implemented using the conceptualized artifact. The 2022 study investigates changes in the origin of competitive advantages through the rise of AI. To achieve this, they employ a case study on chess tournaments to explore the impact of AI on competitive advantage (Krakowski et al., 2023a). Although the methodological experimental design of this article is perfectly fine, the test could have been conducted as an experiment. According to B. Gniewosz (2011), an experiment would have been the preferred methodology in this field (Gniewosz, 2011), and the authors themselves note that further research in this area should be conducted through experiments (Krakowski et al., 2023a). Therefore, this study is a perfect candidate to validate the conceptualized artifact. Nonetheless, it must additionally be noted that the artifact's implementation of the study is merely a sample implementation. Figure 10 shows the four UI activities of the final artifact implementing the study of S. Krakowski et al. (2022). For this purpose, a custom activity (Subfigure 10d) was created that contains a chess game playable between two people. Figures 10a, 10b, and 10c display the conceptualized standard functionalities of the artifact. The individual fields visible in the UI are dynamically populated with data from the corresponding data sources. The experiment data specifies the order of the screens as follows: (1) Information Screen, (2) Participant Selection Screen, (3) Questionnaire Screen, (4) Custom Chess Game Activity. The participant data contains dummy data for five participants who have not filled out any prior information. The application is emulated on a Pixel 4 XL API 30.

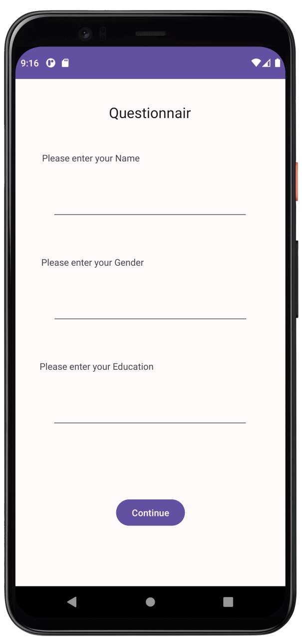
For changing or adding the individual "experiment steps" of the application or modifying the questions within the questionnaire step, only the experiment or participant data needs to be changed, without the need for additional coding, showcasing the simplicity and reusability of the artifact. The same applies to the information screen step. New experiments can build upon the existing use cases in the domain layer and extend them using object-oriented programming. If additional experiment steps involve UI interactions, these can be implemented



(a) Choose Subject Step



(b) Questionnair Step



(c) Info Screen Step

(d) Chess Game

Figure 10: User Interface Implementation

through new Android activities. The name of the activity must then only be stored in the experiment data as a string in order to call it. The full program code can be found in the GitHub repository at: *https:github.comschneemaxmaster_thesis*

7 Evaluation of the Solution

To validate the developed prototype, the previously established test cases are verified for the individual requirements. In addition, the fulfillment of requirements N3.1 (Simplicity), N4.3 (Openness of Platform), and N7.1 (Advanced User Interface), which could not be validated by test cases, is discussed. Subsequently, the usability of the UI of the prototype is assessed via a user interface test.

7.1 Prototype Validation

This section looks at the individual test cases from section 4.3. In summary, all requirements could be verified with the help of the established test cases. Tables 7 and 8 show the corresponding requirements and the associated test cases that prove and verify them. A detailed description of the fulfillment of each of the test cases is included in appendix D. One exception to this are the non-functional requirements N3.1 (Simplicity), N4.3 (Openness of Platform), and N7.1 (Advanced User Interface), which cannot be verified by test cases alone due to their subjectivity. Nevertheless, as already discussed in section 4.3, these requirements represent important specifications for the artifact. For this reason, the fulfillment of these non-functional requirements will be addressed as feasibly as possible without the use of test cases. For this purpose, the experience gained from the implementation of the artifact and the documentation on the technologies utilized are used. Besides the fact that Android and Java are generally considered to be simple, beginner-friendly technologies (Ullenboom, 2017), the best practice architecture that was implemented in the course of this application is a big indicator of the simplicity of the application (Google, 2023a). The individual functions and code modules have been divided into reusable use cases, and all UI activities have been consolidated into Android activities. In addition, the implementation of new custom capabilities for individual experiments has been made extremely easy by the general architecture that was conceptualized. For example, the definition of a new order of “experiment steps” is realized within the corresponding data of the experiment and does not have to be implemented separately by program code. The implementation of custom experiments is also streamlined, which makes it possible for the person performing the experiment to set up their experiment without having to worry about basic tasks like group allocation or data completion. Due to these conditions, the requirement N3.1 (Simplicity) is considered to be fulfilled by the artifact.

The last non-functional requirement which could not be validated by a test case alone is the requirement N7.1 (Advanced User Interface). The conceptualized Android application utilizes MaterialUI developed by Google for the UI elements. Google claims that MaterialUI is distinguished from other UI technologies by its responsive design, motion and animation, consistency, accessibility, cross-platform support, and other design-related features. Furthermore, MaterialUI is characterized by active development on the part of Google and is regularly provided with updates. Thus, the underlying UI technology that was used for the development of the artifact represents a UI technology that will deliver a modern UI as of 2023, and the regular updates by Google can be assumed to guarantee this circumstance in the long run. In summary, the N7.1 requirement is considered to be fulfilled as well as possible through the use of the MaterialUI interface in combination with its update guarantee by Google (Google, 2023c, Google, 2023a).

In conclusion, all requirements could be verified using and fulfilling the corresponding test cases. In addition, it was possible to show argumentatively on the basis of various sources that the non-functional requirements N3.1 (Simplicity), N4.3 (Openness of Platform), and N7.1 (Advanced User Interface) are also fulfilled and verified. Hence, all requirements for the artifact are fulfilled and verified.

7.2 Application Performance and Usability

After all the requirements for the artifact have been fulfilled and the functionality of the artifact has been demonstrated, this section briefly validates the UI of the application. As described in this thesis, the UI of the artifact was implemented based on the different “experiment process steps” that require user inputs. For this purpose, the respective activities in the processes were used and implemented on the individual screens by the corresponding

Requirement	Testcase	Fulfilled
Information		
F1.1 Displaying Information	T1	✔
F1.2 Debrefing Info	T1	✔
Data Collecting		
F2.1 Participant Data	T2, T5	✔
F2.2 Meta-Data	T3, T8	✔
F2.3 Post-Interview	T2, T5	✔
Pre-Loading		
F3.1 Pre-Loading Data	T4	✔
F3.2 Selecting Data	T4	✔
Experiment Setup		
F4.1 Additional Logic	T5	✔
F4.2 Participant Input	T2, T5, T7	✔
F4.3 Proactive System	T2, T5	✔
Groups		
F5.1 Different Groups	T6	✔
F5.2 Communication of Groups	T7	✔
F5.3 Targeted Assignment	T6	✔
F5.4 Random Assignment	T6	✔

Table 7: Fulfillment of Functional Requirements

Requirement	Testcase	Fulfilled
Time-space non-reliance		
N1.1 Distand Communication	T7	✔
N1.2 Time-Flexibility	T7	✔
Data Postprocessing		
N2.1 Evaluation of Data	T8	✔
N2.2 Vizualize Final Data	T8	✔
Simplicity		
N3.1 Simplicity		
Reusable and Interoperable		
N4.1 Reusable	T9	✔
N4.2 Interoperability	T10	✔
N4.3 Openness of Platform		
Monitoring		
N5.1 Monitoring of Study	T8, T11	✔
Pre-Loading		
N6.1 Multi-Source	T4	✔
Advanced User Interface		
N7.1 Advanced User Interface		

Table 8: Fulfillment of Non-Functional Requirements

UI components of an Android application using the MaterialUI design. The UI thus adheres to the proven and tested UI concepts of Google's MaterialUI design philosophy. Nevertheless, this part of the thesis is intended to test the rough layout and usability of the artifact's UI. The goal of this section is not to identify the best possible or most beautiful UI, but to verify that the UI used is a suitable interface for the artifact and its processes. For this purpose, a clickable mock-up prototype is built using screenshots of the artifact. The prototype feature of SAP Build.me was used for this purpose (SAP, 2023). This prototype was then clicked through by business informatics students working at a large German DAX software company. Their task simply being to click through the prototype. Both the clicks and the time needed to navigate through the individual screens and the prototype were measured. The goal of this test is to verify that the UI of the prototype is fundamentally usable and to show any flaws or disadvantages of it. The test would also show potential for improvement if some participants did not manage to click to the end of the prototype, took an unusually long time to do so, or clicked several times at a point that was not intended to be interacted with. In total, 25 individual participants took part in the study. The participants took part in the test anonymously.

The results of the test do not indicate any negative design decisions. The average time it took the participants to click through the prototype is reasonable, and the recorded clicks do not show any unusual hotspots or anomalies. A full list of the anonymized participants and an overview of the clicks they did is included in appendix D. In general, the UI appears to be usable and did not present any major or new challenges to the participants of the test. Once more, it should be pointed out that this test only serves to verify that the activities from the processes in connection with Google MaterialUI result in a meaningful and usable UI and that the usability properties claimed by Google about MaterialUI can be fulfilled in the context of the artifact. Nevertheless, the test shows that the UI of the artifact is usable and, in combination with the claims made by Google about MaterialUI, is considered sufficiently functional for the artifact.

8 Conclusion

The objective of this thesis was the development of an artifact that would improve experimental research in behavioral research in Data Analytics. This was ensured by the following three objectives: (1) the review of prior research on data analytics and its methodological procedures, (2) the development of an artifact that improves the research process in the field of behavioral research in data analytics, and (3) the validation of the artifact through the exemplary realization of a study in said field utilizing said artifact. These objectives were successfully achieved. Not only was an artifact developed that streamlines the execution of experiments and thus contributes to more efficient and effective research in the area of data analytics, but it was also validated by implementing a sample study. Moreover, the current state of knowledge in the field of behavioral research in data analytics was analyzed, and gaps in research on behavioral and experimental research in data analytics were identified. Therefore, the developed artifact in the form of an Android application not only serves as a framework to better conduct experiments in the area of behavioral research in data analytics, but the findings and requirements that were identified through it also serve as an important analysis of the state of research in the area of experimental setups. The three objectives that this thesis was intended to fulfill could thus be met.

The practical findings and implications also serve as a counterpart to the theoretical knowledge that can be found in various journals or books and show the challenges and particularities of how these theoretical concepts are applied in practice. To give one example, this highlights the challenge of different group assignments or the collection of participant feedback in practice. As already described in Krakowski S. et al. (2022) paper, technologies such as ML, AI, or data analytics already represent alternatives to human decision-making, and the artifact developed in this thesis allows these "black-box" technologies to be better understood and thus used, which has a direct impact on the success of a company (Krakowski et al., 2023b). Amankwah-Amoah and Adomako examine in their study big data usage on business failure and come to the conclusion that the correct use of data analytics and big data is an important factor in the success of a company. An improved state of research made possible by the artifact that was created within this thesis could therefore not only lead to new scientific findings but also have a direct influence on the success of companies.

The theoretical implications of research are very comprehensive, especially for the improved execution of experiments. Not only can new research be conducted more efficiently and effectively, but previous results and findings can be more easily verified and re-examined. The use of the developed artifact would make an essential contribution to the creation of new knowledge and the consolidation of already existing knowledge. Accordingly, the artifact should be used in the future to conduct experiments in the field of data analytics as well as to re-evaluate already conducted studies with the help of an experimental setup. The limitations of the developed artifact are mainly related to technological limitations. Due to the technology-independent development of the basic processes, the results of the work represent generally valid research. Nevertheless, the technological implementation of the developed artifact should be re-evaluated at a later point in time, should new technologies become available that might better reflect the collected and generally valid requirements.

In conclusion, in the course of this thesis, it was possible not only to find a literature gap in the area of behavioral research in data analytics and the general execution of experiments but also to identify generally applicable requirements for an application for the execution of experiments based on various sources. In the process, new knowledge was created and existing knowledge was validated. At the same time, an artifact in the form of an Android application was developed and evaluated, which can be used to conduct experiments in the field of data analytics more efficiently and effectively. Thus, this work not only contributes to the current state of research but also enables future researchers to better create new knowledge and consolidate existing knowledge through the developed artifact.

Abbasi, A., Sarker, S., & Chiang, R. H. L. (2016). Big data research in information systems: Toward an inclusive research agenda. *Journal of the Association for Information Systems*, *17*(2).

Abramson, C., Currim, I. S., & Sarin, R. (2005). An experimental investigation of the impact of information on competitive decision making. *Management Science*, *51*(2), 195–207.

Alain Abran, James W. Moore. (2004). *Swebok: guide to the software engineering body of knowledge*. IEEE Computer Society.

Allen, T. J., & Cohen, S. I. (1969). Information flow in research and development laboratories. *Administrative Science Quarterly*, *14*(1).

Amankwah-Amoah, J., & Adomako, S. (2019). Big data analytics and business failures in data-rich environments: An organizing framework. *Computers in Industry*, *105*.

Bag, S., Gupta, S., & Wood, L. (2022). Big data analytics in sustainable humanitarian supply chain: Barriers and their interactions. *Annals of Operations Research*, *319*(1), 721–760.

Balzert, H. (Ed.). (2011). *Lehrbuch der softwaretechnik: Entwurf, implementierung, installation und betrieb*. Spektrum Akademischer Verlag. https://doi.org/10.1007/978-3-8274-2246-0

Barrett, M., & Oborn, E. (2010). Boundary object use in cross-cultural software development teams. *Human Relations*, *63*(8), 1199–1221. https://doi.org/10.1177/0018726709355657

Bhatti, S. H., Hussain, W. M. H. W., Khan, J., Sultan, S., & Ferraris, A. (2022). Exploring data-driven innovation: What's missing in the relationship between big data analytics capabilities and supply chain innovation?. *Annals of Operations Research*, 1–26.

BIS Research. (2018). Umsatz mit big data im bereich healthcare weltweit nach anwendung in den jahren 2016 und 2025 (in milliarden us-dollar). https://de.statista.com/statistik/daten/studie/997352/umfrage/umsatz-mit-big-data-im-bereich-healthcare-nach-anwendung/

Bitkom. (2017). Welches sind die wichtigsten it-trends des jahres 2017? https://de.statista.com/statistik/daten/studie/675726/umfrage/die-wichtigsten-trends-in-der-itk-branche/

Bitkom. (2018). Umsatz mit big-data-lösungen in deutschland in den jahren 2016 und 2017 und prognose für 2018 (in milliarden euro). https://de.statista.com/statistik/daten/studie/257976/umfrage/umsatz-mit-big-data-loesungen-in-deutschland/

Carlile, P. R. (2002). A pragmatic view of knowledge and boundaries: Boundary objects in new product development. *Organization Science*, *13*(4), 442–455. https://doi.org/10.1287/orsc.13.4.442.2953

Caudle, G. (2009). *Streamlining business requirements : The xcellr8 approach* (1st edition).

Chakraborty, I., Hu, P. J.-H., & Cui, D. (2008). Examining the effects of cognitive style in individuals' technology use decision making. *Decision Support Systems*, *45*(2), 228–241.

Chen, D., Schonger, M., & Wickens, C. (2016). Otree—an open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance*, *9*. https://doi.org/10.1016/j.jbef.2015.12.001

Chen, D. Q., Preston, S. D., & Swink, M. (2021). How big data analytics affects supply chain decision-making: An empirical analysis. *Journal of the Association for Information Systems*, *22*(5). https://doi.org/10.5194/gi-2016-11-RC2

Chen, L., Liu, H., Zhou, Z., Chen, M., & Chen, Y. (2022). It-business alignment, big data analytics capability, and strategic decision-making: Moderating roles of event criticality and disruption of covid-19. *Decision Support Systems*, *161*, N.PAG.

Chou, S.-W., & Chang, Y.-C. (2008a). The implementation factors that influence the erp (enterprise resource planning) benefits. *Decision Support Systems*, *46*(1), 149–157.

Chou, S.-W., & Chang, Y.-C. (2008b). The implementation factors that influence the erp (enterprise resource planning) benefits. *Decision Support Systems*, *46*(1), 149–157.

Columbia Experimental Laboratory for Social Sciences. (2023). *Resources for online experiments*. Retrieved July 17, 2023, from https://celss.iserp.columbia.edu/content/resources-online-experiments

Cross, R. L., & Parker, A. (2004). *The hidden power of social networks: Understanding how work really gets done in organizations*. Harvard Business School Press.

Currie, G., & Kerrin, M. (2004). The limits of a technological fix to knowledge management. *Management Learning*, *35*(1), 9–29. https://doi.org/10.1177/1350507604042281

Czekster, R. M., De Carvalho, H. J., Kessler, G. Z., Kipper, L. M., & Webber, T. (2019). Decisor: A software tool to drive complex decisions with analytic hierarchy process. *International Journal of Information Technology & Decision Making*, *18*(1), 65–86.

Demoulin, N. T., & Coussement, K. (2020). Acceptance of text-mining systems: The signaling role of information quality. *Information & Management*, *57*(1), N.PAG.

Donghyuk, S., Shu, H., Gene Moo, L., Whinston, A. B., Cetintas, S., & Kuang-Chih, L. (2020). Enhancing social media analysis with visual data analytics: A deep learning approach. *MIS Quarterly*, *44*(4), 1459–1492.

Dresch, A., Lacerda, D. P., & Antunes, J. A. V. (2015). Design science research. In *Design science research: A method for science and technology advancement* (pp. 67–102). Springer International Publishing. https://doi.org/10.1007/978-3-319-07374-3_4

Du, Wnyu, Pan, S. L., Xie, K., & Xiao, J. (2020). Data analytics contributes to better decision-making beyond organizational boundaries. *MIS Quarterly Executive*, *19*(2).

Elgendy, N., & Elragal, A. (2016). Big data analytics in support of the decision making process. *Procedia Computer Science*, *100*, 1071–1084. https://doi.org/10.1016/j.procs.2016.09.251

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996a). From data mining to knowledge discovery in databases. *AI Magazine*, *17*(3).

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996b). The kdd process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27–34. https://doi.org/10.1145/240455.240464

Fernandes, J., Machado, R.-J., & Seidman, S. (2009). A requirements engineering and management training course for software development professionals. *Software Engineering Education Conference, Proceedings*, 20–25. https://doi.org/10.1109/CSEET.2009.24

Fink, L., Yogev, N., & Even, A. (2017). Business intelligence and organizational learning: An empirical investigation of value creation processes. *Information & Management*, *54*(1), 38–56.

Fischbacher, U. (2006). Z-tree: Zurich tool box for ready-made economic experiments. *Experimental Economics*, *10*.

Foerderer, J., Kude, T., Schuetz, S. W., & Heinzl, A. (2019). Knowledge boundaries in enterprise software platform development: Antecedents and consequences for platform governance. *Information Systems Journal*, *29*(1), 119–144. https://doi.org/10.1111/isj.12186

Fosso Wamba, S., Queiroz, M. M., Wu, L., & Sivarajah, U. (2020). Big data analytics-enabled sensing capability and organizational outcomes: Assessing the mediating effects of business analytics culture. *Annals of Operations Research*, 1–20.

Fulcher, A. J., & Hills, P. (1996). Towards a strategic framework for design research. *Journal of Engineering Design*, *7*(2), 183–193. https://doi.org/10.1080/09544829608907935

Ghasemaghaei, M. (2019). Does data analytics use improve firm decision making quality? the role of knowledge sharing and data analytics competency. *Decision Support Systems*, *120*, 14–24.

Ghasemaghaei, M., Ebrahimi, S., & Hassanein, K. (2017). Data analytics competency for improving firm decision making performance. *The Journal of Strategic Information Systems*, *27*(1), 101–113. https://doi.org/10.1016/j.jsis.2017.10.001

Ghasemaghaei, M., Hassanein, K., & Turel, O. (2017). Increasing firm agility through the use of data analytics: The role of fit. *Decision Support Systems*, *101*, 95–105.

Giamattei, M., Yahosseini, K., Gächter, S., & Molleman, L. (2020). Lioness lab: A free web-based platform for conducting interactive experiments online. *Journal of the Economic Science Association*, *6*. https://doi.org/10.1007/s40881-020-00087-0

Gniewosz, B. (2011). Experiment. In H. Reinders, H. Ditton, C. Gräsel, & B. Gniewosz (Eds.), *Empirische bildungsforschung: Strukturen und methoden* (pp. 77–84). VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-531-93015-2_6

Goodhue, D. L., Kirsch, L. J., Quillard, J. A., & Wybo, M. D. (1992). Strategic data planning: Lessons from the field. *16*(1).

Google. (2023a). *Android developer guides*. Retrieved July 28, 2023, from https://developer.android.com/guide/

Google. (2023b). *Firebase*. Retrieved July 28, 2023, from https://firebase.google.com/

Google. (2023c). *Materialui*. Retrieved August 28, 2023, from https://m3.material.io/

Guven–Uslu, P., Blaber, Z., & Adhikari, P. (2020). Boundary spanners and calculative practices. *Financial Accountability & Management*, *36*(4), 439–460. https://doi.org/10.1111/faam.12266

Han, S., Datta, A., & Joshi, K. D., Chi, Lei. (2017). Innovation through boundary spanning: The role of it in enabling knowledge flows across technological and geographical boundaries. *International Journal of Knowledge Management*, *13*(4), 90–110. https://doi.org/10.4018/IJKM.2017100105

Heckman, J. J. (2010). Selection bias and self-selection. In S. N. Durlauf & L. E. Blume (Eds.), *Microeconometrics* (pp. 242–266). Palgrave Macmillan UK. https://doi.org/10.1057/9780230280816_29

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105. Retrieved July 16, 2023, from http://www.jstor.org/stable/25148625

Hofmann, H., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software*, *18*(4), 58–66. https://doi.org/10.1109/MS.2001.936219

Holsapple, C., Lee-Post, A., & Pakath, R. (2014). A unified foundation for business analytics. *Decision Support Systems*, *64*, 130–141.

Hornby, A. S. (2015). *Oxford advanced learner's dictonary of current english* (9th). Cornelsen.

IEEE. (1990). Ieee standard glossary of software engineering terminology - ieee std 610.12-1990.

Inman, J. J., Winer, R. S., & Ferraro, R. (2009). The interplay among category characteristics, customer characteristics, and customer activities on in-store decision making. *Journal of Marketing*, *73*(5), 19–29.

Işık, Ö., Jones, M. C., & Sidorova, A. (2013). Business intelligence success: The roles of bi capabilities and decision environments. *Information & Management*, *50*(1), 13–23.

ISO Org. (2019). *Norm iso 9241*. Retrieved September 30, 2019, from https://www.iso.org/standard/77520.html

ISO/IEC 25010 (Ed.). (2011). *Systems and software engineering - systems and software quality requirements and evaluation (square)*. ISO/IEC. Retrieved March 23, 2021, from https://www.iso.org/standard/35733.html

Jha, A. K., Agi, M. A., & Ngai, E. W. (2020). A note on big data analytics capability development in supply chain. *Decision Support Systems*, *138*, N.PAG.

Ji, G., Yu, M., Tan, K. H., Kumar, A., & Gupta, S. (2022). Decision optimization in cooperation innovation: The impact of big data analytics capability and cooperative modes. *Annals of Operations Research*, 1–24.

Johnson, J. P., Lenartowicz, T., & Apud, S. (2006). Cross-cultural competence in international business: Toward a definition and a model. *Journal of International Business Studies*, *37*(4), 525–543.

Kankanhalli, A., Ye, H. (, & Teo, H. H. (2015). Comparing potential and actual innovators: An empirical study of mobile data services innovation. *MIS Quarterly*, *39*(3), 667–682. Retrieved July 6, 2023, from https://www.jstor.org/stable/26629625

Karhade, P., & Dong, J. Q. (2021). Innovation outcomes of digitally enabled collaborative problemistic search capability. *MIS Quarterly*, *45*(2), 693–718. https://doi.org/10.25300/MISQ/2021/12202

Kim, B. J., & Tomprou, M. (2021). The effect of healthcare data analytics training on knowledge management: A quasi-experimental field study. *Journal of Open Innovation: Technology, Market, and Complexity*, *7*(1), 60. https://doi.org/10.3390/joitmc7010060

Klingebiel, R., & Meyer, A. D. (2013). Becoming aware of the unknown: Decision making during the implementation of a strategic initiative. *Organization Science*, *24*(1), 133–153. Retrieved July 6, 2023, from http://www.jstor.org/stable/23362104

Korschun, D. (2015). Boundary-spanning employees and relationships with external stakeholders: A social identity approach. *Academy of Management Review*, *40*(4), 611–629. https://doi.org/10.5465/amr.2012.0398

Kotlarsky, J., Scarbrough, H., & Oshri, I. (2014). Coordinating expertise across knowledge boundaries in offshore-outsourcing projects: The role of codification. *MIS Quarterly*, *38*(2).

kotlinlang.org. (2023). *Concise. cross□platform. fun.* Retrieved July 27, 2023, from http://kotlinlang.org

Krakowski, S., Luger, J., & Raisch, S. (2023a). Artificial intelligence and the changing sources of competitive advantage. *Strategic Management Journal (John Wiley & Sons, Inc.)*, *44*(6), 1425–1452.

Krakowski, S., Luger, J., & Raisch, S. (2023b). Artificial intelligence and the changing sources of competitive advantage. *Strategic Management Journal (John Wiley & Sons, Inc.)*, *44*(6), 1425–1452.

Lancelot Miltgen, C., Popovič, A., & Oliveira, T. (2013). Determinants of end-user acceptance of biometrics: Integrating the "big 3" of technology acceptance with privacy context. *Decision Support Systems*, *56*, 103–114.

Lebovitz, S., Lifshitz-Assaf, H., & Levina, N. (2022). To engage or not to engage with ai for critical judgments: How professionals deal with opacity when using ai for medical diagnosis. *Organization Science*, *33*(1), 126–148.

Lee, G. M., He, S., Lee, J., & Whinston, A. B. (2020). Matching mobile applications for cross-promotion. *Information Systems Research*, *31*(3), 865–891.

Lehmann, D., Fekete, D., & Vossen, G. (2016). *Technology selection for big data and analytical applications* (ERCIS Working Paper No. 27). Münster, Westfälische Wilhelms-Universität Münster, European Research Center for Information Systems (ERCIS). http://hdl.handle.net/10419/156084

Leidner, D. E., & Elam, J. J. (1995). The impact of executive information systems on organizational design, intelligence, and decision making. *Organization Science*, *6*(6), 645–664.

Levina, N., & Vaast, E. (2005). The emergence of boundary spanning competence in practice: Implications for implementation and use of information systems. *MIS Quarterly*, *29*(2).

Levina, N., & Vaast, E. (2006). Turning a community into a market: A practice perspective on information technology use in boundary spanning. *Journal of Management Information Systems*, *22*(4), 13–37. https://doi.org/10.2753/MIS0742-1222220402

Levina, N., & Vaast, E. (2008). Innovating or doing as told? status differences and overlapping boundaries in offshore collaboration. *MIS Quarterly*, *32*(2).

Li, H., Lu, K., & Meng, S. (2015). Bigprovision: A provisioning framework for big data analytics. *IEEE Network*, *29*(5), 50–56.

Li, M. (, Huang, Y., & Sinha, A. (2020). Data-driven promotion planning for paid mobile applications. *Information Systems Research*, *31*(3), 1007–1029.

Liberatore, M. J., & Stylianou, A. C. (1995). Expert support systems for new product development decision making: A modeling framework and applications. *Management Science*, *41*(8), 1296–1316. Retrieved July 6, 2023, from http://www.jstor.org/stable/2632787

Lindgren, R., Andersson, M., & Henfridsson, O. (2008). Multi-contextuality in boundary-spanning practices. *Information Systems Journal*, *18*(6), 641–661. https://doi.org/10.1111/j.1365-2575.2007.00245.x

Lucas, H. C. (1981). An experimental investigation of the use of computer-based graphics in decision making. *Management Science*, *27*(7), 757–768. Retrieved July 6, 2023, from http://www.jstor.org/stable/2630917

Lukyanenko, R., Parsons, J., Wiersma, Y. F., & Maddah, M. (2019). Expecting the unexpected: Effects of data collection design choices on the quality of crowdsourced user-generated content. *MIS Quarterly*, *43*(2), 623–647. https://doi.org/10.25300/MISQ/2019/14439

Lurie, N. H., & Mason, C. H. (2007). Visual representation: Implications for decision making. *Journal of Marketing*, *71*(1), 160–177.

Majchrzak, A., More, P. H. B., & Faraj, S. (2012). Transcending knowledge differences in cross-functional teams. *Organization Science*, *23*(4), 951–970. https://doi.org/10.1287/orsc.1110.0677

Mäkelä, K., Barner-Rasmussen, W., Ehrnrooth, M., & Koveshnikov, A. (2019). Potential and recognized boundary spanners in multinational corporations. *Journal of World Business*, *54*(4), 335–349. https://doi.org/10.1016/j.jwb.2019.05.001

Marchena Sekli, G. F., & de La Vega, I. (2021). Adoption of big data analytics and its impact on organizational performance in higher education mediated by knowledge management. *Journal of Open Innovation: Technology, Market, and Complexity*, *7*(4), 221. https://doi.org/10.3390/joitmc7040221

Mell, J. N., Knippenberg, D., Ginkel, W. P., & Heugens, P. P. M. A. R. (2022). From boundary spanning to intergroup knowledge integration: The role of boundary spanners' metaknowledge and proactivity. *Journal of Management Studies*, *Not yet published*. https://doi.org/10.1111/joms.12797

Minbaeva, D., & Santangelo, G. D. (2018). Boundary spanners and intra-mnc knowledge sharing: The roles of controlled motivation and immediate organizational context. *Global Strategy Journal*, *8*(2), 220–241. https://doi.org/10.1002/gsj.1171

Montgomery, A. L., Li, S., Srinivasan, K., & Liechty, J. C. (2004). Modeling online browsing and path analysis using clickstream data [Copyright - Copyright Institute for Operations Research and the Management Sciences Fall 2004; Document feature - references; tables; graphs; equations; Last updated - 2023-07-03; CODEN - MARSE5]. *Marketing Science*, *23*(4), 579–595. https://www.proquest.com/scholarly-journals/modeling-online-browsing-path-analysis-using/docview/212288093/se-2

Mueller, O., Fay, M., & vom Brocke, J. (2018). The effect of big data and analytics on firm performance: An econometric analysis considering industry characteristics. *Journal of Management Information Systems*, *35*(2), 488–509. https://doi.org/10.1080/07421222.2018.1451955

Mummendey, H., & Grau, I. (2014). *Die fragebogen-methode: Grundlagen und anwendung in persönlichkeits-, einstellungs- und selbstkonzeptforschung*. Hogrefe Verlag GmbH & Company KG. https://books.google.de/books?id=6aYVBAAAQBAJ

Nutt, P. C. (1998). How decision makers evaluate alternatives and the influence of complexity. *Management Science*, *44*(8), 1148–1166. Retrieved July 6, 2023, from http://www.jstor.org/stable/2634692

of Zurich, U. (2023). *Zurich toolbox for readymade economic experiments*. Retrieved July 18, 2023, from https://www.ztree.uzh.ch/en.html

Pawlowski, S. D., & Robey, D. (2004). Bridging user organizations: Knowledge brokering and the work of information technology professionals. *28*(4).

Peffers, K., Tuunanen, T., Gengler, C., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The design science research process: A model for producing and presenting information systems research. *Proceedings of First International Conference on Design Science Research in Information Systems and Technology DESRIST*.

Peng, Y., & Sutanto, J. (2012). Facilitating knowledge sharing through a boundary spanner. *IEEE Transactions on Professional Communication*, *55*(2), 142–155. https://doi.org/10.1109/TPC.2012.2188590

Pil Han, S., Park, S., & Oh, W. (2016). Mobile app analytics: A multiple discrete-continuous choice framework. *MIS Quarterly*, *40*(4), 983–A42.

Po-An Hsieh, J. J., Rai, A., & Xin Xu, S. (2011). Extracting business value from it: A sense-making perspective of post-adoptive use. *Management Science*, *57*(11), 2018–2039.

Popovič, A., Hackney, R., Coelho, P. S., & Jaklič, J. (2012). Towards business intelligence systems success: Effects of maturity and culture on analytical decision making. *Decision Support Systems*, *54*(1), 729–739.

Pour, M. J., Abbasi, F., & Sohrabi, B. (2023). Toward a maturity model for big data analytics: A roadmap for complex data processing. *International Journal of Information Technology & Decision Making*, *22*(1), 377–419.

Qiqi, J., Chuan-Hoo, T., Choon Ling, S., & Kwok-Kee, W. (2019). Followership in an open-source software project and its significance in code reuse. *MIS Quarterly*, *43*(4), 1303–1319.

Redman, T. C. (2008). *Data driven: Profiting from your most important business asset*. Harvard Business Review Press.

Richter, A. W., West, M. A., van Dick, R., & Dawson, J. F. (2006). Boundary spanners' identification, intergroup contact, and effective intergroup relations. *Academy of Management Journal*, *49*(6), 1252–1269. https://doi.org/10.5465/amj.2006.23478720

Richter, E. (2019). *Android-apps programmieren : Praxiseinstieg mit android studio* (2. Auflage 2019.).

Runkler, T. A. (2020). *Data analytics*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-29779-4

Russell, S., Gangopadhyay, A., & Yoon, V. (2008). Assisting decision making in the event-driven enterprise using wavelets. *Decision Support Systems*, *46*(1), 14–28.

Ryder, B., Gahr, B., Egolf, P., Dahlinger, A., & Wortmann, F. (2017). Preventing traffic accidents with in-vehicle decision support systems - the impact of accident hotspot warnings on driver behaviour. *Decision Support Systems*, *99*, 64–74.

SAP. (2023). *Build.me*. Retrieved September 3, 2023, from https://www.build.me/

Seacord, R. C., Plakosh, D., & Lewis, G. A. (2003). *Modernizing legacy systems: Software technologies, engineering processes, and business practices*. Addison-Wesley. Retrieved February 21, 2021, from https://learning.oreilly.com/library/view/modernizing-legacy-systems/0321118847/

Seagate. (2018). Prognose zum volumen der jährlich generierten digitalen datenmenge weltweit in den jahren 2018 und 2025 (in zettabyte). https://de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/

Sharda, R., Barr, S. H., & McDonnell, J. C. (1988). Decision support system effectiveness: A review and an empirical test. *Management Science*, *34*(2), 139–159. Retrieved July 6, 2023, from http://www.jstor.org/stable/2632057

Sharma, R., Mithas, S., & Kankanhalli, A. (2014). Transforming decision-making processes: A research agenda for understanding the impact of business analytics on organisations.

European Journal of Information Systems, *23*(4), 433–441. https://doi.org/10.1057/ejis.2014.17
Simon, H. A. (1996). *The sciences of the artificial (3rd ed.)* MIT Press.
Sommerville, I. (2011). *Software engineering* (9th ed.). Pearson.
Song, P., Zheng, C., Zhang, C., & Yu, X. (2018). Data analytics and firm performance: An empirical study in an online b2c platform. *Information & Management*, *55*(5), 633–642.
Spiller, S. A., Reinholtz, N., & Maglio, S. J. (2020). Judgments based on stocks and flows: Different presentations of the same data can lead to opposing inferences. *Management Science*, *66*(5), 2213–2231.
statcounter. (2023). *Operating system market share worldwide - june 2022 - june 2023*. Retrieved July 18, 2023, from https://gs.statcounter.com/os-market-share
Statista. (2019). *Big data projects success rate among corporations in the united states and worldwide as of 2019, by area*. Retrieved July 14, 2023, from https://www.statista.com/statistics/742935/worldwide-survey-corporate-big-data-initiatives-and-success-rate/
Statista. (2022). *Size of the big data analytics market worldwide from 2021 to 2029*. Retrieved July 15, 2023, from https://www.statista.com/statistics/1336002/big-data-analytics-market-size/
Statista. (2023). *Mobile android operating system market share by version worldwide from january 2018 to january 2023*. Retrieved July 28, 2023, from https://www.statista.com/statistics/921152/mobile-android-version-share-worldwide/
Tang, H., Liao, S. S., & Sun, S. X. (2013). A prediction framework based on contextual data to support mobile personalized marketing. *Decision Support Systems*, *56*, 234–246.
Trieu, V.-H. (2017). Getting value from business intelligence systems: A review and research agenda. *Decision Support Systems*, *93*, 111–124.
Trieu, V.-H., Burton-Jones, A., Green, P., & Cockcroft, S. (2022). Applying and extending the theory of effective use in a business intelligence context. *46*(1).
Ullenboom, C. (2017). *Java ist auch eine insel : Das umfassende handbuch. aktuell zu java 8* (12., aktual. u. überarb. Aufl.). http://openbook.galileocomputing.de/javainsel/
van Osch, W., & Steinfield, C. W. (2016). Team boundary spanning: Strategic implications for the implementation and use of enterprise social media. *Journal of Information Technology*, *31*(2), 207–225. https://doi.org/10.1057/jit.2016.12
Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, *26*(2).
Wixom, B. H., & Watson, H. J. (2001). An empirical investigation of the factors affecting data warehousing success. *MIS Quarterly*, *25*.
Wook, M., Hasbullah, N. A., Zainudin, N. M., Jabar, Z. Z. A., Ramli, S., Razali, N. A. M., & Yusop, N. M. M. (2021). Exploring big data traits and data quality dimensions for big data analytics application using partial least squares structural equation modelling. *Journal of Big Data*, *8*(1). https://doi.org/10.1186/s40537-021-00439-5
Wright, R. T., Jensen, M. L., Bennett Thatcher, J., Dinger, M., & Marett, K. (2014). Influence techniques in phishing attacks: An examination of vulnerability and resistance. *Information Systems Research*, *25*(2), 385–400.

Zaccagnino, C. (2020). *Programming flutter : Native, cross-platform apps the easy way* (1st edition).
ZAVADSKAS, E. K., VAINIŪNAS, P., TURSKIS, Z., & TAMOŠAITIENĖ, J. (2012). Multiple criteria decision support system for assessment of projects managers in construction. *International Journal of Information Technology & Decision Making*, *11*(2), 501–520.
Zhang, Q., & Li, J. (2021). Can employee's boundary-spanning behavior exactly promote innovation performance? the roles of creative ideas generation and team task interdependence. *International Journal of Manpower*, *42*(6), 1047–1063. https://doi.org/10.1108/IJM-06-2019-0302
Zhao, Z. J., & Anand, J. (2013). Beyond boundary spanners: The 'collective bridge' as an efficient interunit structure for transferring collective knowledge. *Strategic Management Journal*, *34*(13), 1513–1530. https://doi.org/10.1002/smj.2080

Appendices

Appendix A Identification of the Problem

Online database	Subject Focus
ABI/INFORM Collection	Business and management
Business Source Premier	Accounting, business, economics, management
EconBiz	Business and economics
ProQuest One Business	Business
AIS Electronic Library	Informatics
MIS Quarterly Website	Business informatics
Web of Science	Multiple databases that provide access to different academic topics
Google Scholar	Web search engine for scholarly literature across an array of disciplines

Table 9: Databases Used in the Literature Review

List of Keywords (First Search)

- Data Analytics
- Data AND Boundary
- Organization AND Data Analytics
- Big Data
- boundary
- boundary theory
- boundary spanning
- boundary objects
- boundary spanner

List of Keywords (Second Search)

- Data Analytics
- Data AND Boundary
- Organization AND Data Analytics
- Big Data
- boundary
- boundary theory

- boundary spanning
- boundary objects
- boundary spanner
- Decision
- Decision Making
- Action

Senior Scholars' Basket of Journals

- European Journal of Information Systems
- Information Systems Journal
- Information Systems Research
- Journal of AIS
- Journal of Information Technology
- Journal of MIS
- Journal of Strategic Information Systems
- MIS Quarterly

UT Dallas Top 100 Business School Research Rankings

- The Accounting Review
- Journal of Accounting and Economics
- Journal of Accounting Research
- Journal of Finance
- Journal of Financial Economics
- The Review of Financial Studies
- Information Systems Research
- Journal on Computing
- MIS Quarterly
- Journal of Consumer Research
- Journal of Marketing
- Journal of Marketing Research
- Marketing Science
- Management Science
- Operations Research

- Journal of Operations Management
- Manufacturing and Service Operations Management
- Production and Operations Management
- Academy of Management Journal
- Academy of Management Review
- Administrative Science Quarterly
- Organization Science
- Journal of International Business Studies
- Strategic Management Journal

Appendix B Definition of Objectives for a Solution

Online database	Subject Focus
Business Source Premier	Accounting, business, economics, management
EconBiz	Business and economics
AIS Electronic Library	Informatics
JSTOR	Covers a wide range of subjects
ACM Digital Library	Computer science and related topics
IEEE Xplore / Electronic Library	Computer science

Table 10: Databases Used in the Literature Review for Requirements

Some of the databases listed in table 10 did not contain articles in the corresponding journals that could be found by using the search terms and are therefore not included in table 11.

	Academic Journals in Business Source Premier			JSTOR			EconBiz		
	Search Phrase: Data Analyt-ics (36,028)	Search Phrase: Business Intelli-gence (12,694)	Search Phrase: big data (39,062)	Search Phrase: Data Analyt-ics (41,269)	Search Phrase: Business Intelli-gence (145,917)	Search Phrase: big data (316,288)	Search Phrase: Data Analyt-ics (2,753)	Search Phrase: Business Intelli-gence (6,775)	Search Phrase: big data (4,583)
The Accounting Review	0	0	0	182	187	685	1	0	1
Journal of Accounting and Economics	0	0	0	0	0	0	0	0	0
Journal of Accounting Research	0	0	0	67	45	373	1	3	8
Journal of Finance	0	0	0	103	91	870	4	25	15
Journal of Financial Economics	0	0	0	0	0	0	1	3	1
The Review of Financial Studies	0	0	0	107	70	441	0	0	2
Information Systems Research	61	41	0	79	243	187	2	51	28
Journal on Computing	43	18	0	0	0	0	0	0	0
MIS Quarterly	70	49	52	139	327	312	6	5	5
Journal of Consumer Research	0	0	0	34	128	366	0	0	0
Journal of Marketing	52	33	50	295	988	1508	7	4	14
Journal of Marketing Research	123	17	97	151	210	347	1	1	0
Marketing Science	50	31	54	119	99	256	1	1	4
Management Science	194	60	108	361	411	752	11	55	36
Operations Research	89	96	186	350	230	520	11	12	17
Journal of Operations Management	45	0	0	0	0	0	0	6	4
Manufacturing and Service Operations Management	0	0	0	0	0	0	0	0	1
Production and Operations Management	0	0	0	0	0	0	5	1	3
Academy of Management Journal	0	23	0	50	279	583	0	0	1
Academy of Management Review	0	26	0	36	244	241	0	0	0
Administrative Science Quarterly	0	0	0	25	157	374	0	0	1
Organization Science	0	21	0	48	280	435	0	1	0
Journal of International Business Studies	0	26	0	42	237	401	0	0	0
Strategic Management Journal	0	67	0	73	252	451	0	0	0
Decision Support Systems	242	151	166	0	0	0	0	0	0
European Journal of Information Systems	0	0	0	0	0	0	0	0	0
Information & Management	75	29	59	94	373	221	18	99	51
Information and Organization	0	0	0	0	0	0	0	0	0
Information Systems Journal	0	0	0	94	373	221	5	15	13
Journal of the AIS	0	0	0	0	0	0	0	0	0
Journal of Information Technology	56	24	50	0	0	0	3	8	1
Journal of Strategic Information Systems	0	0	0	0	0	0	0	0	0
Journal of MIS	0	0	0	0	0	0	0	0	0
Total	**874**	**571**	**720**	**2449**	**5224**	**9544**	**77**	**290**	**206**

Table 11: Number of Articles Identified for Requirements and Their Respective Database

Appendix C Design and Development of the Artifact

```
btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String nextStep = GetNextExperimentStepUseCase.getInstance().
            getNextExperimentStep();
        try {
            Class<?> c = Class.forName(nextStep);
            Intent intent = new Intent(ChooseTestSubjectActivity.this, c);

            startActivity(intent);
        } catch (ClassNotFoundException e) {
            System.out.println("Error␣" + e.getMessage());
        }
    }
});
```

Listing 1: Activity Call on Button Press

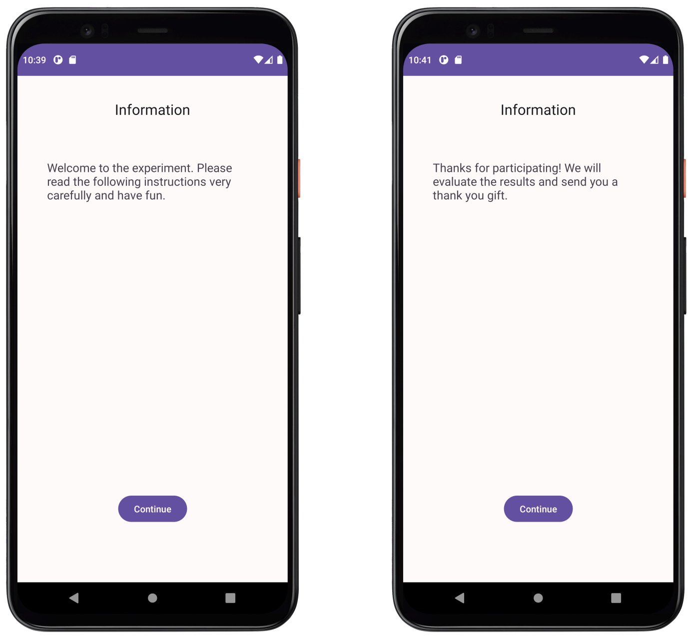
Use Cases

- **GetNextActivity**: (*Load next experiment step*) Get the next step of the experiment from the experiment data.
- **GetParticipantDataUseCase**: (*Load Participant Data*) Get all participant IDs or names from the participant data.
- **CreateNewParticipantUseCase**: (*Create New Participant*) Create a new participant ID or name in the participant data.
- **GetGroupAllocationPUseCase**: (*Read Group Allocation Participant*) Get the manual group allocation for a specific participant from the participant data.
- **GetGroupAllocationEUseCase**: (*Read Group Allocation Experiment*) Get the general group allocation from the experiment data.
- **AllocateGroupsUseCase**: (*Allocate Groups*) Automatically allocate groups and save the respective group allocation for every participant ID or name in the participant data.
- **SetGroupAllocationPUseCase**: (*Save group allocation*) Save the new group allocation in the participant file for the respective participant.
- **GetNextDataEntryUseCase**: (*Load next entry of data for participant*) Get the next data entry for a participant ID or name from the participant data.
- **InsertDataPUseCase**: (*Insert Missing Data Entry*) Save a data entry for a corresponding participant ID or name in the participant data.
- **LogMetaDataUseCase**: (*Log metadata*) Save metadata from the current process step in the experiment data.
- **GetInfoScreenDataUseCase**: (*Load info screen data*) Get information that should be displayed from the experiment data.
- **SetCurrentParticipant**: (*Set current participant*) Set the participant currently performing the experiment.
- **GetCurrentParticipant**: (*Get current participant*) Get the participant currently participating in the experiment.

Appendix D Evaluation of the Solution

T1: A welcome and goodbye message is displayed

The implementation of this test case can be done without further development by utilizing the standard functionalities of the developed prototype. For this purpose, the information ccreen activity is specified as the first and last step in the experiment data. Furthermore, a corresponding welcome and farewell text is stored in the experiment data. Figure 11 shows that the functional requirements F1.1 and F1.2 to show information at the beginning of the experiment and to provide the participants with debriefing information can be completely fulfilled by the artifact.



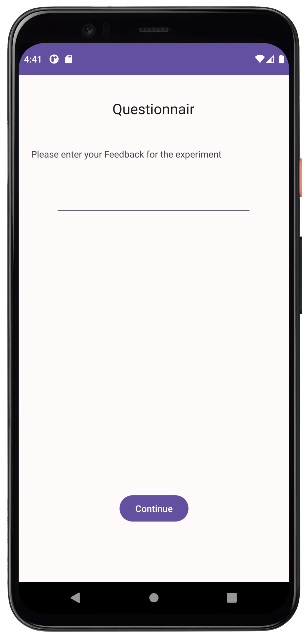
(a) Welcome Message (b) Goodbye Message

Figure 11: T1: User Interface

T2: Participants are prompted to input their age at the beginning and prompted to input how they liked the experiment at the end.

To perform this test, the experiment data must be adjusted first. For this, the questionnaire step must be placed at the beginning and the end of the experiment step order within the experiment data. In addition, the participant data must contain the respective data as an empty field to be queried as an attribute, since the questionnaire process only queries for missing data. These data entries are "age" and "experiment feedback" as defined by the test case. The two screens on which the participants are asked for their age at the beginning of the experiment and for feedback on the experiment at its end are shown in Figure 12. The test T2 and thus the requirements F2.1, F2.3, F4.3, and F4.2 can thus be fulfilled.

(a) Welcome Message



(b) Goodbye Message

Figure 12: T2: User Interface

T3: The information about how long the experiment took is collected.

To fulfill this test case, which checks requirement F2.2, the time necessary to perform the experiment is measured. For this purpose, the lines of code from listing 2 can be used on the start activity. These store the start time of the experiment in the metadata in the experiment data. The experiment is then run, and the code lines in listing 4 are called when the experiment is completed. These code lines retrieve the start time of the experiment from the experiment data and calculate the total elapsed time during the experiment using the current system time. This is then also stored in the metadata of the experiment data. Thus, the test case T3 can be completely fulfilled, and consequently, the requirement F2.2 can be satisfied.

```
String currentTime = new SimpleDateFormat("HH:mm:ss", Locale.getDefault()).format(new
    Date());
LogMetaDataUseCase.getInstance().setMetaData(currentTime);
```

Listing 2: T3: Collecting the Time Needed to Conduct an Experiment (a)

```
long difference = date1.getTime() - date2.getTime();
LogMetaDataUseCase.getInstance().setMetaData(difference);
```

Listing 3: T3: Collecting the Time Needed to Conduct an Experiment (b)

T4: The gender and weight of the participant are pre-loaded into the experiment from different files. The gender of the participant is deleted.

For test case T4, which verifies requirements F3.1, F3.2, and N6.1, the files to be read are accessed through corresponding Java interfaces in the ParticipantData class. Java supports the import of various file formats (Ullenboom, 2017). Additionally, the imported data can also be formatted or adapted. In test case T4, the gender and weight of the participants are imported, and the gender information is subsequently deleted.

```java
File csvfile = new File(Environment.getExternalStorageDirectory() + ”/participantData
    .csv”);
CSVReader reader = new CSVReader(new FileReader(csvfile.getAbsolutePath()));
String[] nextLine;
while ((nextLine = reader.readNext()) != null) {
    // nextLine[] is an array of values from the line
    ParticipantEntity participant = new ParticipantEntity(Integer.parseInt(nextLine
        [0]));
    participant.setName(nextLine[0]);
    participant.setEducation(nextLine[0]);
    participant.setGender(null);

    data.add(participant);
}
```

Listing 4: T4: Loading of Data into the Artifact

T5: Adding a Chess Game as Custom Logic

In test case T5, a chessboard game is to be implemented following the study by Krakowski et al., 2023a. This test case is intended to verify F4.1, F4.2, and F4.3, which deal with the implementation of custom logic. The chessboard serves as proof of the functionality to implement custom logic in the application. Since the checkerboard is an experimental step that requires interaction with the user, this test case is implemented as part of a new activity. This activity must be added to the sequence of activities in the experimental data to ensure that it is called. A screenshot of this activity is shown in Figure 13. The detailed coding of this custom activity will not be discussed further since it only serves to verify the requirements. Through this custom activity and the fact that Java is a Turing Complete programming language, test case T5 is fulfilled.

Figure 13: T5: Chess Board as Custom Experiment Logic

T6: Two groups are created, one of the groups is specifically chosen, and the other one is randomly selected

In general, groups are defined by adding a group ID (e.g., group A or group 1) in the user data. If the corresponding group assignment field is not filled, the assignment is performed automatically in the AllocateGroupsUseCase use case. The code snippet in listing 5 shows the standard code that assigns participants who have no group assignment to a random group. The corresponding group assignment is then stored in the previously empty attribute. Test case T6, thus, verifies requirements F5.1, F5.3, and F5.4. Furthermore, customized or extended group assignments can be implemented in the AllocateGroupsUseCase use case.

```
currentParticipant = getCurrentParticipantUseCase.getCurrentParticipant();
currentParticipantGroup = participantRepository.getParticipant(currentParticipant).
    getGroupAllocation();

if(currentParticipantGroup == null){

    if(experimentRepository.getExperiment().getGroupAllocation() == ”random”){
        Random random = new Random();
        int randomNumber = random.nextInt(2); // Generates either 0 or 1

        if(randomNumber == 0 ){
            setGroupAllocationPUseCase.setGroupAllocation(”Group␣A”,
                currentParticipant);
        } else {
            setGroupAllocationPUseCase.setGroupAllocation(”Group␣B”,
                currentParticipant);
        }
    };
}
```

Listing 5: T6: Allocation of Groups

T7: A chess turn is played by both parties, not using the same device

Test case T7 deals with the interaction between multiple participants. For this purpose, the chess game in test case T5 is to be played by several participants on different devices. The implementation of this test case lies within the corresponding activity itself, in this case, the ChessGameActivity. To achieve this, a client class is needed for sending the current game state and a server class for receiving and offering the current game state. These classes are nested as private classes within the activity itself. Listing 7 shows the code for the server class, listing 6 the code for the client class, and Listing 8 the initiation of both classes at the start of the chess game. The server and client are initiated using threads to ensure correct and fast communication between the two playing parties. In this way, after each move played, the current state of the board is sent via the client of one device to the server of the other device. Since this test case only serves as a proof-of-concept, Java sockets were used to implement the communication. These are characterized by their simplicity, but their functionality is limited, especially for modern applications. The use of third-party interfaces or externally hosted servers such as Firebase is, therefore, conceivable, depending on the requirements of the experiment (Google, 2023b). Communication between two experiment participants on different devices could be demonstrated by playing a chess move in each case. Test case T7, and thus requirements F4.2, F5.2, N1.1, and N1.2, could thus be fulfilled and confirmed.

```java
    private class ClientThread extends Thread {
        @Override
        public void run() {
            try {
                Socket socket = new Socket(server_ip, server_port);
                PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
                out.println(message);
                out.close();
                socket.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
```

Listing 6: T7: Server Code

```java
private class ServerThread extends Thread {
        @Override
        public void run() {
            try {
                ServerSocket serverSocket = new ServerSocket(client_port);
                Socket clientSocket = serverSocket.accept();

                BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket
                    .getInputStream()));
                String message = in.readLine();
                in.close();
                clientSocket.close();
                serverSocket.close();

                handleMessage(message);

            } catch (IOException e) {
                e.printStackTrace();
            }
        }
```

```
}
```

Listing 7: T7: Client Code

```
new ClientThread().start();
new ServerThread().start();
```

Listing 8: T7: Starting Server and Client Threads

T8: The results of the experiment are retrieved and displayed in third-party software

To fulfill test case T8 and thus verify requirements F2.2, N2.1, N2.2 and N5.1 metadata are exported from the application and then imported into external software. As mentioned above, Android in combination with Java supports a variety of file formats for export. The export of the metadata is implemented in the LogMetaDataUseCase use case via the printOutMetaData method and is shown in listing 9. For simplicity and as a proof of concept in the course of this test case, metadata in the form of processing times for the experiment is exported as a CSV file. These are then imported into Excel during the course of this test case and visualized using an Excel representation. Figure 14 shows this Excel graphic. In principle, however, the CSV file could be processed in any other way, provided that the software used for further processing supports CSV files.

```
File file = new File(Environment.getExternalStorageDirectory() + "/participant" +
    GetCurrentParticipantUseCase.getInstance().getCurrentParticipant() + "TimeData.
    csv");
try {
    // create FileWriter object with file as parameter
    FileWriter outputfile = new FileWriter(file);

    // create CSVWriter object filewriter object as parameter
    CSVWriter writer = new CSVWriter(outputfile);

    // adding header to csv
    String[] header = { "id", "time␣in␣milliseconds"};
    writer.writeNext(header);

    //Getting participant information
    ArrayList<ParticipantEntity> participantEntities = GetParticipantDataUseCase.
        getInstance().getParticipantData();

    //Writing current participant time into file
    //String[] data = {String.valueOf(GetCurrentParticipantUseCase.getInstance().
        getCurrentParticipant()), String.valueOf(timeDifference)};

    Iterator iter = participantEntities.iterator();
    while (iter.hasNext()) {
        String[] data = {String.valueOf(((ParticipantEntity)iter.next()).getId()),
            String.valueOf(((ParticipantEntity)iter.next()).getExperimentTime())};
        writer.writeNext(data);
    }
    //closing writer connection
    writer.close();
}
catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
    System.out.println("Error");
}
```

Listing 9: T8: Result Collection and Export

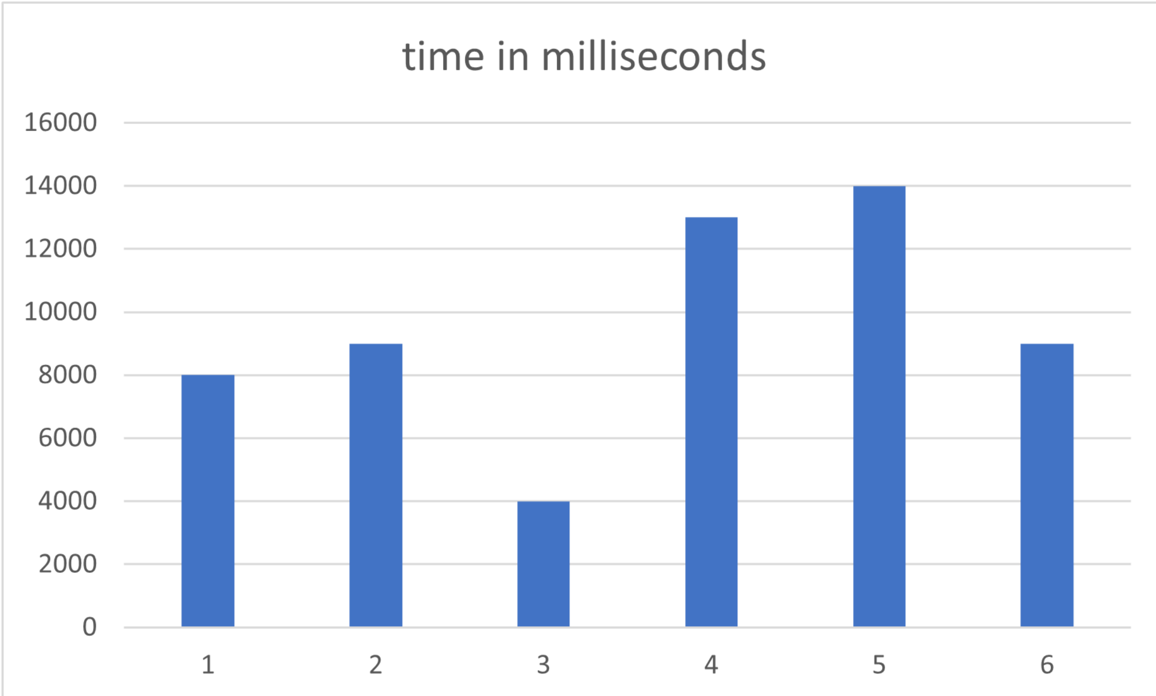


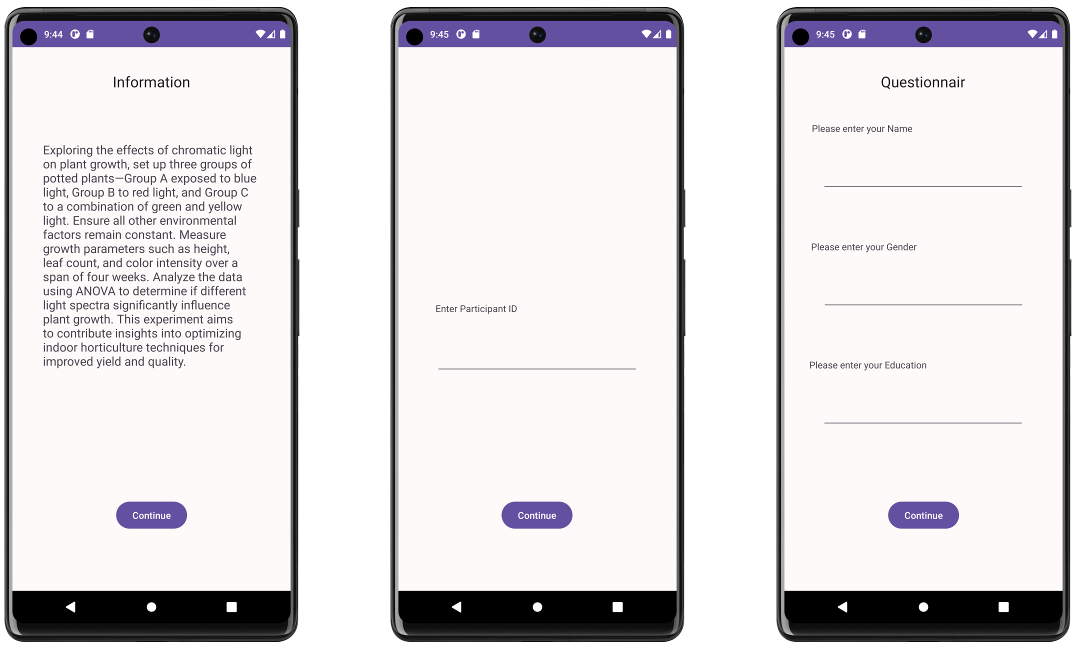
Figure 14: T8: Data Loaded into and Displayed with Excel

T9: The experiment is redone a second time, and another experimental setup is implemented

The re-execution of an experiment verifying requirement N4.1 can be implemented by re-reading the same experiment data. For this purpose, a generic experiment procedure was set up as an example, and the experiment data were copied and used to perform a second experiment with the help of the artifact. At the same time, care was taken to ensure that the data of the participants in the two sample experiments were different. The second example experiment was congruent with the first experiment in terms of structure and execution. Only due to the different participant data, other attributes were queried in the questionnaire activity.

T10: The experiment is conducted on different devices

To verify requirement F4.2 and confirm test case T10, the developed artifact was exported to several other Android devices. Originally tested during development, the artifact was on a Pixel 4 XL. Therefore, to verify this test case, the artifact was installed and tested on a Pixel 6 Pro. The individual activities on this device are shown in Figure 15. Although the two devices have different hardware, software, and screen sizes, the artifact can be used on both devices and does not show any errors or bugs. This means that test case T10 can be fulfilled.



(a) Info Screen (b) Questionair (c) Choose Test Subject

Figure 15: T10: Artifact Run on a Pixel 6 Pro

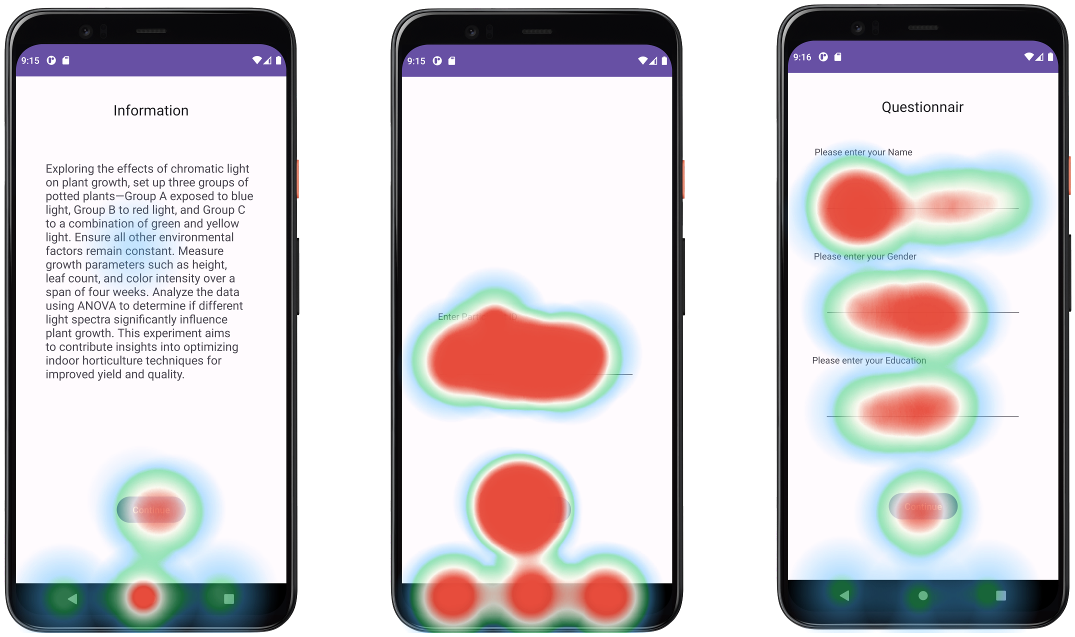
T11: During the experiment, the current state of the chessboard is exported to the conductor of the experiment

Test case T11 can be tested similarly to test case T7. For this purpose, a message about the current status of the experiment is sent to a server via a client class in the same way as in test case T7. In contrast to test case T7, no messages are sent back and forth, but only the current status of the experiment is exported to a monitor. Therefore, only one client class is needed in the application itself. The server to which the current status is sent can take different forms; only the Internet Protocol (IP) address of the server must be known, and it must be able to process the client message. Analogous to test case T7, the implementation of this communication is also possible via other ways and means should more complex functions be required that go beyond the capabilities of Java sockets. Test case T11 and thus requirement N5.1 can thus be verified.

User Interface Test

Participant Name	Time on Study in Seconds
Participant 1	27
Participant 2	22
Participant 3	33
Participant 4	8
Participant 5	20
Participant 6	12
Participant 7	17
Participant 8	10
Participant 9	15
Participant 10	19
Participant 11	34
Participant 12	9
Participant 13	9
Participant 14	11
Participant 15	22
Participant 16	32
Participant 17	29
Participant 18	22
Participant 19	19
Participant 20	9
Participant 21	13
Participant 22	31
Participant 23	23
Participant 24	26
Participant 25	12

Table 12: Participants for User Interface Test



(a) Info Screen (b) Participant Selection (c) Questionnair Step

Figure 16: Artifact User Interface - Heatmap

Affidavit

I hereby declare that I have developed and written the enclosed master thesis entirely on my own and have not used outside sources without declaration in the text. Any concepts or quotations applicable to these sources are clearly attributed to them. This master thesis has not been submitted in the same or a substantially similar version, not even in part, to any other authority for grading and has not been published elsewhere. This is to certify that the printed version is equivalent to the submitted electronic one. I am aware of the fact that a misstatement may have serious legal consequences.

I also agree that my thesis can be sent and stored anonymously for plagiarism purposes. I know that my thesis may not be corrected if the declaration is not issued.

Mannheim, September 5, 2023

Max Darmstadt