



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

Master Thesis
of

MAX DARMSTADT

September 2, 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.....	iv
List of Tables.....	v
List of Algorithms	vi
List of Abbreviations	vii
Abstract.....	viii
1 Introduction.....	1
1.1 Background and Motivation	1
1.2 Objective and Expected Contribution.....	3
2 Theoretical foundations.....	4
2.1 Design Science Research Methodology	4
2.2 Data Analytics	5
2.3 Information Value Chain.....	5
3 Identification of the Problem	6
4 Definition of Objectives for a solution.....	12
4.1 Requirements Elicitation	13
4.1.1 Studies in Data Analytics - A Literature Review	14
4.1.2 Further relevant reference resources	17
4.2 Requirements Specification.....	18
4.3 Requirements Validation.....	21
5 Design and Development of the Artefacts.....	23
5.1 Process Conceptualization	23
5.2 Technology Selection.....	27
5.2.1 Android and Android Studio.....	28

5.2.2	Java	28
5.3	System Architecture Development.....	30
5.3.1	Data Layer.....	31
5.3.2	Domain Layer	33
5.3.3	User Interface Layer	34
5.4	Consolidated System Architectural Summary.....	36
6	Demonstration of the Artifact.....	38
7	Evaluation of the Solution	40
7.1	Prototype validation.....	40
7.2	App Performance and Usability.....	41
8	Conclusion	44
	Bibliography	46
	Appendix A: Identification of the problem	55
	Appendix B: Definition of Objectives for a solution.....	59
	Appendix C: Design and Development of the Artefact	61
	Appendix D: Evaluation of the Solution	63

List of Figures

1	Information Value Chain.....	6
2	Experiment - Swim lane	24
3	Data input step - Swim lane.....	25
4	Choose test subject step - Swim lane.....	25
5	Group allocation step - Swim lane.....	26
6	Questionair step - Swim lane	26
7	Info screen step - Swim lane.....	27
8	User Interface Prototype of Artefact.....	35
9	Complete Architecture of the Artefact	37
10	User Interface Implementation	39
11	T1: User Interface.....	63
12	T2: User Interface.....	64
13	T5: Chess Board as Custom Experiment Logic	67
14	T8: Data Loaded into and Displayed with Excel	72
15	T10: Artefact Run on a Pixel 6 Pro	74
16	Artefact User Interface - Heatmap.....	75

List of Tables

1	Results Assigned to the Information Value Chain.....	8
2	Research Approach Used in the Literature.....	9
3	Research Methods for Literature Search on Requirements	16
4	Requirements Uncatagorized.....	17
5	Functional Requirements Structured	19
6	Non-Functional Requirements Structured.....	20
7	Fulfillment of Functional Requirements.....	42
8	Fulfillment of Non-Functional Requirements	42
9	Databases Used in the Literature Review	55
10	Databases Used in the Literature Review for Requirements	59
11	Number of Articles Identified for Requirements and Their Respective Database	60
12	Participants for User Interface Test	76

Listings

1	Activity Call on Button Press	61
2	T3: Collecting the Time Needed to Conduct an Experiment (a)	65
3	T3: Collecting the Time Needed to Conduct an Experiment (b)	65
4	T4: Loading of Data into the Artefact	66
5	T6: Allocation of Groups	68
6	T7: Server Code	69
7	T7: Client Code	70
8	T7: Starting Server and Client Threads	70
9	T8: Result Collection and Export	71

List of Abbreviations

AI	Artificial Intelligence
CSV	comma-separated values
DSR	Design Science Research
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IS	Information Systems
IT	Informationstechnologie
UI	User Interface
JVM	Java Virtual Machine

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 usage of new technologies that utilize the collecting 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 rising at the same time (Seagate, 2018) and the pressure to use these data volumes effectively in order to gain a business advantage rises. This new trend, often coined “big data” after the fact that never before seen amounts of data are generated and are 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 come to the conclusion that the mere possession of big data as an asset has no positive effects 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 therefore 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 holding back the utilization of data analytics. Section 3 of this thesis uses the information-

value-chain to look at 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 show that there is a lack of research on non-technical aspects of the information-value-chain. These literature gaps mainly consist of areas like decision-making and behavioral research. These results have been confirmed by studies in the past focussing on related fields (Trieu, 2017), which might indicate a persistant issues. This only becomes more apparent as new technologies, which overlap with data analytics like machine learning and artificial intelligence get more widespread. These “black-box”technologies are already alternatives 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 big impact on the effectiveness of data analytics (Holsapple et al., 2014). This lack of research on non technical regards could become a huge issue in the future, specifically for the decision-making process as firms’ top priorities focus more and more 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 hypothesis. The experiment being 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 hypothesis 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 lack of experimental research generally found in data analytics (refering to the results of the literature search in section 3). Improving the experimental research process in the field of data analytics could thereby significantly improve the future state of knowledge on said topic, whilst also allowing organizations to succeed in the fast paste economical 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 their methodological procedure (2) the development of an artefact which improves the research process in the field of behavioral research in data analytics (3) the validation of the artefact through the exemplary realization of a study in said field utilizing said artefact. In order to accomplish the creation of this artefact 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 Dev of artefacts*, *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 in order 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 artefact, utilizing the *Requirement Engineering* approach (Sommerville, 2011, Alain Abran, James W. Moore, 2004). These requirements are then used to design and develop an artefact. Subsequently, the resulting artefact is then demonstrated and assessed by implementing a real experimental study in data analytics as an example to evaluate the artefact and its benefits for the experimental research process. This exemplary implementation is then also used to validate the aforementioned requirements. The last step of the DSR framework, which focusses 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 artefact is created which 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 artefact, but can also be used in off-topic areas beyond the use cases of this thesis.

2 Theoretical foundations

This section represents the theoretical fundamentals of this elaboration by defining the term “data analytic”, the concept of the “information value chain” and the experimental research process in general.

2.1 Design Science Research Methodology

In order to accomplish the goal of this thesis, to create an artifact for the improvement of the research process in the field of data analytics, the DSR approach the 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 to 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) artefact which deals with a certain problem (Hevner et al., 2004), making the 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 “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 in order 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 Dev of artefacts, Demonstration of the Artifact, Evaluation of the solution and Communication* (Peffers et al., 2006).

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 it is discussed in. In this elaboration, data analytics refers to the processing of large amounts of data, also referred to as “big data”, through mathematical procedures or machine learning 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. The goal of data analytics, as is 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 to 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 time, 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 like 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 for 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 contains the further phases “decisions” and “actions”, which deal with the influence of the processed data. These phases reflect the impact of data analytics, since 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.

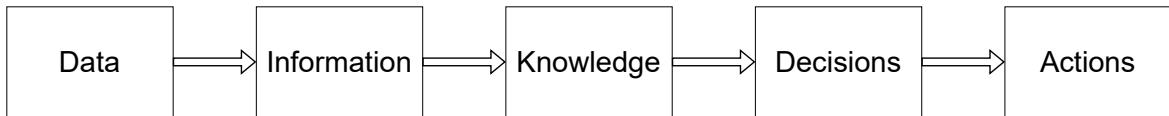


Figure 1: Information Value Chain

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 search is first conducted to find potential gaps in the field of data analytics and methods used in this field. Subsequently, applications and tools are examined, which had been intended to support the research process.

In order to identify constraints on the research on data analytics a literature search is conducted. The main objective of it is to analyze the existing literature to find research gaps, particularities and interrelationships between literature. This is supposed to give insights into the current state of research and to find out which part of the research process on data analytics still has room for improvements. Consequently, relevant literature was identified and reviewed. Afterwards, the identified literature was categorized and analyzed. Initially, it was assumed that the topic of data analytics lies both in the field of information systems and business (Abbasi et al., 2016, Levina and Vaast, 2005). For this reason, the literature search was mainly conducted in literature databases that focused on these topics (Table 10). The

literature search was conducted using a keyword search. A full list of keywords that were used is included in the appendix .

In order to ensure the quality of the identified literature initially, only publications from certain journals were considered. These journals consist of 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 later includes journals in the area of business administration. A full list of included journals is listed in the appendix. 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 meet 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 relevant to the topic. This process yielded 35 research publications. The results were then assigned to different phases of the aforementioned information value chain, their content best represents. This was done to find literature gaps in the general process of data processing. Additionally, the identified literature was categorized by their research methodology in order to find patterns and similarities in the literature. By mapping the literature found, 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.

Table 1 shows an overabundance of literature that got assigned to the "knowledge" phase of the information value chain. The significantly fewer entries for the other phases

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

indicate less research on these sub-parts 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 application of data. Their main research object does not directly consist of any 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”) are mostly considered together in the further elaboration, since the literature which was assigned to these phases lies thematically very closely together.

Only four publications were assigned to the “decisions” phase and none to the “actions” phase. These results in particular call into question if 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 overspanning 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 focussed

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. The full list of keywords is included in the appendix. 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 number 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 the fact that the topic of boundaries in data analytics is not researched extensively.

In order to further analyze the literature and to potentially draw further conclusions, the found literature was also categorized regarding the research method that was 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 show 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 analysis 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 (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 Columbian Experimental Laboratory for Social Sciences. 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 Readymade Economic Experiments: The software component z-Tree was developed as a toolbox for conducting economic experiments. The technical set-up 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 meta data which 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 a 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 removed 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 by Python, requiring programming expertise. Experiments implemented in oTree are implemented 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 setup 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, 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.

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 which enables experimental research in the field of behavioral research in data analytics.

In order to define an objective for a possible solution in 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 sys-

tem 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). See also: design requirement; functional requirement; implementation requirement; interface requirement; performance requirement; physical requirement.” 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 to provide a certain quality these requirements are established using the *requirement engineering* approach for analysis and evaluation of requirements is utilized (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, the *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. 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 in order to find out whether these requirements are actually representative of the desired artefact (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 artefact. These requirements are discovered by analyzing applications and frameworks that

were created to support experimental research in related fields and through a literature review of studies that have utilized experiments in the field of behavioral research in data analytics.

4.1.1 Studies in Data Analytics - A Literature Review

In order to establish further requirements a second literature search is conducted, which focusses 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 both in the fields of information systems and business administration the same databases were used as for the literature review in section 3. Table 10 gives an overview of the databases 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 altering 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. The full list of these journals can be found in appendix 8 and 8. 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 assumption based on a sample size, which does not

represent the full population (Heckman, 2010). A simple example of Selection Bias would be to calculate 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 atrificial intelligence change the compatitive advantage by being substitutes to 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 similair conclusion, stayting 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 backwards and forwards 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. As already mentioned, these terms were used to search the databases from appendix 8 and 8 for matching articles. The search was furhter limited to articles written in the english language. The abstracts of all articles were then analyzed. The number of found articles for each database are displayed in Table 11. This table indicates

how many articles per corresponding term are contained in the respective database, which were published in one of the relevant journals. In this way, a total of 19,955 articles were considered by inspecting their headline and, if applicable their abstracts. The other databases listed in the appendix (among others, AIS Electronic Library, ACM Digital Library, IEEE Xplore / Electronic Library, ProQuest) did not contain articles in the corresponding journals that could be found by using the search terms and are therefore not included in the table.

This process initially yielded 46 research publications, which were supplemented by 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 review 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 artefact which is subject of thesis.

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

The requirements derived from this are included in Table A. As already mentioned, not only requirements of performed experiments were included, but also requirements of set-ups 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” 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 to already performed experiments are discovered, in order to counteracted selection bias.

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
Debriefing Info	1	0			

Table 4: Requirements Uncatagorized

4.1.2 Further relevant reference resources

In addition to already developed applications and the relevant articles themselves, this section includes other sources that can be derived into 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, H. Reinders, H. Ditton and C. Gräsel 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 goal of the experiment to the participants after it has been conducted (Gniewosz, 2011). As already mentioned in the theoretical part about the theory behind the execution of experiments, 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 the measures “Parallelization of confounding variables of the test subjects”, “Randomization of confounding variables of the subjects” and “Control group” represent measures which can 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, the deployment on non windows platforms, the open-source approach and the possibility to add own program code for the experiments based on the analysis of z-Tree and oTree. Further requirements based on the analysis of Lionness Lab are 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 explanation 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 pricetag in euros and in dollars. Non-functional characteristics describe on the other hand the behavior of a system (Seacord et al., 2003) and go thereby beyond the 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	Debriefing 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 artefact 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 Distant 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 Visualize Final Data	The data can be retrieved in a suitable way for further visualization
Simplicity	
N3.1 Simplicity	The artefact is simple to use
Reusable and Interoperable	
N4.1 Reusable	Experiments with the artefact are easy to re-do
N4.2 Interoperability	The artefact is interoperable
N4.3 Openness of Platform	The artefact is open to changes and enhancements
Monitoring	
N5.1 Monitoring of Study	The study conducted with the artefact 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 Artefact 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 in subsection ?? are also conform 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 A and B 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 F2.1. Requirement F2.1 ensures that additional logic and program code can be implemented within the artefact, 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. However, this point will be discussed in more detail in the implementation of the artifact. At this point in time, this point is considered to be fulfilled. 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. An important point, since the artefact must be tested for all requirements following the DSR approach. In order to be able

to formally confirm the criterion, a test-case is therefore designed which checks the artifact against the 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 they 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 conductor of the experiment	N5.1

The non-functional requirements simplicity, openness of platform and 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 requirement more precisely and for example to replace “advanced user interface” with “a user interface that was developed in 2023”. This would satisfy the verifiability criterion, but no longer the validity criterion, since this requirement is actually fairly subjective and the test might not capture the initial intention of the require-

ment. For this reason, the three requirements Simplicity, openness of platform and advanced user interface are valid requirements, but no test cases are included for them. Nevertheless, they are taken into account in the development of the artefact and finally evaluated as best as possible.

By successfully setting up these tests, the “Verifiability” criterion can thus also be confirmed. Thus, the established artifacts meet the criteria Validity, Consistency, Completeness, Realism, and Verifiability and are thus valid requirements for the artifact.

5 Design and Development of the Artefacts

The artifact, which is intended to improve experimental research in behavioral research in 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. For this purpose, the application is first fundamentally conceptualized. This involves analyzing possible technologies and the corresponding processes. The architecture is then implemented in practice on the basis of this basic conceptualization.

5.1 Process Conceptualization

In order 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 of this is to illustrate the individual processes in a technology-independent manner. The reason for this is that the individual processes are initially presented in generalized form, irrespective of technological restrictions or limitations. The following Swim Lane diagrams represent the individual steps of the experiment. 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 large part of the final functionalities can be divided into two different categories. An example of this are the requirements (F2, F3 and N5.1), whereby the application should offer both the possibility to read in and save user data and data corresponding to the experiment setup itself. These requirements can be summarized in functions and data concerning the experiment and functions and data concerning the

participants. The two lanes “Participant Repository” and “Experiment Repository” are representative for those two categories and data sources. The “User” lane shows the activities performed by the test subjects, the “Experimental Repository” lane shows all activities that can be associated with the experimental setup, and the “Participant Repository” lane shows all data and activities that are related to the test subjects.

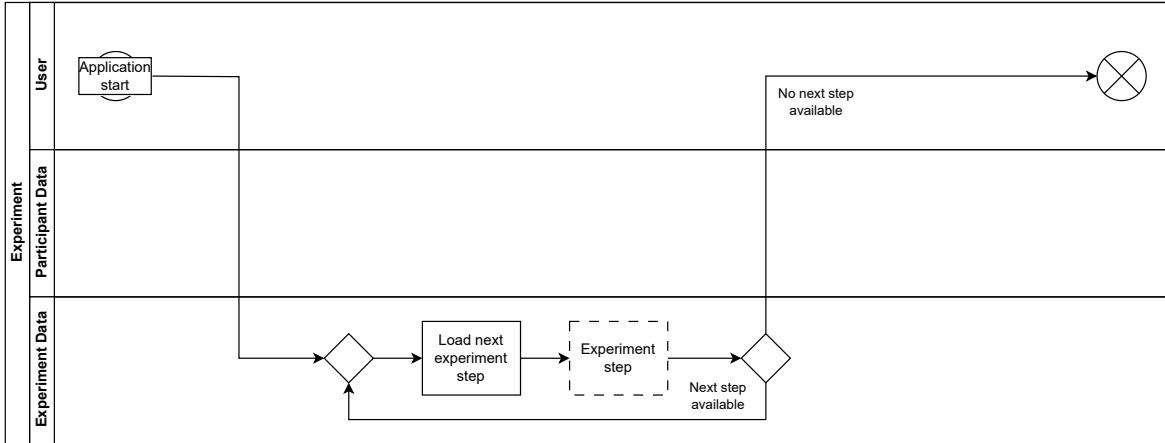


Figure 2: Experiment - Swim lane

Figure 2 shows the basic structure of an experiment with the conceptualized application. 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 the most flexible and adaptable construction kit for an experiment. The individual steps of the experiment, which are executed one after the other, can be customized by the person performing the experiment as well as mapped by standard steps. The standard steps represent steps which have to be used in a multitude of experiments and thus partially represent requirements. In the following, some of these experiment steps which illustrate standard steps that are included in the application are illustrated with the help of swim lane diagrams. Figure 3, 5, 6 and 7 represent these standard steps. The dashed box “Experimental Step” of figure 2 can be replaced by any number of these. For this reason, the graphics mentioned also begin and end in the “Experimental Repository” lane.

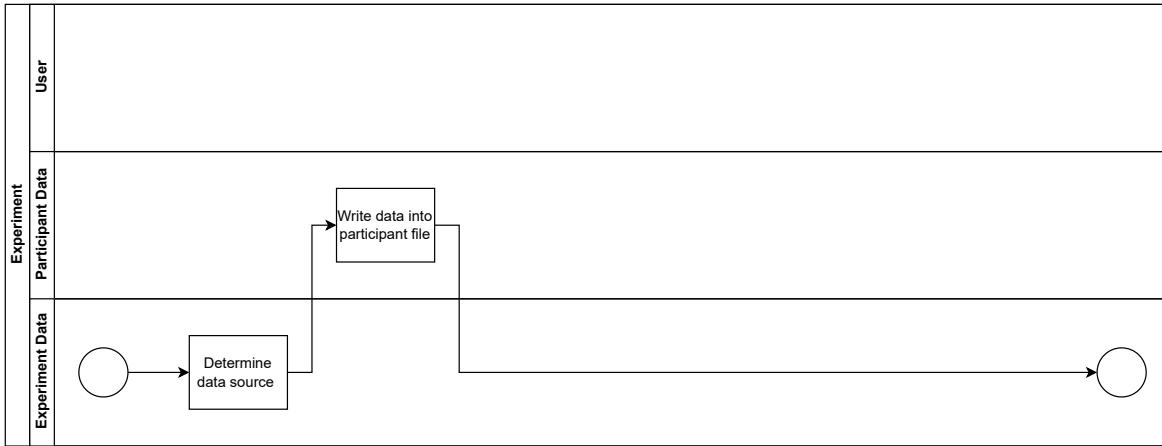


Figure 3: Data input step - Swim lane

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.

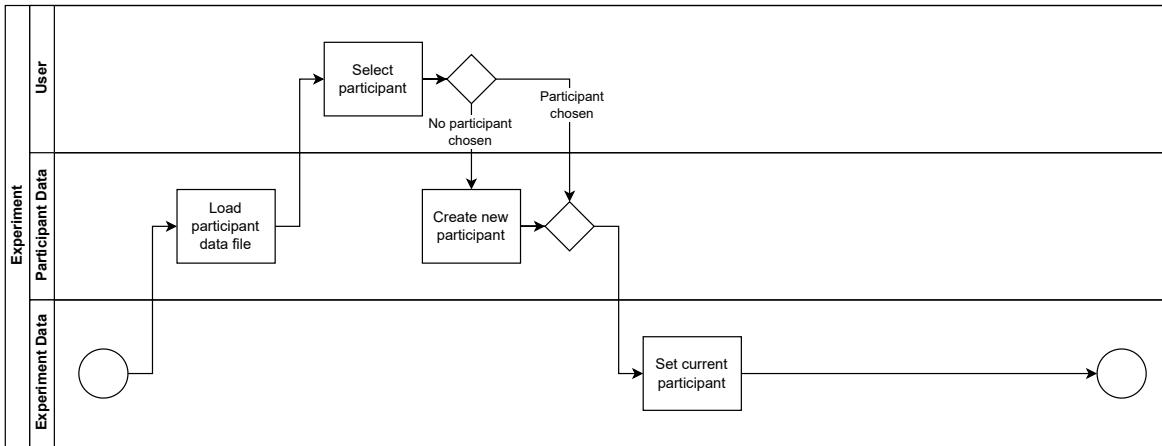


Figure 4: Choose test subject step - Swim lane

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

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

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. This process is

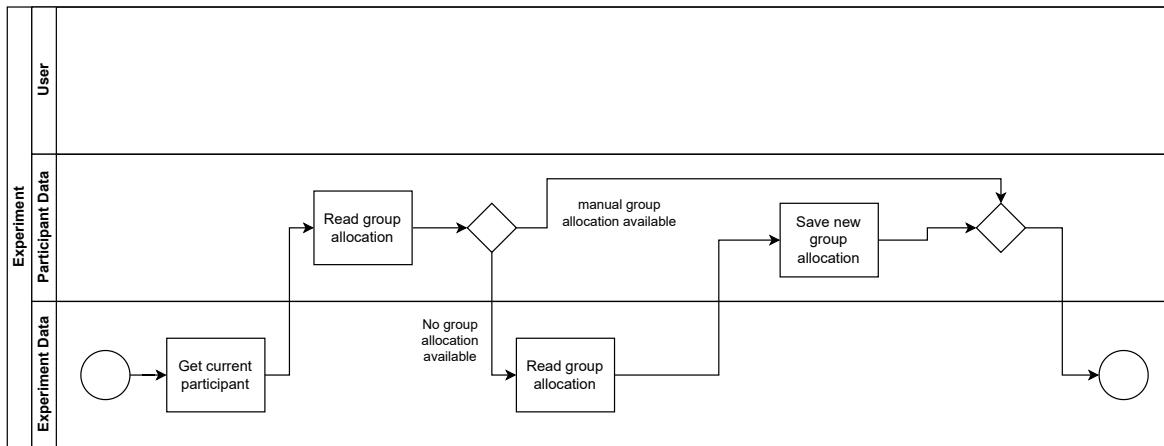


Figure 5: Group allocation step - Swim lane

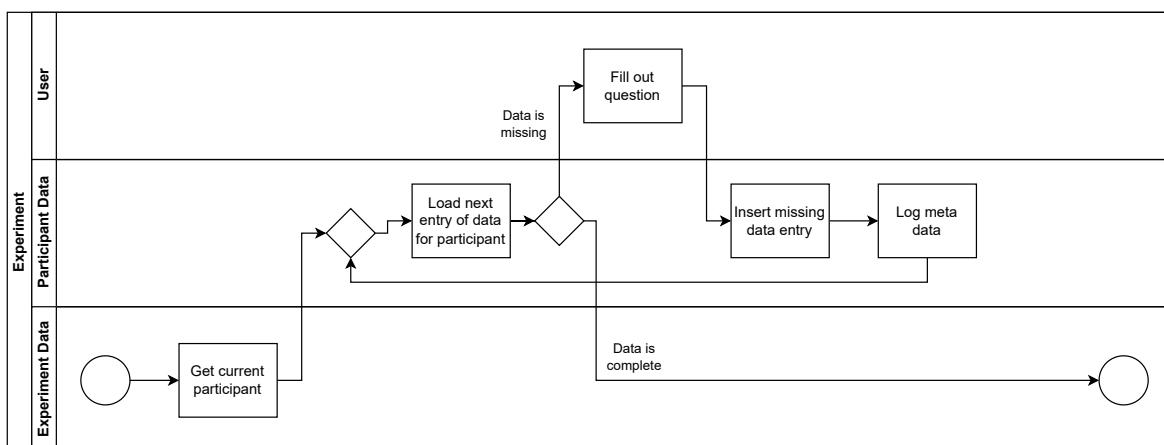


Figure 6: Questionair step - Swim lane

conceptualized similarly to the general experiment setup, so that any number of questions or no questions are displayed for completion, depending on the number of missing data points about a test subject. An example would be a test subject for which the age is missing. An input field is automatically displayed for the participant to fill in the missing data about himself. If all data for a test subject is complete, nothing is displayed.

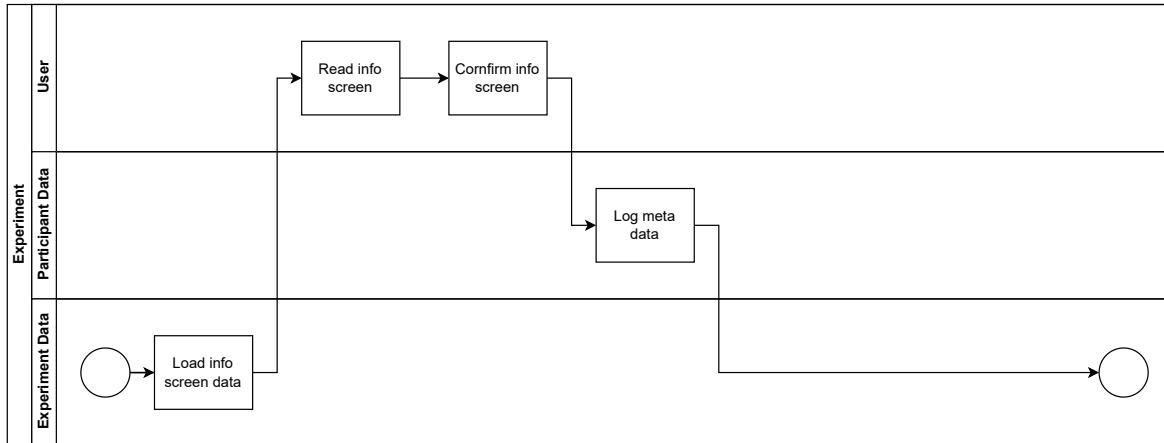


Figure 7: Info screen step - Swim lane

Figure 7 displays information to the test subjects and can be used both as a means of notification during an experiment and 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 N3.1 (Evaluation of Data), N3.2 (Vizualize Final Data), N7.1 (Multi-Source) are represented by the availability of various interfaces. Furthermore, the two non-functional requirements N1.2 (Time-Flexibility) and N6.1 (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 (Distand Communication), N4.1 (Simplicity), N5.1 (Reusable), N5.2 (Interoperability), N5.3 (Openness of Platform), N8.1 (Advanced User Interface) and the afformentioned availability of interfaces as requirements for selecting a suitable technology. In the following chapters, various 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 over all, being used in almost every second device (statcounter, 2023, E. Richter, 2019). The standard development environment to develop Android 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 openness of platform and interoperability requirement (E. Richter, 2019). Applications for the Android operating system are called apps. These apps are programs designed for touch inputs, which are specifically designed 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 distant communication requirements can be fulfilled through the usage of Android. The two programming languages that can be used to develop these Android apps are Java and Kotlin. These apps then can be tested either directly on an Android device or on a variety of virtual devices integrated in 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 user interface as an input function, in addition to network functionalities and a variety of interfaces. Android therefore has a wide range of design guidelines, interface functionalities and is updated at very regular intervals (Statista, 2023, E. Richter, 2019). Due to these regular updates, the widespread use of Android and the focus on User Interface (UI) intensive use cases, it can be assumed that an app developed in Android supports the latest user interface technologies and therefore fulfills the advanced user interface requirement.

5.2.2 Java

Java is a programming language originally developed by Microsystems. Since 2009 Java is part of the product portfolio of Oracle Corporation. Java is an object-oriented program-

ming language, which makes it an 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 byte code, 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 byte code, which is then executed via an interpreter in a virtual environment which is 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 beside Java itself also the execution of some other programming languages as for example Kotlin (kotlinlang.org, 2023). Due to this fact, Java is especially suitable for the implementation of the artefact based on the requirements N5.1 (Reusable) and N5.2 (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 (Distand 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 very easy to understand and beginner-friendly (Ullenboom, 2017), fulfilling the requirement N4.1 (Simplicity). In addition to technical aspects, Java is also Open-Source, extremely widespread, popular and a variety of literature for it is available (Ullenboom, 2017), which generally indicates an openness of platform (Requirement N5.3 Openness of Platform) and convenient to develope code (N4.1 (Simplicity)). Disadvantages of Java, which are also mentioned for the sake of completeness, mainly relate to very specific platform-dependent use cases. Since Java was developed as a general-purpose programming language and platform-independent, it is very difficult to access hardware or drivers directly. However, these use cases are irrelevant in the course of this work (Ullenboom, 2017).

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 remaining 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 these 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 the best option. Furthermore, there is the possibility that at a later point in time another technology combination may be able to better reflect the requirements. At this point in time, however, the requirements can best be met by the combination of Android and Java. In addition, for the reasons mentioned, it can be assumed that this combination of technologies will remain the best way to implement the requirements in the long term. However, the fact that these technologies will no longer be up to date at a later point in time exists, but cannot be ruled out for any other technology too. On the contrary, Android and Java are a sensible and sustainable choice in the long term for the above-mentioned reasons according to the current state of research. 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 app developed on this version runs on about 95.4% of all Android devices. At the same time, the app can easily be ported to a new version if newer features are needed for the implementation of an experiment.

The recommended architecture for an Android app is shown in Figure ?? and consists of three layers, the UI layer, the domain layer, and the data layer. The UI layer displays application data and the app itself to the user. The UI layer is further divided into “UI elements” and “State Holders”. “UI elements” correspond to the displayed screen elements and the “State Holders” to temporary data of the current state of the UI. The domain layer is op-

tional. It is used for abstraction and structuring of the data layer and is recommended above all if the application is to represent very complex business cases or the application must be designed to be very reusable (Google, 2023a). Due to the requirements N5.1 (Reusable), a domain layer is therefore used in the conceptualized application. 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 *GetTime* 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, to centralize changes to the data and to resolve 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 data base (Google, 2023a). The communication between the individual layers is solved in Java via so-called call-back methods. Call-back methods are special methods to which program code is passed, which is executed at a certain time or event and which then notifies the higher-level program. In this way, the call-back function does not have to be modified each time, but said program code can be passed. An example would be a data retrieval, the code for this is passed using a call-back method, which is then executes when the event that the queue is processed is triggered, which then notifies the parent program of its execution and result (Zaccagnino, 2020). 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 between none and any number. The individual data sources depend on the respective experiment. For this reason, a simple local 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 2, 3, 4, 5, 6 and 7 as these 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 artefact from potential data sources that get implemented within experiments two repositories called *ParticipantRepository* and *ExperimentRepository* are implemented. The *ParticipantRepository* could also contain multiple data sources for information about the participants. This architecture is illustrated in Figure 9. As already mentioned, the conceptual design of the artifact is limited to a single data source for the experiment itself and the participants for the sake of simplicity. This is reflected by the colored elements of Figure ??, the gray parts are only exemplary placeholders for an extension of the data sources.

The Android development scheme for 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 is performed in the domain layer. For this reason, the “Data Input Step” (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 is passed on to the repositories above it and the domain layer in the form of a 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 app and the avoidance of large classes by splitting the tasks (Google, 2023a). The domain layer classes are accessed in the same way by the UI as repositories of the data layer are accessed. A simple non related 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 with 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* lane from section 5.1. Entire processes (swim lanes) that do not contain user input can also be mapped by these domain layer use case classes. This encapsulates the business logic of the processual flow in these. 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 the appendix.

Figure 9 shows the modeling of the domain layer in relation to the data layer. The green use cases refer to use cases that exclusively access participant data, blue to use cases that exclusively access experiment data, and cyan to entire processes that access both participant and experiment data.

For the implementation of the individual UseCases, a Java class is created for each

case, which contains the corresponding logic and the UseCase calls as corresponding interfaces through methods.

5.3.3 User Interface Layer

The concept of graphical user interfaces 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 app’s interaction with the user” (Google, 2023a). The required UI elements of the app are generated in these activity. An activity corresponds to a single screen of an application. Apps can contain several different UI screens and thus several different activities. Activities can also call other activities and navigate to them as desired. An app is therefore a sequence of different activities that represent different screens. Usually, an activity serves as the entry point to the app. 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 app, Activities are only very 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 makes 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*, *questionair* and *info screen* step, depicted in figure 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

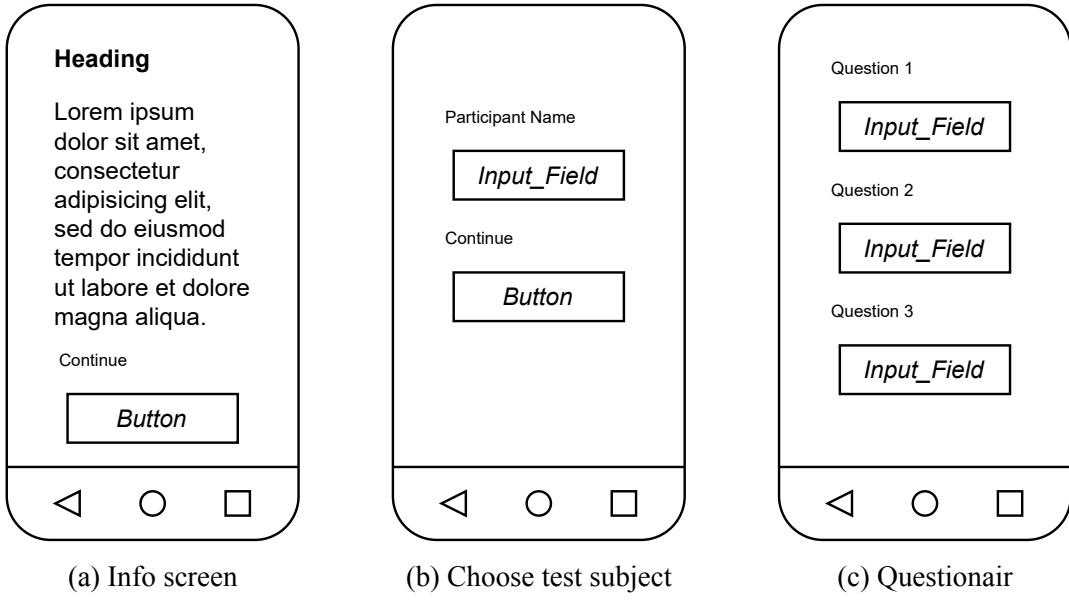


Figure 8: User Interface Prototype of Artefact

experiment steps is guaranteed.

The individual activities only hold a temporary data set of information that is displayed or required in the UI. As already explained, all other business logic is outsourced to the domain layer. This can 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 process which the designed screens in figure 8 are for are all non time sensitive, which is why they all contain a “continue” button in order 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 apps, but in the data of the experiment. As conceptualized in Process 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 shown in listing 1. The navigation to experiment steps that do not contain any user interaction and are therefore not

bound in any activity, but in the domain layer, takes place directly before the call of the next activity. An example of this would be that an activity calls 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.

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 the diagrams represents the interaction of the user with the artefact. 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 usecase, with the exception of the process 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” are represented by a single usecase 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 inlcuding the UI, domain and data layer are represented in figure 9.

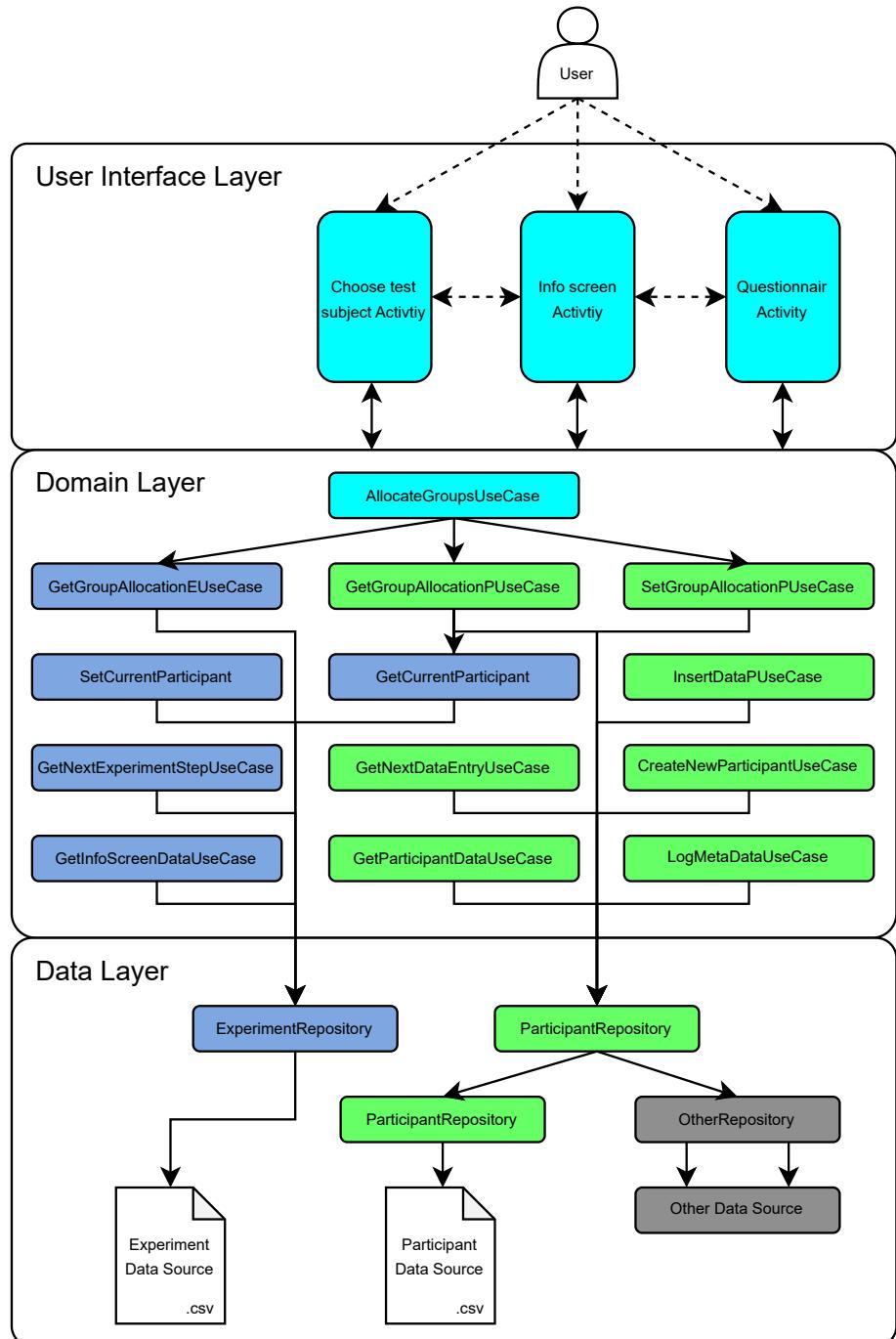
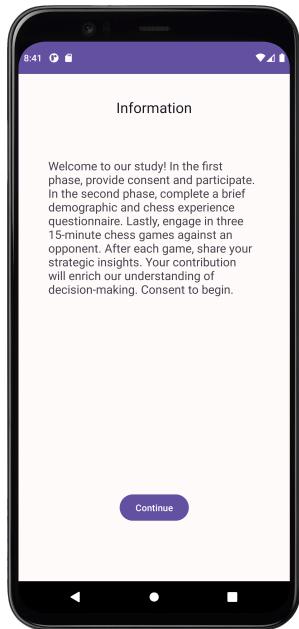


Figure 9: Complete Architecture of the Artefact

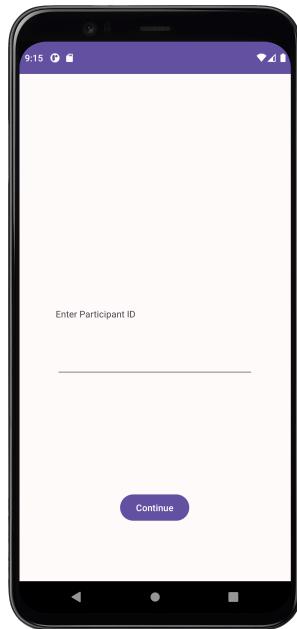
6 Demonstration of the Artifact

In order to validate the conceptualized artifact and to show 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 researches changes in the origin of competitive Advantages through Artificial intelligence. For this, they use a case study on chess tournaments to explore what impact Artificial Intelligence (AI) has on competitive advantage (Krakowski et al., 2023a). Although the methodological experimental design of this article is perfectly fine, the test could have been done by experiment. According to B. Gniewosz (2011) an experiment would even have been the preferred methodology in this field (Gniewosz, 2011). 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 that can be played between two people. Figure 10a, 10b and 10c show the conceptualized standard functionalities of the artefact. The individual fields that are visible in the UI are dynamically filled with data from the corresponding data sources. These data sources are filled with dummy data. The experiment data specifies the order of the screens as follows (1) Information Screen (2) Participant Selection Screen (3) Questionnaire Screen. The participant data contains 5 participants that have no prior information filled out about them. The application is emulated on a Pixel 4 XL API 30.

For changing or adding the individual steps of the application, only the sequence data of the experiment must be changed. The same applies to the Information Screen or the Questionnaire step. New experiments can build on the existing UseCases in the domain layer and extend them in the course of object-oriented programming. If additional experiment steps contain UI interactions, these are implemented via 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 repo at: https://github.comschneemaxmaster_thesis



(a) Choose subject step



(b) Questionair step



(c) Info screen step



(d) Chess game

Figure 10: User Interface Implementation

7 Evaluation of the Solution

To validate the developed prototype, the previously established test cases are verified for the individual requirements. The testing of the test cases provides information about the fulfillment of the requirements. In addition, the fulfillment of requirements that could not be validated by test cases is considered. Subsequently, the usability of the UI of the prototype is assessed.

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. Table 7 and 8 shows the corresponding requirements and the associated test cases that prove and verify them. A detailed description of the fulfillment of the test cases is included in the appendix. One exception 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 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 individual technologies is used. Besides the fact that Android and Java are generally considered to be simple beginner friendly technologies by the developer community, the best practice architecture that was implemented in the course of this application is a big indicator for the simplicity of the application. The individual functions and code modules have been divided into reusable UseCases and all user interface activities have been commulated into android activities. In addition, the implementation of new custom capabilities for individual experiments has been made extremely easy. For example, the definition of the experiment steps is realized through the experiment data 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 his experiment without having to worry about the basic framework. Despite the subjectivity of the "simplicity" requirement,

the requirement N3.1 simplicity is considered to be fulfilled for the artefact due to the usage of technologies which are considered to be simple and beginner friendly, the reusable and streamlined best practice architecture and functionalities of the android application and the encapsulation of standard functionalities. Requirement N4.3 Openness of platform describes the openness of the artifact for changes and enhancements. In general, it could be shown in the already verified requirements that the artifact is extensible. By using the best practice architecture in combination with the general concept of object orientation on which Java is based, it is also possible to argue that the application can be easily enhanced. In addition, to verify requirement N7.1, the artifact must enable the use of modern user interface components. As already explained, this requirement is also a subjective but nevertheless important requirement for the artifact. In the user interfaces of the final artifact, the MaterialUI developed by Google is used. Google claims that the Material UI is distinguished from other user interface 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 user interface in 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 (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 why all other requirements that cannot be substantiated by test cases are also considered to be fulfilled and verified. Hence, all requirements for the artifact are fulfilled and verified.

7.2 App Performance and Usability

After all the requirements for the artifact have been met 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 processes

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 Distant Communication	T7	✓
N1.2 Time-Flexibility	T7	✓
Data Postprocessing		
N2.1 Evaluation of Data	T8	✓
N2.2 Visualize 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

that require user inputs. For this purpose, the respective activities in the said processes were used and implemented on the individual screens by the corresponding UI components of an Android application with Material UI design. The UI thus adheres to the proven and tested UI concepts of Google's Material UI design philosophy. Nevertheless, this part of the work is intended to test the rough layout and usability of the artifact's user interface. 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 usable interface. For this purpose, a paper prototype is built using the screenshots from the artifact with the help of SAP Build. This prototype will be sent to 50 business informatics students of a large German Dax software company. They have the task to click through the prototype. Both the clicks and the time needed to navigate through the individual screens and the prototype are measured. The goal of this small test is to verify that the UI of the prototype is basically usable. The test would show potential for improvement if some participants did not manage to click to the end of the paper prototype, took an unusually long time to do so, or clicked several times at a point that was not intended to be interact with. In total, 32 of the invited participants were able to take part in the study. The participants took part in the test anonymously, for this reason the names of the participants in Appendix 8 are filled with placeholders. The participants spent an average of 23.9375 seconds to complete the paper prototype. The users' summarized clicks are shown in the heatmaps in Figure 16a, 16b, and 16c in the appendix. The results of the test do not indicate any serious negative design decisions. The average time it took the participants to click through the prototype is reasonable and the heatmap in Figure 16 does not show any unusual hotspots. In general, the user interface 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 Material UI result in a meaningful and usable user interface 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 user interface of the artifact is usable and, in combination with the claims made by Google about the MaterialUI, is considered sufficiently functional for the artifact.

8 Conclusion

The objective of this thesis was the development of an artifact which would improve Experimental Research in Behavioral research in Data Analytics. This was to be ensured by the following three Objectives: (1) the review of prior research on data analytics and their methodological procedure (2) the development of an artefact which improves the research process in the field of behavioral research in data analytics (3) the validation of the artefact through the exemplary realization of a study in said field utilizing said artefact. These three objectives were 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 app 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 machine learning, AI or data analytics already represent alternatives to human decision-making, and the artifact developed in this paper 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 artefact 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. 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 on the basis of 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 app 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.
- 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., & Coussemant, 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., Yogeve, 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>
- for Social Sciences, C. E. L. (2023). *Resources for online experiments*. Retrieved July 17, 2023, from <https://celss.iserp.columbia.edu/content/resources-online-experiments>
- 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.), *Microeconomics* (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 dictionary 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., Popović, 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 ready-made 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 infor-

- mation 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.
- 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., & Maretz, 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>

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

Academic Journals in Business Source Premier	JSTOR		EconBiz	
	Search Phrase: Data Analytics (36,028)	Search Phrase: big data (39,062)	Search Phrase: Data Analytics (41,269)	Search Phrase: big data (316,288)
The Accounting Review	0	0	0	182
Journal of Accounting and Economics	0	0	0	0
Journal of Accounting Research	0	0	0	67
Journal of Finance	0	0	0	103
Journal of Financial Economics	0	0	0	91
The Review of Financial Studies	0	0	0	107
Information Systems Research	61	41	0	79
Journal on Computing	43	18	0	0
MIS Quarterly	70	49	52	139
Journal of Consumer Research	0	0	34	128
Journal of Marketing	52	33	50	295
Journal of Marketing Research	123	17	97	151
Marketing Science	50	31	54	119
Management Science	194	60	108	361
Operations Research	89	96	186	350
Journal of Operations Management	45	0	0	0
Manufacturing and Service Operations Management	0	0	0	0
Production and Operations Management	0	0	0	0
Academy of Management Journal	0	23	0	50
Academy of Management Review	0	26	0	36
Administrative Science Quarterly	0	0	0	25
Organization Science	0	21	0	48
Journal of International Business Studies	0	26	0	42
Strategic Management Journal	0	67	0	73
Decision Support Systems	242	151	166	0
European Journal of Information Systems	0	0	0	94
Information & Management	75	29	59	94
Information and Organization	0	0	0	0
Information Systems Journal	0	0	0	94
Journal of the AIS	0	0	0	0
Journal of Information Technology	56	24	50	0
Journal of Strategic Information Systems	0	0	0	0
Journal of MIS	0	0	0	0
Total	874	571	720	2449
				9544
				77
				290
				206

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

Appendix C: Design and Development of the Artefact

```
1  btn.setOnClickListener(new View.OnClickListener() {
2      @Override
3      public void onClick(View view) {
4          String nextStep = GetNextExperimentStepUseCase.getInstance().
5              getNextExperimentStep();
6          try {
7              Class<?> c = Class.forName(nextStep);
8              Intent intent = new Intent(ChooseTestSubjectActivity.this, c);
9              startActivity(intent);
10         } catch (ClassNotFoundException e) {
11             System.out.println("Error " + e.getMessage());
12         }
13     }
14 );
```

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 participants 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*) Allocate groups automatically 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 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 meta data*) Save meta data 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 that currently performs the experiment.
- **GetCurrentParticipant**: (*Get current participant*) Get the participant that currently participates 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 Screen 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 artefact.

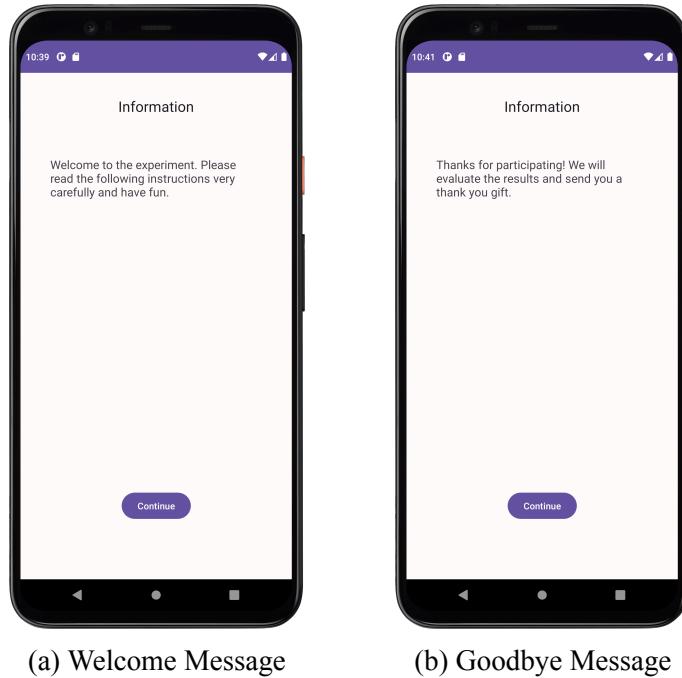


Figure 11: T1: User Interface

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 query of 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.

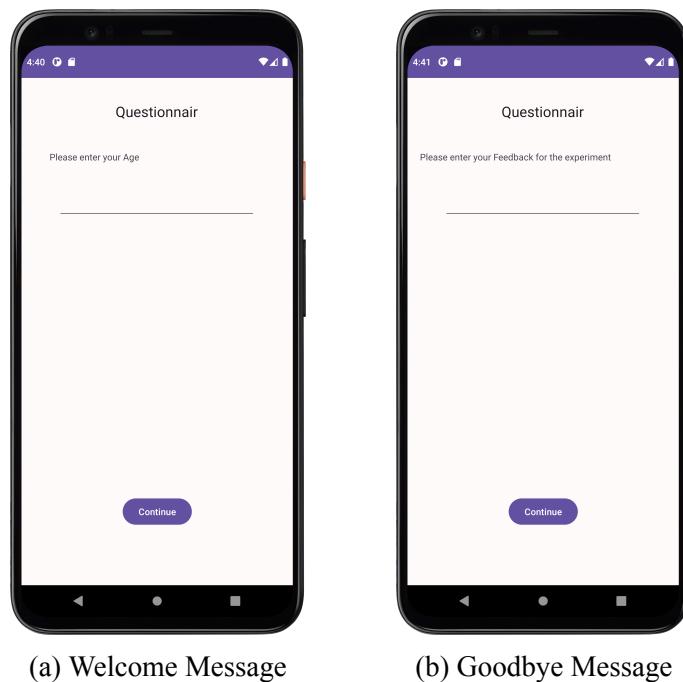


Figure 12: T2: User Interface

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, the lines of code from Listing 2 can be used on the start activity. These store the start time of the experiment in the meta data 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 meta data of the experiment data. Thus, the test case T3 can be completely fulfilled and consequently the requirement F2.2.

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

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

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

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

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

For test case T4, which verifies requirement F3.1, F3.2 and N6.1 the files to be read in are read via corresponding Java interfaces in the ParticipantData class. Java supports the import of a variety of different file formats (Ullenboom, 2017). At the same time, the read-in data can also be formatted or adapted. In test case T4, the gender and weight of the participants is read in and the information about the gender is then deleted.

```
1  File csvfile = new File(Environment.getExternalStorageDirectory() + "/participantData
   .csv");
2  CSVReader reader = new CSVReader(new FileReader(csvfile.getAbsolutePath()));
3  String[] nextLine;
4  while ((nextLine = reader.readNext()) != null) {
5      // nextLine[] is an array of values from the line
6      ParticipantEntity participant = new ParticipantEntity(Integer.parseInt(nextLine
         [0]));
7      participant.setName(nextLine[0]);
8      participant.setEducation(nextLine[0]);
9      participant.setGender(null);
10
11     data.add(participant);
12 }
```

Listing 4: T4: Loading of Data into the Artefact

T5: A chess game is added as custom logic

In test case T5, a chessboard game is to be implemented following the study of 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 a 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 in the course of a new activity. This activity has to be added to the order of the activities in the experiment 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 particularly chosen the other one randomly selected

Generally, 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 the participants who have no group assignment to a random group. The corresponding group assignment is then stored in the previously empty attribute. The test case T6 and thus the requirements F5.1, F5.3 and F5.4 are verified. Furthermore, a customized or extended group assignment would be implemented in the AllocateGroupsUseCase use case.

```
1  currentParticipant = getCurrentParticipantUseCase.getCurrentParticipant();
2  currentParticipantGroup = participantRepository.getParticipant(currentParticipant).
   getGroupAllocation();
3
4  if (currentParticipantGroup == null){
5
6      if (experimentRepository.getExperiment().getGroupAllocation() == "random"){
7          Random random = new Random();
8          int randomNumber = random.nextInt(2); // Generates either 0 or 1
9
10         if (randomNumber == 0){
11             setGroupAllocationPUseCase.setGroupAllocation("GroupA",
12                 currentParticipant);
13         } else {
14             setGroupAllocationPUseCase.setGroupAllocation("GroupB",
15                 currentParticipant);
16         }
17     };
18 }
```

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 in the corresponding activity itself. In this case in the ChessGameActivity. For 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 in 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.

```
1  private class ClientThread extends Thread {
2      @Override
3      public void run() {
4          try {
5              Socket socket = new Socket(server_ip, server_port);
6              PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
7              out.println(message);
8              out.close();
9              socket.close();
10         } catch (IOException e) {
11             e.printStackTrace();
12         }
13     }
14 }
```

Listing 6: T7: Server Code

```
1  private class ServerThread extends Thread {
2      @Override
3      public void run() {
4          try {
5              ServerSocket serverSocket = new ServerSocket(client_port);
6              Socket clientSocket = serverSocket.accept();
7
8              BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket
9                  .getInputStream()));
10             String message = in.readLine();
11             in.close();
12             clientSocket.close();
13             serverSocket.close();
14
15             handleMessage(message);
16         } catch (IOException e) {
17             e.printStackTrace();
18         }
19     }
20 }
```

Listing 7: T7: Client Code

```
1  new ClientThread().start();
2  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 meta data are exported from the application and then read into external software. As mentioned above, Android in combination with Java basically 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, meta data in the form of processing times for the experiment is exported as a comma-separated values (CSV) file. These are then read into Excel in 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.

```
1  File file = new File(Environment.getExternalStorageDirectory() + "/participant" +
2      GetCurrentParticipantUseCase.getInstance().getCurrentParticipant() + "TimeData.
3      csv");
4  try {
5      // create FileWriter object with file as parameter
6      FileWriter outputfile = new FileWriter(file);
7
8      // create CSVWriter object filewriter object as parameter
9      CSVWriter writer = new CSVWriter(outputfile);
10
11     // adding header to csv
12     String[] header = { "id", "time\u2014in\u2014milliseconds"};
13     writer.writeNext(header);
14
15     // Getting participant information
16     ArrayList<ParticipantEntity> participantEntities = GetParticipantDataUseCase.
17         getInstance().getParticipantData();
18
19     // Writing current participant time into file
20     //String[] data = {String.valueOf(GetCurrentParticipantUseCase.getInstance().
21         getCurrentParticipant()), String.valueOf(timeDifference)};
22
23     Iterator iter = participantEntities.iterator();
24     while (iter.hasNext()) {
25         String[] data = {String.valueOf(((ParticipantEntity)iter.next()).getId()),
26                         String.valueOf(((ParticipantEntity)iter.next()).getExperimentTime())};
27         writer.writeNext(data);
28     }
29 }
```

```

23         }
24         //closing writer connection
25         writer.close();
26     }
27     catch (IOException e) {
28         // TODO Auto-generated catch block
29         e.printStackTrace();
30         System.out.println("Error");
31     }

```

Listing 9: T8: Result Collection and Export

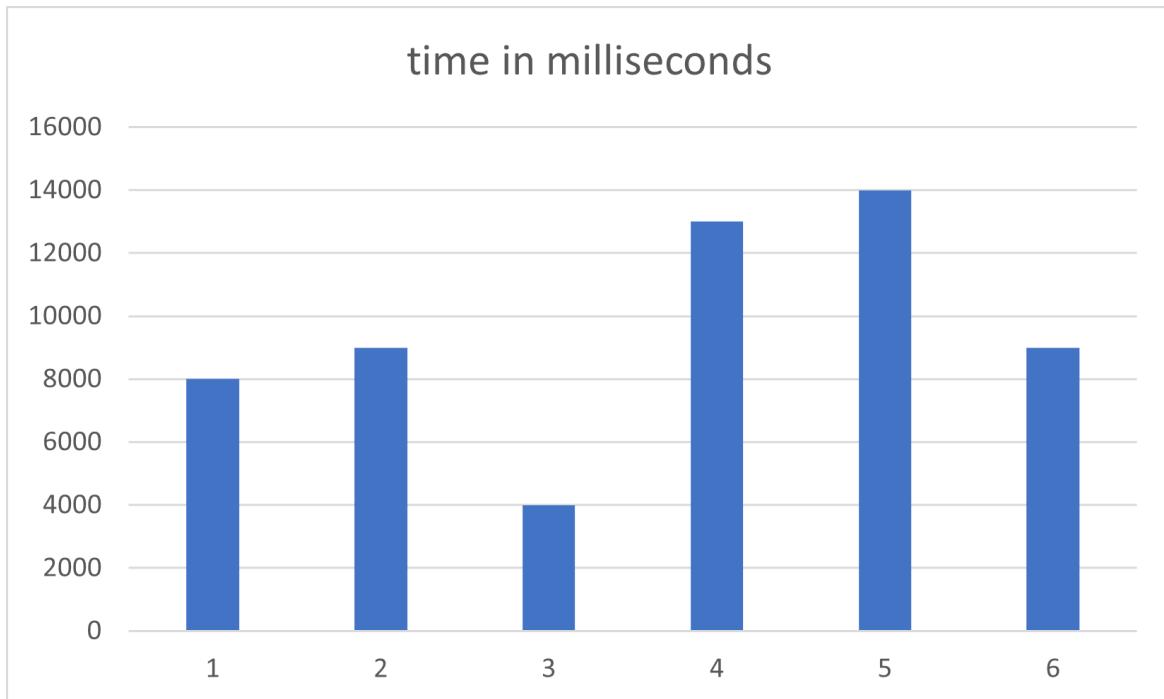


Figure 14: T8: Data Loaded into and Displayed with Excel

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

The re-execution of an experiment verifying the 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, the experiment data was 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 questionair 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.

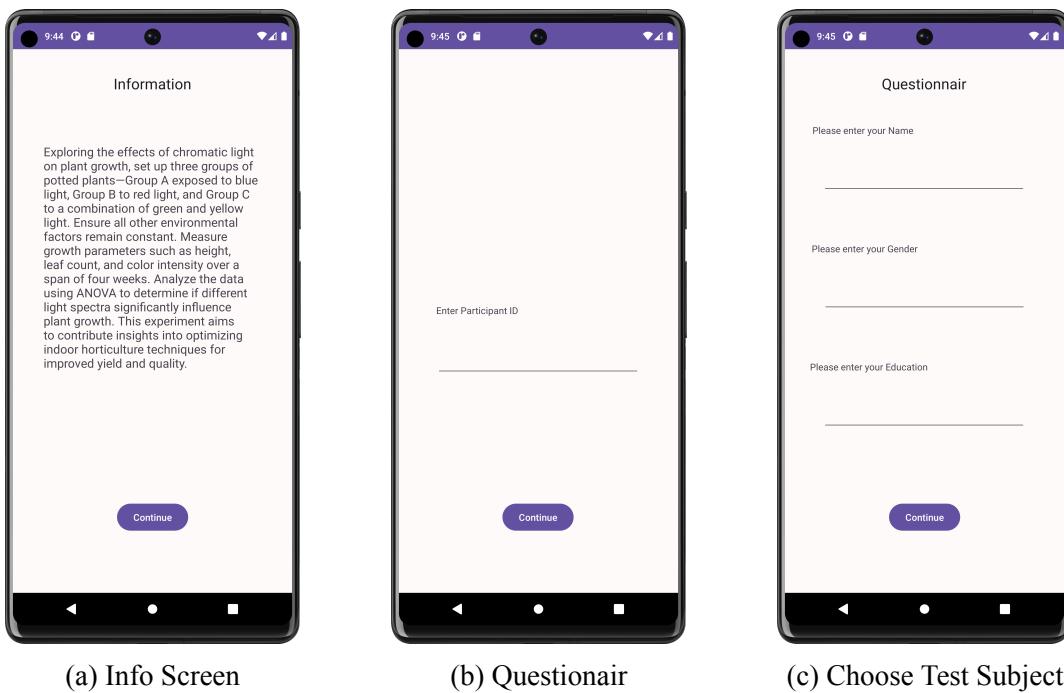


Figure 15: T10: Artefact Run on a Pixel 6 Pro

T11: During the experiment the current state of the chess board 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 therefore take different forms, only the 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 the Java sockets. Test case T11 and thus requirement N5.1 can thus be verified.

User Interface Test

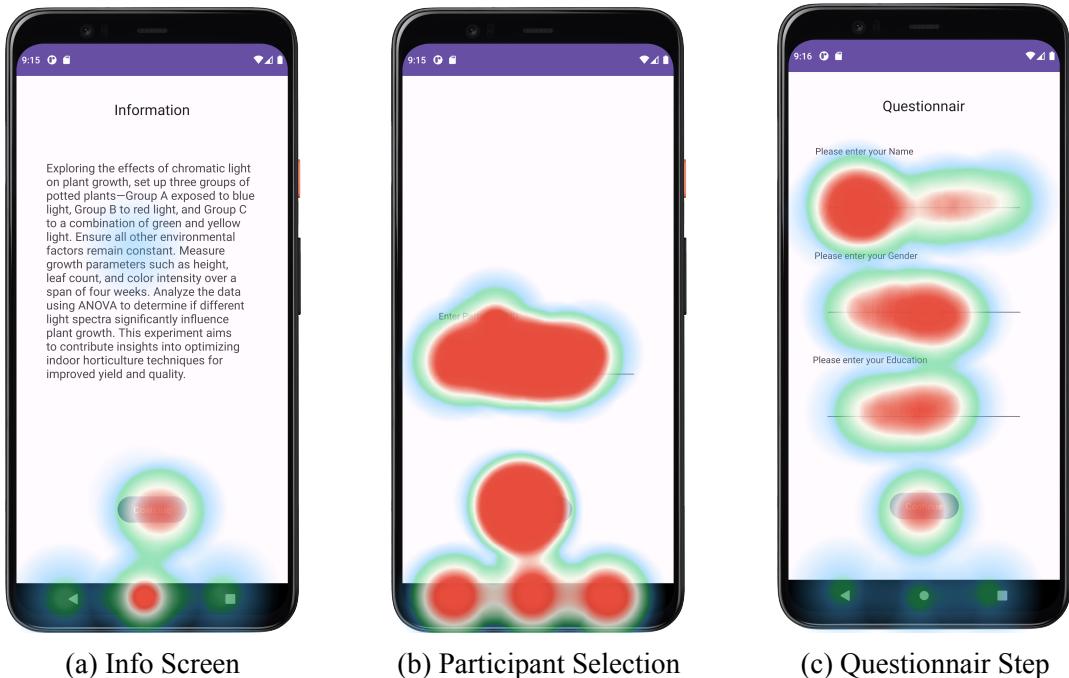


Figure 16: Artefact User Interface - Heatmap

Participant Name	Time on Study in Seconds
Participant 1	27
Participant 2	22
Participant 3	72
Participant 4	33
Participant 5	31
Participant 6	8
Participant 7	20
Participant 8	12
Participant 9	17
Participant 10	10
Participant 11	15
Participant 12	19
Participant 13	34
Participant 14	9
Participant 15	9
Participant 16	11
Participant 17	22
Participant 18	19
Participant 19	32
Participant 20	29
Participant 21	22
Participant 22	94
Participant 23	19
Participant 24	34
Participant 25	9
Participant 26	13
Participant 27	31
Participant 28	23
Participant 29	26
Participant 30	12
Participant 31	22

Table 12: Participants for User Interface Test

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 2, 2023

Max Darmstadt