Hochschule Bremen - City University of Applied Sciences

In cooperation with DLR German Aerospace Center

Implementation of electronic checklists for muscle biopsies, tilt table experiments and spiro-ergometry in order to increase medical safety and scientific quality during biomedical studies at the DLR

Malte Fuhrmann

A thesis submitted in partial fulfillment of the requirements for the Degree of

Master of Engineering

August 1, 2016

Supervisor

Prof. Dr.-Ing. Jörg J. Buchholz
Hochschule Bremen - City University of Applied Sciences

Second Examiner

Dr. med. Ulrich Limper
DLR German Aerospace Center

Malte Fuhrmann:
Implementation of electronic checklists for muscle biopsies, tilt table experiments and spiro-ergometry in order to increase medical safety and scientific quality during biomedical studies at the DLR

Implementierung von elektronischen Checklisten zur Erhöhung der medizinischen Sicherheit und wissenschaftlichen Qualität bei der Durchführung von Muskelbiopsien, Kipptischversuchen und Fahrradspiroergometrien im Rahmen von biomedizinischen Studien am DLR

Master Thesis
Hochschule Bremen

Preparation time: May 02, 2016 - August 1, 2016

Statement of Originality

This thesis has been performed independently with support by my supervisor. It contains no material that has been accepted for the award of a degree in this or any other university. To the best of the candidate's knowledge and belief, this thesis contains no material previously published or written by another person except where due reference is made in the text.

Bremen, August 1, 2016

Malte Fuhrmann

Abstract

Background Checklists are a powerful safety tool in aviation. Their use in cockpits has proven to be effective for many years to decrease the number of mistakes made by pilots in the environment of the "complex system" airplane. Safety checklists are more and more being introduced in medicine and clinical operations but not in human biomedical research so far. Thus the goal of this thesis is to implement medical safety checklists in the context of complex human biomedical studies and evaluate whether or not they can improve medical safety and scientific data quality.

Methods Between January and April 2016, the second campaign of a long-term bed rest study was hosted at the :envihab facility of the German Aerospace Center (DLR) in Cologne. Three complex experiments were chosen which were performed at least twice for each subject: 1. Muscle biopsy, 2. Tilt table tests, 3. Bicycle Spiro-ergometry. Before developing and implementing checklists for the above-mentioned experiments, data was collected to evaluate the performance regarding subject safety and scientific data quality. During the baseline data collection phase in January, 36 experiments on 12 subjects were analyzed. The actual bed rest phase, which lasted for 60 days, was used to generate a Visual Basic based website, which allows electronic checklists to be conducted on tablet computers. The application enables users to open and conduct predefined checklists and save comments and photos for documentation purposes. Furthermore, checklist templates can be edited and thus adapted to changing requirements of upcoming studies. During the recovery phase at the end of the campaign, each of the three experiments was repeated for every subject. The electronic checklists were introduced to and tested by DLR personnel conducting the experiments. The performance was analyzed again using the same indicators to evaluate the effect checklists have on medical safety and data quality.

Results The overall performance combining all three types of experiments was 67% before the checklists were introduced and rose to 97% afterward. Subject safety performance was 39% at baseline and 90% after the introduction of checklists. Considering the quality of scientific data, the performance was increased from 64% to 99% with the help of electronic checklists.

Conclusions Implementation of electronic checklists in human biomedical studies was associated with concomitant increase in medical safety and scientific data quality among subjects participating in complex experiments during a long-term bed rest study at the German Aerospace Center in Cologne.

Acknowledgments

First and foremost I would like to offer my sincerest gratitude to my supervisor Professor Jörg J. Buchholz from the Hochschule Bremen. He had already been the supervisor of my Bachelor thesis in 2014. During the preparation of this thesis, he has once more supported me with his patience and knowledge. His door was always open, when I needed assistance or advice with the code. I attribute the level of my Masters degree to his encouragement and effort. One simply could not wish for a kinder or better supervisor.
Secondly, I would like to express the deepest appreciation to my second examiner Dr. Ulrich Limper for the permission to write my thesis at the German Aerospace Center in Cologne. I consider it an honor to work with him on implementing checklists in scientific research. He consistently allowed this paper to be my own work, but steered me in the right direction when it was necessary. His enthusiasm for aerospace research and medical science, as well as his ability to encourage and inspire others had lasting effect.
I would also like to thank the whole team of the Flight Physiology Department and the study coordination office of the DLR for their passionate participation during the process of creating and testing the electronic checklists. Without their helpful input and support, the checklists could not have been successfully implemented.
In addition, I would like to thank my friend and former flight instructor Alexander Gerhard-Madjidi for providing me with pictures of electronic checklists of an Airbus A380. He is a role model for me and has provided me with unconditional support and funny anecdotes of his life as a pilot for many years now.
Finally, I would like to thank my wife and parents for providing me with continuous support throughout the years of my study and during the process of writing this thesis. This accomplishment would not have been possible without them. Thank you all very much.

Malte Fuhrmann

Contents

List of Figures

List of Tables

Nomenclature

Android . Android is a operating system developed by Google used on mobile devices

ASP.NET open-source framework for web server applications and dynamic websites

BDC ..... Baseline Data Collection

Cookie .. A cookie is a small data piece which is sent to a user's web browser to save information, e.g. for authentication or information on the state of a website

Diastolic Blood Pressure The diastolic blood pressure describes the arterial pressure between heartbeats when the heart muscle is resting

DLR ..... Deutsches Zentrum für Luft- und Raumfahrt, German abbreviation for the Aerospace Center of the Federal Republic of Germany

GUID ... Globally Unique Identifier - a unique reference number used for identification in software development

HTML .. HyperText Markup Language (HTML) is a markup language used for websites

JPEG ... JPEG is a method for compressing digital pictures using an adjustable compression rate

LBNP ... stands for Lower Body Negative Pressure. Negative pressure is applied to a hood which covers the lower extremities of a subject's body to intensify orthostatic hypotension.

OGTT ... Oral Glucose Tolerance Test

Orthostatic Hypotension Orthostatic hypotension (or short orthostasis) describes the resulting low blood pressure in the human body after standing up rapidly. The blood rushes from the upper body into the lower extremities which causes the blood pressure to drop.

Rongeur . Rongeur is french for rodent. A rongeur is a surgical instrument with a scoop-shaped tip with sharp edges. A rongeur is used to remove tissue or bone material during surgery or dissection.

RSL ..... Reactive Jumps in a Sledge Jump System as a Countermeasure during Longterm Bed rest (short: RSL study), described in detail in Chapter 3

SHA-512 A secure hash algorithm developed by the National Security Agency with a hash value of 512 bits

SMTP ... Simple Mail Transfer Protocol, standard for transmission of emails

SSL ..... Secure Sockets Layer, obsolete for Transport Layer Security (TLS), is a protocol that secures communication in a computer network

Systolic Blood Pressure The systolic blood pressure describes the arterial pressure when the human heart beats

Thorax .. Colloquially called chest. Anatomical term to describe the chest region between the neck and the abdomen

Vasovagal Syncope Colloquially referred to as fainting; A vasovagal syncope is caused by the vagus nerve as a reaction to insufficient blood flow to a human's brain. It can be triggered by, inter alia, standing up rapidly

Visual Studio Development environment by Microsoft, which supports many different languages

XML .... Extensible Markup Language (XML) is a markup language that is used to format text files, that are both readable from humans and computers

Chapter 1

Introduction

Checklists are a powerful tool to increase safety in aviation. They have been used in cockpits for many years and have proven to be effective to decrease the number of mistakes made by pilots. Aircraft have become far too complex for one person to perform all necessary tasks by memory without eventually making a blunder. Checklists allow users to perform actions with high reliability by giving the opportunity to pause and reflect on their actions before performing the next steps. The concept of checklists has been introduced as early as 1935 in aviation and is a primary component of key operating methods of airlines and military ever since. Nowadays, even small aeroplanes for private pilots feature checklists. Normal and abnormal checklists are available for every phase of flight and for most common failures that can occur. They are read at pre-defined stages of flight and comprise a list of items that have to be accomplished and verified by pilots at a certain time. In high workload situations like an engine on fire, pilots have to react quickly and perform first actions by memory. An abnormal checklist is read afterwards to ensure that all necessary tasks have been completed and to submit solutions to resolve the emergency situation.

Since the introduction of the first checklist, airplanes have evolved from mechanical and analogous flying machines to high complex fly-by-wire jets featuring computers that can calculate and execute precise flying maneuvers. Nevertheless, the layout and concept of checklists has not changed during this period. Modern airlines feature computer-aided checklists which are displayed by the Flight Management System, suitable to the current phase of flight. The concept itself, however, is very similar to the one of the first paper checklist being introduced over 80 years ago. No matter the format, the major function of a checklist is to ensure that the airplane is properly configured and that the crew maintains this level of safety and quality throughout the entire flight and in every flight.

The successful concept of checklists has been adapted and transferred to surgery by the World Health Organization (WHO) in 2008. Aviation and modern medicine are both complex industries where highly skilled professionals have the responsibility for other people's lives and work in a team to achieve safe results. Decisions are often high-consequence and even though human error is inevitable, harm to passengers or patients as a result of these errors is not. The WHO successfully introduced a Surgical Safety Checklist that helped to reduce the rates of death and major complications significantly. The checklist initiated a culture change in clinical operations which lead to vast improvements in surgical team member's willingness to speak up and report any potential hazards to the safety of patients without fear. Employee satisfaction is improved and a relationship of trust with superiors is established when all team members introduce one another before a surgical intervention and function as a team. Ongoing peer monitoring as part of routine tasks helps to reduce medical errors and increases patient safety.

Unlike in aviation, surgical safety checklists are not only used as a tool for checking the completeness of tasks, but also for documentation purposes. Before a patient leaves the operating room, the key concerns regarding the recovery and care of the patient are reviewed to ensure that all instructions by the surgeon are correctly passed on to nursing staff and to complete the patient's medical record documentation. Possible improvements that could be considered for future interventions are discussed, which helps to raise the standard and further increase the safety of clinical operations.

Safety checklists are more and more introduced in other fields of medicine besides surgery, but not in human biomedical research so far. Biomedical studies are similarly complex as clinical operations in surgery or intensive care units. They involve high complex experiments that are performed by changing teams, just as in cockpits or operating rooms. Effectiveness of experiments and subject safety depend on consistency and completeness of tasks. Results have to be reliable and reproducible for the experiment to be successful. Looking at the benefits for safety checklists brought to cockpits and operating rooms, it is reasonable to believe that checklists are the suitable tool to make experiments in the context of complex human biomedical studies safe and reliable.

The goal of this thesis is to implement medical safety checklists in the context of a long-term bed rest study at the German Aerospace Center (DLR) and evaluate whether or not they can improve medical safety and scientific data quality. Between January and April 2016, the second campaign of a long-term bed rest study is hosted at the :envihab facility of the DLR in Cologne. The purpose of the study is to test a newly developed exercise device as a countermeasure for physiological changes due to weightlessness or long term bed rest. During two campaigns, a total of 24 male subjects, 20 to 45 years of age, participate in the study. After a baseline data collection phase which

lasts for two weeks, the subjects lie in bed for 60 days at an inclination of 6° so that the head is lower than the body. After the bed rest, the subjects undergo post measurements for another two weeks. Twelve scientific groups have planned about 90 experiments, many of which are developed and performed by international scientists. For the checklists to be usable not only for the mentioned campaign, but also for further studies, three complex experiments were chosen that are performed by the Space Physiology Department of the DLR. These specific experiments are part of almost all bed rest studies and astronaut examinations performed at the DLR. When proven to be effective, the checklists are intended to be used for numerous studies in the future. The DLR hosts approximately one large bed-rest study every two years in Cologne and is responsible for astronaut screenings and examinations before and after space missions. The selected experiments are described briefly below:

Muscle biopsy Biopsies are performed by a medical doctor on the *m. soleus* (calf muscle) and *m. vastus lateralis* (lateral thigh muscle) three times for each subject during the study. Muscle specimens are cut in small pieces and immersed in liquid nitrogen for transportation. After the study, composition of muscle fiber types and molecular-biological changes in the tissue are examined.

Tilt table Tilt table experiments are performed twice during the campaign. The subject lies down on the tilt table in a supine position. After resting for 45 minutes, the tilt table is rapidly erected to an upright position which causes an orthostatic hypotension. The experiment is supervised by a scientist and a medical doctor who monitor the subject's vital parameters continuously throughout the experiment.

Spiro-ergometry Physical fitness and endurance performance of subjects are assessed using a bicycle ergometry test before and after the bed rest. While incrementally increasing exertion on the subject, heart rate, blood pressure, oxygen consumption and carbon dioxide output are monitored. The experiment is supervised by a medical doctor and a scientific researcher and lasts approximately one hour.

For a checklist to be effective, is has to cover all relevant safety and quality issues of an experiment. It is crucial to identify those aspects to be able to usefully integrate them into a checklist. Prior to the start of the study, the operation of experiments was discussed in detail with all personnel involved with the execution of the mentioned experiments. A protocol sheet was created for each type of experiment, reflecting a complete list of actions that need to be accomplished for a successful and safe experiment with reproducible results.

Protocol sheets must not be confused with checklists. It would be absolutely pointless to include each and every action in a checklist. It would require far too much time to read and thus it would not be practicable. Only the important items threatening a successful and safe operation when omitted must be included in a checklist.

During the Baseline Data Collection Phase at the beginning of the campaign, all three types of experiments were assessed for each subject with the help of the protocol sheets. If an action has been performed correctly and in time, the associated item was checked. The experimenters were observed silently and have not been given hints or advice during this phase. The collected data was evaluated during the bed rest phase of the subjects. If an item on a protocol sheet has been omitted or forgotten by experimenters on one or more occasions, it has been included in the checklists. All items that are self-evident components of the experiments are not included in the checklists. Only items thought to improve the overall performance of experiments in regard to safety and quality of results are covered by the checklists.

Apart from the contentual conception, the time during the bed rest phase was used to create a web server application to display the checklists. Since all three experiments are performed in different rooms and even buildings, the checklists need to be displayed on portable devices. A web server application has been developed with ASP.NET using Visual Studio 2015. The HTML and ASPX pages are touch-optimized through implementation of jQuery Mobile. The website can be easily accessed from tablet computers during experiments, as well as from desktop computers while editing checklist templates. The application holds pages for displaying checklists and for administrative tasks such as editing of templates and downloading of stored data. All information entered by users while conducting a checklist is stored in XML files on the server. Users can take pictures with the tablet's built-in camera and upload them to the server. Since this information is sensitive, the application is password protected to prevent unauthorized access. Due to the fact that the checklist has to serve the purpose of documentation, the server automatically generates an email containing all information entered by a user after an experiment has been conducted. The email is sent to the Study Coordination office, where it is printed and added to the subject's records.

The application as well as the checklists itself were tested during the recovery phase of the campaign. All of the scientists and medical doctors conducting experiments were briefed on how to handle checklists on the tablet computers. The application was used during all post measurements and thoroughly tested. While the experimenters conducted the electronic checklists, they were monitored using the same protocol sheets that have been used before. After comparing the performance with and without checklists, the effect of checklists on subject safety and data quality can be determined.

This thesis is divided into four major parts apart from Introduction and Discussion:

Chapter 2 describes the concept of checklists in aviation as well as in medicine. The chapter describes the introduction of the first checklist in aviation and the development it went through since its invention. In addition, the implementation process of checklists in clinical operations is explained with regards to the different concepts of checklists in medicine and aviation. The chapter is concluded by an assessment how suitable checklists are for biomedical research.

Chapter 3 covers the contentual conception of checklists to be used during the long-term bed rest study. It thoroughly describes the preconditions at the DLR and the purpose of the RSL study. The three experiments selected for testing the checklists are explained in detail with information about the intention and execution during the campaign.

Chapter 4 depicts the entire process of developing the web server application to display electronic checklists. The chapter contains listings of code to demonstrate the functionality of the application and how it was achieved. Screenshots of all pages allow an insight into structure and functional extent.

Chapter 5 presents the results of the performance evaluation. For each experiment, three indicators for the effectiveness of checklists in biomedical studies have been selected. Firstly, overall performance is measured, taking the entire list of actions into account that are performed during an experiment. Secondly, performance with regard to subject safety is measured. Thirdly, the input of checklists on the quality of scientific data is analyzed. The chapter is concluded by an evaluation of feedback questionnaires indicating the acceptance and contentedness of checklist application users.

Chapter 2

The Concept of Checklists

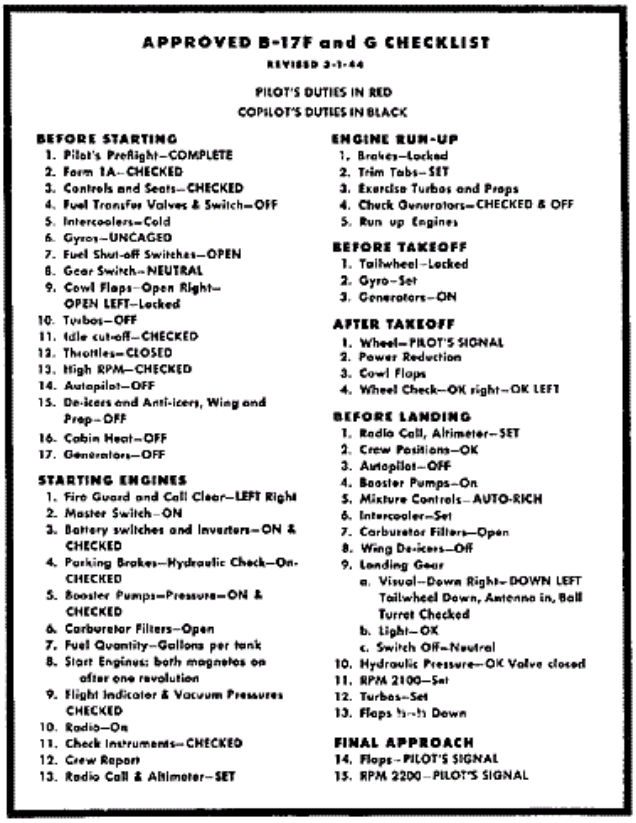
In 1903, the era of controlled flight with motored aeroplanes began with a flight performed by the Wright brothers. Self-propelled, fixed wing flying machines carried out short distance flights and laid the foundation for modern airplanes. Military forces quickly recognized the potential of being able to spy out hostile territories from the air. The competition between nations lead to rapid development of even more powerful aircraft to carry camera equipment and bombs for warfare. During the First World War, several manufacturers had established themselves in the market of aircraft development, including Boeing.

2.1 Development of Checklists in Aviation

In 1935, a new Boeing Model 299 was presented, featuring four engines and a high bomb load[1]. After five years of development, the aircraft was undergoing test flights performed by Boeing's chief test pilot at that time, Leslie Tower. On October 30, 1935, the aircraft crashed, killing both pilots and severely injuring others on board. During the following investigation it was determined that a simple control lock that had not been removed before take off has caused the crash[2]. How could an experienced pilot forget such a simple thing? Pilots during that time were expected to perform all necessary steps to conduct a flight by memory. It was realized that the systems of an aircraft had become quite complex. It was not possible anymore to memorize all required steps for configuring a complicated aircraft. Boeing came up with a simple yet innovative and effective solution: a checklist of the size of a postcard. It enlisted mundane tasks a pilot should perform during taxiing, taking off and landing.

[1] [Har12, Historic Wings]
[2] [Har12, Historic Wings]

After implementing the checklist, the Model 299 traveled 1.8 million miles without a single accident and played a significant role in the Second World War[3]. Figure 2.1 shows the checklist that was introduced by Boeing after the crash of Model 299 in its original version as published in the airplane's training manual[4]. Please note that the model has been re-designated the B-17 *Flying Fortress*. The checklist is divided into sections for different phases of flight. For each section, a number of *items* are listed that have to be accomplished and verified by the pilots.



APPROVED B-17F and G CHECKLIST

REVISED 3-1-44

PILOT'S DUTIES IN RED
COPILOT'S DUTIES IN BLACK

BEFORE STARTING
1. Pilot's Preflight—COMPLETE
2. Form 1A—CHECKED
3. Controls and Seats—CHECKED
4. Fuel Transfer Valves & Switch—OFF
5. Intercoolers—Cold
6. Gyros—UNCAGED
7. Fuel Shut-off Switches—OPEN
8. Gear Switch—NEUTRAL
9. Cowl Flaps—Open Right—OPEN LEFT—Locked
10. Turbos—OFF
11. Idle cut-off—CHECKED
12. Throttles—CLOSED
13. High RPM—CHECKED
14. Autopilot—OFF
15. De-icers and Anti-icers, Wing and Prop—OFF
16. Cabin Heat—OFF
17. Generators—OFF

STARTING ENGINES
1. Fire Guard and Call Clear—LEFT Right
2. Master Switch—ON
3. Battery switches and Inverters—ON & CHECKED
4. Parking Brakes—Hydraulic Check—On—CHECKED
5. Booster Pumps—Pressure—ON & CHECKED
6. Carburetor Filters—Open
7. Fuel Quantity—Gallons per tank
8. Start Engines: both magnetos on after one revolution
9. Flight Indicator & Vacuum Pressures CHECKED
10. Radio—On
11. Check Instruments—CHECKED
12. Crew Report
13. Radio Call & Altimeter—SET

ENGINE RUN-UP
1. Brakes—Locked
2. Trim Tabs—SET
3. Exercise Turbos and Props
4. Check Generators—CHECKED & OFF
5. Run up Engines

BEFORE TAKEOFF
1. Tailwheel—Locked
2. Gyro—Set
3. Generators—ON

AFTER TAKEOFF
1. Wheel—PILOT'S SIGNAL
2. Power Reduction
3. Cowl Flaps
4. Wheel Check—OK right—OK LEFT

BEFORE LANDING
1. Radio Call, Altimeter—SET
2. Crew Positions—OK
3. Autopilot—OFF
4. Booster Pumps—On
5. Mixture Controls—AUTO-RICH
6. Intercooler—Set
7. Carburetor Filters—Open
8. Wing De-icers—Off
9. Landing Gear
   a. Visual—Down Right—DOWN LEFT Tailwheel Down, Antenna in, Ball Turret Checked
   b. Light—OK
   c. Switch Off—Neutral
10. Hydraulic Pressure—OK Valve closed
11. RPM 2100—Set
12. Turbos—Set
13. Flaps [illegible] Down

FINAL APPROACH
14. Flaps—PILOT'S SIGNAL
15. RPM 2200—PILOT'S SIGNAL

Figure 2.1: Excerpt of Boeing's B-17 Checklist

[3][Wor16, Patient Safety Alliance]
[4][Uni44, B-17 Training Manual]

At first, pilots felt that checklists insulted their ability to memorize all required steps to conduct a safe flight in an aircraft. Soon, the techniques of using checklists were practiced and perfected which resulted in a lower workload for pilots and increased margins of safety[5].

Since the introduction of the first checklist, airplanes have evolved from mechanical and analogous flying machines to high complex fly-by-wire jets featuring computers that can calculate and execute precise navigation maneuvers. However, checklists have not undergone any conceptual rethinking or design changes during this period[6]. A modern airliner checklist is very similar to the B-17 checklist shown in Figure 2.1 on the preceding page. They all follow the same strategy:

1. Reading an item from the checklist
2. Performing the associated task or verifying the correct setting
3. Verbally confirming the performed action

Nowadays, even small aircraft for private pilots feature checklists. Large airliners have normal and abnormal checklists for every phase of flight and for most common failures that can occur during a flight. Modern aircraft like the Airbus A 380 feature computer-aided checklists. The Flight Management System displays the checklist that suits the present state of flight. The pilot does not have to leaf through a handbook to find the right checklist and has both hands free while reading it. Figure 2.2 on the next page exemplarily shows a normal checklist of an A 380 that is read before takeoff.

The airplane automatically ticks off all items that are detected to be correct by the system itself and thus informs the pilot about the status of each checklist item. The pilot has to perform the tasks associated with the items that are remaining. Once items are accomplished, the computer steadily clears the screen which results in enhanced clarity and a reduction in time needed to conduct the checklist.

[5][Har12, Historic Wings]
[6][DW90, Degani and Wiener]

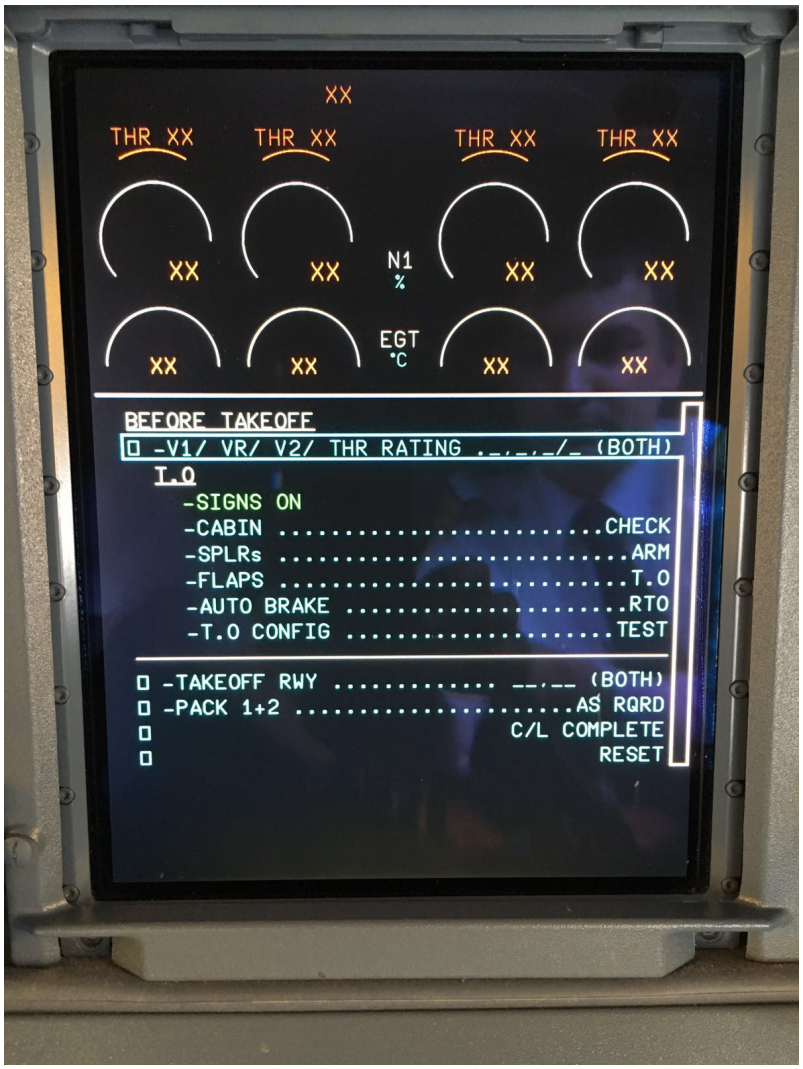


Figure 2.2: Computer-Aided Checklist in an Airbus A380 (© Alexander Gerhard-Madjidi, 2016)

No matter the format, the major function of a checklist is to ensure that the crew will properly configure the plane for flight and maintain this level of quality throughout the flight and in every flight.[7]

[7] [DW90, Degani and Wiener]

2.2 Introduction of Checklists in Medical Operations

In the complex world of modern medicine, as in aviation, human error is inevitable. Harm to patients as a result of these errors is not and can be significantly reduced. Checklists allow users to perform actions with high reliability by giving the opportunity to pause and reflect on their actions before performing the next steps. In 2004, the World Health Organization (WHO) launched the *World Alliance for Patient Safety* to put focus on problems of patient safety. Every two years, a *Global Patient Safety Challenge* is created. The second challenge in 2006 paid attention to safe surgery. Therefore, a Surgical Safety checklist[8] (which can be seen below in Figure 2.3) has been published, which takes the concepts and principles of aviation checklists and applies them to surgery.

Surgical Safety Checklist

World Health Organization | Patient Safety
A World Alliance for Safer Health Care

Before induction of anaesthesia → **Before skin incision** → **Before patient leaves operating room**

(with at least nurse and anaesthetist) | (with nurse, anaesthetist and surgeon) | (with nurse, anaesthetist and surgeon)

Has the patient confirmed his/her identity, site, procedure, and consent?
- ☐ Yes

Is the site marked?
- ☐ Yes
- ☐ Not applicable

Is the anaesthesia machine and medication check complete?
- ☐ Yes

Is the pulse oximeter on the patient and functioning?
- ☐ Yes

Does the patient have a:

Known allergy?
- ☐ No
- ☐ Yes

Difficult airway or aspiration risk?
- ☐ No
- ☐ Yes, and equipment/assistance available

Risk of >500ml blood loss (7ml/kg in children)?
- ☐ No
- ☐ Yes, and two IVs/central access and fluids planned

- ☐ **Confirm all team members have introduced themselves by name and role.**
- ☐ **Confirm the patient's name, procedure, and where the incision will be made.**

Has antibiotic prophylaxis been given within the last 60 minutes?
- ☐ Yes
- ☐ Not applicable

Anticipated Critical Events

To Surgeon:
- ☐ What are the critical or non-routine steps?
- ☐ How long will the case take?
- ☐ What is the anticipated blood loss?

To Anaesthetist:
- ☐ Are there any patient-specific concerns?

To Nursing Team:
- ☐ Has sterility (including indicator results) been confirmed?
- ☐ Are there equipment issues or any concerns?

Is essential imaging displayed?
- ☐ Yes
- ☐ Not applicable

Nurse Verbally Confirms:
- ☐ The name of the procedure
- ☐ Completion of instrument, sponge and needle counts
- ☐ Specimen labelling (read specimen labels aloud, including patient name)
- ☐ Whether there are any equipment problems to be addressed

To Surgeon, Anaesthetist and Nurse:
- ☐ What are the key concerns for recovery and management of this patient?

This checklist is not intended to be comprehensive. Additions and modifications to fit local practice are encouraged.

Revised 1 / 2009

Figure 2.3: Surgical Safety Checklist

[8] [Wor08a, World Health Organization]

An estimated number of 234 million major operations are performed around the world each year, corresponding to one operation for every 25 people alive[9]. Surgical care is of high importance for the health care of human population worldwide. International experts worked in groups to identify areas that need improvement and to define a core set of safety standards that can be applied in all WHO member states[10]. The result is a checklist that is a simple and practical tool that can be used by surgical teams around the world.

The checklist's form and content is divided into three parts. The first part of the list is read before the anesthesia is inducted to check whether or not the correct patient is in the operating room and all necessary preparations have been performed. At this point, at least one nurse and an anesthesia professional are with the patient.

The second part of the checklist is read and confirmed by the entire operating team before a skin incision is performed. All team members have to be introduced to one another by name and role, which is important for team building and clear role assignment. The procedure and anticipated risks are reviewed by the surgeon in order to prepare the team of what to expect from the surgical operation.

The third and last part of the checklist assures that the number of surgical tools matches the count before the operation. If a specimen has been taken from the patient, it has to be correctly labeled with his or her name. The key concerns regarding the recovery and care of the patient are reviewed aloud to ensure that all instructions by the surgeon are correctly passed on to nursing staff.

The results of a year-long pilot study of the Surgical Safety checklist in eight developed and developing countries were published in January 2009, in the New England Journal of Medicine[11]. Introduction of the WHO checklist was associated with marked improvements in surgical outcomes. Postoperative complication rates fell by 36% on average, and death rates fell by a similar amount[12]. Thus it can be said, that the WHO Surgical Safety checklist can improve the safety of surgical patients in diverse clinical and economic environments significantly[13].

[9][Wor08b, Safe surgery saves lives, page 8]
[10][Wor08b, Safe surgery saves lives, page 12]
[11][Wor16, Checklist Effect]
[12][HWB+09, New England Journal of Medicine, page 496]
[13][HWB+09, New England Journal of Medicine, page 496]

2.3 Adaptation of Checklists for Scientific Research

Biomedical studies are similarly complex as clinical operations in surgery or intensive care units. The previous subsections have shown that a checklist can be useful whenever a system becomes too complex. At some point, one person cannot possibly perform all necessary steps from memory without eventually making a mistake. In aviation as in medicine, decisions are high-consequence. Even though a checklist is very useful in normal operations, there are situations when there is simply no time to read a checklist. In high workload situations like a patient with cardiac arrest in surgery or an engine on fire in an airliner, actions have to be performed by memory. After the so-called *memory items* have been performed, a checklist can help the user to reflect his or her actions to ensure that no step has been forgotten. Thus, it has to be distinguished for each purpose whether or not a checklist is feasible.

It is reasonable to think that checklists can be successfully transferred to other areas. Biomedical studies involve high complex experiments which are performed by changing teams. Effectiveness and subject safety depend on consistency and completeness of tasks during experiments. In situations where there is a lot of routine, but also a lot of stress, checklists are a useful tool in making experiments safe and reliable. The positive experiences with checklists in aviation and modern medicine are combined and adapted to suit the requirements of scientific research. Checklists shall not only be used for checking purposes as in aviation, but also for documentation. The following chapters describe thoroughly the implementation of electronic checklists for three complex experiments during biomedical studies at the German Aerospace Center (DLR).

Chapter 3

Content Conception

The following chapter describes the preconditions at the German Aerospace Center and the contentual conception of checklists to be used during the bed rest study RSL.

3.1 Bed Rest Study RSL

The purpose of the study is to test a newly developed exercise device as a countermeasure for physiological changes due to weightlessness or long term bed rest. The *Sledge Jump System* allows subjects to 'jump' in a horizontal position as well as in space using low-pressure cylinders to recreate gravity[1].

During two campaigns, a total of 24 male subjects, 20 to 45 years of age, participate in the study. For a duration of 60 days, the subjects lie in bed at an inclination of 6° so that the head is lower than the body. Two weeks before and after the study, the participants are measured to chart changes in their bodies. Similar to astronauts, people on earth suffer from long term bed rest as their bones and muscles lose strength from underuse.

Twelve scientific groups have planned about 90 experiments that examine the cardiovascular system, how the brain copes with the head down tilt bed rest and the effect of simulated gravity on specific organs, just to name a few. Many of the experiments are developed and performed by international scientists, who come to Cologne especially for this study. The checklists are to be tested during the second RSL campaign, but have to be usable for further studies as well. Therefore, three core data experiments were selected that are performed by the Space Physiology Department of the DLR. The experiments are part of almost all bed rest studies conducted at the DLR. The checklists are intended to be used for numerous experiments in the future, when proven to be effective.

[1] [ESA15, ESA]

3.2 Selection of Experiments

The effectiveness of checklists in biomedical research studies is to be tested during three different experiments to begin with. The three most complex experiments of the RSL bed rest study were selected.

3.2.1 Spiro-ergometry

A subject's physical fitness and endurance performance before and after the bed rest is assessed using a bicycle ergometry test. While incrementally increasing exertion on the subject, heart rate, blood pressure, oxygen consumption and carbon dioxide output are monitored. The experiment lasts approximately one hour and is performed once before and once after the bed rest. The test is supervised by a scientific researcher as well as a medical doctor. The doctor monitors the subject's vital parameters using cuff/stethoscope auscultation and a device called Innocor Innovision, which is shown in Figure 3.1. The device measures cardiac output and oxygen uptake during exercise using a breathing mask and pulse oxymetry[2].

Figure 3.1: Spiro-ergometry (© COMPEK Medical Services, s.r.o., 2016)

[2][Cer, COMPEK Medical Services, s.r.o.]

3.2.2 Muscle Biopsy

The RSL study includes muscle biopsies on three days: before and at the end of the bed rest as well as ten days after bed rest. Biopsies are performed on the *m. soleus* (calf muscle) and *m. vastus lateralis* (lateral thigh muscle). The procedure is performed by experienced medical doctors in accordance with hygiene requirements and clinical surgery standards.

After local anesthesia, an incision of skin and fascia of the respective muscle is performed. Muscle samples are extracted using a rongeur or a biopsy needle. Figure 3.2 shows a surgeon releasing muscle material from a rongeur on to the sample preparation table. The biopsy is cut in small pieces and immersed in liquid nitrogen.

Figure 3.2: Muscle Biopsy (© DLR, 2016)

With the help of the biopsied material, composition of muscle fiber types and molecular-biological changes in the tissue are examined. The effectiveness of the newly developed training method can thus be verified. This experiment is by far the most complex one because it involves more personnel and equipment than the other experiments. Therefore, the number of items that are to be covered by a checklist is the highest as described in Chapter 3.3 on page 17.

3.2.3 Tilt Table

A tilt table is an apparatus that is used to rapidly change a subject's position from supine to standing up and vice versa. When being erected, the body has to cope with orthostastic hypotension. The blood rushes from the thorax into the lower extremities which results in a decrease in systolic and diastolic blood pressure. This effect can even be intensified by applying negative pressure to the lower body. Figure 3.3 shows a subject lying on the tilt table with the LBNP cover installed. The pressure can be decreased in steps of 10 mm Hg to shorten the time needed to cause a vasovagal syncope.

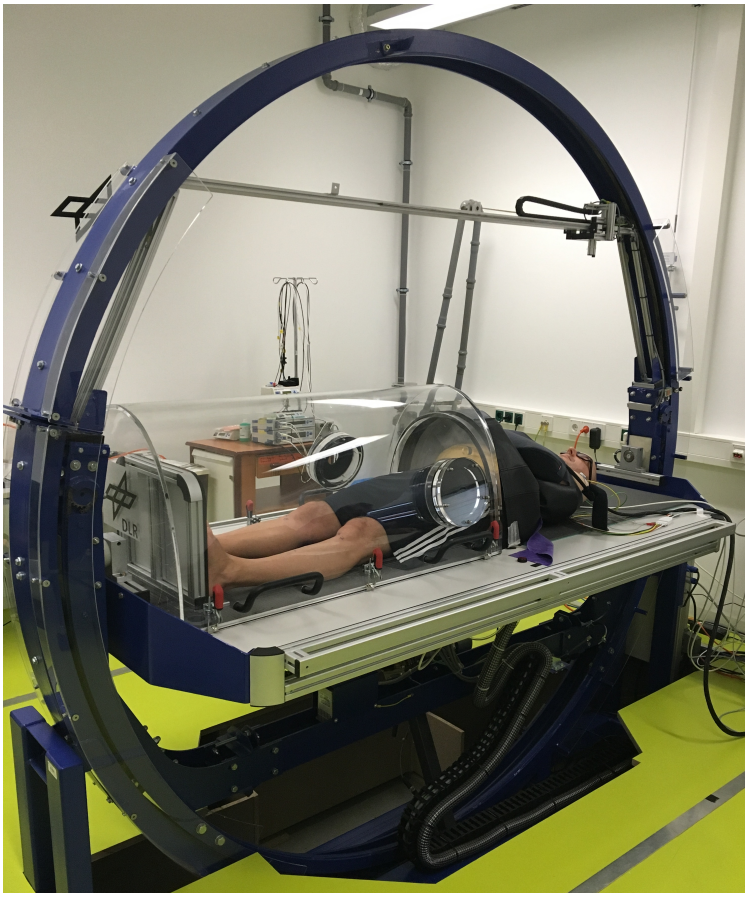


Figure 3.3: Tilt Table (© DLR, 2016)

Tilt table experiments are performed twice during the RSL study, once before and once at the end of the bed rest. A scientific supervisor and a medical doctor monitor the vital parameters of the subject throughout the experiment. The subject rests for 45 minutes before the tilt table is erected. Before the LBNP session is started, cardiovascular data is recorded for fifteen minutes. The test is canceled shortly before a syncope.

3.3 Determination of Evaluation Criteria

In order for a checklist to be effective, it has to cover all relevant safety and quality issues of an experiment. It is important to identify those aspects to be able to usefully integrate them into a checklist. Prior to the start of the second RSL campaign, the operation of experiments was discussed in detail with the Flight Physiology Department and the Study Coordination Team. All persons involved described their individual working steps and tasks during an experiment. Thus, each protocol sheet reflects a complete list of actions that lead to a successful and safe experiment with reproducible results.

Figure 3.4 shows an excerpt of a protocol sheet for muscle biopsies. As stated before, it is by far the longest and most complex one. A copy of the complete protocol sheets for all three experiments can be found in Attachments *A* through *C*.

DLR **Experiment Protocol**	Implementation of electronic checklists for muscle biopsies, tilt table experiments and spiro-ergometry in order to increase medical safety and scientific quality during biomedical studies at the DLR
Muscle Biopsy	
Date:	Started at:
Subject:	Ended at:
1) Initial check during pick-up	
Identity of subject confirmed	☐
Subject is ready for surgical intervention	☐
Calf and thigh shaved'	☐
Subject wears short pants	☐
Subject went to restroom w/i past 15 min.	☐
Wheelchair, urine bottle and route card on hand	☐
If OGTT, then:	☐
Subject did not eat during the past 10h	☐
Blanket/reading material etc. ready	☐
I.v. in place	☐
2) Pre-anesthesia check	
Biopsy room prepared according to DLR hygienic standards	☐

Figure 3.4: Protocol Sheet for Data Collection during Muscle Biopsies

Each protocol sheet is divided into three separate parts. The first part covers all actions performed when a subject is picked up. The twelve subjects for each campaign live in a self-contained living module which they are not allowed to leave unattended. Therefore, subjects have to be picked up and dropped off for an experiment. The second part of the protocol covers the

experiment itself and the third part eventually the drop off. This division has proven to be highly effective, hence the checklists are split into three separate parts as well. It is advantageous to have an identical framework for each experiment in order to simplify programming and to minimize familiarization time for experimenters.

Protocol sheets must not to be confused with checklists. It would be absolutely pointless to check each and every item on the sheet for each experiment. This procedure would require far too much time and would not be practicable. Only the important items have to be implemented to assure medical safety and scientific quality.

The RSL study campaign includes three phases (listed in chronological order):

1. *Baseline Data Collection* Phase (BDC): 14 days for each subject
2. *Head-Down Tilt* Phase (HDT): 60 days for each subject
3. *Recovery* Phase: 15 days for each subject

During the BDC phase, all three types of experiments were monitored for each subject with the help of the protocol sheets. If an action has been performed correctly and in time, the associated check box was checked. The experimenters were observed silently and have not been given hints or advice at this time.

The collected data was transferred to an Excel worksheet for evaluation. Muscle biopsies were performed twice for each subject on the same day with a break in between for a glucose tolerance test (OGTT). Therefore, two muscle biopsy data sets exist for each subject. All observed experiments were evaluated during the HDT phase. If an item on a protocol sheet has been omitted or forgotten by the experimenter, it has been included in the corresponding checklist. All items that are self-evident components of the experiments do not need to be specifically mentioned in a checklist. Consequently, the checklists only include items thought to improve the overall performance of experiments in regard to safety and quality of results.

The protocol sheets were used again in unmodified form during the recovery phase while the checklists were introduced. Thus, after comparing the results from BDC and recovery phase, it can be determined whether or not the checklists are effective. The results are thoroughly discussed in Chapter 5 on page 71.

Chapter 4

Web Server Application Development

The goal of this thesis is to implement electronic checklists in order to enhance medical safety in complex human biomedical studies at the DLR. In the beginning, three different experiments are to be covered by the checklists, each taking place in different rooms with various prerequisites. The challenge is to develop an application that meets all the requirements and can easily be adapted for further experiments.

As stated in Chapter 3, a checklist consists of three parts, of which two are read in the living module when a subject is picked up or dropped off. The middle part of the checklist is read right before the start of the actual experiment in the assigned room. Therefore, a portable device is needed to display the checklists. The choice came down to a Samsung Galaxy Tab S, which features a 8.4 inch screen and long battery life.

Since Galaxy tablets use the operating system Android, it stands to reason to develop a mobile app. This would have the benefit of being independent of local circumstances and infrastructure. However, the disadvantages outweigh the advantages of a mobile app. Firstly, even the smallest modifications have to be realized through updates. Secondly, the checklists are not only designed to enhance safety, but also to serve the purpose of documentation. Using a mobile app, transferring data from several tablet computers to a central computer would be a cumbersome and tedious task. A web server is needed in any case for storing and evaluation of conducted checklists, hence it can be used to store the whole application.

In the final analysis, a web server application is the best choice to meet the requirements. The website can easily be accessed from tablets when performing an experiment, as well as from desktop computers when modifying checklists. The only necessary condition for this to work is that the tablets have access to a wireless network, which is fulfilled in this case.

The web server application has been developed with ASP.NET using Visual Studio 2015. The HTML and ASPX pages have been touch-optimized using jQuery Mobile to make them more usable on a tablet. The structure and functionality of the application are described in detail in the following chapters.

4.1 Application Structure

The application holds a total of twelve ASPX pages, three XML files, one HTML page and four folders, containing additional XML files and pictures. Figure 4.1 shows the application structure and access rights of different users. The administrator is the only user who has access to all functions. Experimenters can only choose a template and conduct checklists. The following chapters will explain the functions of the application in detail.

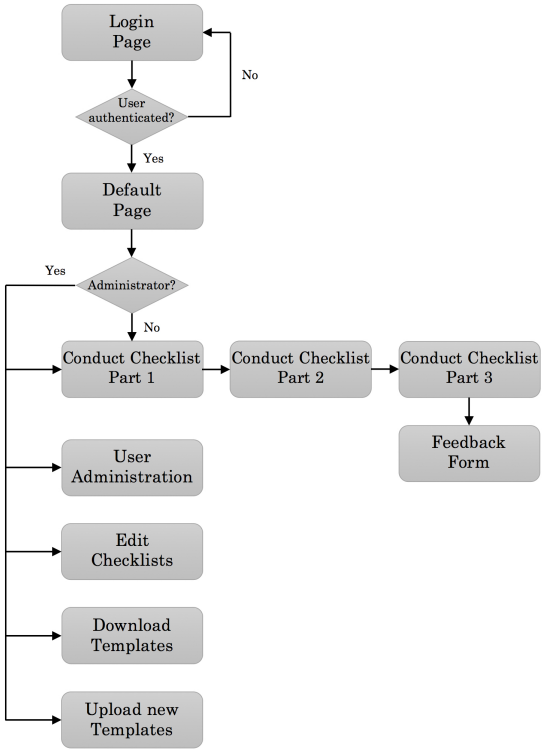


Figure 4.1: Application Flowchart

4.2 Configuration and User Authorization

The web server holds sensitive information, therefore the application has to be protected from unauthorized access. If a user has not logged on to the site and

requests access to a page, he is redirected to the login page. The configuration of the login page and the mode of authentication is done in the web.config file of the web server application.

4.2.1 Web.config

ASP.NET allows different ways of user authentication. The most convenient one for this application is a form-based authentication, since only a few users are using the application. This method allows the programmer to choose how to store the password and keeps the amount of code required to a minimum.

The forms authentication method uses a cookie named ".ASPXAUTH", which is created when a user logs on to the page. The timeout is used to limit the lifetime of the authentication session. Its default value is only 30 minutes, which has to be significantly increased. Most experiments, which this application is being used for, last longer than half an hour. To increase the security of the authentication ticket, *protection* is set to *all*, which means that the authentication cookie is both encrypted and digitally signed.

```
<configuration>
        <system.web>
                <compilation debug="true" strict="false"
                    explicit="true"
                        targetFramework="4.5" />
                <httpRuntime targetFramework="4.5" />
                <customErrors mode="Off" />
            <authentication mode="Forms">
                        <forms name=".ASPXAUTH" loginUrl="Login.aspx"
                            timeout="300"
                                protection ="All" />
                </authentication>
                <authorization><deny users="?"/></authorization>
        </system.web>
        <appSettings>
                <add
                    key="ValidationSettings:UnobtrusiveValidationMode"
                    value="None" />
        </appSettings>
</configuration>
```

The authorization element specifies, that all users are to be denied, if they are not properly authenticated. Finally, the unobtrusive validation mode must be disabled to allow for custom validation settings in ASPX forms. This feature is automatically included in the .NET 4.5 framework and has to be manually disabled.

4.2.2 Login.aspx

The login page hosts a simple form which allows the user to insert his credentials as seen in Figure 4.2.

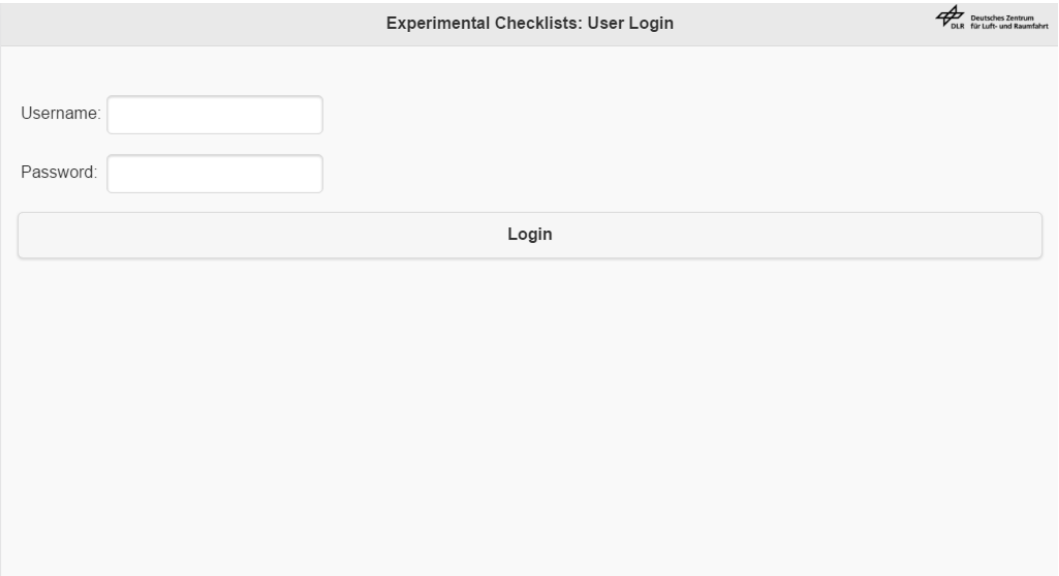


Figure 4.2: Login Page

A click on the *Login* button starts a sub procedure, which compares the entered username and password to the stored credentials. Since storing passwords in plain text would be very insecure, the password is hashed first. Hashing a password means that one takes a password of variable length and creates a cryptic password of a fixed length. In case the XML file, in which the passwords are stored, gets compromised, hashed passwords are very hard to reverse. The public function shown below uses SHA-512 (Secure Hash Algorithm with a length of 512 bits) to encrypt the passwords.

```
1 Public Function Hash512(password As String, salt As String) As String
2         Dim convertedToBytes As Byte() =
              Encoding.UTF8.GetBytes(password & salt)
3         Dim hashType As HashAlgorithm = New SHA512Managed()
4         Dim hashBytes As Byte() =
              hashType.ComputeHash(convertedToBytes)
5         Dim hashedResult As String = Convert.ToBase64String(hashBytes)
6         Return hashedResult
7 End Function
```

To increase the security of the hash, a randomly generated value called salt is used. The combination of plain password and salt value is then hashed, which makes it even harder to decode.

The password and salt value for each user is stored in a file called *users.xml*. Here is one example for an user entry:

```
<user>
        <username>user</username>
        <hashedpassword>
                cu1Si4RFgyDkRJjjYq7L3I5Kr...431h6s79dhRO4dP+bKcNXSvziw==
        </hashedpassword>
        <salt>rcHdUgZFG7Ta1eXFPgHu</salt>
</user>
```

When a user clicks the login button, the procedure opens the *users.xml* file to check if an entry with that username exists. If that is the case, the entered password is salted with the stored salt value and then hashed. If the two hashed passwords match, the user is authenticated and the application redirects to the default page. If the credentials are not correct, access is denied.

```
Sub Button_Login_Click(Sender As Object, E As EventArgs)
 If Page.IsValid Then
 Try
  Dim users_xml = XElement.Load(MapPath("~/users.xml")
  Dim password_to_compare =
   From stored_user In users_xml.<user>
   Where stored_user.<username>.Value = textbox_username.Text
   Select
        stored_password_salted_hashed=
                stored_user.<hashedpassword>.Value.ToString,
        stored_salt = stored_user.<salt>.Value.ToString

   Dim entered_password_salted_hashed =(Hash512(textbox_password.Text,
        password_to_compare.FirstOrDefault.stored_salt))

   If entered_password_salted_hashed =
        password_to_compare.FirstOrDefault.stored_password_salted_hashed
   Then
    FormsAuthentication.RedirectFromLoginPage(textbox_username.Text,False)
   Else
        Errormessage.Text = "The credentials you have entered are not
            correct. Please try again."
   End If
   Catch ex As Exception
  End Try
  End If
End Sub
```

4.2.3 UserAdministration.aspx

The administrator of the application has access to a few exclusive features such as the user administration page shown in Figure 4.3. This page allows the administrator to add or delete users as well as assigning new passwords.

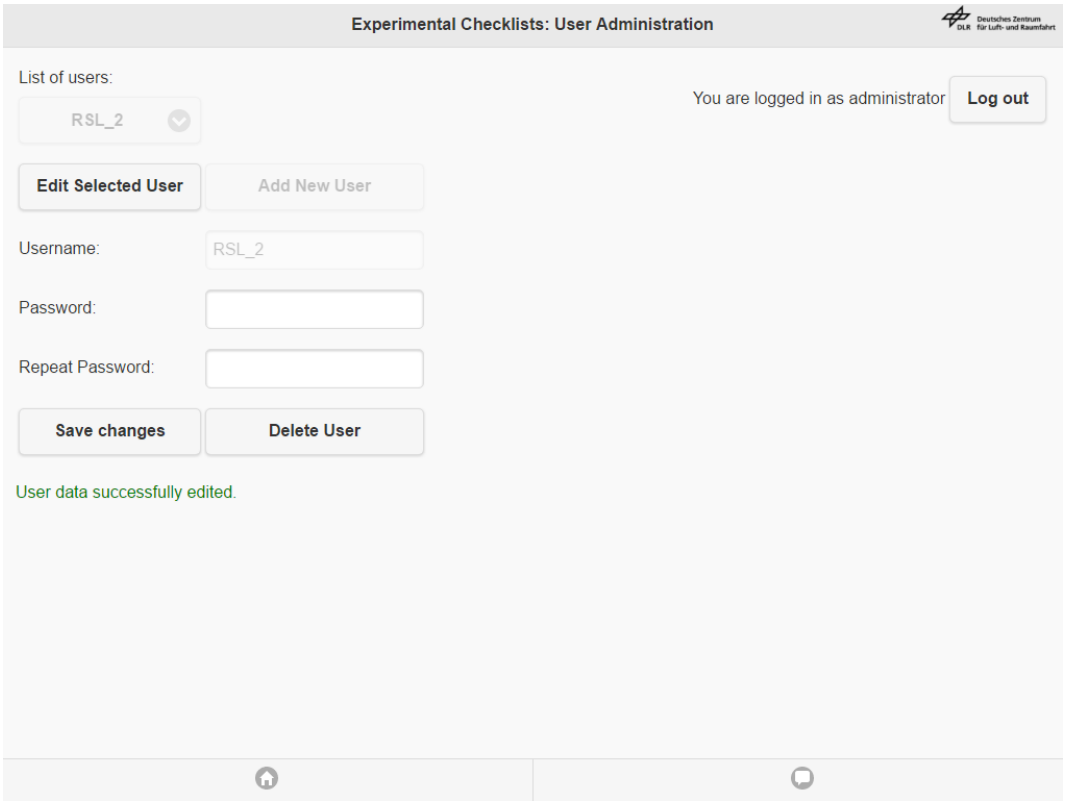


Figure 4.3: User Administration

As described in Chapter 4.2.2, a salt value is used to hash passwords. The following excerpt from the code-behind file shows the function that generates a random salt:

```
1  Public Function CreateRandomSalt() As String
2   Dim mix As String =
       "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"
3   Dim salt As String = ""
4   Dim rnd As New Random
5   Dim sb As New StringBuilder
6   For i As Integer = 1 To 20
7         Dim x As Integer = rnd.Next(0, mix.Length - 1)
8         salt &= (mix.Substring(x, 1))
9   Next Return salt
10 End Function
```

The salt is generated using a string builder. It contains twenty characters, which are randomly chosen from a list of lower and upper case letters as well as numbers. The value of the salt is unique to each user and to each password. Each time the administrator sets a new password for a user or saves a new user, a different salt is created. The procedure for saving a new user can be seen below:

```
1 Protected Sub Button_save_new_user_click(ByVal sender As Object,
    ByVal e As System.EventArgs)
2  Dim entered_username = textbox_username.Text
3  Dim used_salt = CreateRandomSalt()
4  Dim entered_password_salted_hashed = (Hash512(textbox_password.Text,
     used_salt))
5  Dim repeated_password_salted_hashed =
     (Hash512(textbox_repeat_password.Text, used_salt))
```

The username does not need to be encrypted, so only the two passwords entered by the administrator are hashed. As described before, a new salt is created and added to the plain password to create the hash. Both hash values then need to be compared to make sure the administrator repeated the password correctly. If that is not the case, an error message informs the administrator about the inconsistency.

Before the new user information can be saved in an XML file, it needs to be checked if a user with the same username already exists. The old user data has be overwritten or to be more precise, it has to be deleted before the new information can be stored. Afterwards, a new XElement is created and added to *users.xml*. After the page reloads, a message informs the administrator about the success. At the same time, the dropdownlist in the upper left corner is filled dynamically with a list of all users stored in the XML file including the latest one just added.

```
If entered_password_salted_hashed = repeated_password_salted_hashed
    Then
  Dim users_file = MapPath("~/users.xml")
  Dim users_xml = XElement.Load(users_file)
  Dim current_user_xml =
   From users In users_xml.<user>
   Where users.<username>.Value = entered_username
  current_user_xml.Remove()
  users_xml.Save(users_file)
  Dim new_user_password_xml As XElement =
        <user>
                <username>
                        <%= entered_username %>
                </username>
                <hashedpassword>
                        <%= entered_password_salted_hashed %>
                </hashedpassword>
                <salt>
                        <%= used_salt %>
                </salt>
        </user>

  users_xml.Add(new_user_password_xml)
  users_xml.Save(users_file)
  Successmessage.Text = "New user successfully added."
 Else
  Errormessage.Text = "The passwords you have entered do not match.
      Please try again."
 End If
End Sub
```

For reasons of simplicity, the whole functionality of this website is not discussed here. A copy of the complete source code is attached to this thesis on a CD-ROM.

4.3 Default Page

After logging in, the user is redirected to the default page. It alters the functions it offers depending on which user is logged in at the moment. As described before, an experimenter can only conduct checklists. Therefore the options *Checklist Editing*, *Upload Checklists* and *User Administration* are only shown to the administrator. Figure 4.4 shows an overview of all functions available on the default page.

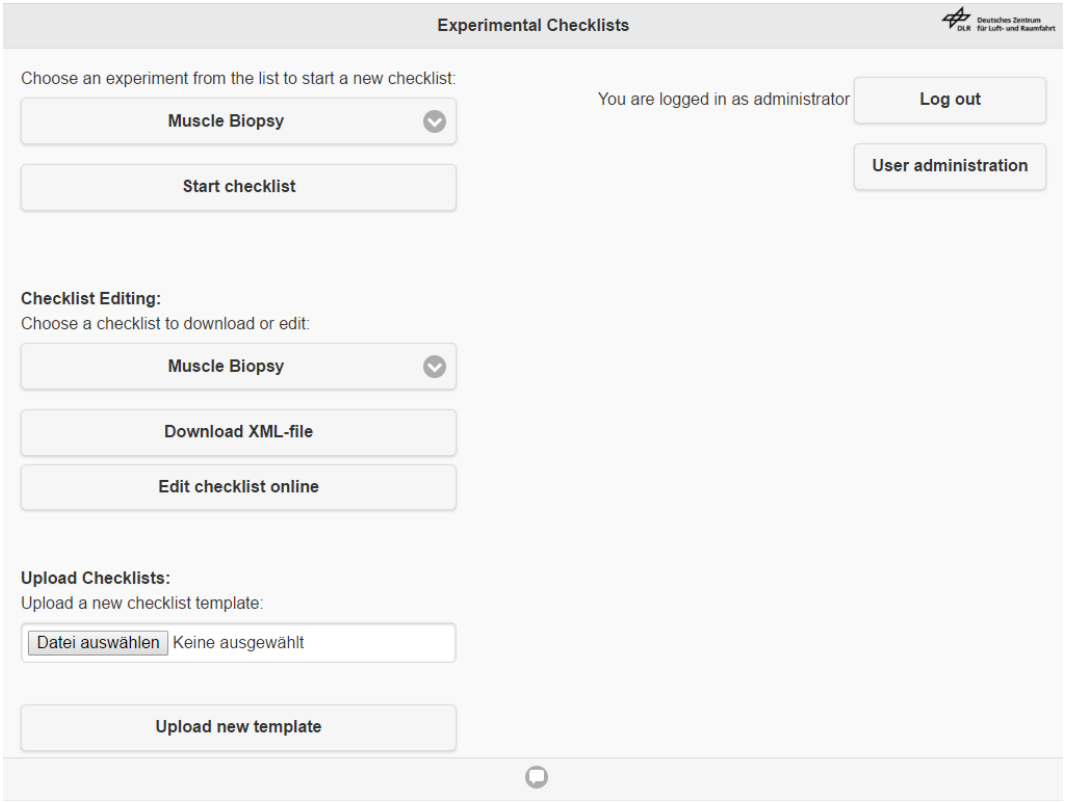


Figure 4.4: Default Page

During page load, the identity of the user is inquired. In case the user is not authenticated, he is redirected to the login page. Upon successful authentication, the current username is shown in a label in the upper right corner. The next step is to fill the dropdownlist showing the editable templates with content. That only takes place, when the user is the administrator. If a different user logs in , the administrator's controls are hidden.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles Me.Load
 If Not IsPostBack Then
  If (Context.User.Identity.IsAuthenticated) Then
   label_current_user.Text = "You are logged in as " +
      Context.User.Identity.Name
   If Context.User.Identity.Name.ToString = "administrator" Then
    Dim editable_templates_xml =
        XElement.Load(MapPath("~/List_of_templates.xml"))
    Dim list_of_editable_templates =
   From template In editable_templates_xml.<template>
   Select
    title = template.<title>.Value,
    ID = template.<GUID>.Value
   Order By title

    ddl_checklists.DataSource = list_of_editable_templates
    ddl_checklists.DataTextField = "title"
    ddl_checklists.DataValueField = "ID"
    ddl_checklists.DataBind()
   Else
    ddl_checklists.Visible = False
    ...
    label_upload.Visible = False
   End If
```

The top dropdownlist showing the templates available for conducting a checklist is visible to all users. It has to be filled with the same information as the second list, described before. The information needed is stored in an XML file named *List_ of_ templates.xml* which is shown in the following listing:

```
<dataroot>
       <template>
              <GUID>74D6B777-8399-4FE1-B55B-7048963DECE2</GUID>
              <title>VO2max</title>
       </template>
       <template>
              <GUID>98A17DEE-3C43-4C44-9ACE-B3E04531CF0A</GUID>
              <title>Tilttable</title>
       </template>
       <template>
              <GUID>DC18C032-DB3D-4D9E-B0F7-E8CF9FF2B665</GUID>
              <title>Muscle Biopsy</title>
       </template>
</dataroot>
```

At this stage, the application is only used for three different experiments. The list will become longer, when templates are edited and more experiments are added over time. Thus it makes sense to store this information in a separate file, even though it may not look like it right now. In case two templates accidentally have the same name, the unique ID's help to distinguish between them.

After the dropdownlists have been filled with the templates available, the user can choose one and start a checklist. Before being redirected to the next page, the user's browser needs to store information on which template is selected. A session variable would normally be the first choice, because it works like a short-term memory buffer. It stays active as long as the user's session is valid and can transfer important information between different pages. For this purpose however, a cookie is more suitable, since some experiments last several hours, in which the tablets are in stand-by mode for long periods of time. Closing a browser and shutting off a device can cause sessions to end. A cookie survives this unscathed and remains in the browser's memory for a predefined amount of time. In this case, the lifetime of the cookie is set to twelve hours.

```
1 Protected Sub Button_start_checklist_click(ByVal sender As Object,
    ByVal e As System.EventArgs)
2  Dim selected_template = ddl_experiments.SelectedValue.ToString
3  Dim new_cookie As New HttpCookie("selected_template")
4  new_cookie.Value = selected_template
5  new_cookie.Expires = DateTime.Now.AddHours(12)
6  Response.Cookies.Add(new_cookie)
7  Response.Redirect("Conduct_checklists.aspx")
8 End Sub
```

The procedure for starting and editing a checklist is very similar. In both cases, a cookie named *selected_ template* is created storing the ID of the template selected. Only the destination page, the user is redirected to, is different.

One feature available solely to the administrator is the download of checklist templates. Even though a whole page is available for online editing of checklists, the possibility of editing and creating templates offline with a software of one's choice is very advantageous. The following sub procedure provides the user with the selected XML file using the browser's download function.

```
Protected Sub Button_download_xml_Click(ByVal sender As Object, ByVal
    e As System.EventArgs)
 Dim selected_template_file = ddl_checklists.SelectedValue
 Dim filename As String =
     HttpContext.Current.Server.MapPath("Checklist_Templates/" &
     selected_template_file & ".xml")
 Dim downloadFile = New System.IO.FileInfo(filename)
 HttpContext.Current.Response.Clear()
 HttpContext.Current.Response.AddHeader("Content-Disposition","attachment;
  filename=" + downloadFile.Name)
 HttpContext.Current.Response.AddHeader("Content-Length",
  downloadFile.Length.ToString())
 HttpContext.Current.Response.ContentType = "application/xml"
 HttpContext.Current.Response.WriteFile(downloadFile.FullName)
 HttpContext.Current.Response.End()
End Sub
```

Before the XML file is streamed to the user's browser, the corresponding file name and extension are defined. In this case, the file name is equal to the title of the experiment and the extension is certainly XML. According to the browser's settings, the file is downloaded to the default download folder of the user's computer.

Upon editing the template, the administrator most likely wants to upload the XML file to make it available for all users. ASP.NET offers a file upload class which displays a text box control and a browse button that enables the user to select a file to upload to the server[1].

```
Protected Sub Button_upload_click(ByVal sender As Object, ByVal e As
    System.EventArgs)
 If FileUpload_XML.HasFile Then
  FileUpload_XML.SaveAs("Destination Folder" &
      FileUpload_XML.FileName)
  Successmessage.Text = "File upload successful."
 Else
  Errormessage.Text = "You have not selected a file."
  Errormessage.Visible = True
 End If
End Sub
```

[1][Mic, FileUpload Class]

Needless to say, the page informs the user whether or not the upload was successful. A regular expression validator ensures that only XML files are uploaded. If a user tries to upload a file of a different type, an error message is displayed. Also if the user fails to select a file and hits the upload button, an error message pops up.

```
<asp:label ID="label_FileUpload" runat="server">
       Upload a new checklist template:
</asp:label>
<asp:FileUpload ID="FileUpload_XML" runat="server" />
    <asp:RegularExpressionValidator ID="regexValidator"
    runat="server" ForeColor="red"
        ControlToValidate="FileUpload_XML"
        ErrorMessage="Only xml Files are allowed."
        ValidationExpression="(.*\.([Xx][Mm][Ll])$)">
            </asp:RegularExpressionValidator>
<asp:Label id="Errormessage" runat="server"
    ForeColor="red"></asp:Label>
<asp:Label ID="Successmessage" runat="server"
    ForeColor="Green"></asp:Label>
```

4.4 Execution of Checklists

A checklist always comprises three parts, no matter the type of experiment. Each part is being read at a specific time. This chapter describes all three parts in chronological order, starting with the pick-up list.

4.4.1 Part 1 (Pick-Up Checklist)

Most experiments are performed in special rooms, which need to be prepared before the subject arrives. When all personnel and equipment is ready, the subject is picked up and accompanied to the experiment. This action is owed to the often tight schedule of the subjects in complex studies such as RSL. To minimize waiting time, it is crucial that the subject is ready when he is being picked up. In order to ensure that, the first part of the checklist is read while the subject is still inside the living module. Figure 4.5 demonstrates what a Muscle Biopsy checklist looks like.

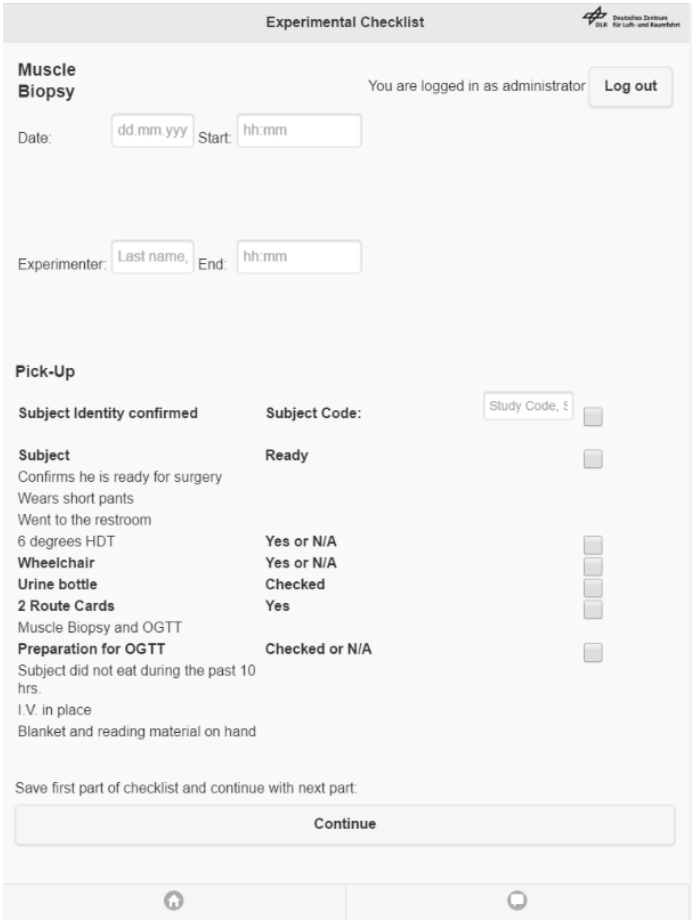


Figure 4.5: Checklist Part 1

The experimenter enters the current date, scheduled start and end times as well as his name into the text boxes before he starts to read the actual checklist. Please note that this is only the case for muscle biopsies and spiro-ergometry, where there is solely one experimenter in charge. During tilt table experiments, there is one scientific supervisor and one medical monitor who carry out the experiment together. For this reason, it is determined whether or not the text box for a medical monitor needs to be displayed.

```
If list_of_templates.FirstOrDefault = "Tilttable" Then
       Label_Medical_Monitor.Visible = True
       TextBox_Medical_Monitor.Visible = True
Else
    Label_Medical_Monitor.Visible = False
    TextBox_Medical_Monitor.Visible = False
End If
```

The corresponding text box can be seen in Figure 4.6.

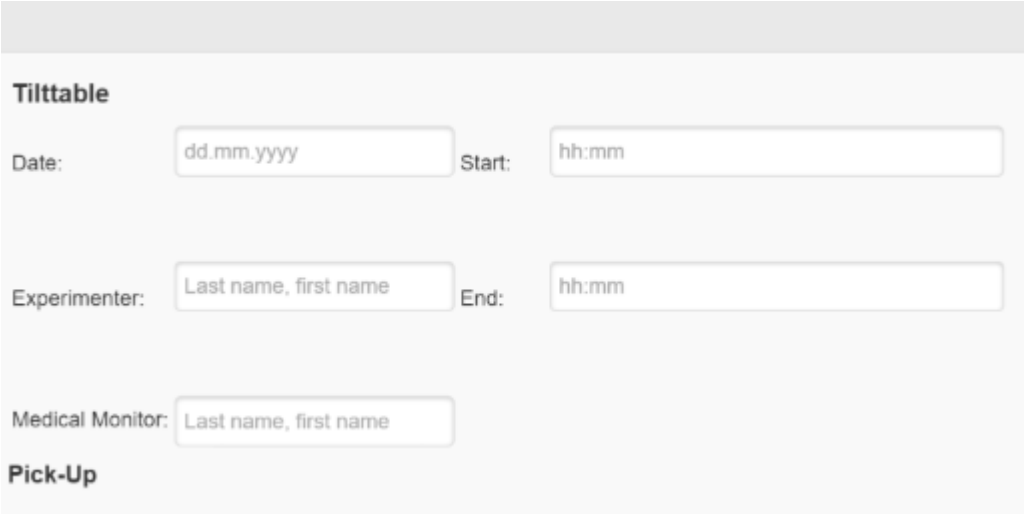


Figure 4.6: Medical Monitor during Tilt Table Experiments

Without the information from the four or five (depending on the experiment) text boxes it would be hard to identify an experiment from a list of conducted checklists. Thus, they are mandatory. When the site is validated - usually when the user tries to complete the current part of the checklist - the required fields are scanned to see if they are filled in.

```
<asp:TextBox ID="Textbox_Starttime" runat="server"
    Placeholder="hh:mm"></asp:TextBox>
<asp:RequiredFieldValidator
        id="Requiredfieldvalidator_Starttime" runat="server"
            EnableClientScript="false"
        ErrorMessage="Please enter a start time." Display="Static"
            ForeColor="Red"
        ControlToValidate="Textbox_Starttime">
</asp:RequiredFieldValidator>
```

The *Required field validator* only checks if there is any text in the box. In order to achieve a uniform documentation style, it is necessary that all users enter a time or date in the same format. That is a job for the *Regular expression validator* shown below:

```
<asp:RegularExpressionValidator
        ID="RegularExpressionValidator_Starttime" runat="server"
        ErrorMessage="Please insert the start time in the given format
            (24 hours!)"
        ControltoValidate="Textbox_Starttime" ForeColor="Red"
        Validationexpression="^[0-2][0-9]:[0-5][0-9]$">
</asp:RegularExpressionValidator>
```

If the user wants to enter twelve o'clock for instance and fails to enter the minutes, an error message will remind him. The regular expression for validating a time is rather short. The most complex regular expression used on this site validates the date entered by the user. It even takes leap years into account and verifies if the date is plausible.

Leap years can be identified using the following three criteria[2]:

1. The year can be evenly divided by 4.
2. The year is not a leap year, if it can be evenly divided by 100, unless
3. the year is also evenly divisible by 400.

There is a number of free online tools to generate and test regular expressions, which is very useful when it gets more complex. The website `www.debuggex.com` can be used to visualize self-created regular expressions as can be seen in Figure 4.7 on the following page.

[2][TA, Time and Date]

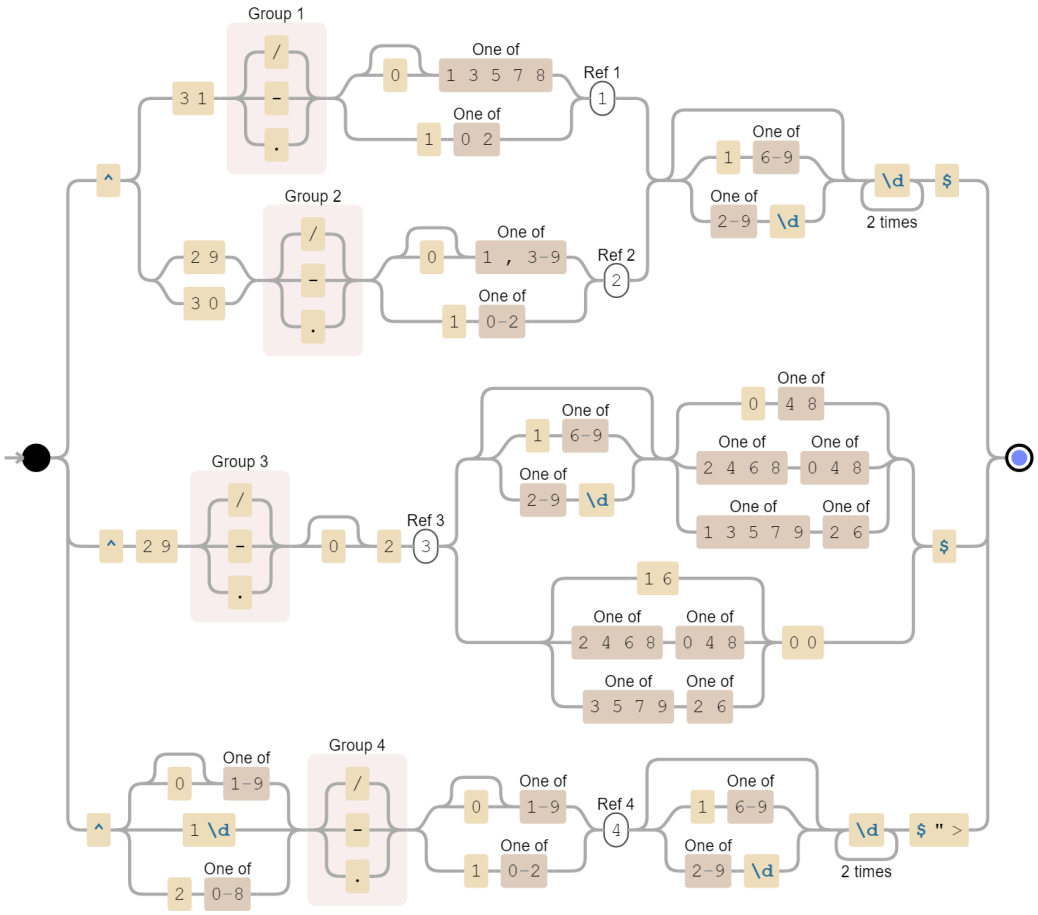


Figure 4.7: Regular Expression Validator for Dates

The static ASPX page contains an empty table, which will be filled by the code-behind procedures:

```
<asp:table id="Table_Content_Part1" runat="server" Width="90%" />
```

In order to be able to display the content of the checklist, the page load procedure needs to open the correct XML file. As described in Chapter 4.3, the unique ID of the template chosen by the user is stored in a cookie. If the cookie did not exist, the page would redirect the user to the default page. The ID is needed for two things. First, to find the right entry in the *list_of_templates.xml* file and to display the associated title of the checklist. Second, to open the correct XML file, that contains the actual content.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles Me.Load
 If (Context.User.Identity.IsAuthenticated) Then
  Session.Timeout = "300"
  If (Request.Cookies("selected_template") Is Nothing) Then
   Response.Redirect("Default.aspx")
  End If
  Dim selected_template As String
  selected_template =
      Server.HtmlEncode(Request.Cookies("selected_template").Value)
  Dim list_of_templates_file =
      XElement.Load(MapPath("~/List_of_templates.xml"))
  Dim list_of_templates =
   From template In list_of_templates_file.<template>
   Where template.<GUID>.Value = selected_template
   Select
   title = template.<title>.Value

   Label_Title.Text = list_of_templates.FirstOrDefault

   Dim template_file = XElement.Load(MapPath("~/Checklist_Templates/"
   &selected_template & ".xml"))
...
```

All three checklist templates are stored in a separate folder *Checklist_ Templates* in order to improve and simplify the file structure. The XML file for a muscle biopsy is shown below by way of example:

```
<?xml version="1.0" encoding="utf-8"?>
       <dataroot>
              <header>
                     <title>Muscle Biopsy</title>
                     <experimenter></experimenter>
                     <date></date>
                     <starttime></starttime>
                     <endtime></endtime>
              </header>
       <Part1>
              <title>Pick-Up</title>
              <Row>
              ...
```

The first piece of information extracted from the XML file is the title for the first part of the checklist, in this case *Pick-Up*. It is displayed on the ASPX page just below the header.

```
Dim template_content =
       From template In template_file.<Part1>
    Select
    title_Part1 = template.<title>.Value

Label_Title_Part1.Text = template_content.FirstOrDefault
```

The next step is to read the content from the XML file row by row. Each row's content is defined by up to six parameters:

<Question></Question> A checklist item consists of a question and a corresponding answer. The text of the question is displayed on the left side, whereas the answer stands on the right side.

<Bold></Bold> However, not all rows contain actual checklist items. Some rows only consist of sub items that help clarify the item above. For better recognition, items are printed in bold and sub items in normal font. This element can either be true or false.

<Answer></Answer> The answer usually consists of a phrase such as *checked* or *ready.* Answers are always printed in bold, since they only exist alongside items.

<Textbox></Textbox> Sometimes, standard phrases are not functional as answers. If an items queries a specific number like a date or a weight, the user must be able to enter the number into a text box. When the element is set to true, a text box is displayed next to the answer text.

<Placeholder></Placeholder> To ensure that the user enters the right information into the text box, a placeholder shows the demanded format or unit. Needless to say, this feature is only needed when the row contains a text box.

<No_Content></No_Content> If this element is set to true, the right side of the row does not contain any content. The last three rows in Figure 4.5 are an example for this.

To start displaying the content for each row, all *row* elements for the first part of the checklist need to be excerpted from the XML file:

```
Dim rows_content =
        From row In template_file.<Part1>.<Row>
    Select
    row_number = row.<Number>.Value,
    question_text = row.<Question>.Value,
    bold_check = row.<Bold>.Value,
    answer_text = row.<Answer>.Value,
    textbox_check = row.<Textbox>.Value,
    placeholder_text = row.<Placeholder>.Value,
    no_content_check = row.<No_Content>.Value
    Order By row_number
```

The content is going to be displayed in a table. The cells in that table are created dynamically. In order to obtain the number of rows needed in that table, the *row* elements in the XML file are counted.

Please note that the number of elements is subtracted by one, because the first row in the table will be assigned the number *0*, not *1*.

```
Dim number_of_rows_in_xml =
        Aggregate row In template_file.<Part1>.<Row>
    Into Count()

Dim number_of_rows_in_table = number_of_rows_in_xml - 1
Session("number_of_rows") = number_of_rows_in_table
```

Before the cells for the table are created, the corresponding web controls need to be defined. Each text box receives an assigned required field validator to ensure that the user enters something into the text box.

```
Dim Label1 As System.Web.UI.WebControls.Label
Dim Label2 As System.Web.UI.WebControls.Label
Dim Textbox1 As System.Web.UI.WebControls.TextBox
Dim RequiredFieldValidator1 As
    System.Web.UI.WebControls.RequiredFieldValidator
Dim Checkbox1 As System.Web.UI.WebControls.CheckBox
```

Each row comprises of four cells: two for the question and answer text as well as one for an optional text box and one for a check box. Each cell is created dynamically including the corresponding controls before it is being added to the temporary table. The listing below shows the process for a label containing a question's text.

```
1  Dim row_num As Integer
2  For row_num = 0 To number_of_rows_in_table
3      Dim tempRow As New TableRow()
4      Label1 = New Label()
5      Label1.ID = "Label_Question" & row_num.ToString
6      Label1.Text =
           rows_content.ElementAt(row_num.ToString).question_text
7      If rows_content.ElementAt(row_num.ToString).bold_check.ToString =
           "true" Then
8       Label1.Font.Bold = True
9      End If
10     Dim tempCell_1 As New TableCell()
11     tempCell_1.Controls.Add(Label1)
12     tempRow.Cells.Add(tempCell_1)
13     tempCell_1.Width = Unit.Percentage(40)
14     ...
15     Table_Content_Part1.Rows.Add(tempRow)
16    Next
```

First, a new label is defined. Please note that the name *Label1* is never changed throughout the for loop. *Label1* always defines the question's text for each row in the table. The actual ID of each individual label is defined in the next step: it is named *Label_ Question* plus the row number, starting with *0*. The text is taken from the associated *Row* entry in the XML file.

After determining if the text is to be printed in bold, the label is added to the temporary table cell. Continuingly, more temporary cells containing web controls are added to the temporary row, before the entire row is added to the table until the end of the loop.

With all items being displayed, the user can start to use the checklist. When he has checked all items of the first part of the checklist, he most likely wants to save this part and continue with the next one. So far, the application has only read static content from XML files. The information entered by the user is saved only then when the user hits the *Continue* button at the bottom of the checklist.

```
Protected Sub Button_save_click(ByVal sender As Object, ByVal e As
    System.EventArgs)
        If Page.IsValid Then
        Dim current_user = Context.User.Identity.Name.ToString()
        Dim title = Label_Title.Text
        Dim title_part_1 = Label_Title_Part1.Text
        Dim entered_date = TextBox_Date.Text
        Dim entered_starttime = Textbox_Starttime.Text
        Dim entered_endtime = Textbox_Endtime.Text
        Dim entered_medical_monitor = ""
        If Label_Title.Text = "Tilttable" Then
                Try
                entered_medical_monitor = " with " &
                    TextBox_Medical_Monitor.Text
                Catch ex As Exception
                End Try
        End If
        Dim entered_experimenter = TextBox_Experimenter.Text &
            entered_medical_monitor.ToString
```

After verifying that the page is valid, all entries from the header are stored in variables. In case the current experiment is a tilt table, the medical monitor's name is added to the *entered_ experimenter* definition.

Up to this point, only the template ID was of interest, but since the first part of the checklist has already been conducted, a new GUID for this specific checklist has to be created. This unique ID is then saved in a new cookie named *checklist_ id*, so the following websites can access the information as well.

```
Dim selected_template As String
selected_template =
    Server.HtmlEncode(Request.Cookies("selected_template").Value)
Dim new_checklist_id = System.Guid.NewGuid.ToString()
Dim new_cookie As New HttpCookie("checklist_id")
new_cookie.Value = new_checklist_id
new_cookie.Expires = DateTime.Now.AddHours(12)
Response.Cookies.Add(new_cookie)
```

The information entered by the user will be saved in a new XML file. To make this file easily accessible, there needs to be a list containing every single checklist that was ever conducted with this application. This file is called *List_ of_ conducted_ checklists.xml*.

```
Dim checklist_file = MapPath("~/List_of_conducted_checklists.xml")
Dim checklist_xml = XElement.Load(checklist_file)
Dim new_checklist_xml_entry As XElement =
        <checklist>
                <GUID_of_Checklist>
                        <%= new_checklist_id %>
                </GUID_of_Checklist>
                <GUID_of_Template>
                        <%= selected_template %>
                </GUID_of_Template>
                <user>
                        <%= current_user %>
                </user>
                <date>
                        <%= entered_date %>
                </date>
                <experimenter>
                        <%= entered_experimenter %>
                </experimenter>
                <starttime>
                        <%= entered_starttime %>
                </starttime>
                <endtime>
                        <%= entered_endtime %>
                </endtime>
        </checklist>

checklist_xml.Add(new_checklist_xml_entry)
checklist_xml.Save(checklist_file)
```

The element holds all the information needed to identify a particular checklist. This makes data analysis easy, even with a huge number of conducted experiments. The administrator can filter search results by experimenters, date, type of experiment or time.

A finished XML entry looks like this:

```
<checklist>
	<GUID_of_Checklist>
		64c2057d-d42f-46ba-83e9-e82600e0ef32
	</GUID_of_Checklist>
	<GUID_of_Template>
		DC18C032-DB3D-4D9E-B0F7-E8CF9FF2B665
	</GUID_of_Template>
	<user>administrator</user>
	<date>01.08.2016</date>
	<experimenter>Fuhrmann, Malte</experimenter>
	<starttime>10:00</starttime>
	<endtime>11:30</endtime>
</checklist>
```

One piece of information not added to the entry in the list of conducted checklists is the subject code. It is only stored in the XML file holding the actual answers and comments from the checklist, given by the user. This file does not exist yet and needs to be created at this point.

```
Dim entered_subject As String = ""
Try
Dim subject_textbox As TextBox = form1.FindControl("Textbox_Answer0")
entered_subject = subject_textbox.Text
Catch ex As Exception
End Try
Dim new_checklist_xml_file As XElement =
<dataroot>
	<header>
		<subject>
			<%= entered_subject %>
		</subject>
		...
	</header>
	<title_part_1>
		<%= title_part_1 %>
	</title_part_1>
</dataroot>
Dim new_checklist_file = MapPath("~/Conducted_Checklists/" &
    new_checklist_id & ".xml")
new_checklist_xml_file.Save(new_checklist_file)
```

For reasons of simplicity, not all the information stored in the header is repeated here. It is congruent with the *new_checklist_xml_entry* except for the subject code mentioned above.

Analogous to creating the table to display the checklist, it is being read cell by cell to save the contained information. The state of a control, for instance a check box, can easily be detected, if it has been created by hand in the process of building a website. Dynamically created controls are a bit tougher to read. The key to detecting the state of a control, which has been created programatically is knowing its ID. Since all controls that have been added to the table have been assigned a unique name, they can be easily distinguished from one another.

```
1 Dim number_of_rows_in_table = Table_Content_Part1.Rows.Count - 1
2 Dim number_of_rows_to_save As Integer
3
4 For number_of_rows_to_save = 0 To number_of_rows_in_table
5         Dim text_question As Label =
              form1.FindControl("Label_Question" &
              number_of_rows_to_save)
6         Dim question_bold As String = "false"
7         If text_question.Font.Bold = True Then
8                 question_bold = "true"
9         End If
```

For each row of the table, starting with row number *0*, the state of the controls is saved. The first information excerpted is the text of the question label. Using the command *FindControl* and the assigned name a particular label is addressed. Since a XML file does not differentiate between normal font and bold print, this information has to be stored in a separate variable.

The associated answer to the question is saved in the same way followed by the text box query. Most items on a checklist do not need a text box. Therefore, the default value is *No Textbox*. If a text box exists, the value is replaced with the text inside the text box.

```
1 Dim text_answer_text As String = ""
2 Try
3         Dim text_answer As Label = form1.FindControl("Label_Answer" &
              number_of_rows_to_save)
4         text_answer_text = text_answer.Text
5 Catch ex As Exception
6 End Try
7 Dim text_textbox As String = "No Textbox"
8 Try
9         Dim textbox As TextBox = form1.FindControl("Textbox_Answer" &
              number_of_rows_to_save)
10        text_textbox = textbox.Text
11 Catch ex As Exception
12 End Try
```

It is similar with the check box, which is queried last. If a check box exists, it is saved whether the box is checked or not:

```
Dim checkbox_checked As String = "No Checkbox"
Try
        Dim checkbox As CheckBox = form1.FindControl("Checkbox" &
            number_of_rows_to_save)
        If checkbox.Checked.ToString = "true" Then
                checkbox_checked = "Checked"
        ElseIf checkbox.Checked.ToString = "False" Then
                checkbox_checked = "Not Checked"
        End If
Catch ex As Exception
End Try
```

After all five pieces of information are extracted from the first row of the table, they need to be saved in the XML file created for this particular checklist. If the data was not saved at this point, the variables would be overwritten by the loop. All data would be lost except for the last row. To avoid this, a new XML entry is created and saved for each row, before the loop starts over.

```
Dim row_xml_entry As XElement =
        <row_part_1>
                <number>
                        <%= number_of_rows_to_save %>
                </number>
                <question>
                        <%= text_question.Text %>
                </question>
                <question_bold>
                        <%= question_bold %>
                </question_bold>
                <text_answer>
                        <%= text_answer_text %>
                </text_answer>
                <textbox>
                        <%= text_textbox %>
                </textbox>
                <checkbox>
                        <%= checkbox_checked %>
                </checkbox>
        </row_part_1>
        conducted_checklist_xml.Add(row_xml_entry)
        conducted_checklist_xml.Save(new_checklist_file)
Next
```

After looping through all rows of the table, the XML file contains all the relevant information entered by the user. A new cookie named *part1_completed* is created and the user is redirected to the second part of the checklist. The cookie serves as a safety net in case the user closes the browser or turns off the device. Because of the cookie, the browser will even then remember that the first part of the checklist has already been finished and redirects him straight to the second part.

```
1 Dim new_completed_cookie As New HttpCookie("part1_completed")
2 new_completed_cookie.Value = "true"
3 new_completed_cookie.Expires = DateTime.Now.AddHours(12)
4 Response.Cookies.Add(new_completed_cookie)
5 Response.Redirect("Conduct_checklists_part2.aspx")
```

4.4.2 Part 2 (Pre-start Checklist)

The second part of the checklist is a continuation of the part described in the last chapter. The layout is very similar, only the header containing information about the date and time is missing. Figure 4.8 shows the second part of the muscle biopsy checklist exemplarily.

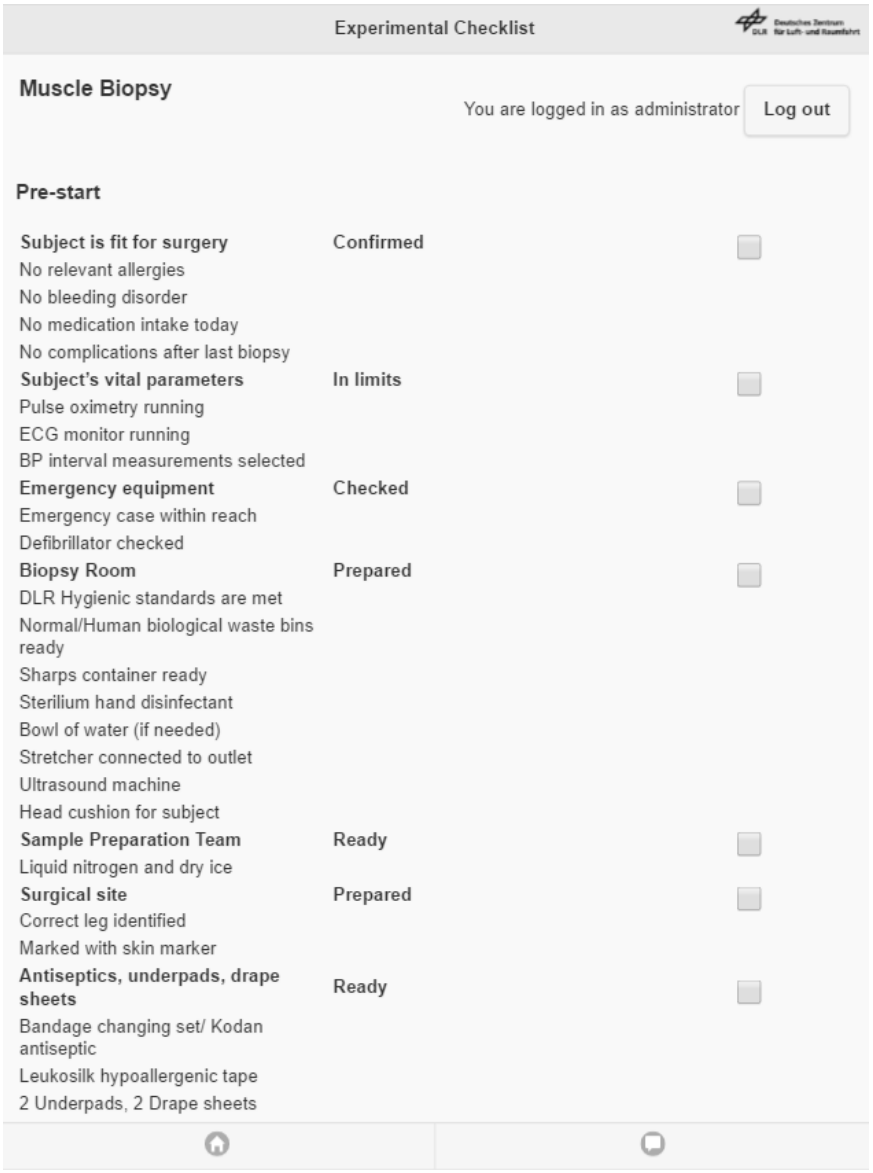


Figure 4.8: Checklist Part 2

When the user is redirected to the *Conduct_ checklists_ part2* page, information about necessary cookies needs to be obtained.

```
If (Request.Cookies("selected_template") Is Nothing) Then
        Response.Redirect("Default.aspx")
End If
If (Request.Cookies("checklist_id") Is Nothing) Then
        Response.Redirect("Conduct_checklists.aspx")
End If
If (Request.Cookies("part1_completed") Is Nothing) Then
        Response.Redirect("Conduct_checklists.aspx")
End If
If Not (Request.Cookies("part2_completed") Is Nothing) Then
        Response.Redirect("Conduct_checklists_part3.aspx")
End If
```

The server redirects the user to predefined pages if one of the essential pieces of information is missing. The same procedure applies when the user requests a part of a checklist which has already been completed. The transferals are very intuitive, even though these situations will rarely arise.

After having assured that all cookies are accessible for displaying the checklist, the selected template needs to be loaded. Of importance thereby is the type of experiment which is going to be performed. Tilt table and spiroergometry both have a long list of termination criteria. To help the experimenters and medical monitors refresh their memories, a PDF file containing the list can be accessed. The feature has to be disabled for muscle biopsy lists:

```
Dim selected_template As String
selected_template =
    Server.HtmlEncode(Request.Cookies("selected_template").Value)
If list_of_templates.FirstOrDefault = "VO2max" Then
        Button_newtab.Visible = True
        Label_newtab.Visible = True
ElseIf list_of_templates.FirstOrDefault = "Tilttable" Then
        Button_newtab.Visible = True
        Label_newtab.Visible = True
Else
        Button_newtab.Visible = False
        Label_newtab.Visible = False
End If
```

The button that opens the PDF documents in a new tab can be seen in Figure 4.9 on the following page.

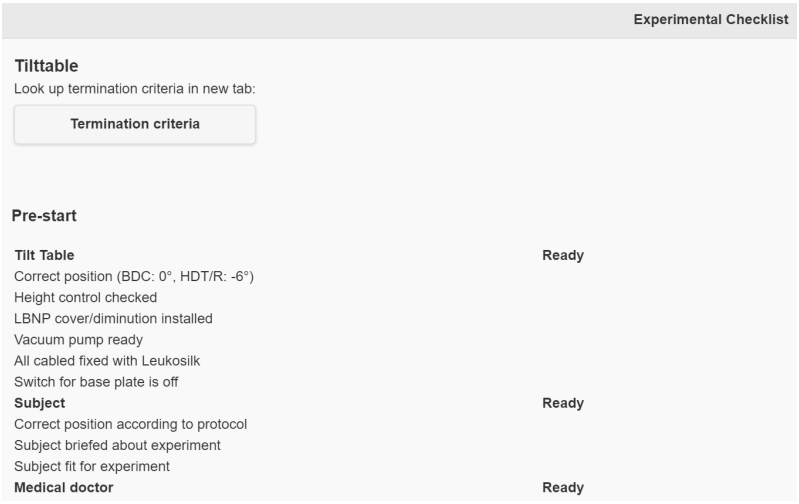


Figure 4.9: List of Termination Criteria

The PDF shall be accessible throughout the whole experiment without keeping the user from using the checklist. Therefore, a new tab is opened in the user's browser when the button is clicked:

```
Protected Sub Button_newtab_Click(ByVal sender As Object, ByVal e As
    System.EventArgs)
        Session("kind_of_checklist") = Label_Title.Text
        Dim url As String = "Display_files.aspx"
        Dim sb As New StringBuilder()
        sb.Append("<script type = 'text/javascript'>")
        sb.Append("window.open('")
        sb.Append(url)
        sb.Append("');")
        sb.Append("</script>")
        ClientScript.RegisterStartupScript(Me.GetType(),
        "script", sb.ToString())
End Sub
```

The procedure opens an ASPX page named *Display_files* which does not have any content. Solely the code-behind file contains the following commands to display the PDF file assigned to the current experiment:

```
Dim kind_of_checklist = Session("kind_of_checklist")
Response.ContentType = "Application/pdf"
Dim FilePath As String = MapPath("~/Termination_Criteria/" &
    kind_of_checklist.ToString & ".pdf")
Response.WriteFile(FilePath)
Response.End()
```

One of the major advantages of using this application is the documentation of all experiments that are performed. Especially for muscle biopsies it can be useful to save pictures of the surgical sites or the biopsy material. At the end of the checklist items, the user has the possibility to upload an unlimited quantity of pictures.

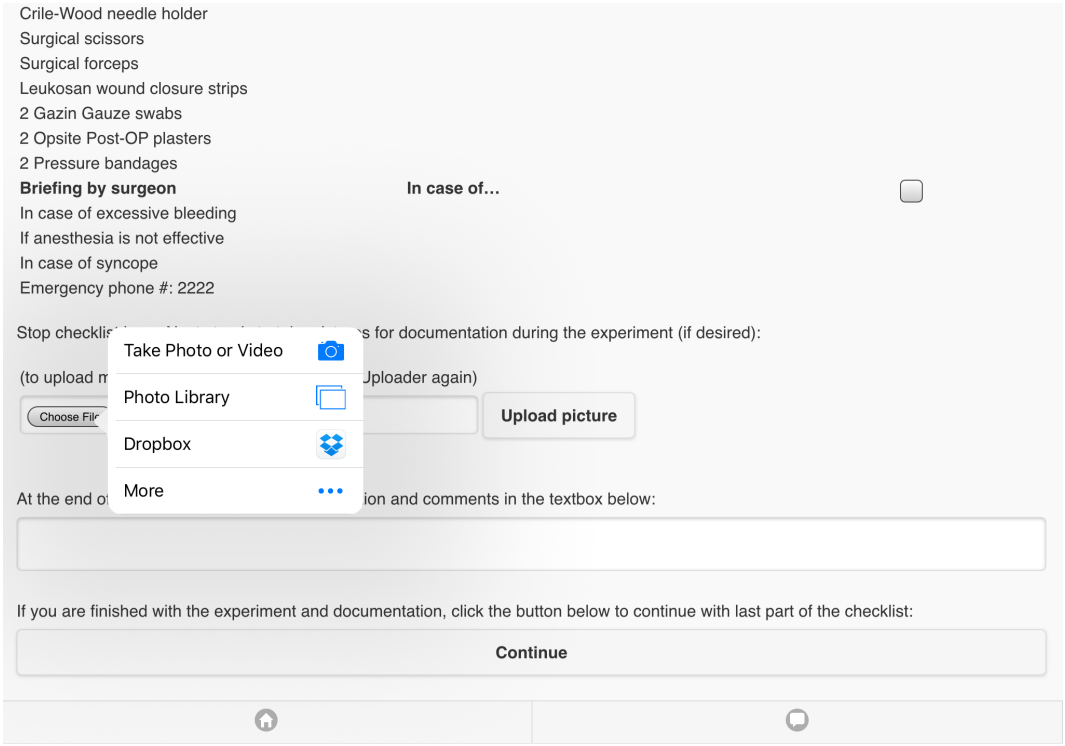


Figure 4.10: Picture Upload

As one can see in Figure 4.10, pictures can be uploaded from a variety of sources such as the tablet's camera, photo library or from a Dropbox folder. It would be negligent not to restrict the possible file types to pictures or to be more precise JPEG's. Otherwise, crafty users could upload source code files that would alter the application's behavior. The Regular Expression Validator used for that reviews the file extension and notifies the user where appropriate.

```
<asp:FileUpload ID="FileUpload_Pictures" runat="server" />
<asp:RegularExpressionValidator
        ID="regexValidator"
        runat="server" ForeColor="red"
        ControlToValidate="FileUpload_Pictures"
        ErrorMessage="Only jpeg Files are allowed."
        ValidationExpression="(.*\.([Jj][Pp][Gg])$)"/>
<asp:Label ID="Errormessage" runat="server" ForeColor="red"/>
<asp:Label ID="Successmessage" runat="server" ForeColor="Green"/>
```

The ASP FileUpload control makes accepting uploads from users very easy. All that is left to do is choosing a name and a destination folder for the file. The picture to be saved already has a file name, which was assigned by the device it was taken with. Using this name could lead to duplicate files and is therefore not practicable. A new unique ID is created and the file is named accordingly.

To associate the picture with the current checklist, a new entry containing the file name and the time of the upload is added to the XML file:

```
Protected Sub Button_upload_click(ByVal sender As Object, ByVal e As
    System.EventArgs)
        Dim checklist_id As String
        checklist_id =
            Server.HtmlEncode(Request.Cookies("checklist_id").Value)
        Dim new_upload_id = System.Guid.NewGuid.ToString()
        If FileUpload_Pictures.HasFile Then
                FileUpload_Pictures.SaveAs("Destination Folder" &
                        new_upload_id & ".jpeg")
                Successmessage.Text = "File uploaded successfully."
                Dim time_saved =
                    DateTime.Now.ToString("dd-MM-yy_HH-mm")
                Dim checklist_file = MapPath("~/Conducted_Checklists/"
                    & checklist_id & ".xml")
                Dim checklist_xml = XElement.Load(checklist_file)
                Dim new_checklist_xml_entry As XElement =
                        <picture>
                                <time>
                                        <%= time_saved %>
                                </time>
                                <GUID>
                                        <%= new_upload_id %>
                                </GUID>
                        </picture>

                checklist_xml.Add(new_checklist_xml_entry)
                checklist_xml.Save(checklist_file)
        Else
                Errormessage.Text = "You have not selected a file."
        End If
End Sub
```

In the rare event the user clicks the upload button without having specified a file before, an error message is displayed. The user can then select a picture and try to upload it again. The error message disappears and instead, a success report is being displayed.

The second part of the checklist is most likely to be completed by the experimenter in charge of the experiment. The next and last part is not necessarily performed by the same person, but rather by someone who drops off the subject. The experimenter can give special instructions by writing a comment in a text box at the end of the checklist. The text is then displayed on the next page, so it is not forgotten during drop off.

```
Protected Sub Button_save_click(ByVal sender As Object, ByVal e As
    System.EventArgs)
        If Page.IsValid Then
                        Dim comment = Textbox_comments.Text.ToString
                        comment = comment.Replace("&", "")
                        comment = comment.Replace("<", "")
                        comment = comment.Replace(">", "")
                        comment = comment.Replace("*", "")

Dim new_checklist_xml_entry_comment As XElement =
        <comment>
                <time>
                        <%= time_saved %>
                </time>
                <text>
                        <%= comment %>
                </text>
        </comment>

checklist_xml.Add(new_checklist_xml_entry_comment)
checklist_xml.Save(checklist_file)
```

The comment is added to the XML file in a similar way as the pictures. As a security precaution, all special characters, that could be misinterpreted as commands by the XML file, are disposed of before saving.

The rest of the checklist is processed as described for the first part in Chapter 4.4.1.

4.4.3 Part 3 (Drop-off Checklist)

The third and last part of the checklist is conducted after the actual experiment is over. It is intended to brief the subject about the dos and don'ts, when he is dropped off at the living module. Figure 4.11 shows the final part of a muscle biopsy checklist. As one can see, comments from the experimenter are displayed at the top of the page. The person who reads the list can write down comments for documentation as well.

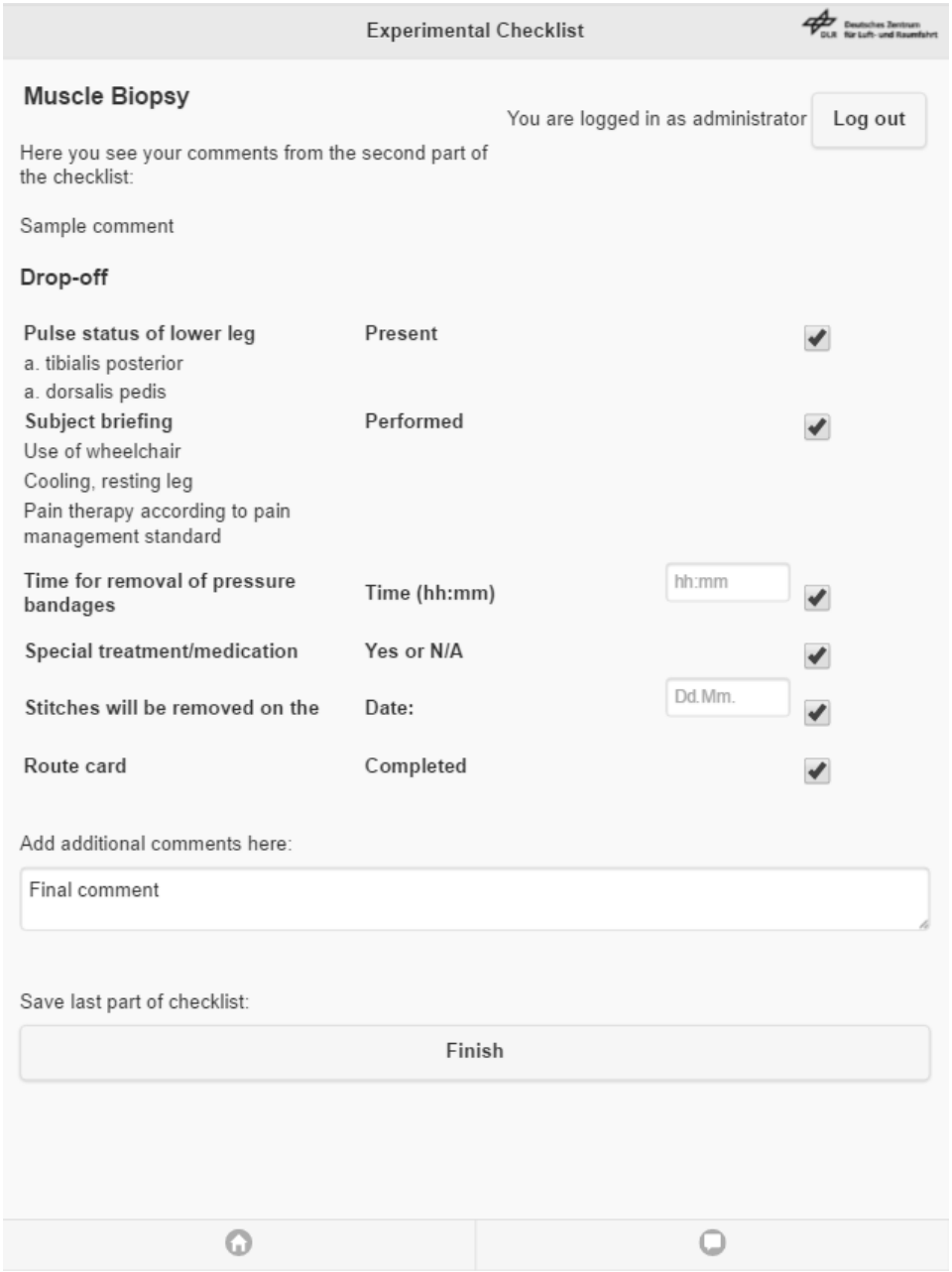


Figure 4.11: Checklist Part 3

The content of the list is displayed and saved in exactly the same way as the two parts before, therefore it is not discussed in detail here again. However, the page has more to offer than just saving the items into an XML file. When the user clicks the *Finish* button, the server automatically generates an email containing all three parts of the checklist, the comments by the user and all pictures that where uploaded. This email can then be printed out and added to the subject's record. Please refer to Attachment D on page M for a full example of a server generated email.

The simplest way of transferring information about the experiment would be to send the corresponding XML file per email. Due to its length and structure, the XML file is confusingly complex though. Therefore, it is more suitable to prepare the data and display it in plain text. The best way to achieve congruent layouts across different checklists is an email template.

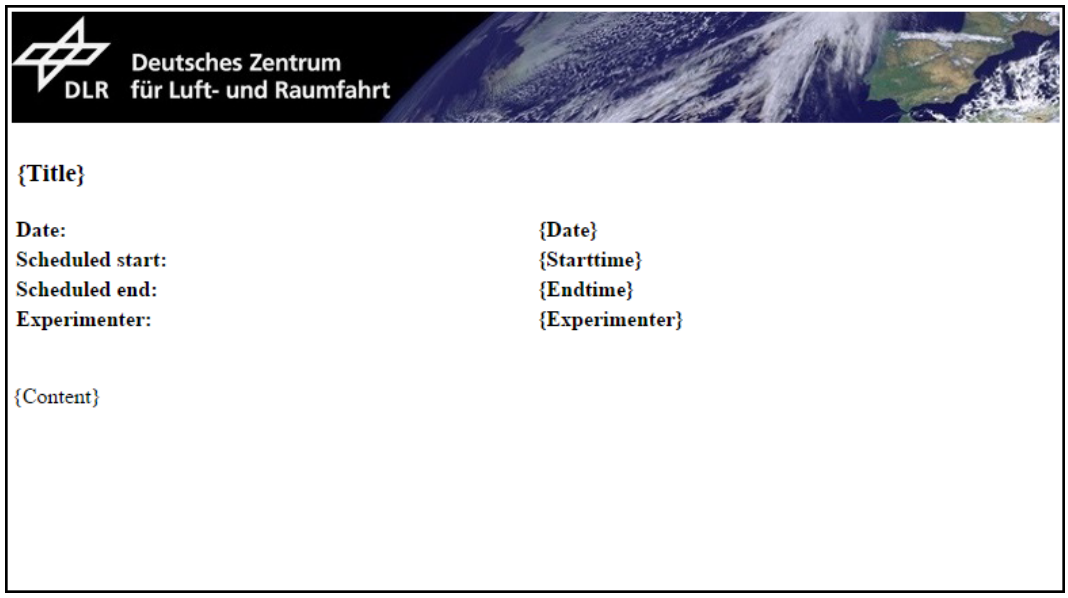


Figure 4.12: Email Template

The template is nothing but a simple HTML page, which contains placeholders. Those are written in curly brackets and later replaced programatically. The header consists of an image file stored on the server:

```
<div data-role="page">
 <img src="http://www.fbm.hs-bremen.de/.../email-header.jpg" />
 <div data-role="content">
       <table id="Table_Header" style="width:100%">
              <tbody>
                     <tr><td colspan="2"><h3>{Title}</h3></td></tr>
                     ...
```

A sub procedure named *Send_ Email* is used to create a new mail message. After defining the sender address, the necessary data to substitute the placeholders has to be retrieved from the XML file:

```
Protected Sub Send_Email()
        Dim new_email As MailMessage = New MailMessage()
        new_email.From = New MailAddress("malte.fuhrmann@dlr.de")
        Dim checklist_id As String
        checklist_id =
            Server.HtmlEncode(Request.Cookies("checklist_id").Value)
        Dim checklist_file = MapPath("~/Conducted_Checklists/" &
            checklist_id & ".xml")
        Dim checklist_xml = XElement.Load(checklist_file)
        Dim header_content =
                From header In checklist_xml.<header>
                Select
                        entered_title = header.<title>.Value,
                        entered_subject = header.<subject>.Value,
                        ...
                        entered_endtime = header.<endtime>.Value
                Order By entered_title
Dim title_from_xml = header_content.FirstOrDefault.entered_title
Dim subject_from_xml = header_content.FirstOrDefault.entered_subject
...
Dim endtime_from_xml = header_content.FirstOrDefault.entered_endtime
```

Knowing the title of the experiment and the subject code, the message subject field can be created. The message's body starts off with an empty string, which is filled with the template HTML page. The placeholders of the header are then substituted with the values defined before.

```
new_email.Subject = "Documentation of a " & title_from_xml & " for
    subject " & subject_from_xml

Dim body As String = String.Empty
Dim reader As StreamReader = New
    StreamReader(Server.MapPath("~/EmailTemplate.html"))
body = reader.ReadToEnd
body = body.Replace("{Title}", title_from_xml & " for subject " &
    subject_from_xml)
body = body.Replace("{Date}", date_from_xml)
body = body.Replace("{Starttime}", starttime_from_xml)
body = body.Replace("{Endtime}", endtime_from_xml)
body = body.Replace("{Experimenter}", experimenter_from_xml)
```

Displaying the content of the checklist is more complex due to the amount of data. To achieve a smooth layout, a table is created which will hold all three parts of the checklist. The first row of the table only contains the title of the first part:

```
Dim new_string_builder As New StringBuilder("<table>")
Dim part1_title =
        From title In checklist_xml.<title_part_1>
        Select title.Value
new_string_builder.Append("<tr><td><b>" & part1_title.FirstOrDefault
    & "<br/></b></td></tr>")
```

So far, the table only consists of one row. In order to know, how many rows need to be added for the content, the corresponding *<row>* entries in the XML file need to be counted. This number is subtracted by one, because row numbering starts with *0*, as for the tables before.

```
Dim part1_content =
        From row In checklist_xml.<row_part_1>
        Select
                number = row.<number>.Value,
                question_text = row.<question>.Value,
                answer_text = row.<text_answer>.Value,
                textbox_text = row.<textbox>.Value,
                CheckBox = row.<checkbox>.Value
Dim number_of_rows_in_xml_part1 =
        Aggregate row In checklist_xml.<row_part_1>
Into Count()
Dim number_of_rows_in_table_part1 = number_of_rows_in_xml_part1 - 1
```

For each row, four table cells have to be created, containing the following information:

1. The question
2. The answer
3. Information, the user entered in a text box, where appropriate
4. The state of the corresponding check box, where appropriate

In case a row does not contain a text or a check box, the related cell remains empty.

```
Dim row_num As Integer
For row_num = 0 To number_of_rows_in_table_part1
 Dim cell1 = part1_content.ElementAt(row_num.ToString).question_text
 Dim cell2 = part1_content.ElementAt(row_num.ToString).answer_text
 Dim cell3 = part1_content.ElementAt(row_num.ToString).textbox_text
 Dim cell4 = part1_content.ElementAt(row_num.ToString).CheckBox
 If cell3 = "No Textbox" Then
        cell3 = ""
 End If
 If cell4 = "No Checkbox" Then
        cell4 = ""
 End If
```

With the help of a string builder, the four cells are assembled to a table row. The procedure is repeated for all rows in the three parts of the checklist. The body contains almost all relevant pieces of information about the experiment at this point. Before comments and pictures are added to the email, the table is finalized.

```
new_string_builder.Append("<tr><td>" & cell1 _
        & "    </td><td>" & cell2 _
        & "    </td><td>" & cell3 _
        & "    </td><td>" & cell4 & "   </td></tr>")
Next
new_string_builder.Append("</table><br/><br/>")
```

As described in Chapter 4.4.2 on page 46, it is mandatory for the experimenter to write a comment at the end of the second part of the checklist. An optional comment can be left at the end of the drop-off list. Both comments are summarized and added to the body of the message with line breaks in between. The *Try-Catch* Statement handles a possible error that may occur if no additional comment can be found.

```
Try
        Dim comment_in_xml =
        From comment In checklist_xml.<comment>
        Select
                comment_text = comment.<text>.Value

        Dim add_comment_in_xml =
        From add_comment In checklist_xml.<add_comment>
        Select
                add_comment_text = add_comment.<text>.Value

        new_string_builder.Append("Comments from Experimenter:<br/>")
        new_string_builder.Append(comment_in_xml.FirstOrDefault &
            "<br/>")
        new_string_builder.Append(add_comment_in_xml.FirstOrDefault &
            "<br/>")
Catch ex As Exception
End Try
body = body.Replace("{Content}", new_string_builder.ToString)
new_email.Body = body
new_email.IsBodyHtml = True
```

At this point, the placeholder *{Content}* can be substituted with the string builder containing the table and comments. For the commands defining the table to work, the body of the email has to be defined as HTML code. Otherwise, a confusing mixture of code and text would be displayed, which is not intended.

Finally, pictures are attached to the email, where appropriate:

```
Dim picture_id_from_xml =
        From picture In checklist_xml.<picture>
        Select
                picture_id = picture.<GUID>.Value
        Order By picture_id
For Each picture_id In picture_id_from_xml
        new_email.Attachments.Add(New Attachment(MapPath("~/Uploads/"
            & picture_id.ToString & ".jpeg")))
Next
new_email.To.Add(New MailAddress("sample@mail.com"))
new_email.Priority = MailPriority.High
```

With the use of a SMTP client, the email is sent to the responsible study coordinators. With a click of a button, the email is printed out and added to the subject record. Recipients and priority of the message are set accordingly.

The client uses an existing email account on a SMTP server, in this case a DLR mail server. For the site to be able to access the server, a username and password have to be provided. Also the connection needs to be secured by SSL (a requirement by the email program used).

```
Dim smtp As SmtpClient = New SmtpClient()
Try
        smtp.Host = "smtp.dlr.de"
        smtp.EnableSsl = True
        Dim network_credentials As NetworkCredential = New
            System.Net.NetworkCredential()
        network_credentials.UserName = "username"
        network_credentials.Password = "password"
        smtp.Credentials = network_credentials
        smtp.UseDefaultCredentials = True
        smtp.Port = 587
        smtp.Send(new_email)
Catch ex As Exception
Finally
        smtp = Nothing
        new_email = Nothing
End Try
End Sub
```

After the email has been transmitted, both variables defining the SMTP client as well as the email are cleared. This ensures that every time the procedure is called, a new email is generated and sent.

4.5 Feedback Form

During the development and testing phase, it is crucial to actively integrate the whole staff working with the application. It is important to know if errors occurred or what can be improved in order to increase the acceptance of the newly introduced checklists. A feedback questionnaire has been implemented which consists of eleven questions. Development and evaluation of the questionnaire are thoroughly discussed in Chapter 5 on page 71. This chapter covers the process of programming the website that displays the feedback form.

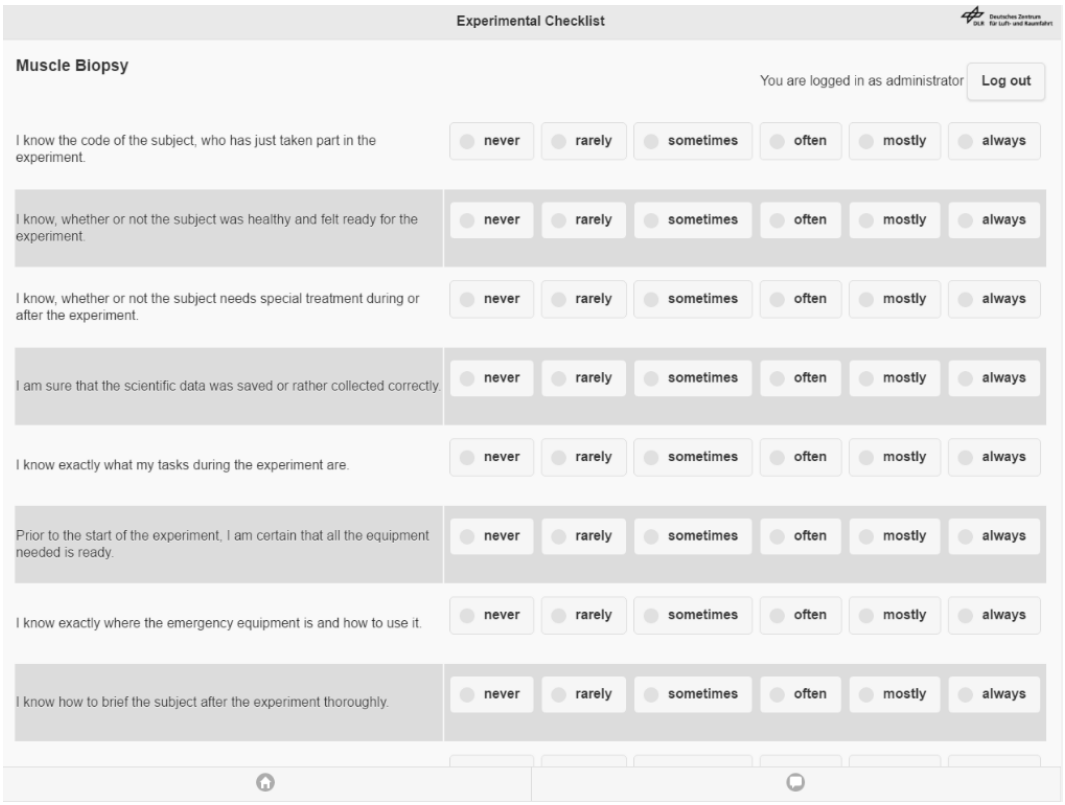


Figure 4.13: Feedback Form

The ASPX page hosting the questionnaire contains a table with eleven rows, each comprised of a question text and six answer options. A list of radio button covers a scale from *never* to *always*. Each list is controlled by a required field validator to ensure that a user answers every single question.

Below the eleven questions is a text box which enables the user to report any technical malfunction, if applicable. When the feedback is submitted, a sub procedure saves the user's selection into a XML file.

```
Protected Sub Button_submit_click(ByVal sender As Object, ByVal e As
    System.EventArgs)
        Dim selected_answer_1 = radiobuttonlist1.SelectedItem.Value
        Dim selected_answer_2 = radiobuttonlist2.SelectedItem.Value
        ...
        Dim selected_answer_11 = radiobuttonlist11.SelectedItem.Value
        Dim technical_problem =
            Textbox_technical_problems.Text.ToString
        Dim checklist_id As String
        checklist_id =
            Server.HtmlEncode(Request.Cookies("checklist_id").Value)
        Dim date_time = DateTime.Now.ToString("dd-MM-yy_HH-mm")
        Dim title = Label_Title.Text.Trim
```

The value of all lists of radio buttons is stored in variables, as well as the inputs from the text box. To help identify an experiment, the type of checklist used and the time saved are recorded, too. A new XElement is created which is saved in a separate folder named *FeedbackForms* on the server.

```
Dim new_feedback_xml_file As XElement =
 <dataroot>
        <title>
                <%= title %>
        </title>
        <date_time>
                <%= date_time %>
        </date_time>
        <checklist_id>
                <%= checklist_id %>
        </checklist_id>
        <question1>
                <%= selected_answer_1 %>
        </question1>
        ...
    <question11>
                <%= selected_answer_11 %>
        </question11>
        <technical_problems>
                <%= technical_problem %>
        </technical_problems>
 </dataroot>

Dim new_feedback_form_file = MapPath("~/FeedbackForms/" & title & "_"
    & date_time & ".xml")
new_feedback_xml_file.Save(new_feedback_form_file)
```

In addition to saving the feedback in a XML file, an email is sent out to the system administrator containing the same information. This has proven to be very useful during the test phase, because the reaction time to errors is very short. The email is dispatched in the same way as described in Chapter 4.4.3 on page 52 with one difference. Due to the shortness of the list, a template is not necessary.

```
1  Dim message_body As String
2  message_body = "Question 1: <b>" & selected_answer_1 & "</b><br/>" _
3  & "Question 2: <b>" & selected_answer_2 & "</b><br/>" _
4  & "Question 3: <b>" & selected_answer_3 & "</b><br/>" _
5  ...
6  & "Question 11: <b>" & selected_answer_11 & "</b><br/><br/><br/>" _
7  & "Comments: <br/>" & technical_problem
8
9  Dim new_email As MailMessage = New MailMessage()
10 ...
11 new_email.Attachments.Add(New Attachment(MapPath("~/FeedbackForms/" &
       title & "_" & date_time & ".xml")))
12 ...
```

4.6 Editing of Templates

The testing phase of the application has shown that every user has a different idea of how the application should behave. Involving colleagues into the development process is crucial in order to enhance acceptance and level of satisfaction. This results in many little changes that need to be made though in order meet everyone's liking. Speaking from experience, most changes are of contentual nature.

To modify a checklist's template, one has to download an XML file, edit it with Visual Studio, for instance, and upload it again. That may be convenient when changing large parts of a template, but certainly not when only fixing a typing error. This is where the template editing page comes in:

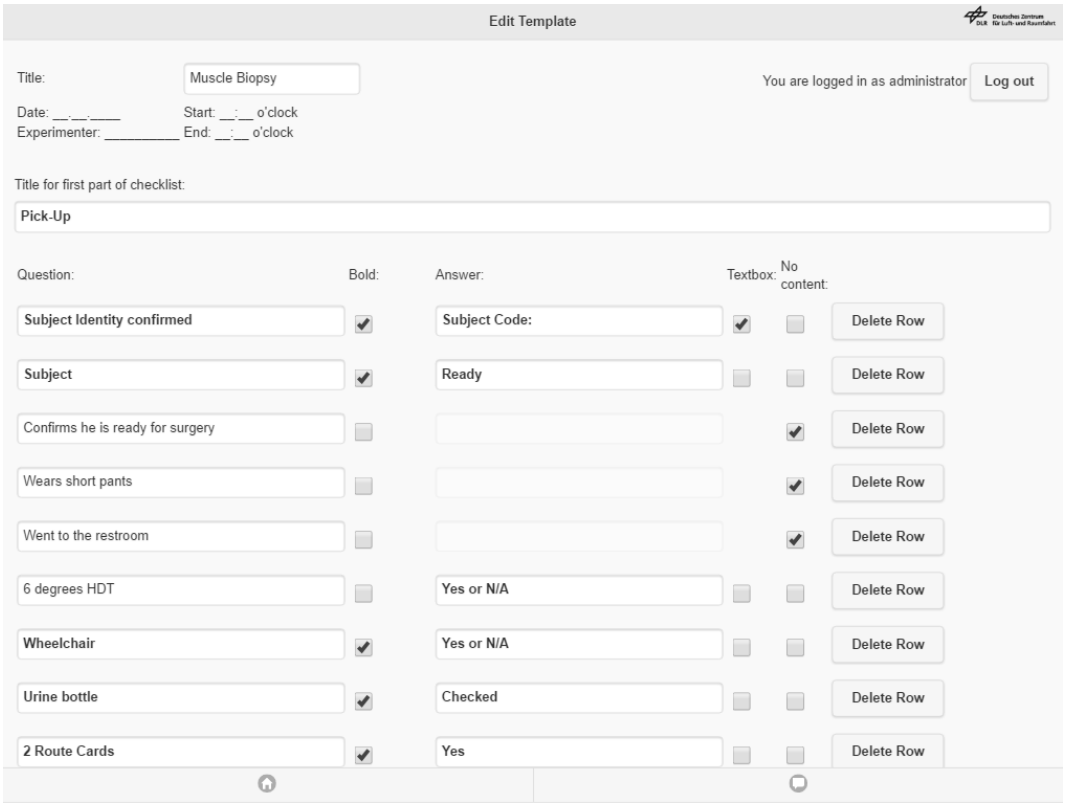


Figure 4.14: Template Editing

As described in Chapter 4.1 on page 20, only the administrator can access the editing page shown in Figure 4.14. The page takes a template and breaks down each line into its elements for convenient editing.

All three parts of a checklist are displayed in one table, that is defined in the ASPX page as follows:

```
<label for="Textbox_Title_Part1" style="font:bold">
        Title for first part of checklist:
</label>
<asp:TextBox ID="Textbox_Title_Part1" runat="server"
    Font-Bold="true"></asp:TextBox>
<asp:table id="Table_Content_Part1" runat="server" Width="90%">
        <asp:TableRow>
                <asp:TableCell>Question:</asp:TableCell>
                <asp:TableCell>Bold:</asp:TableCell>
                <asp:TableCell>Answer:</asp:TableCell>
                <asp:TableCell>Textbox:</asp:TableCell>
                <asp:TableCell>No content:</asp:TableCell>
        </asp:TableRow>
</asp:table>
```

Each part is delimited by a header comprising a label and a text box, which allows the user to enter a new title. The static part of the table serves as a legend for the individual columns.

The rest of the table is filled dynamically. In order to achieve this, the content has to be retrieved from a XML file. With the help of the template ID, the associated file is loaded:

```
Dim selected_template As String
selected_template =
    Server.HtmlEncode(Request.Cookies("selected_template").Value)

Dim template_file = XElement.Load(MapPath("~/Checklist_Templates/" &
    selected_template & ".xml"))
Dim list_of_templates_file =
    XElement.Load(MapPath("~/List_of_templates.xml"))
Dim list_of_templates =
        From template In list_of_templates_file.<template>
        Where template.<GUID>.Value = selected_template
        Select
        title = template.<title>.Value
Textbox_Title.Text = list_of_templates.FirstOrDefault

Dim template_content_part1 =
        From template In template_file.<Part1>
        Select
        title_Part1 = template.<title>.Value
Textbox_Title_Part1.Text = template_content_part1.FirstOrDefault
```

After displaying the title of the entire checklist as well as the title for each individual part, the content has to be loaded. For simplification, only the first part is described here. All *<row>* elements of the first part are retrieved from the XML file and counted in order to obtain the number of rows necessary.

```
Dim rows_content =
From row In template_file.<Part1>.<Row>
Select
       question_text = row.<Question>.Value,
       bold_check = row.<Bold>.Value,
       answer_text = row.<Answer>.Value,
       textbox_check = row.<Textbox>.Value,
       no_content_check = row.<No_Content>.Value
Dim number_of_rows_in_xml =
       Aggregate row In template_file.<Part1>.<Row>
Into Count()
Dim number_of_rows_in_table = number_of_rows_in_xml - 1
Session("number_of_rows") = number_of_rows_in_table
```

This process is comparable to the one described in Chapter 4.4.1 on page 32. The big difference is the way, the checklist is displayed. The whole content has to be editable, therefore question and answer texts are displayed in text boxes. All configuration options are represented by check boxes. A button is added to each line to allow for deletion of a whole row. Its functionality is thoroughly described on page 67.

```
Dim Textbox1 As System.Web.UI.WebControls.TextBox
Dim Textbox2 As System.Web.UI.WebControls.TextBox
Dim Checkbox1 As System.Web.UI.WebControls.CheckBox
Dim Checkbox2 As System.Web.UI.WebControls.CheckBox
Dim Checkbox3 As System.Web.UI.WebControls.CheckBox
Dim Button1 As System.Web.UI.WebControls.Button
```

The six controls are created for all rows, starting with *0*. Each control is surrounded by its own table cell. In the listing below, it is described how a text box is added. Please note that an event handler is added to the control. Changes made by the user shall be visible instantly. Therefore it is necessary that the server registers changes made to a text box. In order for this to work, the text box has to automatically post back changes to the server.

```
Dim row_num As Integer
For row_num = 0 To number_of_rows_in_table
        Dim tempRow As New TableRow()
        Textbox1 = New TextBox()
        Textbox1.ID = "Textbox_Question_" & number_of_parts.ToString &
            "_" & row_num.ToString
        Textbox1.Text =
            rows_content.ElementAt(row_num.ToString).question_text
        Textbox1.AutoPostBack = True
        Dim tempCell_1 As New TableCell()
        tempCell_1.Controls.Add(Textbox1)
        AddHandler Textbox1.TextChanged, AddressOf
            Me.Textbox_Question_1_Textchanged
        tempRow.Cells.Add(tempCell_1)
        tempCell_1.Width = Unit.Percentage(40)
```

The process is repeated for the remaining text box as well as for two check boxes. The button mentioned before is added at last. Please note that the address of the handler is the same for all buttons, added for the first part of the checklist. Combining a large number of buttons in only one click event substantially reduces the code lines required. It goes without saying that the ID is unique to each button. Otherwise it would not be possible to distinguish between them.

```
Button1 = New Button()
Button1.ID = "Button_delete_row_" & number_of_parts.ToString & "_" &
    row_num.ToString
Button1.Text = "Delete Row"
Dim tempCell_6 As New TableCell()
tempCell_6.Controls.Add(Button1)
AddHandler Button1.Click, AddressOf Me.Button_Delete_Part1_Click
tempRow.Cells.Add(tempCell_6)
Table_Content_Part1.Rows.Add(tempRow)
```

The controls are depending on one another. If the check box *bold* is checked, the text in the *question* text box ix printed in bold. Moreover, the check box *no_ content* determines if the right part of the row is greyed out or not. The configuration options enable the user to modify all parts of the checklists as he pleases.

As has been stated before, one single event handler covers the delete buttons of all rows in the first part of the checklist. When one of those buttons is clicked, a sub procedure is started. First, it has to be precisely determined, which of the buttons was the originator. With the help of the *CType* function, the expression *sender* can easily be converted to a button class.

```
1  Private Sub Button_Delete_Part1_Click(ByVal sender As System.Object,
       ByVal e As System.EventArgs)
2          Dim selected_template As String
3          selected_template =
               Server.HtmlEncode(Request.Cookies("selected_template").Value)
4          Dim name_of_button = CType(sender, Button).ID
5          Dim row_number As String = 0
6          If name_of_button.Length = 21 Then
7                  row_number =
                       name_of_button.Substring(name_of_button.Length - 1)
8          Else
9                  row_number =
                       name_of_button.Substring(name_of_button.Length - 2)
10         End If
11         row_number = row_number + 1
12         Dim template_file = MapPath("~/Checklist_Templates/" &
               selected_template & ".xml")
13         Dim template_xml = XElement.Load(template_file)
14         Dim rows_content =
15                 From row In template_xml.<Part1>
16         rows_content.Elements.ElementAt(row_number).ReplaceWith("")
17         template_xml.Save(template_file)
18         Response.Redirect(HttpContext.Current.Request.Url.ToString(),
               True)
19 End Sub
```

The ID of the button includes the row designator, which is either a single-digit or double-digit number at the end of string. Removing all characters before the number and adding *1* you receive the row number, the click originated from.

The associated *<row>* entry in the XML file is located and removed. To be more precise, the element is selected and then replaced with nothing. After saving the XML file, the page is reloaded to display the changes.

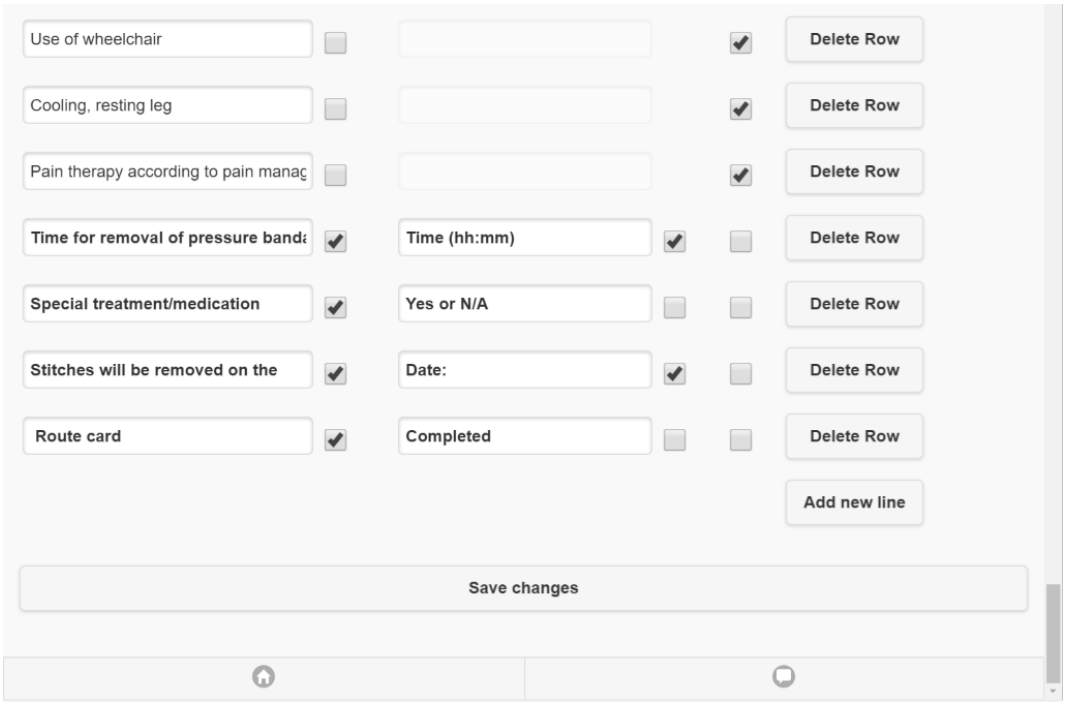


Figure 4.15: Adding and Deleting Rows in a Template

Of equally importance as removing entries is the ability to add new lines in order to extend a checklist. To achieve this, a button is added at the end of each part of the template. When clicked, a new *<row>* entry is added to the XML file and the page is reloaded:

```
Dim template_file = MapPath("~/Checklist_Templates/" &
    selected_template & ".xml")
Dim template_xml = XElement.Load(template_file
Dim last_entry_in_xml =
        From row In template_xml.<Part1>
Dim new_row As XElement =
        <Row>
                <Question></Question>
                <Bold>false</Bold>
                <Answer></Answer>
                <Textbox>true</Textbox>
                <No_Content>false</No_Content>
        </Row>

last_entry_in_xml.Elements.Last.AddAfterSelf(new_row)
template_xml.Save(template_file)
Response.Redirect(HttpContext.Current.Request.Url.ToString(), True)
```

It has been stated before that all text boxes on the page have to automatically post back to the server if the user entered new text. Otherwise, all changes would be lost during the next page reload. Analogous to the *Delete row* button, one event handler covers all text boxes that contain questions. After identifying the specific text box that has changed since the last page load, the new information has to be stored into the XML file. The *CType* function cannot only retrieve the ID of the sender, but also the text. From the text, a new *XElement* is created. It is meant to replace the text that is currently saved in a sub-element of the *<row>* element. Following the server's counting method, the question's text is sub-element number *0*, because it is the first one.

```
1  Protected Sub Textbox_Question_1_Textchanged(sender As Object, ByVal
       e As EventArgs)
2   Dim selected_template As String
3   selected_template =
        Server.HtmlEncode(Request.Cookies("selected_template").Value)
4   Dim name_of_textbox = CType(sender, TextBox).ID
5   Dim row_number As String = 0
6   If name_of_textbox.Length = 20 Then
7    row_number = name_of_textbox.Substring(name_of_textbox.Length - 1)
8   Else
9    row_number = name_of_textbox.Substring(name_of_textbox.Length - 2)
10  End If
11  Dim new_text_of_textbox = CType(sender, TextBox).Text
12  Dim new_entry As XElement =
13        <Question><%= new_text_of_textbox %></Question>
14  row_number = row_number + 1
15  Dim template_file = MapPath("~/Checklist_Templates/" &
        selected_template & ".xml")
16  Dim template_xml = XElement.Load(template_file)
17  Dim rows_content =
18        From row In template_xml.<Part1>
19  rows_content.Elements.ElementAt(row_number).Elements.ElementAt(0).
        ReplaceWith(new_entry)
20  template_xml.Save(template_file)
21  Response.Redirect(HttpContext.Current.Request.Url.ToString(), True)
22 End Sub
```

All input options of the editing page are equipped with event handlers similar to the one described above. Thus, all changes made by the user are recognized by the server and directly transferred to the associated XML file. For reasons of simplicity, not all sub procedures are discussed in this chapter. A copy of the complete source code is attached to this thesis on a CD-ROM.

4.7 Error Reporting Page

Software almost always contains errors, especially if it is as complex as this application. The entire Visual Studio project contains almost 5,000 lines of code, not including XML files. Since errors cannot be completely precluded, it is important to learn from their occurrence. A short sub procedure is included in the code-behind files of all ASPX pages. It is intended to save the type of error and the page which it occurred on in session variables before redirecting the user to a pre-defined error landing page:

```
1 Public Sub Page_Error(Sender As Object, e As EventArgs) Handles
    Me.Error
2  Dim last_error As Exception = Server.GetLastError()
3  Dim error_URL As String = Request.Url.ToString()
4  Session("last_error") = last_error
5  Session("error_URL") = error_URL
6  Response.Redirect("Error_Page.aspx")
7 End Sub
```

The user is forwarded to a website called *Error_page*, which informs him about the error which has occurred. Using the text box, additional information about circumstances or previous actions that might have led to the error can be supplied.

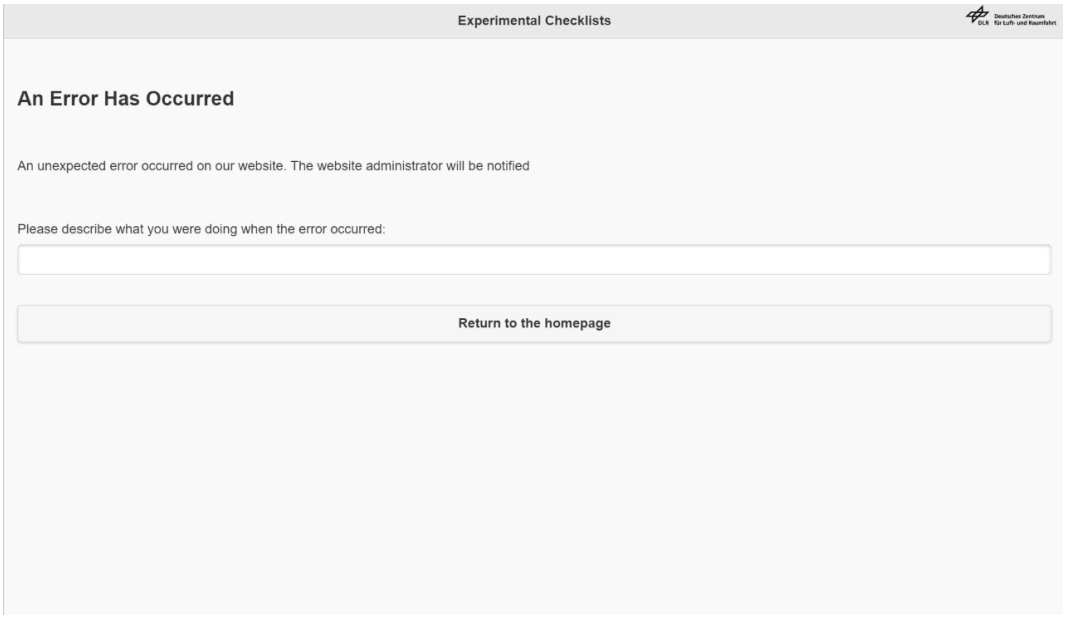


Figure 4.16: Error Reporting Page

A sub procedure is started when the *Return to home* button is clicked. It serves two purposes: one is to send an email to the system administrator giving as much information about the error as possible and the second is to redirect the user to the default page.

A sample email for reporting an error can be found in Attachment F on page R. It contains information retrieved from the server and from the user, if applicable:

```
Dim last_error = Session("last_error")
Dim error_URL = Session("error_URL")
Dim comment_fromuser As String = "No comment"
If TextBox_comment.Text IsNot Nothing Then
        comment_fromuser = TextBox_comment.Text.ToString()
End If
Dim error_description As String = "<b>Page, on which the error
    occured: </b>" & error_URL &
 "<br><b>Error message: </b>" & last_error.Message.ToString() &
 "<br/><b>Comment from user: </b>" & comment_fromuser &
 "<br/><b>Stack trace:</b><br>" & last_error.ToString()
```

As discussed in previous chapters, the email is generated and sent using a SMTP client. The subject line and the high priority of the message point out the importance of the email. The body then gives detailed information which makes trouble shooting as easy as possible.

After clearing the error on the server, the user is redirected to the default page, enabling him to restart a checklist.

```
Dim new_email As MailMessage = New MailMessage()
        new_email.From = New MailAddress("malte.fuhrmann@dlr.de")
        new_email.Body = error_description
        new_email.IsBodyHtml = True
        new_email.Subject = "An error occurred in the DLR Checklist
            Application"
        ...

        Server.ClearError()
        Response.Redirect("Default.aspx")
```

An error page is a very handy tool during the process- of improving an application. During the testing phase, a few minor problems occurred, mainly because of changes to the code-behind files when adding functions. All of the errors have been resolved, leaving a reliable and stable application to work with.

Chapter 5

Results

Checklists were introduced and tested during the recovery phase of the second RSL campaign in April, 2016. For reasons of simplicity, results are displayed separately for each type of experiment, starting with spiro-ergometries. In the table below, the different medical doctors who performed experiments are assigned the corresponding colors included in the following diagrams. For reasons of data protection, the names have been replaced by anonymous designations.

Experimenter	Color
Doctor A	Green
Doctor B	Orange
Doctor C	Yellow
Doctor D	Blue

Table 5.1: Legend of Medical Doctors performing Experiments

For each type of experiment, three indicators for efficacy have been selected. Firstly, overall performance is measured, taking the entire list of actions into account that are performed during an experiment. Secondly, only items are measured that deal with the safety of subjects. Thirdly, the input of checklists on the quality of scientific data is analyzed taking the associated items into account. The results are displayed in graphs to show the development of performance over time as well as the effect different experimenters have on experiments.

The chapter is concluded by an evaluation of feedback questionnaires indicating the acceptance and contentedness of checklist application users.

5.1 Spiro-ergometry

During the BDC phase, spiro-ergometries have been performed for each subject. Unfortunately, one subject dropped out of the study for medical reasons shortly before bed rest. Therefore, only eleven subjects participated in spiro-ergometry tests during the recovery phase. Figure 5.1 shows that the first ten experiments before the bed rest have been performed by *Doctor A* and the last two ones by *Doctor C,* whereas all experiments during the recovery phase were conducted by *Doctor A*.

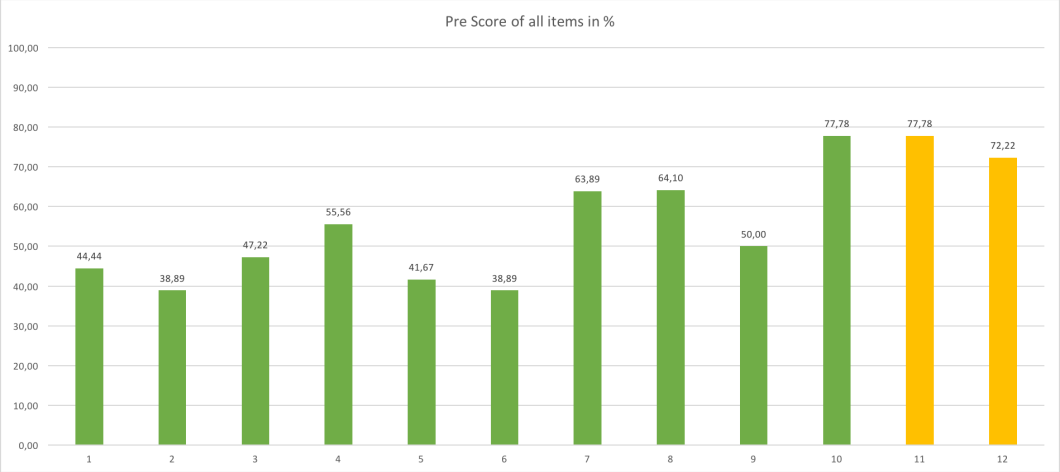


Figure 5.1: Pre Phase Analysis of all VO2max Items

The average overall performance during the experiment without the use of checklists amounted to 56%, which was significantly increased to 98% during the recovery phase.

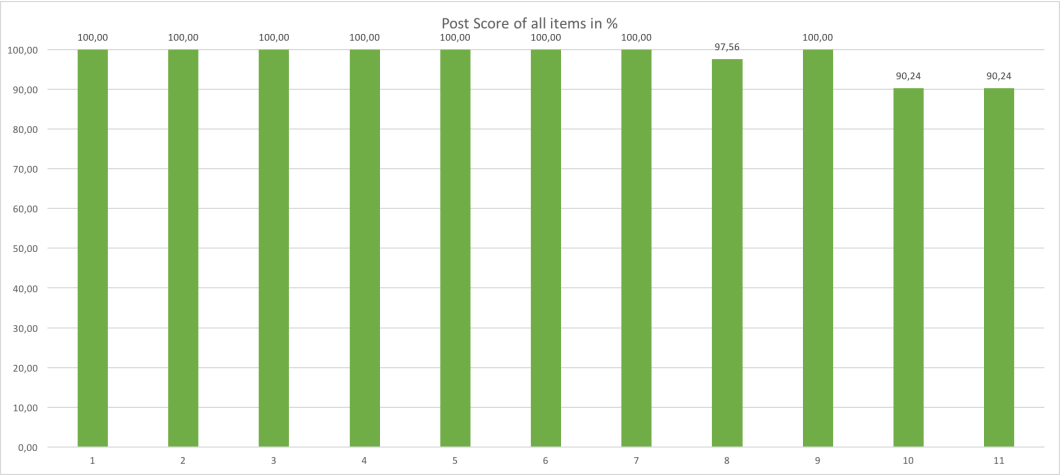


Figure 5.2: Post Phase Analysis of all VO2max Items

In order to measure the impact of checklists on subject safety, five indicators were defined as can be seen below in Table 5.2. The complete list of items that were used to evaluate spiro-ergometries can be found in Attachment A.

Item #	Safety Item
1)	Subject confirms that he is healthy and feels capable for physical performance test
2)	Physical examination performed by medical doctor (heart and lung auscultation)
3)	Emergency equipment checked (emergency case and defibrillator)
4)	Subject has been briefed by medical doctor about the experiment
5)	Special treatments ordered by doctor are clearly communicated and marked on timetable, if applicable

Table 5.2: List of VO2max Safety Items

Taking only the five items into account that reflect essential necessities for a safe experiment with regard to subjects, the performance before bed rest was slightly below the overall performance. The score of 51% could be increased to 96% by the use of checklists as can be seen in Figure 5.4 on the following page.

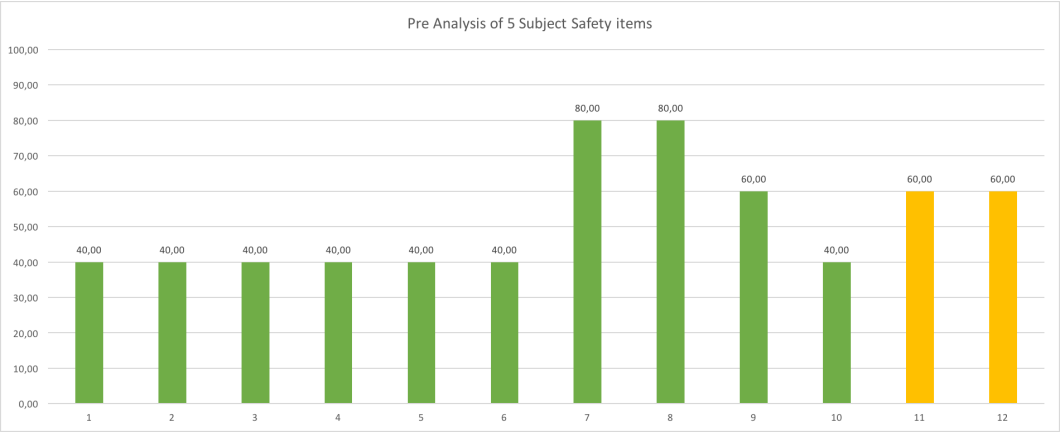


Figure 5.3: Pre Phase Analysis of VO2max Safety Items

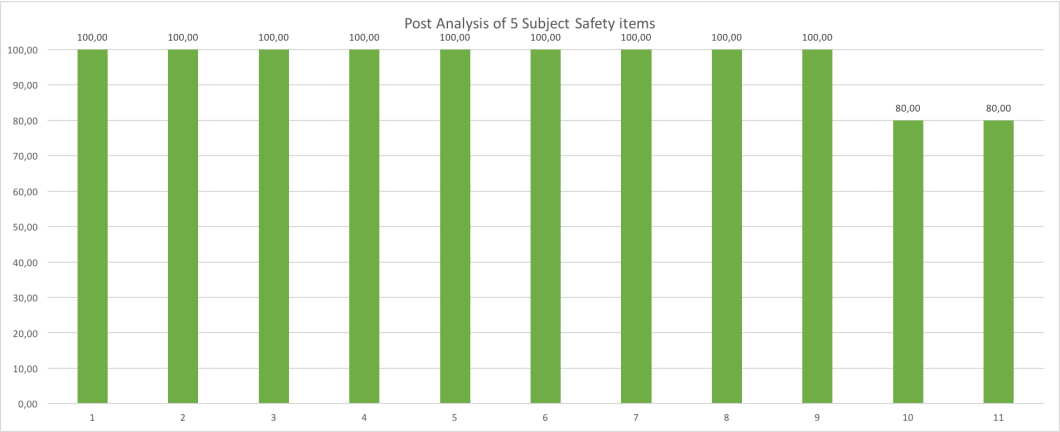


Figure 5.4: Post Phase Analysis of VO2max Safety Items

Besides subject safety, the result of an experiment is very important. The scientific data has to be reproducible and significant for an experiment to be effective. Therefore, ten items have been selected that make the scientific quality of an experiment measurable:

Item #	**Quality Item**
1)	Identity of subject confirmed
2)	Subject's face is shaved
3)	Subject has warm hands
4)	File name is correct
5)	Innocor is calibrated and running
6)	Gas bottle is locked tight and pressure is $>$ 10 bar
7)	Breathing mask is tight
8)	Full batteries for holter ECG
9)	Scientific ECG running (leads are correct)
10)	Data saved and backup performed

Table 5.3: List of VO2max Quality Items

When attention is paid to all ten items, the results of an experiments are most likely useful. If, for instance, a measuring apparatus is not calibrated correctly or a battery is not sufficiently charged to last for the duration of an experiment, data will be lost. Naming files correctly and performing a data backup is of similar importance. Figure 5.5 shows the performance during the BDC phase with regard to data quality:

Figure 5.5: Pre Phase Analysis of VO2max Quality Items

The effect of checklists on the quality of data is significant. Results as low as 40% were achieved before the bed rest. With the use of checklists, no item assuring data quality was forgotten which increased the score to 100% as can be seen below:

Figure 5.6: Post Phase Analysis of VO2max Quality Items

5.2 Muscle Biopsy

Five days before the start of the HDT phase, muscle biopsies were performed on each subject. The experiment comprises two parts due to the combination with OGTT. Therefore, two data sets have been recorded for each subject. The first one includes picking up the subject and the first part of the biopsy. The second one represents the last part of the biopsy and subject drop off.

The procedure was repeated after the bed rest with the same segmentation. Please note that for the recovery phase only 22 data sets exist, due to the drop out of one subject as described before. The first two subjects had their biopsies taken by *Doctor B*, whereas all subsequent biopsies were performed by *Doctor A*.

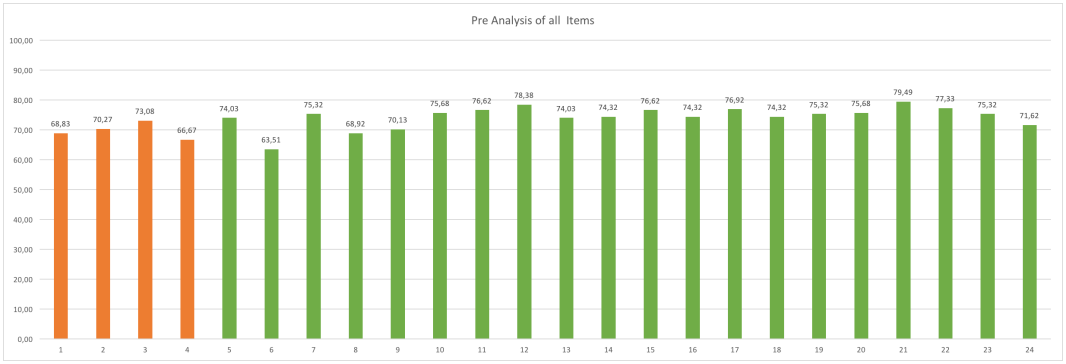


Figure 5.7: Pre Phase Analysis of all Biopsy Items

Figure 5.7 shows the overall performance of muscle biopsies in chronological order for the BDC phase. The average performance of 74% has been increased to 95% by the use of checklists as can be seen below:

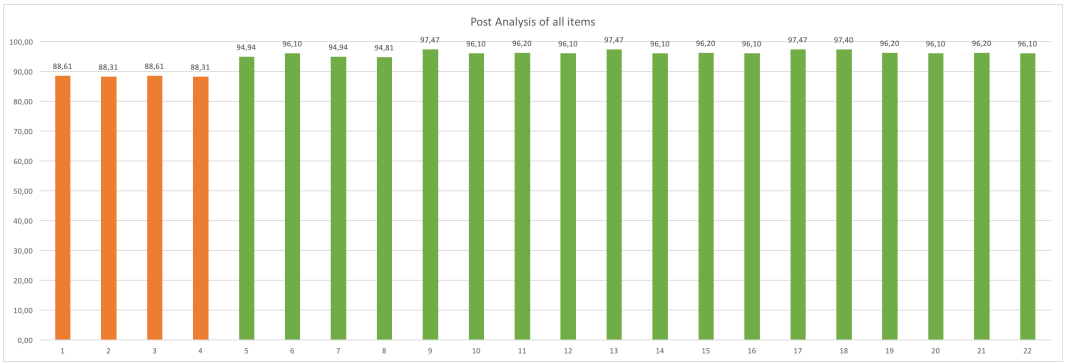


Figure 5.8: Post Phase Analysis of all Biopsy Items

A complete list of all 83 items included in this analysis can be found in Attachment B.

Since subject well-being and health is of paramount importance during muscle biopsies, nine indicators have been selected to point out the effect of checklists on subject safety:

Item #	Safety Item
1)	Subject is ready for surgical intervention
2)	Biopsy room prepared according to DLR hygienic standards
3)	Emergency equipment checked (emergency case and defibrillator)
4)	Best before date of all surgical tool checked
5)	Surgeon and assistant ready (hands washed and sanitized, surgical clothing, etc.)
6)	Datex Ohmeda ECG monitor running
7)	Subject has no relevant allergies
8)	Pulse status of lower leg checked
9)	Special treatment ordered by doctor are clearly communicated and marked on timetable, if applicable

Table 5.4: List of Muscle Biopsy Safety Items

The list covers all items that could endanger a safe experiment. Especially before the first biopsy to be performed on a subject it is important to know if he encountered complications during similar interventions. Bleeding disorders, allergy to the anesthesia to be used or inability to see blood are just a few possibilities. On the other side, the operating room and all of the equipment and personnel involved has to meet hygienic and medicinal standards. After the intervention, pressure bandages must not be too tight in order to allow sufficient blood flow through the lower extremities.

Figure 5.9 clearly shows that the level of subject safety is lower than the overall performance during biopsies. Only 41% were achieved during the BDC phase, with a maximum of 57% on the second to last biopsy day.

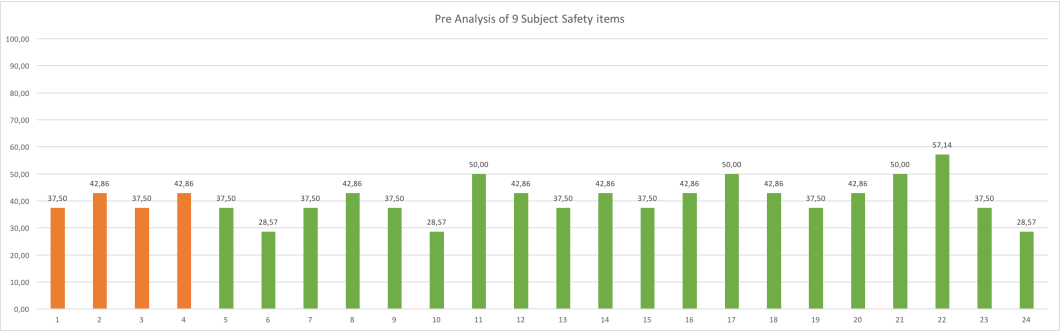


Figure 5.9: Pre Phase Analysis of Biopsy Safety Items

The muscle biopsy checklist places particular emphasis on subject safety. Thus, the score could be increased by 30% to 73%. The value for each experimenter is level and more consistent as before.

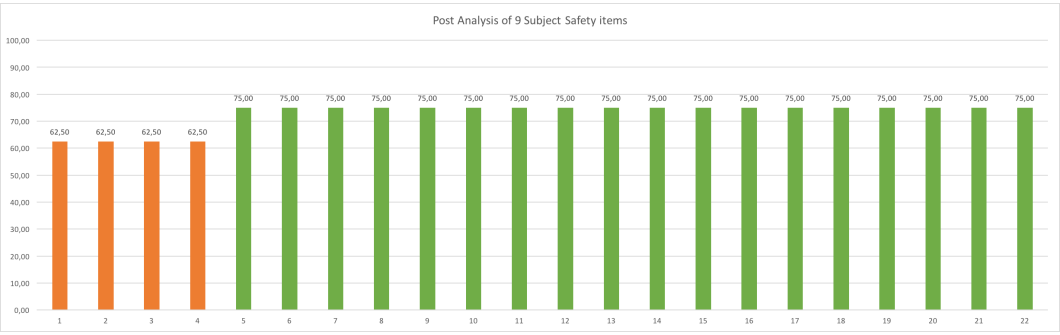


Figure 5.10: Post Phase Analysis of Biopsy Safety Items

Quality of data is very important for biomedical studies. If an experiment does not work out as planned, it can usually be repeated, if necessary. The more complex an experiment gets, the higher the effort to repeat it. Muscle biopsies are by far the most complex interventions performed during the RSL study. It takes a long time to prepare due to the combination with the glucose tolerance test. The day on which the experiment is performed, is decisive for comparable results. Therefore, the experiment cannot be repeated if the samples are not satisfactory. In order to eliminate as many factors threatening a successful operation, six items have been defined that contribute to data quality. The complete list can be found on the next page.

Item #	Quality Item
1)	Identity of subject confirmed
2)	Subject did not eat during the 10 hours prior to the experiment (only applicable if biopsy is performed in combination with OGTT)
3)	All surgical tools ready
4)	Ultrasound machine in reach
5)	Liquid nitrogen available in sufficient volume
6)	Subject uses wheelchair after experiment

Table 5.5: List of Muscle Biopsy Quality Items

It goes without saying that the unambiguous allocation of sample and subject code is extremely important for significant results. Therefore, the identity of each subject has to be confirmed before a surgical intervention. The subject is not allowed to eat during the last ten hours before OGTT in order not to compromise the measured blood glucose levels. The equipment needed for the intervention has to be complete and prepared, so that nobody has to leave the room during a biopsy. The performance during the BDC phase, taking only the six points mentioned above into account can be seen on the next page.

Figure 5.11: Pre Phase Analysis of Biopsy Quality Items

The score varies between one low of only 40% and highs of 80% with a mean of 72%. Figure 5.12 shows that the performance was tremendously increased to 100% throughout all 22 post measurements.

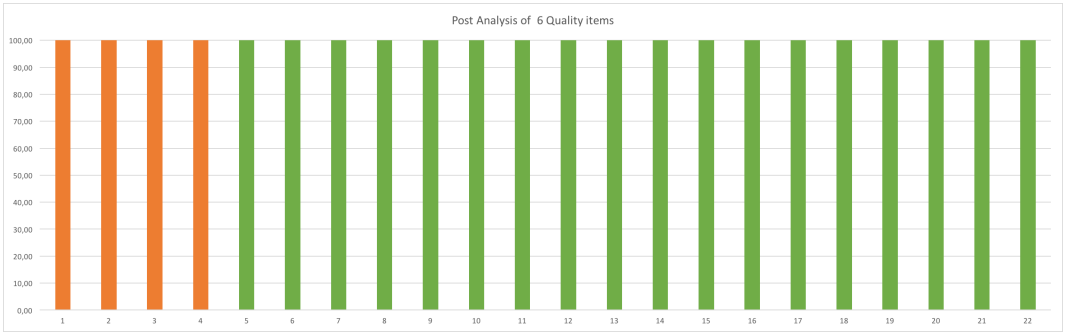


Figure 5.12: Post Phase Analysis of Biopsy Quality Items

5.3 Tilt Table

The tilt table experiment is the core element of a HDT bed rest study. It is the first experiment to be performed on the first day of the recovery phase in order to make the effect of the head down tilt on the cardiovascular system visible. A baseline test has to be performed before the bed rest, so that the collected data can be compared. A familiarization is performed for each subject at the beginning of the study to thoroughly explain the processes during the experiment. During the BDC phase, two familiarization tests and twelve normal tests have been observed, all of which have been monitored by *Doctor C.* The overall performance has been measured, taking all items of the protocol sheet into account, which can be found in Attachment C.

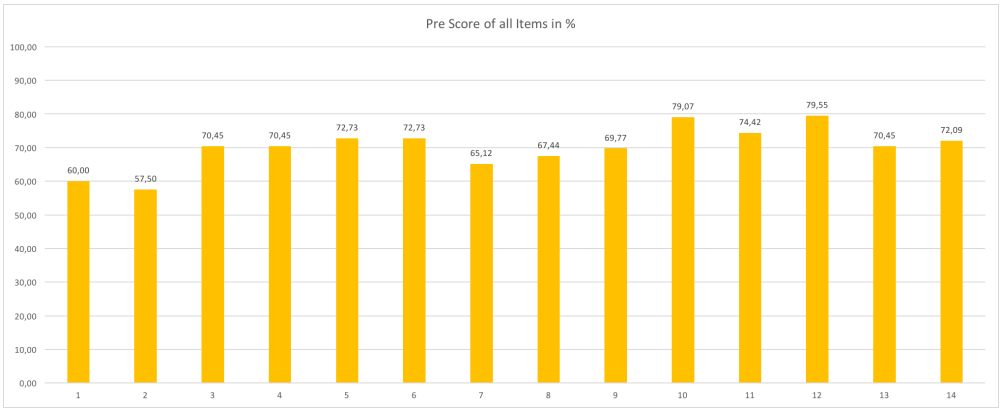


Figure 5.13: Pre Phase Analysis of all Tilt Table Items

The overall performance without the use of checklists varied between 58% and 80% with a mean of 70%. Due to the use of checklists, the performance was significantly enhanced to 98% with a low of only 94% during the eighth experiment.

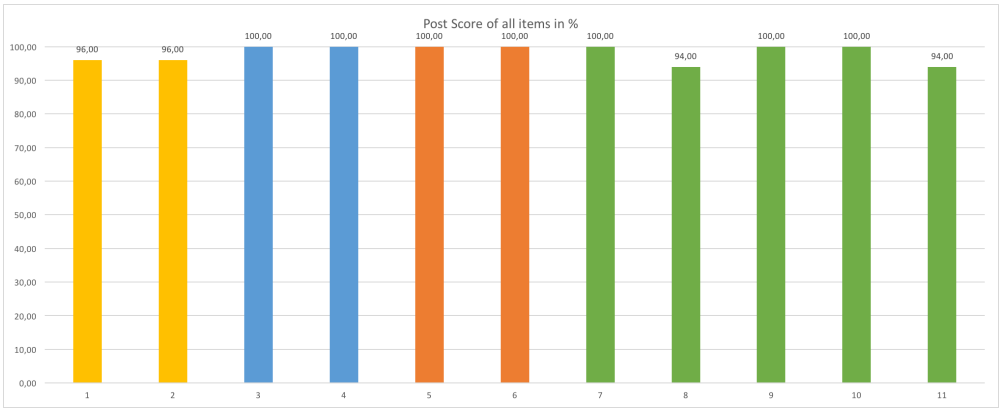


Figure 5.14: Post Phase Analysis of all Tilt Table Items

As for the other two types of experiments, a list of safety items has been created to make the effect of checklists on subject safety measurable:

Item #	Safety Item
1)	Scientific supervisor does not leave controls during experiment
2)	Subject confirms that he feels physically fit for experiment
3)	Medical ECG running, alarms are turned off
4)	Emergency case checked and in reach
5)	Special treatment ordered by doctor are clearly communicated and marked on timetable, if applicable

Table 5.6: List of Tilt Table Safety Items

Taking only the above mentioned five items into account, the performance during the BDC phase varied between 20% and 40% with a mean of 37%.

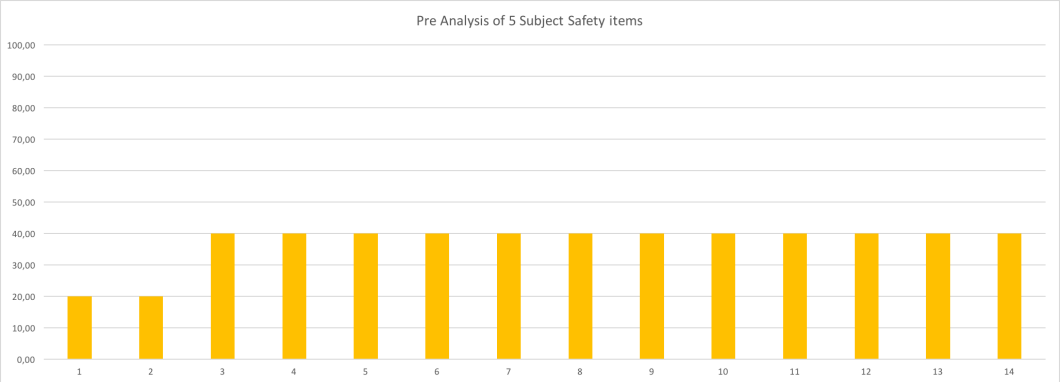


Figure 5.15: Pre Phase Analysis of Tilt Table Safety Items

Even though the experiments during the recovery phase were conducted by four different medical doctors, all safety items have been adhered to which results in a 100% performance during all eleven tests as shown in Figure 5.16 on the following page

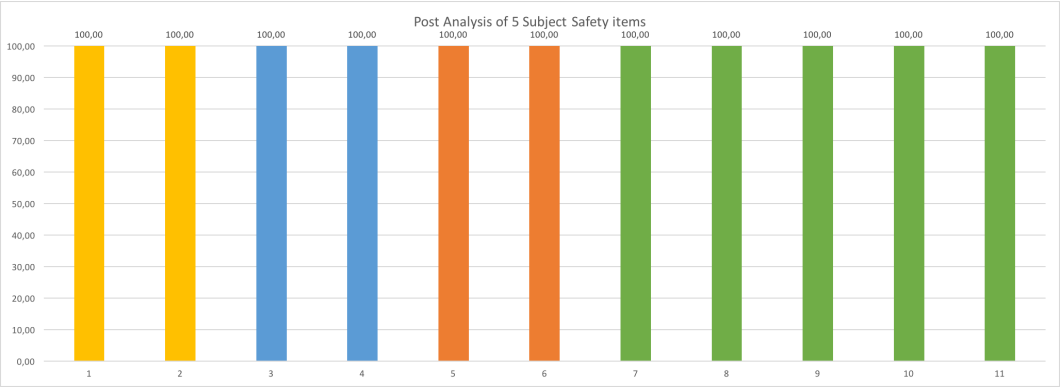


Figure 5.16: Post Phase Analysis of Tilt Table Safety Items

Quality of data is of high importance during tilt table testing, because the measuring instruments are very sensitive. Therefore, 10 indicators have been selected that assure comparable and reproducible results:

Item #	**Quality Item**
1)	Identity of subject confirmed
2)	No body lotion on subject's chest
3)	Subject has warm hands
4)	Holter ECG in place
5)	Subject code, age, height, weight entered correctly; right protocol is selected
6)	Scientific ECG running, signals clear, color code correct
7)	Microvasculature team ready
8)	Tilt table ready
9)	Subject is ready
10)	Scientific data saved and data backup performed

Table 5.7: List of Tilt Table Quality Items

The performance regarding data quality was higher than the overall performance to start with. Compared to a mean total performance of 70%, 81% were achieved during the BDC phase as shown below:

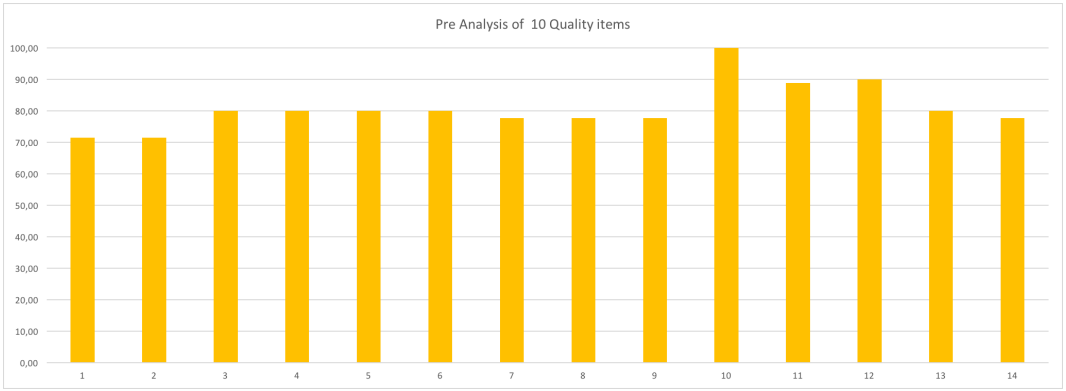


Figure 5.17: Pre Phase Analysis of Tilt Table Quality Items

Nevertheless, the data quality performance was increased by the checklists to 96% during the recovery phase with lows of 80% and highs of 100%.

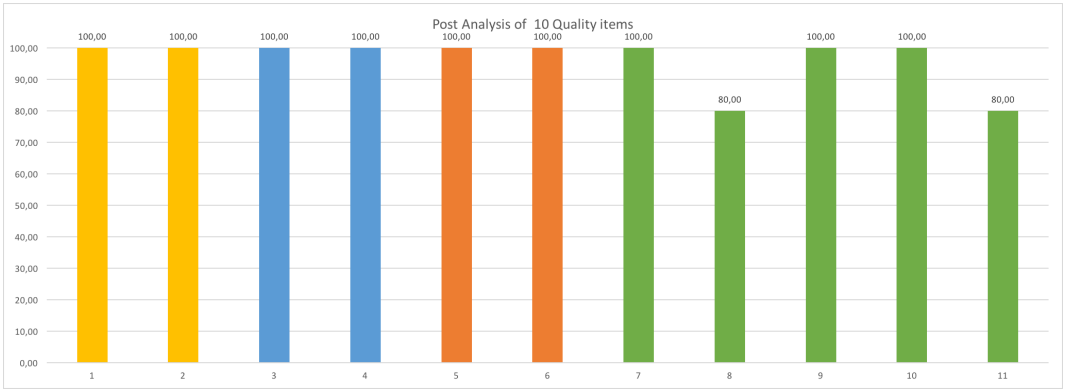


Figure 5.18: Post Phase Analysis of Tilt Table Quality Items

5.4 Feedback Evaluation

As described in Chapter 4.5 on page 59, a feedback form has been implemented into the checklist application in order to make the contentedness and satisfaction of the users measurable.

The results are displayed individually for each of the eleven questions. Figure 5.19 shows the legend for interpreting the results.

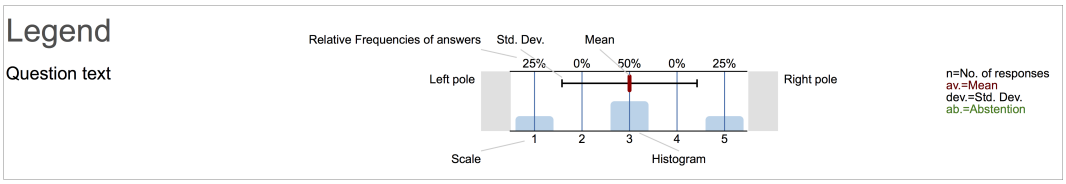


Figure 5.19: Legend for Statistical Analysis

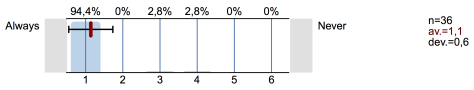
In total, 36 questionnaires have been filled out by users right after performing an experiment and using the checklist application. The scale ranges from *Always* to *Never*, showing the frequency of answers for each intermediate step. The numerical value for each bar is displayed above the scale. Mean value and standard deviation make the data statistically analyzable.

All users state that they know the code of the subject they have just performed an experiment with. Unambiguous allocation of samples or data and subject code is extremely important for significant results.

91,7% of all experimenters state that they have always checked the vital parameters of a subject before starting an experiment. Only 2,8% claim that they only sometimes check a subject's health.

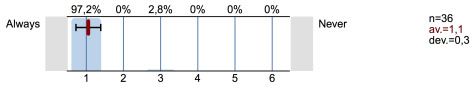
Experimenters are almost always aware of special treatment a subject might need during or after an experiment.

1.4) I am sure that the scientific data was saved or rather collected correctly



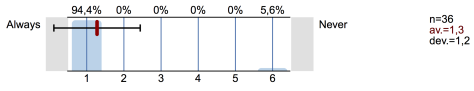
94,4% are sure that the scientific data or samples have been collected successfully after an experiment.

1.5) I know exactly what my tasks during the experiment are



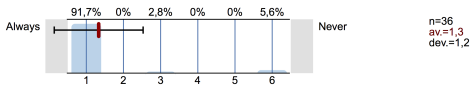
Over 97% of participants in experiments involving subjects know exactly what their tasks are. Only 2,8% answer with *Often.*

1.6) Prior to the start of the experiment, I am certain that all the equipment needed is ready



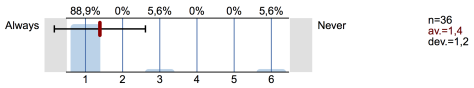
The standard deviation on the sixth question is relatively high, because 5,6% of all users stated that they are never certain before an experiment that all of the equipment needed was ready.

1.7) I know exactly where the emergency equipment is and how to use it



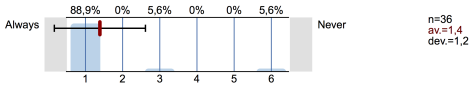
While 91,7% of experimenters are always sure where the emergency equipment is located and how to use it, 5,6% are never sure.

1.8) I know how to brief the subject after the experiment thoroughly



Since experiments like muscle biopsies are not ordinary for most subjects, post briefings are of high importance. 88,9% of experimenters state that they know how to thoroughly brief a subject on the do's and don'ts after an experiment.

1.9) I feel as a part of a team



Experiments are performed by scientists and medical doctors collectively. It is important that all personnel participating in an experiment feel as part of

a team in order to assure maximum performance. Only 5,6% state that they never feel as a part of a team.

The results shown in Chapter 5 speak for themselves. The performance was increased in all areas and all types of experiments. Nevertheless, the subjective impression of the users counts as well. Almost 89% stated that the checklist application prevented them from making a mistake. Only 2,8% are sure that they would not have made a mistake even without a checklist.

One important factor for the acceptance of checklists in biomedical studies is the time needed to read them. The more time a checklist takes to read, the higher the possibility that a user is tempted to conduct the next experiment without the checklist. Almost 89% stated that the checklist application in its current form does not take too long to read.

The results are summarized in Figure 5.20 for better comparability:

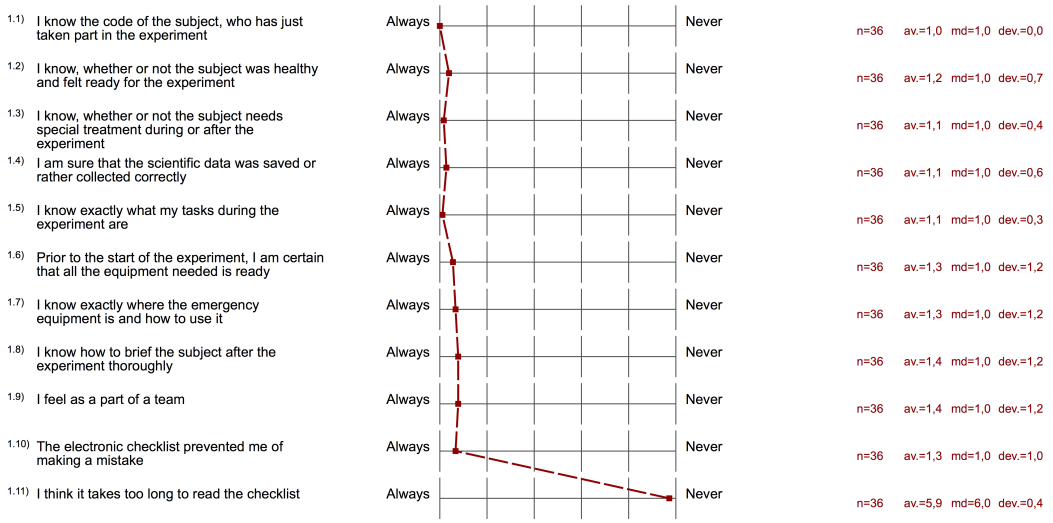


Figure 5.20: Profile Line (Mean)

Chapter 6

Discussion

Introduction of medical safety checklists for three complex experiments during a long-term biomedical bed rest study was associated with marked improvements in subject safety and quality of scientific data. In all three types of experiments, the number of mistakes were significantly reduced. Overall performance rose from 67% to 97% due to the use of checklists. Performance in view of subject safety was 39% at baseline and 90% after the introduction of checklists. Considering the quality of scientific data, the performance was increased from 64% to 99%. All experimenters experienced an increase in performance when using a checklist. Although the effect of the intervention was stronger during some experiments than during others, no single experiment was responsible for the overall effect, nor was the effect confined to one single experimenter exclusively. The increase in performance suggests that medical safety checklists can significantly increase the level of subject safety and the quality of scientific data in diverse experiments during biomedical studies.

While the evidence of enhancement in safety and quality of experiment performance is substantial and strong, the exact mechanism and cause of enhancement is not fully clear and quite likely multi-factorial. The use of checklists involved both changes in culture and attitude of individual experimental team members. Checklists create breaks in operation for teams to reflect on their actions before continuing with the next task. Furthermore, preceding team introductions, briefings and debriefings have previously been shown to be associated with improved safety processes and attitudes[1]. The checklists put emphasis on oral confirmation and process-oriented briefings. Also, mandatory checks of availability and completeness of emergency equipment before each experiment have been omitted often before the introduction of checklists. This philosophy proved to be new for most of the experimental team members. Other potentially lifesaving measures became routine tasks as well, including heart and lung auscultation and querying possible allergies to anesthesia or

[1] [HWB+09, New England Journal of Medicine, page 497]

known complications during previous interventions. Although individual actions have still been omitted, the overall performance with regards to safety and quality has been increased from 54% to 95%. The sum of individual changes in routine and attitude of experimenters could account for the large improvements observed.

Another mechanism that could attribute to the improvement of performance is the Hawthorne effect, also referred to as the observer effect. The effect describes a reaction in which individuals improve their performance in response to their knowledge of being observed[2]. The results of the experiments cannot easily be extricated from the contribution of the Hawthorne effect. The checklists encourage personnel to actively take care that all processes ensuring a safe conduction of experiments are obeyed in a team. However, an observer was present during all experiments covered by the results displayed. During the Baseline Data Collection phase, the experimenters were not shown the protocol sheets and thus did not know exactly which actions were observed. It is reasonable to think that the Hawthorne effect raised the level of performance during pre-measurements, if at all.

This thesis has several limitations. The small number of experiments performed in one single campaign was chosen because it was not possible to include more studies in the short amount of time given to prepare this thesis. The next long-term bed rest study hosted by the DLR in Cologne will not start until early 2017. Furthermore, the experiments were performed by a small number of scientists and medical doctors, including one examiner of this thesis who has been involved in the contentual conception of the checklists from the beginning. Because the content of protocol sheet has not been revealed to other scientists or assistants, this circumstance had no effect on other persons participating in experiments. Another limitation of this thesis is the learning curve effect. Some types of experiments were solely supervised by one single experimenter. Data analysis clearly shows that the performance was increased during consecutive experiments. Checklists have proven to reduce the failure rates during the first experiments. There might have also been a learning curve effect in the process of observing experiments and collecting data. It is likely that increasing numbers of failures would be listed over time, which would lead to an underestimation of the effect checklists have on safety and quality. One additional limitation is that data collection was restricted to three types of experiments. The effect of checklists on other types of experiments is unknown. This limitation is particularly relevant for experiments that are less complex or that are performed by a larger number of personnel.

A lesson learned from the implementation and development process of checklists is to focus on the user. It is of paramount importance to convince users of the positive effect checklists have on performance and safety.

[2][May03, The human problems of an industrial civilization]

Checklists must not be received as a tool to control or evaluate users, but as a safety feature. In order to achieve this, all prospective users have to be given opportunity to help shape checklists and contribute their own demands. The analysis of feedback questionnaires proves that the conduction of checklists is neither lengthy nor discouraging. The medical safety checklists improve the team spirit and establish a culture of active communication about safety concerns while flattening hierarchies.

One additional concern is how feasible the implementation process is in other areas. Electronic checklists only require three significant resources: an adequate number of tablet computers, a wireless network and disk space on a web server. Tablet computers and web servers can be leased to minimize fixed capital. A wireless network is presumably available in almost every major institution. The phase of testing the application and training experimenters on how to conduct medical safety checklists lasted only ten days. Thus it can be said that implementation of electronic checklists is neither costly nor lengthy.

There are still concerns that need to be addressed after the submission of this thesis. The procedures during experiments vary slightly between studies. All three checklist templates have to be revised and adapted to meet the requirements of the next study. In addition, checklists could be created for further experiments to acquire more experience with them. At this point, the application is running on a web server of the University of Applied Sciences in Bremen. The next logical step is to transfer the application to a web server hosted by the German Aerospace center, which brings many advantages. The login procedure could be integrated into the present infrastructure. Experimenters could use their DLR credentials to log on to the checklist application. Additionally, the server could directly access DLR network printers for making documentation available without the interim steps of sending an email and manually having to print it out.

Biomedical studies are a cornerstone of modern science and research. Great achievements have been derived for the medical and aerospace sector because of subjects participating in aforesaid studies. Health and safety of subjects must always have top priority. Aviation has learned from human errors that come with high consequences in many cases. Modern medicine benefits from this experience by implementing safety checklists in clinical operations. In this thesis, a checklist based application was associated with marked improvements in subject safety and quality of scientific data during a long-term biomedical study. Applied to a broader basis, checklists have the potential to significantly reduce the failure rate and improve overall performance in a variety of complex systems. Further study is needed to fully understand the effect of checklists and to initiate a sustained change in culture and behavior of individuals working in scientific research teams worldwide.

Bibliography

[Cer] Cermak, Petr: *Innovision Innocor*. http://www.compek.cz/pool/vzor/upload/Obrazky/Ergospirometrie/Innovision/RZ_Inncor_Innovision_EN_4c.jpg. – accessed June 23, 2016

[DW90] Degani, Asaf ; Wiener, Earl L.: *Human Factors of Flight-Deck Checklists: The Normal Checklist*. Contractor Report. https://ti.arc.nasa.gov/m/profile/adegani/Flight-Deck_Checklists.pdf. Version: 05 1990. – accessed on July 22, 2016

[ESA15] ESA: *60 Days and 60 Nights ... In Bed*. http://www.esa.int/Our_Activities/Human_Spaceflight/Bedrest_studies/60_days_and_60_nights_in_bed. Version: August 2015. – accessed on June 28, 2016

[Har12] Hare, Thomas van: *From The Ashes Of The Model 299*. http://fly.historicwings.com/2012/10/from-the-ashes-of-the-model-299/. Version: October 2012. – accessed on July 23, 2016

[HWB+09] Haynes, Alex B. ; Weiser, Thomas G. ; Berry, William R. ; Lipsitz, Stuart R. ; Breizat, Abdel-Hadi S. ; Dellinger, E. P. ; Herbosa, Teodoro ; Joseph, Sudhir ; Kibatala, Pascience L. ; Lapitan, Marie Carmela M. ; Merry, Alan F. ; Moorthy, Krishna ; Reznick, Richard K. ; Taylor, Bryce ; Gawande, Atul A.: A Surgical Safety Checklist to Reduce Morbidity and Mortality in a Global Population. In: *New England Journal of Medicine* 360 (2009), Nr. 5, 491-499. http://dx.doi.org/10.1056/NEJMsa0810119. – DOI 10.1056/NEJMsa0810119. – PMID: 19144931

[Ind15] India, Health B.: *The Checklist*. http://patientsafetyalliance.in/the-checklist/. Version: June 2015

[May03] Mayo, Elton: *The human problems of an industrial civilization.* London : Routledge, 2003 http://swb.eblib.com/patron/FullRecord.aspx?p=182337

[Mic] Microsoft Corporation: *FileUpload Class.* Microsoft Developer Network. https://msdn.microsoft.com/de-de/library/system.web.ui.webcontrols.fileupload(v=vs.110).aspx. – accessed June 15, 2016

[TA] Time ; Aksjeselskap, Date: *What Is a Leap Year?* http://www.timeanddate.com/date/leapyear.html. – accessed June 19, 2016

[Uni44] United States Army Air Force: *B-17 Training Manual.* PIF 2-10-1, November 1944. http://www.303rdbg.com/checklist-cockpit.html. – accessed on July 22, 2016

[Wor08a] World Alliance for Patient Safety: *WHO surgical safety checklist and implementation manual.* http://www.who.int/patientsafety/safesurgery/ss_checklist/en/. Version: 2008. – accessed on July 25, 2016

[Wor08b] World Health Organization: *Safe Surgery Saves Lives.* Version: 2008. http://apps.who.int/iris/bitstream/10665/70080/1/WHO_IER_PSP_2008.07_eng.pdf accessed on July 25, 2016

[Wor16] World Health Organization: *The checklist effect.* http://who.int/patientsafety/implementation/checklists/background/en/. Version: 2016. – accessed on May 23, 2016

Appendix A

Experiment Protocol for Spiro-Ergometry

DLR **Experiment Protocol**	Implementation of electronic checklists for muscle biopsies, tilt table experiments and spiro-ergometry in order to increase medical safety and scientific quality during biomedical studies at the DLR
VO2max	
Date:	Started at:
Subject:	Ended at:

1) Initial check during pick-up	
Identity of subject confirmed	☐
Subject wears sportswear and shoes	☐
Subject went to the restroom in the last 15 minutes	☐
Subject's face is shaved	☐
Subject's hands are warm (if necessary, use cherry pit bag)	☐
Last meal was at least 1 hour ago	☐
Subject has been weighed w/o wearing a shirt and before application of Holster ECG (weight written down)	☐
No drink allowed	☐
Towel is on hand	☐
If R+0: wheelchair is on hand	☐

2) Pre-start check	
Room temperature w/i limits (19 – 23 °C)	☐
Filename correct	☐
Innocor calibrated and running	☐
Gas bottle is locked tight and pressure is ≥ 10 bar	☐
Breathing mask is tight	☐
Full batteries for ECG holter	☐
ECG running and leads are correct	☐
Pulse oximeter running	☐
Blood pressure monitored	☐
Protocol sheet ready	☐
Borg scale ready	☐
No complications during last VO2max	☐
Subject did not take medication relevant for the test today	☐
Subject confirms he or she is healthy and feels capable of physical performance test	☐
Physical examination by medical doctor (heart and lung auscultation)	☐
Subject briefed about experiment	☐
Emergency equipment in reach: Emergency case checked and in reach Defibrillator checked and in reach Medical doctor and nurse in the room	☐ ☐ ☐ ☐
Briefing by medical doctor (termination criteria)	☐
Stretcher free	☐

3) Return check	
Data backup	☐
Subject weighted w/o shirt or holter ECG	☐
Weight difference communicated w/ kitchen	☐
Subject feels fine/not too exhausted	☐
Special treatment defined by doctor (if applicable): Blood pressure and heart rate measurements Examination by doctor Additional water in the next 30 minutes Necessary treatments marked on timetable	☐ ☐ ☐ ☐ ☐

Appendix B

Experiment Protocol for Muscle Biopsies

DLR **Experiment Protocol**	Implementation of electronic checklists for muscle biopsies, tilt table experiments and spiro-ergometry in order to increase medical safety and scientific quality during biomedical studies at the DLR
Muscle Biopsy	
Date:	Started at:
Subject:	Ended at:

1) Initial check during pick-up	
Identity of subject confirmed	☐
Subject is ready for surgical intervention	☐
Calf and thigh shaved'	☐
Subject wears short pants	☐
Subject went to restroom w/i past 15 min.	☐
Wheelchair, urine bottle and route card on hand	☐
If OGTT, then:	☐
Subject did not eat during the past 10h	☐
Blanket/reading material etc. ready	☐
I.v. in place	☐

2) Pre-anesthesia check	
Biopsy room prepared according to DLR hygienic standards	☐
Emergency equipment checked: Emergency case Defibrillator	☐ ☐ ☐
Prep razor	☐
Leukosilk hypoallergenic tape	☐
Bandage changing set/ Kodan antiseptic	☐
Cutasept spray antiseptics	☐
1 pair Vasco OP Protect sterile gloves	☐
Latex gloves	☐
MoliNea underpads	☐
Drape sheets	☐
Aperture drape sheet	☐
Two-piece adhesive aperture drapes	☐
Surgical skin marker	☐
Ultrasound machine	☐
Lidoject Anesthetics	☐
Disposable Hypodermic-needles (sizes 1 and 20)	☐
5 ml sterile syringe	☐
Sterilium hand disinfectant	☐
Cardboard MediBowls	☐
Surgical disposable scalpel	☐
1 Rongeur/1 Acecut automatic biopsy needle	☐
Crile-Wood needle holder	☐
Monocryl Antibacterial suture	☐
Surgical scissors	☐
Surgical forceps	☐

Best before date of all surgical tools checked	☐
Leukosan wound closure strips	☐
2 Gazin Gauze swabs	☐
2 Opsite Post-OP plasters	☐
Pressure bandages	☐
Bowl of cold water	☐
Normal waste bin	☐
Human biological waste bin	☐
Sharps container	☐
Liquid nitrogen	☐
Dry ice	☐
Surgeon: hands washed	☐
Surgeon: sanitized hands	☐
Surgeon: sterile gloves	☐
Surgeon: Surgical face mask	☐
Surgeon: Foliodress cap	☐
Surgeon: Surgical clothing	☐
Assistant: Hand disinfection performed	☐
Assistant: Gloves	☐
Assistant: Surgical face mask	☐
Assistant: Foliodress cap	☐
Assistant: Surgical clothing	☐
Pulse oximetry running	☐
Datex-Ohmeda ECG monitor running	☐
Blood pressure interval measurements	☐
Stretcher connected to outlet	☐
Head cushion for subject	☐
Surgical site confirmed on right leg and marked with pen	☐
Surgical site sanitized	☐
Subject has no allergies	☐
Subject does not have bleeding disorder	☐
Subject did not take medicine that day	☐
No complications after last biopsy	☐
Briefing by surgeon performed: In case of excessive bleeding… If anesthesia is not effective If subject loses consciousness Emergency telephone # announced	☐ ☐ ☐ ☐ ☐
3) After surgical intervention check	
Pulse status of lower leg checked	☐
Documentation of sample weight	☐
Documentation of sample site and time	☐
Documentation of complications (if applicable)	☐

4) Return check	
Subject arrived sitting in wheelchair	☐
Time for removal of pressure bandages has been communicated to the subject as well as the study office and has been marked on the timetable	☐
Subject briefed (cooling, no stress, etc.)	☐
Pain therapy according to pain management standard	☐
Special treatment/medication if applicable	☐
Time to remove stitches	☐
Route card filled out	☐

Appendix C

Experiment Protocol for Tilt table

DLR **Experiment Protocol**	Implementation of electronic checklists for muscle biopsies, tilt table experiments and spiro-ergometry in order to increase medical safety and scientific quality during biomedical studies at the DLR
Tilt table	
Date:	Started at:
Subject:	Ended at:

1) Initial check during pick-up	
Identity of subject confirmed	☐
Subject wears short pants	☐
Subject wears socks	☐
Subject is non fasting	☐
Subject went to the restroom during the last 15 minutes	☐
No body lotion on chest	☐
Warm fingers (if not, warmed with cherry pit bags)	☐
Urine bottle	☐
2 Route Cards	☐
Subject doesn't wear accelerometer	☐
Holter ECG applied	☐
If R+0: wheelchair is on hand	☐

2) Pre-start check	
Room temperature w/i limits (19 – 23 °C)	☐
Protocol sheet ready	☐
Subject code, age, height, weight entered correctly / right protocol selected	☐
Door is closed and all personnel is inside	☐
Scientific supervisor ready: Doesn't leave controls Scientific ECG running, signals are clear Finger blood pressure monitored All Cables fixed with Leukosilk	☐ ☐ ☐ ☐ ☐
Microvasculature team ready	☐
Tilt table is ready: Correct position (0° during BDC, -6° during HDT/R phase) LBNP cover in position and locked, all leaks taped Vacuum pump ready Table free to move (cables checked) Switch for base plate is off Diminution of LBNP installed	☐ ☐ ☐ ☐ ☐ ☐ ☐
Subject is ready: position according to protocol has been briefed about the experiment confirms that he/she feels physically fit for experiment	☐ ☐ ☐ ☐
Physical examination by medical doctor (heart and lung auscultation)	☐

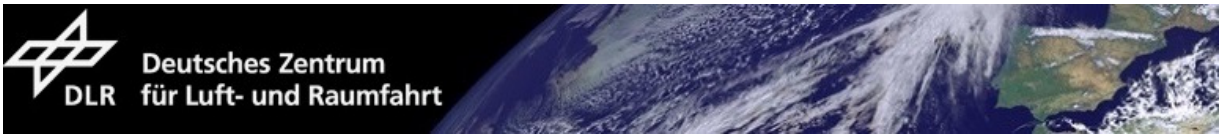
Medical ECG running / alarms turned off	☐
Blood pressure monitored and w/i normal range	☐
Emergency case checked and in reach	☐
Defibrillator checked	☐
Briefing by medical doctor performed: confirms that subject has been pre-examined briefing of termination criteria	☐ ☐ ☐
3) Return check	
Scientific data has been saved	☐
Route card filled out correctly	☐
Subject feels fine and doesn't faint when standing up	☐
R+0 only: No walking before PG/DGI test No walking of far distances during the first day No walking outside M3 during the second day Use of Event button on holster ECG	☐ ☐ ☐ ☐ ☐
Special treatment defined by doctor (if applicable): Blood pressure and heart rate measurements Examination by doctor 12 lead ECG recording Additional water in the next 30 minutes Necessary treatments marked on timetable	☐ ☐ ☐ ☐ ☐ ☐

Appendix D

Server generated Email containing Documentation of an Experiment

From: **Malte Fuhrmann** malte.fuhrmann@dlr.de
Subject: Documentation of a Muscle Biopsy for subject Test Subject
Date: 23. Juni 2016 um 08:30
To: Malte Fuhrmann mfuhrmann@me.com

MF



Muscle Biopsy for subject Test Subject

Date:	**01.08.2016**
Scheduled start:	**10:00**
Scheduled end:	**11:30**
Experimenter:	**Fuhrmann, Malte**

Pick-Up

Subject Identity confirmed	Subject Code: Test Subject	Checked
Subject	Ready	Checked
Preparation for OGTT	Checked or N/A	Checked
Subject did not eat during the past 10 hrs.		
I.V. in place		
Blanket and reading material on hand		
Confirms he is ready for surgery		
Wears short pants		
Went to the restroom		
6 degrees HDT	Yes or N/A	Checked
Wheelchair	Yes or N/A	Checked
Urine bottle	Checked	Checked
2 Route Cards	Yes	Checked
Muscle Biopsy and OGTT		

Pre-start

Subject is fit for surgery	Confirmed	Checked
No relevant allergies		
Emergency case within reach		
Defibrillator checked		
Biopsy Room	Prepared	Checked
DLR Hygienic standards are met		
Normal/Human biological waste bins ready		
Sharps container ready		
Sterilium hand disinfectant		
Bowl of water (if needed)		
Stretcher connected to outlet		
Ultrasound machine		
No bleeding disorder		
Head cushion for subject		
Sample Preparation Team	Ready	Checked
Liquid nitrogen and dry ice		
Surgical site	Prepared	Checked
Correct leg identified		
Marked with skin marker		
Antiseptics, underpads, drape sheets	Ready	Checked
Bandage changing set/ Kodan antiseptic		
Leukosilk hypoallergenic tape		
2 Underpads, 2 Drape sheets		
No medication intake today		
MediBowls		
Surgical instruments and anesthesia	Prepared	Checked
Lidoject Anesthetics		
Disposable Hypodermic-needles (sizes 1 and 20)		
5 ml sterile syringe		
Surgical disposable scalpel		
1 Rongeur/1 Acecut automatic biopsy needle		
Sewing equipment and bandages	Checked	Checked
Monocryl Antibacterial suture		
Crile-Wood needle holder		
No complications after last biopsy		

No complications after last biopsy			
Surgical scissors			
Surgical forceps			
Leukosan wound closure strips			
2 Gazin Gauze swabs			
2 Opsite Post-OP plasters			
2 Pressure bandages			
Briefing by surgeon	In case of…		Checked
In case of excessive bleeding			
If anesthesia is not effective			
In case of syncope			
Subject's vital parameters	In limits		Checked
Emergency phone #: 2222			
Pulse oximetry running			
ECG monitor running			
BP interval measurements selected			
Emergency equipment	Checked		Checked

Drop-off

Pulse status of lower leg	Present		Checked
a. tibialis posterior			
Route card	Completed		Checked
a. dorsalis pedis			
Subject briefing	Performed		Checked
Use of wheelchair			
Cooling, resting leg			
Pain therapy according to pain management standard			
Time for removal of pressure bandages	Time (hh:mm)	12:00	Checked
Special treatment/medication	Yes or N/A		Checked
Stitches will be removed on the	Date:	14.08.	Checked

Comments from Experimenter:
Sample comment

Appendix E

Server generated Email containing Feedback

From: **Malte Fuhrmann** mfuhrmann@me.com
Subject: You received new feedback for a VO2max experiment
Date: 23. Juni 2016 um 08:25
To: Malte Fuhrmann malte.fuhrmann@dlr.de

UR

Question 1: **always**
Question 2: **always**
Question 3: **always**
Question 4: **always**
Question 5: **always**
Question 6: **always**
Question 7: **always**
Question 8: **always**
Question 9: **always**
Question 10: **always**
Question 11: **never**

Comments:
No technical problems.

Attachment:
<VO2max_22-06-18_18-03.xml>

Appendix F

Server generated Email reporting a sample Error

An error occured in the DLR Checklist Application

June 13, 2016 at 6:37 PM

From malte.fuhrmann@dlr.de

Page, on which the error occured: http://localhost:2064/Conduct_checklists.aspx
Error message: Der Wert darf nicht NULL sein.
Comment from user:
Stack trace:
System.ArgumentNullException: Der Wert darf nicht NULL sein. bei Conduct_checklists.Button_save_click(Object sender, EventArgs e) in C:\Users\Malte\Documents\Visual Studio 2015\WebSites\DLRchecklists\Conduct_checklists.aspx.vb:Zeile 147. bei System.Web.UI.WebControls.Button.OnClick(EventArgs e) bei System.Web.UI.WebControls.Button.RaisePostBackEvent(String eventArgument) bei System.Web.UI.WebControls.Button.System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(String eventArgument) bei System.Web.UI.Page.RaisePostBackEvent(IPostBackEventHandler sourceControl, String eventArgument) bei System.Web.UI.Page.RaisePostBackEvent(NameValueCollection postData) bei System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeStagesAfterAsyncPoint)

Appendix G

CD-ROM

The CD-ROM contains three folders, each comprised of the following:

1. Digital Copy of the Thesis
2. Protocol Analysis
3. Visual Studio Files

To test the website, the following credentials are needed:

Username: *administrator*

Password: *admin123*

Please note that the email account that is used by the server to send emails is only valid until August 31, 2016. After this date, a different SMTP server has to be entered.