

UNIVERSITY OF TARTU  
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE  
Institute of Computer Science  
Software Engineering Curriculum

**Kaarel Kruus**

**Partiture: An Interoperable Music Stand  
Application for Mobile Devices**

**Master's Thesis (30 ECTP)**

Supervisor: Jüri Kiho, Prof.

Tartu 2014

# **Partiture: An Interoperable Music Stand Application for Mobile Devices**

## **Abstract:**

Almost every accomplishment can be retreated as an effort of teamwork and collaboration today – so can be creating and performing music. Composers, lyricists, arrangers, interpreters, collectives, conductors, choirmasters, music teachers and students – all their contribution intersects at some level. There are many tools and programs available, which assist for example composers in creating the artwork, arrangers in modifying it for concrete collective types, students in learning to play some instruments, etc. However, the circle of real-time collaboration supporting means is practically non-existent. Conductors, music teachers still do communicate all the context related notices to their mentees verbally, which often causes misunderstandings and information loss. This thesis presents the concept for interconnected interoperable music stand applications, which can enhance the cooperation between the musicians in the form of sharing page turns during performing, creating/sharing remarks and comments during practice sessions as well as locating retake positions fast and unambiguously, working over the air using ad-hoc based temporary networks. It is the introduction into the first phase of developing the corresponding software system (Partiture).

## **Keywords:**

Interoperability, collaboration, teamwork, music stand, score following, shared remarks, temporary networks, multipeer connectivity, performance testing, dispersion, mobile applications, requirements analysis, software system development, architecture planning;

## **Partiture: Koostöövõimeline noodipuldirakendus mobiilsetele seadmetele**

### **Lühikokkuvõte:**

Peaaegu kõik saavutused tänapäeval taanduvad meeskonnatööle ja ühistegevusele. Erandiks ei ole ka muusika loomine ja esitamine. Heliloojad, luuletajad, seadete autorid, interpreedid, kollektiivid, dirigendid, koormeisterid, muusikaõpetajad – kõigi panused lõikuvad mingil tasandil teiste panustega. Leidub palju programme ja tööriistu, mis näiteks abistavad heliloojaid komponeerimisel, seadjaid seade loomisel, õpilasi mingi instrumendi õppimisel, jne. Siiski on reaalajas muusika-alast kaastööd võimaldavate vahendite hulk praktiliselt olematu. Dirigendid, muusikaõpetajad jätkuvalt edastavad kontekstiga seotud märkusi oma juhendatavatele suuliselt, mis tihtipeale põhjustab arusaamatusi ja informatsiooni kadu. Käesoleva töö eesmärgiks on välja töötada omavahel ühenduses olevate koostöövõimeliste noodipultide kontseptsioon, mis võiks muuta muusikute omavahelist suhtlemist tõhusamaks jagatud noodilehe keeramiste, märkuste kui ka tagasivõtu kohtade määramisel kiirelt ja üheseltmõistetavalt, kasutades selleks spetsiaalseid lokaalselt ülesseatud ajutisi traadita võrke. Tegemist on vastava tarkvarasüsteemi (Partiture) arendamise esimese etapiga.

### **Võtmesõnad:**

Koostöövõimelisuus, kaastöö, meeskonnatöö, noodipult, takti jälgimine, jagatud märkmed, ajutised võrgud, võrdõigusvõrgud, koormustestid, hajusus, mobiilirakendused, nõuete analüüs, tarkvarasüsteemi arendus, arhitektuuri planeerimine;

## Table of Contents

Acknowledgements .....	6
1 Introduction .....	7
1.1 Problem statement .....	7
1.2 Contribution .....	9
1.3 Scope .....	9
1.4 Document structure .....	10
2 Background .....	11
2.1 Targeted platform .....	11
2.2 Development tools .....	11
2.3 Key concepts .....	11
2.4 External components .....	12
3 Related work .....	13
3.1 Non-collaboration supporting solutions .....	13
3.2 Collaboration supporting solutions .....	16
3.3 Other indirectly similar solutions .....	17
4 Requirements analysis .....	18
4.1 User stories .....	19
Business requirements .....	19
Functional requirements .....	21
Non-functional requirements .....	23
4.2 Implementation scope .....	25
5 Architecture .....	27
5.1 Overview .....	27
5.2 Logical view .....	28
5.3 Development view .....	30
5.4 Process view .....	31
Restoring application state .....	31
Authenticating user .....	32
Downloading scores .....	33
Building up the (re)connection .....	33
5.5 Physical view .....	35
5.6 Scenario view .....	36
5.7 Communication mechanisms .....	38

Current implementation .....	38
Alternative solutions .....	39
6 Validation .....	41
6.1 Functionality testing .....	41
6.2 Performance testing .....	41
Test data .....	41
Test method .....	42
Test devices .....	44
Test layouts.....	44
Hypotheses and conclusions.....	50
7 Future work .....	52
7.1 Extendable screen support .....	52
7.2 Extractable scores.....	53
7.3 Multiplatform support.....	54
7.4 Digital rights management .....	54
7.5 Score following aid.....	55
8 Conclusions .....	56
9 References .....	58
Resümee.....	60
Appendix.....	61
I. Performance tests graphical results.....	61
II. Storyboards .....	68
III. Prototypes .....	84
IV. License .....	85

## Acknowledgements

This thesis was completed in cooperation with Tartu Heino Eller Music School<sup>1</sup> teaching staff and students, who were a great help in defining and elaborating the requirements for the system and sharing interesting ideas for future work purposes.

I would like to acknowledge sincerely my supervisor; prof. Jüri Kiho, for the professional as well as moral support, guidance and inspiration, he has been offering and sharing with me during preparing for and writing this thesis.

I would also like to thank my partner and family members for their patience, and giving me the strength to complete this thesis.

Estonian Information Technology Foundation (EITF)<sup>2</sup> and Skype Technologies OÜ<sup>3</sup> were supporting the author with a scholarship in the academic year 2013/2014.

---

<sup>1</sup> <http://www.tmk.ee/en> (May, 2014)

<sup>2</sup> <http://www.hitsa.ee/en> (May, 2014)

<sup>3</sup> <http://www.skype.com/en/> (May, 2014)

<sup>4</sup> The exact value depends on various aspects, such as extracted score layouts or the characteristics (e.g.

# 1 Introduction

In this chapter the motivation and problem statement formulation for this thesis are presented, and the main contribution highlighted. Additionally the scope of the thesis project and a brief overview of the document are outlined.

## 1.1 Problem statement

Organizing the work of a 60-member choir or 100-member symphony orchestra is not a trivial task. There are usually many people behind the preparation of the practice sessions and / or live performances. Some of the time-consuming background processes, mostly invisible for the audience, could be accelerated or even automated by using modern technological solutions. It would enable to put more efforts and resources into the processes e.g. polishing the pieces during practice sessions. For example if a conductor selects a new musical piece for a choir or an orchestra, it has to be photocopied for all members, which costs. Additionally it has to be distributed among all members. Now when the conductor finds a printing error or (s)he wants to interpret (e.g. change some parts) something in his / her way, in many cases all the members have to „synchronize“ the changes on their copies (by using a pencil) as well – again another time-consuming move for the whole collective. Also not losing track of the currently singed or played measures and rustling page turnings during performances is one of many aspects to be considered. All these steps could be completed in a much faster and efficient way – by replacing traditional paper-based sheet music with locally interoperable digital music stands (displaying e-scores), which will also be the main focus point in this thesis.

There are numerous use cases, how interoperable music stands could support the musicians during practice and performance sessions. For example, professional artists, especially instrumentalists, often perform music by following the score. However, musical pieces usually contain repetitions, which also reflects in the score physical layout and this can sometimes lead to intense page turnings. There are several approaches, how this to is overcome, e.g.:

- the transition related measures are skipped during the page turning (e.g. a piano player can release one hand for turning the page, while playing with the other)
- the transition related prior (or subsequent) measures are memorized and the page turning will be performed during the last (or first) rest on the (next) page

- human assistance (standing or sitting next to the artist) is used for the page turning task

Using interoperable music stands, the assistant can comfortably sit somewhere nearby (e.g. even within the audience) and turn the pages remotely over the air.

Players in an orchestra are usually grouped pairwise and therefore can share the page turning task among them. However, when considering the whole orchestra – up to 50 percent<sup>4</sup> of the staff might be occupied with turning the page instead of creating music at a time. Using interoperable music stands, this could be handled from one stand for the whole orchestra.

During the practice sessions, the conductor can stop the music at a random point to explain his / her wishes or to highlight some mistake(s). Setting the unambiguous retake position can be tricky and take quite a lot of time, since many scores do not have measure indexes assigned. Therefore the conductor might return to a well-known point (e.g. “*a tempo*”), even if it is located 20 measures prior the real desired retake position, in order to avoid confusion and have everybody on the same spot. An interoperable music stand application can help to quickly distribute the exact location to the whole staff by tapping on the conductor’s stand (see Figure 1) – with less than a second.



Figure 1. Distributing pinpoint events.

Often the need arises to highlight something in the score (e.g. repeating mistakes, score printing errors, author’s / conductor’s interpretation wishes, breath mark changes, etc.) whether for someone privately or for the whole collective. The latter case can become quite a challenge, since there are almost always members, who do not pay attention in

<sup>4</sup> The exact value depends on various aspects, such as extracted score layouts or the characteristics (e.g. rests) in the piece.



time, do not always carry a pencil around, have been missing some practice sessions, etc. For example, even if a single choir member should sing in *forte fortissimo* whereas all others are singing in *piano pianissimo*, the conductor has to stop and explain once more the dynamics of the current context (measure, phrase, etc.). This can have a great impact not only on the rehearsal time, but also the conductor's and the whole collective's motivation, which can be reflected during the performance time. Using interoperable music stands, important remarks, created by the conductor, could be distributed (synchronized) within seconds to all the members. Even a member should have missed a number of practices, (s)he will be able to retrieve them all during the session time over the air.

## **1.2 Contribution**

There are a couple of systems available (listed in sub-section 3.2), which theoretically could provide a solution for these problems, when extending their functionality, but nothing like the planned system Partiture exists yet. Here, the uniqueness lies in the key concept of not requiring a persistent internet connection for the main operating work mode – events (e.g. page turning, pinpointing) and data (e.g. remarks, modifications) between the applications are exchanged over the air, using wireless technologies like Bluetooth and Wi-Fi. This means that the system can easily be transported to any preferred location. It also means that the system does not rely on the internet connection or some external services (except the user and score management system), which in case of going offline for only a few seconds could easily put an end to the concert.

As a contribution, a set of requirements for such kind of a system are collected, analyzed and outlined; possible communication mechanism options are discussed and briefly compared; the stability of a communication framework will be evaluated by conducting performance tests and summarizing the results; an application prototype will be developed; a set of further enhancement ideas will be given.

## **1.3 Scope**

Since time resource limitations, the application prototype will be developed for one mobile platform – Apple iOS only. Adding later the support for other mobile platforms like Android, should take less time because of the clearly emerged communication protocol and other workflows (sequences), forming the system. The most important requirements, based on their priority values, will be considered as first, all other requirements have to be considered later, outside the scope of this thesis project.

## **1.4 Document structure**

The first chapter of this document gave the overview of the main research problem and the proposal for a solution. The second chapter will give an overview of the background, such as which tools will be used, which frameworks have to be included. The third chapter will give an overview of similar solutions and concepts used by other solutions, related to this thesis. The fourth chapter will define and prioritize the requirements for the planned application. The fifth chapter will give an overview of the system architecture in detail. The sixth chapter presents the testing (validation) results of the application. The seventh chapter provides a starting point for the functionalities to be considered in the next iteration. The eighth chapter will summarize this thesis.

## 2 Background

This chapter will give a brief overview of the technical background, required components and implementation approaches to be used for developing the software application Partiture.

### 2.1 Targeted platform

The application will be initially developed upon Apple iOS SDK 7 and its core frameworks, running fully functional on iPads and iPad Minis. The minimal supported OS version will be iOS 7, which automatically means that the following device versions will be supported:

- iPad 2nd generation and newer versions
- iPad Mini 1st generation and newer versions

The decision for selecting Apple iOS as the targeted platform was related to the authors' previous experiences with it. Also other aspects, like the user interface responsiveness latency of different mobile platforms and device types [1] were considered.

### 2.2 Development tools

For developing purposes the native development tool (IDE) Xcode<sup>5</sup> (major version 5) running on the Mac OS X platform is used. For custom actions (e.g. prebuild, postbuild) UNIX-based scripting tools like Bash<sup>6</sup> can be used.

### 2.3 Key concepts

The core functionality – event and data exchange between the applications, will take place over the air, using the Multipeer Connectivity Framework, which will be outlined more detailed in sub-section 5.7. This technology supports the most important key aspects – connection speed and reliability, which are required for this real-time system to be efficient.

---

<sup>5</sup> <https://developer.apple.com/xcode/> (May, 2014)

<sup>6</sup> [http://en.wikipedia.org/wiki/Bash\\_\(Unix\\_shell\)](http://en.wikipedia.org/wiki/Bash_(Unix_shell)) (May, 2014)

## 2.4 External components

Since developing the whole application prototype from scratch would not fit into the thesis scope, some external frameworks / 3rd party components, like Julius Oklamcak's PDF Reader Core for iOS<sup>7</sup> are used.

---

<sup>7</sup> <https://github.com/vfr/Reader> (May, 2014)

## 3 Related work

### 3.1 Non-collaboration supporting solutions

There are quite many software applications available for public usage, which are already successfully replacing paper-based sheet music with digital sheet music for many categories of musicians. Some of these applications may require also special hardware, like SamePage Performance Stations<sup>8</sup>, but most of them are designed to work on more common and widespread platforms and devices – especially on tablet devices, in order to reach a larger number of potential users. All these applications include basic key features like:

- storing a large amount of scores (sheet music) on the device memory
- sheet music importing and exporting in PDF format
- organizing (e.g. grouping, filtering) and searching within the stored scores
- page turning using finger swipe and / or special tap gestures
- adding custom annotations, notes on the score (e.g. highlighting partitions) as additional layers

Particular applications may consider more specific features like:

- creating internal links (shortcuts) to jump from one location to another (e.g. NextPage<sup>9</sup> for iPad)
- page turning using specially designed foot pedals, operating over Bluetooth (e.g. AirTurn<sup>10</sup>)
- automatic page scrolling / turning, based on song length estimation (e.g. OnSong<sup>11</sup>)
- storing to / downloading from cloud services like Dropbox
- continuous score scrolling (from top to bottom)
- a built-in metronome (audiovisual) helping to keep the tempo
- instrument tuning assistants
- direct score editing / manipulating on MusicXML formatted files
- playback of recognizable score file formats like MusicXML, MIDI
- recording and playing back live sessions

---

<sup>8</sup> <http://www.samepagemusic.com/> (May, 2014)

<sup>9</sup> <http://www.onstagetechnologies.com/products/nextpage> (May, 2014)

<sup>10</sup> <http://airturn.com/> (May, 2014)

<sup>11</sup> <http://onsongapp.com/> (May, 2014)

- motion detection (eye movement) based page turning

One interesting bundle, which in concept (in an abstract way though) is more related to this thesis, is iGigBook<sup>12</sup> - consisting of the iGigBook Sheet Music Manager for displaying the scores on Apple iPad devices and iGigBook Pager for acting as a remote control for turning pages running on Apple iPhone / iPod Touch devices. The connection between the two applications is established over Bluetooth, which supports the instant data exchange aspect, but also adds restrictions like being operative within ~10 meters range.

A very small number of applications have also attempted to include artificial intelligence modules, in order to highlight the played measures and also trigger page turning events automatically in real-time, while the piece is performed. The intelligence lies in fast input (played music) processing, which is done using algorithms and methods like the Fast Fourier Transform (FFT) analysis [2] and finding the correct place in the score based on the analysis result. Since most of the musical pieces do incorporate recurrences and / or similar fragments (see Figure 2), the input might be noisy and / or invalid – finding the correct spot in real-time requires a lot of computation power and also machine learning support.

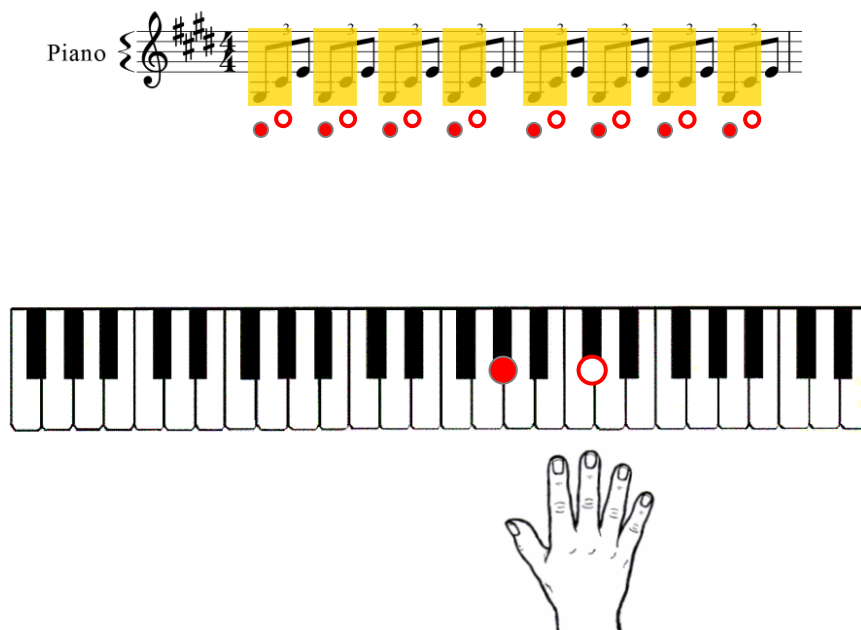


Figure 2. Recurrences in the right hand piano partition.

<sup>12</sup> <http://www.igigbook.com/> (May, 2014)

Also identifying the input for different kind of instruments can be difficult – percussive like instruments (e.g. the piano) create more concrete soundscapes / peaks than string like instruments (e.g. the violin) [3].

One such kind of a non-commercial / open-source project [4] was proposed by a Carleton College Computer Science Department professor David Liben-Nowell. It was started in 2009 and finished in 2010 by a group of students under the code name Piano Hero [5]. The targeted platform was Mac OS X, so it was more like a desktop application. As the name also states, the application was designed to work together with a piano. After performing smoke tests on the application, it was clear that it cannot be used for live performances, though some kind of input processing was done and it also highlighted the measures according to some logic. Nevertheless, looking into the source code and playing around with it helps to get more familiar with the domain and therefore it can be used as an input for further investigation into similar topics.

Another commercial application was first introduced [6] under the name Tonara<sup>13</sup> in November 2011. The targeted platform where it can be installed on is Apple's iOS (iPad). The application is available for free; the business model rather lies in selling sheet music via in-app purchases. As an extra feature it also allows to turn on functionality for real-time score following, which after performing some tests – trivial and more complex – turned out to be more accurate than for example above-mentioned Piano Hero.

There can also be found references for applications like Musebook Score<sup>14</sup> (later renamed to pdf-notes<sup>15</sup>), which initially adopted some score following mechanisms, but later did drop this feature.

Another type of score following applications, such as OrganMuse<sup>16</sup>, relies on MIDI input, which is provided by MIDI outputting devices. So the input is already processed and can directly be used for identifying the correct (currently played) spot in the score. Since most of the natural instruments do not have MIDI interfaces, this approach is feasible only for a small amount of special devices.

---

<sup>13</sup> <http://tonara.com/> (May, 2014)

<sup>14</sup> <http://mbscore.blogspot.com/> (May, 2014)

<sup>15</sup> <http://blog.musebook.com/> (May, 2014)

<sup>16</sup> <http://www.organmuse.com/> (May, 2014)

### 3.2 Collaboration supporting solutions

There are not many solutions available, which support collaborative music resources sharing and editing between musicians during practice and performance sessions.

One such kind of a solution was introduced in November 2012 by Brussels Philharmonica Orchestra, Samsung and a young ICT company named neoScores<sup>17</sup> – an application, which is claimed to be a starting point for replacing paper-based sheet music with digital sheet music. Due to the limited amount of technical / implementation details, business logic<sup>18</sup>, and premature reviews [7] (which most probably indicates that the application will be commercialized), only a few assumptions can be concluded:

- since it is platform independent, it cannot adopt (many) specific platform / native features
- since it is web-based, there has to exist a web-server (most probably accessible over the internet) handling the data exchange part

According to neoScores official website, the go-live release was planned for March 2014. As of 22<sup>nd</sup> of May 2014, only a beta version has been made publicly available, which cannot be used for an extensive evaluation and analysis.

In December 2013 another Belgium company called Scora<sup>19</sup> was introducing their system together with the Leuven Alumni Orchestra during a 3-day concert series to nearly 2000 spectators. According to the available information on their website<sup>20</sup>, the main focus is on creating software and custom hardware bundles for professional orchestras, which means that procuring the system for smaller collectives (e.g. instrumental or vocal ensembles) or music schools might be either impossible or very expensive. From the functionality perspective, Scora is relatively similar to the solution under development in this thesis (Partiture) by implementing key features like:

- sending starting (retake) positions from the conductors device to the members of the orchestra
- creating and sharing annotations (remarks) with the members of the orchestra
- turning pages remotely for the members of the orchestra

---

<sup>17</sup> <http://www.neoscores.com/> (May, 2014)

<sup>18</sup> <http://demo.neoscores.com/about> (November, 2013)

<sup>19</sup> <http://scora.net/> (May, 2014)

<sup>20</sup> <http://scora.net/about.htm> (May, 2014)



Based on the application screenshots and provided technical information, the following statements can be concluded:

- the devices' operating system is Android
- the hardware (devices) is specially designed (e.g. conductor console)
- the synchronization between the tablets and conductors console works over a content server
- Wi-Fi technology is used for communication between the devices

Another similar solution, which supports collaborative music editing, is SamePage<sup>21</sup>, though it is designed to rather support composing music among larger groups. Theoretically it should also be usable for practices and performance sessions, but as a client-server like application, it requires for operating an internet connection, because it stores the data and handles synchronizing the changes via external services.

### **3.3 Other indirectly similar solutions**

Multipeer Connectivity Framework based technology is adopted for multiple Apple / Mac OS related applications and services like:

- AirDrop<sup>22</sup>, which supports file sharing over the air between nearby devices

and also for other applications like:

- FireChat [8]

---

<sup>21</sup> <http://www.samepagemusic.com/> (May, 2014)

<sup>22</sup> [http://ipad.about.com/od/iPad\\_Guide/ss/What-Is-Airdrop-How-Does-It-Work.htm](http://ipad.about.com/od/iPad_Guide/ss/What-Is-Airdrop-How-Does-It-Work.htm) (May, 2014)

## 4 Requirements analysis

The business level requirements as well as functional and non-functional level requirements for the system were defined and elaborated in cooperation with Tartu Heino Eller Music School teaching staff and students (a dedicated interest group).

For gathering the requirements, several methodologies, such as

- visiting lessons (orchestra rehearsals, conductor classes) – aim to observe and study the musicians' daily based workflows and processes
- group interviews – aim to discuss different approaches, generate brainstorm and evaluate upcoming ideas
- private interviews – aim to concentrate on specific needs of different musicians (e.g. percussion instructor)
- prototype presentations – aim to introduce the concept and collect impressions and improvement ideas
- chat and email discussions – aim to communicate sudden ideas instantly

were applied and combined.

Considering the characteristics of the domain, the agility of the project and the timeline, it was decided to outline the requirements in the form of user stories [9]. This approach is also more flexible in terms of changing the requirements more frequently. Furthermore, it allows specifying the needs for the system in the procurer's everyday language. The completeness of the functional requirements (user stories) will be evaluated by comparing the implementation against the corresponding acceptance criteria, defined for the implementation scope of this thesis. All user stories, which did not fit into the scope, are postponed as future work explicitly.

## 4.1 User stories

### Business requirements

Table 1. Business requirements related user stories.

ID	As a(n) <type of user>	I want (need) <some goal>	so that <reason>	Originator	Prio.
BR1	Conductor, Sound engineer	the musician(s) to turn the score pages silently	the rustling noise would not disturb the performances or studio recordings	[name hidden] (Eller)	1
BR2	Musician	to delegate the task of page turning to someone (or something) else	the practice or performance session could run continuously without interruptions	(Eller)	1
BR3	Musician	to purchase or sell remarks (instructions, comments)	work could be organized more efficiently	[name hidden] (Eller)	3
BR4	Guitar teacher	to compile and share score collections	work could be organized more efficiently	[name hidden] (Eller)	3
BR5	Conductor	to distribute scores via a centralized system	further (uncontrolled) distribution (e.g. photocopying) of the scores would not be possible	(Eller, UT)	1
BR6	Musician	the digital score to adopt the characteristics of paper score as closely as possible	the reading angle and light (artificial light, sunlight) reflection would not disturb the practice or performance in any way	(Eller, UT)	1
BR7	Guitar student	share the device (application) with others (e.g. family)	the hardware related costs could be	(Eller, UT)	1

		<b>members)</b>	<b>reduced (splitted)</b>		
--	--	-----------------	-------------------------------	--	--

## Functional requirements

Table 2. Functional requirements related user stories.

ID	As a(n) <type of user>	I want (need) <some goal>	so that <reason>	Originator	Prio.
FR1	Performing artist	to restore the latest state of the application after a failure incident by just opening it again with a single tap on the application icon	I would not have to start searching for the correct score, entering the room, restoring the session manually	(Eller, UT)	1
FR2	Musician	to be able to sign in to and sign out from the application on demand	my scores, rooms, remarks would not be visible to other users of the application	(Eller, UT)	1
FR3	Musician	to see my downloadable and downloaded scores in separated lists	I could quickly differentiate among them	(Eller, UT)	2
FR4	Musician	to be able to search for a particular score by entering the title, composer, lyricists or arranger (partial) value as the keyword	the score could be found faster	(Eller, UT)	2
FR5	Music teacher	to be able to define the desired room name	me and my students could easily find it in the list of available rooms	(Eller, UT)	1
FR6	Music teacher	to be able to define a passkey for the room	uninvited guests could not join it	(Eller, UT)	1
FR7	Musician	to see my created, my joined and not joined rooms in separated lists	I could quickly differentiate among them	(Eller, UT)	2
FR8	Musician	to be able to search for a particular room by entering the name (partial) value as the keyword	the room could be found faster	(Eller, UT)	2

FR9	Musician	to be able to navigate through the score pages with finger swipe gestures	the desired page could be quickly found	(Eller, UT)	1
FR10	Musician	to be able to navigate through the score pages using a quick menu	the desired page could be quickly found without scrolling through all the pages	(Eller, UT)	1
FR11	Conductor	to pinpoint the exact retake location on a score page with tapping on it for the whole collective	extra time does not have to be wasted on finding the spot	(Eller, UT)	1
FR12	Conductor	to turn the pages for the whole collective with a finger swipe on the screen or selection in the quick menu	there would not be any interruptions in the music for some members because of page turning task	(Eller, UT)	1
FR13	Musician	to be able to use a room without making it publicly visible for nearby users / members	I could prepare the room with some remarks first	(Eller, UT)	2
FR14	Music teacher	to be able to create private remarks (visible only for me) as well as public remarks (visible for all members in the room)	the students' scores would not be buried under too much remarks / information	(Eller, UT)	1
FR15	Musician	to be able to add remarks with finger movement gestures on the score	it could be done fast and efficiently	(Eller, UT)	1
FR16	Musician	to be able to move or delete my created remarks on the score	the scores wouldn't contain excessive information or duplications	[name hidden] (Eller)	1

## Non-functional requirements

Table 3. Non-functional requirements related user stories.

ID	As a(n) <type of user>	I want (need) <some goal>	so that <reason>	Originator	Prio.
NFR1	Organist	to layout the scores in both orientations (portrait, landscape)	a larger number of available scores could be used with the system	[name hidden] (Eller)	1
NFR2	Organist	to turn the score pages using also other body parts (gestures) than fingers	the practice or performance session could run continuously without interruptions	(Eller)	3
NFR3	Guitar teacher	to distribute remarks to as many users as possible	work could be organized more efficiently	[name hidden] (Eller)	2
NFR4	Musician	to use different sized scores	they would be observable in different occasions	(Eller)	1
NFR5	Musician	to get an analysis (statistics) after the practice or performance session	the improvement or regression could be measured and the mistakes identified; the result could be compared with others	(Eller)	2
NFR6	Clarinetist	to see the rhythm and pitch deviations compared with the original notation	the mistakes could be identified	[name hidden] (Eller)	2
NFR7	Musician	to use PDF scores	the existing files do not have to be converted to system specific formats	(Eller, UT)	1
NFR8	Conductor	to distribute scores to a certain group of	other parties could not	(Eller, UT)	1

		<b>choir members</b>	<b>download them</b>		
<b>NFR9</b>	<b>Musician</b>	<b>to prepare the score for using it with the system online</b>	<b>I would not have to install additional software</b>	<b>(Eller, UT)</b>	<b>1</b>



## 4.2 Implementation scope

Table 4. Implementation scope with corresponding acceptance criterions.

Requirement ID	Acceptance criterion	Accepted	Comment
FR1	After the application has been terminated from the task manager (has the same influence on the application as a crash), while the user was in a room, the same room should become visible automatically after re-launching it.	YES	
FR2	The user can enter his / her e-mail and password for signing in; The user can sign out himself / herself under the application embedded settings;	YES	
FR3	The downloadable scores and downloaded scores will be displayed in the same list, but in different sections	YES	
FR4	The keyword matching results (list items) will be indicated with a yellow highlight color	YES	
FR5	The user can enter an alphanumeric room name when creating it	YES	
FR6	The user can enter an alphanumeric passkey for the room when creating it	YES	
FR7	The user created, joined and not joined rooms will be displayed in the same list, but in different sections	YES	
FR8	The keyword matching results (list items) will be indicated with a yellow highlight color	YES	
FR9	The user can use left / right swipe gestures for navigating through the score	YES	
FR10	The user can use a quick preview menu at the bottom showing thumb images of the score pages	YES	
FR11	The room owner can tap on the	PARTIAL	The joined room

	<b>desired location in the score, which will be highlighted for the joined room members as well</b>		<b>member has to be on the same page in order to see the pinpoint event</b>
<b>FR12</b>	<b>The room owner controls the page turning for all the joined members of the room (participating members in the room session)</b>	<b>YES</b>	
<b>FR13</b>	<b>The room can be entered without making it visible. In order to start publishing a session in the room a special button on the top toolbar of the workspace view has to be tapped first.</b>	<b>YES</b>	
<b>FR14</b>	<b>The layers (private and public) can be easily switched by selecting between corresponding buttons in the top toolbar of the workspace view</b>	<b>YES</b>	
<b>FR15</b>	<b>The remarks can be added in the editing mode by dragging finger gesture movements on the score</b>	<b>PARTIAL</b>	<b>Only creating horizontal highlights are currently supported</b>
<b>FR16</b>	<b>The own created remarks can be moved or deleted by pressing for 1 second on it</b>	<b>YES</b>	

## 5 Architecture

This chapter describes the design and architecture of the system Partiture, including the relations and dependencies of its components, using the 4+1 architectural view model [10] with some adjustments. Additionally, some technological choices as well as alternative approaches are briefly discussed and compared.

### 5.1 Overview

In general, the system consists of two major key components:

- **web application**<sup>23</sup> – which is needed for providing a centralized user and score management system
- **mobile application** – providing the functionality for working with the downloaded scores

The functional behaviour of the web application has not been fixed yet and will be therefore considered as a „black box“ in the following descriptions.

The functional behavior of the mobile application can be roughly divided into two operating modes:

- **preparation mode** – the mobile application requires a communication link (e.g. over the internet) with the web application (client-server model behaviour), in order to be able to authenticate (if required) and to download accessible scores, which are needed to enter the work mode
- **work mode** – the mobile application can be used independently or in collaboration with other instances of the application (peer-to-peer model behaviour)

One of the key points, which was considered when designing the architecture, is the distribution of the databases and data synchronization between the mobile applications with minimal conflicts. Since there is no central server involved for the main workflow operations (except the user registration, online authentication and score preparation, sharing management), each transferable object is created with a universally unique identifier [11] (hereinafter UUID). The universally unique identifiers, created and managed by the web application are considered as tokens in the mobile application environments.

---

<sup>23</sup> Within the scope of this thesis only a (static) mockup of the web application interface, providing the basic functionality for authentication and score downloading, is implemented.

## 5.2 Logical view

The logical view of the mobile application is outlined by a class diagram. The class names, their corresponding roles and relations (see Figure 4) were designated while bearing in mind an example music lesson (for instance, orchestra rehearsal).

- **User** – an application user, who can log in to the system and take advantage of its functionalities; there can be several users (related BR7) on one device, but only one logged in at a time
- **Score** – an abstract representation of a set of musical sheets, which has been downloaded on to the device by an user
- **Bundle** – shows the relation between the user and the downloaded score(s)
- **Composer** – associated with the corresponding score(s)
- **Lyricist** – associated with the corresponding score(s)
- **Arranger** – associated with the corresponding score(s)
- **Member** – an user (local or remote), who has joined a room
- **Membership** – shows the status and role of the member in a room
- **Room** – a (virtual) room binds one concrete score with the member(s) of it; every time a room is accessed (opened) a new session in the room is started
- **Participation** – shows the participation time of the member in the created session
- **Session** – with each session, one private and one public layer is created
- **Layer** – holds the remarks made by a participant during the session
- **Remark** – an annotation / comment, which was created on a layer by a participating member in the session

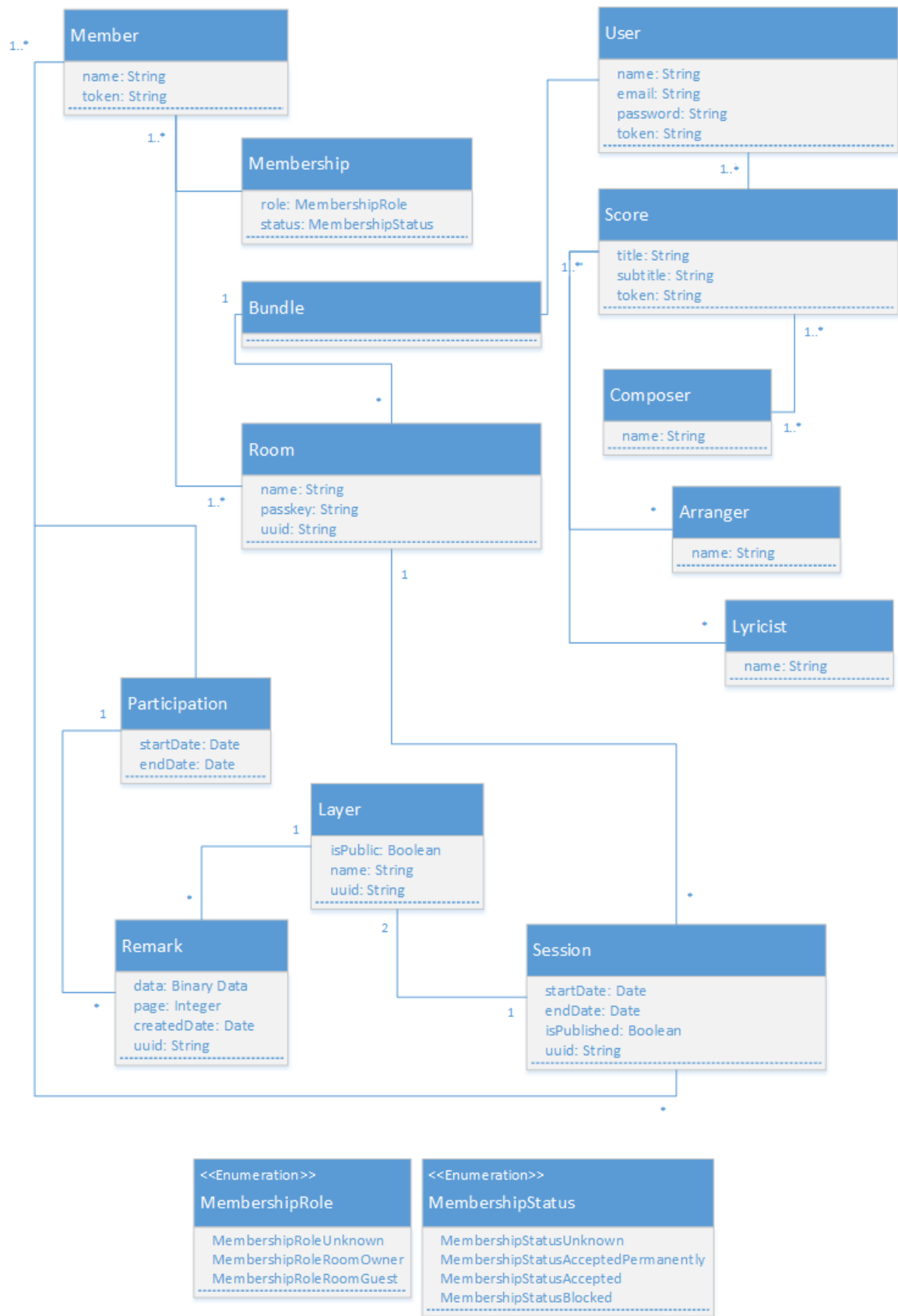


Figure 4. Mobile application class diagram.

### 5.3 Development view

The development view is outlined using a component diagram of the system, identifying the most important components and their relations (see Figure 5).

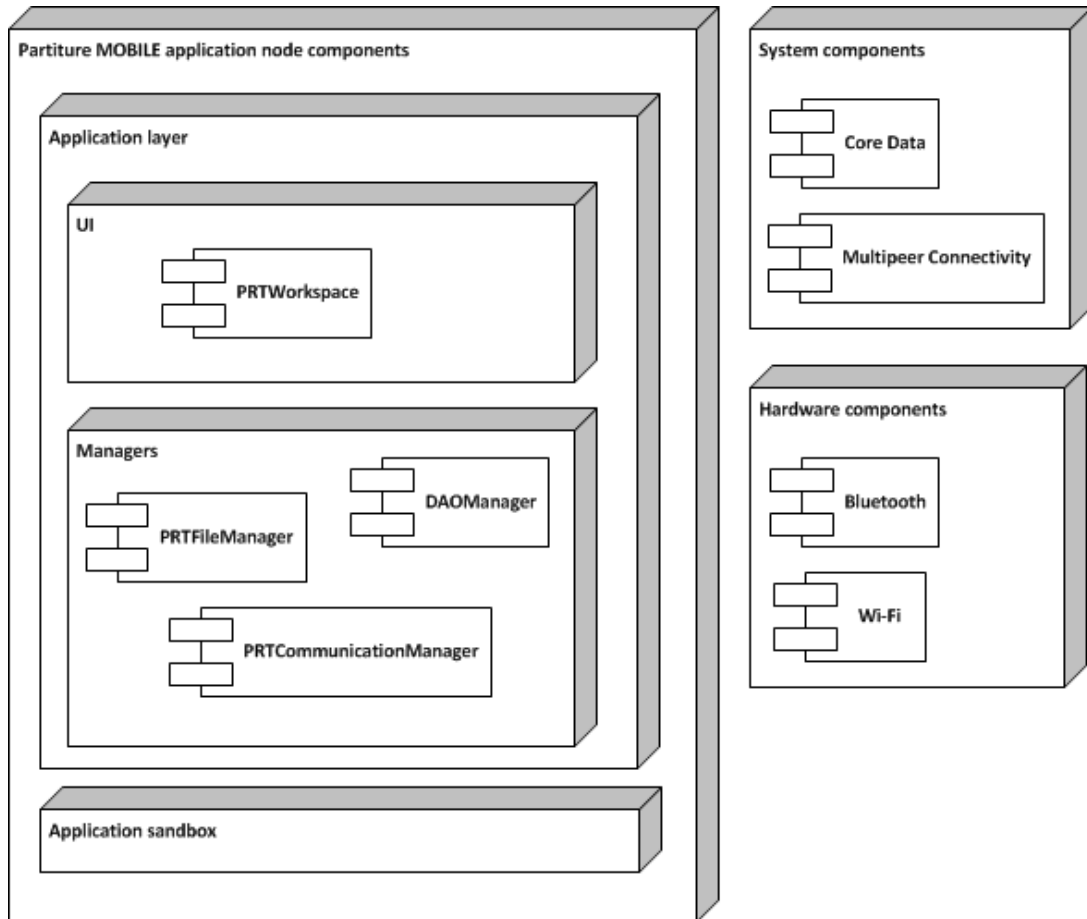


Figure 5. Mobile application component diagram.

## 5.4 Process view

The most important and non-trivial processes in the system are outlined using UML activity diagrams. More trivial processes are elaborated using textual explanations.

### Restoring application state

On (re)launch, the application first tries to restore the latest session (published or unpublished) in a room (related FR1) based on the active participations stored in the database. In case the application is started without an active user or no active participations were found for the active user, alternative actions will be triggered correspondingly (see Figure 6).

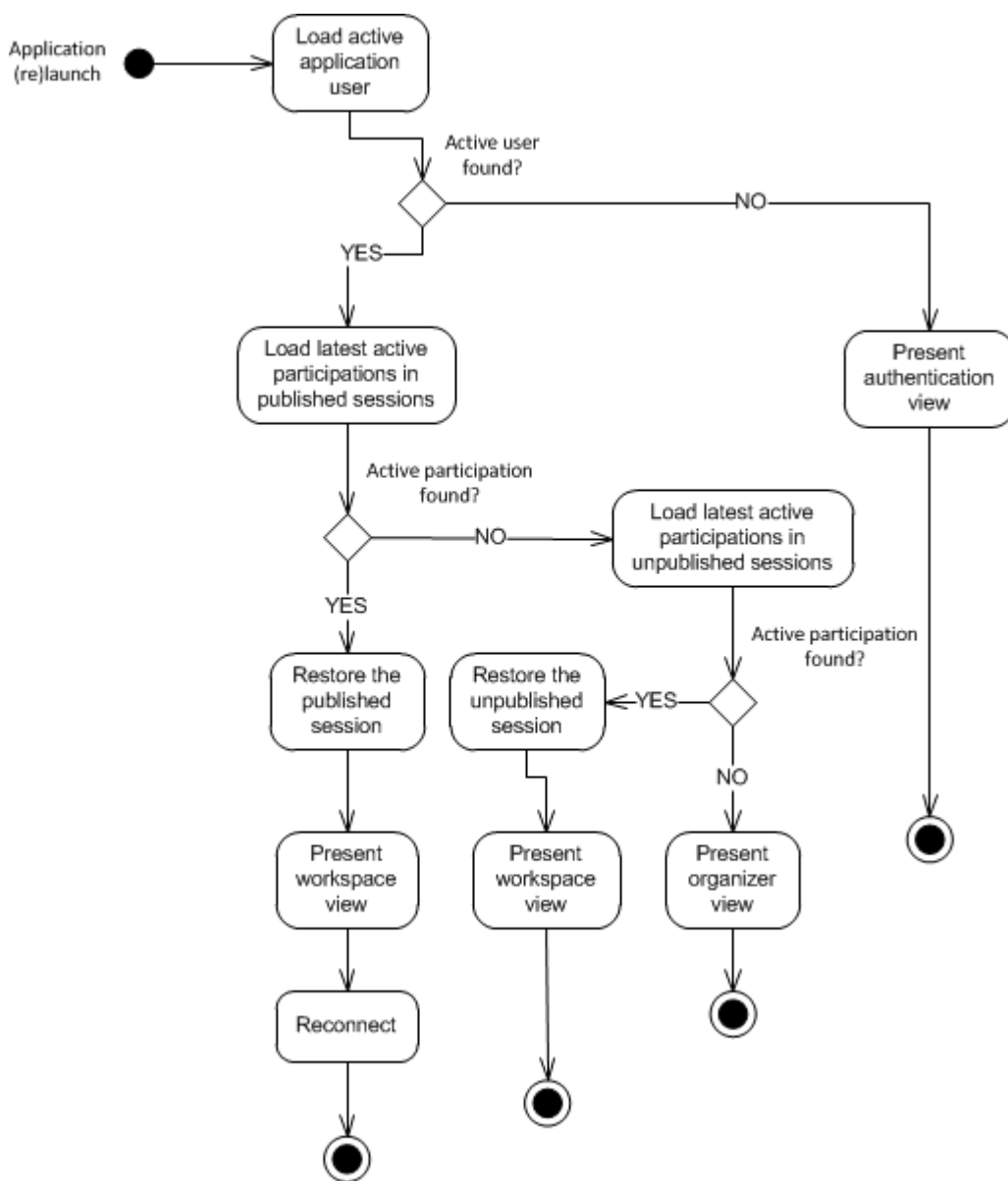


Figure 6. Restoring application state process diagram.

## Authenticating user

In order to be able to use the applications features (incl. downloading scores), the user first has to authenticate himself / herself. When the user information is not stored in the database (e.g. the user authenticates the first time on the particular device) an online authentication process is triggered for acquiring the user token via the web application API (see Figure 7).

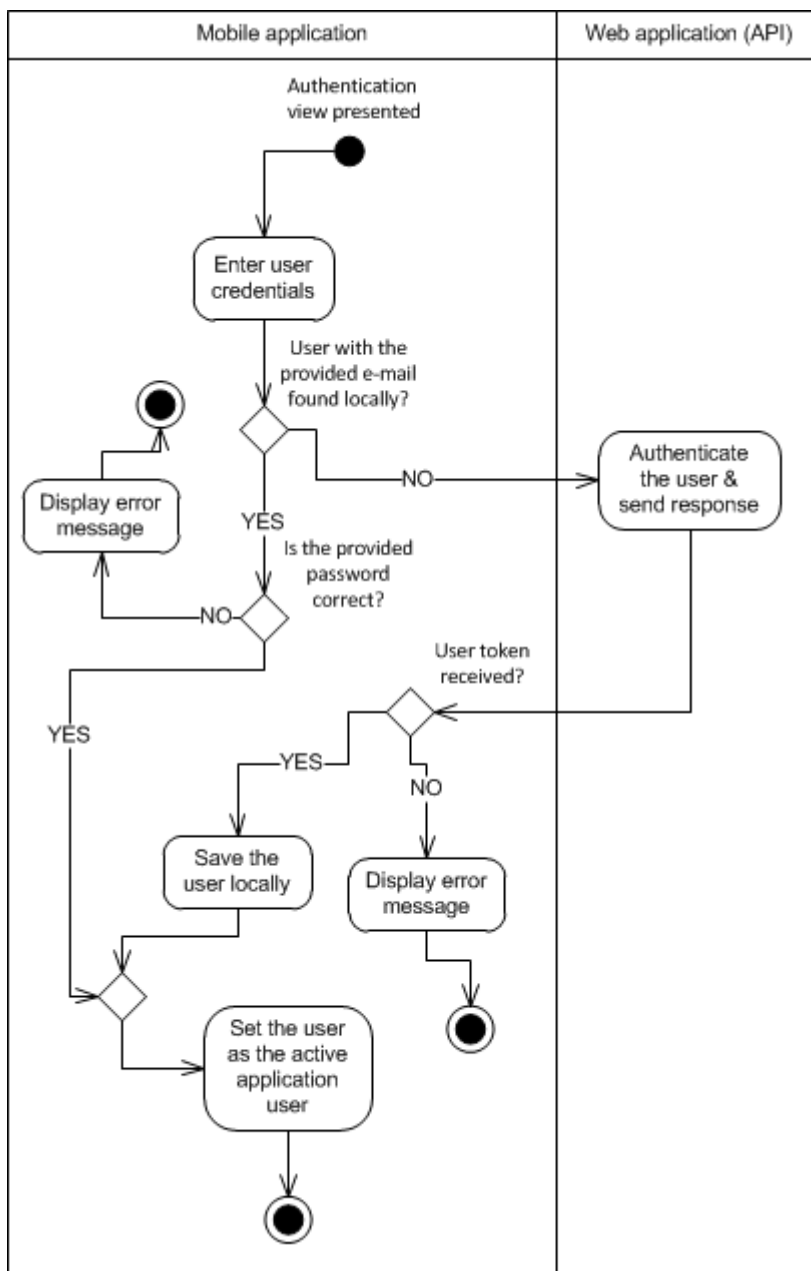


Figure 7. Authentication process diagram.



### **Downloading scores**

When the organizer view is presented, the mobile application asks from the web application API for the list of downloadable scores using the active user token. The list will be composed from the scores, which have been made available for this particular user and can be used for triggering the score download(s).

### **Building up the (re)connection**

The (re)connection process has the same behaviour on establishing the initial connection as well as on post-failure (e.g. connection drop, application crash) reconnecting (see Figure 8).

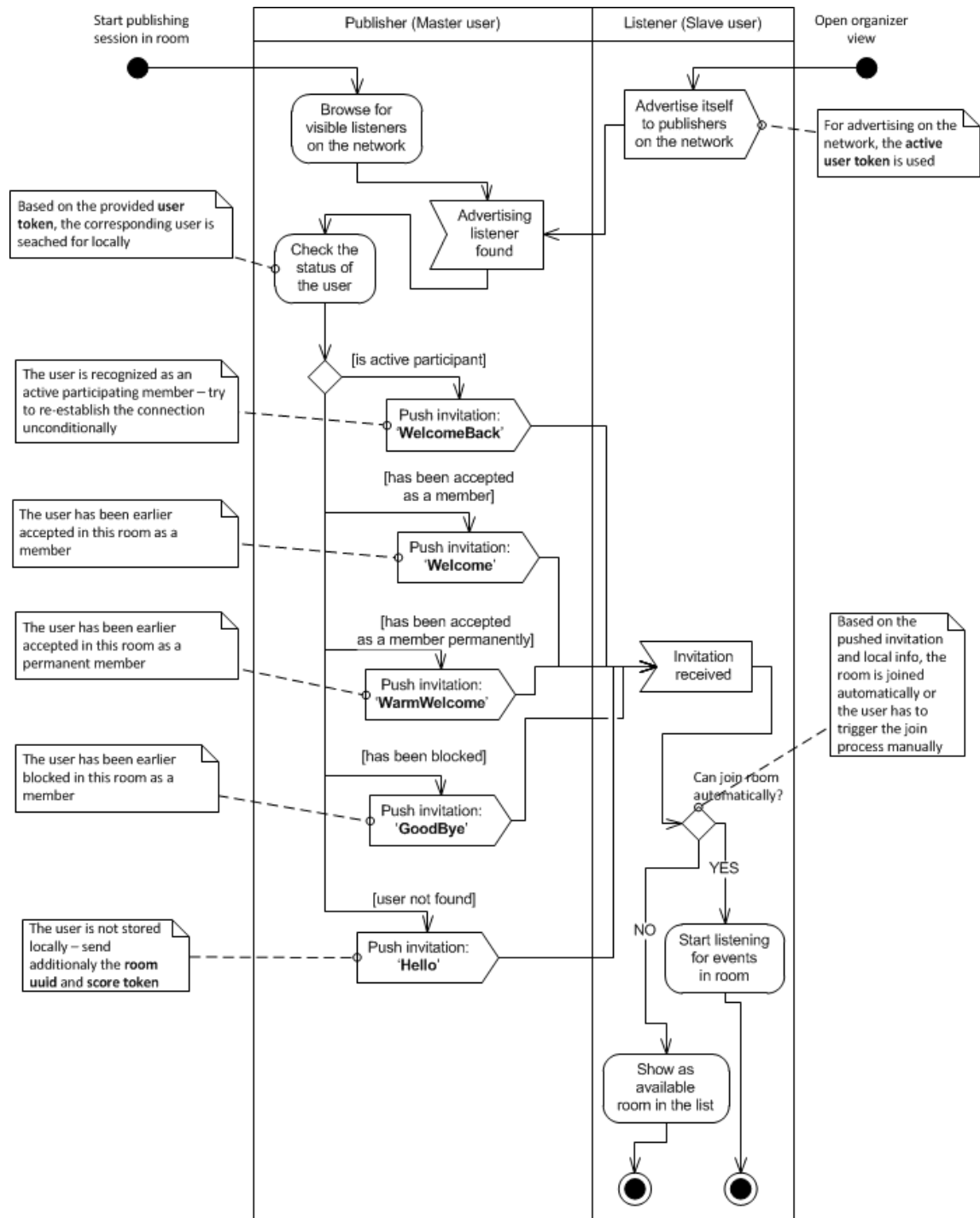


Figure 8. Building up the (re)connection.

## 5.5 Physical view

The physical view is outlined using UML deployment diagrams and considering the two main operating modes – preparation mode (see Figure 9) and work mode (see Figure 10).

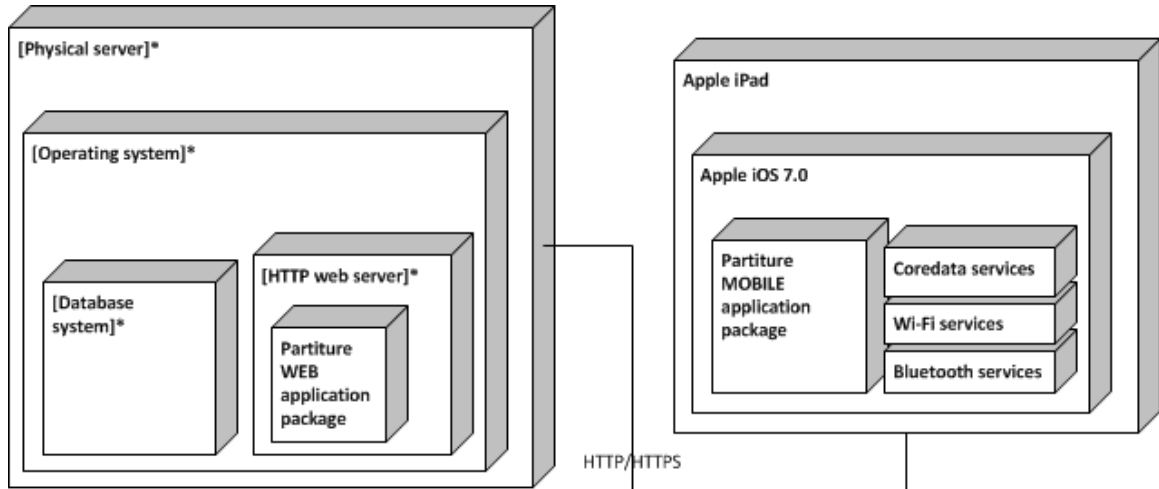


Figure 9. Preparation mode related deployment diagram.

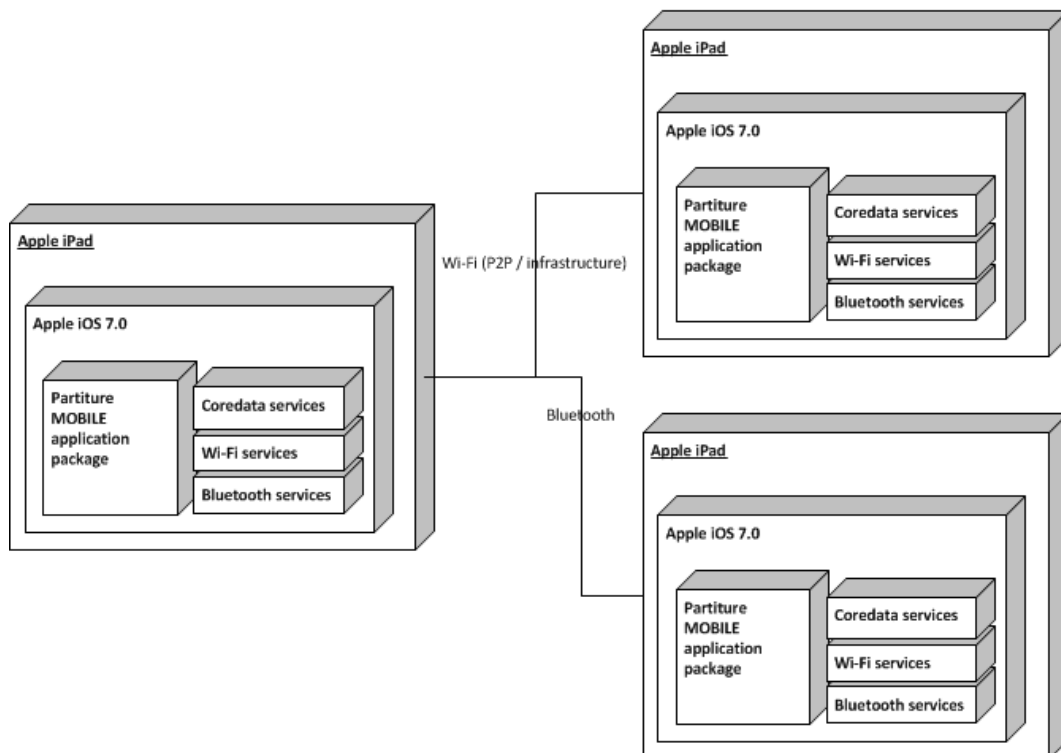


Figure 10. Work mode related deployment diagram.

## 5.6 Scenario view

The scenario view is outlined using a combination of use case diagrams describing the web application main functions (see Figure 11) and mobile application main functions (see Figure 12). Additionally a set of storyboards (see Appendix II), which explain with some concrete examples the main functions of the mobile application, illustrated by some graphical user interface views and object diagram snapshots (pre and post states). The storyboards also serve as user guide for the mobile application.

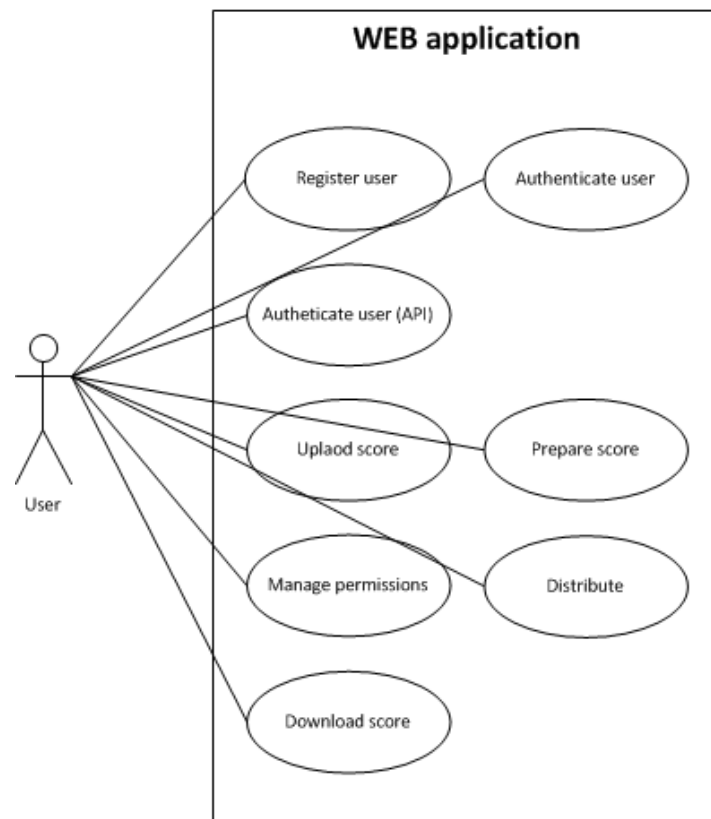


Figure 11. Web application use case diagram

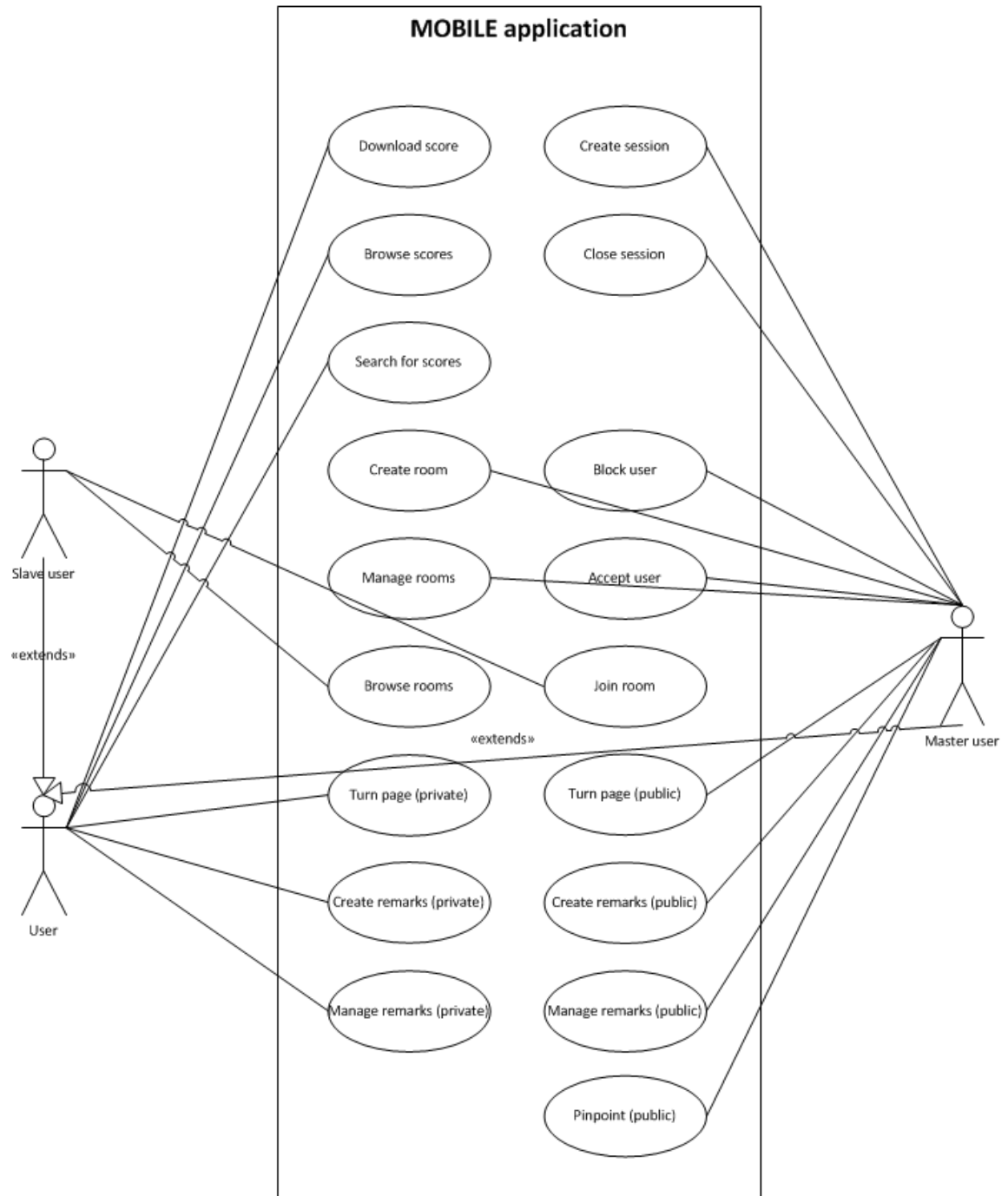


Figure 12. Mobile application use case diagram

## 5.7 Communication mechanisms

### Current implementation

The communication mechanism underneath is based on Apple's Multipeer Connectivity Framework (hereinafter MCF)<sup>24</sup>, which was first introduced with the release of iOS7<sup>25</sup> - a recent major operating system update for iDevices<sup>26</sup>. The MCF provides a set of methods, encapsulated into three main classes, for making peers discoverable, discovering available peers, establishing a connection and transferring data between the peers. Three possible schemes can be used for these tasks:

- Bluetooth Personal Area Network (hereinafter Bluetooth PAN) – this scheme uses the device's capability to set up Bluetooth based wireless networks. Although the Bluetooth standard 4.0, also known as Bluetooth Smart or Bluetooth Low Energy (BLE), is claimed to operate with less power consumption [12] than the previous standards, the effective working range for Bluetooth Class 2 devices remains up to 10 meters [13].
- peer-to-peer Wi-Fi – this scheme uses the device's capability to create nearby Wi-Fi networks on an ad-hoc basis.
- infrastructure Wi-Fi – this scheme requires the presence of at least one Access Point, which the devices can connect to.

According to Apple, the MCF automatically selects the most optimal scheme based on the conditions and available (enabled) interfaces[14]. In some cases, multiple technologies might be used in a cooperative way simultaneously, such as Bluetooth for handling the discovery and establishing a connection between the peers and Wi-Fi for sending and receiving data between the peers. The details of the previously described logic have not been made public by Apple.

---

<sup>24</sup><https://developer.apple.com/library/ios/documentation/MultiPeerConnectivity/Reference/MultiPeerConnectivityFramework/Introduction/Introduction.html> (May, 2014)

<sup>25</sup><https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/Articles/iOS7.html> (May, 2014)

<sup>26</sup><http://en.wikipedia.org/wiki/iDevice> (May, 2014)

## **Advantages**

- multiple schemes for data transmission – which means that fallbacks can be created
- no prior configuration is needed – which means that the nodes are ready to connect without defining IP addresses or ports

## **Disadvantages**

- iOS specific – which means that the system cannot be easily integrated with other platforms

## **Alternative solutions**

### ***Remote server based***

In the remote server based solution a service (e.g. running in a cloud) would be responsible for receiving and transmitting data packages to interested parties.

## **Advantages**

- the communication will be HTTP based – which means that the system can be adopted to a wider range of different platforms (e.g. Android, Windows)

## **Disadvantages**

- requires a persistent (internet) connection to the remote server (service) for operating – which means that the system can be used only in locations with an available link
- higher number of external dependencies (theoretical bottlenecks) – which means that the system does not rely only on the connection provider (ISP), but also on the service provider (host)
- the communication will be only HTTP based – which means that other channels (e.g. Bluetooth) cannot be used for data transmission

### ***Local server based***

In the local server based solution a service (e.g. running on a separate node or on an existing application node) would be responsible for receiving and transmitting data packages to interested parties.

#### **Advantages**

- the communication will be HTTP based – which means that the system can be adopted to a wider range of different platforms (e.g. Android, Windows)

#### **Disadvantages**

- the communication will be only HTTP based – which means that other channels (e.g. Bluetooth) cannot be used for data transmission (if not implemented additionally)



## 6 Validation

This chapter describes the validation results of the functional tests for the application logic and performance tests for one of its sub-components.

### 6.1 Functionality testing

As for testing the application functionality, mainly the white-box testing [15] method was used. Additionally the implemented user stories were (are) validated by comparing the implementation against the corresponding acceptance criterions (see sub-section 4.2).

### 6.2 Performance testing

There are many aspects, environment related conditions and external influences (e.g. electromagnetic interference [16]), which can affect the data transmission speed and reliability between the peers. In order to identify some of the possible bottlenecks and find the optimal configuration, a number of different tests were conducted using a simplified version of the mobile application.

#### Test data

The most important notification in the system, which has to be transmitted from one peer (mobile application instance) to another as fast and reliably as possible, is the page turning notification. Therefore a test data package (see Figure 13) with similar characteristics (structure, size) was also used for running the performance tests.

```
{  
  "id": " bf5b1ea6-2526-46aa-991c-0db34d5969d1", <- UUID of the package  
  "index": "25" <- number of the page  
}
```

Figure 13. Sample performance test data package.

## **Test method**

Since it is nearly impossible to synchronize the test devices clocks with an accuracy of milliseconds, the time difference of sending the test package from the master application instance (peer) to a slave application instance (peer) and back is measured by registering the timestamps. Dividing the result by 2 will give the approximate average latency of transmitting the package in one direction only. The method concept will be illustrated on a high-level basis with the following BPMN model (see Figure 14).

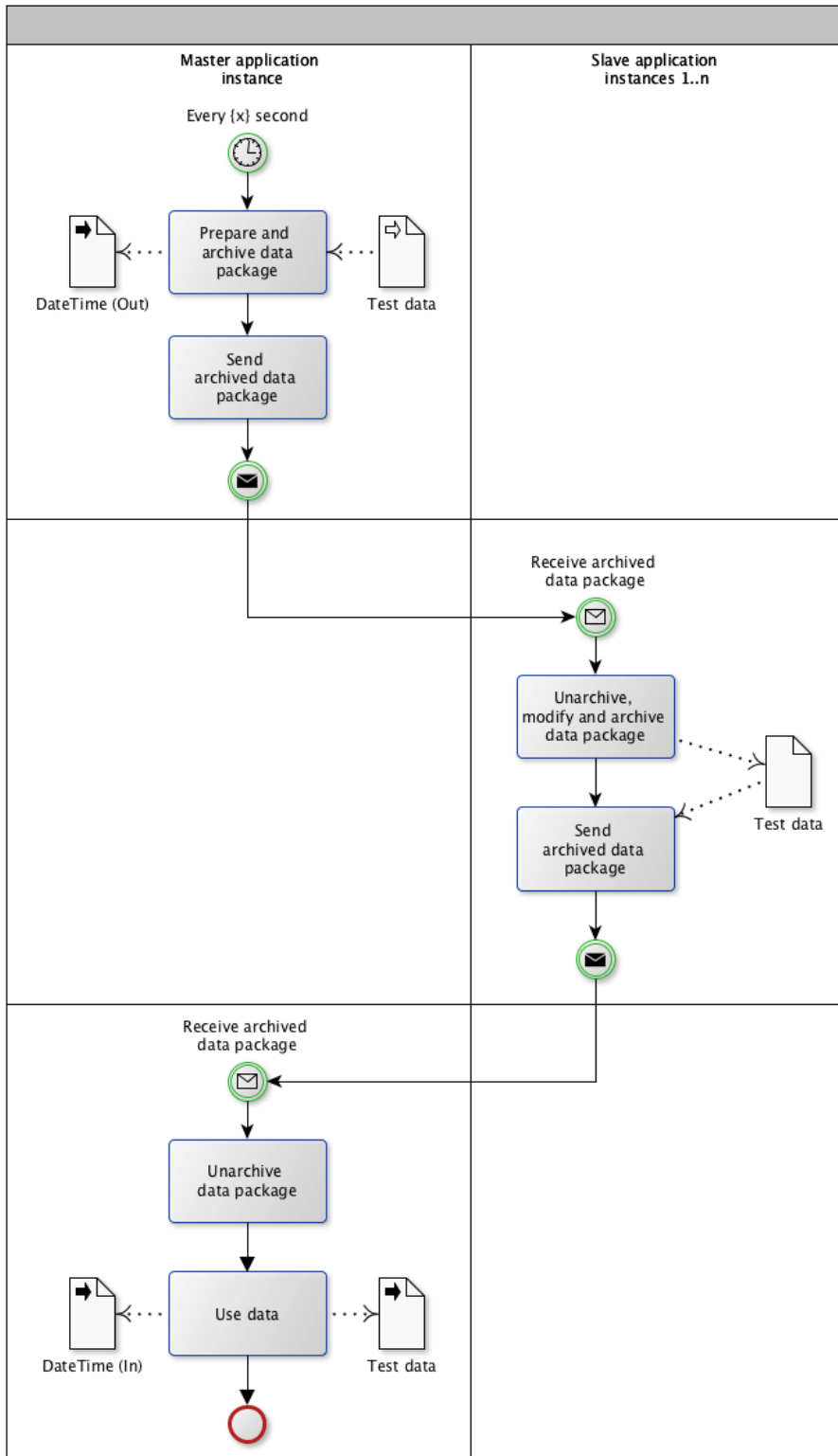


Figure 14. Test method process description (BPMN model).

## Test devices

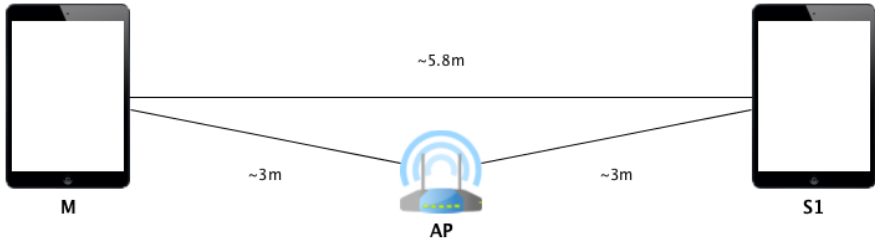
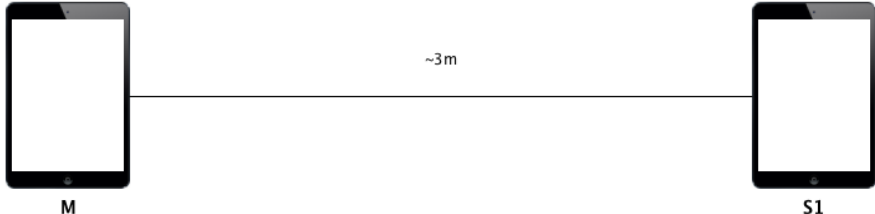
The following devices and configurations were used for testing.

- **AP** – UniFI UAP-Pro Access Point<sup>27</sup> / auto channel selection enabled; password protected AES 128bit
- **M** (master application instance) – iPad Air (5<sup>th</sup> generation iPad) / iOS 7.0.4
- **S1** (slave application instance) – iPad Air (5<sup>th</sup> generation iPad) / iOS 7.0.4
- **S2** – iPad 3 / iOS 7.0.3
- **S3** – iPhone 5 / iOS 7.0.4
- **S4** – iPhone 5 / iOS 7.0.4
- **S5** – iPad 4 / iOS 7.0.3
- **S6** – iPhone 5S / iOS 7.0.3

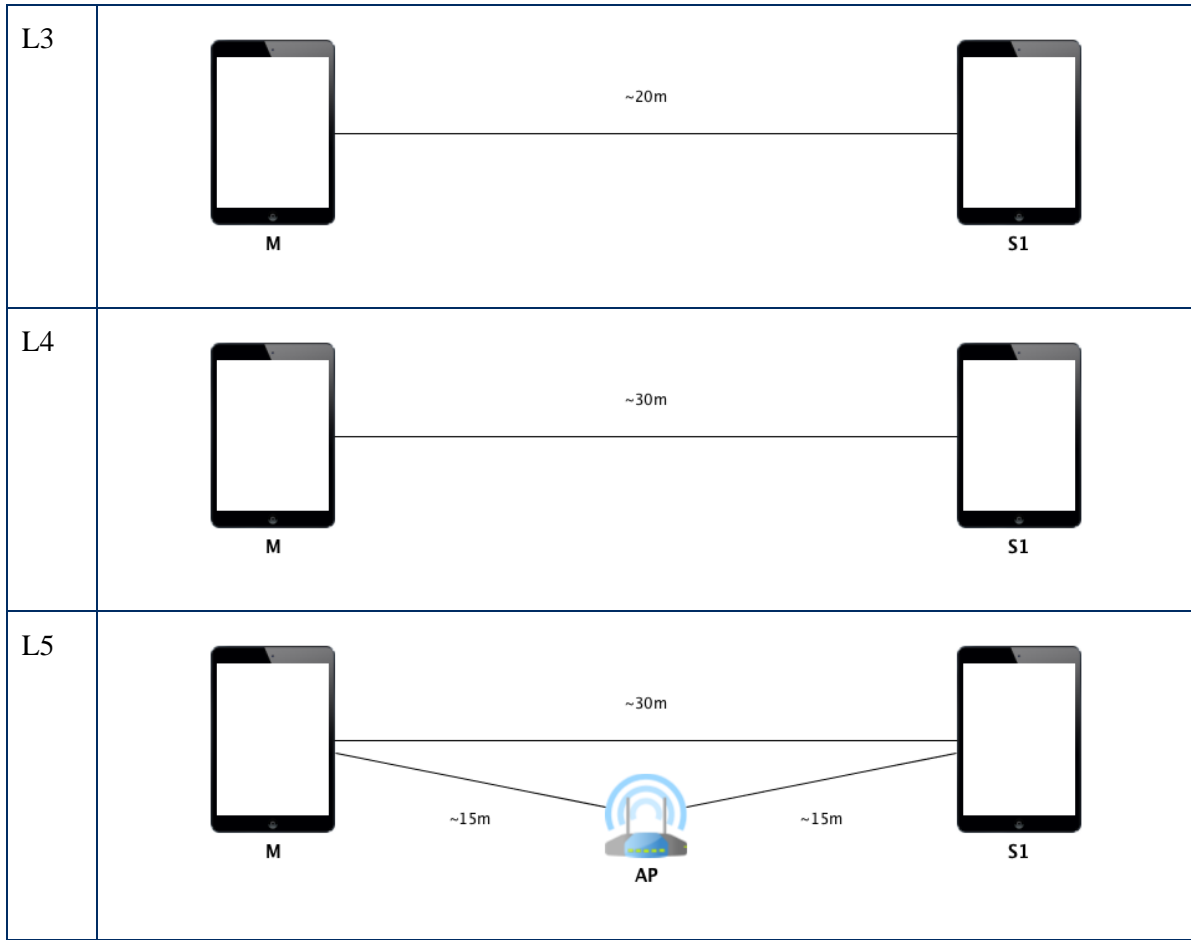
## Test layouts

The following layouts (see Table 5) were used during performing the tests.

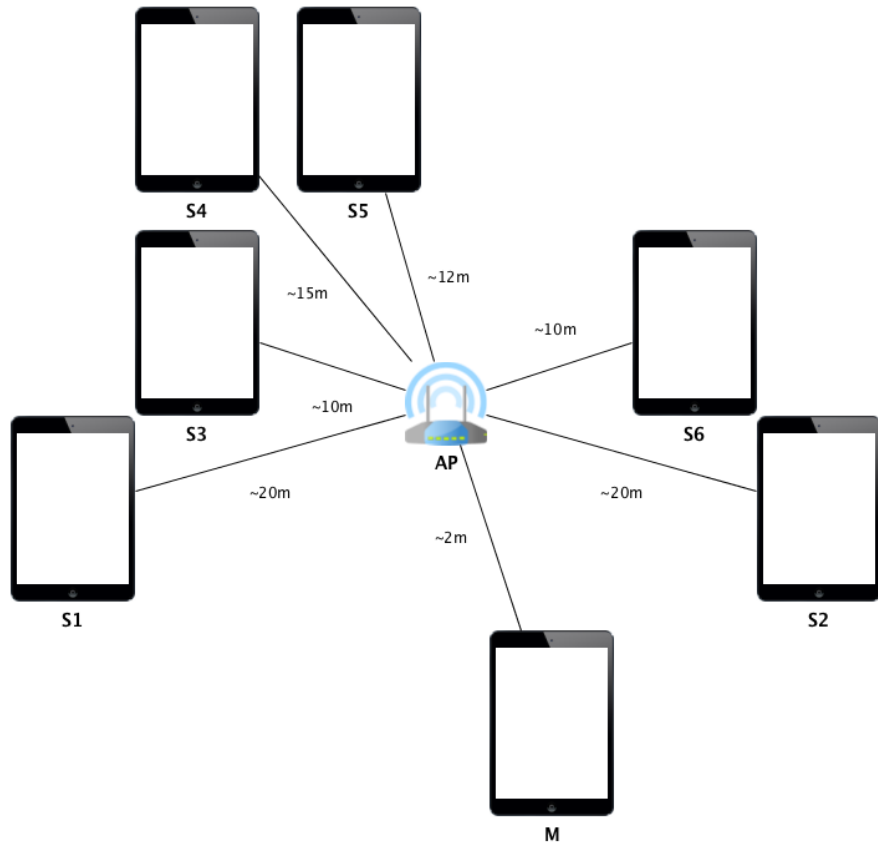
Table 5. Device physical placements for conducting performance tests.

ID	Layout
L1	
L2	

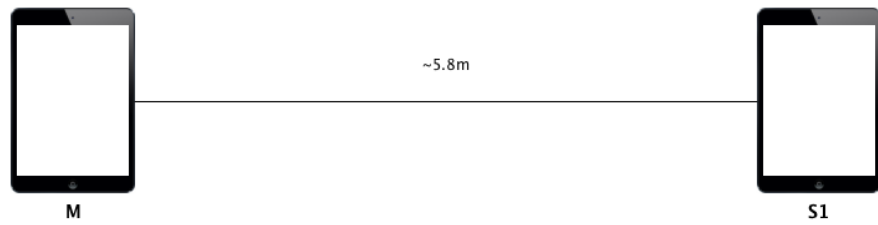
<sup>27</sup> [http://dl.ubnt.com/datasheets/unifi/UniFi\\_AP\\_DS.pdf](http://dl.ubnt.com/datasheets/unifi/UniFi_AP_DS.pdf) (May, 2014)



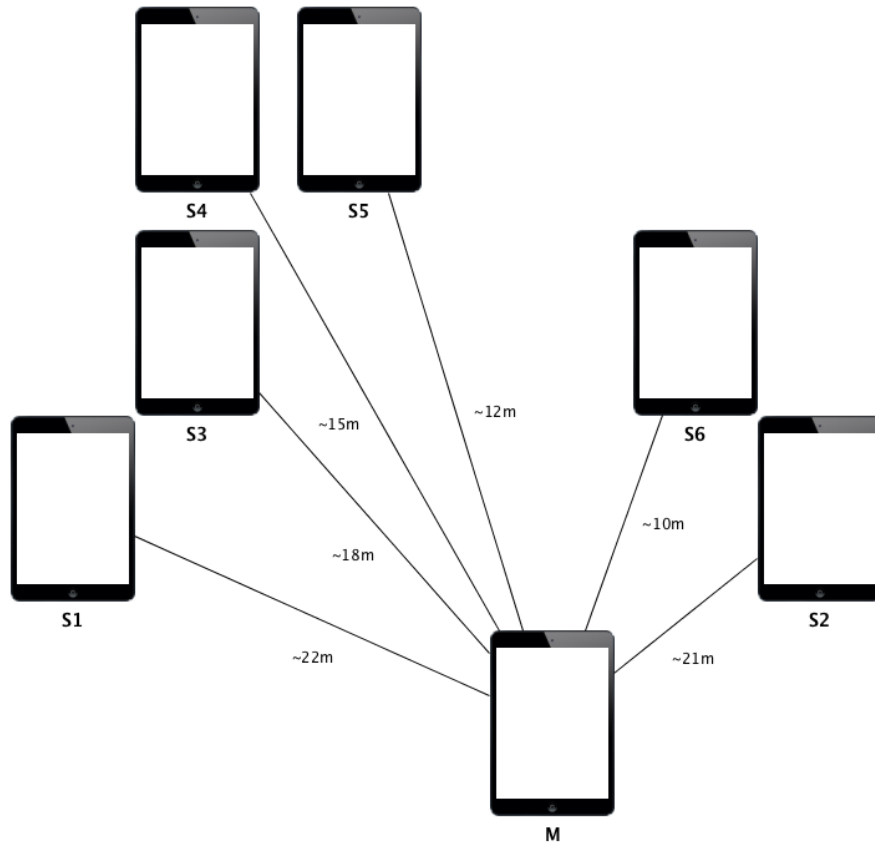
L6



L7



L8



### Legend




























- **Test ID** – identity of the test
- **Layout ID** – used physical layout
- **Slave device ID** – slave device identity
- **Encryption** – specifies whether encryption is enabled (+) or disabled (-)
- **Data transfer mode** – specifies whether the reliable (R) or unreliable (U) mode is used
- **Interval** – specifies how often a package was sent out (in seconds)
- **Sent packages** – specifies how many packages were sent out in total
- **Enabled interfaces** –  (infrastructure Wi-Fi)  (pee-to-peer Wi-Fi)   
(Bluetooth)

Table 6. Performance test results.

Test ID	Layout ID	Slave device ID	Encryption	Data transf. mode	Interval (s)	Sent packages	Enabled interfaces	Min. lat. (s)	Median lat. (s)	Mean lat. (s)	Max. lat. (s)	Lost packages
<b>T001</b>	L1	S1	+	R	1	100		0.006863	0.356500	0.392200	0.995600	0
<b>T002</b>	L1	S1	+	R	1	500		0.00645	0.13050	0.14720	0.91650	0
<b>T003</b>	L1	S1	-	R	1	500		0.006041	0.124800	0.128000	0.743500	0
<b>T004</b>	L1	S1	+	U	1	500		0.006375	0.128700	0.130500	0.348100	0
<b>T005</b>	L1	S1	-	U	1	500		0.006148	0.131700	0.136400	0.442600	3
<b>T006</b>	L7	S1	-	U	1	500		0.01859	0.04930	0.05178	0.12780	0
<b>T007</b>	L7	S1	-	U	1	500		0.01730	0.04494	0.05352	0.11050	0
<b>T008</b>	L7	S1	+	R	1	500		0.02069	0.02341	0.03339	0.11320	0
<b>T009</b>	L7	S1	+	R	1	500		0.02070	0.02920	0.03868	0.14940	0
<b>T010</b>	L2	S1	+	R	1	500		0.006396	0.129900	0.177100	0.998100	0
<b>T011'</b>	L2	S1	-	U	1	500		0.002364	0.126800	0.154800	0.979400	0
<b>T012</b>	L3	S1	-	U	1	500		0.01888	0.05488	0.05842	0.12900	0
<b>T013</b>	L3	S1	+	R	1	500		0.02078	0.03534	0.04164	0.15100	0
<b>T014</b>	L4	S1	+	R	1	500		0.02094	0.02766	0.03751	0.24020	0
<b>T015</b>	L4	S1	+	R	1	500		0.02074	0.02298	0.02733	0.10960	0
<b>T016</b>	L5	S1	+	R	1	500		0.006497	0.122900	0.121900	0.243400	0
<b>T017</b>	L5	S1	+	R	1	500		0.006238	0.124400	0.127600	0.922100	0



<b>T018</b>	L6	S1	+	R	1	500		0.007936	0.124400	0.126100	0.403200	0
		S2						0.00937	0.14430	0.16210	0.86420	0
		S3						0.008501	0.126000	0.127700	0.402900	0
		S4						0.01108	0.15260	0.17240	0.91410	0
		S5						0.01006	0.14750	0.16820	0.85360	0
		S6						0.03148	0.14670	0.16900	0.84550	0
<b>T019</b>	L6	S1	-	R	1	500		0.008995	0.123200	0.128400	0.684500	0
		S2						0.008753	0.127700	0.131900	0.684600	0
		S3						0.02412	0.12510	0.13080	0.68310	0
		S4						0.009692	0.136600	0.142300	0.683700	0
		S5						0.002718	0.238400	0.319100	0.966100	0
		S6						0.00856	0.13060	0.13630	0.68490	0
<b>T020</b>	L6	S1	-	U	1	500		0.006921	0.123900	0.125000	0.373600	0
		S2						0.008002	0.128900	0.136000	0.371500	0
		S3						0.02116	0.06981	0.08066	0.23910	0
		S4						0.02762	0.08811	0.09335	0.27050	0
		S5						0.0158	0.1424	0.1579	0.8816	0
		S6						0.002898	0.204700	0.302700	0.994800	2
<b>T021</b>	L6	S1	+	R	1	500		0.00760	0.12090	0.12420	0.41650	0
		S2						0.009467	0.131000	0.140900	0.701300	0
		S3						0.01012	0.12440	0.12710	0.36760	0
		S4						0.01194	0.13310	0.14650	0.70170	0
		S5						0.0113	0.1330	0.1451	0.7021	0
		S6						0.008926	0.132400	0.145900	0.621400	0
<b>T022</b>	L8	S1	+	R	1	500		0.02086	0.05952	0.07736	0.24910	0
		S2						0.02649	0.11920	0.12190	0.24460	0
		S3						0.02786	0.10280	0.10870	0.23770	0
		S4						0.02207	0.04523	0.06122	0.22060	0
		S5						0.02420	0.08323	0.08971	0.22410	0

		S6						0.02287	0.06293	0.06856	0.21960	0
<b>T023</b>	L6	S1	+	R	5	500		0.02021	0.13990	0.20910	1.85200	0
		S2						0.02269	0.06728	0.08493	0.31730	0
		S3						0.02240	0.06089	0.07250	0.24780	0
		S4						0.02285	0.10730	0.11180	0.30120	0
		S5						0.03141	0.15190	0.22780	1.85200	0
		S6						0.007545	0.139700	0.210000	1.853000	0
<b>T024</b>	L6	S1	+	R	120	15		0.05997	0.14190	0.18570	0.66040	0
		S2						0.02371	0.03145	0.04627	0.12990	0
		S3						0.1276	0.1459	0.2672	0.8575	9 <sup>28</sup>
		S4						N/A	N/A	N/A	N/A	15 <sup>29</sup>
		S5						0.04535	0.14000	0.24940	0.85780	0
		S6						0.06096	0.14390	0.19210	0.66170	0

The graphical results of these performance tests are depicted in the appendices (see Appendix I).

## Hypotheses and conclusions

Before conducting the tests, a number of hypotheses (see Table 7) were built based on the MCF documentation and previous experiences with Bluetooth and Wi-Fi networks and interfaces.

Table 7. Hypotheses and confirmation.

ID	Hypothesis	Result	Comment
<b>H1</b>	<b>data transmission will be faster, if the encryption is turned off</b>	<b>TRUE</b>	<b>compare T002 and T003 results</b>
<b>H2</b>	<b>data transmission will be faster, if the transfer mode is set to unreliable (~ UDP)</b>	<b>TRUE</b>	<b>compare T002 and T004 results</b>
<b>H3</b>	<b>packet loss could be observed, if the transfer mode is set to unreliable (~</b>	<b>TRUE</b>	<b>see T005 (3 lost packages)</b>

<sup>28</sup> Lost packages because of disconnection

<sup>29</sup> Lost packages because of disconnection

	UDP)		
H4	various interfaces (Wi-Fi, Bluetooth) can cause interference, which causes packet loss and/or slows down the connection	TRUE	compare T006 and T007
H5	the Bluetooth connection will get unstable (packet loss and/or connection drops) on the distance of more than 10 m	FALSE	see T022; the tests did show that the bluetooth connection remained stable by up to 30m distances
H6	the bluetooth connection will get unstable (packet loss and/or connection drops) if more devices are connected	FALSE	compare T008 and T022
H7	the peer-to-peer Wi-Fi connection is slower than Bluetooth, but faster than the infrastructure Wi-Fi connection	FALSE	
H8	the peer-to-peer Wi-Fi connection enables the devices to communicate in a range of up to 10m	FALSE	the tests did show that the maximal distance, where the devices were able to connect was 3m
H9	the connection can drop, if the communication takes places less frequently	TRUE	compare T023 and T024

The most remarkable result discovered, was the stability and reliability of the Bluetooth connection. Up to 6 interconnected slaves gave the maximal average one way latency of ~ 0.12 seconds. However connecting more devices simultaneously will most probably show some setbacks and therefore the infrastructure Wi-Fi should be relied on instead.

## 7 Future work

This chapter describes the further work and enhancement possibilities for Partiture, considering the unimplemented user stories and their assigned priorities.

### 7.1 Extendable screen support

The most problematic issue, which in particular affects orchestra conductors, is the lack of screen sizes of the currently marketed mobile devices. There can be relatively many different instrumental (and vocal) sections (parts) in the score that vary depending on the orchestra type and the corresponding musical piece, which will define its physical layout as well. An average chamber orchestra arrangement contains approximately 15 sections per sheet (page), which means that even on an A3 sized sheet, the measures are quite difficult to follow. This aspect usually encourages the conductors to learn the pieces by heart. Since many conductors are also used to having at least two sheets opened simultaneously, it can be stated that the current 9.7-inch screen solution is not satisfactory for gaining an efficient user experience.

The superior solution would be to combine smartphones (or tablets) with next generation display technologies, such as e-papers (electronic papers) [17]. This approach would make the system more flexible in terms of musician's different dimension needs (see Figure 15) and fulfil also some hardware related requirements (related BR6).

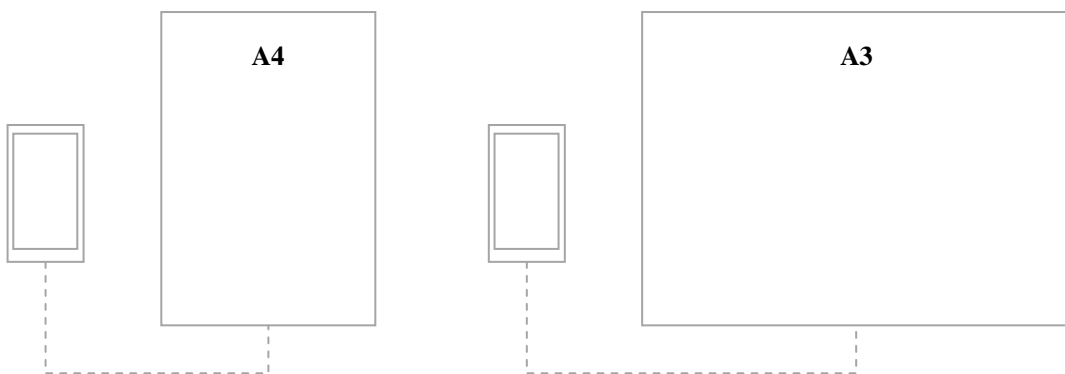


Figure 15. Smartphones combined with different e-paper formats.

As an alternative workaround, ordinary LCD (widescreen) monitors or smart TV-s could be used for rendering or extending the smartphone's (or tablet's) graphical output (see Figure 16). These could be stationary installed at music schools, rehearsal locations and other commonly used spaces (for instance, musicians' homes). It would allow keeping the application logic (data) on the mobile devices and therefore the system still partially mobile.



Figure 16. A smartphone combined with a flat widescreen monitor.

The musicians in an orchestra are usually organized pairwise because of the page turning aspect and therefore they quite often share also the same physical sheet. Technically it would be possible to “bind” the pair member devices in a way that they will act as one together, which would be also a form of screen extending (see Figure 17).

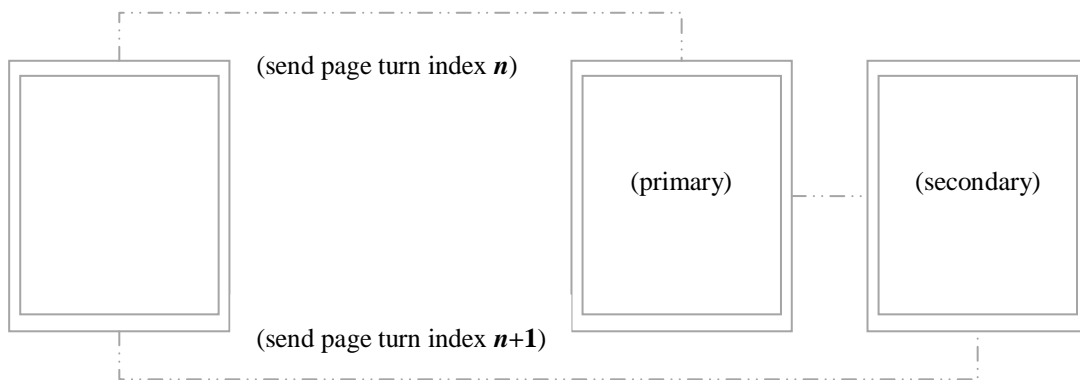


Figure 17. Paired devices as an extended screen.

## 7.2 Extractable scores

Another major issue, which affects the orchestra members, is related to the different variants of the score. Whereas the conductor wants to have the full score for an overview, certain type of players (e.g. a violin group) are only interested in certain (extracted) parts of

it. This means that one extracted score page can hold a different range of measures, which will break the current page turning and remark creating concept, where everybody is having the same variant, layout of the score. In order to solve this problem completely, the PDF file format has to be replaced with MusicXML [18]. This change will also allow implementing new use cases like collaborative composing, direct score manipulation, etc. However introducing MusicXML also has some drawbacks – most of the currently used scores are in PDF format and creating the MusicXML source out of it requires integrating Optical Music Recognition (incl. Optical Measure Recognition [19]) techniques during the score preparation in the web application environment. Increasing data packet shipment amounts can have a critical hit on the performance as well.

### **7.3 Multiplatform support**

Different people have usually different preferences regarding the mobile devices and / or operating systems. In many cases these preferences are related to their previous experiences or already existing devices at hand. The current implementation targets only Apple iPad's (running on Apple iOS), whereas many musicians would like to use also other popular devices and platforms like Android, Linux or Windows. The application logic will be quite trivial to be taken over for other platforms, integrating the communication mechanisms between multiple platforms (devices) much more complex. Probably the most appropriate solution would be to implement a custom cross-platform version of the MCF (e.g. in C/C++), which could be later reused on all targeted platforms.

### **7.4 Digital rights management**

The preparing and distributing of well-formatted scores is a competency of the publishing companies. There are very many different terms and conditions on how the scores can be distributed, photocopied, digitally reproduced, used for practicing or performing and these can also vary depending on the company, concrete piece, the composer, etc.

Currently the score distribution is planned to be handled via the web application environment, where the authorized person (e.g. conductor) can share the selected score(s) among the eligible users (e.g. orchestra members). On the one hand, this approach helps to avoid uncontrolled score distributing, on the other hand it also means that the users have to download the scores first before going to the performance or practice session.

Purchasing the licence(s) for digital reproduction of the scores can be quite expensive; therefore many publishing companies offer various licensing models, such as leasing scores for a specified time period. In order to provide these kind of solutions for the users of Partiture, DRM (Digital rights management) [20] tools have to be integrated in the applications.

## **7.5 Score following aid**

Occupying the conductors constantly with the page turning task is not a sufficient solution for this problem. Some might miss the correct page turns and some might even forget about them.

The first ideas for synchronizing raw audio input and scores in real-time [21], were proposed already 30 years ago, but still there are relatively many unresolved challenges in this field. Several approaches (e.g. [22]) have been tested since then, but without any outstanding results so far. Even if the page turning task cannot be fully automated yet, it would be a great help for the musicians to get it working at least partially (as an aid) by combining different algorithms and strategies (tempo and pitch analysis, timing mechanisms, etc.).

## 8 Conclusions

There have not been many studies conducted on the topic „interoperable music stands“, although the contemporary state of technology definitely has been allowing to implement considerable solutions for the past few years already. However, quite recent developments in this area clearly emerge the need for such kind of systems.

There are numerous use cases, which can be identified for individual, teacher-student(s), conductor-orchestra, choirmaster-choir, instrumentalist-assistant, etc. The usage purposes from the most important aspect to be considered, when collecting and analyzing them. It is necessary to align the final set together with a group of professional musicians.

Although the set of devices, which was used for conducting the performance tests for identifying the bottlenecks with different configurations and measuring latency, was relatively small, the tests still gave very promising results regarding feasibility and some ideas and pointers regarding scaling up the network by adding more nodes to it.

There will be always concerns regarding hardware (component) failures and aspects like decreasing battery levels, therefore in order to gain the musicians trust to start using these systems, especial attention has to be paid on handling errors and exceptional cases at the software level – e.g. if the application should terminate during the concert, relaunching it again should automatically proceed to the prior crash state (e.g. open the correct room and restore the session in it) without requiring a second touch from the musician.

There are several enhancement possibilities on the software as well as hardware side, which could give the system more value and therefore help to spread it among more musicians with different preferences and needs. From the software perspective, one of the main challenges will be to port the system onto other mobile platforms and enable cross-platform communication in a similar way, as implemented currently. Concerning from the hardware perspective, the support for larger (external) screens should be added.

Introducing and importing (music) lesson related familiar terms like room, session, member, participation, desk, workspace in the context of Partiture and reusing them as architectural elements, helps to understand the concept (abstract) of the application in a more comprehensive way for the musicians and other stakeholders.



The area of „interoperable music stands“ overall offers many interesting challenges for musicians, software developers as well as hardware developers as for the next steps and future work.

## 9 References

- [1] Agawi, Inc. (2013, November) TouchMarks II: Touchscreen Latencies in Flagship Tablets. [Online]. <http://appglimpse.com/blog/>
- [2] Zack Settel and Cort Lippe, "Real-time musical applications using frequency domain signal processing," in *Applications of Signal Processing to Audio and Acoustics, 1995., IEEE ASSP Workshop on*, New Paltz, NY, 1995, pp. 230-233.
- [3] Meinrad Müller, *Information Retrieval for Music and Motion*. Heidelberg, Germany: Springer Berlin Heidelberg, 2007.
- [4] Carleton College. (2012, October) An Intelligent Music Stand. [Online]. [http://cs.carleton.edu/cs\\_comps/0910/musicstand](http://cs.carleton.edu/cs_comps/0910/musicstand)
- [5] (2012, October) Piano Hero An intelligent music stand. [Online]. [http://cs.carleton.edu/cs\\_comps/0910/musicstand/final\\_results/index.html](http://cs.carleton.edu/cs_comps/0910/musicstand/final_results/index.html)
- [6] AOL Inc. (2013, October) Tonara's iPad App Looks To Reinvent Sheet Music For The Digital Age. [Online]. <http://techcrunch.com/2011/09/12/tonara-ipads-app-looks-to-reinvent-sheet-music-for-the-digital-age/>
- [7] Technmusiced. (2012, November) Brussels Philharmonic uses Samsung Galaxy Tablets for Performing. [Online]. <http://technmusiced.wordpress.com/2012/11/07/belgium-symphony-uses-samsung-galaxy-tablets-for-performing/>
- [8] AOL Inc. (2014, March) FireChat For iOS Is A Hyperlocal Anonymous Chat Network That Doesn't Need An Internet Connection. [Online]. <http://techcrunch.com/2014/03/20/firechat-for-ios-is-a-hyper-local-anonymous-chat-network-that-doesnt-need-an-internet-connection/>
- [9] Mike Cohn, *User Stories Applied*. Addison Wesley, 2004.
- [10] Philippe Kruchten, "Architectural Blueprints—The “4+1” View Model of Software Architecture," *IEEE Software*, vol. 12, no. 6, pp. 42-50, November 1995.
- [11] Wikimedia Foundation, Inc. (2014, May) Universally unique identifier. [Online]. [http://en.wikipedia.org/wiki/Universally\\_unique\\_identifier](http://en.wikipedia.org/wiki/Universally_unique_identifier)
- [12] (2014, February) Bluetooth SIG. [Online]. <https://developer.bluetooth.org/TechnologyOverview/Pages/BLE.aspx>

- [13] Bluetooth SIG Inc. (2014, February) A Look at the Basics of Bluetooth Technology. [Online]. <http://www.bluetooth.com/Pages/Basics.aspx>
- [14] (2013, December) WWDC 2013 Session Videos. [Online]. <https://developer.apple.com/videos/wwdc/2013/>
- [15] Laurie Williams. (2006) White-Box Testing. [Online]. <http://www.chaudhary.org/WhiteBox.pdf>
- [16] N Golmie, "Interference in the 2.4 GHz ISM band: impact on the Bluetooth access control performance," in *Communications, 2001. ICC 2001. IEEE International Conference on*, Helsinki, 2001, pp. 2540 - 2545.
- [17] Jason Heikenfeld, Paul Drzaic, Jong-Souk Yeo, and Tim Koch, "Review Paper: A critical review of the present and future prospects for electronic paper," *Journal of the Society for Information Display*, vol. 19, no. 2, pp. 129-156, February 2011.
- [18] MAKEMUSIC, INC. (2014, May) musicXML. [Online]. <http://www.musicxml.com/>
- [19] Gabriel Vigliensoni, Gregory Bulet, and Ichiro Fujinaga, "Optical measure recognition in common music notation," in *Proceedings of the International Society for Music Information Retrieval*, Curitiba, 2013, pp. 125-130.
- [20] Qiong Liu, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard, "Digital rights management for content distribution," in *ACSW Frontiers '03 Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, Darlinghurst, 2003, pp. 49-58.
- [21] Roger B. Dannenberg, "An On-Line Algorithm for Real-Time Accompaniment," in *Proceedings of the 1984 International Computer Music Conference (ICMC)*, 1984, pp. 193-198.
- [22] A. Arzt, G. Widmer, and S. Dixon, "Automatic Page Turning for Musicians via Real-Time Machine Listening," in *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, Patras, 2008, pp. 241-245.

# **Partiture: Koostöövõimeline noodipuldirakendus mobiilsetele seadmetele**

Magistritöö (30 EAP)

Kaarel Kruus

## **Resüme**

Töö peamiseks eesmärgiks on täita lünk, mis on jäänud info- ja kommunikatsioonitehnoloogia ja klassikalise muusikamaailma vahele. Nutitelefonide ning tahvelarvutite müük on viimaste aastatega seljatanud personaalarvutite müügi, kasvatades üha enam ka nõudlust uute ning innovaatiliste tarkvararakenduste järele. Mobiilseid seadmeid kasutatakse kõikvõimalikes kohtades ja eluvaldkondades, nagu näiteks tervishoiu asutustes, lennukites jm. tööülesannete täitmist toetavate ning meelelahutust pakkuvate vahenditena. Tavapärasest vähem võib infotehnoloogiliste lahenduste kasutamist märgata aga näiteks klassikalise muusika kontsertidel. Kui sümfooniaorkester mängib kontserdil või harjutab proovis mingit pala, siis miks ei võiks orkestrandid reaalselt jagada sama pala visuaalset esitust, jättes sellised operatsioonid nagu lehekülje keeramine, noodi modifitseerimine ning muud võimalikud tegevused ühe usaldusväärse isiku – dirigendi hooleks?

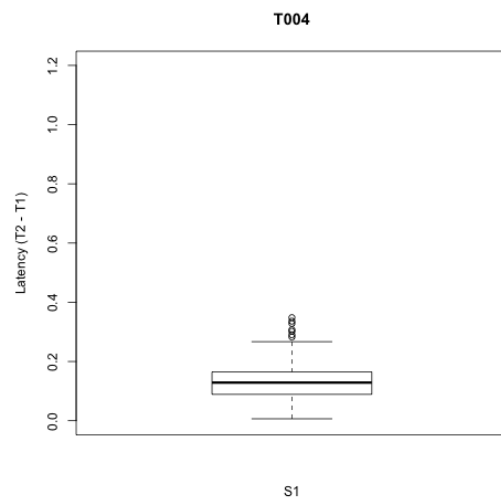
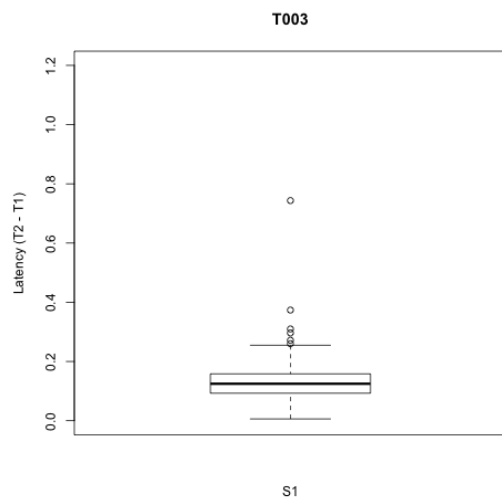
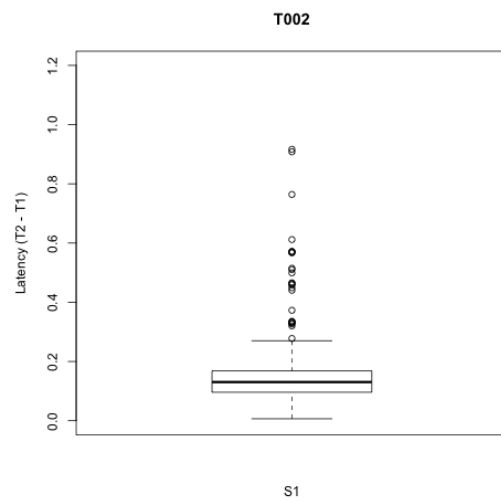
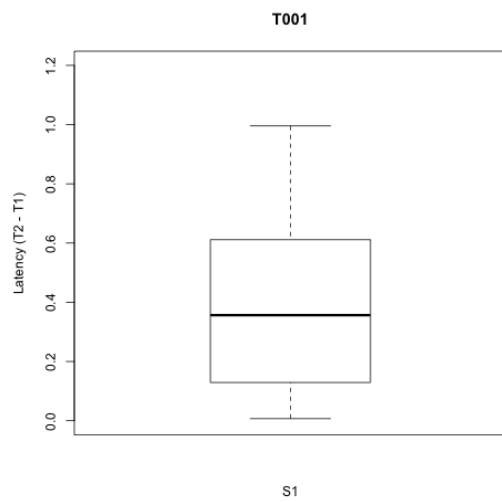
Sellise kontseptsiooniga lahendusi käesoleva töö planeerimisel ja taustauuringu tegemise hetkel olemas (vähemalt avalikult presenteeritud) ei olnud, siiski avalikustati töö kirjutamise jooksul üks üsna sarnane lahendus, mis samas näitab teema aktuaalsust ja vajadust selliste süsteemide järele. Olid küll olemas mõned ideeliselt sarnased rakendused, kuid ükski neist ei suutnud pakkuda mobiilsust ning efektiivsust sellisel tasandil, nagu käesoleva magistritöö raames arendatud.

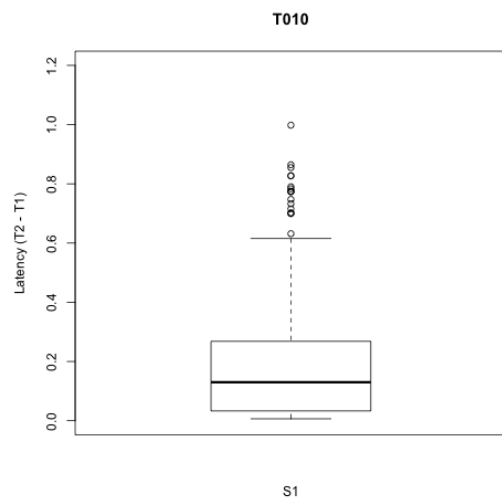
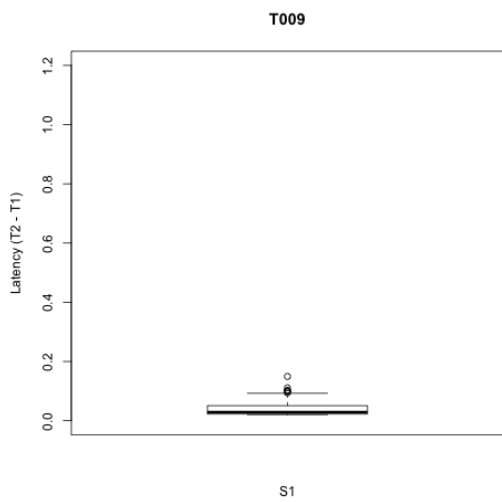
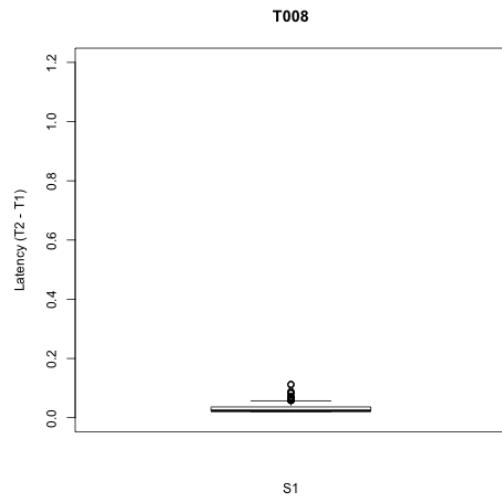
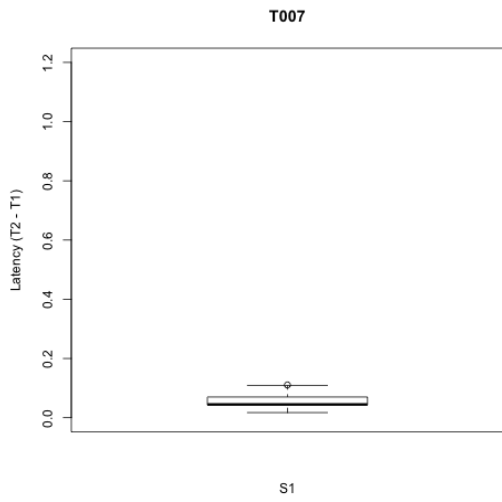
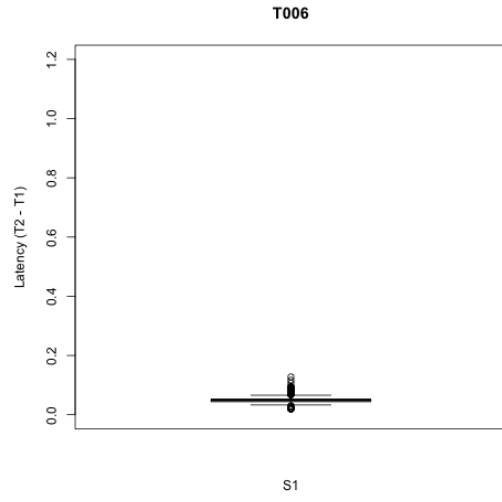
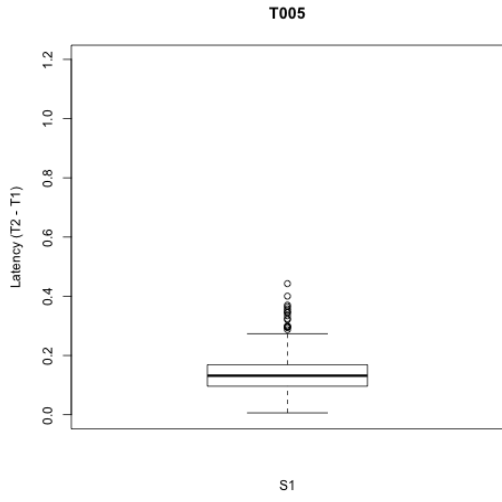
Töö esimeses peatükis defineeritakse probleem ning pakutakse välja lahendus. Teises peatükis antakse ülevaade kasutatavatest tööriistadest, raamistikest ning tuuakse välja see osa, mis lisab väärtust teiste samalaadsete rakenduste ees. Kolmandas peatükis antakse ülevaade samalaadsete või sarnaseid kontseptsioone realiseerivate süsteemide kohta. Neljas peatükk kirjeldab nõudeid arendatavale rakendusele. Viiendas peatükis tutvustatakse süsteemi ja rakenduse arhitektuuri. Kuuendas peatükis esitatakse süsteemi testimisega seotud tulemusi. Seitsmendas peatükis tuuakse välja edasise töö jätkamise pidepunktid. Kaheksas peatükk võtab kokku tööst tulenevad järeldused.

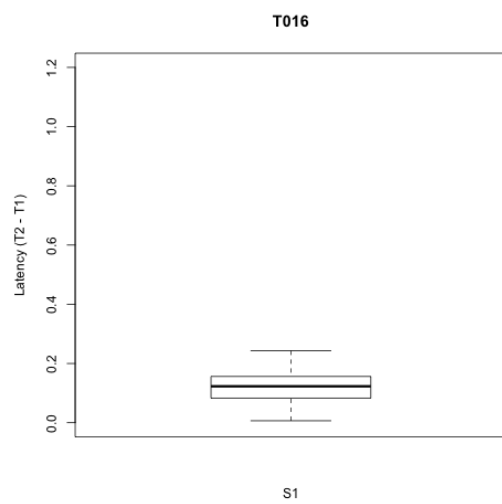
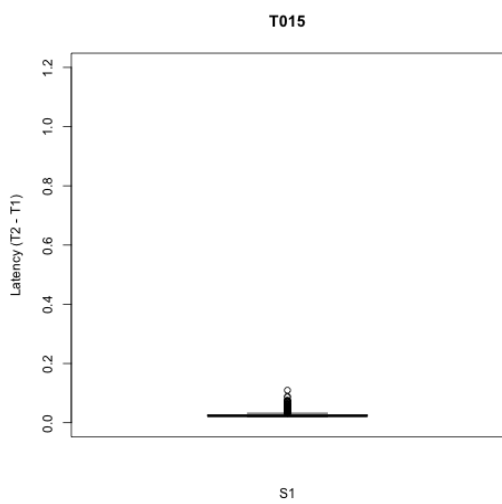
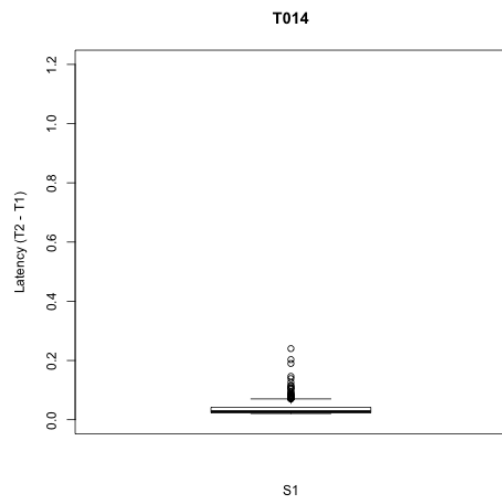
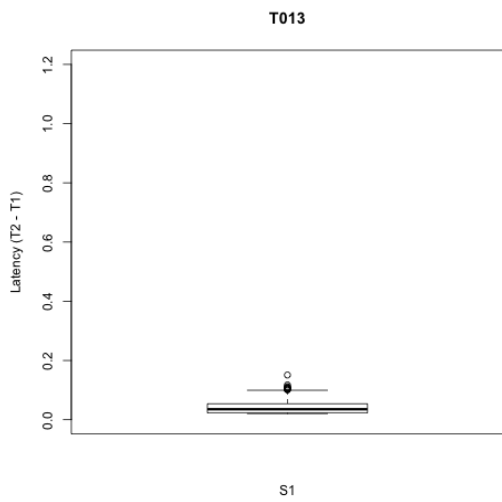
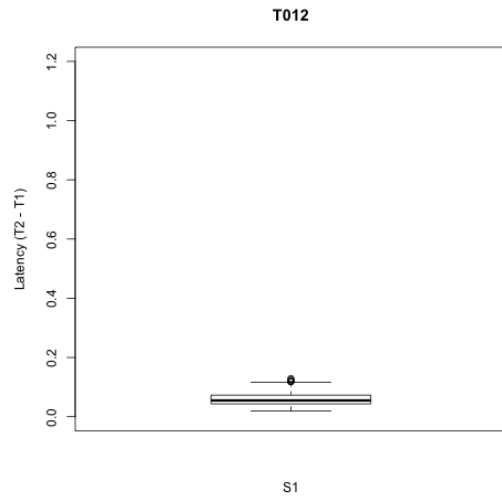
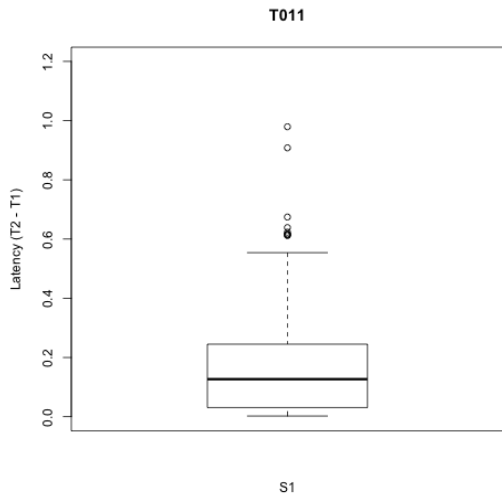
## Appendix

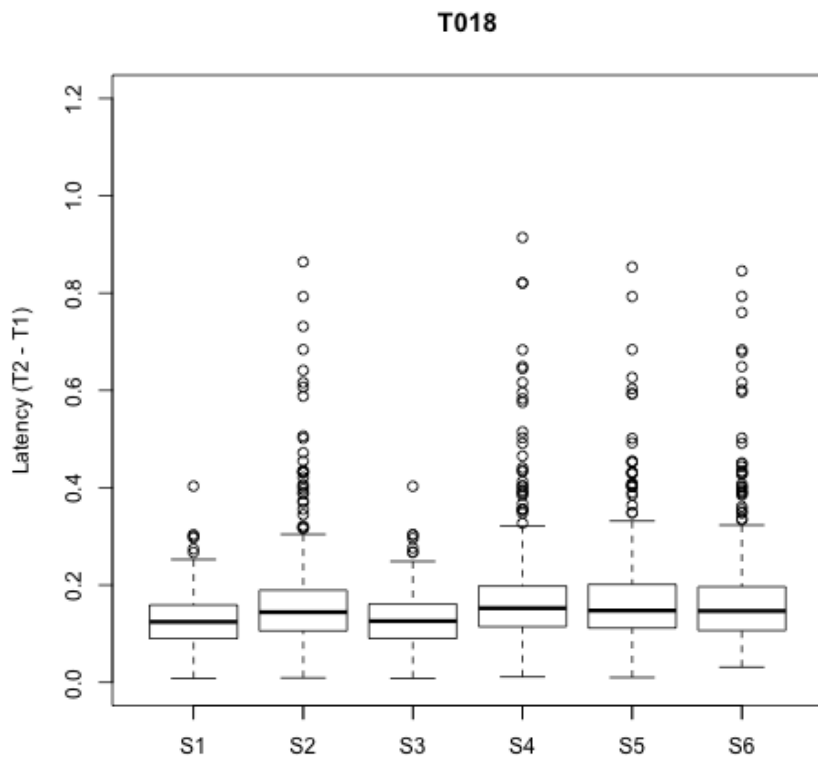
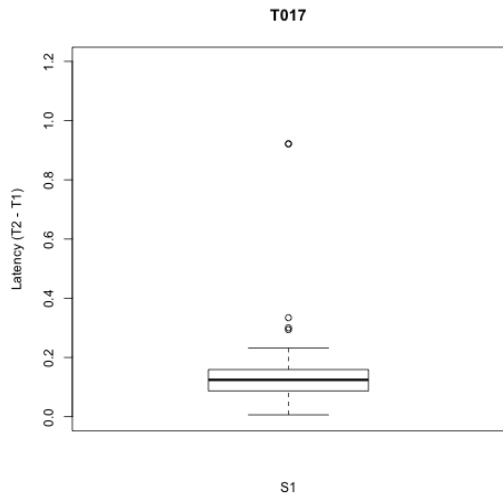
In the following appendices, the performance tests graphical results, an example set of storyboards for some application usage scenarios and information about the prototype application(s) can be located.

### I. Performance tests graphical results



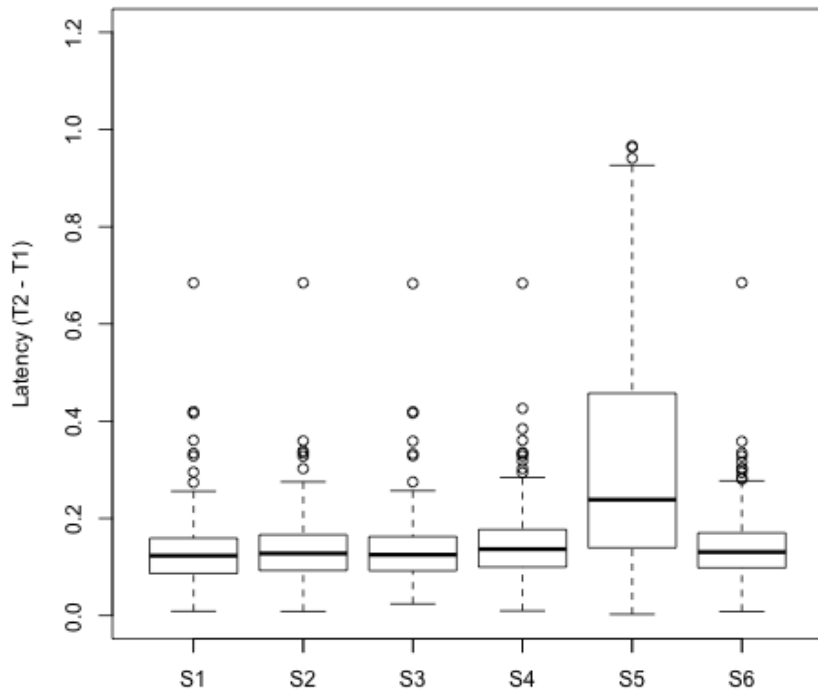




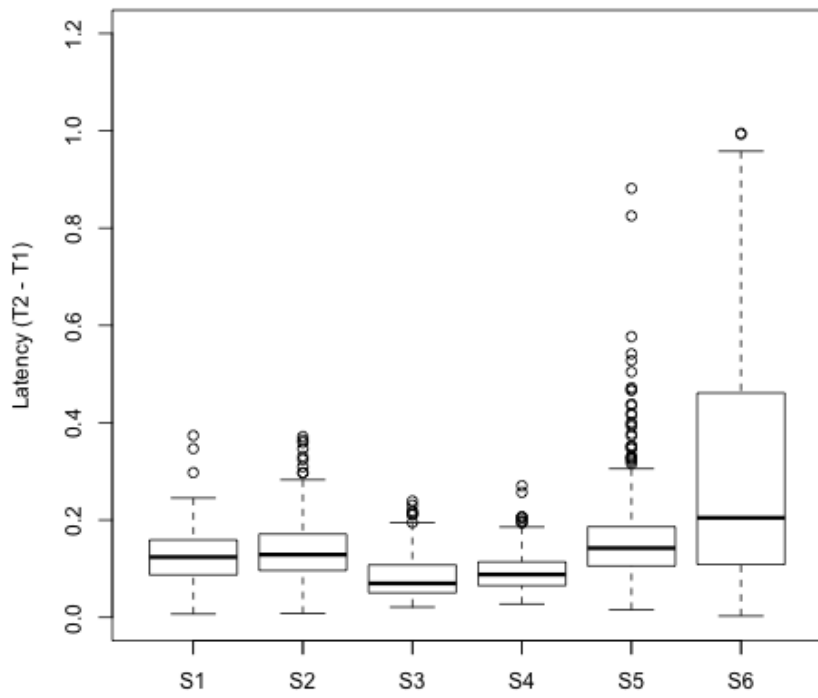




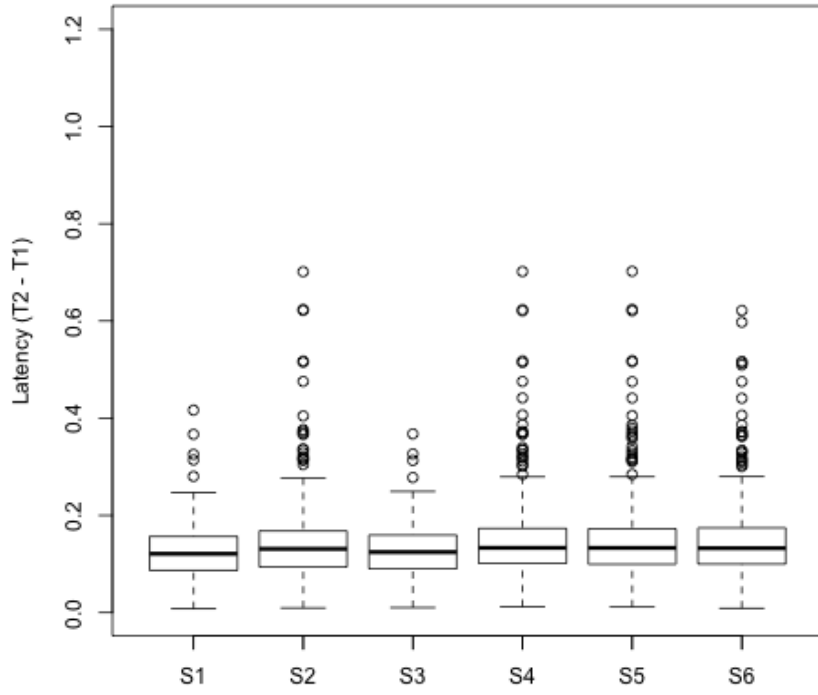
**T019**



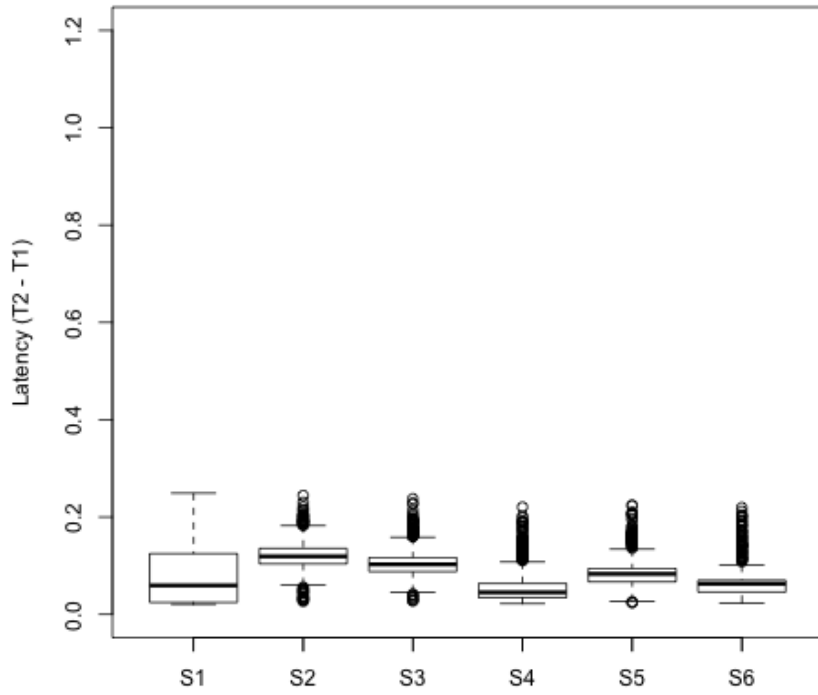
**T020**



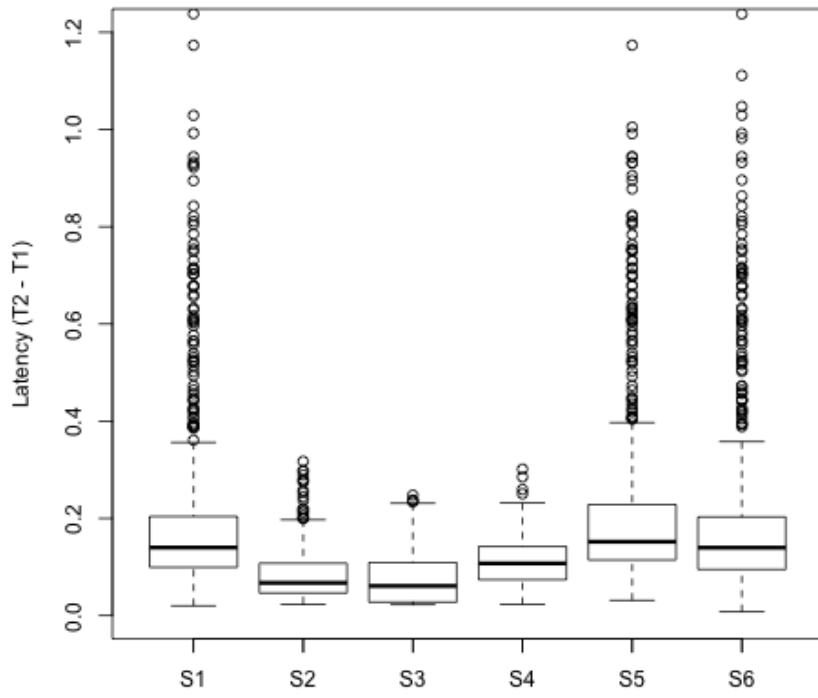
T021



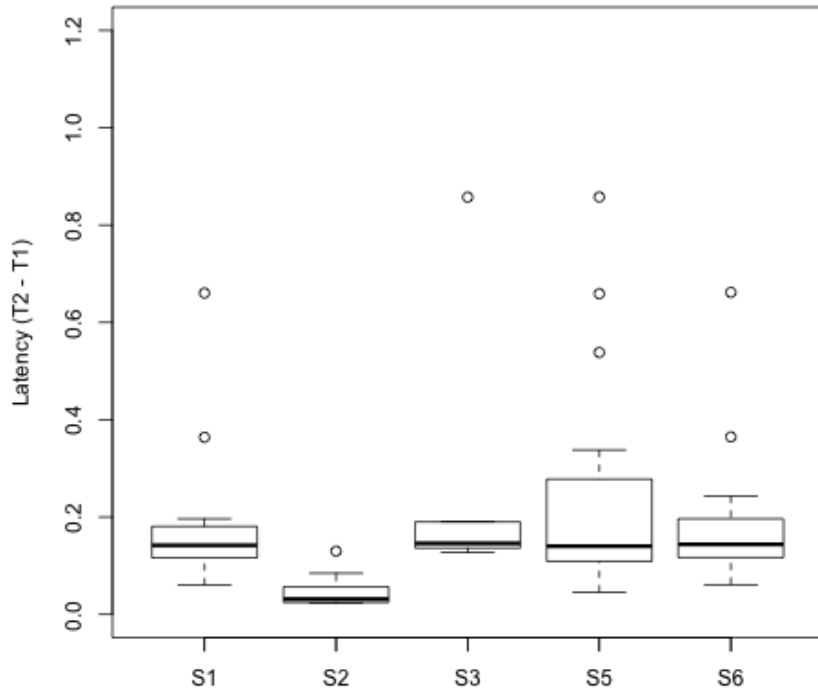
T022



T023



T024



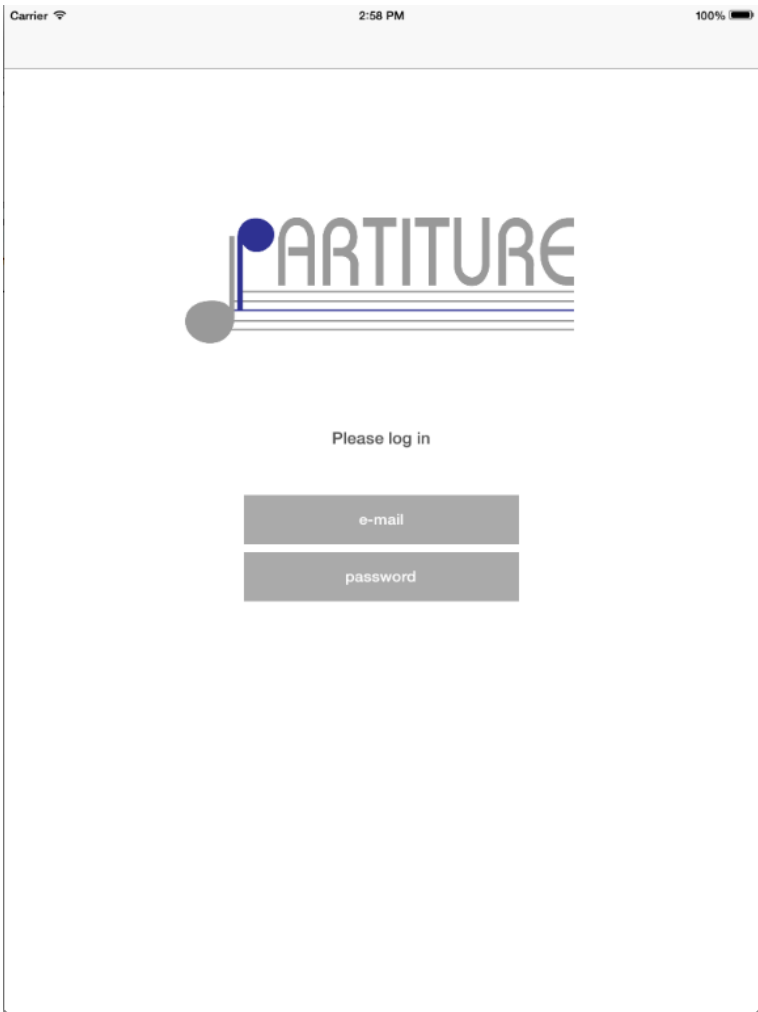
## II. Storyboards

<b>Story 1: User online authentication</b>	
Precondition(s): John Paige has created a user account using the web application.	
John Paige is a piano teacher, who opens the freshly installed mobile application on a device for the very first time. The mobile application will present him the authentication view. In order to be able to start using the mobile application's functionality, John completes a one-time <sup>30</sup> online authentication process <sup>31</sup> by entering his email address and password into the provided text fields and tapping on the button "Sign In". After a successful authentication attempt, the application will present him the score managing view.	
OD (Pre)	N/A

---

<sup>30</sup> Once the user has authenticated oneself, the mobile application can be used without re-authenticating until the user decides to de-authenticate.

<sup>31</sup> When the user has never authenticated oneself on a particular device before, an online authentication is needed (which requires an internet connection) to receive a user token, which is needed for performing further actions.

<p>GUI (Pre)</p>	
<p>OD (Post)</p>	<pre> :User name="John Paige" email="johnpaige@example.com" password=SHA("joHn1") isCurrent=TRUE token="14d678e1-1deb-4f45-8a32-681d6cbd28f1" </pre>

GUI (Post)



## Story 2: User de-authentication

Precondition(s): John Paige has completed Story 1 (or Story 3)

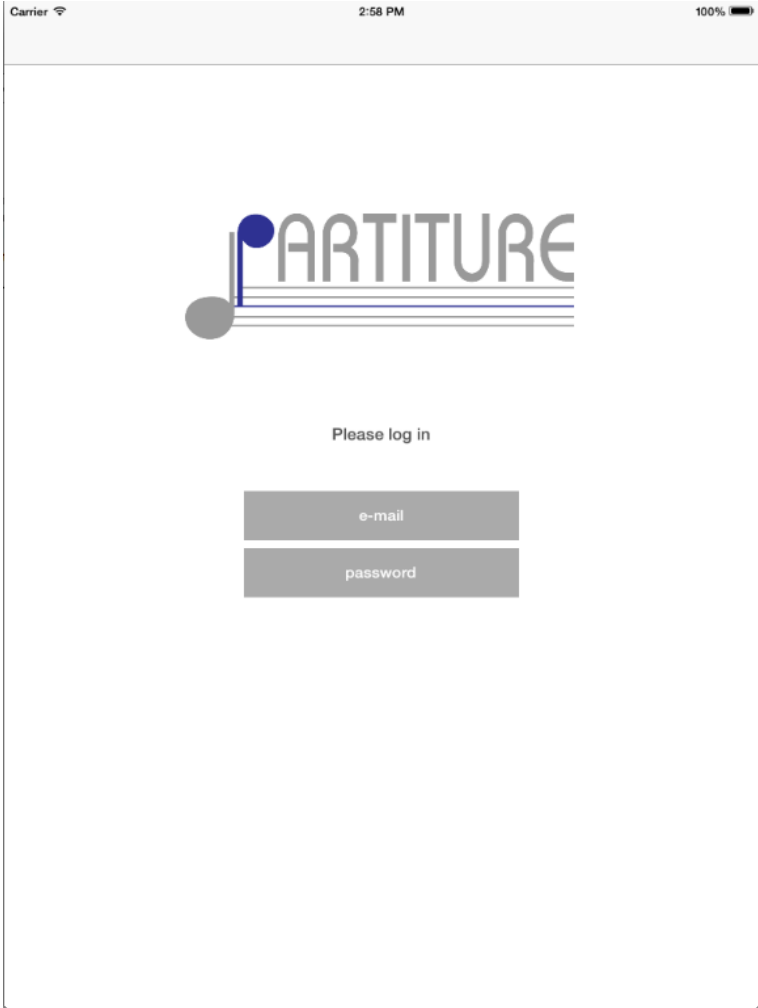
John Paige has used the mobile application for a while and decides to de-authenticate himself, in order to give his colleague the opportunity to work with it. John opens the settings view and taps on the button "Sign Out". After a successful de-authentication attempt, the application will present him the authentication view.

OD (Pre)

```
:User
name="John Paige"
email="johnpaige@example.com"
password=SHA("joHn1")
isCurrent=TRUE
token="14d678e1-1deb-4f45-8a32-681d6cbd28f1"
```

GUI (Pre)



OD (Post)	<pre>:User name="John Paige" email="johnpaige@example.com" password=SHA("joHn1") isCurrent=FALSE token="14d678e1-1deb-4f45-8a32-681d6cbd28f1"</pre>
GUI (Post)	 <p>The screenshot shows a mobile application interface. At the top, the status bar displays 'Carrier', signal strength, '2:58 PM', and '100%' battery. The main content area features the 'ARTITURE' logo, which consists of a stylized musical staff with a blue note and a grey circle. Below the logo, the text 'Please log in' is centered. Underneath, there are two grey rectangular input fields: the top one is labeled 'e-mail' and the bottom one is labeled 'password'.</p>



<b>Story 3: User offline authentication</b>	
Precondition(s): John Paige has completed Story 1 (and Story 2)	
<p>John Paige opens the mobile application on the device for the second time, after having used it once already. The mobile application will present him the authentication view. In order to be able to start using the mobile application's functionality again, John completes a one-time offline authentication process<sup>32</sup> by entering his email address and password into the provided text fields and tapping on the button "Sign In". After a successful authentication attempt, the application will present him the score managing view.</p>	
OD (Pre)	<pre> :User name="John Paige" email="johnpaige@example.com" password=SHA("joHn1") isCurrent=FALSE token="14d678e1-1deb-4f45-8a32-681d6cbd28f1" </pre>
GUI (Pre)	(Similar to Story 1)
OD (Post)	<pre> :User name="John Paige" email="johnpaige@example.com" password=SHA("joHn1") isCurrent=TRUE token="14d678e1-1deb-4f45-8a32-681d6cbd28f1" </pre>
GUI (Post)	(Similar to Story 1)

<sup>32</sup> When the user has once authenticated oneself on a particular device before, only an offline authentication is needed (which does not require an internet connection)

#### Story 4: Downloading a score

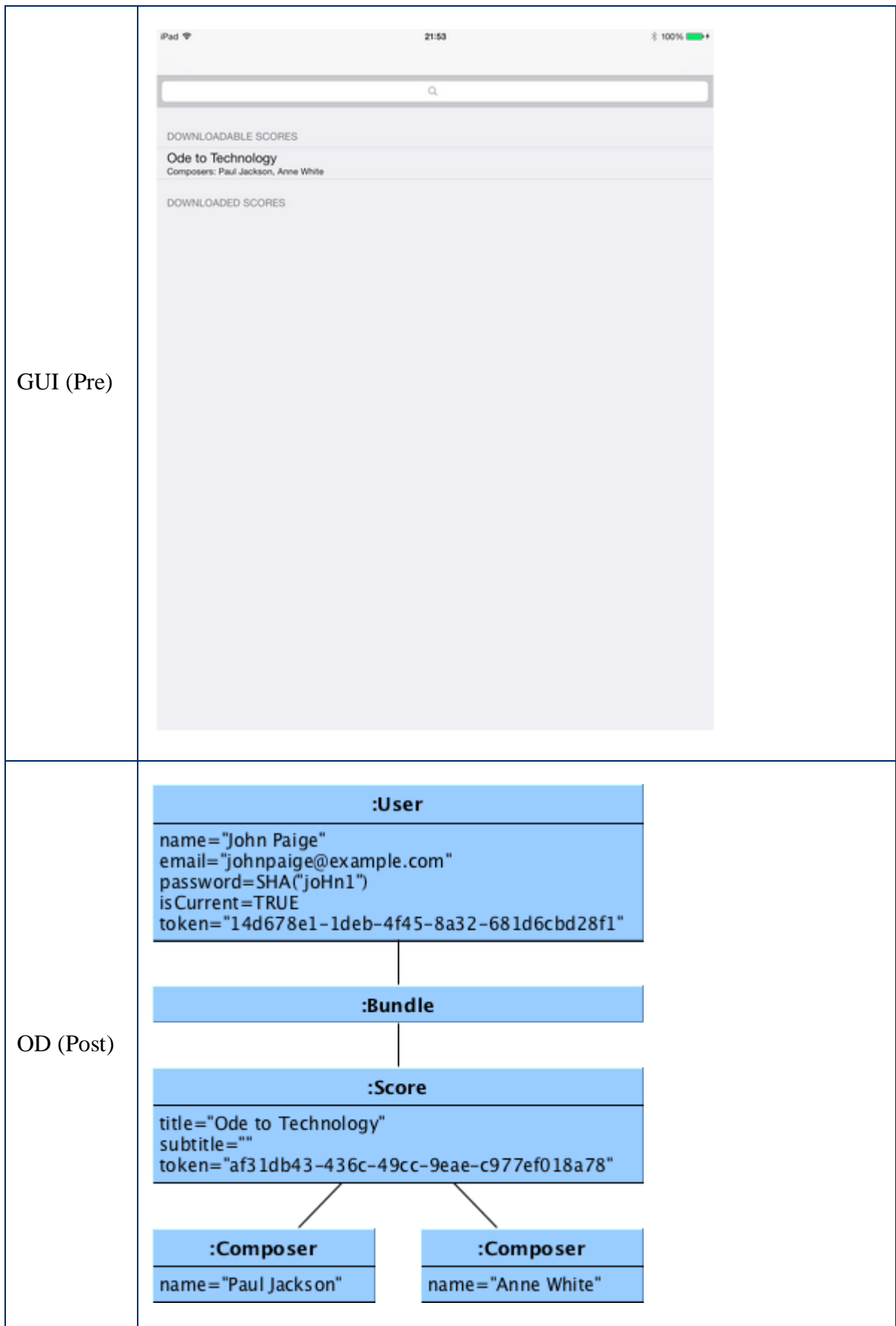
Precondition(s): John Paige has completed Story 1 (or Story 3) and at least one score has been made available (via the web application) for him to download.

John Paige opens the mobile application in a fresh state<sup>33</sup>. The mobile application will present him the score managing view, containing a list of already downloaded and available (downloadable) scores. John taps in the available scores list on the score list item titled "Ode to Technology", which is composed by Paul Jackson and Anne White. The user interaction will be disabled and an overlay with a loading indicator and download percentage presented. After the download process has finished, the overlay will be dismissed and the user interaction enabled again. The score is removed from the available scores list and added under the downloaded scores list.

OD (Pre)

```
:User  
name="John Paige"  
email="johnpaige@example.com"  
password=SHA("joHn1")  
isCurrent=TRUE  
token="14d678e1-1deb-4f45-8a32-681d6cbd28f1"
```

<sup>33</sup> The mobile application will not be resumed from the background (or suspended) mode, but started newly.



GUI (Post)

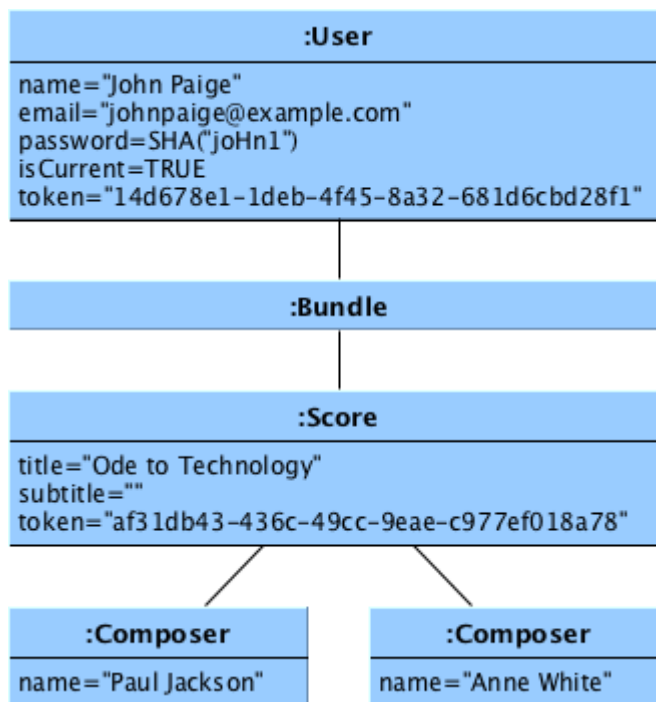


## Story 5: Creating a room

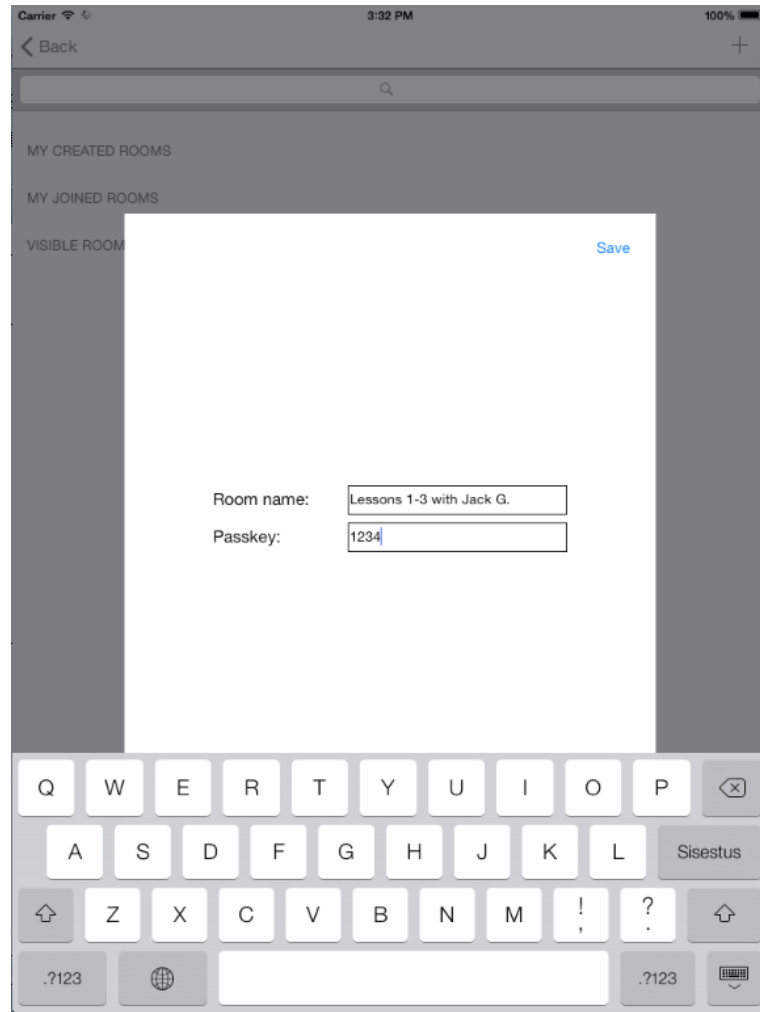
Precondition(s): John Paige has completed Story 4

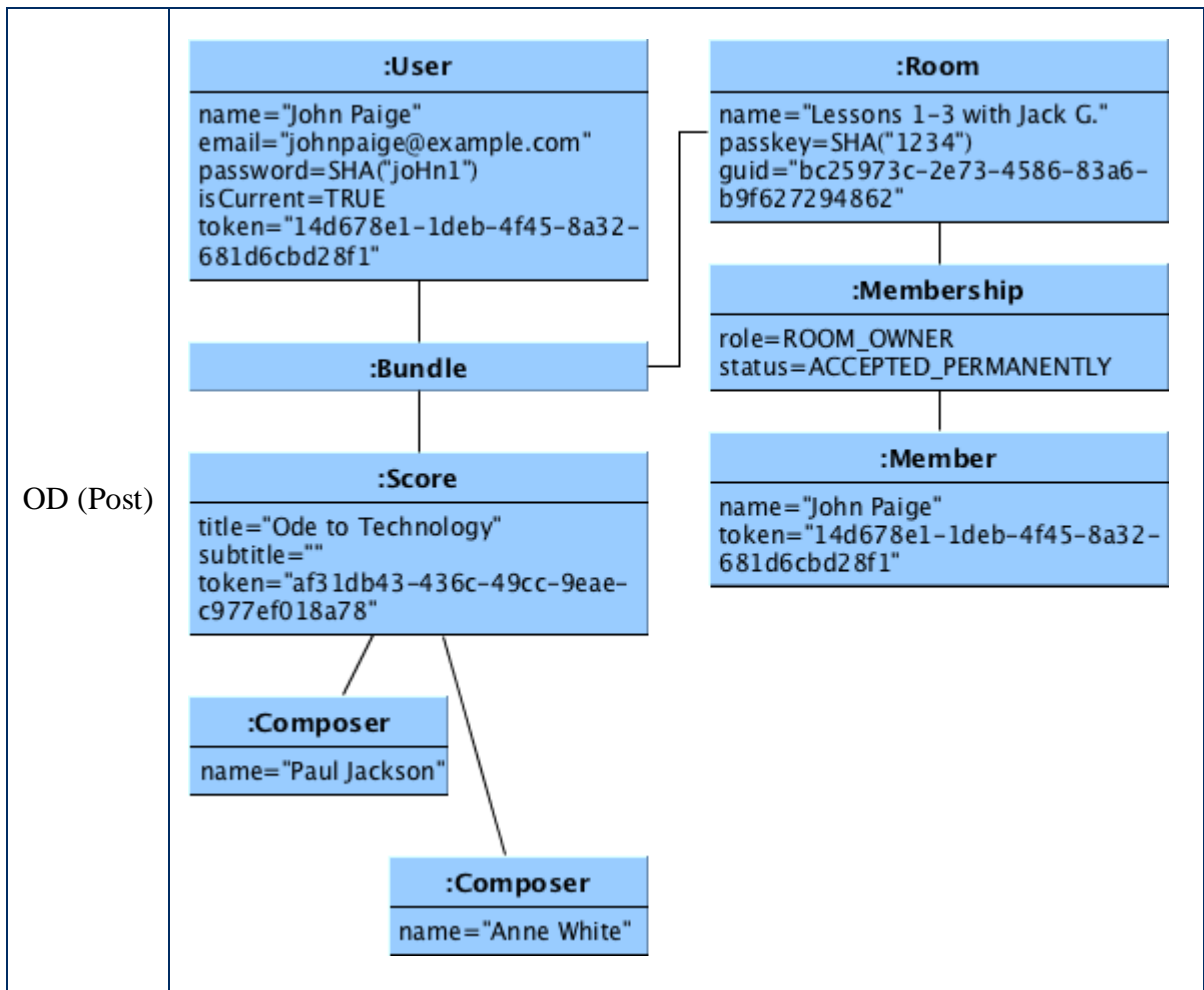
John Paige wants to prepare for the next three piano lessons with his student Jack Green. In order to be able to create a session and work with Jack in an interoperable way (e.g. share remarks), John has to create a room first. John has opened the score managing view, he taps on the score list item “Ode to Technology” in the downloaded scores list, after which the application will present him the room managing view. John taps on the “+” button in the top-right corner of the view, the mobile application will present him a popover prompting the necessary information for creating a new room. John enters “Lessons 1-3 with Jack G.” as the room name and “1234” as the passkey and taps on the button “Save”. The popover will be dismissed and the created room will appear in the room managing view list.

OD (Pre)

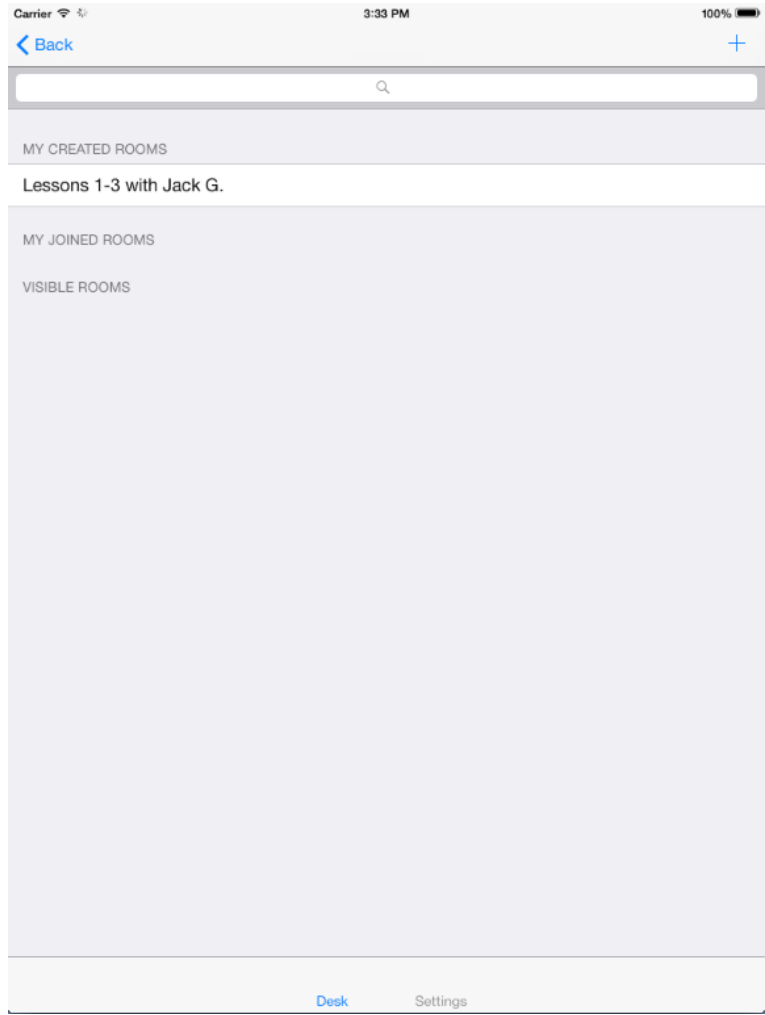


GUI (Pre)





GUI  
(Post)

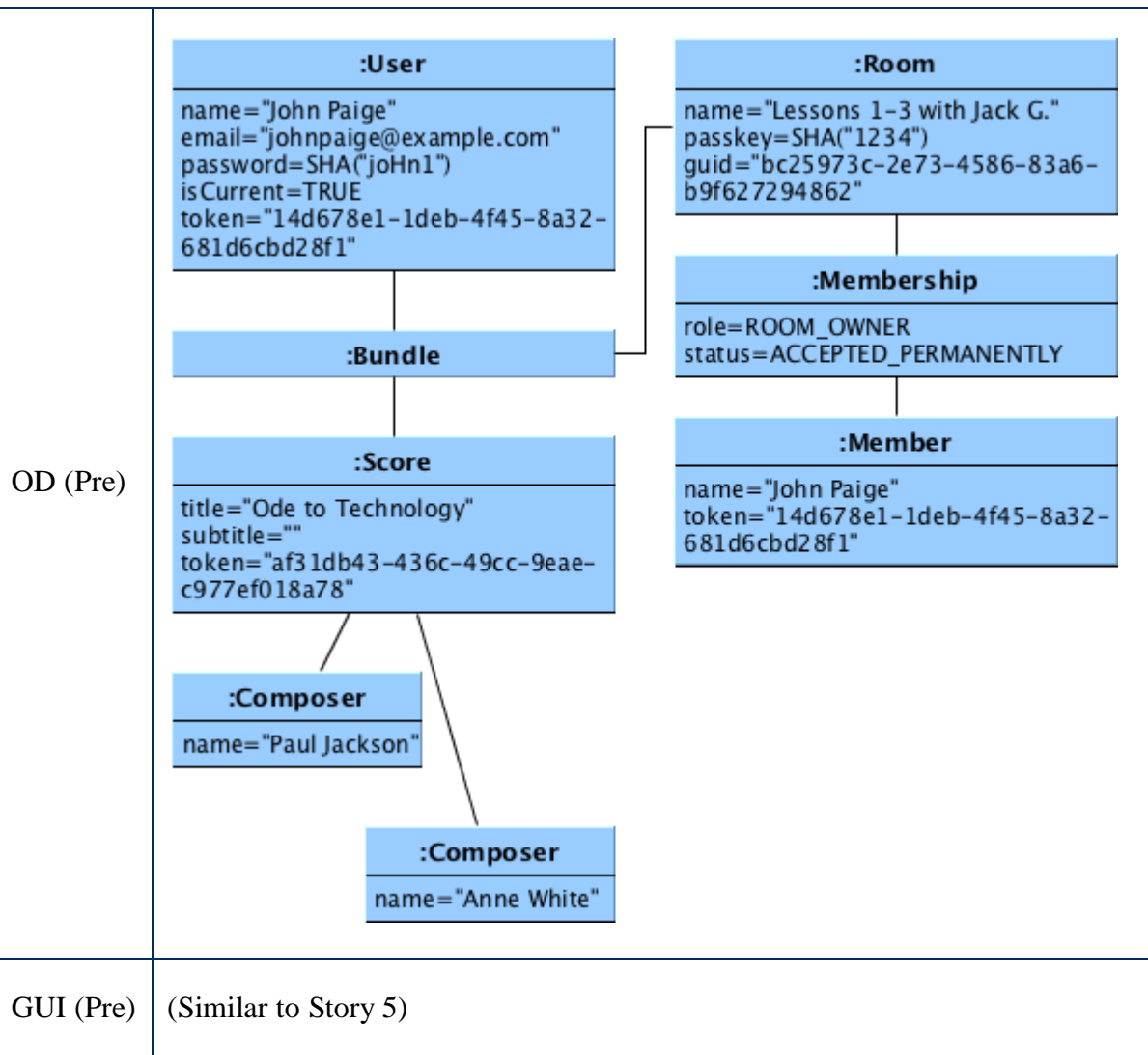


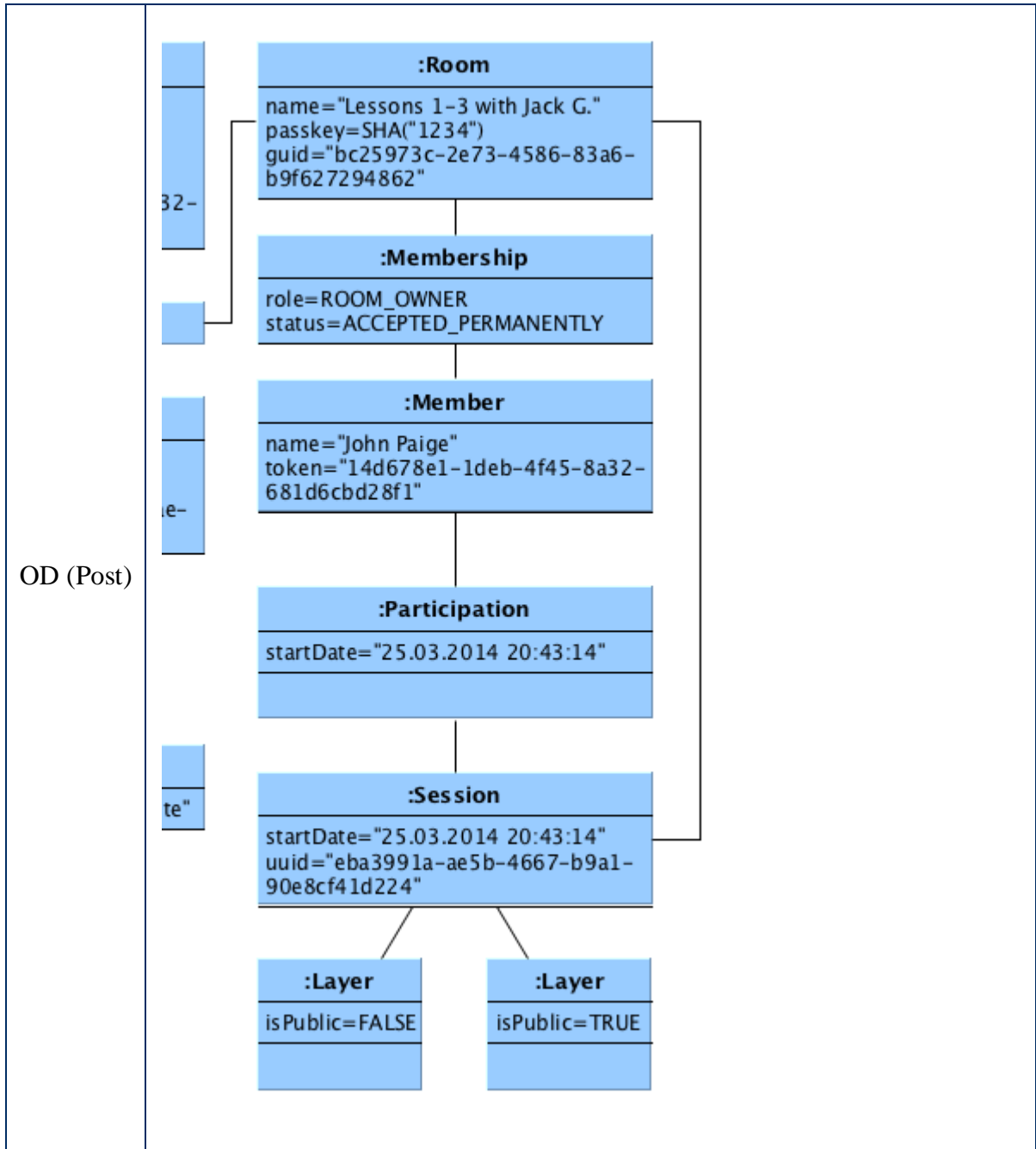


**Story 6: Starting a session**

Precondition(s): John Paige has completed Story 5

John has opened the room managing view, he taps on the room list item “Ode to Technology” in the available rooms list, after which the application will present him the workspace view.





GUI  
(Post)

Done Edit

21 F#m B C#  
Üht - moo - di mö - le - mad siin möö - da läe - vad, kui

23 F#m B C# C# (Ahi)  
vöö - ras puh - kab, ko - ha - lik teeb tööd. Mis maa see

25 S. A. T. B. F#m B C# F#m E G#  
on, kas tões - ti üks - nes or - jaks veel ai - nult kül - bab i - ni - me - ne  
Maa, Maa, maa  
Maa, maa

28 A Bm F#m  
siin? Ja kes sel - le va - lu  
L R

30 E D D Bm Bm7 C#4  
üks - kard kok - ku kor - jaks, et tu - leks ar - mas - tus ja lõp - peks piin.  
ja lõp - peks piin.

2 of 6

### III. Prototypes

As a part of this thesis, two prototype version packages of Partiture will be submitted:

- 1) **pariture\_pre\_prototype.zip** – contains simplified logic for demonstrating the key features of the application by leaving aside executing sub-processes like: authentication, creating rooms, joining rooms, accepting users, etc.
  - a. bi-directional page turns
  - b. bi-directional remark sharing
  - c. bi-directional pinpointing

The communication link will be established between the nodes automatically, as soon as the corresponding scores are opened on the devices – an overlay message („Connected“ / „Disconnected“) notifies the application users about the status. After a successful connection attempt, shared page turn and pinpoint events can be triggered using finger movement gestures.

In order to start adding / moveing shared remarks, the button “Edit“ in the top toolbar menu has to be tapped.

- 2) **pariture\_prototype.zip** – contains more complex logic (rooms, sessions, layers) for demonstrating the applications backend functionality aspects.

**PS!** The integration between the backend and frontend logic has not been finalized yet.

In order to be able to run the application prototypes, the following requirements have to be filled:

- the development tool Xcode 5 has been installed<sup>34</sup>
- the base SDK for iOS 7 has been isntalled (by default comes along with Xcode 5)
- the dependency package manager Cocoapods has been installed<sup>35</sup>

In order to be able to run the applicaion prototypes on the tablet devices (iPad’s), one has to be a member of the Apple iOS Development Program<sup>36</sup>

---

<sup>34</sup> <https://developer.apple.com/xcode/> (May, 2014)

<sup>35</sup> <http://cocoapods.org/> (May, 2014)

<sup>36</sup> <https://developer.apple.com/programs/ios/> (May, 2014)

#### **IV. License**

##### **Non-exclusive licence to reproduce thesis and make thesis public**

I, **Kaarel Kruus** (date of birth: 17.10.1987),

*(author's name)*

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
  - 1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
  - 1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

**Partiture: An Interoperable Music Stand Application for Mobile Devices,**

*(title of thesis)*

supervised by Jüri Kiho,

*(supervisor's name)*

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **26.05.2014**