

Improving the user experience of AMCS on mobile devices

Sinthujan Thanabalasingam

19.01.2020

Abstract

Auditorium Mobile Classroom Service (AMCS) is an Audience Response System (ARS) that is actively developed by several individuals at the TU Dresden since 2012. In the educational context, instructors and lecturers use it to enhance and manage their lectures during the semester. One of the main goals of AMCS is to improve the way knowledge is presented and transferred to the audience by offering interactive polls that students can participate in before, during and after a lecture. Several standalone front end applications for platforms like iOS, Android and web are provided that enable the usage of the system. However, because of its ease of access, a majority of students use the system via their web browsers on different mobile devices such as laptops, tablets and smartphones. Regarding usability, design and consistency, the challenge lies in providing a unified and responsive user interface across all supported platforms that is intuitive to use. This work analyzes AMCS and its web interface under these criteria, conceptualizes a redesign strategy that is implemented in a prototype and concludes with an evaluation of the implemented changes.

1 Introduction

1.1 Background

AMCS (Auditorium Mobile Classroom Service) is an Audience Response System (ARS) currently under development at TU Dresden. It represents a practical solution that was initialized and developed in cooperation between members of both the Chair of Computer Networks of the Faculty of Computer Science and the Chair of Psychology of Learning and Instruction from the Faculty of Psychology back in 2012. Ever since its inception, the system serves as a research prototype used to study and evaluate new technologies in the context of ARS. Numerous features were researched, implemented, and evaluated by students and the AMCS group over the years.

In general, the system's main objective is to leverage technology in order to increase the interactivity and adaptability of lectures. By providing interactive polls and evaluation mechanisms, AMCS aims at increasing the audience's engagement before, during, and after a lecture takes place. Overall, several technologies intended to support the interaction between speakers and the audience were developed and tested with the help of AMCS.

In general, the system is built for and aimed at different individual groups of users in the educational context. On the one hand, AMCS provides a set of features that increase engagement in the audience and try to close the gap between lecturer and listener. One of the system's goals is to support a student's learning process by providing interactive polls, question pools, and self-evaluation mechanisms that work on saved answers. While the polls are often used during lectures to collect immediate feedback from the audience, the latter two tools can be used afterward to identify and repeat difficult questions, to prepare for the next lesson, or to study for the examination.

On the other hand, several features of AMCS are designed to support lecturers and instructors in their role. Professors and docents at TU Dresden use it during lectures to get immediate feedback from participating students. More specifically, lecturers are provided with insights on the understanding of their audience in the form of poll evaluation. For example, poor results in a poll covering a particular topic might suggest that the topic was misunderstood or insufficiently explained. AMCS enables the speaker to precisely identify segments of their lectures that students might struggle with and helps them to focus more easily on repeating and emphasizing these topics in the future. Consequently, AMCS is used during the whole semester to prepare and manage lectures and polls, to analyze learning progress and to evaluate feedback given by students.

1.2 Motivation

AMCS is being used actively at lectures that take place in the Faculty of Computer Science at TU Dresden. While several standalone front end applications for different platforms such as iOS, Android and web are provided to the audience, an analysis of user counts has shown that AMCS is used by the majority of students via its web page across different mobile devices such as laptops, tablets and smartphones. The reason for this is likely how easy it is to access, as web browsers are preinstalled on most devices, rendering the installation of the AMCS standalone app as an avoidable extra step.

Regarding the fact that AMCS sees most of its use via the web page, more of an effort should be made to provide a unified and responsive web-based user interface across all aforementioned device types.

Therefore, the motivation of this work is to create a more user-friendly experience by improving the application's front-end in terms of usability, design, and consistency.

1.3 Outline

This work's central goal is to conceptualize a redesign strategy that, once correctly implemented, will improve the experience of users that access AMCS with the browsers on their mobile devices. In order to reach this objective, the work is organized in the following manner:

In the opening Chapter 2, the scope of this work is defined by showcasing different components of AMCS that users interact with most commonly. A short introduction is given to the look and feel of the central elements of AMCS. In Chapter 3, a usability analysis of the components is conducted. The chapter centers around identifying, listing, and categorizing design flaws and issues of AMCS. Chapter 4 analyses solutions to the identified problems of AMCS that other relevant ARS came up with. In Chapter 5, the concept of this work is presented in the form of a redesign strategy. For each component, several improvement proposals are developed and enhanced iteratively. Chapter 6 describes how the strategy is concretely implemented in the form of a prototype. The chapter covers issues that occurred during the implementation and necessary adaptions, changes, and differences to the initial proposals. In Chapter 7, the prototype is used to evaluate the effectiveness of the redesign strategy. A comparison to the current state of AMCS is drawn. Finally, this work is summarized in Chapter 8, concluding with a look at open questions and an outlook on future work.

Contents

1	Introduction	3
1.1	Background	3
1.2	Motivation	4
1.3	Outline	4
2	Current State of AMCS	7
2.1	Web Technologies	7
2.2	Landing Page and Login	7
2.3	Main View	7
2.3.1	Lectures	10
2.3.2	Course management	10
2.4	Poll View	10
2.5	Course View	11
2.6	Menu and Navigation	11
2.6.1	Student	11
2.7	Summary	13
3	Problem Analysis	16
3.1	Problems Of The Mobile View	16
3.1.1	Main View	16
3.1.2	Course View	18
3.1.3	Poll View	18
3.1.4	Burger Menu and Navigation	19
4	Related Systems	23
5	Redesign Strategy	24
5.1	Main View	24
5.1.1	Layout	24
5.1.2	Navigation	26
5.1.3	Drop-Down Menu	26
5.2	Poll View	27
5.2.1	Layout and Visualization	27
5.2.2	Navigation between questions	28
5.3	Course View	29
5.3.1	Navigation	29
5.4	Summary	30

6 Implementation	31
6.1 Main View	31
6.1.1 Drop Down Menu	31
6.1.2 Actions Menu	32
6.1.3 Lecture Tabs	32
6.1.4 Lecture List	33
6.1.5 Iteration 4	35
6.2 Navigation	35
6.3 Poll View	37
6.3.1 Layout	37
6.3.2 State updates	37
6.3.3 Navigation between Questions	37
6.3.4 Rendering of Questions	38
6.3.5 Problems	38
7 Evaluation	39
8 Conclusion	40
.1 Appendix Title	41

2 Current State of AMCS

AMCS provides front end applications for iOS, Android, and web that enable access to a variety of features that benefit both lecturers and the audience. As already established in the introduction, AMCS is a research prototype used to implement and test new components in the ARS context. Several features were added to AMCS in a modular fashion, resulting in UI components that each differ in varying degrees from each other visually. To keep the scope of this work manageable, it will mostly focus on all features that audience members will come in contact with when using AMCS. This section elaborates on the current state of the system by identifying and analyzing all views that allow for access to the different functionalities of AMCS from a student's point of view.

2.1 Web Technologies

The AMCS front end web page is written using Angular¹, a typescript-based front end framework for building mobile and desktop web applications.

2.2 Landing Page and Login

When accessing the website², students will be shown the landing page of AMCS (see Figure 2.1). A big login button is displayed that will reveal a login form (see Figure 2.2) when pressed. In order to use the system, students have to create an account by providing credentials.

Additionally, a subscription to courses is mandatory in order to use the service reasonably. A PIN code identifies each course. By typing in the optional PIN code, students will automatically subscribe to the corresponding course. From thereon, students have access to the system.

2.3 Main View

After successfully logging in, the user is presented with the *Main View* of the system (see Figure 2.3). It can be scrolled in the vertical direction and consists of a header and a body. On top, the header consists of corporate branding on the left side and a burger menu on the right side. Below it, the view's body organizes information in different sections as follows:

¹<https://angular.io> (last access: 01.11.2019)

²<https://amcs.website> (last access: 01.11.2019)

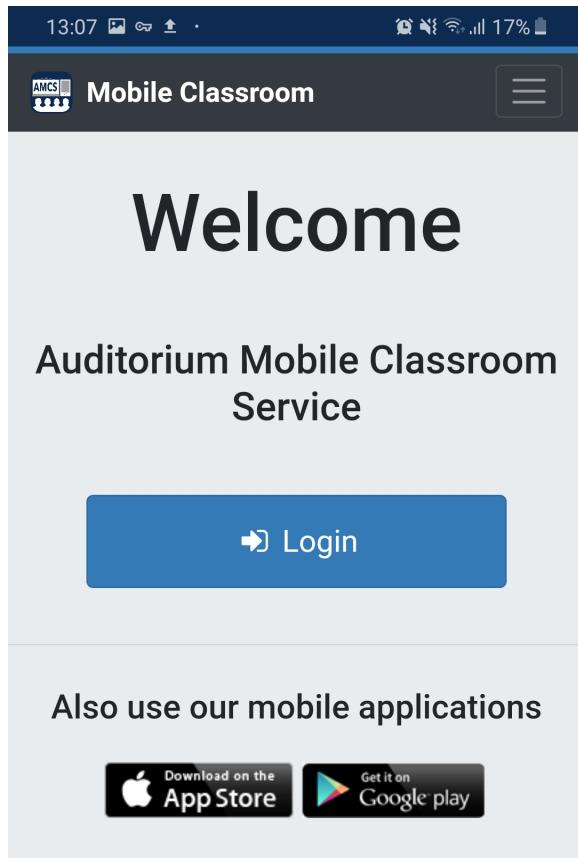


Figure 2.1: The landing page of AMCS. This is the initial screen shown when accessing <https://amcs.website>

The screenshot shows the login form for the AMCS service. At the top, there is a logo for "AMCS" featuring three stylized figures and a smartphone. Below the logo, the text "Please log in" is displayed. A callout box provides instructions for pseudonym creation: "Example for pseudonym creation: • First name of your mother - Silvia • Day of birth of your mother - 1964-04-24 • Own day of birth - 1990-03-06. Pseudonym: SI2406". The main form area contains three input fields: "Pseudonym", "Password", and "PIN (optional)". A blue "Login" button is located at the bottom right of the form. The footer of the screen displays the copyright notice "Auditorium Mobile Classroom Service © 2012-2019 TU" and standard mobile navigation icons.

Figure 2.2: The login form of AMCS. Users can choose a synonym and a password to create an account.

Lectures

Proposed lectures

- Implicit Surfaces II** LIVE
2019-11-10 - 13:00 h 90 min
We talk about implicit surfaces.
Computer Graphics II
- Introduction** LIVE
2019-11-10 - 09:20 h 90 min
Introduction to Application Development.
Application Development for Mobile and Ubiquitous Computing (English)
- Requirements Analysis** AFTER
2019-11-09 - 11:10 h 90 min
Requirements analysis for User Interfaces.
User Interface Engineering

Upcoming lectures (4)

Active lectures (2)

Enrolled courses

Example: PIN_123 Enroll

- Application Development for Mobile and Ubiquitous Computing (English)** DELETE
- Computer Graphics I** DELETE
- Computer Graphics II** DELETE
- Computer Networks** DELETE
- Scientific Visualization** DELETE
- User Interface Engineering** DELETE

Figure 2.3: Main View: Lecture information is provided in different sections for each temporal context. Per default, a section with proposed lectures is expanded.

Figure 2.4: Course management below Lectures: Each course is listed below an enrollment form consisting of a PIN input and a submit button.

2.3.1 Lectures

This section lists all lectures that the user subscribed to (see Figure 2.3). It is organized in subsections that indicate the temporal context of each lecture. These include:

Proposed lectures - Lectures that the system proposes to the user. These lectures either will take place in the next 24 hours or already occurred in the last 24 hours.

Upcoming lectures - Lectures that will take place in the future are shown here.

Active lectures - Lectures that take place right now are shown here.

Past lectures - Lectures that have already taken place are shown here.

Rendering of lectures Each of the subsections mentioned above is organized in a list that contains all corresponding lectures. For each lecture, a box is rendered that uses all horizontal space available to it. The box consists of a blue header with the lecture's name, a white info/detail area, and a light blue footer that contains the course name. A color-coded badge on the top right of the boxes serves as an indicator of the lectures' temporal context. The subsections can be collapsed or expanded by clicking on the sections heading.

2.3.2 Course management

Further down on the page, the section *Enrolled Courses* can be found (see Figure 2.4). It serves two purposes: Primarily, it provides a way to enroll in a course. An enrollment form, shown for this purpose, consists of a text field to enter the course PIN and a blue button that triggers the enrollment. When provided with a valid PIN, pressing the button redirects the user to the *Course View* (described in Section 2.5). Secondly, the view serves as an overview of a students' courses. They are rendered as light blue buttons in a vertical list. A trash can icon on each button provides a way to leave the given course.

add graphic of badges

2.4 Poll View

One of the main functionalities of the system that users will engage with is answering polls. Polls can be reached by clicking on a lecture box from either the *Main View* or the *Course View*. Each poll consists of a set of questions the user can answer. They are rendered in a view that is reused by the system depending on the situation and context. This means that the view might only be accessible under certain circumstances, for example, when the lecture reaches a specific point in time, making it a slide poll (SP). SPs are shown when a specific slide is on display and can only be answered at this very moment. Other types of polls include "global" course polls (CP) that are always accessible during the semester and lecture polls (LP), which can only be answered during the lifetime of a lecture. Active polls are displayed all at once in sections designated to each poll type. The different types of polls that occur in AMCS are further summarized in Table 2.1.

Poll Type	Explanation
Slide Poll (SP)	Active when a specific slide is shown. Commonly used for quizzes after a difficult section in a lecture to make sure that students understood everything correctly.
Preparation Poll (PP)	Active before the lecture takes place. It is commonly used to instruct students to prepare for a certain topic.
Lecture Poll (LP)	Active during the lifetime of a lecture.
Post-Processing Poll (PPP)	Active after a lecture has taken place. Commonly used to check gained knowledge.
Course Poll (CP)	Active during the whole lifetime of the course (commonly during the whole semester).

Table 2.1: Different poll types that the user might encounter when using AMCS.

If no polls for a given lecture are available, the user is presented with the hint shown in .

add graphic here

2.5 Course View

The *Course View* is shown when the user selects one of the courses from the *Course Management* section (see Figure 2.5 and Figure 2.6). Its purpose is essentially to provide a filtered view on the lectures of a single course. The course name and PIN, its description and lists of upcoming, live and past lectures are visible from top to bottom in this order. It reuses the lecture section component described in Section 2.3.1.

2.6 Menu and Navigation

Besides using the *Main View*, additional functionality can be reached by navigating the *Burger Menu* that is shown in the upper-right corner of the screen. It reveals a sub-menu which expands vertically, offering three additional sub-menus (see Figure). In the following, these sub-menus and their functionality are briefly explained.

add section here

2.6.1 Student

This is one of the most important buttons that connect a subset of the main functionalities of AMCS. Upon pressing this button, the menu expands again vertically, showing a list of further sub-menus. Most of the functionalities shown in this list will be touched by the proposals for improvement that are presented in Chapter 3. The functionalities in questions are:

1. Evaluation of answers
2. Question Pool
3. Edit account



← Computer Graphics II (CG_2_2019)

Students have to make sure that their solution is handed in at the submission date. Theoretical submissions must be handed in in written form on A4 paper by 16:00 to the chair staff. We recommend handing in the solutions at the end of the previous lecture. Practical submissions must be uploaded via Opal by 23:59. To get points for their practical submission, each team has to present their work to a tutor on the evaluation date. For this evaluation, each team gets an individual appointment with their tutor (Friday, 3rd - 5th DS). Note that for technical reasons, every team member has to upload a copy of their solution to OPAL in order to get scored.

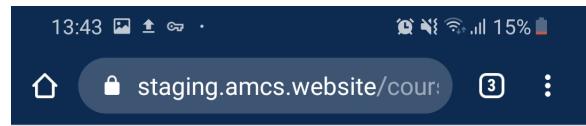
Overview of all lectures of Computer Graphics II. Click on one of the lecture to open the detail view.

Upcoming lectures (1)

Implicit Surfaces II



Figure 2.5: Course View: Details like the course name, description and PIN are displayed.



Overview of all lectures of User Interface Engineering. Click on one of the lecture to open the detail view.

Upcoming lectures (2)

Sketching and Prototyping

2019-11-23 - 09:20 h 90 min
Sketching and prototyping

History of User Interfaces

2019-11-16 - 11:10 h 90 min
The history of User Interfaces.

Past lectures (1)

Requirements Analysis

2019-11-09 - 11:10 h 90 min
Requirements analysis for User Interfaces.

Auditorium Mobile Classroom Service © 2012-2019 TU Dresden.

Figure 2.6: Lectures that belong to a certain course displayed in the Course View.

Evaluation of Answers

Once students have participated in a poll and answered a few questions, they can evaluate their answers by selecting this option. Two drop-down-menus are shown prompting the student to select the course and lecture of interest. After choosing an item from the list, a vertical list of all questions that occurred during this lecture is shown to the student, together with indicators for the given answers. If the student used two attempts to answer the question, a toggle button is provided to switch between the first and the second answer.

add graphic here

Question Pool

By selecting this option from the *Burger Menu*, the student is offered the possibility to create collections of already answered questions. The intent is to provide a way for students to collect and repeat questions that they had difficulty in answering. Similar to , the student is prompted with a drop-down menu to select a course of interest. After selection, the student is presented with a list of all lectures and their polls, respectively. All questions of each poll are grouped and shown to the student in a vertical list. From this list, the student can select all questions that they might be interested in to create a pool of questions. These pools are then processed into polls that the student then can answer again. These polls are rendered in the same manner as stated in Section 2.4.

add section here

How It Works

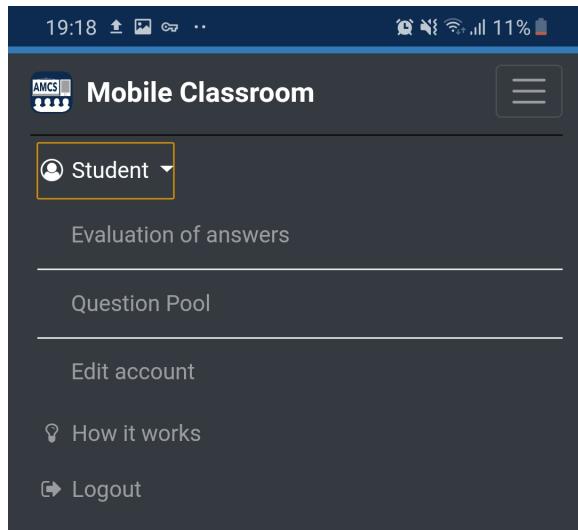
Pressing this button will redirect to a page that shows tutorial instructions on how to use AMCS. This help page is rendered identical on all mobile devices and therefore falls out of the scope of this paper.

Logout

As the name already states, pressing this button will logout the user and end the session. If logging out was successful, the *Landing Page* is displayed.

2.7 Summary

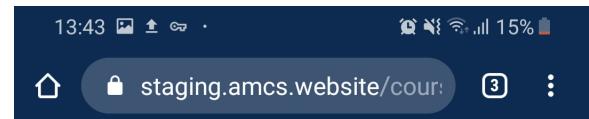
The identified UI components that will be further analyzed and discussed are summarized in the following table:



Lectures

💡 Proposed lectures

Implicit Surfaces II	LIVE
📅 2019-11-10 - 13:00 h	⌚ 90 min
📄 We talk about implicit surfaces.	
Computer Graphics II	
Introduction	LIVE
📅 2019-11-10 - 09:20 h	⌚ 90 min
📄 Introduction to Application Development.	
Application Development for Mobile and Ubiquitous Computing (English)	



Overview of all lectures of User Interface Engineering. Click on one of the lecture to open the detail view.

⊖ Upcoming lectures (2)

Sketching and Prototyping

BEFORE

📅 2019-11-23 - 09:20 h ⌚ 90 min

📄 Sketching and prototyping

History of User Interfaces

BEFORE

📅 2019-11-16 - 11:10 h ⌚ 90 min

📄 The history of User Interfaces.

⊖ Past lectures (1)

Requirements Analysis

AFTER

📅 2019-11-09 - 11:10 h ⌚ 90 min

📄 Requirements analysis for User Interfaces.

Auditorium Mobile Classroom Service © 2012-2019 TU Dresden.

Figure 2.7: Expanded Burger Menu. The *Lecture List* below is displaced significantly.

Figure 2.8: Lectures that belong to a certain course displayed in the *Course View*.

Component	Description
Landing Page	Shown when the web page is opened. Points to the Login mechanism.
Main View	Gives an overview of ongoing lectures and enrolled courses. Allows to enroll to or unsubscribe from courses.
Poll View	Shown whenever a poll should be answered.
Course View	Displays information and lectures of a single course.
Burger Menu	Overarching navigation element visible in all views.
Question Pool	Shown when the creation of a question pool is attempted. It can be reached from the <i>Burger Menu</i> .
Evaluation of answers	Shown when the evaluation of answers to polls is attempted. It can be reached from the <i>Burger Menu</i> .

Table 2.2: Summary of UI Components discussed in this work.

3 Problem Analysis

This chapter analyzes the current state of the application and identifies problems that occur when using AMCS via a web browser on mobile devices. The system will be analyzed in terms of typical usage by an audience member to narrow down the extent of this work. A characteristic instance of an audience member could be a student who uses AMCS on their smartphone. While doing so, students will interact mostly with the *Main View*, the *Poll View*, and the *Navigation / Burger Menu*. Therefore, this work is centered around but not limited to these components (see Table 2.2).

3.1 Problems Of The Mobile View

The back end of AMCS reacts to requests coming from mobile devices such as smartphones or tablets by providing a responsive mobile view to its clients. However, in some aspects, AMCS struggles to offer a consistent UI experience that guarantees the best usability possible. One challenge lies in the fact that the system has to deal with limited screen space to visualize information with maximum effectiveness. Decisions have to be made on the size and placement of different UI components, depending on the information that they should convey to the user.

User interaction likewise plays a big role. Some users might approach the application with different ways of interaction and navigation that are characteristic of mobile devices. For example, a smartphone user might expect to be able to use swiping gestures to navigate a menu. The user could also expect that information is organized in views consisting of separate tabs, which is a typical technique used to display much information on limited screen estate.

This section lists key issues that lower usability or might confuse mobile users, as described above. A tabular summary of the findings is found in Table 3.1.

3.1.1 Main View

As already described in Section 2.3, the *Main View* relies on a vertically scrolling list view, consisting of different sections. While this is the most intuitive design to chose for use on smartphones, details of the implementation in place cause problems and impair the usability of the application.

General Visualization Problems

Section 2.3.1 covered the fact that lectures are rendered by displaying the title of the lecture in the top section of the box using white letters on a solid blue background. Details about the lecture, such as time, duration, and a textual description, visualized in gray letters and icons on a white background, follow below. Finally, at the bottom of the box, the course name is shown in gray letters on a light blue background. The order *lecture name, details, course name* can cause confusion. The most coarse grain piece of information - the course name - is displayed at the bottom of the box rather than at the top. Generally, when seeking information about active lectures, a student will most likely remember the course name rather than the name of a single lecture, as timetables used by students only contain course names. Therefore, displaying the information in this order could lead students to take a longer time to find the pieces of information that they are seeking.

In addition to that, background colors and font sizes used are inappropriate to differentiate between course name and lecture name, further increasing the ambiguity described above.

To sum it up, the hierarchical and logical relationship between courses and lectures remains unconsidered by the visual representation of course and lecture information.

No Notifications For New Or Unread Content

A commonly used technique to lead the user to new content is an *Unread Content Indicator* (see facebook indicator). The Main View lacks completely of indicators of any kind for new or unread content. The absence of such indicators is a problem because, typically, not all polls are visible to students at the beginning of a semester. One reason is delayed or planned activation of polls. Lecturers can activate their polls at a preferred point in time. To see whether, for a given course or lecture, new polls exist, students have to tap on the lecture to check. This process is unintuitive and adds another layer of indirection to the overall workflow.

add graphic here

Indirection Problems

The boxes that represent each lecture claim a lot of screen space in relation to the information that is displayed to the student (see Figure 2.3). The layout causes plenty of indirection because, per default, the sections for upcoming and active lectures are expanded fully. This might be handy when quickly gathering information about lectures that are or soon will be active, but in every other case, it slows navigation and overall interaction because the *Course Management* section is pushed down to the bottom. A list of only four boxes causes a scroll bar to appear on the widespread screen resolution of 1920x1080 pixels. Therefore, a student who wishes to enroll in a new course has to scroll to the bottom of the *Main View* before they reach the *Enrollment Form*. Similar problems occur when merely looking up current course enrollments or when trying to leave a course altogether.

Furthermore, if a student seeks information regarding a specific course, no filter or search functionality is offered by the lecture list. Instead of typing the course name in a search bar, they have to scroll down to the bottom of the lecture list, scan the course list manually with their eyeballs, find the course and click on the corresponding item. After that, they are redirected to the *Course View*, which then displays a filtered list of lectures belonging to the course. The more courses the student enrolls in, the further the level of indirection increases.

Redundancy

Some visual redundancy is added by the badges that reside on the upper-right corner of each lecture. These badges help to visualize the temporal context of the lecture for each item in the corresponding section. It seems that the badges exist as an additional means to convey the temporal context of the lecture because sorting the lectures in their respective section alone fails to do so. However, the badge's names do not match the section's names, e.g., an upcoming lecture's badge reads *BEFORE* instead of *UPCOMING*.

Some visual redundancy exists in the *Course Management* section below. Each course consists of a box with the course's name along with an unsubscribe button, represented by a trash can icon (seeFigure 2.4). This way of rendering the courses causes redundancy, as repetition (especially of the trash can icon) adds noise to the overall look. Plenty of the vertical screen estate is wasted in this manner.

3.1.2 Course View

Since the *Course View* reuses the lecture list along with the sections *Upcoming lectures*, *Active lectures* and *Past lectures*, likewise the same issues arise as for the *Main View*, as already described in Section 3.1.1. The *Course View* has a vital function in terms of usability due to it acting as a filter for lectures that belong to a certain course. Problems arise when a user wants to switch quickly between different courses. Doing so requires to leave the *Course View* by tapping the back button and then scanning the *Course Management* section for the element of interest, which is slow and cumbersome. This is explained in more detail in Section 3.1.4

3.1.3 Poll View

Visualization

Section 2.4 describes the rendering of questions as boxes that are aligned in a vertical scrolling list. Namely, the extensive use of vertical space on the screen is one problem introduced by this layout. Longer polls that consist of multiple questions unnecessarily take plenty of vertical screen estate. Answering one question usually does not require to see the neighboring questions, but most of the time, two to three questions are in view simultaneously (see Figure 3.2), potentially causing a distraction to some students.

The view also lacks basic but potentially important information such as the number of total and remaining questions. This information might be useful during longer polls for students who want to gain an idea of how many questions are left.

In general, the layout lacks separation and distinction between individual types of polls. Polls of the same type are separated by a heading that denotes the poll type separates all polls of the same type. These groups of polls are then simply appended to the bottom of the list, increasing its length even further.

Local Navigation

The vertical list is difficult to navigate as it requires scrolling between questions. If a student wants to jump from the first to the last question, or vice versa, several swiping gestures are needed to reach the top or the bottom of the list. Similar to the lecture list described in Section 2.3.1, the question list is also segmented into different sections. Lecture and course questions are similarly appended to the bottom of a *Slide Poll*. Students who wish to view these

add
graphic
here

questions must scroll to the bottom of the list. Due to the layout in place, indirection increases further, and overall interaction slows down.

3.1.4 Burger Menu and Navigation

At the time of writing, the ways of navigating the application are problematic and partly confusing. Several layers of indirection introduce problems and may worsen the user experience. Figure 3.1 illustrates click paths a user must take in order to reach different views (illustrated in blue) within AMCS. In general, some views are connected via the *Burger Menu* as the overarching element of navigation (illustrated in green). In contrast, other views are interconnected more tightly. They are reached by clicking on elements inside a view, such as a course or a lecture. The following paragraphs elaborate more on both aspects of navigation.

On the one hand, the interconnected graph of views, as described in Figure 3.1, contains two bigger issues. One example of unexpected behavior is the path that a user takes when he wants to return from the *Poll View* to the *Main View* in order to choose a different poll. It is possible to reach the *Poll View* from the *Main View* with only one tap, for example, by selecting a lecture. However, by pressing the back button, the user is first redirected to the *Course View* and then, with a second tap on the back button, to the *Main View*. A more well-defined implementation would rather directly return to the last element of the *View Stack*, in this case, the *Main View*.

Furthermore, some views like the *Question Pool View* and the *Answer Evaluation View* do not offer buttons that allow navigating back. Consequently, the navigation relies partly on the corporate branding on the upper-left that, when tapped, will redirect to the *Main View*. One could argue that the interconnectedness between different views might not be strong enough.

On the other hand, the *Burger Menu* that connects several aspects and functionalities of AMCS in an overarching manner (as it is part of every view) has some noteworthy issues. One visual problem that arises is the fact that the menu excessively uses vertical screen space and dislocates the rest of the content currently shown when expanding several sub-menus. A reason why this is problematic might be the following scenario: a student wants to evaluate their answers to polls for a particular lecture. When opening the menus on their phone from the *Main View*, information like course name and lecture title are pushed down by the menu, potentially entirely out of the viewport. However, this information is required in the *Answer Evaluation View* because the user has to choose their course and lecture of interest from two drop-down menus. These additional steps could lead to users having to return to the *Main View* to look up the lecture name or other details again so that they can proceed with their selection.

An additional, different issue arises from the labeling of menu entries. The labels can confuse students because the first menu entry is labeled as *Student*. Such labeling implies in general that different user roles exist in AMCS, which is the case. However, the user's role should not be the label of a sub-menu in the navigation. While functionality like *Account Management* might be a plausible function to be found here, it is somewhat unintuitive that the *Question Pool* and *Answer Evaluation* can be accessed via a button labeled *Student*.

Evaluation of answers

As mentioned above, pressing the option *Evaluation of answers* in the expanded *Burger Menu* leads to a view with a drop-down menu from which students can choose a course of their interest. Afterward, a list of expandable items is shown, where each item represents a lecture.

Clicking on one or multiple of these items will expand a vertical list of questions similar to the regular question list described in Section 3.1. Likewise, answers given by the student are shown as well (see Fig.). Multiple problems occur on this view: First of all, the navigation path to reach this view contains unnecessary indirection and might not be intuitive enough, which is illustrated by Figure 3.1. Students might expect this functionality to reside at the *Main View* attached to the elements of the course list or inside the *Course View* itself. Instead, every time evaluation of given answers is attempted, this functionality can only be accessed by using the *Burger Menu*, choosing the appropriate item from the sub-menu, selecting the course in question, and afterward expand the lecture and the corresponding question list. Moreover, the question list suffers from the same rendering and navigation problems already described in Section 3.1.3. Questions are poorly navigable, and a good amount of scrolling is required to jump between questions.

add
graphic
here

The screenshot shows the 'Mobile Classroom' application's poll view. At the top, there's a header with the 'Mobile Classroom' logo, a back arrow, the text 'Lecture 1', a green circular icon with a question mark, 'Slide 1 / 30', and a menu icon. Below the header, there's a section titled 'Answer questions' with a 'Slide questions' link. Three questions are displayed vertically, each in its own blue-bordered box:

- Another question**: What is going on?
 I don't know
 Me neither
 Third answer
 4. possibility
[Answer](#)
- Another question**: What is going on?
 I don't know
 Me neither
 Third answer
 4. possibility
[Answer](#)
- Another question part two**: What is going on?
 I don't know
 Me neither
[Answer](#)

Question Pool

The question pool suffers from the same navigation problems described in the preceding section. Again, a drop down menu for selecting a course is shown before students can see the overview of the *Question Pool*. Once more, students might think that access to this functionality is located near the *Main View* or the *Course View*, which is not the case.

Figure 3.2: *Poll View*: Multiple questions are displayed at the same time.

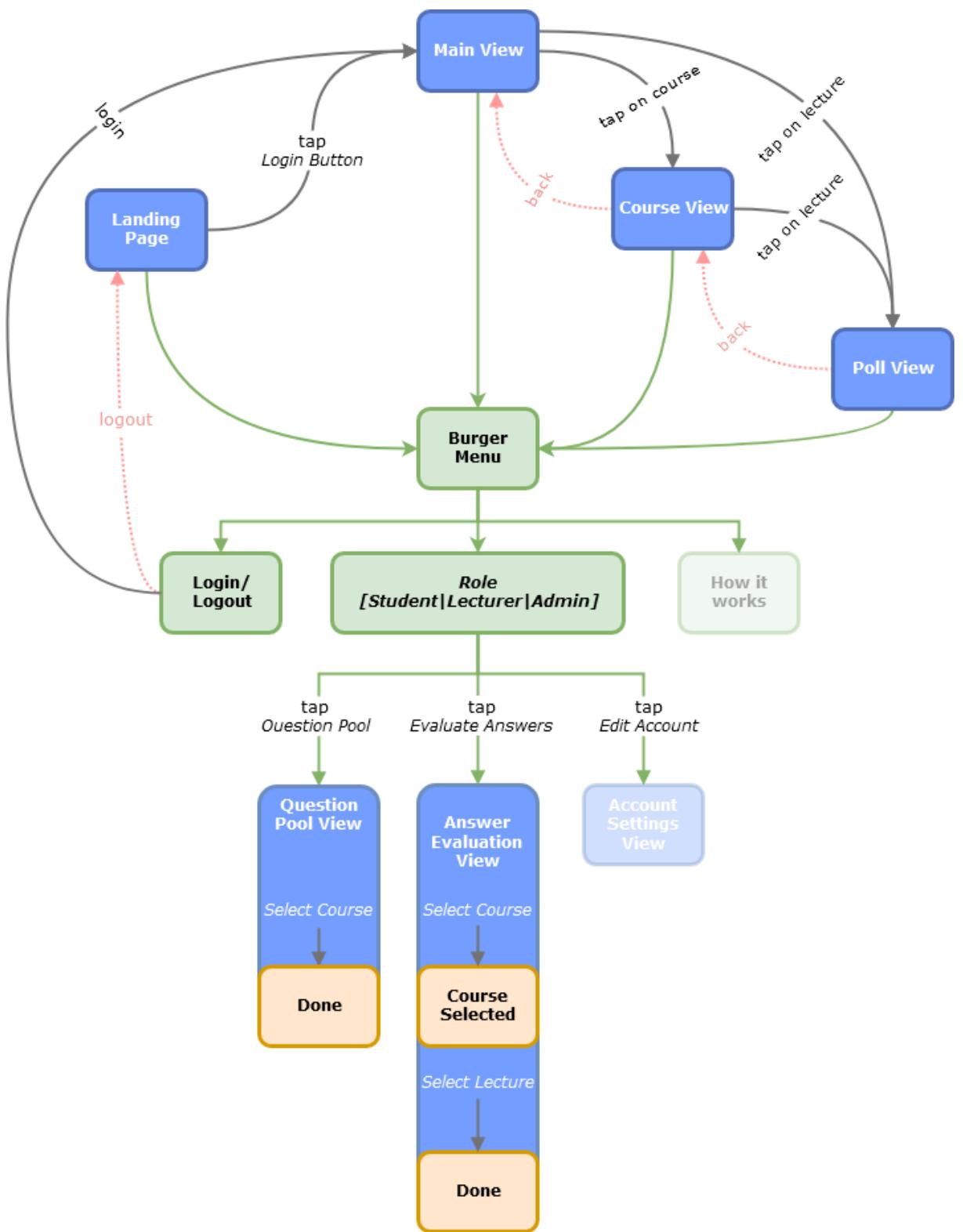


Figure 3.1: Navigation concept of AMCS: Every arrow represents a tap / click the user has to do to reach the desired destination.

ID	Name	Categories	Views	Summary
GV 1	Visualization of Lectures	Visualization	MV	The hierarchical order of courses containing lectures is disregarded.
GV 2	Missing Notifications	Visualization	MV, CV	No notifications for new or unread content are given.
GV 3	Lecture Badges	Visualization, Redundancy	MV, CV	Lecture badges that indicate temporal context are redundant and do not match their respective section in the lecture list.
GV 4	Course List	Visualization, Redundancy	MV	The Course List is placed at the very bottom of the MV. The list contains redundant visual elements.
IN 1	Course Management	Layout	MV	The section <i>Enrollment</i> and the course list are located too far to the bottom of the view.
IN 2	Course Filter	Functional	MV	The view lacks of a filtering mechanism.
PV 1	Poll View Content Displacement	Visualization Layout	PV	The poll layout uses too much vertical space.
PV 2	Missing Poll Information	Visualization	PV	The number of total and remaining questions is missing.
PV 3	Missing Poll Separation	Layout, Navigation	PV	The view lacks of separation between different poll types. Navigation between different poll types requires too much scrolling.
NV 1	Switching Courses	Navigation	MV, CV	Quickly switching between courses requires unnecessary navigation between CV and MV.
NV 2	Confusing Click Paths	Navigation	MV, CV, PV	Under certain circumstances, returning to an earlier view can yield unexpected results.
NV 3	Weak View Inter-connectedness	Navigation	MV, QPV, AEV	Some views miss proper back buttons to return to the previous view.
NV 4	Menu Size	Visualization Layout	BM	Fully expanding the BM uses too much screen space.
NV 5	Menu Labeling	Visualization	BM	Some menu entries are labeled inadequately.
NV 6	Evaluation of Answers	Navigation Layout	AEV	Several drop down menus must be operated to evaluate answers for a given lecture.
NV 7	Question Pool	Navigation	QPV	The QPV is not reachable easy enough, as expanding the BM is required.

Table 3.1: Classification of issues identified in the usability analysis of AMCS. MV = Main View, CV = Course View, AEV = Answer Evaluation View, PV = Poll View, BM = Burger Menu, QPV = Question Pool View

4 Related Systems

5 Redesign Strategy

Several weaknesses and flaws of the web view of AMCS have been identified and analyzed in the previous chapter. They range from issues regarding visualization, layout and space usage to problems with user navigation. A set of proposals that aim at solving these problems is introduced in the subsequent chapter. The focus will predominantly lie on using the available screen space more efficiently, improving local navigation inside polls and global navigation between different views and reducing indirection to a minimum.

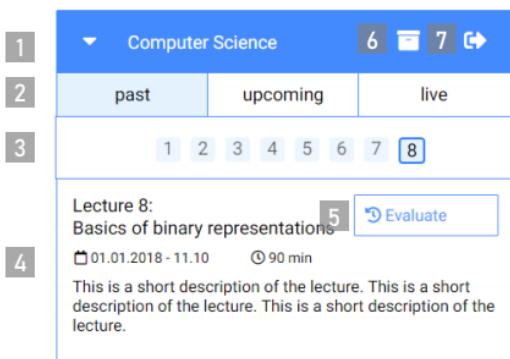


Figure 5.1: Mock-up 1 of the new *Main View*: Currently, the *past courses* tab is shown. A course that has already taken place is selected. Usage of drop down menus and tabs to reduce the amount of vertical space used. (5), (6) and (7) serve as buttons to evaluate given answers for the shown lecture, go to the question pool of this course and unsubscribe from the selected course respectively.

in this section. Figure 5.1, Figure 5.2 and Figure 5.3 respectively show an evolution of mock-ups for the *Main View*.

5.1.1 Layout

Iteration 1 As mentioned in Section 3.1.1, the *Main View* suffers from using the available vertical space not effectively enough. Most noticeably, the course overview and *Enrollment Form* are placed below the *Lecture List*. In order to find information about relevant courses

Each proposal for itself is centered around improving usability aspects of the application while all proposals as a whole aim at keeping the interface consistent and recognizable across all supported platforms. As already mentioned in Chapter 2, AMCS was extended in a modular fashion over the years. This resulted in UI components that each differ in varying degrees from each other visually. No redesign of all modular components is attempted, but rather a redesign of the overall framework in which these components are embedded, narrowing down the focus of this work even further. The result of this chapter is summarized in Table 5.1.

5.1 Main View

Several improvements for the layout and visualization of the *Main View* are elaborated

or to enroll in or leave a course, students are required to scroll all the way to the bottom. Therefore, one proposal is to compress this view by using drop-down menus and tabs. Figure 5.1 represents a mock-up of the proposals described in the following. First of all, the view is restructured to follow the hierarchical concept of courses containing lectures (LV): In the top (1), a button for a drop-down menu is shown next to the currently selected courses' name and two additional buttons (6) and (7). The purpose of the drop-down menu and the buttons is explained later. Following the heading, tabs for *past*, *upcoming* and *live* lectures are laid out side-by-side (2). In the first two mock-ups (see Figure 5.1 and Figure 5.2 respectively), the tabs are followed by the *Navigation Bar*, a numbered horizontal list of clickable dots (3) that each represent one lecture. The currently selected lecture is highlighted with a bold blue border to enhance visibility and orientation. Finally, the information section of the view follows (4) with the title of the lecture, time and duration details and the lecture description. In the details section, an additional button is placed (5) that is labeled as "Evaluate". Another idea would be to hide or deactivate this button for lectures that have not yet entered the state *past*. This layout uses a considerably reduced amount of vertical space. The placement of the "Evaluate" button (5) is motivated by reducing vertically occupied space even further. However, it could be reasonable to place it below the description text of the lecture. On most devices, the proposed layout will reduce the amount of scrolling required. In an effort to keep visual clutter and redundancy at a minimum, the badges that indicate temporal context of the lectures now miss completely. The view will solely rely on the sorting of lectures in their respective tabs, meaning that the tabs themselves visually convey the temporal context of a lecture.

Iteration 2 After collecting feedback from the AMCS group, the mock-up was adjusted slightly (see Figure 5.2). The textual description of the tabs is replaced with iconography. Furthermore, *Notification Bubbles* are introduced to indicate new or unread content. The *Evaluate Button* was moved to a more conventional location at the bottom of the view. The number of simultaneously displayed lectures remains at one.

Iteration 3 Regarding the strict reduction of vertical space used by the layout, one might suggest that the new *Main View* has become too compact. Feedback from the AMCS group led to the realization, that multiple lectures should be visible at the same time for the reason that typically multiple lectures are taking place on the same day. Therefore, the next iteration expands the layout again to display three lectures at once (see Figure 5.3). A page view concept is proposed that tries to find a compromise between minimizing scrolling and maximizing the amount of information displayed at once. The location of the *Evaluate Button* is moved again to the top portion of a lecture box.

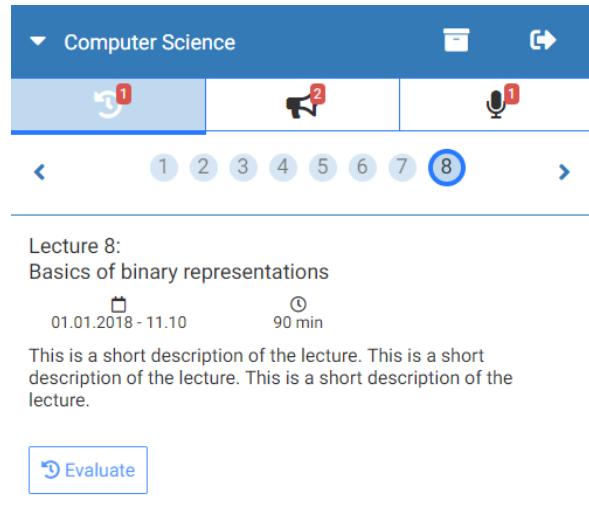


Figure 5.2: Mock-up 2 of the new *Main View*: Textual descriptions for tabs are replaced with iconography, notification bubbles for unread content are introduced. Colors are adjusted to match the corporate design of AMCS.

5.1.2 Navigation

Lecture Tabs (LT) will separate lectures by their temporal context. Selecting a tab will only display lectures that share the respective temporal context, meaning that it should be easier to switch between *past*, *upcoming* and *live* lectures. The *Pagination Bar (PB)* at the bottom of Figure 5.3 is used to ease navigation between lectures that share the same temporal context. A student can use the bar to switch quickly between the oldest and newest past lecture by selecting the corresponding

The mock-up displays a list of four lectures under the course 'Computer Science'. Each lecture card includes a title, a short description, a date and time, a duration, and an 'Evaluate' button. The first three lectures have a red 'C' icon next to the 'Evaluate' button, indicating they are past lectures. The fourth lecture has a blue 'C' icon, indicating it is an upcoming lecture. The header features a 'Mobile Classroom' logo and tabs for 'past lectures', 'upcoming lectures', and 'live lectures'. The footer contains a pagination bar with numbered buttons (1, 2, 3, 4) and navigation arrows.

Figure 5.3: Mock-up 3 of the new *Main View*:The icon indicating an ongoing lecture for a course is brought back.A combination of textual description and iconography is used for tabs.Multiple lectures are displayed at once in a page-based view.

As previously described in Section 2.6.1, the *Evaluation of Answers* requires the user to select a course and a lecture of interest from two drop-down menus. The *Evaluate Button* will eliminate the need to select both the course and lecture of interest. The information required for the back-end request that returns the answers for a given lecture is present. The idea is therefore to reuse the existing *Evaluation of Answers* component, but have the click on the *Evaluate Button* select the course and lecture for the user. This way, two extra steps are removed from the established workflow. All three buttons try to reduce the indirection introduced by the *Burger Menu* to a minimum. Features associated with a course or lecture will be accessible from a view that deals with courses or lectures respectively. The *Burger Menu* will then only have to deal with the profile editing and logout functionalities.

button from the bar. This should improve navigation within the *Main View* as well as in the context of a lecture. Additionally, this proposal aims at removing indirection in the global navigation context by providing certain buttons that serve as shortcuts for the functionality currently found in the *Burger Menu*. As illustrated in Figure 5.3, button (A) serves as a shortcut to the *Question Pool* for the selected course. On click of button (B), the student will be removed from the selected course. Both of these buttons are placed in the header of the layout next to the course's name to indicate that both referenced functionalities operate in the context of courses, whereas the "Evaluate" button (C) operates on the context of lectures (FL).

5.1.3 Drop-Down Menu

The *Drop-Down Menu (DDM)* is introduced to help reducing usage of vertical space even more (see Figure 5.4). Clicking on it reveals its two functions: Besides the now embedded *Enrollment*

Form (EEF), a list of courses a student is already enrolled to is shown below. When selecting a course from this list, only corresponding lectures will be shown in the *Main View*. The *Enrollment Form* still consists of a text field and a button. Both elements are shown next to the text *Enroll....*. The close proximity to the list of courses could make the *Enrollment Form* potentially more traceable to users. The *Enrollment Form* will only be embedded in the drop down menu when the student is enrolled in at least one course beforehand. Otherwise, if a student has not enrolled in any course yet, in place of the *Main View*, only the *Enrollment Form* will be shown. In summary, the *Drop Down Menu* acts as a coarse grain filter to the *Main View* and essentially covers the responsibilities of the *Course View* that is currently in place.

5.2 Poll View

5.2.1 Layout and Visualization

Several issues have been identified regarding layout and visualization of polls in Section 3.1.3. Main problems include the ineffective use of vertical space and a lack of visual separation between poll types. Regardless of their type, polls are simply appended to the bottom of the list. The view therefore grows unnecessarily in size. In order to solve these issues, a tab-based layout in the form of *Poll Tabs (PT)* is proposed once more. Figure 5.5 shows the first mock-up iteration for the redesign of the *Poll View*.

Iteration 1 Beginning at the top, the course's name is displayed in white font on a blue rectangle (1). Following up, to separate the type of polls from one another, a *Tab Menu* is used (2) to differentiate between *Slide Polls*, *Lecture Polls* and *Course Polls*. The tabs are arranged from left to right depending on the poll's lifetime. The most short lived polls, the *Slide Polls* are placed on the left, the *Lecture Polls* take advantage of the middle and the *Course Polls* are displayed to the right. Selecting one of the tabs will cause the layout to show only polls of said type, making them act as a filter to what is currently displayed. This will potentially improve the effectiveness of vertical space used significantly. Below the *Tabs*, a *Navigation Bar* (3) is displayed that contains the question's topic, index and the total number of questions. In case of a slide poll, the current slide number is shown additionally (see Figure 5.5). Furthermore, the *Navigation Bar* introduced in Section 5.1.2 is reused here (4). It serves as a means to navigate between questions more easily and faster. However, the *Navigation Bar* also reduces vertical space used significantly. Only one question at a time is displayed to the student.

Appropriate colors and icons are intended to convey information more efficiently. A blue bold border is used to indicate the current question selected in the *Navigation Bar*, light blue dots signify, that the corresponding question has not been answered yet, whereas bold green or red dots indicate correctly and wrongly answered questions respectively. Below the *Navigation Bar*, only one question at a time is displayed to the student to avoid visual noise and clutter (5). The visualization of questions (*VQ*) is handled as follows: The question is displayed in a blue rectangle with white text. Below the question, an instance of an answering mechanism

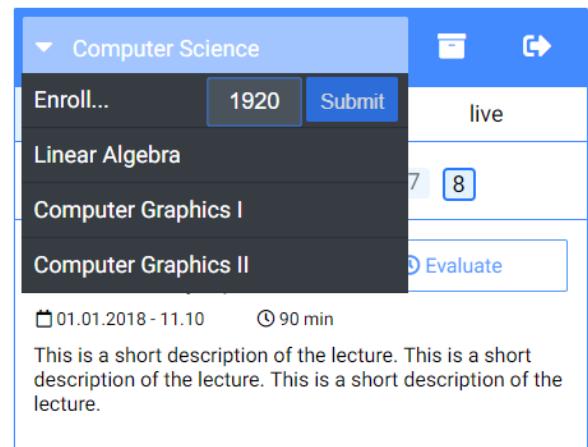


Figure 5.4: Mock-up of the new *Embedded Drop Down Menu*: The enrollment form is integrated in the menu. Users can change the currently selected course from here. The *Main View* then adapts accordingly, allowing the menu to act as a filter for courses.

is displayed (6). Currently, each question is answered individually by selecting the option and then pressing the blue *Answer* button (see Figure 3.2). Afterwards, feedback is shown immediately to the student. In terms of usability, users might find this tedious and redundant. One idea that comes to mind is to send the answers of a poll collectively in bulk to the server at the end of a poll. An advantage with this approach is the reduced amount of requests sent to the server. However, AMCS follows a rather strict principle of providing immediate feedback. Students should directly be informed about the correctness of an answer. Therefore, in the case of SC-, MC-, SCC- or MCC-questions, the button to answer the question is omitted and merely selecting an option will trigger a request to the back end server. Wrong answers are highlighted as before in red, a correct answer in green and it will still be possible to answer twice. Finally, space for textual feedback is given in a box (7) below the answers. This view is reusable and can therefore also be used to display already answered questions when using the *Evaluate answers* functionality. This view will likewise profit from the reduced amount of vertical space used.

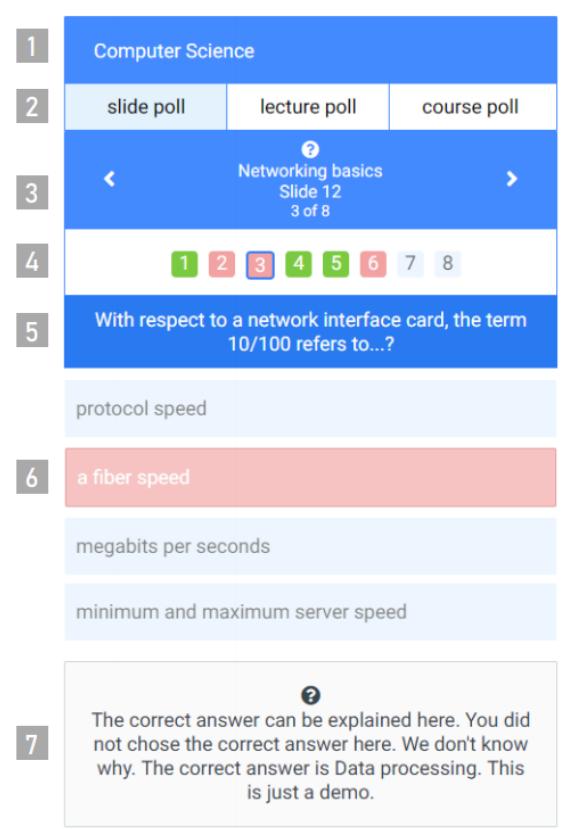


Figure 5.5: Mock-up 1 of the new *Poll View*: The icon indicating an ongoing lecture for a course is brought back. A combination of textual description and iconography is used for tabs. Multiple lectures are displayed at once in a page-based view. introduced.

Iteration 2 Similar to the later iterations of the *Main View* described earlier, the mock-up was adapted to comply with the corporate design of AMCS (see Figure 5.6). The tabs at the top now use textual descriptions and iconography simultaneously to convey their meaning more efficiently, as feedback by the AMCS group led to the assumption that icons alone are not recognizable enough. The tabs are further enhanced by the inclusion of *Notification Bubbles* (**NB**) that indicate the amount of unanswered question in each poll category, potentially reducing the cognitive effort to find unanswered questions. The *Navigation Bar* is further enhanced by using icons that represent the state of a specific question. Green arrows and red crosses are used to visualize correctly or wrongly answered questions respectively to improve accessibility, especially for colorblind students. In order to separate navigation from content, the buttons that allow to jump between questions are moved to the *Navigation Bar*, as they resided previously next to the topic and question. The indicator for the current slide number is modified slightly and pushed to the left. On the right side, the icons that differentiate between question types are reintroduced.

5.2.2 Navigation between questions

Navigation between questions should be made easier for students and focus on one question at a time. Therefore, an improvement would be to introduce two buttons in the *Navigation Bar* (**NVB**) that can be used to navigate one question forward or backwards. Pressing the respective

button will cause to show the next or previous question, regardless of whether the current question has already been answered. This leads to the same level of freedom when navigating polls that the current state of the application allows. Swiping (SW) is a widely spread way of interacting with a user interface on smartphones or tablets. Consequently, a student might expect to be able to use these gestures while using AMCS. Therefore, navigating between questions should be possible by swiping left to go forward or right to go backwards. The combination of swiping and the provision of buttons for navigation, especially next and back buttons (NBB), enhances usability while respecting different platforms and device types. Students on smartphones and tablets will have buttons and swipe gestures simultaneously available to them, while users on laptops and computers without touchscreens can use the buttons. In addition to that, the student can use the indicators inside the *Navigation Bar* to freely select a question they wish to answer or review. This eases navigation within a poll and only very little scrolling is required if any at all.

5.3 Course View

The *Course View* has an important function in terms of usability. It acts as a filtered view on all lectures belonging to a certain course. Users will seek the opportunity to sort and filter lectures, which is why this function must be replicated by the redesign. However, as described in Section 3.1.2, the *Course View* has a redundant nature as it looks and feels nearly identical to the *Main View*. Furthermore, some potentially confusing click paths lead to the *Course View* as elaborated in Section 3.1.4. As outlined in Section 5.1.3, the *Drop-Down Menu* filter lectures after a course they belong to. The tabs below allow for additional filtering after the temporal context of lectures. Therefore, with these proposals in place, the *Course View* is deemed obsolete. Dropping the *Course View* from the redesign results in shorter, fewer click paths and stronger interconnectedness between all views (see [Add graphic of new click paths](#)).

5.3.1 Navigation

By shifting navigation elements from the old *Burger Menu* to components of the *Main View*, the new *Burger Menu* is slimmed down (SBM). It only contains the buttons labeled *How it works*, *Edit account* and *Login/Logout*. The options to view the *Question Pool*, to *Evaluate* a given lecture or to unsubscribe from a course are all moved to the *Main View*, as described in Section 5.1.1 and Section 5.1.3. These changes lead to shorter click paths with well-defined behavior (see).

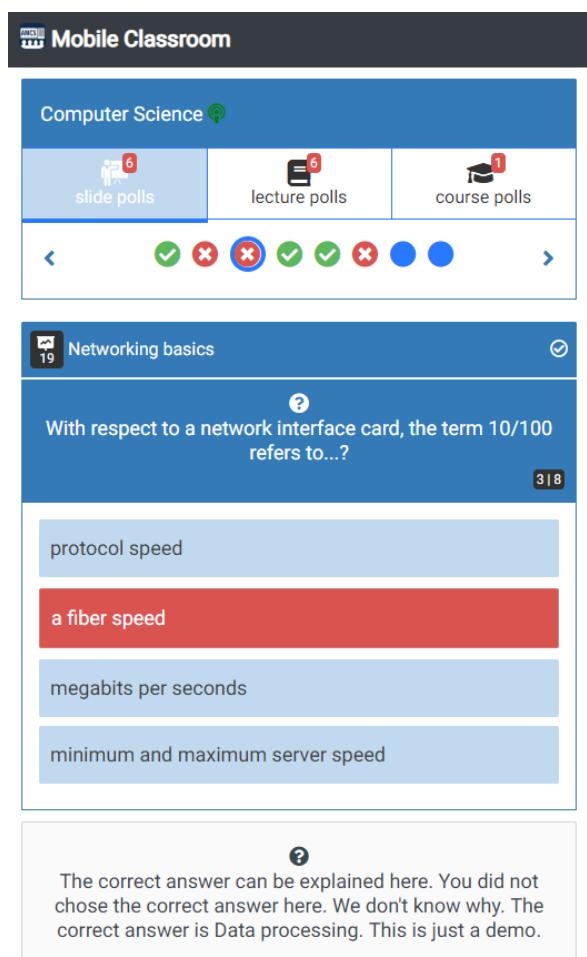


Figure 5.6: Mock-up 2 of the new *Poll View*: Colors are adapted to match the corporate design of AMCS.

add
graphic
here

add
graphic
here

5.4 Summary

All proposals and their respective goals are shortly summarized in Table 5.1. It serves as a reference to which features have to be implemented by the prototype. Therefore, this table will be referenced in Chapter 6.

ID	Name	Views	Goals
LT	Lecture Tabs	MV	<ul style="list-style-type: none"> · Visual separation of lectures · Filter lectures by temporal context · Reduce amount of vertical space used
DDM	Drop Down Menu	MV	<ul style="list-style-type: none"> · Filter lectures by course · Reduce amount of vertical space used
EEF	Embedded Enrollment Form	MV	<ul style="list-style-type: none"> · Reduce cognitive effort to enroll to a course · Reduce amount of vertical space used
LV	Improved Lecture Visualization	MV	<ul style="list-style-type: none"> · Reduce cognitive effort to find information · Reduce visual clutter · Reduce amount of vertical space used
FL	Improved Location of Access To Features	MV	<ul style="list-style-type: none"> · Reduce amount of vertical space used · Reduce click path length · Remove unnecessary click paths
PB	Pagination Bar For Lectures	MV	<ul style="list-style-type: none"> · Reduce amount of vertical space used · Reduce visual clutter · Remove unnecessary click paths
PT	Poll Tabs	PV	<ul style="list-style-type: none"> · Reduce amount of vertical space used · Reduce visual clutter
VQ	Improved Visualization of Questions	PV	<ul style="list-style-type: none"> · Improve visuals and readability · Reduce visual clutter
NVB	Navigation Bar	PV	<ul style="list-style-type: none"> · Improve navigation between questions · Reduce visual clutter · Reduce amount of vertical space used
NBB	Next/Back Buttons	PV	<ul style="list-style-type: none"> · Improve navigation between questions
SW	Support Swiping	PV	<ul style="list-style-type: none"> · Improve navigation between questions · Improve quality of user interaction
SBM	Slimmed Down Burger Menu	BM	<ul style="list-style-type: none"> · Improve global navigation · Reduce click paths and their length · Reduce amount of vertical space used
NB	Notification Bubbles	MV, PV	<ul style="list-style-type: none"> · Improve quality of user interaction · Reduce cognitive effort to find new content

Table 5.1: Classification of proposals for improved usability of AMCS. This classification serves as a feature list for the prototype. MV = Main View, PV = Poll View, BM = Burger Menu

6 Implementation

The previous chapter elaborated on a redesign strategy addressing all of the identified issues that impair the usability of AMCS on mobile devices. This chapter covers the realization of said changes in the form of a prototypical application. However, not all proposals were implemented as specified. One reason for this is that several problems arose during development such as legacy code that could not be changed easily without requiring a complete rewrite. The existing code base is complicated and hard to understand, as several different individuals worked on it over the years.

Another reason is some of the proposals themselves, as they were not thought through to the end or even deemed impractical without necessary improvements. Nevertheless, parts of the concept were adapted and a prototype was successfully implemented. In the following, the implementation process and the required adjustments to the concept will be elaborated by describing different design iterations and the decisions behind each change.

6.1 Main View

The *Main View* has undergone several revisions and does not exactly resemble the mock-ups that were shown in the previous chapter (see Figure 5.3). The general layout stayed roughly the same, keeping the idea of a *Drop Down Menu* and *Lecture Tabs* below. But a lack of consideration for the different *Roles* users can play when using AMCS made a small number of changes and considerations necessary.

The view consists of the *Drop Down Menu* on the top that allows for course selection and *Lecture Tabs* on the bottom that displays lectures sorted by their temporal context. Between the two of them, a context-sensitive *Actions Menu* is displayed.

6.1.1 Drop Down Menu

Generally speaking, the *Drop Down Menu* provides a coarse grain filter that allows the user to select a specific course. Doing so causes the *Lecture Tabs* and their content to change accordingly. More specifically, the *Drop Down Menu* acts in a slightly different manner depending on the current user role. When using it as a *Student* (see Figure 6.1), it contains all courses that said student is enrolled to. On top of that, the *Enrollment Form* is embedded inside the menu, allowing the *Student* to join other courses.



Figure 6.1: The *Drop Down Menu* for the role *Student*: The *Enrollment Form* is embedded.



Figure 6.2: The *Drop Down Menu* for the role *Lecturer*. A menu entry that allows for course creation replaces the embedded *Enrollment Form*.

In contrast, when logged in as a *Lecturer* (see Figure 6.2), the *Drop Down Menu* contains all courses that he owns and manages. Naturally, the *Enrollment Form* is omitted as the *Lecturer* role does not need it. Selecting a course from the list causes an additional component to be displayed: the *Actions Menu*.

6.1.2 Actions Menu

An oversight not covered by the proposals is the fact that the functionality offered by the *Main View* has to change depending on the role of the user that is currently logged in. To address this issue more effectively, the *Actions Menu* is introduced. The *Actions Menu* offers several buttons with different functionality derived from user roles and privileges. The number and kind of buttons displayed in the *Actions Menu* varies depending on role of the user that is currently logged in. *Students* are shown the buttons for either visiting the *Question Pool* or *Leaving* the course altogether. In contrast, if a *Lecturer* is logged in, the *Action Menu* contains more options. A *Lecturer* can *Create a Lecture*, *Edit*, *Export*, *Clone*, *Evaluate* or *Delete* a course. Visualization and placement of the *Actions Menu* is explained in the sections following.

6.1.3 Lecture Tabs

The *Lecture Tabs* are displayed below the *Drop Down Menu*. Each temporal context is associated with a color, label, and icon to allow for easier differentiation between them. Similar to the *Actions Menu*, the concept for displaying lectures had to be adapted to comply with the semantics of different user roles. If a *Student* is logged in, the tabs *Live*, *Upcoming* and *Past* are shown. An additional tab labeled *Offline* is displayed when logged in as a *Lecturer*, as *Lecturers* are able to create lectures in advance and restrict access to them by changing their status to *Offline*.

For each tab, an indicator for the number of lectures belonging to it is provided. Tapping on one of the tabs displays the lectures in a vertical list. Instead of using paging to divide the content into shorter lists as proposed in Figure 5.3, the whole list of lectures is displayed at once. This decision was made partly because the existing code base was difficult to adapt to these criteria.

6.1.4 Lecture List

The visualization of lectures underwent several different iterations before it was finalized for the evaluation. Each iteration introduced small but significant changes to the overall look and feel of the user interface. In general, each lecture is displayed in a box colored according to the temporal context of the lecture. The course name is displayed in a smaller font. Below that, the full title of the lecture is shown in a slightly bigger and bold font. Reading this information from top to bottom conveys the hierarchical structure of courses consisting of lectures.

Iteration 1

Lectures are grouped by date and a visual divider between dates is introduced to emphasize and visualize this grouping (see Figure 6.3). Each lecture is displayed in a color-coded box. On the top, the date of the lecture is displayed along with two buttons. The first button on the left will redirect to the *Evaluation of Answers* functionality. At first, to save space, only the course name and the lecture title are shown. The description of the lecture below is deemed not as important and is therefore retracted per default. The second button on the top right can be used to unveil the lecture description. The *Actions Menu* as such was not part of this iteration yet.

Problems The date divider does not eliminate the redundancy introduced by displaying the date of each lecture individually. Furthermore, the design of the two buttons on the top right might be problematic. For once, it can be difficult to identify these two elements with their placement and lack of textual labels as pressable buttons. While collecting intermediate feedback from the AMCS group, people tended to say that they struggle with identifying the meaning of the iconography used.

Iteration 2

The second iteration addresses the problems described above partly (see Figure 6.4). Most notably, the *Lecture List* resembles more a calendar or a student schedule in its overall design. The layout is divided into two columns: The left column displays the current date, whereas the right column contains the list of courses associated with the date. Essentially, instead of displaying the date for each lecture, this information is extracted to the left column. Now, only the time and duration are displayed at the top for each lecture separately. The description remains retractable.

The screenshot shows the 'Mobile Classroom' application interface. At the top, there is a header with the AMCS logo and the text 'Mobile Classroom'. A three-line menu icon is on the right. Below the header, a section titled 'Kurs auswählen / In Kurs einschreiben' is visible. A dropdown menu is open, showing 'Alle Kurse'. Underneath, a section titled 'Veranstaltungen' is shown with four icons: a green square with a white icon (14), an orange square with a white icon (4), a blue square with a white icon (9), and a red square with a white icon (22). Below this, there are two cards representing lectures. The first card is for 'AMCS MONITORING EVALUATION' on '11.04.2019 @ 09:20' and includes a green 'Details' button and a blue 'Info' button. The second card is for 'DATENBANKEN: Vorlesung' on '11.04.2019 @ 13:00' and also includes a green 'Details' button and a blue 'Info' button. At the bottom, another card is shown for 'BOT COURSE 1000: Bot lecture 1' on '29.03.2019 @ 00:00' with similar buttons.

Figure 6.3: Iteration 1 of the Main View.

Figure 6.4: Iteration 2 of the *Main View*. Four tabs are displayed when logged in as a *Lecturer*. From left to right: *Live*, *Upcoming*, *Past*, *Offline*.

This iteration also includes the introduction of the aforementioned *Actions Menu*. It consists of a vertical list of buttons and remains completely hidden if the entry *All courses* is selected from the *Drop Down Menu*. This design hopes to remove unnecessary clutter by displaying these buttons only when needed.

Problems The introduction of the *Actions Menu* raises new problems. The buttons are organized in a vertical list that pushes all following content further down the page. This problem is essentially identical to the issue that occurred with the original *Burger Menu*. After pressing one of the buttons of the *Actions Menu*, content below might change. However, in some cases, because the menu itself stays in place, these updates of the user interface below might happen completely off screen. Therefore, selecting some of the buttons could feel to the user like an idempotent operation.

The buttons on the top of each lecture have the colors of their background and icon inverted, but they essentially remained the same. Furthermore, each lecture box is given a drop shadow to indicate that the area is pressable. Pressing anywhere on this *card-like* box will redirect students to the *Poll View*, whereas lecturers will arrive at the *Course Dashboard*.

Problems Similar to the ambiguity of the top buttons of each lecture that remains in this iteration, every *tab* is suffering from the same problem. According to feedback from the AMCS group, the used iconography alone fails to convey which temporal context each *tab* is representing.

Iteration 3

Some issues regarding the *Lecture List* are fixed in this iteration. To clarify the meaning of each *tab*, a textual label is added. The colors of each *tab* are adjusted slightly to be more easy on the eyes (see Figure 6.5). Furthermore, improvements to the *Lecture Box* design are made (see Figure 6.6). The time and duration of each lecture are displayed in one line next to each other. The two buttons previously discussed are moved to the bottom. They are expanded to take most of the width of the lecture box. The icons remain the same but they are reduced in size and pushed to the left. A textual label in the center of each button now describes its purpose.

Lectures



Figure 6.5: Improved *Lecture Tabs* of *Iteration 3* with textual labels. The *Offline* tab is displayed only to *Lecturers*.

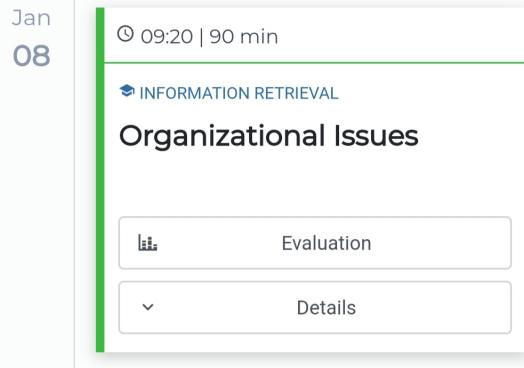


Figure 6.6: Improved *Lecture Box* of *Iteration 3*. The two buttons, previously at the top and unlabeled, are now located at the bottom.

6.1.5 Iteration 4

Iteration 4 represents the last incremental update to the *Main View* (see Figure 6.7). Most prominently, the *Action Menu* is now hidden per default. A new button next to the *Drop Down Menu* suggests additional options that concern the selected course. Tapping this button shows the *Action Menu* (see Figure 6.9). Selecting one of the options from the *Actions Menu* causes it to close and update the view below accordingly. Furthermore, changes to the *Lecture Tabs* are made. The *Notification Bubbles* are changed slightly in style. They now serve as an absolute counter for the number of elements inside a tab. Besides this change, a *Notification Bubble* is displayed for each *Lecture Box*, indicating new and unanswered questions for a given lecture.

6.2 Navigation

As proposed earlier in Chapter 3, the *Burger Menu* was slimmed down considerably. A logged in user can only find the menu entries *Edit account*, *How it works* and *Logout* (see Figure 6.10). The *Question Pool* and *Evaluation of Answers* is moved to the respective lectures in the *Main View*. Apart from these changes, the *Burger Menu* stayed the

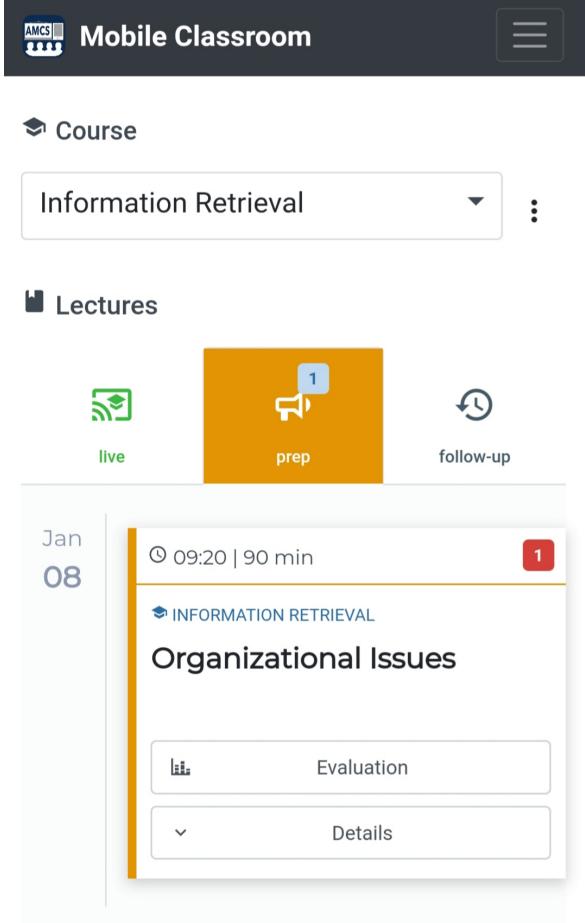


Figure 6.7: *Iteration 4*: The *Action Menu* is now accessed by pressing the *Three-Dots-Button* next to the *Drop Down Menu*. Furthermore, an indicator for unanswered polls is added to the *Lecture Box*.

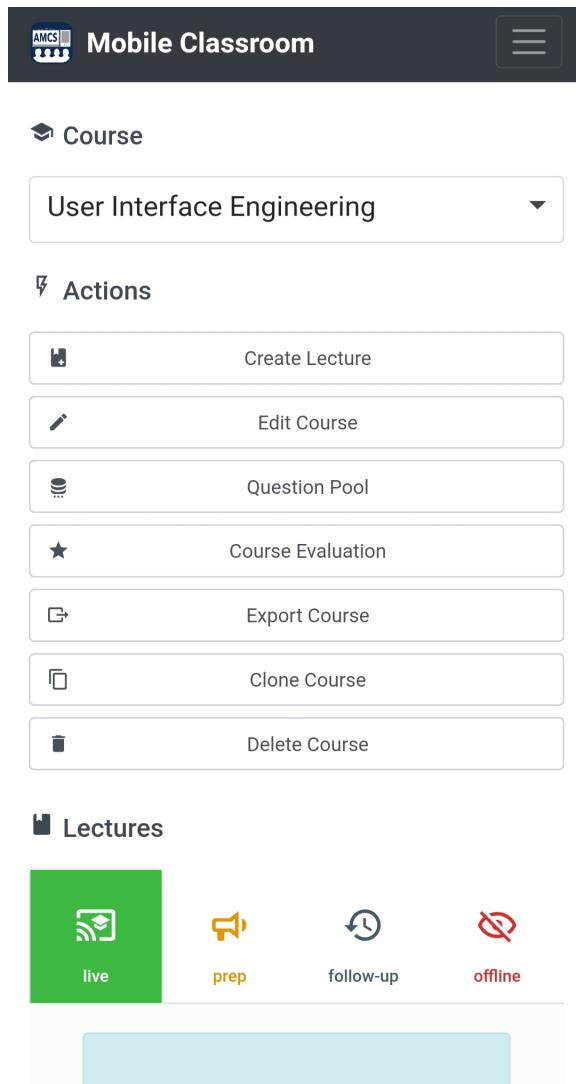


Figure 6.8: Actions Menu for Lecturer in Iteration 3: The Lecture Tabs and any other content below is always displaced by the long list of buttons, because it never disappears.

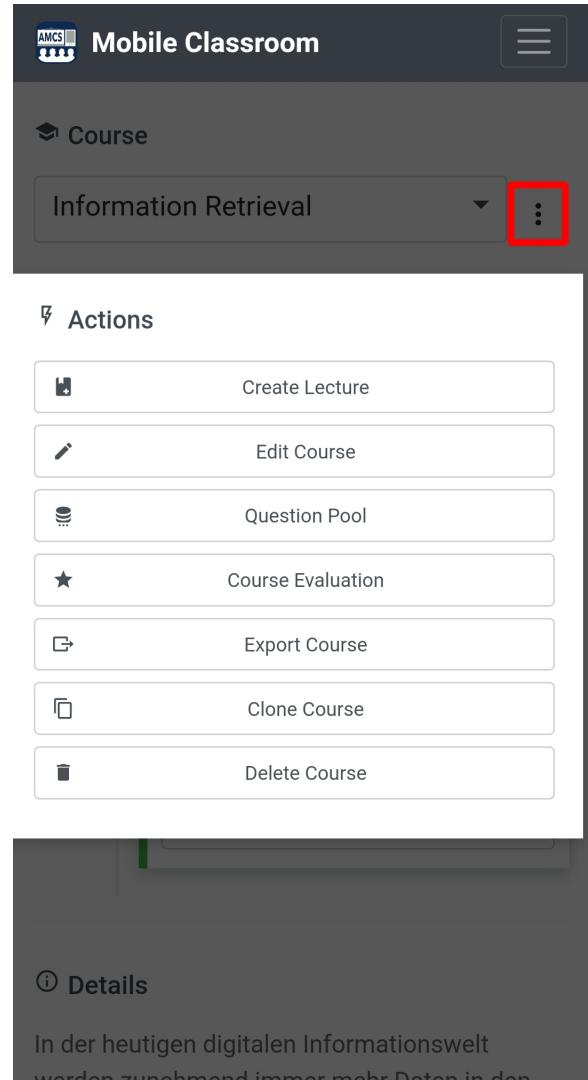


Figure 6.9: Improved Actions Menu of Iteration 4: An extra button denoted by three dots (in the red square) toggles an overlay that contains the Actions Menu. It closes once a selection is made, preventing displacement of the content below.

same. Content below now has a higher chance to stay inside the visible view-port.

6.3 Poll View

The *Poll View* is implemented closer to the proposals made in Chapter 5 than the *Main View* previously described. Nevertheless, several changes had to be made to account for the hard to extend codebase.

6.3.1 Layout

The layout of the *Poll View* overall stayed close to the proposals in Section 3.1.3 (see Figure 6.11). An exception is the *Back Button* that was added at the top left. When pressed, it returns the user to the *Main View*. Similar to the *Lecture Boxes* of the *Main View*, first the course name is displayed, followed by the lecture name in a bigger, uppercase font. As tabs are already used by the *Main View*, a reusable *Tab Component* was implemented to avoid duplicate code in the already complicated codebase. The *Poll View* uses an instance of the component. Each poll type has a tab designated to it, complete with *Notification Bubbles* showing the remainder of unanswered questions for each category respectively. To be more precise, the *Slide Polls* are displayed in the *Slide Tab* and the *Course Polls* reside in the *Course Tab*. However, the *Preparation-, Lecture-, and Post-Processing-Polls* are all located in the *Lecture Tab*. The content that is displayed in this tab therefore directly depends on the state of the lecture.

6.3.2 State updates

The implementation behaves like the existing system in the same manner that lecture state updates or the answering of the question cause the view to update accordingly. This is achieved by leveraging the existing code that relies on a *WebSocket* connection. No manual polling via a refresh by the user is required for the view to stay up to date.

6.3.3 Navigation between Questions

As described in Section 5.2, the *Poll View* features the *Navigation Bar*, albeit slightly modified. Between the buttons for navigating *forward* and *backward*, each question is represented by a clickable circle.

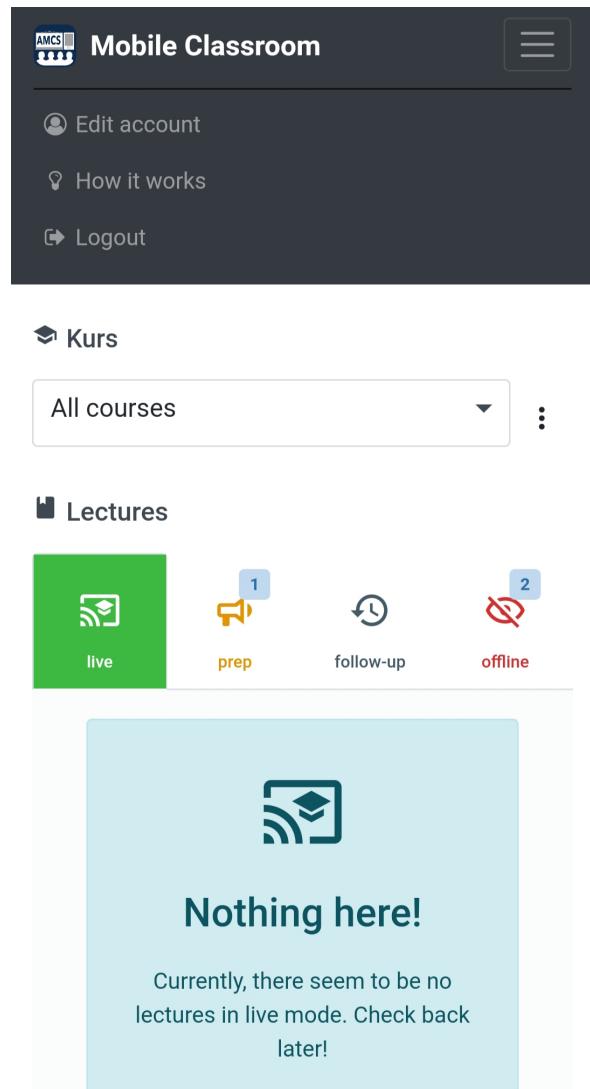


Figure 6.10: Iteration 4: The *Burger Menu* displaces the content that follows in a significantly reduced magnitude.

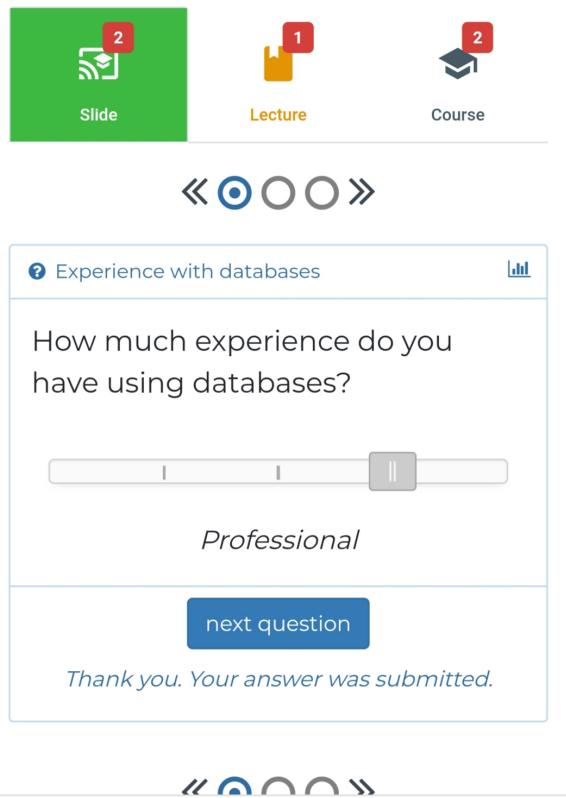
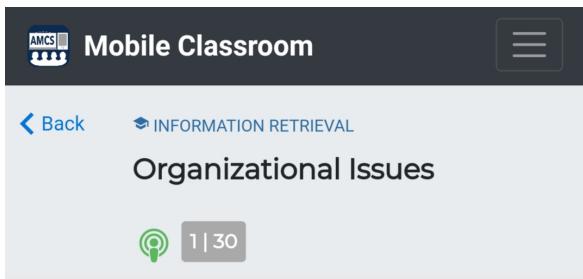


Figure 6.11: The improved *Poll View* using tabs to separate polls, the *Navigation Bar* and the *Previous/Next Question* buttons to ease navigation.

The current question is highlighted visually by usage of a different, blue icon. To ease navigation even further, the *Navigation Bar* is displayed a second time below the question. After answering a question, a button for advancing to the next question is shown (if a next question does exist).

6.3.4 Rendering of Questions

The existing implementations for each question type were adapted to be visually more appealing. The colors used in the prototype differ significantly from the design proposed in Figure 5.6. Most notably, the font size of the question is increased significantly to improve readability. To avoid collision with the established design of the *Main View*, background colors were adjusted to be mostly white instead of a solid blue. Border colors remain in the corporate design of AMCS.

6.3.5 Problems

Not all features were implemented, as problems occurred during development. For example, due to time constraints the feature that allows the *Navigation Bar* indicates the correctness for a given answer through respective iconography was not implemented. The *Next Question*-Button is only displayed, if a next question exists. Furthermore, the prototype implementation is not sophisticated enough to offer a button that allows to jump to the closest unanswered question nearby.

7 Evaluation

8 Conclusion

.1 Appendix Title

This is the text of the appendix, if you need one.

Bibliography

[Smith et al.(2009)Smith, Jones] P. Q. Smith, and X. Y. Jones. ...reference text...