



# Improving the user experience of AMCS on mobile devices

Sinthujan Thanabalasingam

15.11.2019

**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, speakers 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 lays in providing a unified and responsive user interface across all supported platforms that is intuitive to use. This work analyzes AMCS and it's web interface under these criteria, formulates 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 the TU Dresden. Several members of the Faculty of Computer Science initialized the project in 2012 and since then, the system and its community have grown continuously and countless features were added over the years.

In general, the system's main objective is to improve the way knowledge is transferred from a speaker to their audience. By providing interactive polls and evaluation mechanisms, AMCS aims at increasing the audience's engagement before, during and after a lecture takes place, overall strengthening the interaction between speaker and audience.

On the one hand, several features of AMCS are designed to support speakers in their role. Regarding the educational context, lecturers at university use it not only during their lecture to get immediate feedback from participating students. Furthermore, it is used during the whole semester to prepare and manage lectures, to analyze learning progress and to evaluate feedback given by students. More specifically, a speaker can gain insight on the understanding of their audience by evaluating answers to their polls. For example, poor results to a poll covering a certain topic might suggest that the topic was misunderstood or badly explained. AMCS enables the speaker to precisely use this information, helping to identify parts of their lectures that students might struggle with. Speakers can focus more easily on repeating and emphasizing these topics in the future.

On the other hand, AMCS provides a set of features that try to increase engagement and to close the gap between speaker 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, the latter two tools can be used afterwards to identify and repeat difficult questions, to prepare for the next lesson or to study for the examination.

## 1.2 Motivation

Several standalone front end applications for different platforms such as iOS, Android and web are provided to the audience. However, 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 web page being the most common way AMCS is used, 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 in terms of usability, design and consistency.

## 1.3 Outline

This work's central goal is to formulate a redesign strategy that when implemented accordingly 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. A short introduction is given to the look and feel of the central elements of AMCS. In chapter 3, an 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. Chapter 5 then proceeds by mapping out the 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 necessary adaptations, changes and differences to the initial proposals. In chapter 7, the prototype is used to evaluate the redesign strategy. A comparison to the current state of AMCS is drawn. Finally, this work is concluded in chapter 8 with open questions and a short outlook on future work.

## 2 Current State of AMCS

At the time of writing, AMCS offers front end applications for iOS, Android and web that students can use. AMCS does offer a variety of features to the audience and to speakers. They were designed and developed by different people over the years and therefore all differ variously in terms of UI design and layout. To keep the scope of this work manageable, this work 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 Technology

The AMCS web page is written using Angular<sup>1</sup>, a typescript-based front end framework for building mobile and desktop web applications.

### 2.2 Landing Page and Login

When accessing the website<sup>2</sup>, students will be shown the landing page of AMCS (see fig. 2.1). A big login button is displayed that when pressed will reveal a login form (see fig. 2.2). 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 reasonably use the service. By typing in an optional PIN code, students will subscribe automatically 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 fig. 2.3). It can be scrolled in the vertical direction and is divided into header and 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:

---

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

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

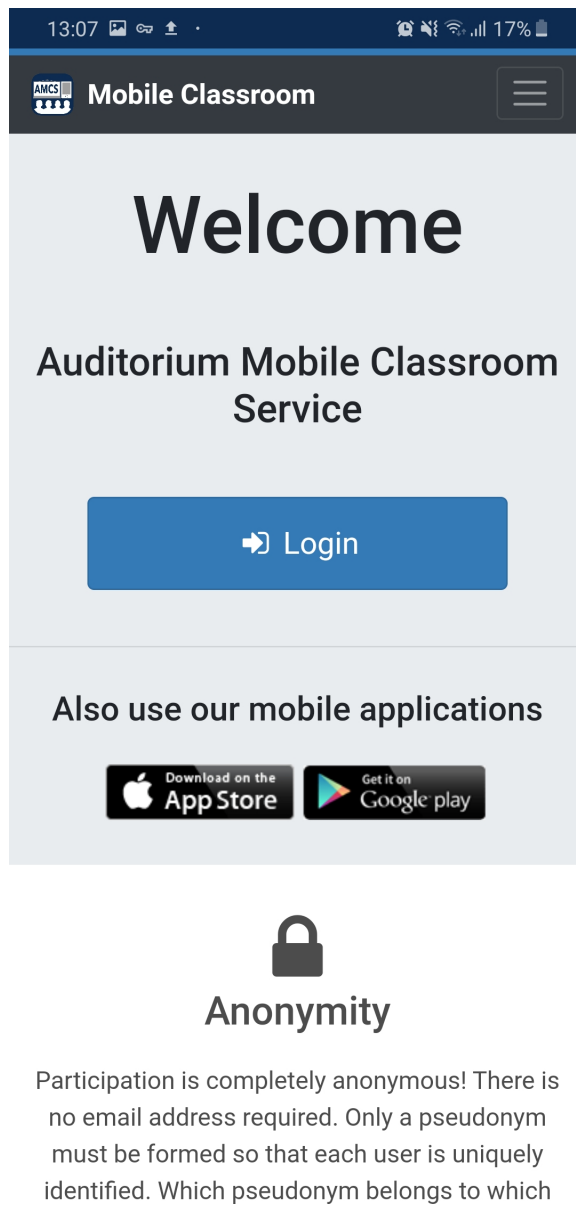


Figure 2.1: The landing page of AMCS. This is the initial screen shown when accessing <https://amcs.website>

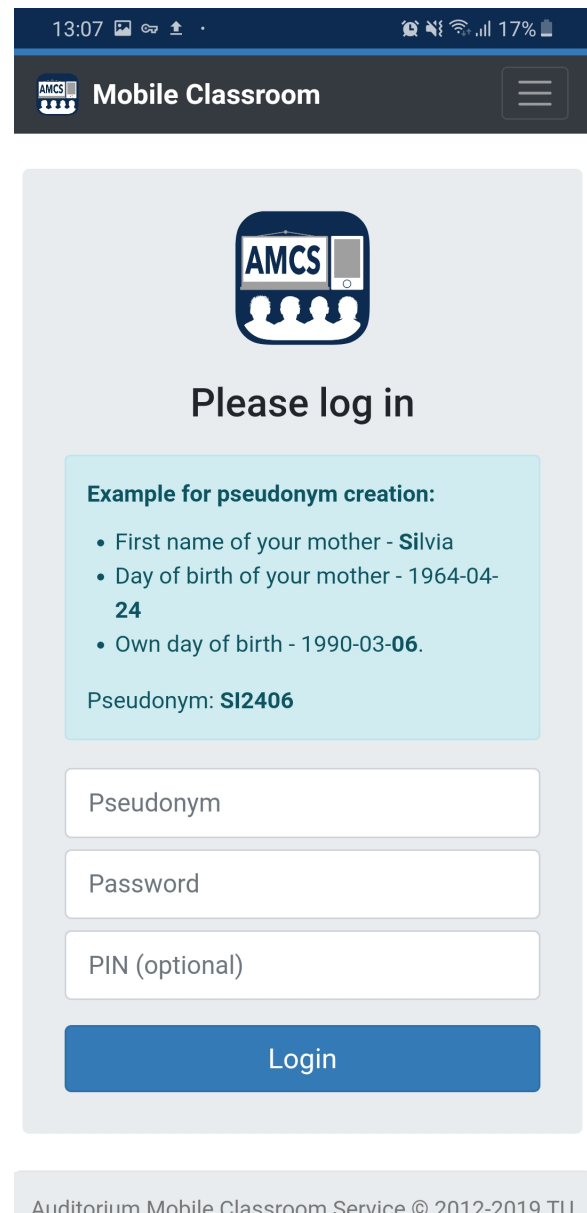


Figure 2.2: The login form of AMCS. Users can choose a synonym and a password to create an account.

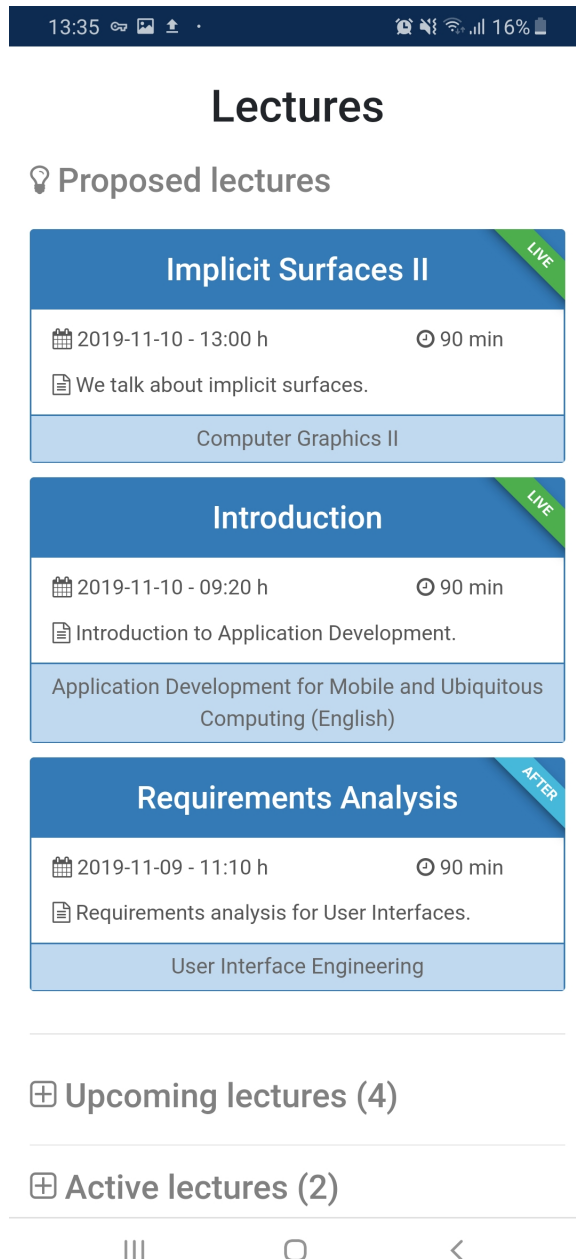


Figure 2.3: Main View: Lecture information is provided in different sections for each temporal context. Per default, a section with proposed lectures is expanded.

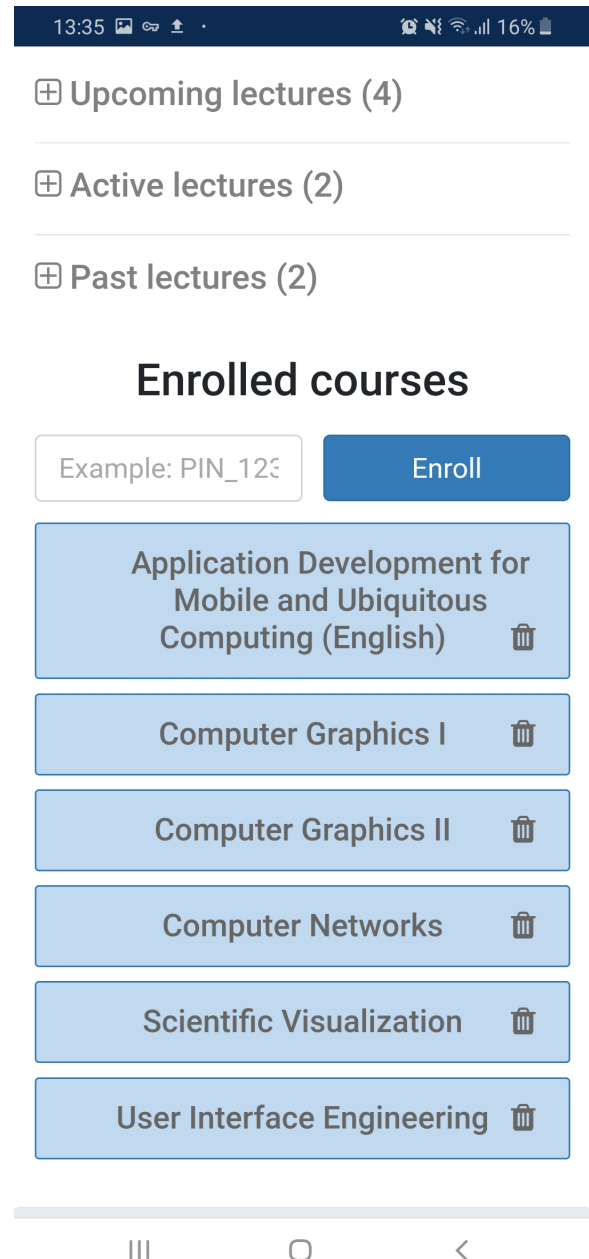


Figure 2.4: Course management below Lectures: Each course is listed below an enrollment form consisting of a PIN input and a submit button.

### 2.3.1 Lectures

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

**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 aforementioned subsections 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 for the temporal context of the lecture.

### 2.3.2 Course management

Further down on the page, the section *Enrolled Courses* can be found (see fig. 2.4). It serves two purposes: Primarily, it provides a way to enroll into a course. An enrollment form is shown that consists of a text field to enter the course PIN and a blue button to trigger the enrollment. When provided with a valid PIN, pressing the button redirects the user to the *Course View* (described in section 2.5) on successful enrollment. Secondly, the view shows all courses the student is currently enrolled in. 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.

## 2.4 Poll View

Answering polls is one of the main functionalities of the system that users will engage with. 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 in 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 life time 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.

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

## 2.5 Course View

The *Course View* is shown when the user selects one of the courses from the course management section (see fig. 2.5 and fig. 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,

add  
graphic  
of  
badges

add  
graphic  
here



Poll Type	Explanation
Slide Poll (SP)	Active when a specific slide is shown. Commonly used for pop 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. 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.

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  
sec-  
tion  
here

### 2.6.1 Student

This is one of the most important buttons that connects a subset of 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. Question Pool
2. Evaluation of answers
3. Edit account

### 2.6.2 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.

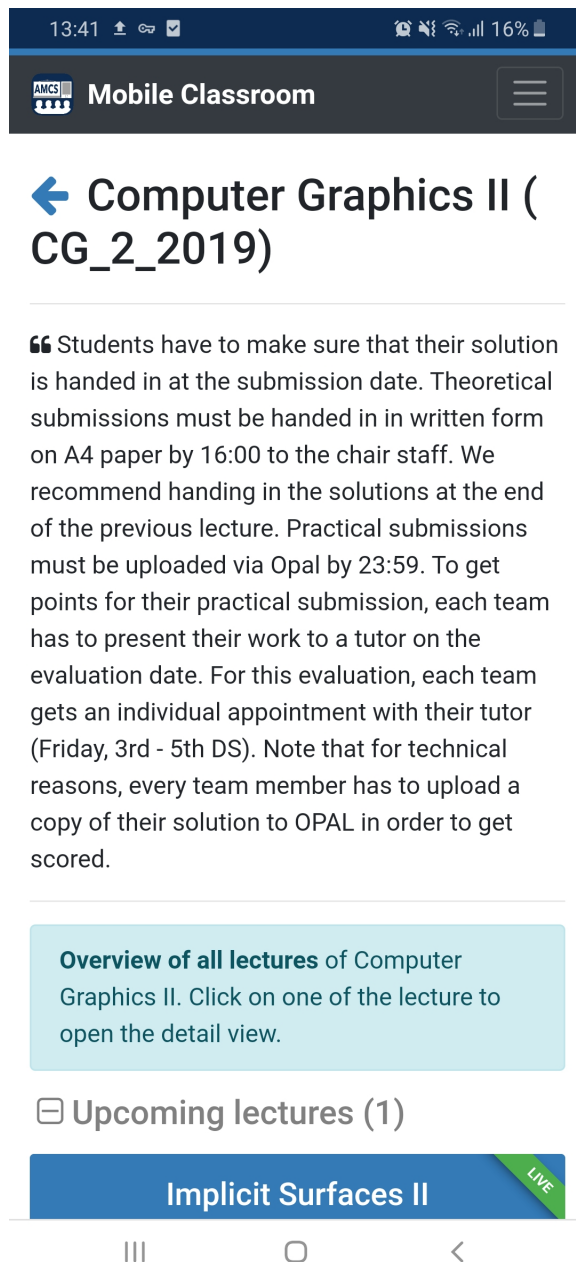


Figure 2.5: *Course View*: Details like the course name, description and PIN are displayed.

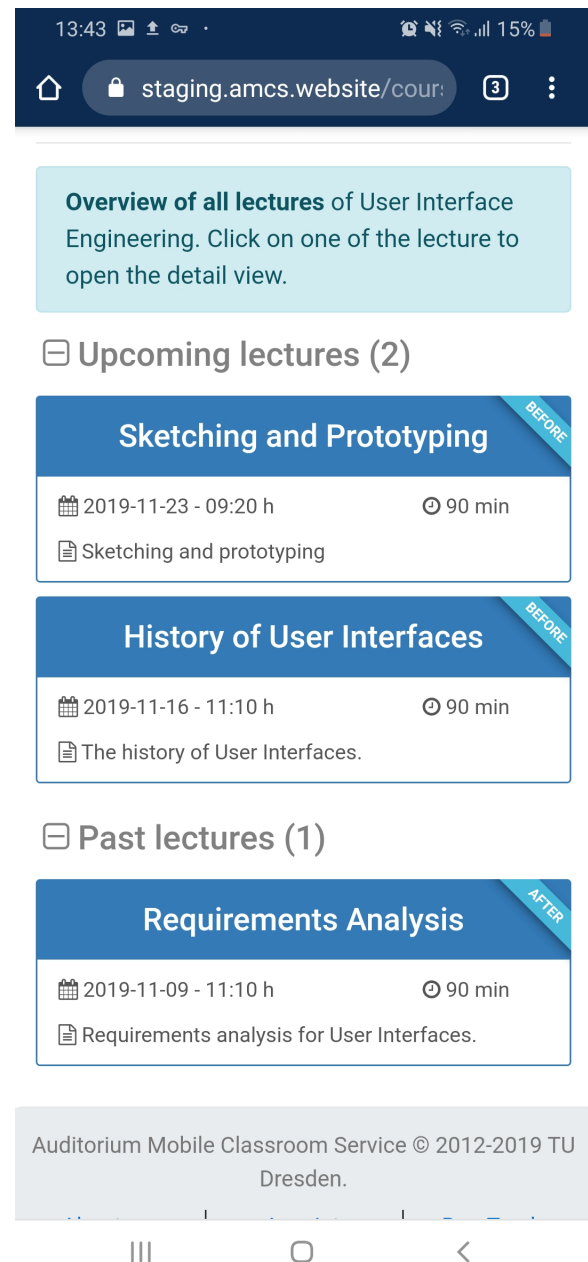


Figure 2.6: Lectures that belong to a certain course displayed in the *Course View*.

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. Can be reached from the <i>Burger Menu</i> .
Evaluation of answers	Shown when the evaluation of answers to polls is attempted. Can be reached from the <i>Burger Menu</i> .

Table 2.2: UI Components of AMCS touched by this work.

### 2.6.3 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.

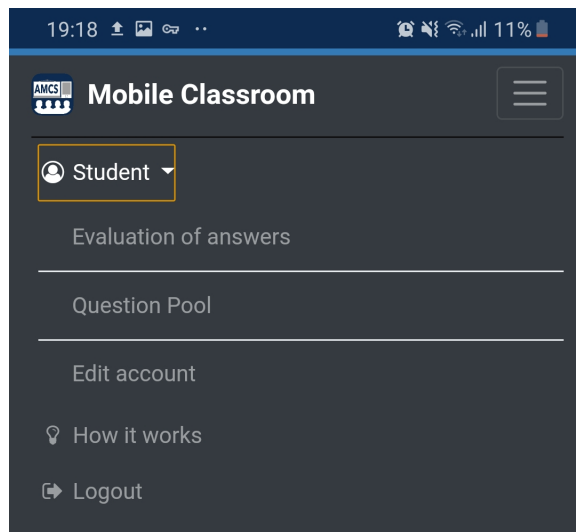
### 2.6.4 Question Pool

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 they are interested in. 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 composed into polls that the student then can answer again. These polls are rendered in the same manner as stated in section 2.4.

add  
sec-  
tion  
here

## 2.7 Related Systems

Add  
this



## Lectures

### 💡 Proposed lectures

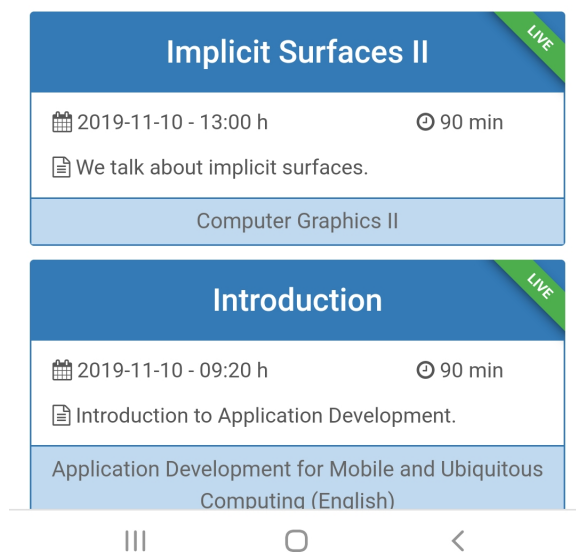


Figure 2.7: Expanded *Burger Menu*.

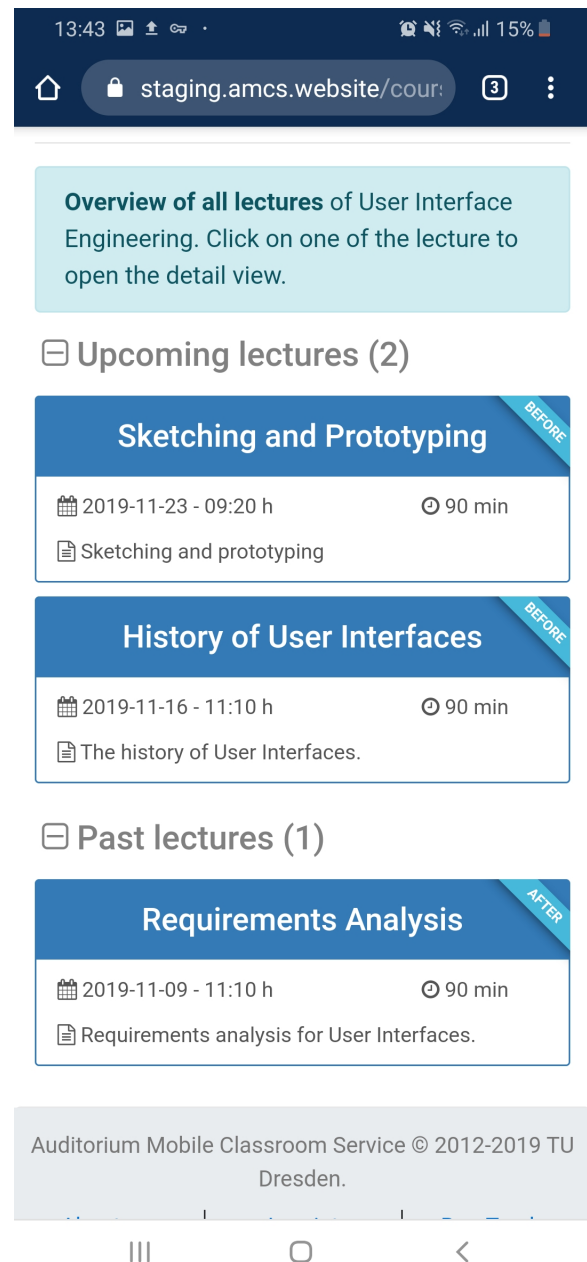


Figure 2.8: Lectures that belong to a certain course displayed in the *Course View*.

## 3 Problem Analysis

This chapter will analyse and identify problems that occur when using AMCS via a web browser on mobile devices. To narrow down the extend of this work, the system is analyzed from the point of view that an audience member like a student has while using AMCS on their smartphone. When 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 on 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 uniform UI experience that guarantees high usability. One challenge lays in the fact that the system has to deal with limited screen space to visualize information as effectively as possible. Additionally, users might approach the application with different ways of interaction and navigation in mind that are typical for mobile devices. For example, a smartphone user might expect to be able to use swiping gestures to navigate a menu or that information is organized in views consisting of separate tabs. This section lists key issues that lower usability or might cause confusion to mobile users. A summary of the findings can be found in Table 3.1 and Table 3.2 respectively.

#### 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. The layout in place causes problems regarding usability aspects of the application.

##### General Visualization Problems

subsection 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. It is followed by details about the lecture such as time, duration and a textual description, visualized in gray letters and icons on a white background. 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 to students take longer time to find the pieces of information that they are looking for.

In addition to that, inappropriate background colors and font sizes are used 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 is disregarded.

## **No Notifications For New Or Unread Content**

The *Main View* lacks completely of visual indicators for new or unread content. Typically, not all polls are visible to students at the beginning of a semester - either because the polls do not exist yet or for the reason that polls can be activated by lecturers at a later point in time. In the current version of AMCS, notifications for new or unread content are completely missing. For example, in order to see whether or not for a given course or lecture new polls exist, students have to tap on the lecture to check. This is unintuitive and adds another layer of indirection to the overall workflow.

## **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 a lot 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 of the page. A list of only four boxes causes a scroll bar to appear on the very common screen resolution of 1920x1080 pixels. A student that navigates to the *Main View* to enroll into a new course therefore always has to scroll to the bottom of the page before they reach the enrollment form. The same problem likewise occurs when simply seeking information about what courses a student is already enrolled in or when trying to leave a course altogether.

Further more, if a student looks for information regarding a specific course, no filter or search functionality is offered by the lecture list. Instead, 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. Finally, they are redirected to the *Course View*, which then displays a filtered list of lectures belonging to the course. This level of indirection is only further increased the more courses the student is enrolled in.

## **Redundancy**

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

In the course management section below, each course is represented by a box with the course's name along with an unsubscribe button, represented by a trash can icon(see Figure 2.4). This is a redundant way of rendering the courses which adds noise to the overall look and uses a lot of the vertical screen estate.

### 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 very important function in terms of usability as it acts as a filter for lectures belonging 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 subsection 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. Bigger polls that consist of multiple questions unnecessarily take a lot of vertical screen estate. The view also lacks of basic information such as total number of questions or number of remaining questions. This information might be useful in bigger polls if students want to gain an idea on how many questions are left. 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). This might be distracting to some students.

Additionally, the layout lacks of separation and distinction between types of polls. Different types of polls are 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 subsection 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*. This layout requires that a student who wants to view these questions has to scroll all the way to the bottom of the list, further increasing indirection and slowing interaction down.

### 3.1.4 Burger Menu and Navigation

At the time of writing, the ways of navigating the application can be described as 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 and can be reached by clicking on elements inside a view such as a course or lecture. The following paragraphs elaborate more on both aspects of navigation.

On the one hand, the interconnected graph of views as it is described in Figure 3.1 contains two main issues. One example of unexpected behavior is the fact that when a user wants to return from the *Poll View* to the *Main View* in order to chose a different poll, they are first redirected to the *Course View*. However, it is possible to reach the *Poll View* from the *Main View* with one tap, by selecting a lecture for example. Furthermore, some views like the *Question Pool*

add  
graphic  
here

*View* and the *Answer Evaluation View* do not offer buttons that allow to navigate back. The navigation relies partly on the corporate branding on the upper-left that when tapped will redirect to the *Main View*. This means that the interconnectedness between different views might not be strong enough.

On the other hand, the *Burger Menu* connects several aspects and functionalities of AMCS in an overarching manner, as it is part of every view. One visual problem that arises is the fact that the menu uses a lot of vertical screen space and delocates the rest of the content that is currently shown when several sub menus are expanded. An example as to why this is problematic might be the following scenario: a student wants to evaluate their answers to polls for a certain 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. But this information is required in the *Answer Evaluation View* because the user is asked to chose their course and lecture of interest from two drop down menus. This could lead to users having to return to the *Main View* to look up the lecture name or other details again. In addition, the menu can confuse students because the first menu entry is labeled as *Student*. This implies to users in general that different roles exist that a user can play in AMCS, which is the case. However the user's role should not be the label of a sub menu in the navigation as it is not intuitive that functionality like account management, the Question Pool and Answer Evaluation can be accessed via a button labeled *Student*.

## Evaluation of answers

Clicking on 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 that they are interested in. Afterwards, 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 be located 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 afterwards 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 lot of scrolling is required to jump between questions.

add  
graphic  
here

## 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. Käsekuchen



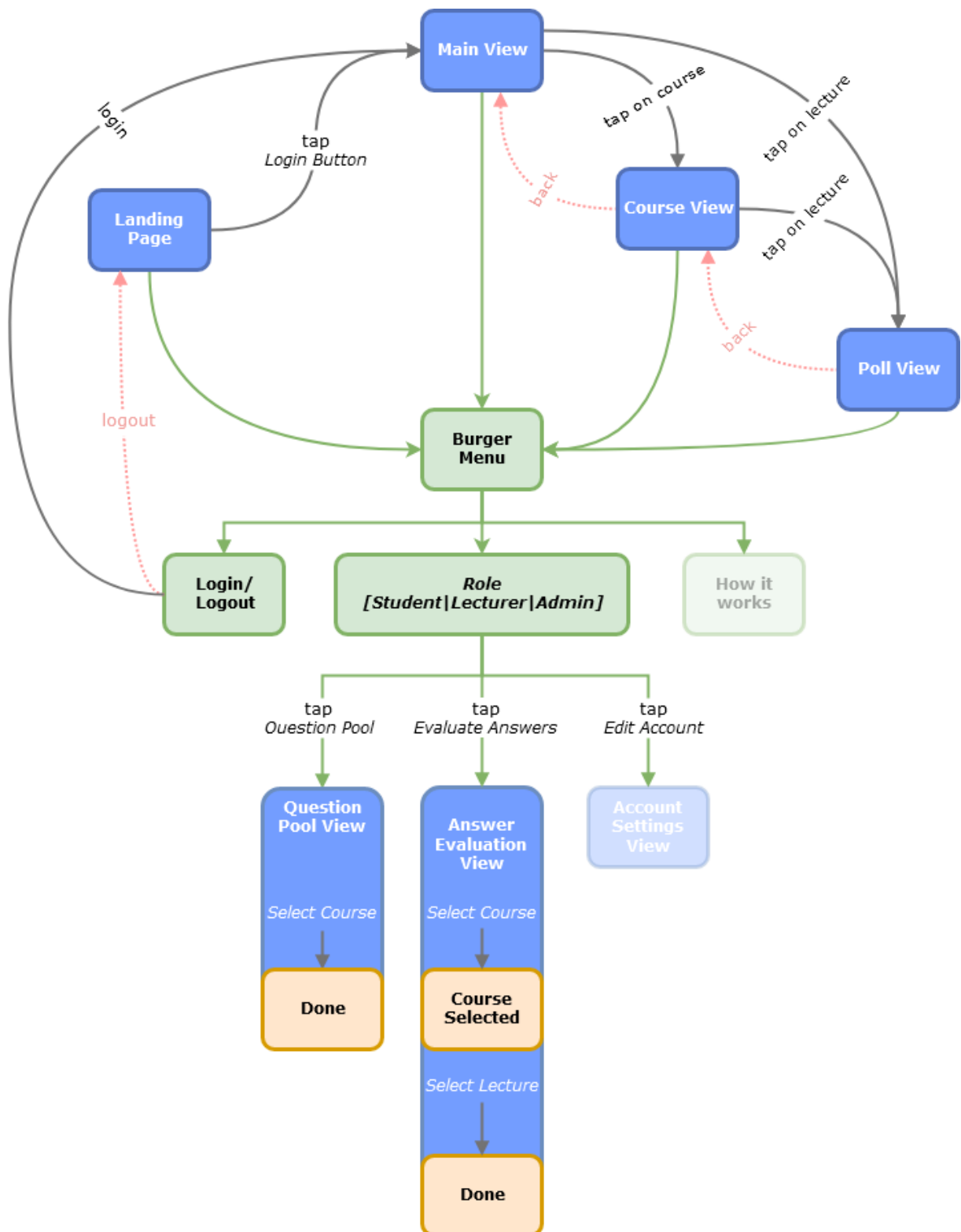


Figure 3.1: Navigation concept of AMCS: Every arrow represents a tap / click the user has to do to reach the desired destination.

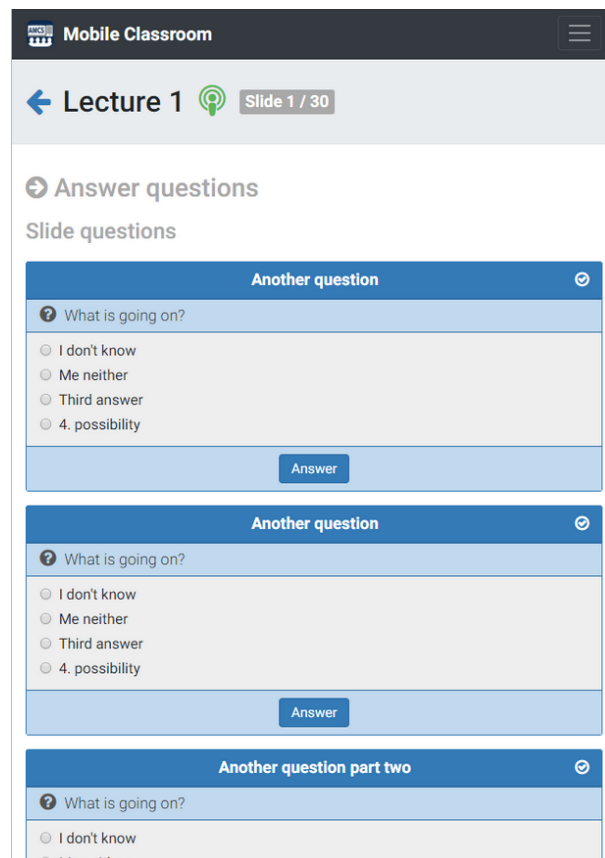


Figure 3.2: *Poll View*: Multiple questions are displayed at the same time.

ID	Name	Categories	Components	Summary
GV 1	Visualization of Lectures	Visualization	Main View	The hierarchical order of courses containing lectures is disregarded.
GV 2	Missing Notifications	Visualization	Main View Course View	No notifications for new or unread content are given.
GV 3	Lecture Badges	Visualization Redundancy	Main View Course View	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	Main View	Courses are each rendered in a box as a list on the bottom of the <i>Main View</i> .
IN 1	Course Management	Layout	Main View	The section <i>Enrollment</i> and the course list are located too far to the bottom of the view.
IN 2	Course Filter	Functional	Main View	The view lacks of a filtering mechanism.
PV 1	Poll View Layout	Visualization Layout	<i>Poll View</i>	Polls use too much vertical space.
PV 2	Missing Poll Information	Visualization	<i>Poll View</i>	Number of total and remaining questions is missing.
PV 3	Missing Poll Separation	Layout Navigation	<i>Poll View</i>	The view lacks of separation between different poll types. Navigation between different poll types requires too much scrolling.
NAV 1	Switching Courses	Navigation	Main View Course View	Quickly switching between courses requires unnecessary navigation between <i>Course View</i> and <i>Main View</i> .
NAV 2	Confusing Click Paths	Navigation	<i>Main View</i> <i>Course View</i> <i>Poll View</i>	Under certain circumstances, returning to an earlier view can yield unexpected results.
NAV 3	Weak View Interconnectedness	Navigation	<i>Main View</i> <i>Question Pool View</i> <i>Answer Evaluation View</i>	Some views miss proper back buttons to return to an earlier view.
NAV 4	Menu Size	Visualization Layout	<i>Burger Menu</i>	Fully expanding the <i>Burger Menu</i> uses too much screen space.

Table 3.1: Classification of issues identified in the usability analysis of AMCS.

ID	Name	Categories	Components	Summary
NAV 5	Menu Labeling	Visualization	<i>Burger Menu</i>	Some menu entries are labeled inadequately.
NAV 6	Evaluation of Answers	Navigation Layout	<i>Answer Evaluation View</i>	Several drop down menus must be operated to evaluate answers for a given lecture.
NAV 7	Question Pool	Navigation	<i>Question Pool View</i>	The <i>Question Pool View</i> is not reachable easy enough, as expanding the <i>Burger Menu</i> is required.

Table 3.2: Classification of issues identified in the usability analysis of AMCS.

## 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 or user navigation. A set of proposals that aims at solving these problems is introduced in the subsequent chapter. Mainly, the focus will lay 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. Each proposal for itself is centered around improving usability of the application while all proposals as a whole aim at keeping the interface consistent and recognizable across all supported platforms.

### 5.1 Main View

Several improvements for the layout and visualization of the *Main View* follow 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 vertical space available inefficiently. Most noticeably, the course overview and enrollment form are placed below the list of lectures. In order to find information about relevant courses or to enroll / 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. A mock-up of the proposals described in the following can be seen in Figure 5.1. First of all, the view is restructured to follow the hierarchical concept of courses containing lectures: 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 functionality 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 tab bar is 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 entered the state *past*. This layout reduces the amount of vertical space used drastically. The placement of the "Evaluate" button (5) is motivated by

reducing vertically occupied space as much as possible, but it could be reasonable to place it below the description text of the lecture. On most devices, the amount of scrolling required should be reduced with the proposed layout. In an effort to reduce redundancy as possible from the view, the badges that indicate temporal context of the lectures now miss completely.

**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 in 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 expanded the layout again to display 3 lectures at once (see Figure 5.3). A page view concept is proposed that minimizes scrolling and maximizes the amount of information displayed at once. The location of the *Evaluate Button* has moved again to the top portion of a lecture box.

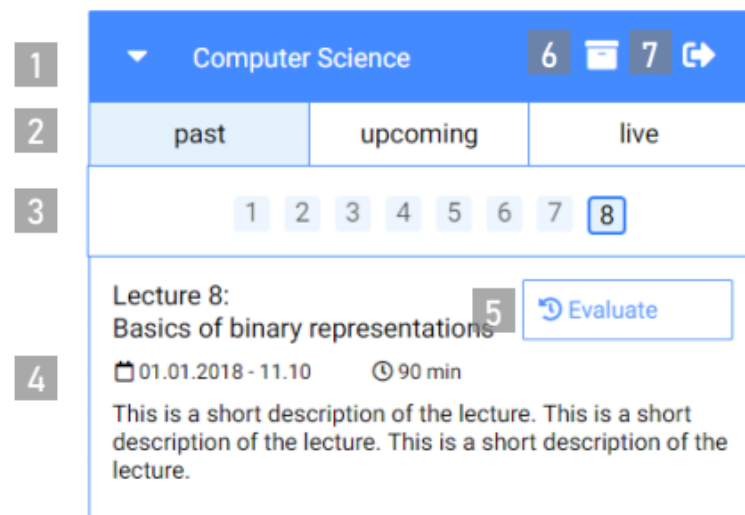


Figure 5.1: Mock-up 1 of the new *Main View*: Currently, the *past courses* tab is shown. A course that has already taken place is selected. Usage of drop down menus and tabs to reduce the amount of vertical space used. (5), (6) and (7) serve as buttons to evaluate given answers for the shown lecture,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

### 5.1.2 Navigation

Tabs (2) should 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

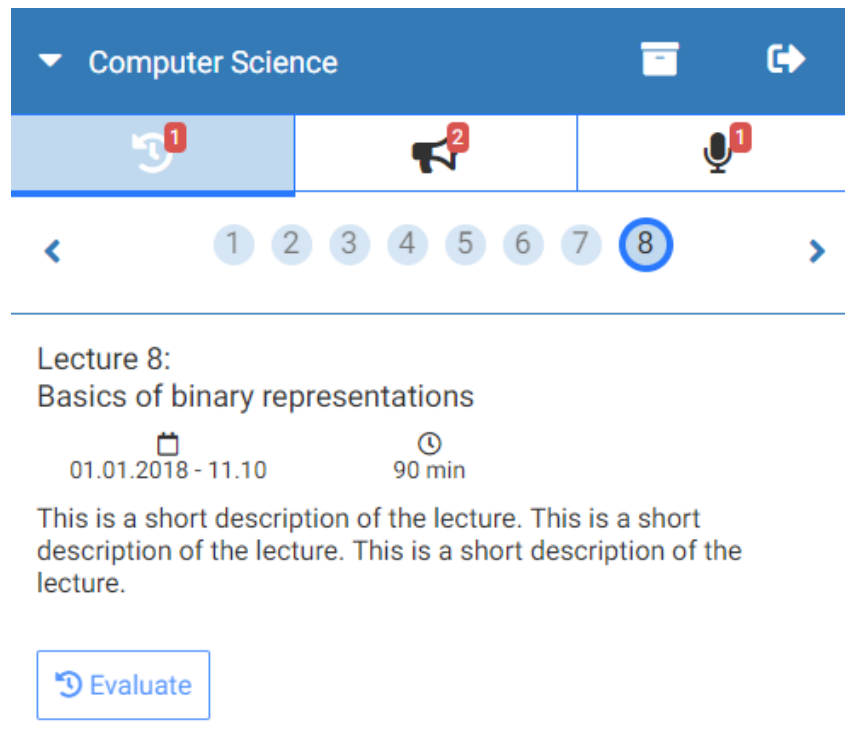


Figure 5.2: Mock-up 2 of the new *Main View*: Textual descriptions for tabs are replaced with iconography, notification bubbles for unread content are introduced. Colors are adjusted to match the corporate design of AMCS.

between *past*, *upcoming* and *live* lectures. The *Navigation Bar* (3) in Figure 5.3 is used to ease navigation between lectures that share the same temporal context. A student can use the bar for example to switch quickly between the oldest and newest past lecture by selecting the corresponding button from the bar. This should improve navigation within the *Main View* as well as within a lecture context. Additionally, this proposal aims at removing indirection as much as possible on a global scale by providing certain buttons that serve as shortcuts for the functionality that is currently found in the *Burger Menu*. Button (6) serves as a shortcut to the *Question Pool* for the selected course. On click of button (7), 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 on a course scale, whereas the "Evaluate" button (5) operates on a lecture scale. The *Evaluate Button* eliminates the need to select a course from a drop down menu and even choosing from a list of lectures is not necessary anymore, as all of these information needed to send the request to the back-end are present. Therefore, multiple layers of indirection are removed from the established workflow. All three buttons try to remove the indirection introduced by the *Burger Menu* as possible. Functionality that is associated with a course or lecture is triggered from a view that deals with courses or lectures respectively. The *Burger Menu* would then only have to deal with the profile editing and logout functionalities.

### 5.1.3 Embedded Drop down menu

The drop down menu 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, a list of courses a student is already enrolled to is shown below. The enrollment 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 enrolled courses could make this functionality 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, in place of the *Main View*, only the enrollment form should be shown. When selecting an item from list of courses below, corresponding lectures will be brought up in the *Main View*. In summary, the *Drop Down Menu* acts as a filter to the *Main View* and essentially covers the responsibilities of the *Course View* currently in place.

#### 5.1.4 Poll View

#### 5.1.5 Layout and Visualization

Several issues have been identified regarding layout and visualization of polls in section section 3.1.3. Main problems include the ineffective use of vertical space in this view and a lack of separation between types of polls. Different types of polls are simply appended to the bottom of the list, making it even longer. In order to solve these issues, a tab-based layout is used 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 greatly.

Below the *Tab Menu*, 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 subsection 5.1.2 is reused here (4). It serves as a means to navigate between questions more easily and faster but 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, where as bold green or red dots indicate correct 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 question is displayed in a blue rectangle with white text. Below the question, an instance of an answering mechanism is displayed (6). Currently, each questions 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 a wrong answer. Therefore, in the case of SC-, MC-, SCC- or MCC-question, the button to answer the question is omitted and mere selection of 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 can 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.

**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* 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. The buttons that allow to jump between questions are moved to the *Navigation Bar*, as they resided previously next to the topic and question in order to separate navigation from content. The indicator for the current slide number is modified slightly and pushed to the left. On the right side, icons that differentiate between question types are reintroduced.

### 5.1.6 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* 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 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 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 PCs 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, no scrolling is required anymore.

## 5.2 Course View

The *Course View* has an important function in terms of usability since it acts as a filter for lectures belonging to a certain course. Users must be given the opportunity to sort and filter a list of elements, which is why this function has to be preserved in the redesign. However, as described in subsection 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 subsection 3.1.4. As outlined in segment subsection 5.1.3, the *Drop Down Menu* serves as a filter for courses, rendering the *Course View* obsolete. It is therefore dropped by the redesign. This results in an additional side effect in form of reduced amounts of click paths and stronger interconnectedness between all the views (see ).

add  
sec-  
tion  
here

## 5.2.1 Navigation

By shifting navigation elements from the *Burger Menu* to components of the *Main View*, the new *Burger Menu* is slimmed down. It will only contain 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 have all been moved to the *Main View*, as described in sections subsection 5.1.1 and subsection 5.1.3. The changes lead to shorter click paths in general with more well-defined behavior (see ).

add  
graphic  
here

Add  
a  
table  
that  
names  
and  
de-  
scribes  
all  
pro-  
pos-  
als  
so  
that  
the  
im-  
ple-  
men-  
ta-  
tion  
chap-  
ter  
can  
refer  
to  
this

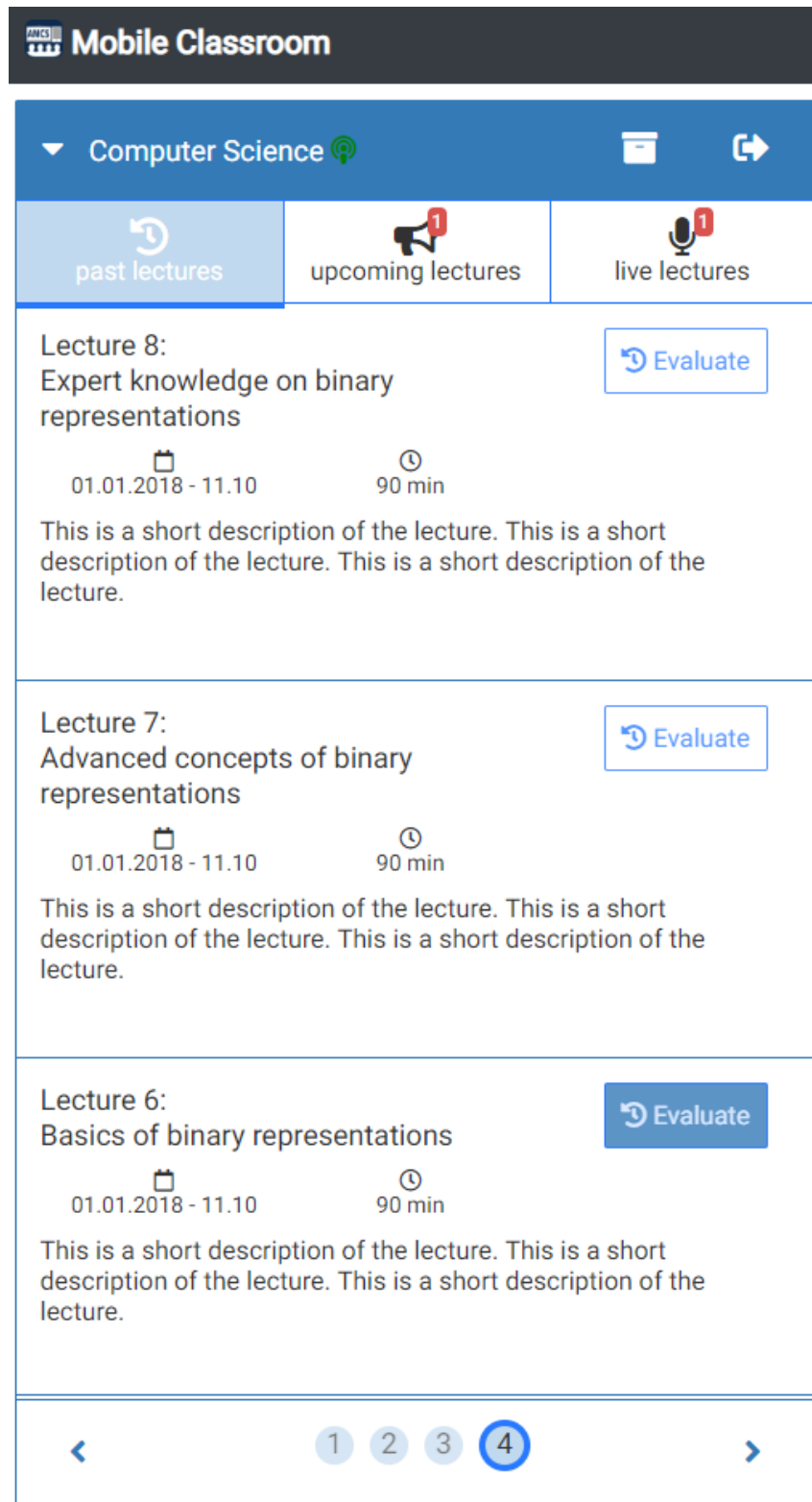


Figure 5.3: Mock-up 3 of the new *Main View*: The icon indicating an ongoing lecture for a course is brought back. A combination of textual description and iconography is used for tabs. Multiple lectures are displayed at once in a page-based view.

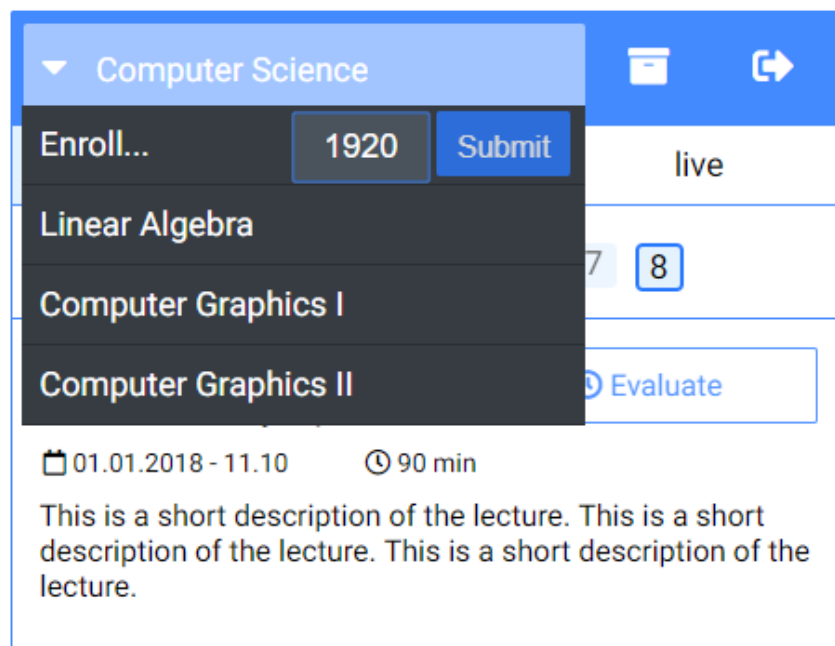


Figure 5.4: Mock-up of the new *Embedded Drop Down Menu*: The enrollment form is integrated in the menu. Users can change the currently selected course from here. The *Main View* then adapts accordingly, allowing the menu to act as a filter for courses.

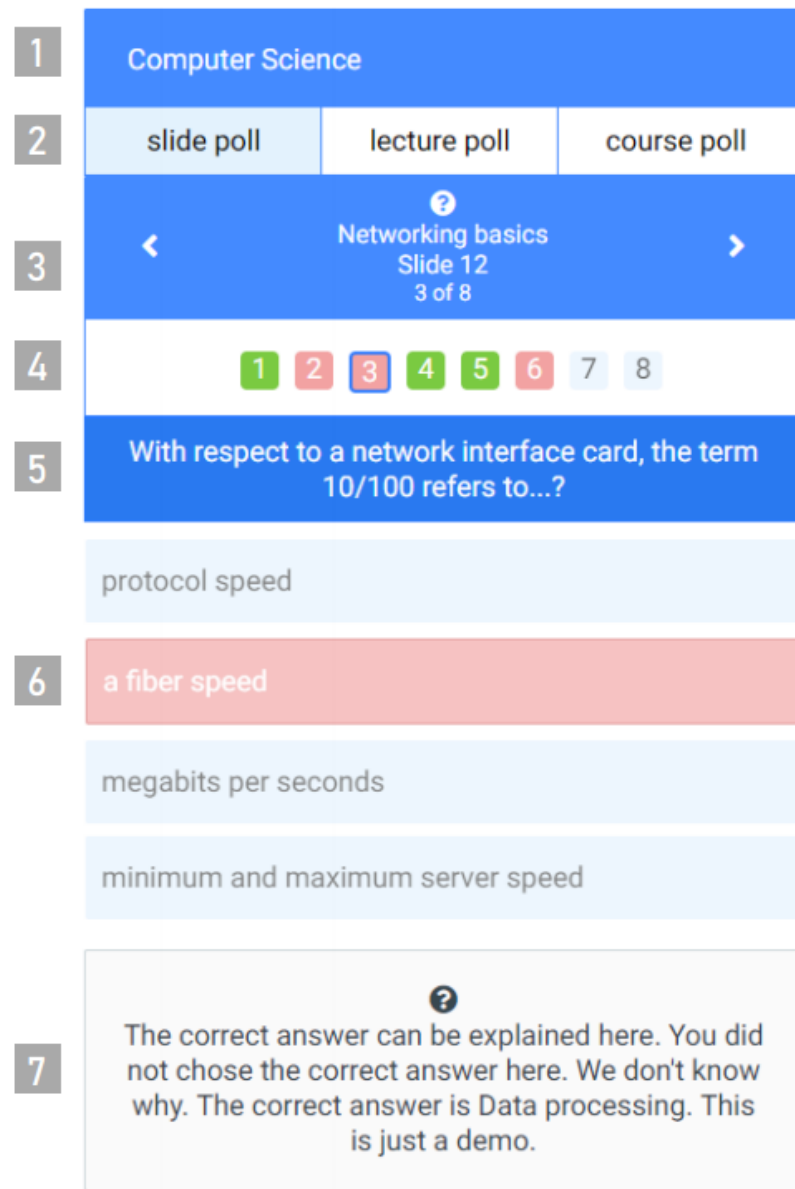


Figure 5.5: Mock-up 1 of the new *Poll View*: The icon indicating an ongoing lecture for a course is brought back. A combination of textual description and iconography is used for tabs. Multiple lectures are displayed at once in a page-based view.

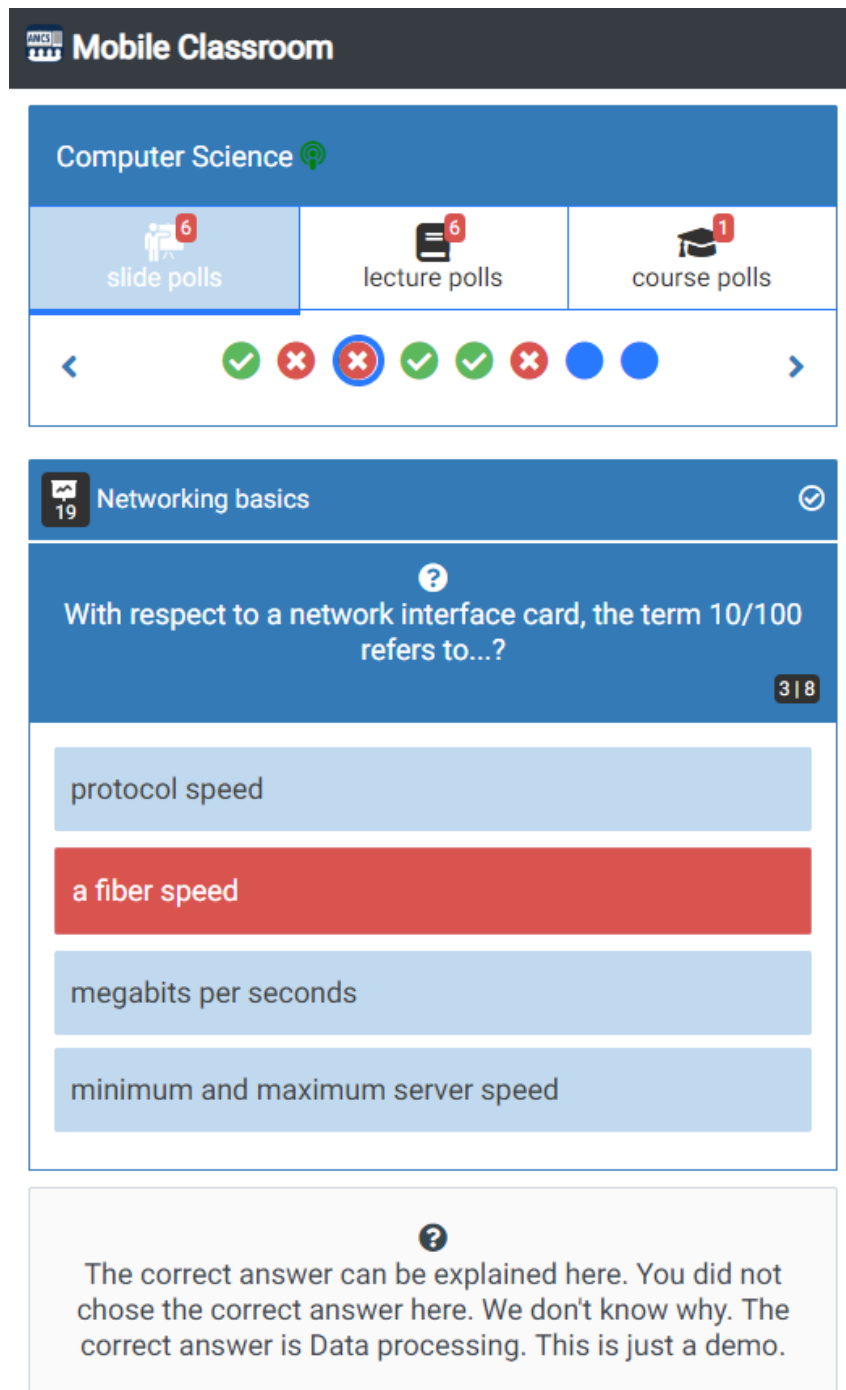


Figure 5.6: Mock-up 2 of the new *Poll View*: Colors are adapted to match the corporate design of AMCS.

## 6 Implementation



## 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...