

Conceptual Modeling & Design. Project report.

“WeStudy” academic social network.

Group composition

Joachim Alvarez-Rodriguez (Joachim.Julio.Alvarez-Rodriguez@vub.be)

Aldar Saranov (Aldar.Saranov@ulb.ac.be)

Description

The application will give a platform for students to discuss topics of a course with each other. Help from the teachers can also be asked. More detailed description can be found in the submitted description paper.

UI design method

UI design method was chosen in order to perform design process. It starts from specifying user classes.

User classes

Two major types of users are supposed to interact with the system - student, teacher. There is also a special kind of user, school. The user type and privileges are determined automatically at the login stage.

Student

Students can follow courses. Students can discuss topics with each other or ask the teacher for help. A student can make a new post on one of his courses, this new post can be hidden. He should also be able to add a comment on one of the existing posts. While inside a post he can choose to request help from the course teacher. Also he should be able

to change his profile settings, like edit his password or change his avatar.

- Type of user: direct.
- Motivation/goals: discussing course topics with other students or the teacher.
- Task Experience: advanced beginner to competent performer.
- Frequency of use: Occasionally to regular.
- Task knowledge: low.
- Use: mandatory.
- Computer experience: advanced beginner to competent performer.
- Number of users: more than 10 000.
- Training: not required.
- Tasks performed: see later.

Teacher

Teacher can also follow courses that he/she provides. A teacher can make a new post on one of his courses and comment thereof. Similarly he/she can edit his profile settings. In addition, a teacher should be able to view all the hidden posts, and also tag certain posts as pinned.

- Type of user: direct.
- Motivation/goals: Help and discuss with the students on course topics.
- Task Experience: advanced beginner to competent performer.
- Frequency of use: Occasionally.
- Task knowledge: average.
- Use: mandatory.
- Computer experience: competent performer.
- Number of users: more than 1 000.
- Training: 1-2 hours.
- Tasks performed: see later.

School

School are the predefined accounts of the school supervisors, the main purpose of which is to create student and teacher accounts. Course pages can also be created or deleted by the school.

- Type of user: direct.
- Motivation/goals: managing the system accounts and course pages.
- Task Experience: competent performer.
- Frequency of use: daily.
- Task knowledge: high.
- Use: mandatory.
- Computer experience: competent performer.
- Number of users: more than 3.
- Training: a day.
- Tasks performed: see later.

Usability requirements

Allow users to make a new post for a course.

User classes:

Student, Teacher

Associated usability requirements

- Making a new post should be possible in less than 5 minutes

Motivation: in order to announce a message related to the study subject, the student needs to be able to do so easily in a straightforward way. Otherwise he will get too annoyed to use the application in the first place.

Measuring concept: Quality of task performance

Measuring method: Task scenario

- Result: Time to perform the task

Criteria for judging:

- Current level: n.a.
- Worst level: 10 min
- Planned level: 5 min
- Best level: 1 min.

- Making a new post should be possible after minimal training.

Motivation: since this is one of the primary functions of the system and the system is casual user oriented, it should not be difficult to do this.

Measuring Concept: Learnability

Measuring Method: Task scenario after a short training

- Result: Percentage of errors

Criteria for judging:

- Current level: n.a.
- Worst level: 10 %
- Planned level: < 5%
- Best level: 2 %.

Allow low level post options.

User classes:

Student

Associated usability requirements

- hidden option

Motivation: We want to ensure that students can exchange private information with the teacher without bothering other students.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.

- Best level: 100%.

- requested option

Motivation: If help is required for the topic, then a student should be able to ask for a teacher.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

Allow high level post options.

User classes:

Teacher

Associated usability requirements

- pinned option

Motivation: some posts have valuable information, and thus have to be shown in the highest priority.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

- requested option

Motivation: a teacher should be able to untag the request posts after the problem is resolved.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

Allow users to add a comment to a post.

User classes:

Student, Teacher

Associated usability requirements

- Making a new comment should be possible in less than 2 minutes

Motivation: in order to announce a message related to the study subject, the student needs to be able to do so easily in a straightforward way. Otherwise he will get too annoyed to use the application in the first place.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Time to perform the task

Criteria for judging:

- Current level: n.a.
- Worst level: 4 min
- Planned level: 2 min
- Best level: 1 min.

- Making a new comment should be possible after minimal training.

Motivation: since this is one of the primary functions of the system and the system is casual user oriented, it should not be difficult to do this.

Measuring Concept: Learnability

Measuring Method: Task scenario after a short training

– Result: Percentage of errors

Criteria for judging:

– Current level: n.a.

– Worst level: 10 %

– Planned level: < 5%

– Best level: 2 %.

Allow users to edit their profile settings.

User classes:

Student, Teacher

Associated usability requirements

- Allow the user to edit their password.

Motivation: For security reasons, the password should be able to be edited on user's accord without special training. This will ensure higher account safety.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

– Result: Percentage of tasks successfully completed

Criteria for judging:

– Current level: n.a.

– Worst level: 90%.

– Planned level: 95%.

– Best level: 100%.

- Allow the user to edit their avatar.

Motivation: Having avatars provides a higher user experience, give a personal touch and makes it feel more like the forums that everyone is already used to.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

Allow users to edit their course list.

User classes:

Student, Teacher

Associated usability requirements

- Allow the user to join a new course.

Motivation: In order to be able to exchange information related to the courses from the new study year, that he/she was assigned to, the user should be able to join new courses.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

- Allow the user to leave his/her course.

Motivation: Since a user might not want to receive irrelevant information from the courses that he/she passed or was unassigned from, the user should be able to leave those.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

Obtain a special overview of the posts.

User classes:

Teacher.

- Allow to see hidden posts per course

Motivation: Having an overview of the hidden posts will make it easier for the user to go through them.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

- Allow to see requested posts per course

Motivation: Having an overview of the requested posts will make it easier for the user to see where his assistance is needed.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

Allow to manage user accounts.

User classes:
School.

Associated usability requirements

- Allow to create student and teacher accounts.

Motivation: In order to prevent the registration of random people, the accounts should be issued by the responsible person, selected by the school.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

Allow to manage courses.

User classes:
School.

Associated usability requirements

- Allow to create course pages.

Motivation: In order to prevent the course creation by unauthorised people, these should be done by the responsible person.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

- Result: Percentage of tasks successfully completed

Criteria for judging:

- Current level: n.a.
- Worst level: 90%.

- Planned level: 95%.
- Best level: 100%.

- Allow to delete course pages.

Motivation: Once certain courses have completed their term or were abolished, they should be deleted by the responsible person, in order not to overload the system content by courses that are not actual.

Measuring Concept: Quality of task performance.

Measuring Method: Task scenario

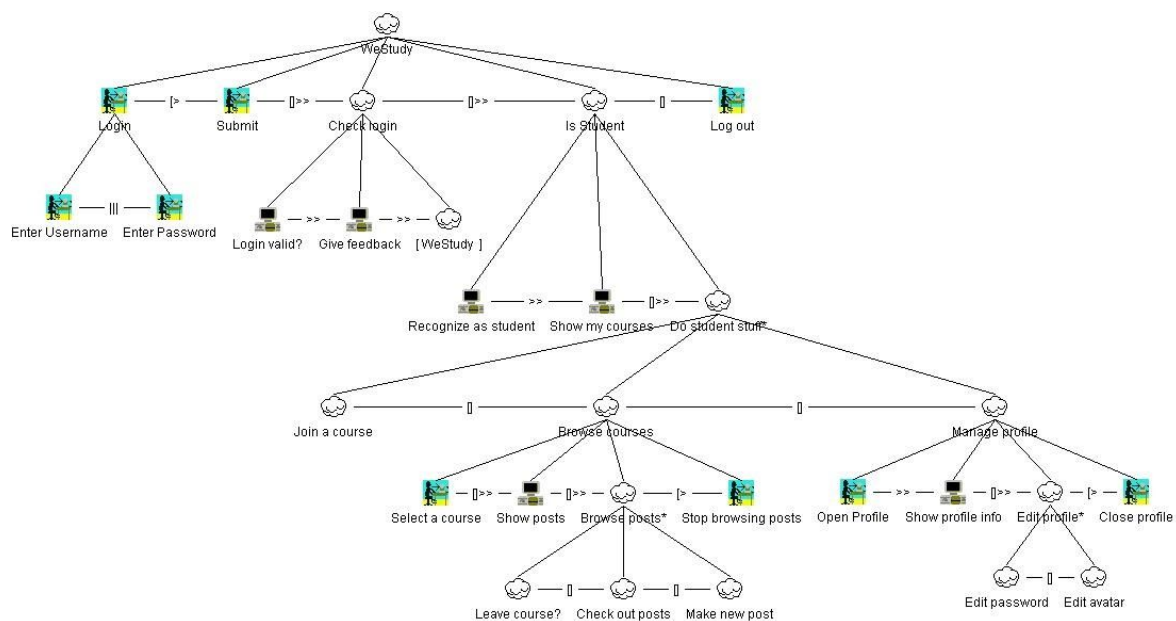
- Result: Percentage of tasks successfully completed

Criteria for judging:

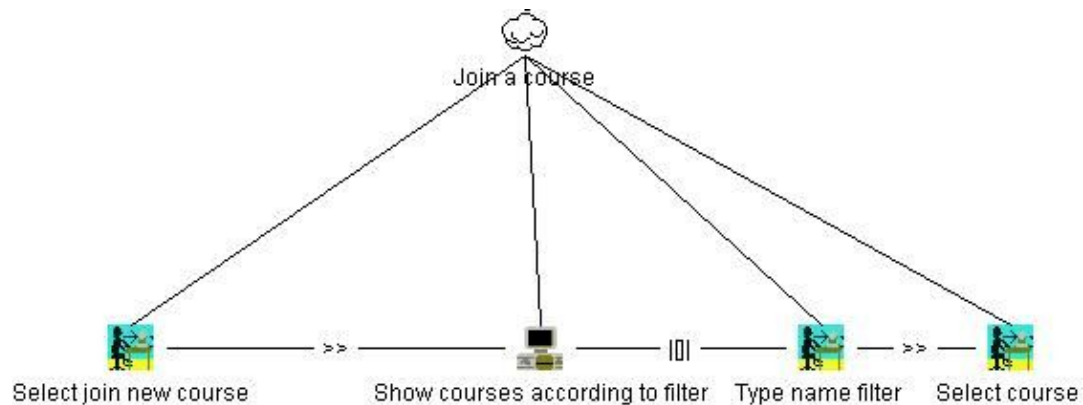
- Current level: n.a.
- Worst level: 90%.
- Planned level: 95%.
- Best level: 100%.

User tasks - student

The complete CTT.



Student - task 'join a course'



Type: typical

Situation: given a course name, a student wants to join this course.

Script:

[Student]: selects 'join new course'.

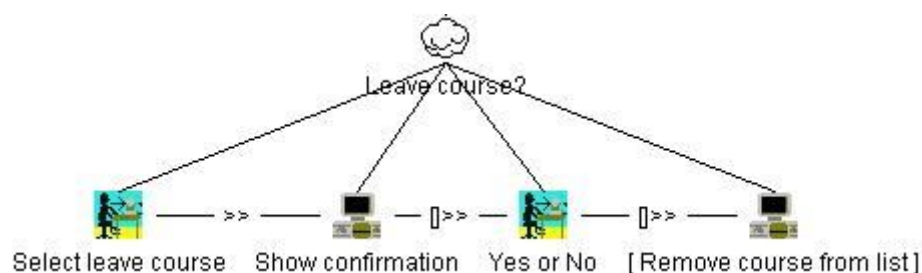
[System]: shows course list without filter applied

[Student]: types the filter name.

[System]: shows course list with regard to the filter.

[Student]: selects a course.

Student - task 'leave a course'



Type: typical

Situation: given a course name, a student wants to leave this course.

Script:

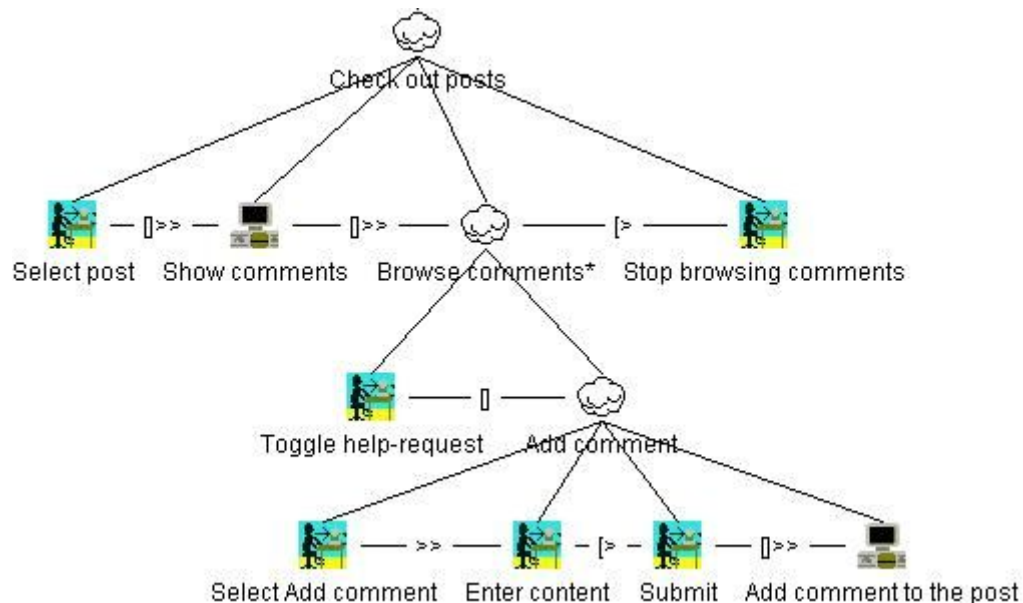
[Student]: selects 'leave course' option.

[System]: shows confirmation of leaving.

[Student]: clicks yes.

[System]: removes this course from this student's list.

Student - task 'check out posts'



Type: typical

Situation: having entered a course page, toggle help-request to a given post and add a new comment.

Script:

[Student]: selects the specified post in the post list.

[System]: shows the comment list.

[Student]: toggles help-request.

[Student]: selects 'add comment' option.

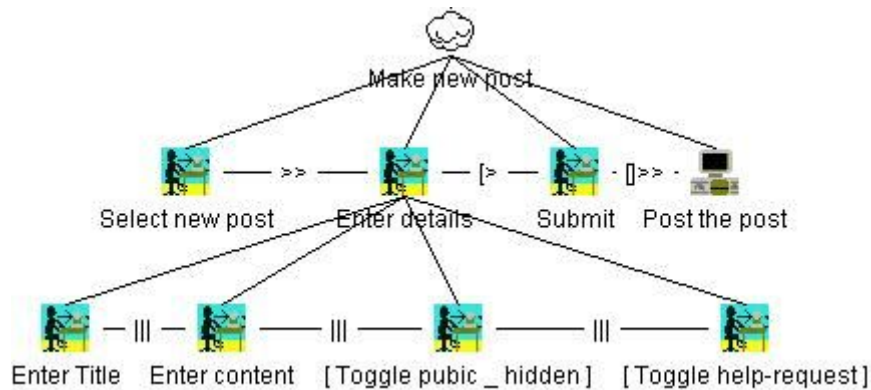
[Student]: enters the content.

[Student]: submits the comment.

[System]: adds the comment to the database and updates the view.

[Student]: stops browsing comments.

Student - task 'make a new post'



Type: typical

Situation: having entered a course page, make a new hidden post.

Script:

[Student]: selects 'new post' option.

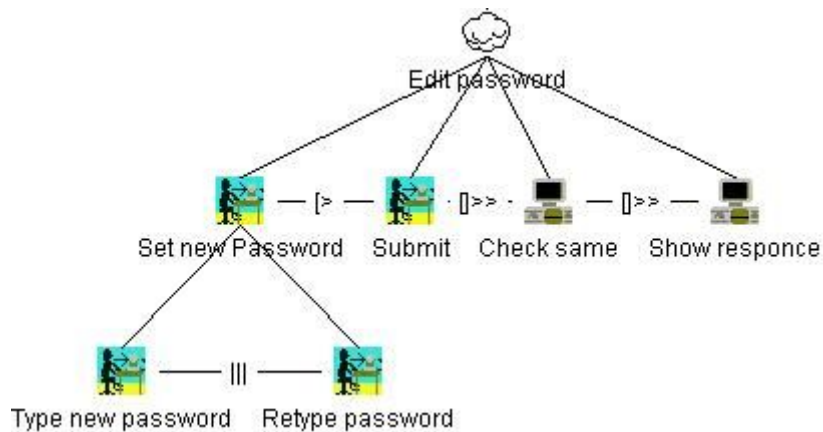
[Student]: enters the post title.

[Student]: enters the post content.

[Student]: toggles the hidden flag.

[Student]: submits the post.

Student - task 'edit password'



Type: occasional

Situation: having entered the profile window, successfully edit the account's password.

Script:

[Student]: types new password.

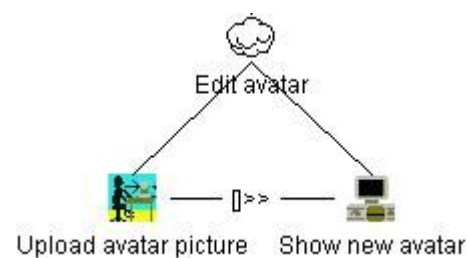
[Student]: re-types new password.

[Student]: submits new password.

[System]: checks that the two passwords are the same. They are same.

[System]: shows positive response of the password changing.

Student - task 'edit avatar'



Type: occasional

Situation: having entered the profile window, successfully edit the account's avatar.

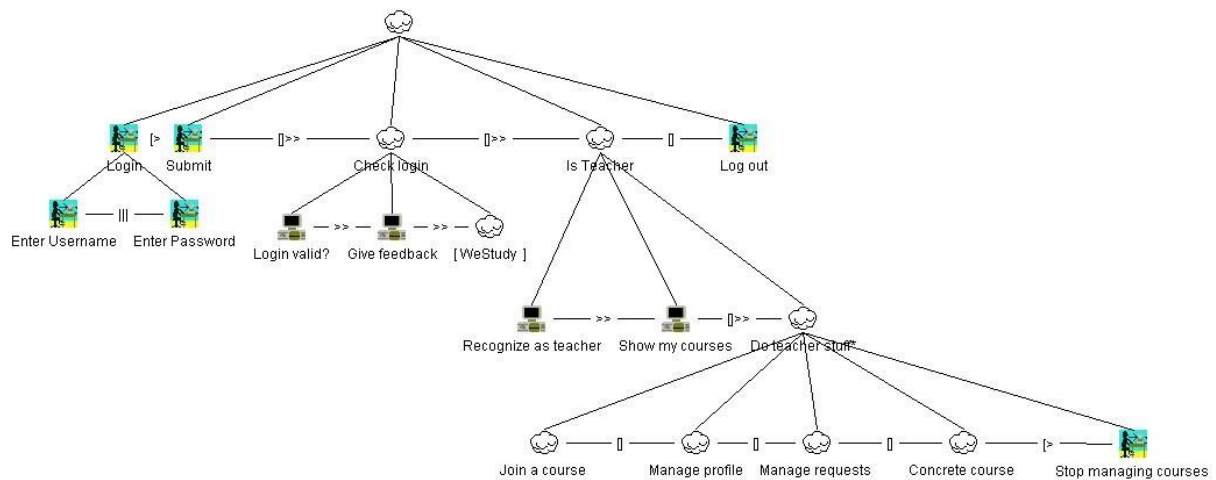
Script:

[Student]: uploads an avatar picture.

[Student]: shows new avatar.

User tasks - teacher

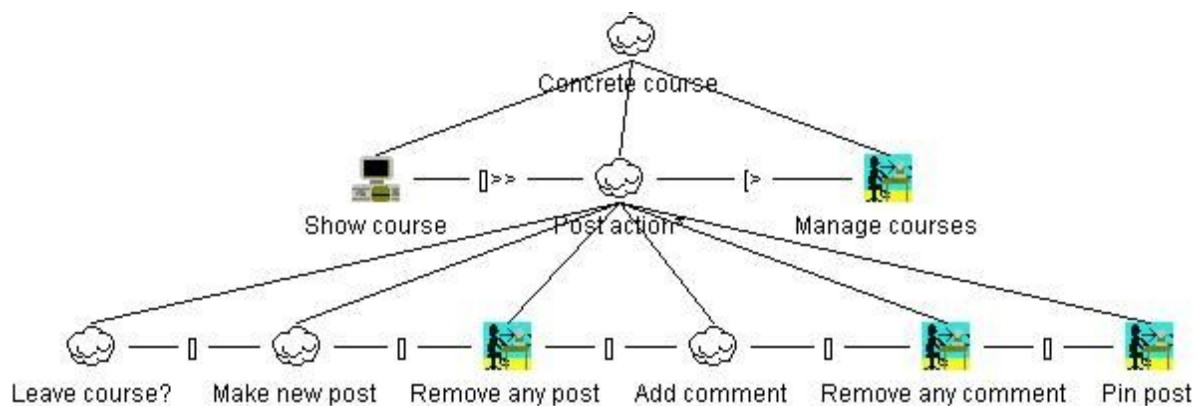
The complete CTT.



Remark:

- Teacher 'Join a course' task is identical to the student's one.
- Teacher 'Manage profile' task is identical to the student's one.

Teacher - task 'concrete course'



Type: typical

Situation: having entered the course page, remove a spam comment, add one new comment and pin one of the posts.

Script:

[System]: shows the course.

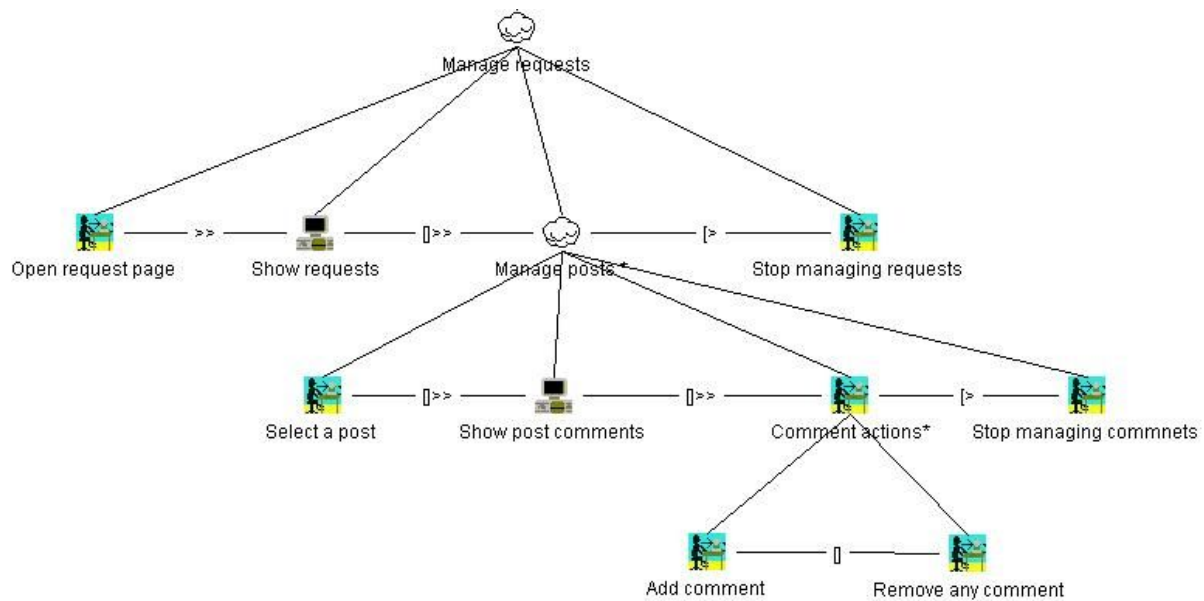
[Teacher]: removes the specified spam comment.

[Teacher]: adds a new comment.

[Teacher]: pins the specified posts.

[Teacher]: goes to 'manage courses' page.

Teacher - task 'manage requests'



Type: typical

Situation: having entered the teacher services, go to an obtained request and answer by a comment.

Script:

[Teacher]: opens request page.

[System]: shows requests.

[Teacher]: selects a post.

[System]: shows post comments.

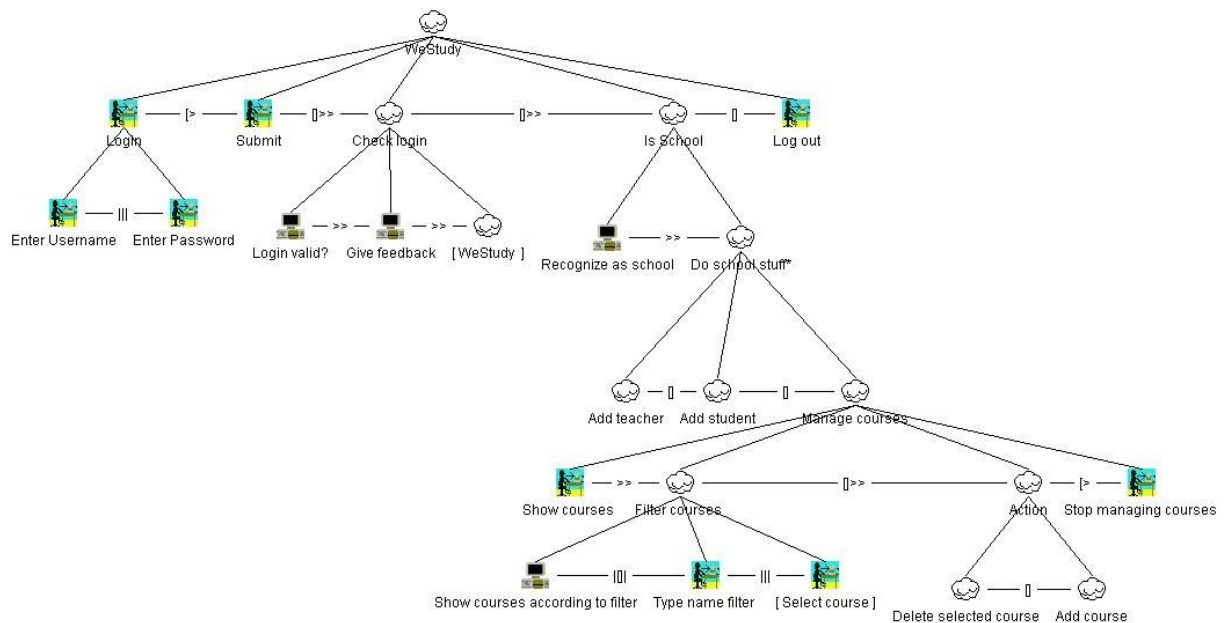
[Teacher]: Add comment.

[Teacher]: stop managing comments.

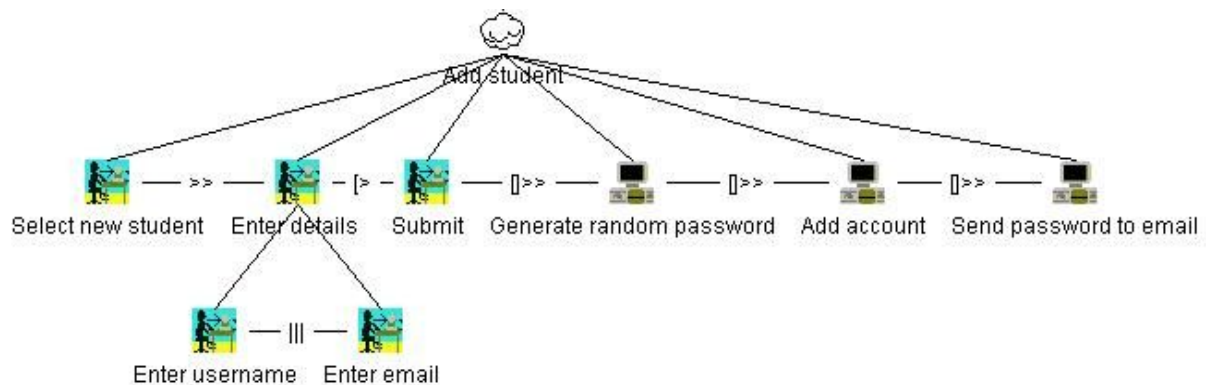
[Teacher]: stop managing requests.

User tasks - school

The complete CTT.



School - task 'add a new student account'



Type: massively several times a year.

Situation: having entered the school services, add a new student account.

Script:

[School]: selects a 'new student' option.

[School]: enters username.

[School]: enters e-mail.

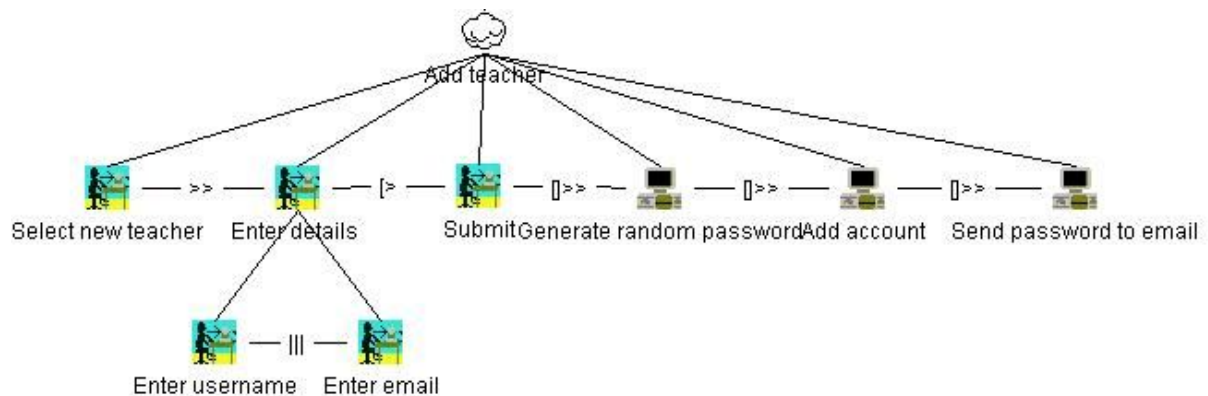
[School]: submits the student.

[System]: generates a random account password.

[System]: adds the account to the database.

[System]: sends the password via e-mail.

School - task 'add a new teacher account'



Type: occasionally.

Situation: having entered the school services, add a new teacher account.

Script:

[School]: selects 'new teacher' option.

[School]: enters teacher's username.

[School]: enters e-mail.

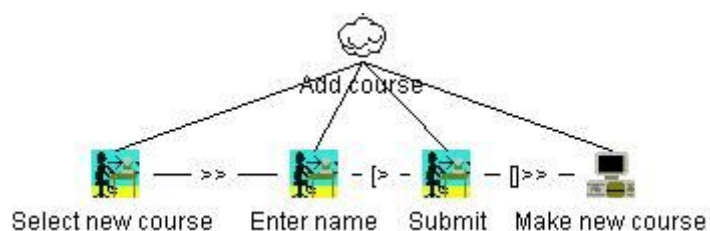
[School]: submits the account.

[System]: generates a random account password.

[System]: adds the account to the database.

[System]: sends password via e-mail.

School - task 'add a new course'



Type: very occasionally.

Situation: having entered the school services, add a course to the program.

Script:

[School]: selects 'new course' option.

[School]: enters the course name.
[School]: submits the course.
[System]: adds the course to the database.

School - task 'delete selected course'



Type: very occasionally.

Situation: having entered the school services, delete the specified course.

Script:

[School]: selects delete course.

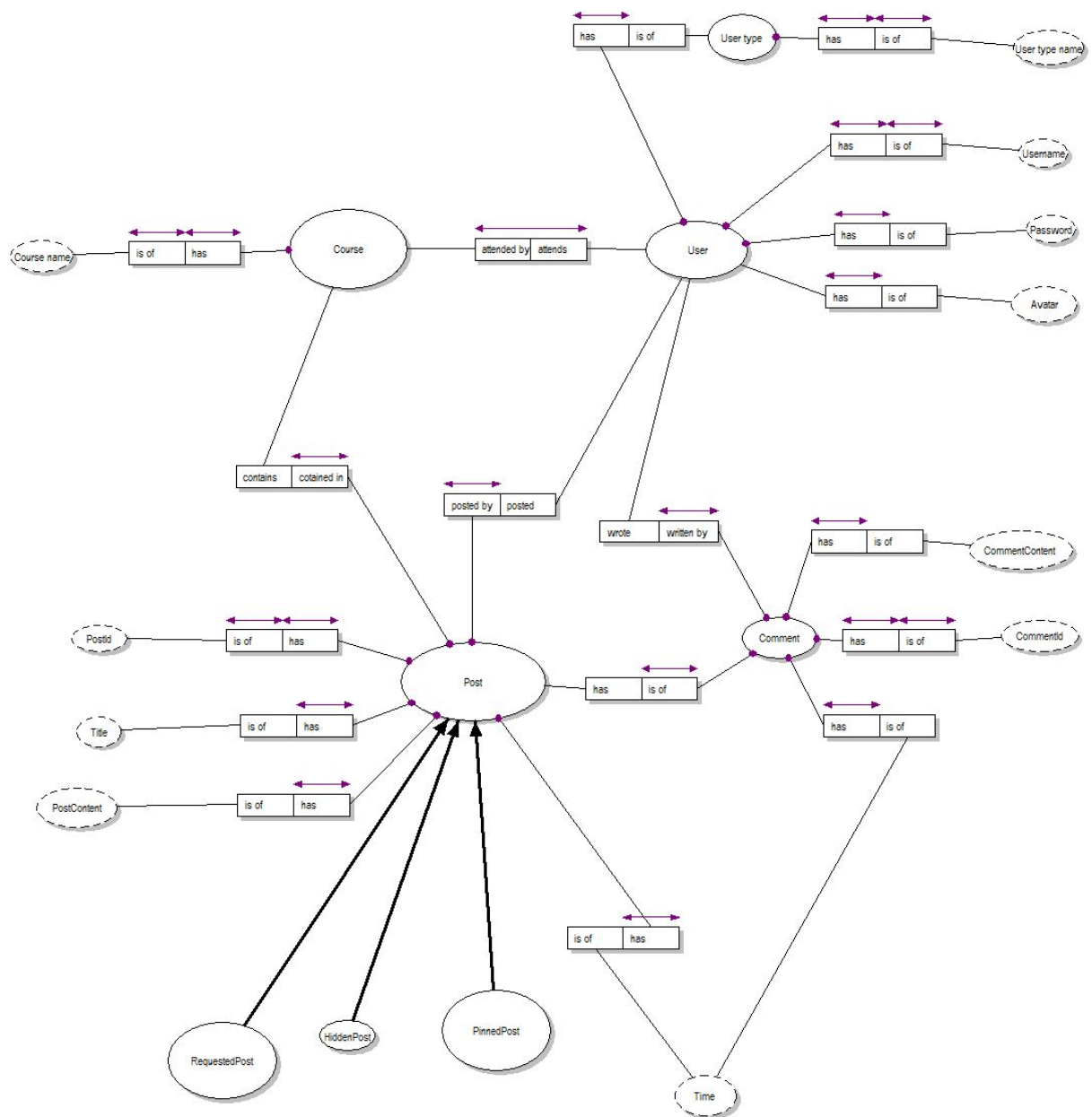
[System]: shows confirmation of the delete process.

[School]: confirm deletion.

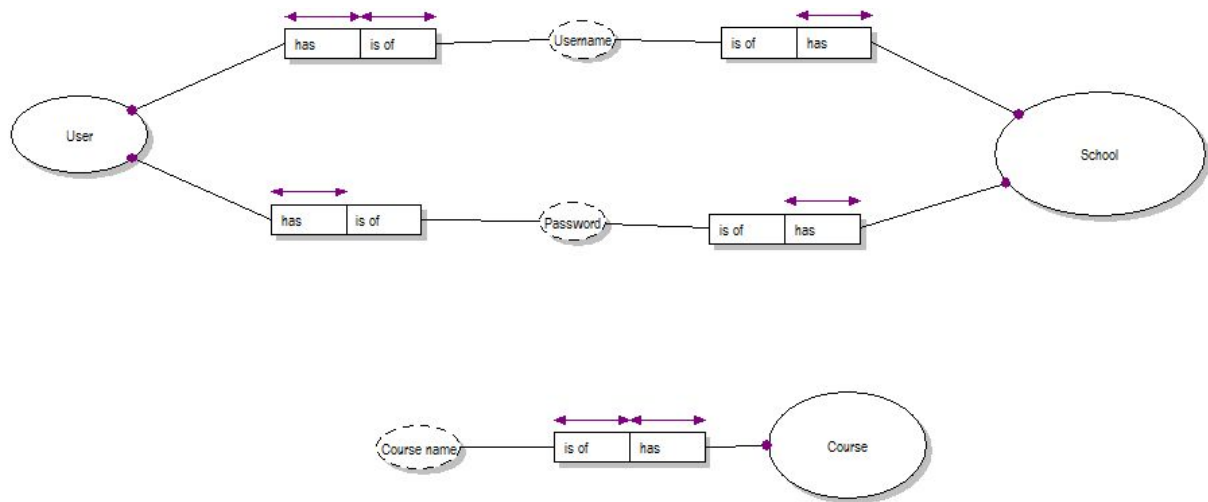
[System]: deleted the course from the database

User object models

Normal users (student or teacher) have an object model as following:



School responsible supervisors have an object model as following:



The style guide fragments

Since we are developing an application for Android device, we have to use the corresponding set of visual component tools for the UI design. These components work for any Android device of the specified version, without depending on concrete model, screen size and so on. The style guide is application-oriented, not corporate since we have nothing to inherit the styles from.

However we are using the standard Android visual components styles description of which can be found here:

<https://developer.android.com/design/index.html>

Android toolbar

One of the visual components that is common for all the activities (visual views) for this application is the toolbar. Toolbar is located in the top part of the screen and provides some options that are specific for a certain activity and a logout option.

A drop down menu is located in the right corner of the toolbar and provides the toolbar options. Some toggles can be also found in toolbars

which are used for switching request, hidden, pinned statuses of the posts.

The content is placed below, on the rest space of the screen.

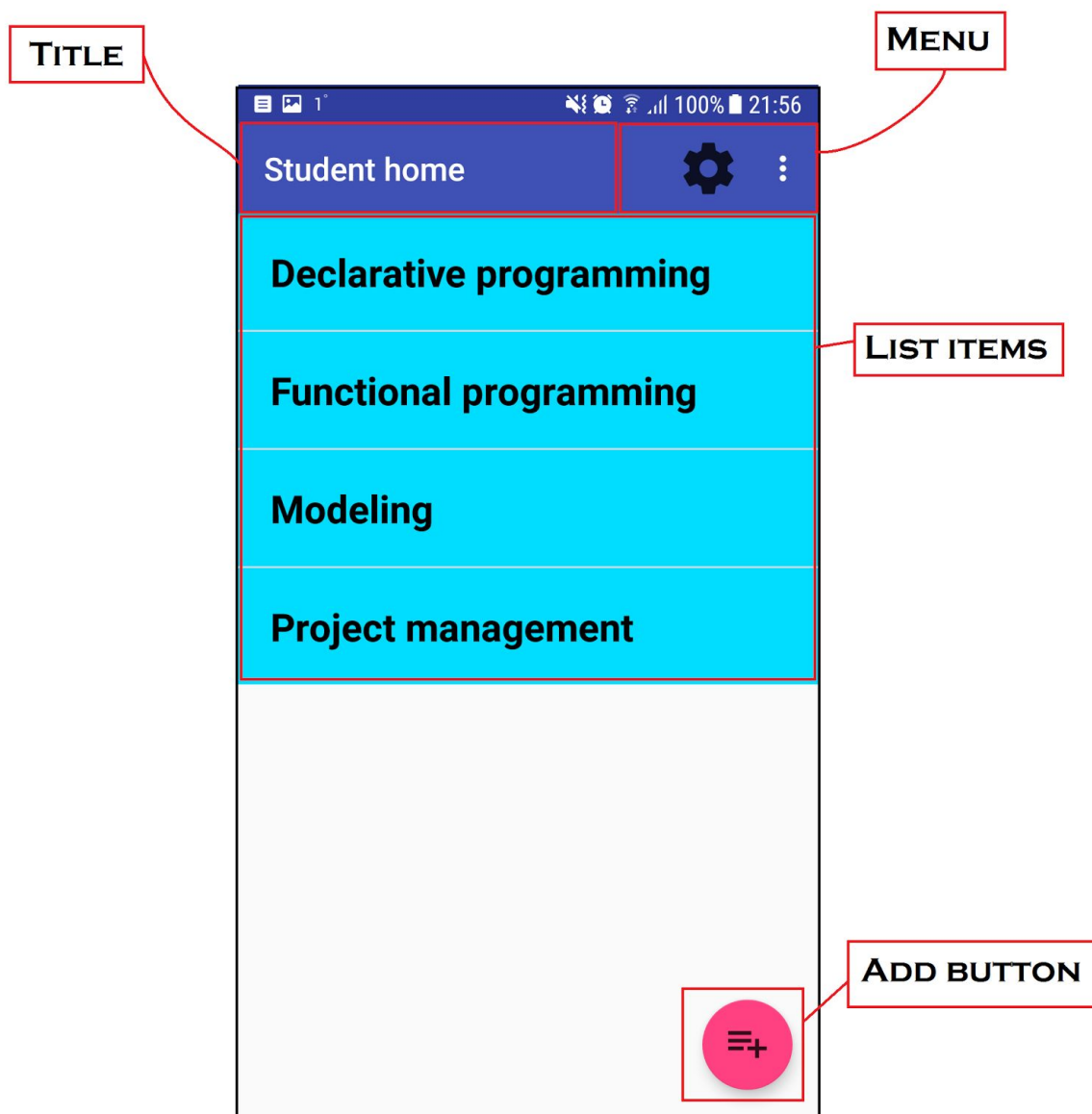
The screenshot shows a mobile application interface for creating a new post. At the top, there is a dark blue header bar with the text "New post for Fun" in white. Below the header, the main content area is light green and contains two input fields: "Title" and "Content". A white dropdown menu is open over the "Content" field, displaying three options: "Hidden" with an unchecked checkbox, "Requested" with an unchecked checkbox, and "Log out". The status bar at the very top of the screen shows various icons including a menu, camera, signal strength, and battery level at 100%, along with the time 23:01.

Android list view

List view is used for demonstrating such things as course list, post list and comment list. They consume the whole screen width and allow top-down navigation over the items.

Android floating action button

These buttons are normally used for adding some content (such as new posts and comments).

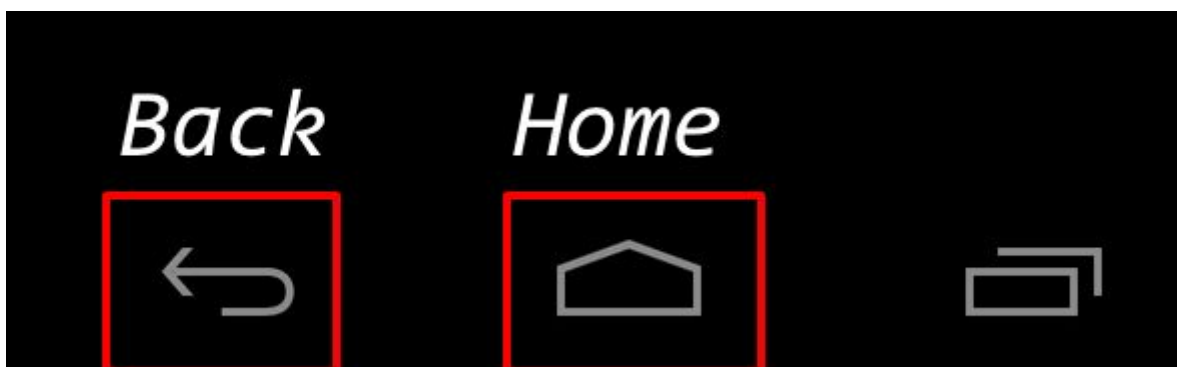


Password field

For sake of security, the password text field should hide the specified characters and use stars instead.

Standard application panel

This panel allows to use the return button in order to return to the previous activity of the application. The application can also be paused & hidden if one touches the “Home” button.

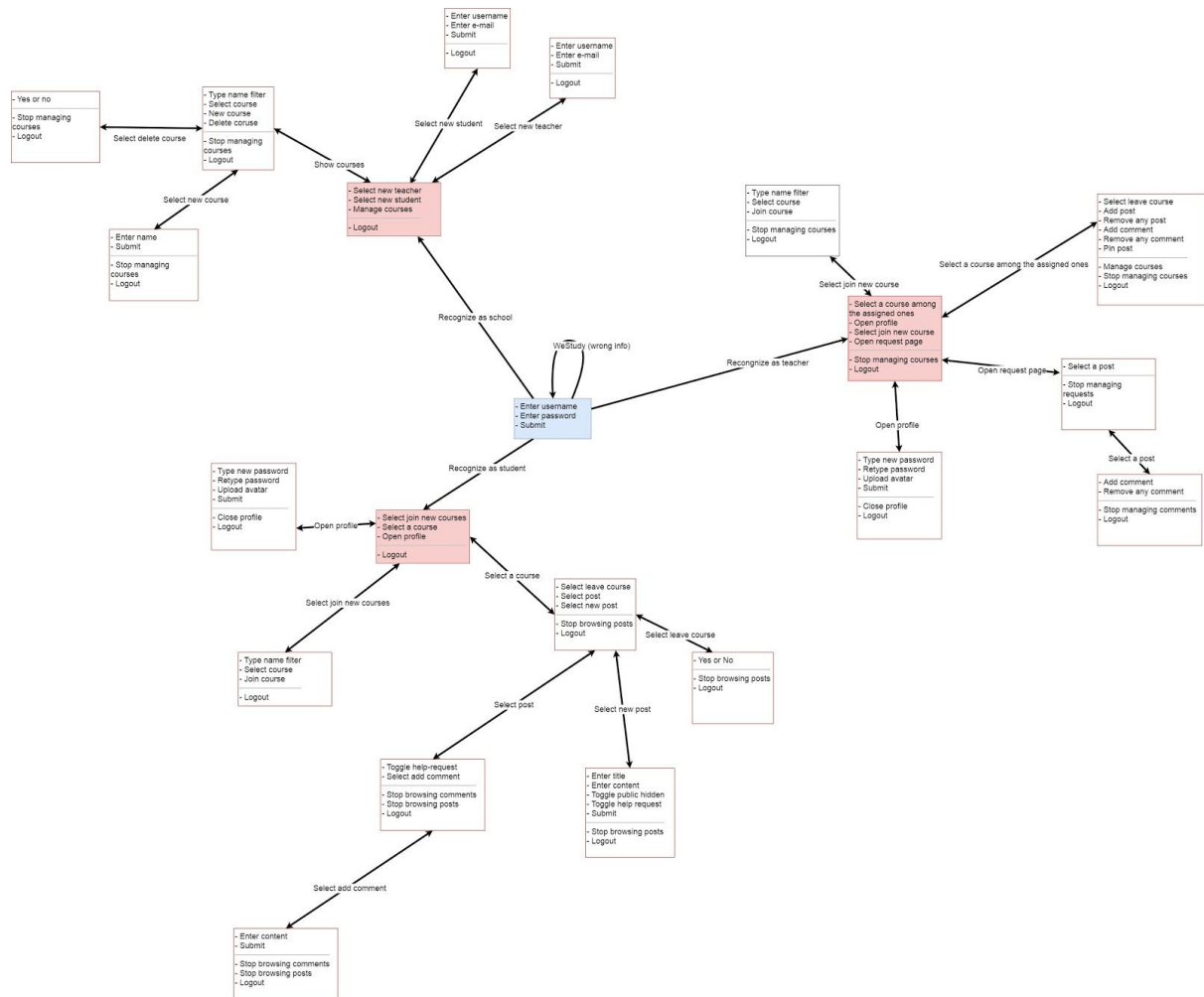


Standards for data types:

- Date: YYYY-MM-DD
- Hours: 24h format: hh:mm:ss
- Boolean: Checkbox / switch
- Overview of items: listview

Design UI

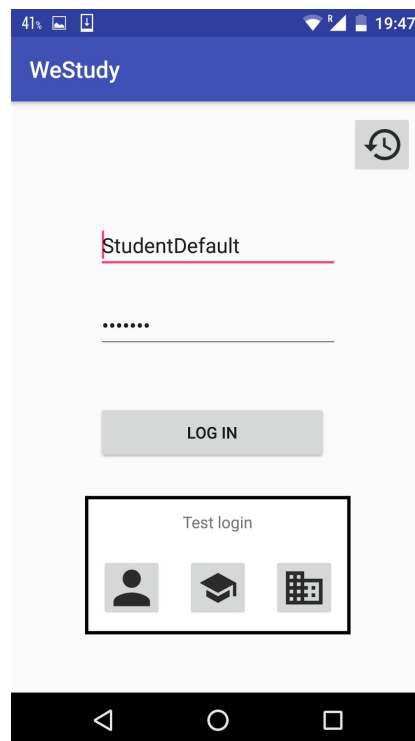
The following enabled task set was derived from the CTTs:



In our case every separate task set should be shown in a separate view, i.e. a separate activity because we use android UI notation.

Prototype UI

Log-in (common view)



The image shows a mobile application interface for 'WeStudy'. At the top is a blue header bar with the text 'WeStudy'. Below the header, there is a light gray background area. In the top right corner of this area is a circular icon with a clock and a refresh symbol. The main content area contains a text input field with the placeholder text 'StudentDefault', followed by a password input field represented by seven dots. Below these fields is a gray button labeled 'LOG IN'. At the bottom of the main content area is a box titled 'Test login' which contains three icons: a person, a graduation cap, and a building. The entire screen is framed by a black Android-style navigation bar at the bottom with back, home, and recent apps buttons.

41% 19:47

WeStudy

StudentDefault

.....

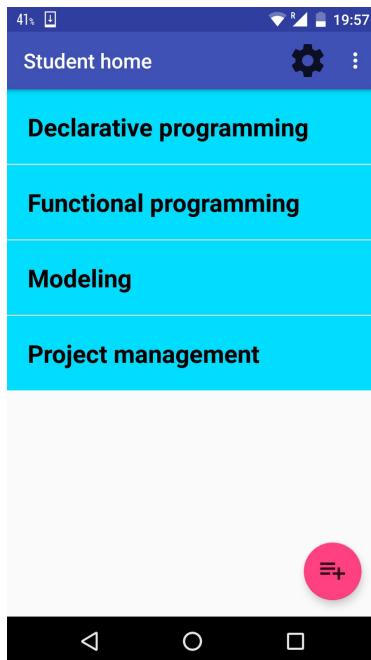
LOG IN

Test login

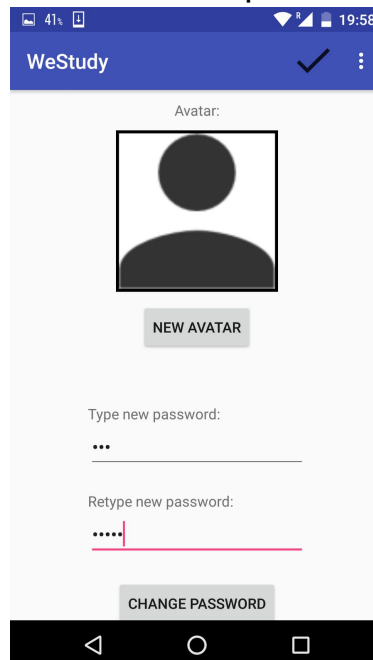
Person icon, Graduation cap icon, Building icon

Student

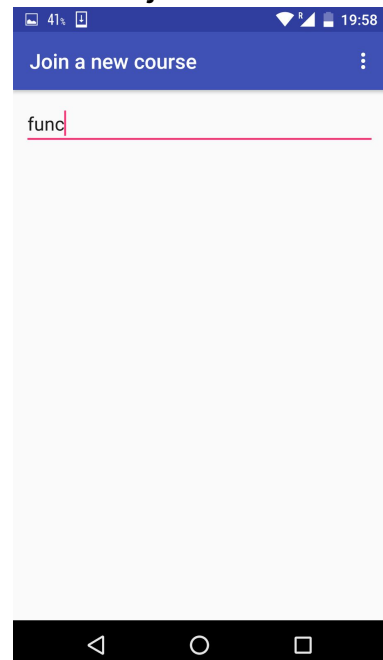
Student home



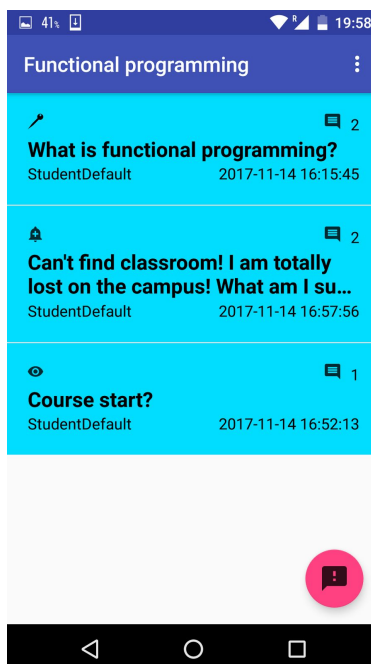
Edit student profile



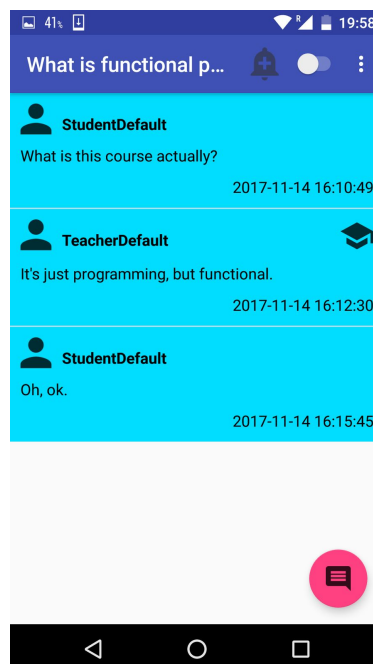
Student joins a course



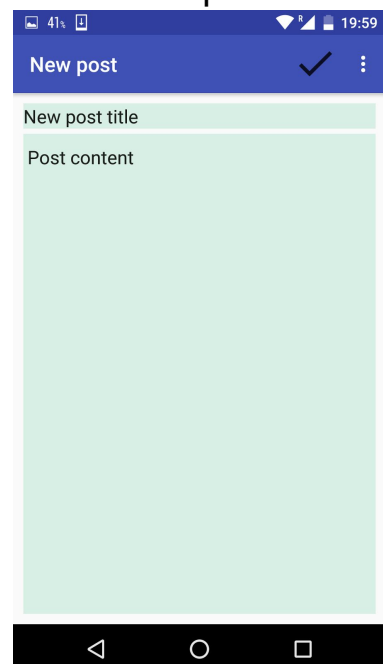
Course list



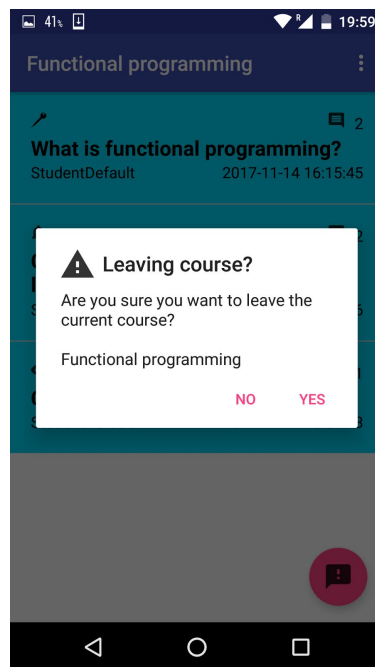
Comment list



New post

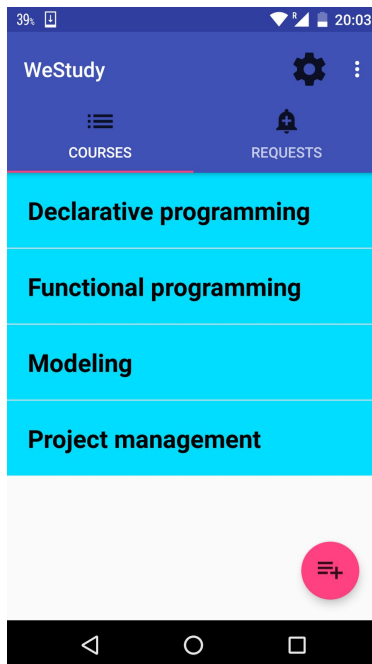


Confirm leaving a course

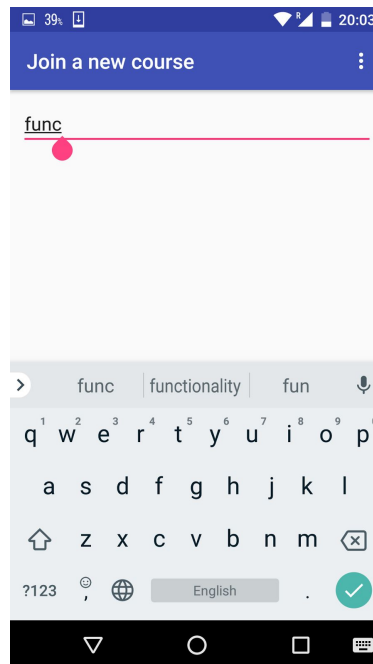


Teacher

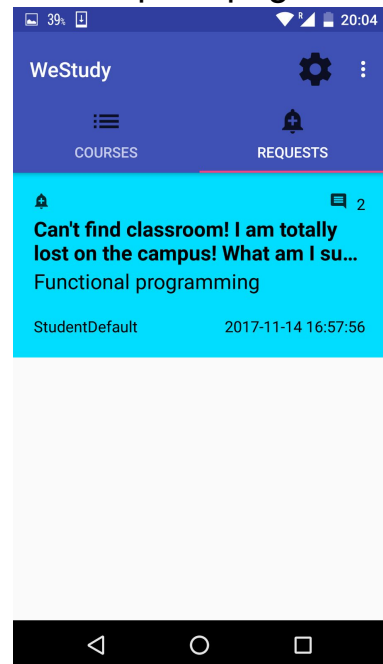
Course list



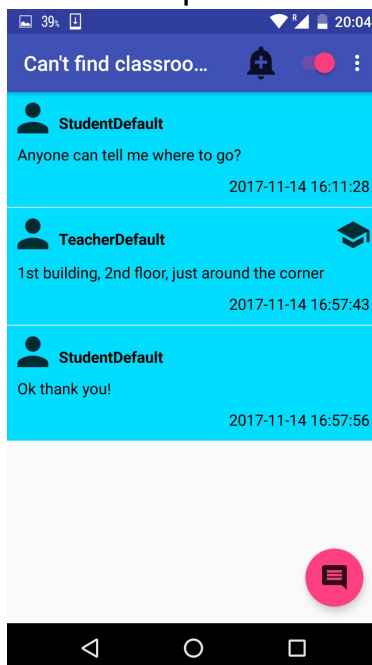
Joins a new course



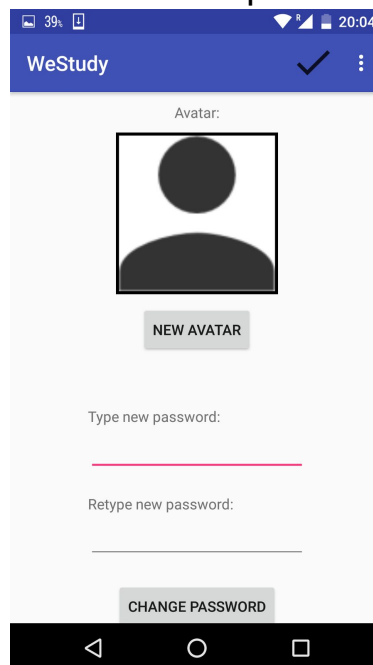
Request page



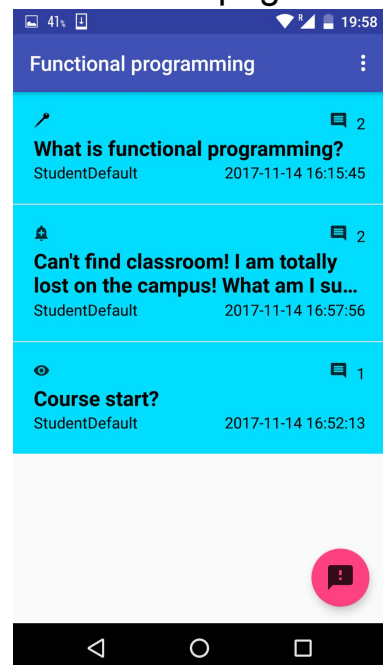
Course posts list



Edit teacher profile

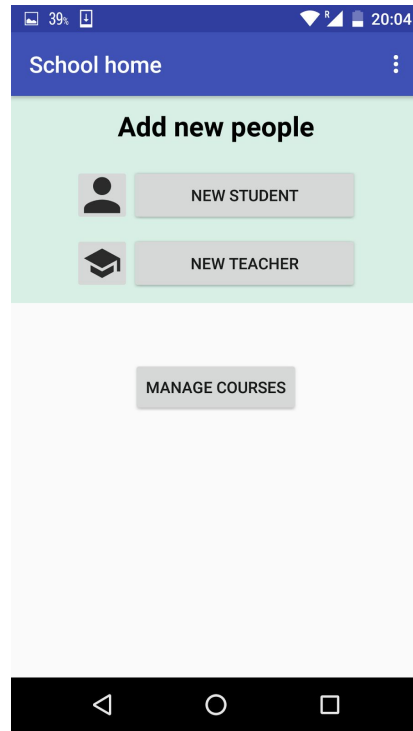


Course page

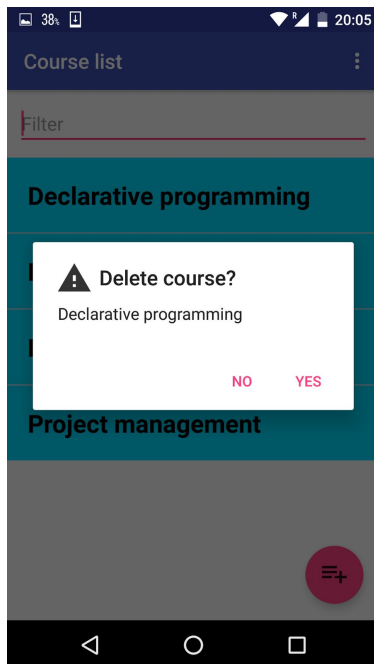


School

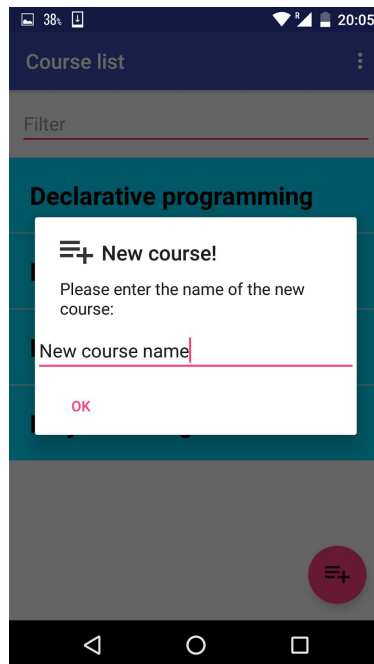
School home page



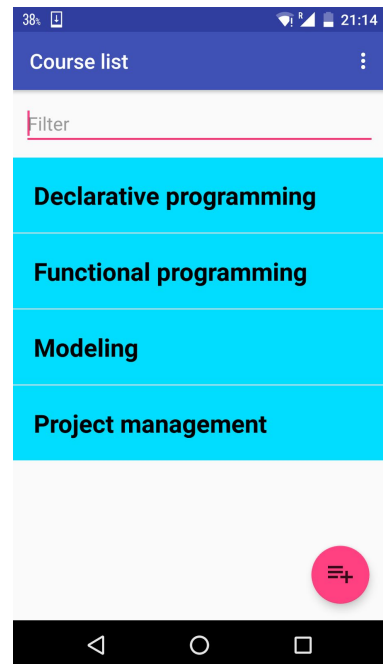
Confirm delete course



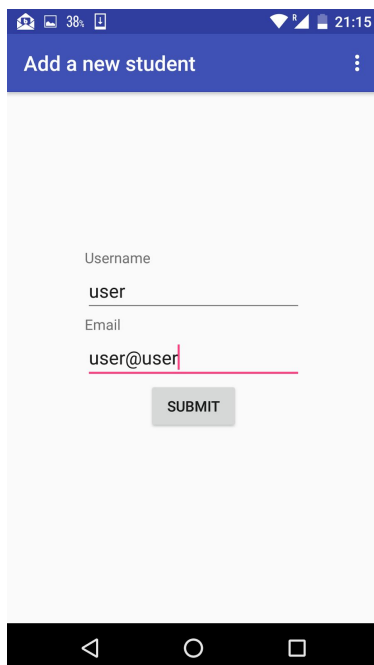
Create a new course



Find courses



Add a new student



Add a new teacher

