1. INTRODUCTION

1.1 PURPOSE

The purpose of this Software Requirements Specification (SRS) document is to provide a detailed description of the functionalities of the eVoter system. This document will cover each of the system's intended features, as well as offer a preliminary glimpse of the software application's User Interface (UI)

1.2 DOCUMENT CONVENTIONS

This document features some terminology which readers may be unfamiliar with. See Appendix A (Glossary) for a list of these terms and their definitions.

1.3 PROJECT SCOPE

The eVoter system is composed of two main components: a client-side application (for teacher and student) which will run on Android handsets and browser, a server-side application which will support and interact with various client-side features.The system is designed to facilitate the process of sending questions and collecting information in real time.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The eVoter project is a self-contained product intended for use on the Android and web platform. While the eVoter mobile application is the main focus of the project, there is also a server-side component which will be responsible for database and synchronization services. The scope of the project encompasses both server- and client-side functionalities, so both aspects are covered in detail within this document.

2.2 PRODUCT FEATURES

The following list offers a brief outline and description of the main features and functionalities of the eVoter system

1. USER REGISTRATION AND LOGIN

1.1 User sign up

- Student and teacher will go to a certain link and register his information (his email will be validated) and request an account.
- The account is an identity ID (created by him and unique in the system) and a password (created by him and more than 8 characters with at least one special

character and one number) sent via his email.

1.2 User login

- Users will just login once when they open the application by their account registered with the system

1.3 User log-out

- When the student logs-out the client-side application, he could not vote any more and his name will disappear on the screen of teacher.
- When teacher logs-out, his current session will be deactivated and he could not receive any vote from his students.

1.4 Password recovery

- The system can provide user with an new password if user can enter exactly his email or identity.

2. QUESTION MANAGEMENT

2.1 Create a question

Teacher can create a new question, send it immediately or save and send later.
Teacher can select a question, edit and save it as a new one.

2.2 Edit a question

Teacher can update something new for an existed question created by his sessions.

2.3 Delete a question

Teacher can delete an existed question created by his sessions. That means the answers and the statistics of this question will be deleted too.

2.4 Send a question.

Teach can send one by one question in his list to the audience list in the active session.

2.5 View a question

Teacher can view a list of question and view more in detail question and its answers.

2.6 View statistic of the number of votes for one question

Student can compare their answer with class's for each question
Teacher can view the vote statistic of each question after sending. However, he could not see the individual votes.

2.7 Save statistic

The statistic can be saved automatically for both kinds of user: teacher and student.

2.7 Search question

Users enter some keywords and a list of question is returned.
This function is also used for teachers when creating a question. They can search and save as a new one after editing.

3. SESSION MANAGEMENT

3.1 Create a session

Teacher can create one session for his lecture anytime. This session will be

saved default after creating.

3.2 Delete a session

When the teacher logs-in to the system, he can view his sessions and free delete one or many of them. That means all questions, their vote statistics are deleted too.

3.3 View audiences

Teacher can view the number of students and unknown users accessing his session at that time. He could not the answers of each student except unknown users.

3.4 View session

In his session, teacher can manage questions, see list of audience and view feedback statistic of his questions.

3.5 Active session

When the teachers log-in to the system, they can view his sessions, select one session and start it.

4. ANSWER MANAGEMENT

- Student can only send an answer once after receiving a question.
- Student can send a(an) happy/unhappy feedback during the session.
- The answers will be saved in three databases of server, teacher and student. So teacher can view the historical number of votes of the questions.
- The feeling feedback will not be saved in the database.
- The answer is only accepted before the teacher “STOP”

5. HISTORY MANAGEMENT

- Users can view their the history data votes under graph format and can delete it.

2.4 OPERATING ENVIRONMENT

- Server Model can run on Windows/Linux/MacOS
- Application client side for both teacher and students can Android platform and web browser.

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

The primary design constraint is the mobile and web platform. Since the application is designated for mobile handsets, limited screen size and resolution will be a major design consideration. Creating a user interface which is both effective and easily navigable will pose a difficult challenge. Other constraints such as limited memory and processing power are also worth considering. eVoter is meant to be quick and responsive, even when dealing with large audiences and transactions, so each feature must be designed and implemented with efficiency in mind.

2.6 USER DOCUMENTATION

The primary goal of eVoter is to facilitate the process of replacing clicker hardware and viewing statistic of number of votes in real time. Consequently, the application will be designed to be as simple to use as possible. Nonetheless, users may still require some supplementary information about each component of the eVoter system. The application will contain two features that offer this: the Tutorial and the Help menu

3. SPECIFIC FUNCTIONAL FEATURES

3.1 USER REGISTRATION

The user will go to registration link and this screen prompts the user to create an account on the eVoter server using an valid email address. Completing this process will create and store an account for the user on the server, enabling all of the application's synchronization capabilities.

3.1.1 STIMULUS/RESPONSE SEQUENCE

Step 1 eVoter application client launches
Step 2 The user is prompted to enter
Step 3 The information is sent to server and stored in the database
Step 4 Registration is completed and user is taken to main screen

3.1.2 USER REQUIREMENT

The user could not be proceed until a valid email address is entered. The application will verify that the user's input is consistent with the format of an email address.

3.1.3 SYSTEM REQUIREMENT

The system ensures that user's information has to encrypted and stored safely.

3.2 USER LOGIN

The user will go to log in link and this screen prompts the user to enter user ID and password. eVoter server will validate user's account.

3.2.1 STIMULUS/RESPONSE SEQUENCE

Step 1 eVoter application client launches
Step 2 The user is prompted to enter user ID and password
Step 3 The information is sent to server and validated in the database
Step 4 Login is completed and user is taken to main screen

3.2.2 USER REQUIREMENT

The user could not be proceed until user ID and password matching database.

3.3. USER LOGOUT

3.3.1 STIMULUS/RESPONSE SEQUENCE

Step 1: User clicks on logout button
Step 2: Application prompts alert to make sure user log-out the system
Step 3: Log-out is completed and user is taken to log-in screen

3.3.2 USER REQUIREMENT

NONE

3.3.3 SYSTEM REQUIREMENT

The system guarantees that this user will disappear on teacher's screen

3.4 PASSWORD RECOVERY

3.4.1 STIMULUS/RESPONSE SEQUENCE

Step 1: User enters an email or user ID

Step 2: The system will validate the user's input in the database
Step3: Show an announcement that user should open his email to get the new password and a link to go back log-in screen

3.4.2 USER REQUIREMENT
The user could not be processed until he enters a correct email or user ID
3.4.3 SYSTEM REQUIREMENT
The system ensures that user's new information will be updated

3.5 CREATE SESSION

3.5.1 STIMULUS/RESPONSE SEQUENCE
Step 1: Teacher clicks 'create session' button
Step 2: Teacher enters session name and clicks 'save' button
Step3: Teacher will see a new tab opened
3.5.2 USER REQUIREMENT
Application- client side is opened and teacher logs-in to the application
3.5.3 SYSTEM REQUIREMENT
New session will be saved in the database on server.

3.6 DELETE SESSION

3.6.1 STIMULUS/RESPONSE SEQUENCE
Step 1: Teacher clicks 'history' button on left side menu
Step 2: A list of session of the teacher will be displayed
Step 3: Select one or many sessions by checking and click 'delete' button
3.6.2 USER REQUIREMENT
Application- client side is opened and teacher logs-in to the application
3.6.3 SYSTEM REQUIREMENT
The system should delete this/these session(s) out database and all of questions as well as answers will be deleted either.

3.7 VIEW AUDIENCE

3.7.1 STIMULUS/RESPONSE SEQUENCE
Step 1: Teacher clicks 'audience' button on left side menu
Step 2: A screen displayed students and unknown users name under icons to show the number of users connected to his session
3.7.2 USER REQUIREMENT
Teacher has to logged-in successfully application client side
Teacher is opening a session
3.7.3 SYSTEM REQUIREMENT
During time the teacher looks at the audience screen, if there is any user logging-in successfully, the system should update immediately.

3.8 CREATE QUESTION

3.8.1 STIMULUS/RESPONSE SEQUENCE

Step 1:Teacher clicks ‘create question’
Step 2: A question form is displayed including name, content, answers type and answer
Step 3: User enters valid data to form
Step 4: User clicks ‘save question’

3.8.2 USER REQUIREMENT

Teacher has to logged-in the application successfully and the session is opening.
The question only created if user can fulfil the form.

3.8.3 SYSTEM REQUIREMENT

The system guarantees that this question is owned by this teacher and only he can delete or edit. The question should be saved into database on server and consistent with all applications.

3.9 DELETE QUESTION

3.9.1 STIMULUS/RESPONSE SEQUENCE

Step 1: Teacher click ‘questions’ on the left side menu
Step 2: The questions list created by this user are displayed
Step 3: User selects one or many questions
Step 4: User clicks ‘delete’ button

3.9.2 USER REQUIREMENT

Teacher has to logged-in the application successfully and the session is opening.
The questions have to be selected before clicking ‘delete’ button

3.9.3 SYSTEM REQUIREMENT

The system guarantees that list of questions displayed is owned by this teacher. The question should be saved into database on server and consistent with all applications.

3.10 EDIT QUESTION

3.10.1 STIMULUS/RESPONSE SEQUENCE

Step 1: Teacher click ‘questions’ on the left side menu
Step 2: The questions list created by this user are displayed
Step 3: User selects one question
Step 4: User clicks ‘edit’
Step 5: Question loaded into a form
Step 6: User click ‘save’

3.10.2 USER REQUIREMENT

Teacher has to logged-in the application successfully and the session is opening.
The questions have to be selected before clicking ‘edit’ button

3.10.3 SYSTEM REQUIREMENT

The system guarantees that list of questions displayed is owned by this teacher. The question should be saved into database on server and consistent with all applications.

3.11 VIEW QUESTION

3.11.1 STIMULUS/RESPONSE SEQUENCE

Step 1: Teacher click ‘questions’ on the left side menu
Step 2: The questions list created by this user are displayed

Step 3: User selects one question
Step 4: User clicks ‘detail’
Step 5: Question loaded into a form
Step 6: User click ‘close’

3.11.2 USER REQUIREMENT

Teacher has to logged-in the application successfully and the session is opening.
The questions have to be selected before clicking ‘detail’ button.
Network connection is available

3.11.3 SYSTEM REQUIREMENT

The system guarantees that list of questions displayed is owned by this teacher. The question should be saved into database on server and consistent with all applications.

3.12 SEND ANSWER

3.12.1 STIMULUS/RESPONSE SEQUENCE

Step 1: Student opens client application
Step 2: The screen will display the question and answers
Step 3: User selects the answer

3.12.2 USER REQUIREMENT

Students can see the question and answers only if the teacher sends it.
Network connection is available.
Teacher’s session is opening.

3.12.3 SYSTEM REQUIREMENT

The system guarantees that teacher can receive all the answers from students in the class. The number of votes will updated real time until teacher stops and moves to another question.
These answers will be saved in database of server, teacher and student. They have to be consistent.

3.13 SEND FEELING FEEDBACK

3.13.1 STIMULUS/RESPONSE SEQUENCE

Step 1: Student opens application client.
Step 2: Student presses happy/unhappy button on the screen.

3.13.2 USER REQUIREMENT

Network connection is available.
Teacher’s session is opening.

3.13.3 SYSTEM REQUIREMENT

The system guarantees that teacher can receive all the feedback from students in the class. The number of votes will updated real time until teacher ends his session.
These answers will not be saved in database.

3.14 VIEW HISTORY

3.14.1 STIMULUS/RESPONSE SEQUENCE

Step 1: User opens application client
Step 2: User click ‘history’ tab on left side of screen

Step 3: User can see a list of session
Step 4: User selects one session
Step 5: User click ‘view’ options

3.14.2 USER REQUIREMENT

User has to log-in successfully

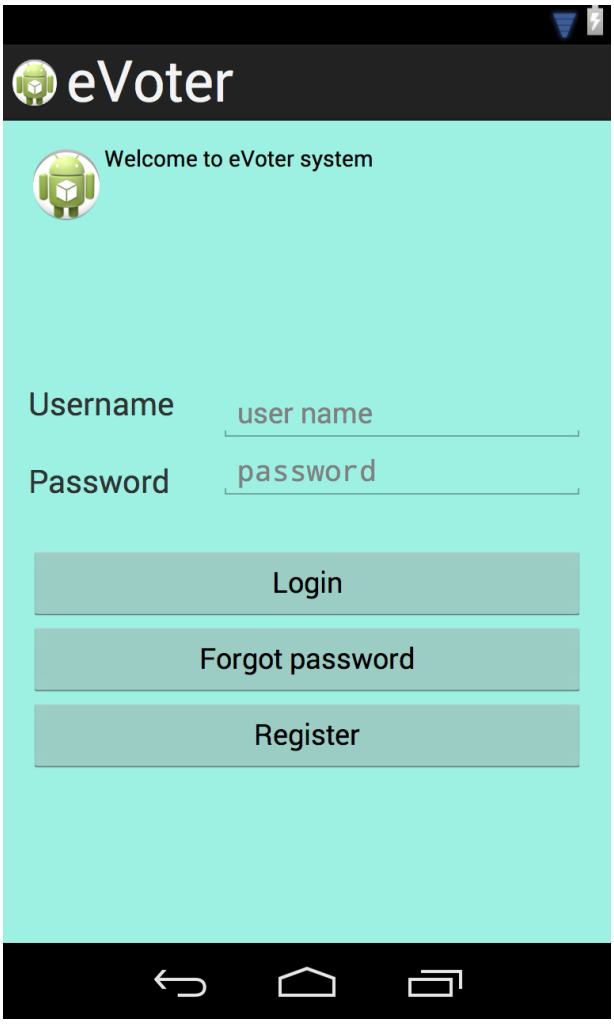
3.14.3 SYSTEM REQUIREMENT

The systems only loads the sessions owned by logged-in user.

The system allows the users delete their history such as session and statistic vote

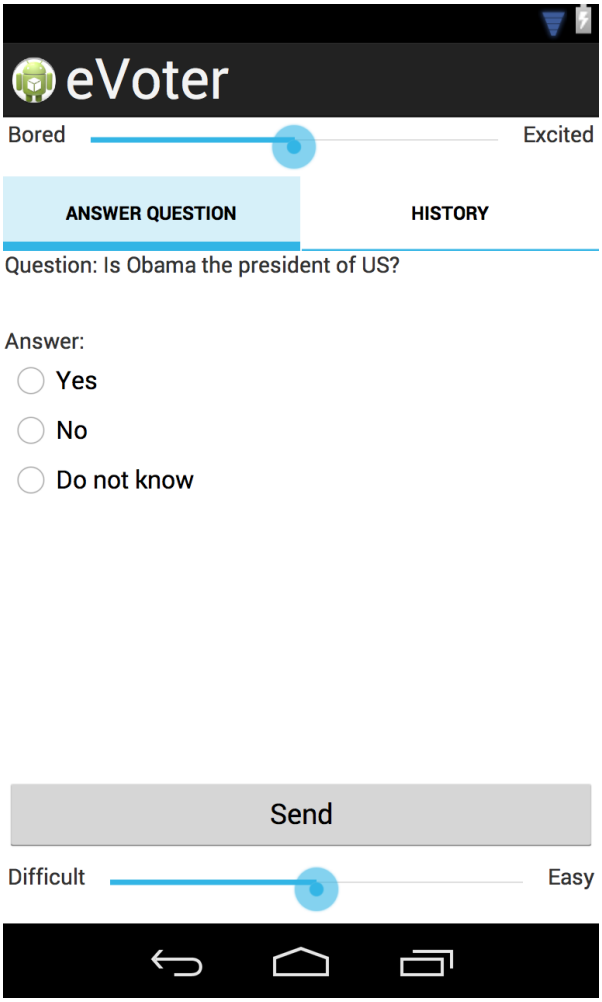
4. USER INTERFACE

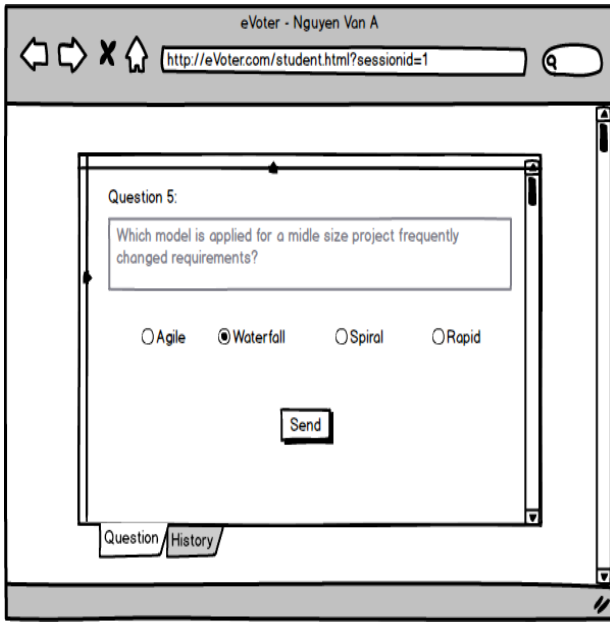
4.1 User management interface



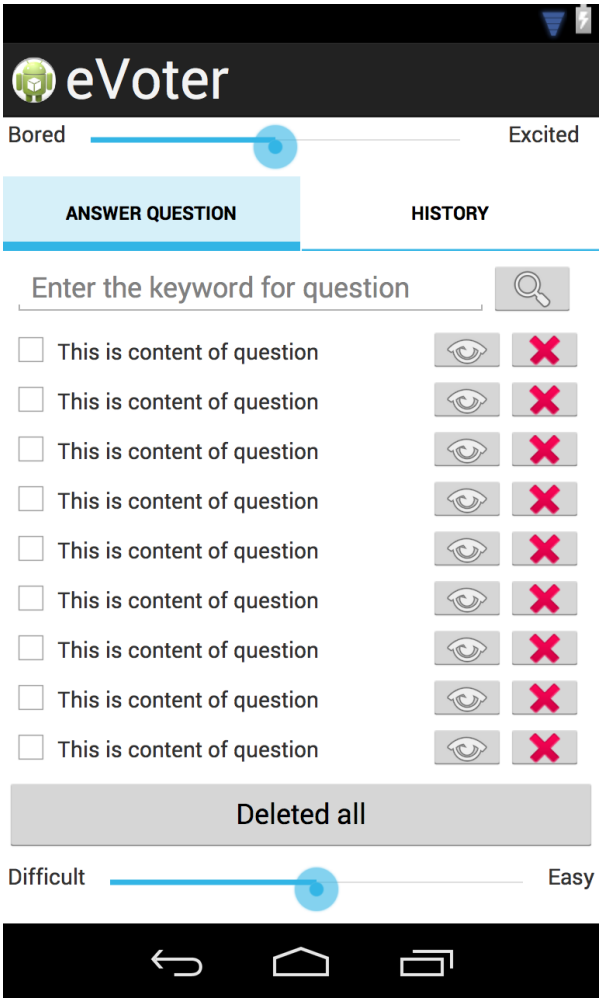
4.2 Student management interface

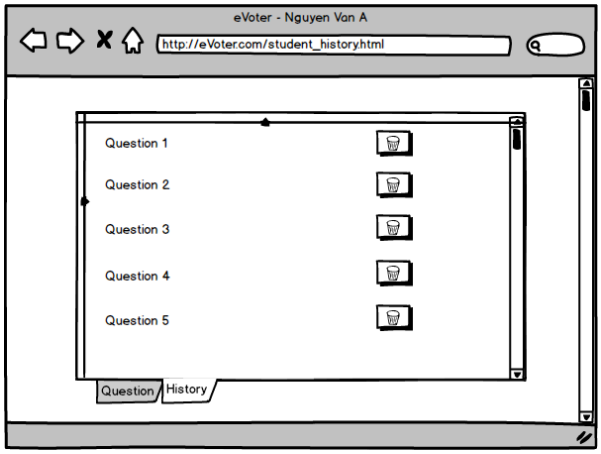
4.2.1 Answer a question



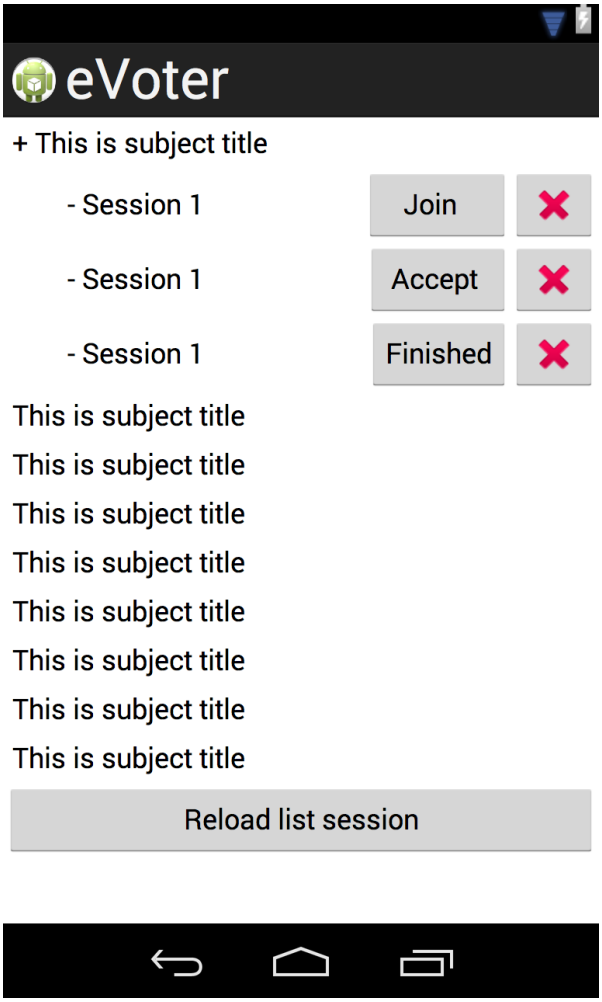


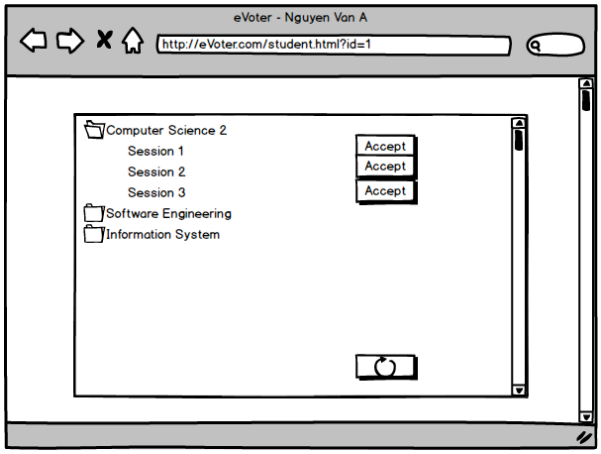
4.2.2 View question's history



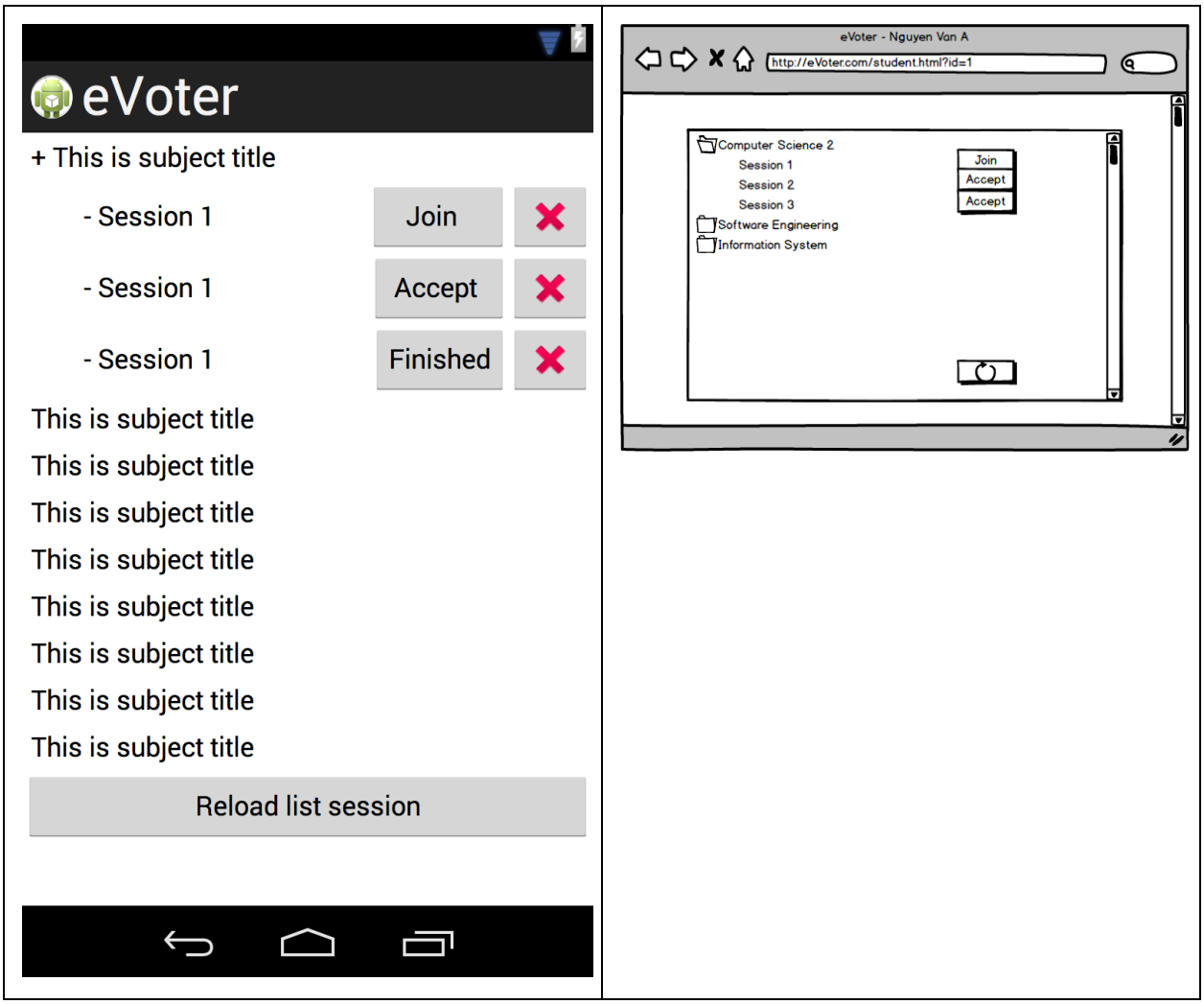


4.2.3 View a session

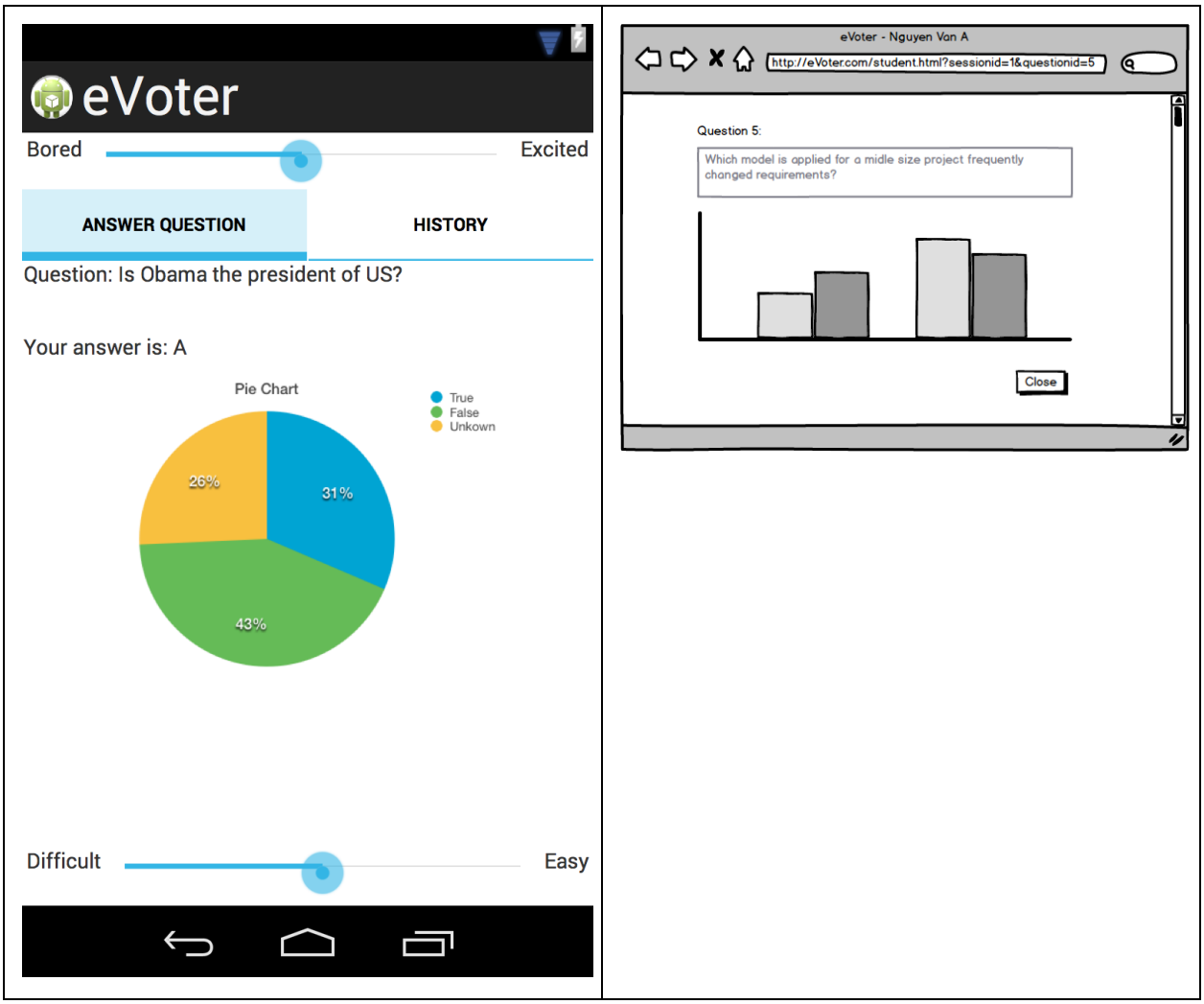




4.2.4 Join a session

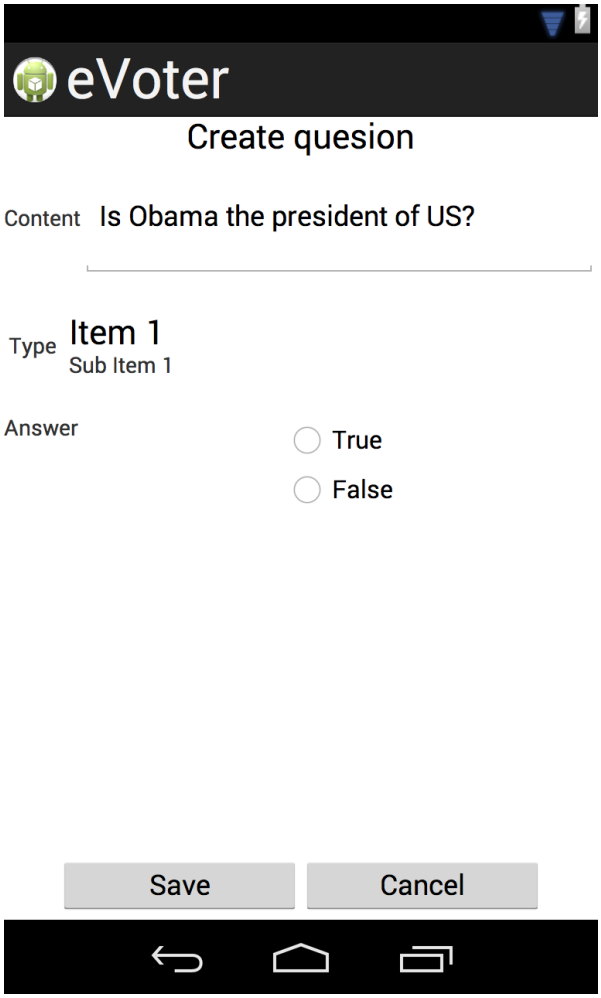


4.2.5 View answer result in real-time

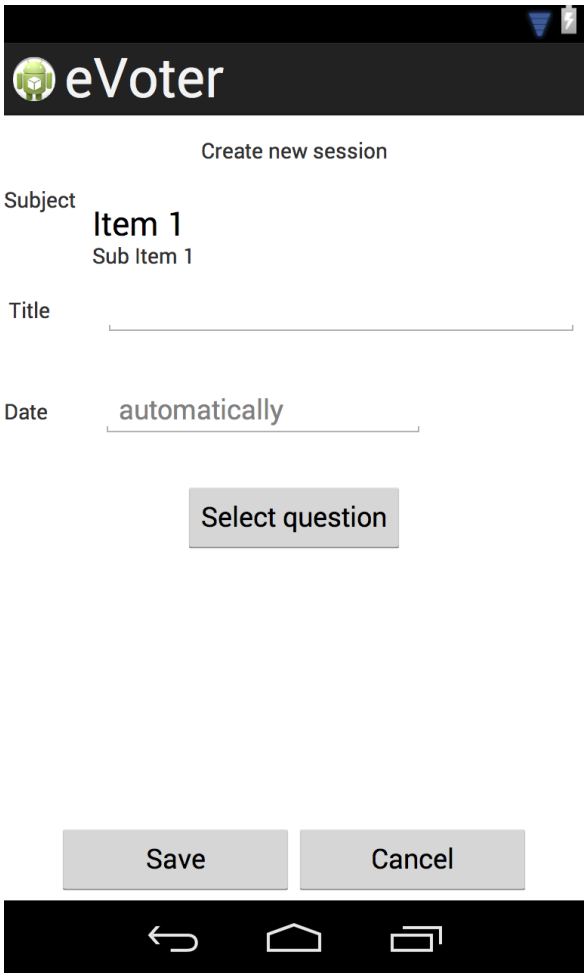


4.3 Teacher management interface

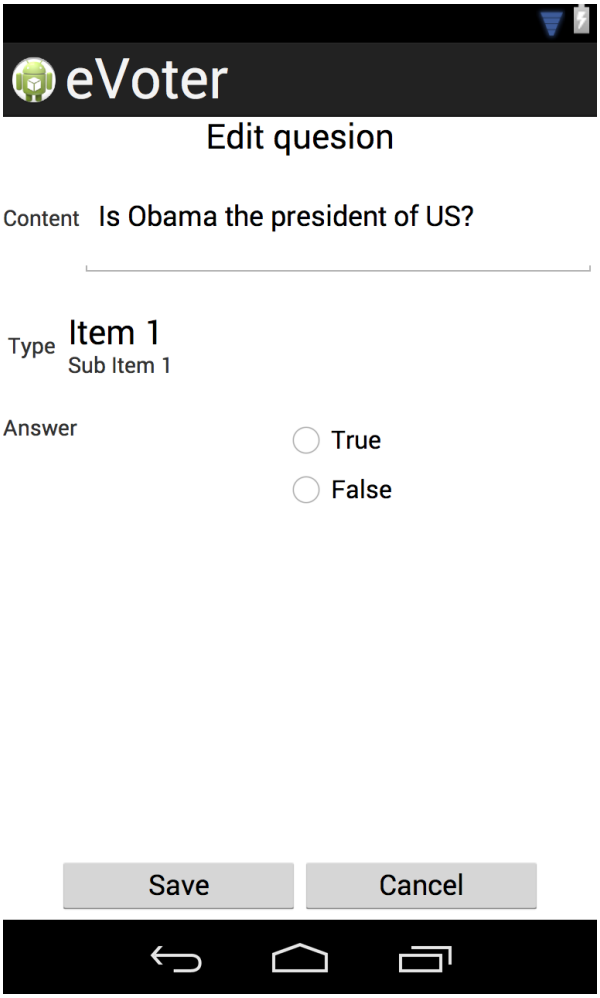
4.3.1 Create a new question



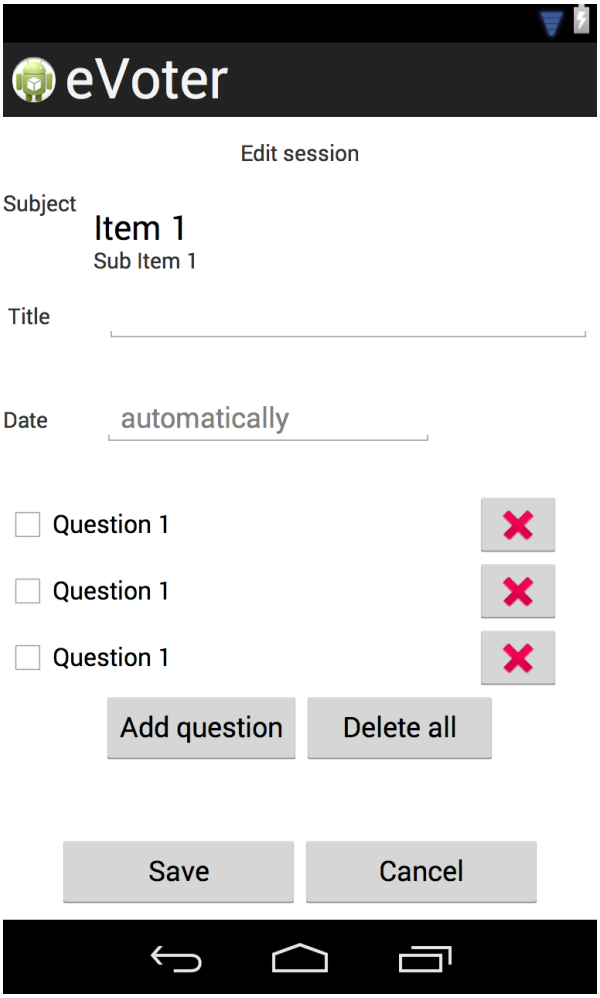
4.3.2 Create a new session



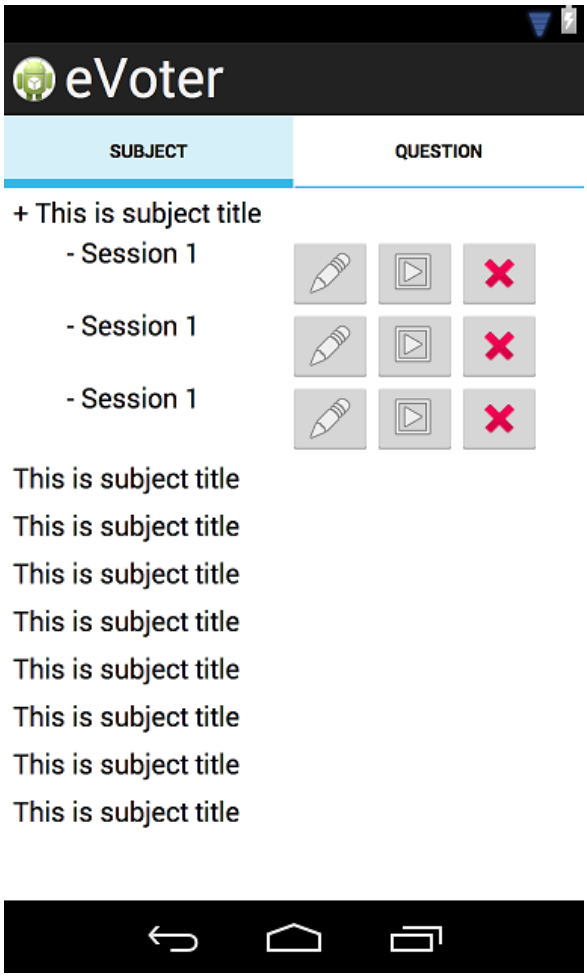
4.3.3 Edit a question



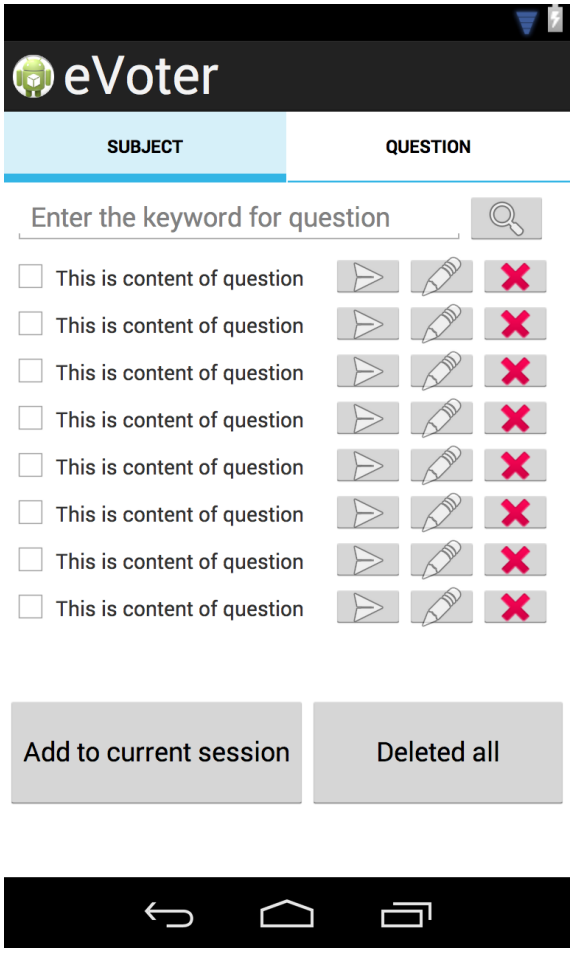
4.3.4 Edit a session



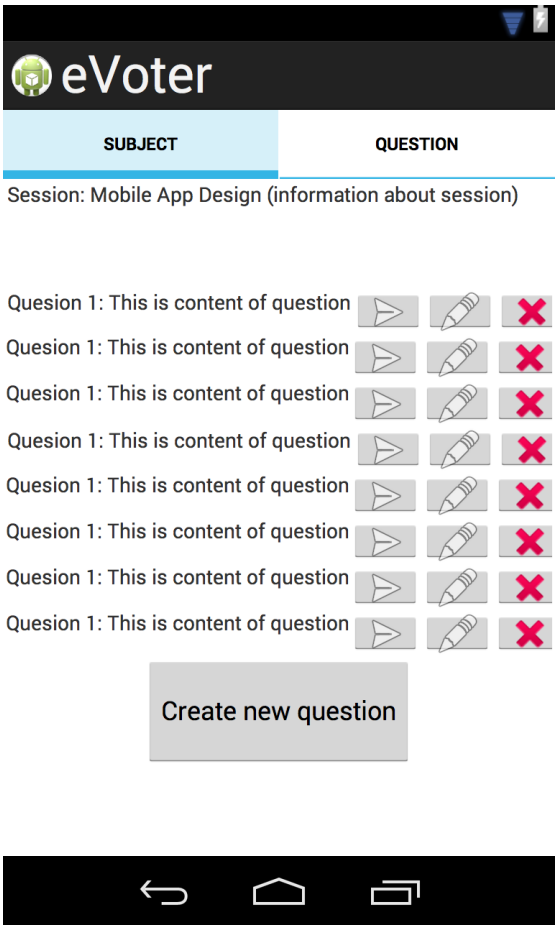
4.3.5 View subjects



4.3.6 Search a question

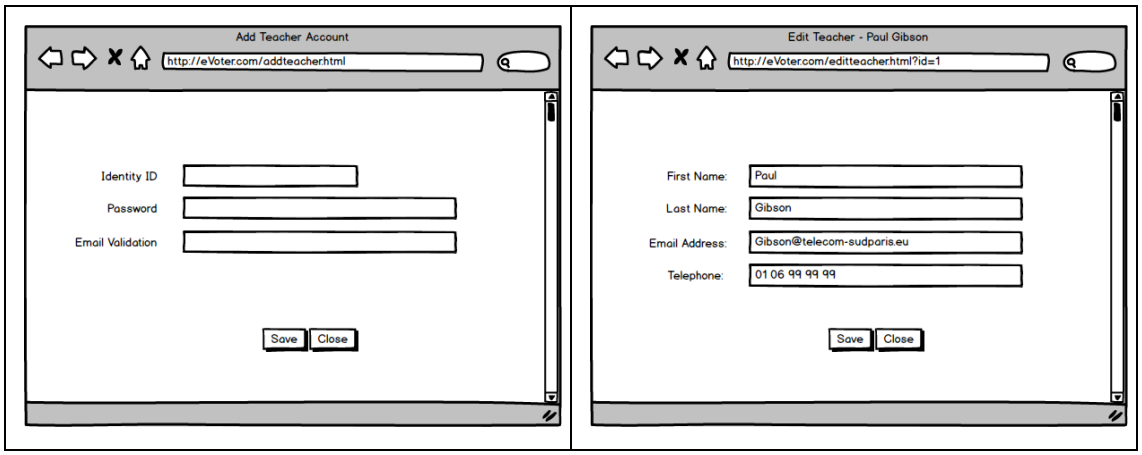


4.3.7 View an active session

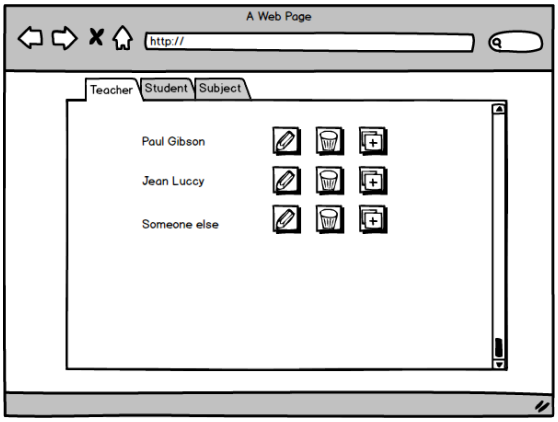


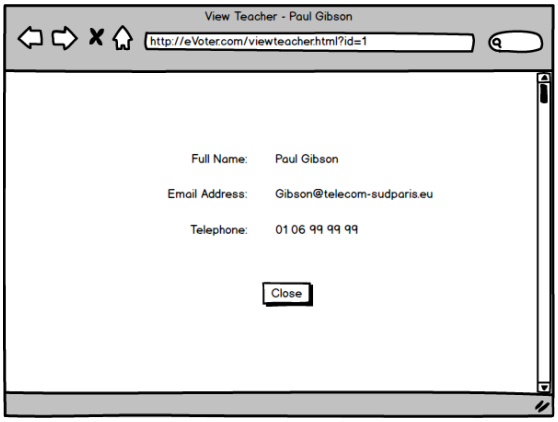
4.4 System Management Interface

4.4.1 Teacher Management



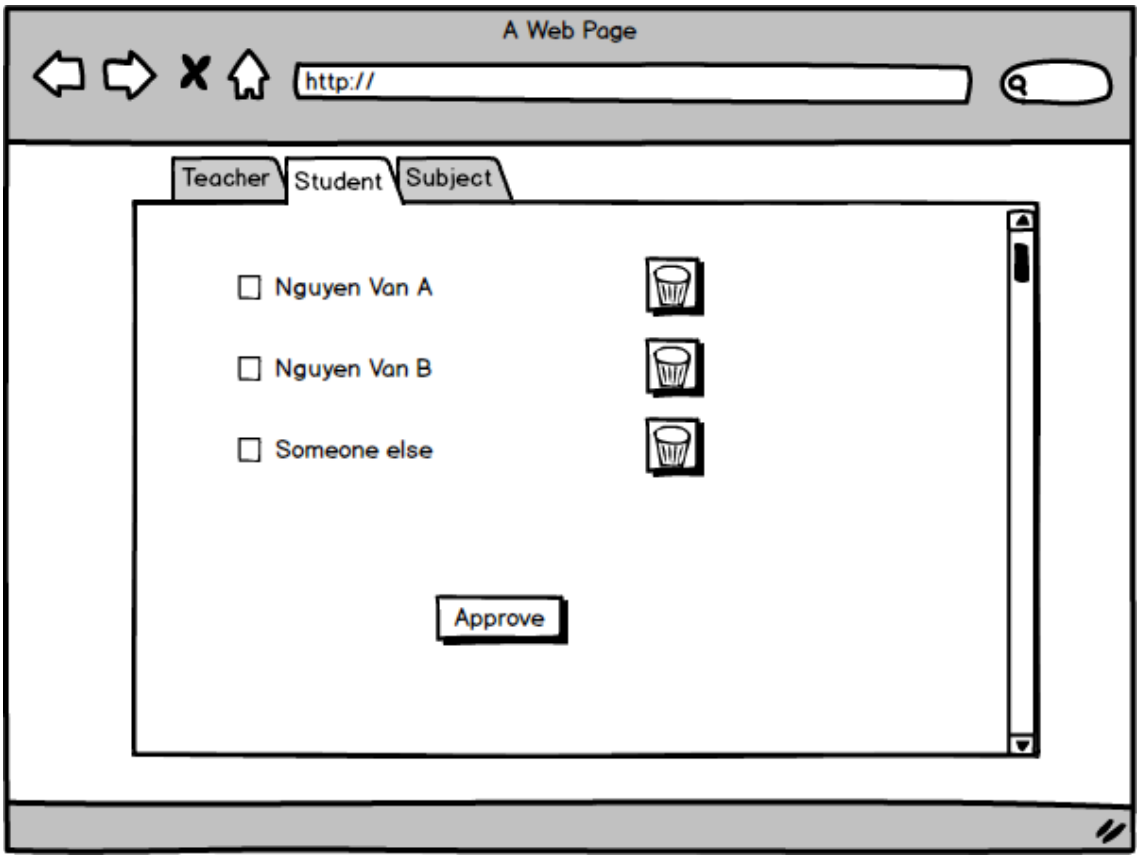
Create teacher account	Edit teacher information





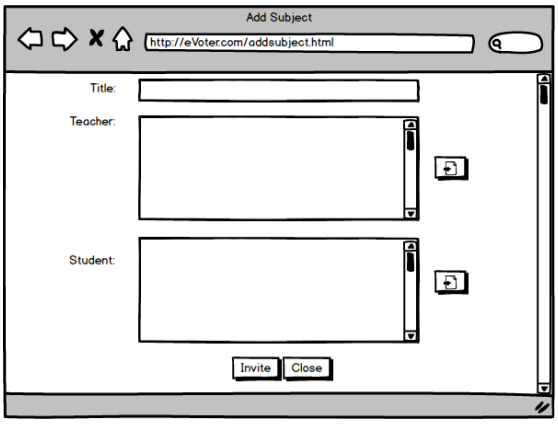
View teacher list	View teacher information in detail

4.4.2 Student Management

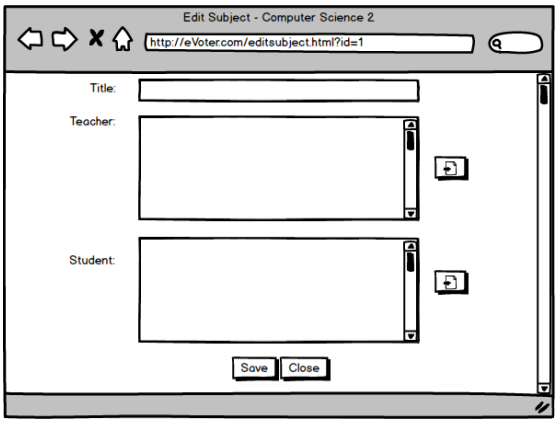


Approve student registration request

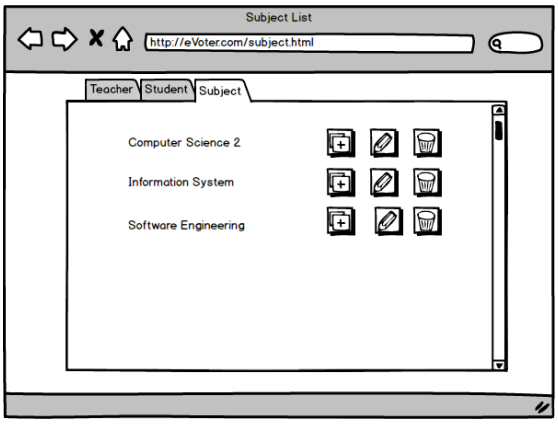
4.4.3 Subject Management



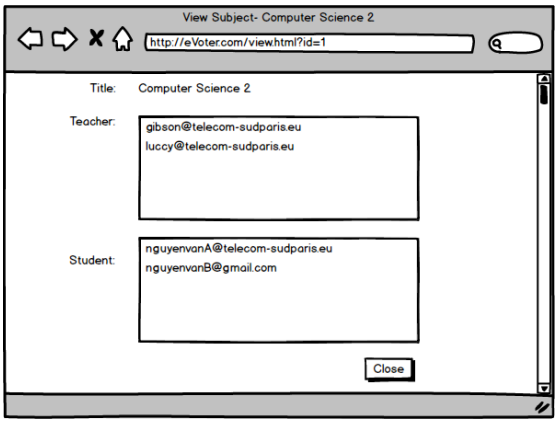
Add a subject



Edit a subject



View subject list



View subject in detail