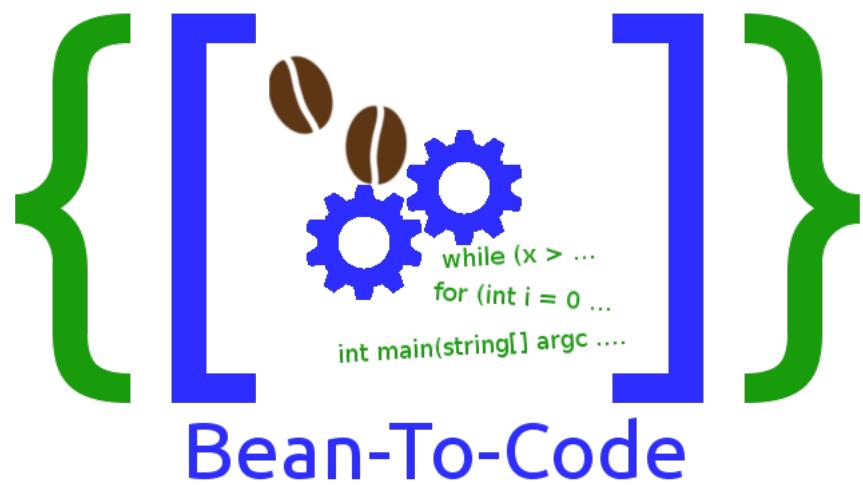


COS 301 Software Documentation

Melany Barnes 12030466
Dieter Doman 11002566
Johan Esterhuyse 10043285
Rudiger Roach 11004322

Version 1.0
GitHub link: https://github.com/RudigerRoach/301_main_emma.git



Contents

1 Vision and Scope	3
1.1 Vision	3
1.2 Scope	3
2 Architecture requirements	5
2.1 Architecture requirements	5
2.1.1 Architectural scope	5
2.1.2 Quality requirements	5
2.1.3 Integration and access channel requirements	5
2.1.4 Architectural constraints	6
2.2 Use of reference architectures and frameworks	6
2.3 Technologies and languages	6
3 Functional requirements and application design	7
3.1 Introduction	7
3.2 Required Functionality	7
3.2.1 Login and Auto Login	7
3.2.2 Create Judging Session	7
3.2.3 Voting	8
3.3 Use case prioritization	9
3.4 Use case/Services contracts	9
3.5 Process specifications	11
3.6 Domain objects	13
4 Testing Information	13
4.1 Unit testing	13
4.2 Usability testing	14
4.3 How tests where executed	15
4.4 Test coverage	15
5 Project management	15
5.1 Software development process	15
5.2 Issue management	15
5.3 Team profile	16
5.4 Project progress	16
5.5 Un-implemented functionality	17
5.6 Main risks and challenges faced	17
6 User manual for server	18
6.1 Creating a session	18
6.2 Controlling the session:	20
6.3 Importing project for further development	20
6.4 Installing the server	20
7 User manual for mobile application	20
8 Glossary	23

1 Vision and Scope

1.1 Vision

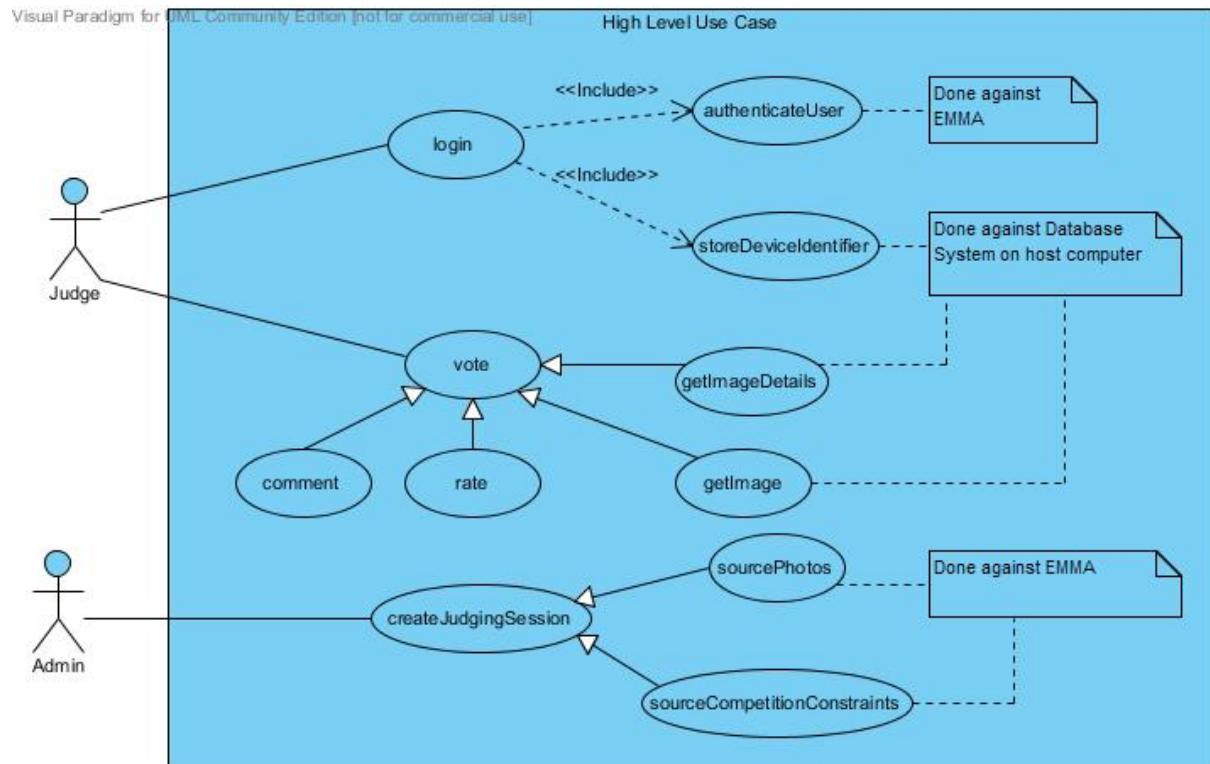
Our client creates software for camera club event management. A big part of an event comprises of an image judging process. Currently the process is completed by using Infra-red remotes and receivers, but this configuration is limited in terms of usability and the amount of judges that can judge concurrently.

The proposed solution will replace the hardware remote with a software application to run on a mobile device. The mobile application should alleviate all of the issues caused by the current setup, and should be developed with a server component that plugs into the existing EMMA system.

1.2 Scope

Create a software solution that:

- Runs on IOS and Android mobile devices.
- Allows as many as 20+ judges on the night.
- Allows judges to register against the event (in order to score) by capturing an email address.
- Remembers the scoring device for future meetings such that registration is not required again.
- Caters for realtime scoring.
- Can display a thumbnail image of that currently being judged.
- Caters for simple score entry bound within a variable range, as well as pluggable scoring methods that could include boolean scoring.
- Reports meaningful error messages, in a clear way.
- Allows for quick correction and re-capture.
- Can notify a judge of outstanding scores.



High Level Use Case Diagram

2 Architecture requirements

2.1 Architecture requirements

2.1.1 Architectural scope

- Provide an infrastructure for a judge to rate photos on a mobile device.
- Provide a database to link a judge's phone id to his email address.

2.1.2 Quality requirements

- Usability

99 % of users should be able to use the system with little to no prior training.

- Scalability

The deployed system must be able to operate effectively under the load of 50 concurrent users. The 50 users will be handled by Jetty which creates a thread for each person that connects to certain servlet at a point in time thus it will be able to handle the concurrency. It will be tested by creating a thread pool of 50 threads and doing 50 unit tests concurrently.

- Installability

It should be easy to install the server side component and the effort to get it running each club night should be minimal. There will be a computer at the event running the EMMA server component and our server should be installed on it. The application should also be easy to download and install on the judges phone.

- Performance requirements

All operations on application should respond within less than 1 second.

- Testability

All services offered must be accompanied by unit tests. The tests should ensure that all pre-conditions are met before the service is delivered and that all post-conditions are met after the service has been delivered.

- Security

The systems functionality should be only available to users who can be authenticated through the EMMA system. The users email address will only be used with no password. New users have to create an account before being granted access to the application if the sessions is closed. If the session is opened any person should be allowed to use it.

2.1.3 Integration and access channel requirements

- Integration requirement

The production version of this application will need to integrate with EMMA. EMMA is Java a based application.

- Access channels

The mobile application will have to go through a web-service which will be the public interface for the server-side component. It uses HTTP communication between then mobile and the server Jetty component.

2.1.4 Architectural constraints

- The mobile application should run on Android and iOS operating systems.
- The PC's that will be running the server side of the application and EMMA component will generally not be the latest technology(limited memory and processing power).
- There will be limited to no internet connection.
- The communication between the mobile device and server PC will be done over a wifi network.
- The server side component of this project should be able to run on Windows and OS X operating systems.

2.2 Use of reference architectures and frameworks

- JIRA Framework for the SCRUM agile method.
- Appcelerator Titanium framework which is an open-source software development kit for cross-platform mobile development.
- Jetty for hosting server and servlet handling.
- TiJasmine framework to run javascript unit testing in the Titanium framework.
- JUnit for java unit testing framework.

2.3 Technologies and languages

- Java
- JavaScript
- db4o Object database
- JavaFX
- FXML

3 Functional requirements and application design

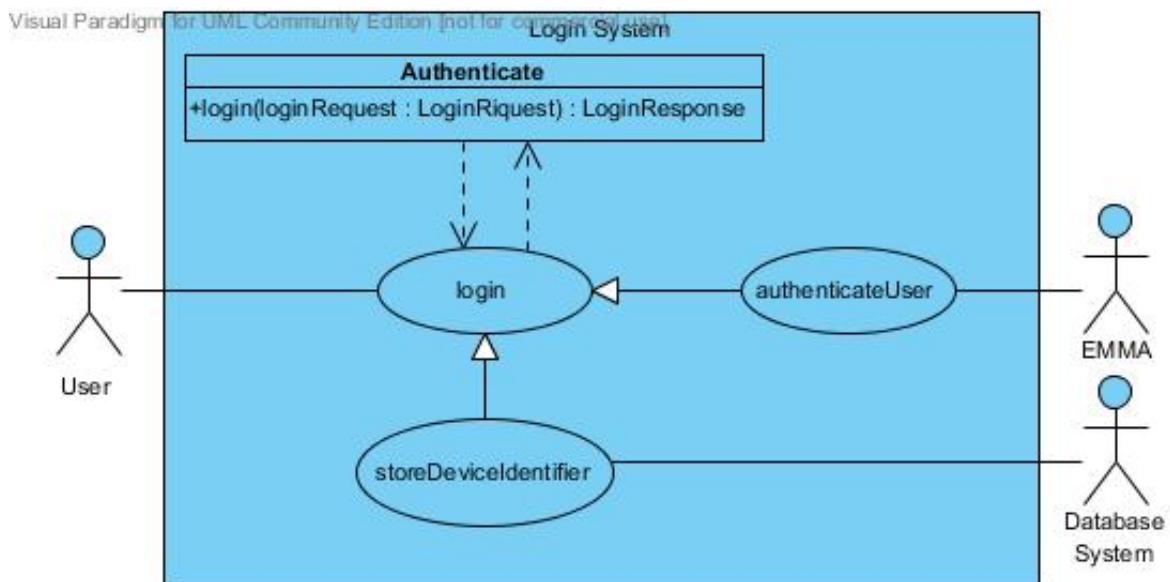
3.1 Introduction

This section discusses the functional requirements for the mobile judging system.

3.2 Required Functionality

3.2.1 Login and Auto Login

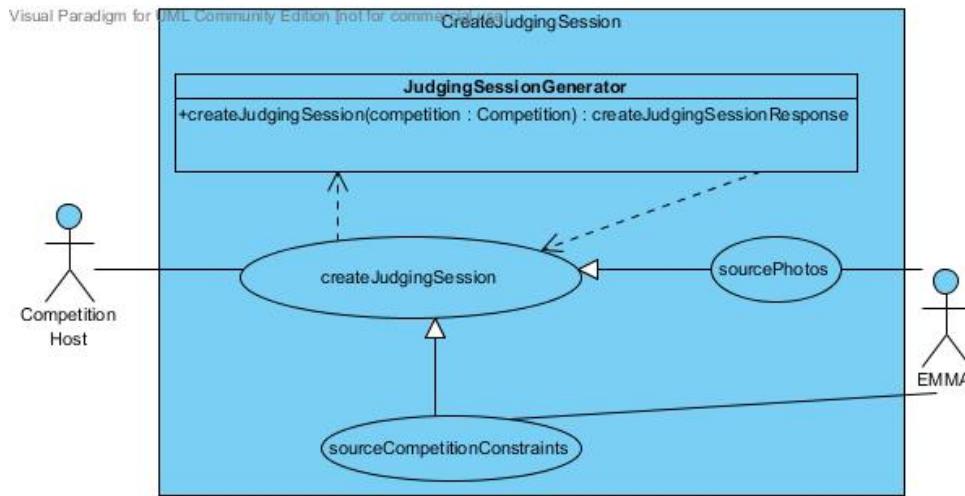
To login for the first time a user will have to enter his email address. The email provided will be authenticated by EMMA. If login fails the user will be informed that he is not registered to be a judge for the current session. If login is successful the device's unique identifier will be sent to the server to be stored in the database so that the device can be remembered on the system. This will allow for auto login - if a user attends a session where he is able to judge his phone will automatically be logged into the system when he enters the application. The user will then be able to use the rest of the system.



Login Use Case Diagram

3.2.2 Create Judging Session

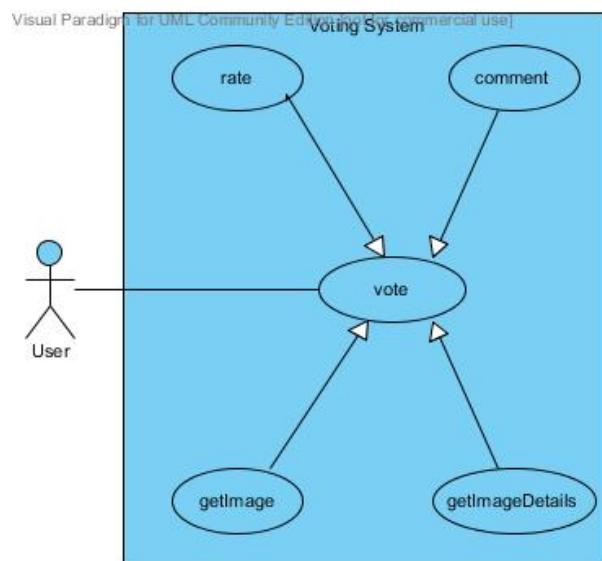
When the server is started, it will request that the session's photos as well as all the competition constraints be sent to it. The constraints will contain the type of session (Open event, Closed event, Yes/No, Winner), the range for a valid score and if comments are enabled.



Create Judging Session Use Case Diagram

3.2.3 Voting

If a user logs in before the event starts, a loading screen will be displayed until the event starts. The server will inform the user's application when the event starts. The server will then pass through the first image and the details about the image. The details will contain the image name, the bottom and top score ranges as well as if comments must be enabled. The users will vote for the image and will be able to leave a comment if the comments are enabled. The server will either have a time limit per image or the server will check if all users have submitted their vote. If not all users have submitted their vote, the server will notify those users. If all users have submitted their vote, the next image and its details will be displayed. This will continue until all images have been scored. The user will be notified that voting is done.



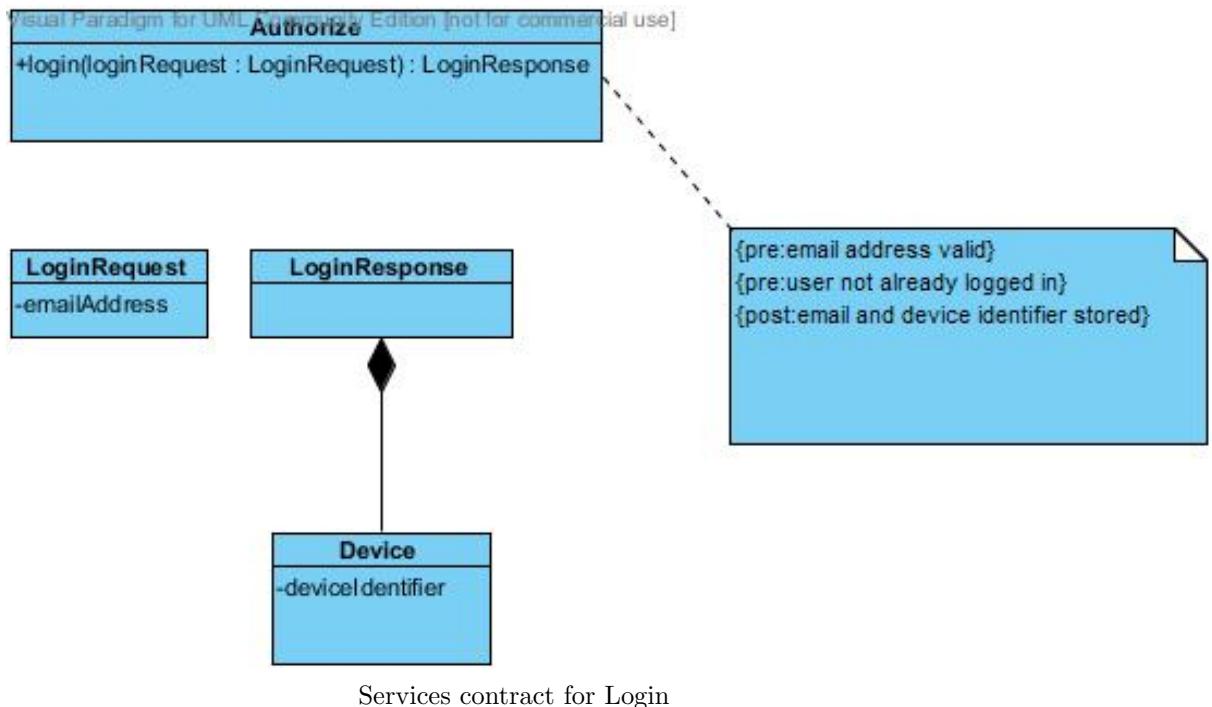
Voting Use Case Diagram

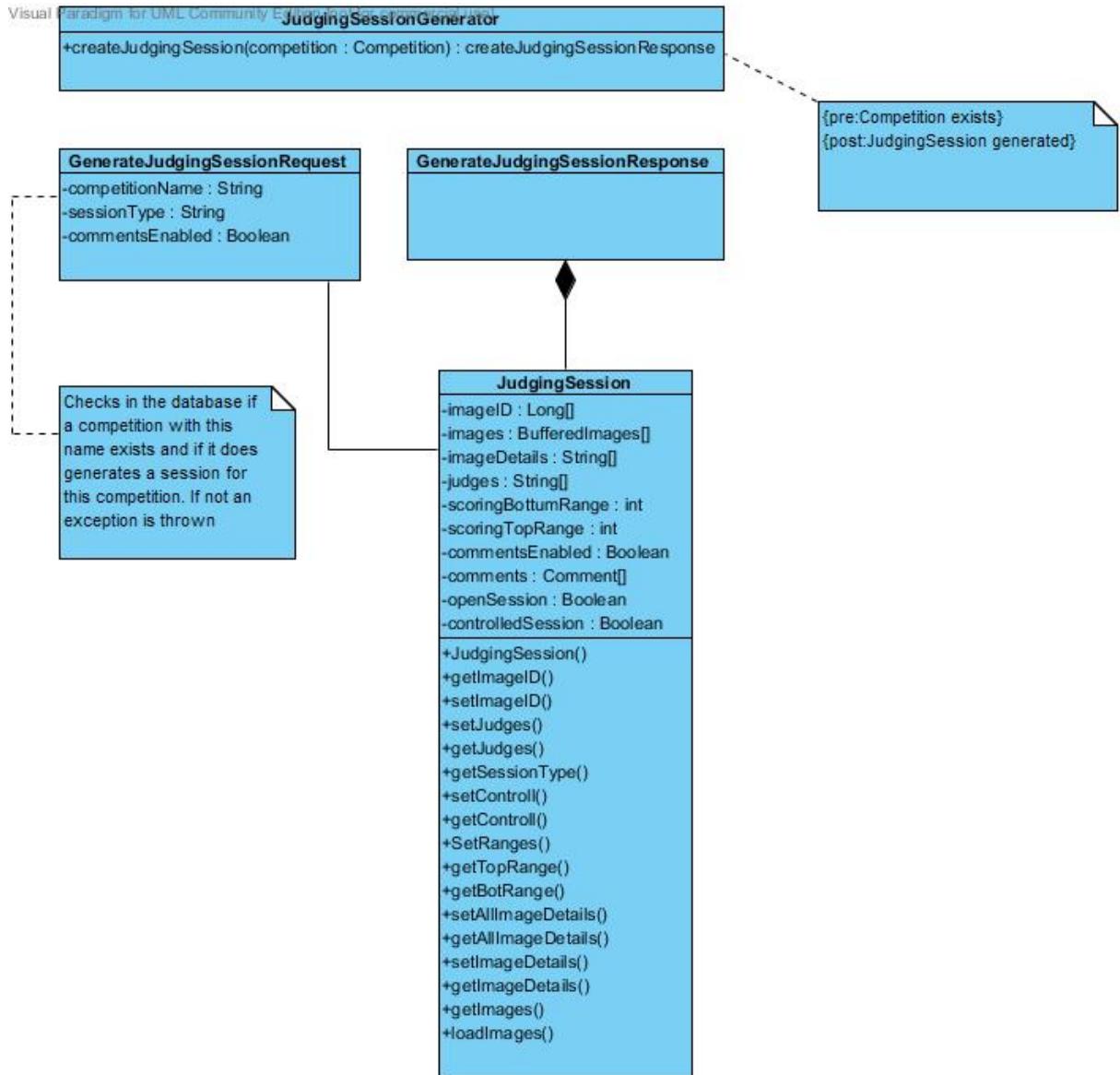
3.3 Use case prioritization

Critical Use Cases are the main cases that the system is made up of namely: Login, Create Judging Session and Voting. Without these cases the system will have limited to no functionality which will lead to a system that is not required by anyone.

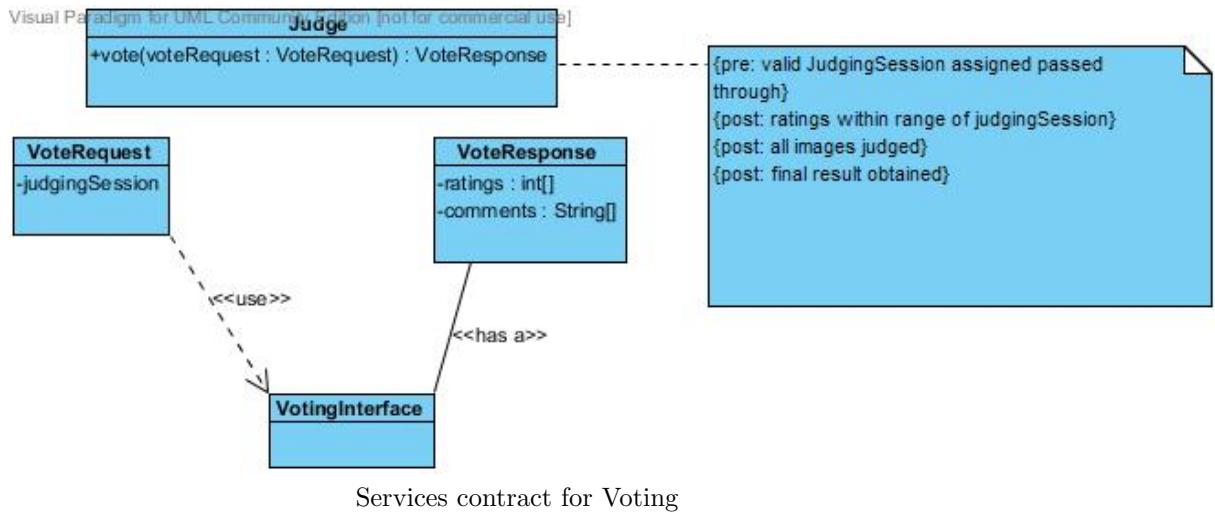
Important Use Cases are the cases that improves the critical use cases and introduces a wider variety of functionality. These cases are Auto Login.

3.4 Use case/Services contracts

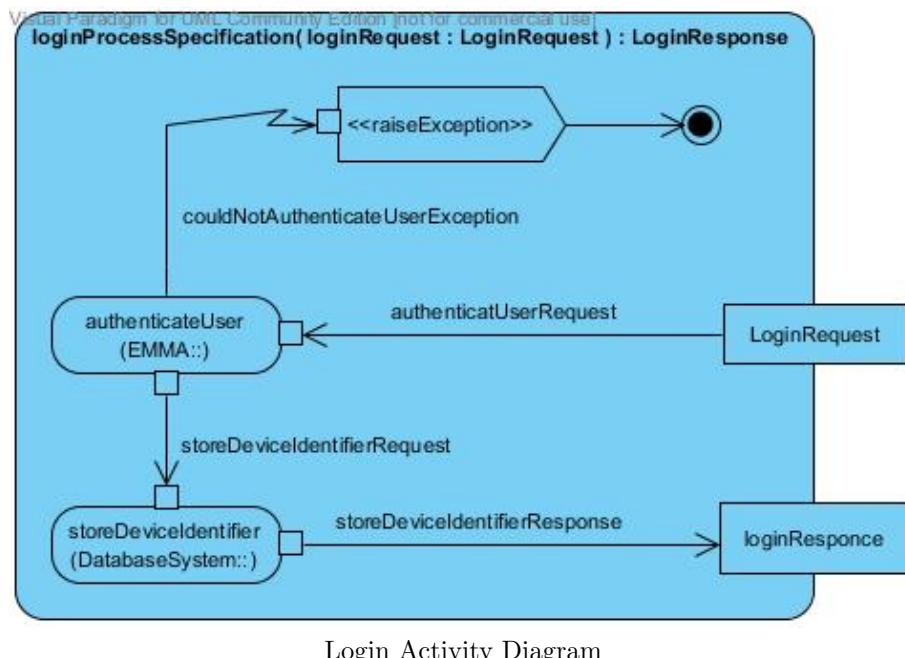


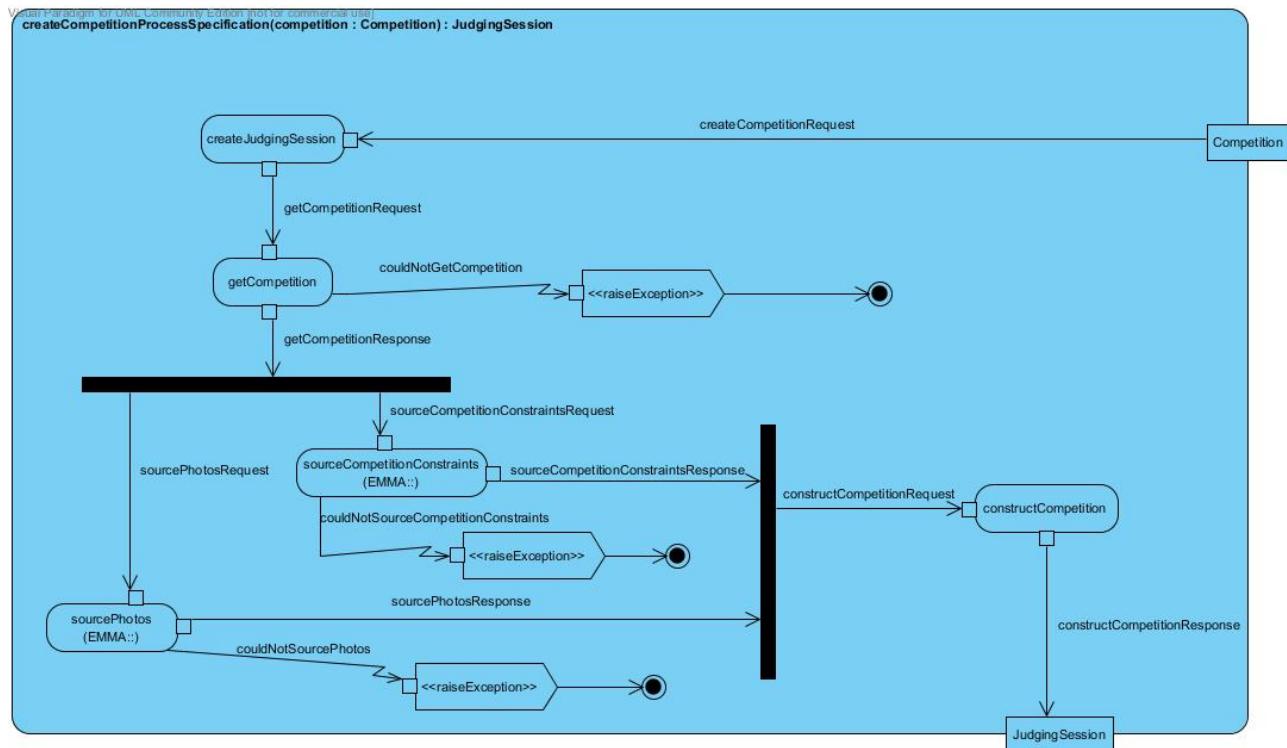


Services contract for Create Judging Session

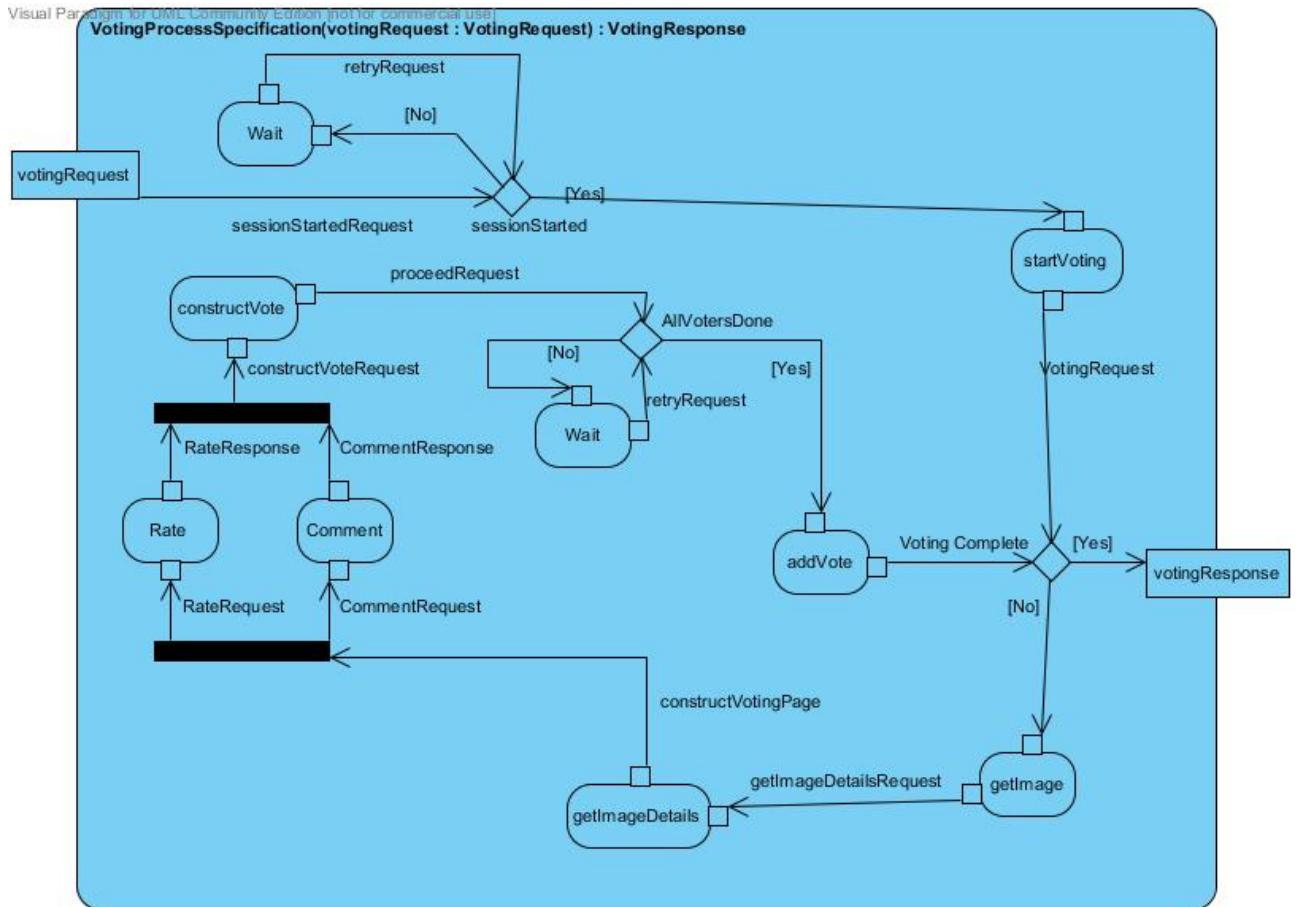


3.5 Process specifications





Create Judging Session Activity Diagram



Voting Activity Diagram

3.6 Domain objects

4 Testing Information

4.1 Unit testing

For unit testing we used the Jasmine framework for the mobile side of our system which is a javascript unit testing framework. We used an mock java server for mobile unit testing and mock data to test specific functions in the code.

For the server side component of our system we used a JUnit testing framework. Using HTTP default clients to perform mock network calls for the tests and mock data to test certain components of the server.

Mock data was set up to test the pre- and post conditions of the software contract.

4.2 Usability testing

Our mobile side was developed focusing on user centered development. Throughout the design and implementing of the mobile side users were consulted in short feedback cycles, getting their inputs and new ideas on how to make the application as usable as possible.

During our test we had a user group of 6 random people. We ran the 3 different types of sessions with them. We gave them some tasks to do such as rotating the screen, change the language, and to evaluate the flow of voting.

Consent Form

Subject – COS301
Group – Bean-To-Code
Department of Computer Science
University of Pretoria

This study will test only the usability of our product and not the skills of the user. **Thus it is the product that is being tested not the user.**

All information obtained in this study will be kept strictly **confidential and anonymous**. The results of this study will be presented as a group and no individual will be identified without their explicit permission.

Your participation is completely voluntary. You may withdraw from this study at any time, without penalty.

By signing this consent form, you are indicating that you fully understand the above information and agree to participate in this study.

Participant's signature _____

Date _____

Bean-To-Code member signature _____

Date _____

Task that needed to be performed

1. Change the server address in the settings tab.
2. Change the comment setting in the settings tab.
3. Login into the session.
4. Submit one score using a textbox.
5. Submit one score using a sidebar.
6. Finish the session.
7. For an elimination session, push yes then no then submit.
8. Scroll between photos to select a winner.
9. Add a comment and submit your winner.

Data recorded from usability test

TASK										Total	% Average overall tasks
	1	2	3	4	5	6	7	8	9		
One (Easy)	4	4	6	3	6	6	3	5	5	40	74%
Two		2		1			2		1	6	11%
Three	2			1			1	1		5	9%
Four				1						1	2%
Five (Hard)										0	0%

4.3 How tests were executed

Junit netbeans plugin for the server side. TiJasmine modules and manual mock-server execution for the mobile side.

4.4 Test coverage

- Approximately 60 percent test coverage on the mobile side.
- Approximately 70 percent test coverage on the server side.

5 Project management

5.1 Software development process

We followed the scrum software development methodology. Its agility, in terms of requirements, meant that the project was able to grow during development, while we were discovering new requirements from the client.

5.2 Issue management

On the clients suggestion, we used the Jira issue tracking system, combined with git and GitHub for version control.

5.3 Team profile

Melany Barnes:

- Team Lead.
- Developing the bulk of the mobile app front-end.

Dieter Doman:

- Developing the back-end of the server app.

Johan Esterhuyse:

- Developing the front end for the server app.
- Paying special attention to documentation and diagrams.

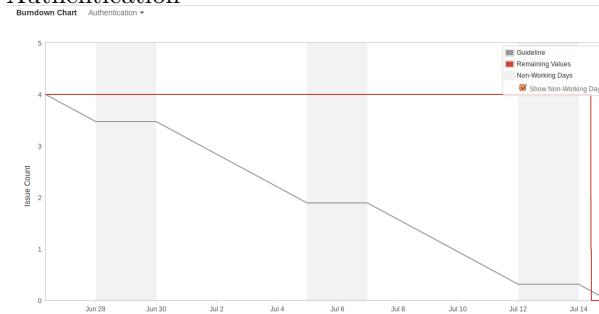
Rudiger Roach:

- Developing the back-end of the mobile app.
- Developing the database for the server app.
- Assisted in development of the mobile app front-end.

5.4 Project progress

We aimed to provide burndown charts for the entire project progress but unfortunately The issue tracking system ran from a server at the clients' residence and it experienced a lot of down-time. Luckily we can provide burndown charts for the following sprints:

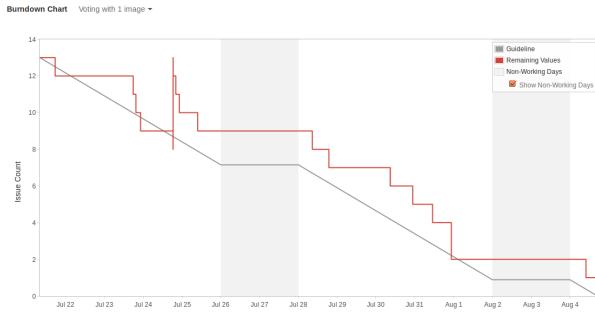
- Authentication



- Auto login



- Voting with 1 image



5.5 Un-implemented functionality

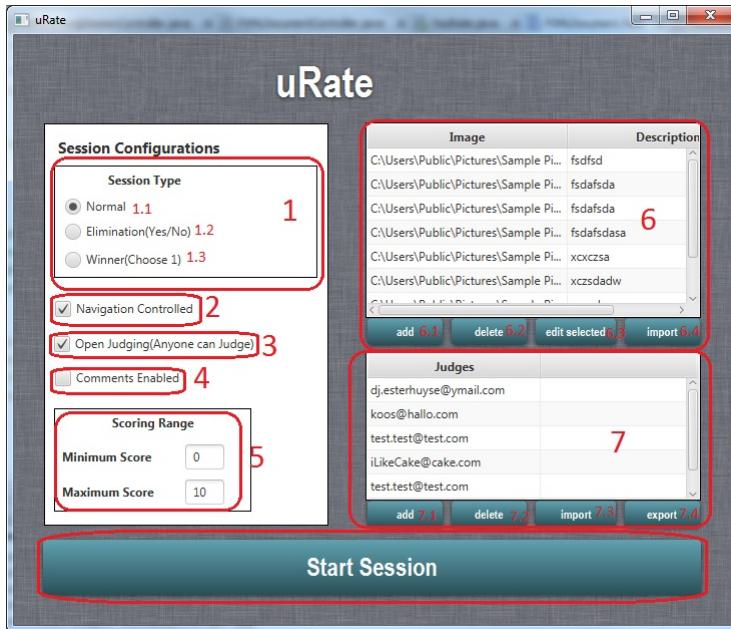
- Displaying results after voting has completed after.
- The mobile app is unaware of a server crash and will wait indefinitely in this case.

5.6 Main risks and challenges faced

- Getting to know Jira and issue tracking.
- Getting to know the Appcelerator platform.
- Getting to know JavaFX using FXML.
- Multi-platform support.
- Multi-screen-resolution support.

6 User manual for server

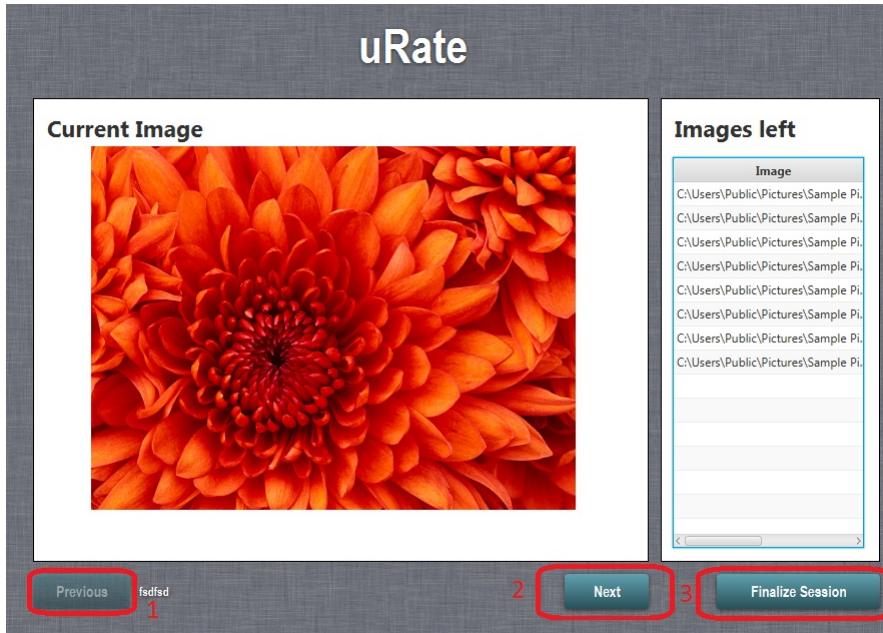
6.1 Creating a session



1. Specify the Session type:
 - (a) Normal is a session where the user rates each picture based on general impression.
 - (b) Yes/No is a session where the user states whether they think a picture should go to the next round or not.
 - (c) Winner is a session where the user chooses one winner and the picture with the most votes wins.
2. Specify if the session is a controlled session or not.
 - (a) If the creator of the session would like to guide the judges through voting then tick the box.
 - (b) If the creator of the session would like judges to judge on their own time then don't tick the box.
3. Specify if the session should be open to all judges.
 - (a) If anyone can judge then tick the box.
 - (b) If only specific people may judge then don't tick the box.
4. Specify whether comments should be enabled.
 - (a) If you want the judges to give their opinion then tick the box.
 - (b) If you don't want judges to give their opinion then don't tick the box.
5. Specify the minimum and maximum scores.

- (a) Minimum has to be smaller than maximum.
6. Add Images to the voting session.
 - (a) Click the "+" button to add a single image.
 - i. Click browse and select an image.
 - ii. Type in a description.
 - (b) Click the "++" button to import multiple images.
 - i. Click the browse next to the "select images" and select the images.
 - ii. Click the browse next to the "select description file" and select the description file.
 - A. The description file is a file which specifies the details for each image one line at a time.
 - (c) Click the gear button to edit a selected image.
 - i. First select the image.
 - ii. Click the gear.
 - iii. Specify the changes.
 - (d) Click the "-" button to delete a selected image.
 - i. First select the image.
 - ii. Click the "-" button to remove the selected image.
7. Add Judges to the voting session(only neccesary if open judging(see 3. of manual) is not enabled).
 - (a) Click the "add" button to add a judge.
 - i. Specify the judges email address and click add.
 - (b) Click the "delete" button to remove the selected judge.
 - i. First select the judge.
 - ii. Click the "-" button to remove the selected judge.
 - (c) Click the "import" button to import judges.
 - i. Click browse and select a judges file.
 - A. A judges file is a file that specifies the email addresses of judges one at a time.
 - (d) Click the "export" button to export images.
 - i. Browse to where you want to save the file and click save.
 8. Click "Start Session" in order to start running a session.
 - (a) If any data is missing, the creator will be informed.

6.2 Controlling the session:



1. Click the "Next" button in order to preview the next image.
 - (a) If this is a controlled session(See session creation point 2), this will take the judges(using app) to the next image.
2. Click the "Previous" button to go back to the previous image.
 - (a) This is not possible if the session is controlled(See session creation point 2).
3. When all judges are done voting click on "Finalize Session".
 - (a) This will end the judging and then display the results.

6.3 Importing project for further development

Since by the client's request the only wanted an netbeans project. Thus open project in netbeans after pulling from our git repository and import the all the external libraries under external libraries folder.

6.4 Installing the server

Run following command after building project in netbeans(which will create jar file in dist folder) in command line in the dist folder: "java -jar YouRate Simulator.jar"

7 User manual for mobile application

When the application opens up the app will first try to automatically log in (1.1). If the user has logged in using the device before, the server will recognise the device

and automatically log the user into the app. If automatic login fails, the user will be redirected to the login screen.

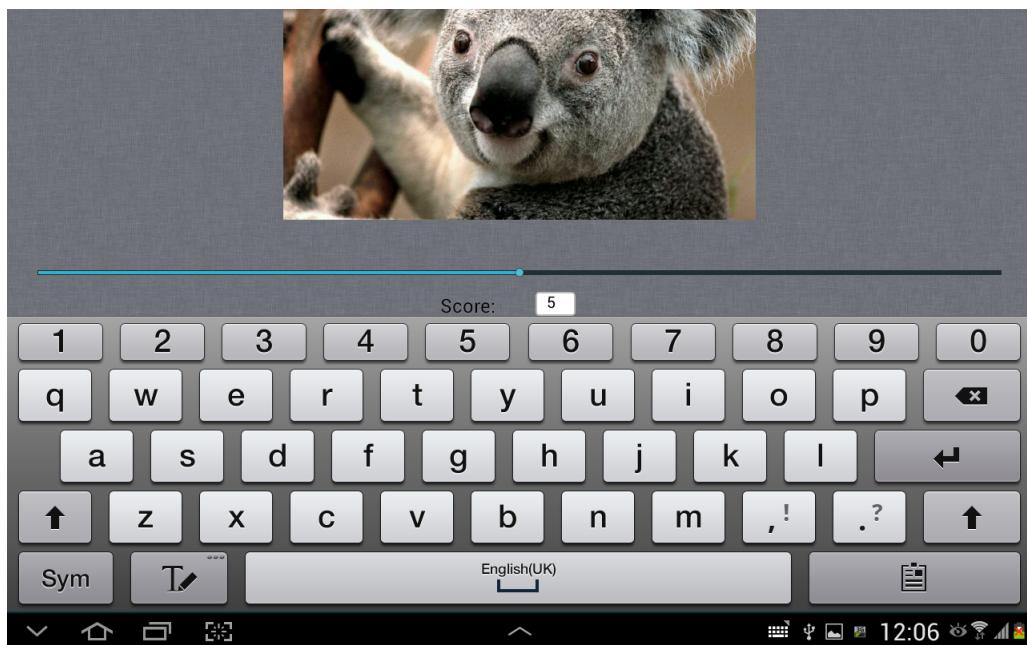
Waiting screens with different messages might be encountered:

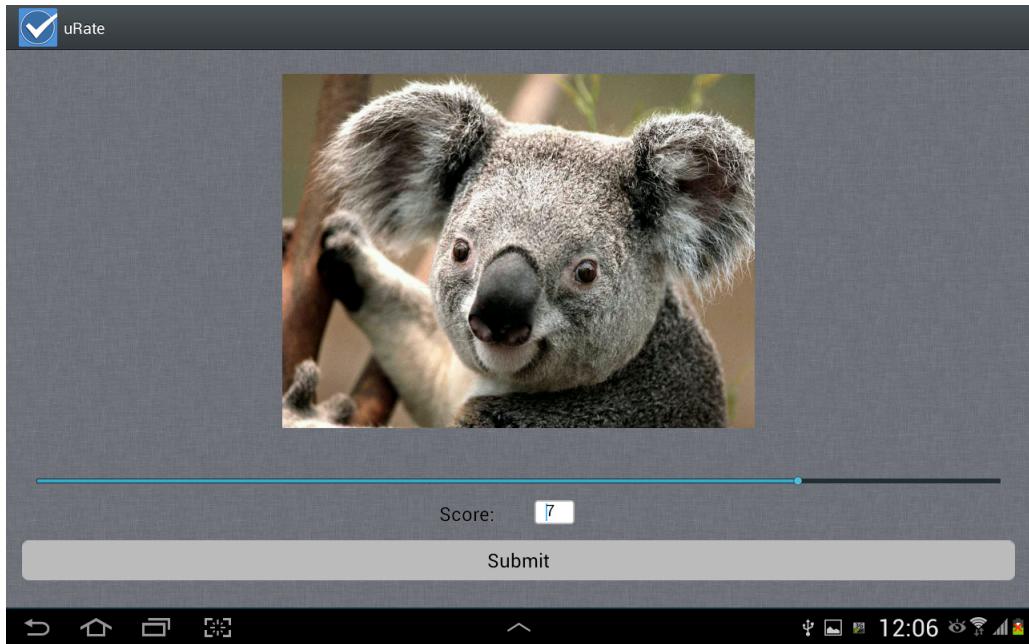
- Attempting automatic login... (See 1.1)
- Attempting to log in... (See 1.2)
- Waiting for server... (See 1.3)

The email address is validated, thus a valid email address must be entered. A message will be displayed if an invalid email address was entered and the login button was clicked. If the email address is valid the app will try to log in and a waiting screen (1.2) will be displayed. When the user is logged in successfully, waiting screen (1.3) will be shown. When an image on the server is ready to be displayed, the application will continue to the correct type of voting page (1.4, 1.5, 1.6)

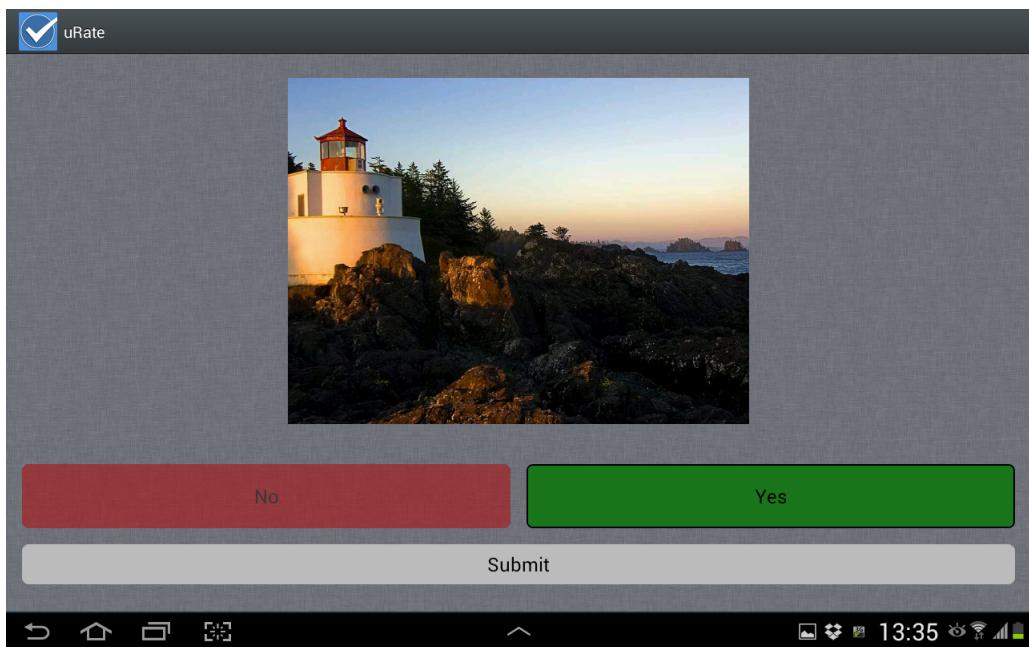
Number based scoring (1.4):

The score can be changed by moving the slider or by clicking on the score text area. When a user clicks on the text area, the device onscreen keyboard will pop up and the user can enter a score.



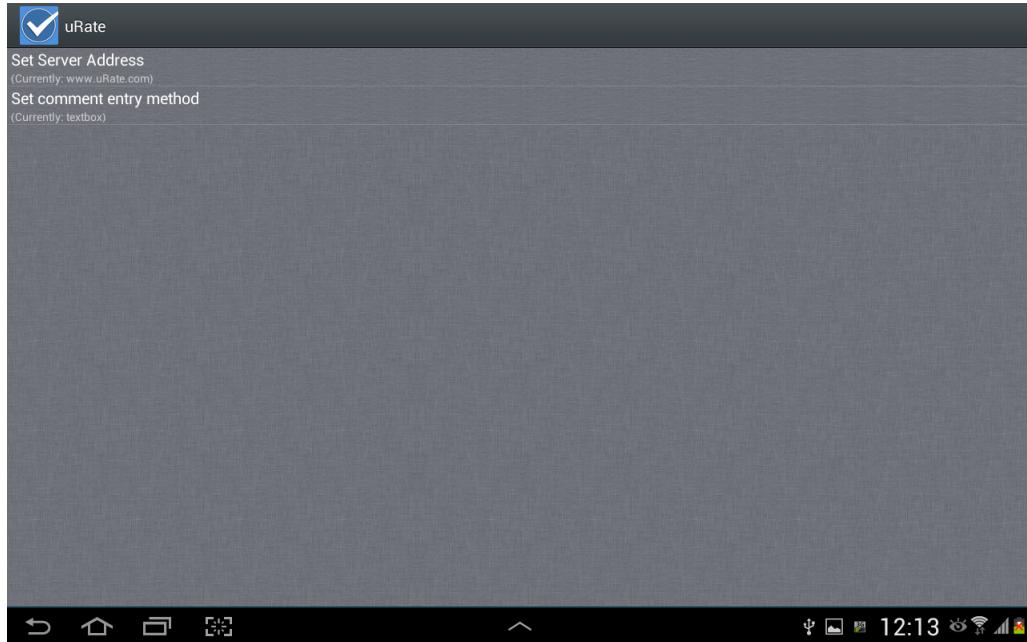


Elimination round (1.5): The user can choose between yes or no per image and then click on the submit button to submit their result.



Choose a winner (1.6): The user can swipe through the images and choose 1 as the winner and then click on the submit button to submit their result.

Changing the settings: The settings can be changed by clicking the device hardware menu button, if the device does not have a hardware menu button a settings icon will be displayed on the screen. The settings page will look as follows:



The server address can be changed by clicking on “Set Server Address”. To successfully connect to the server, the correct server address must be entered. Example address: 192.168.0.100 or www.uRate.com To change the comment entry method, the “Set comment entry method” can be clicked. Comments can be entered via clicking a button or by having a textbox always visible on the screen.

8 Glossary

EMMA - Entry and Member Management Application

His - Refers to his/her

He - Refers to he/she