inTV

CRDT DOCUMENT

Bitirsoft

Table of Contents

1. SYSTEM REQUIREMENTS

1.1. Definition

inTV is an application for android devices.

In this application we aim to develop an android based application which provides second-screen for the TV programs. This second screen shall provide real time relevant data about the selected program, such as guests of the talk show, actor of the movies or series. Our app also shall support second screen for presentations. In addition, our app shall provide interaction with the show. Interaction part of the app includes giving votes in a poll related to a show, taking part in online quiz and sharing comments.

1.2. System overview, tools and components

inTV is an application for android devices. It uses only internet connection of the phone. Our app runs on Android 5.0 or better version of the Android operating system. We also need server to be able to store information and to make communication. Our server is local now. However we will get online cloud server, soon. In addition to these our app has admin side. Admin side of the application is not available on the mobile app. It is only used on the computer via website which is designed for this purpose.

Moreover our application requires some components to be operational. These components are Android gui, admin gui, communication between database and application, information crawl, entity extraction, polling system and QR code system. To implement these components we also need some tools, of course. These tools are Android Studio, Jdbc, Microsoft Visual Studio, Microsoft Sql Management Studio, Sql, Java and C# as a programing language. Some of these components and tools are replaced or removed and some new components and tools are added. Instead of using Jdbc, we use Django for our database procedures. Also instead of using QR code, we have decided to use pin code because of the efficiency. Microsoft Sql Management Studio is removed because we do not need to use it. Finally we have decided to use Python as a programming language because of its functionality.

1.3. Use Case Diagrams

1.3.1. User Use Case Diagram

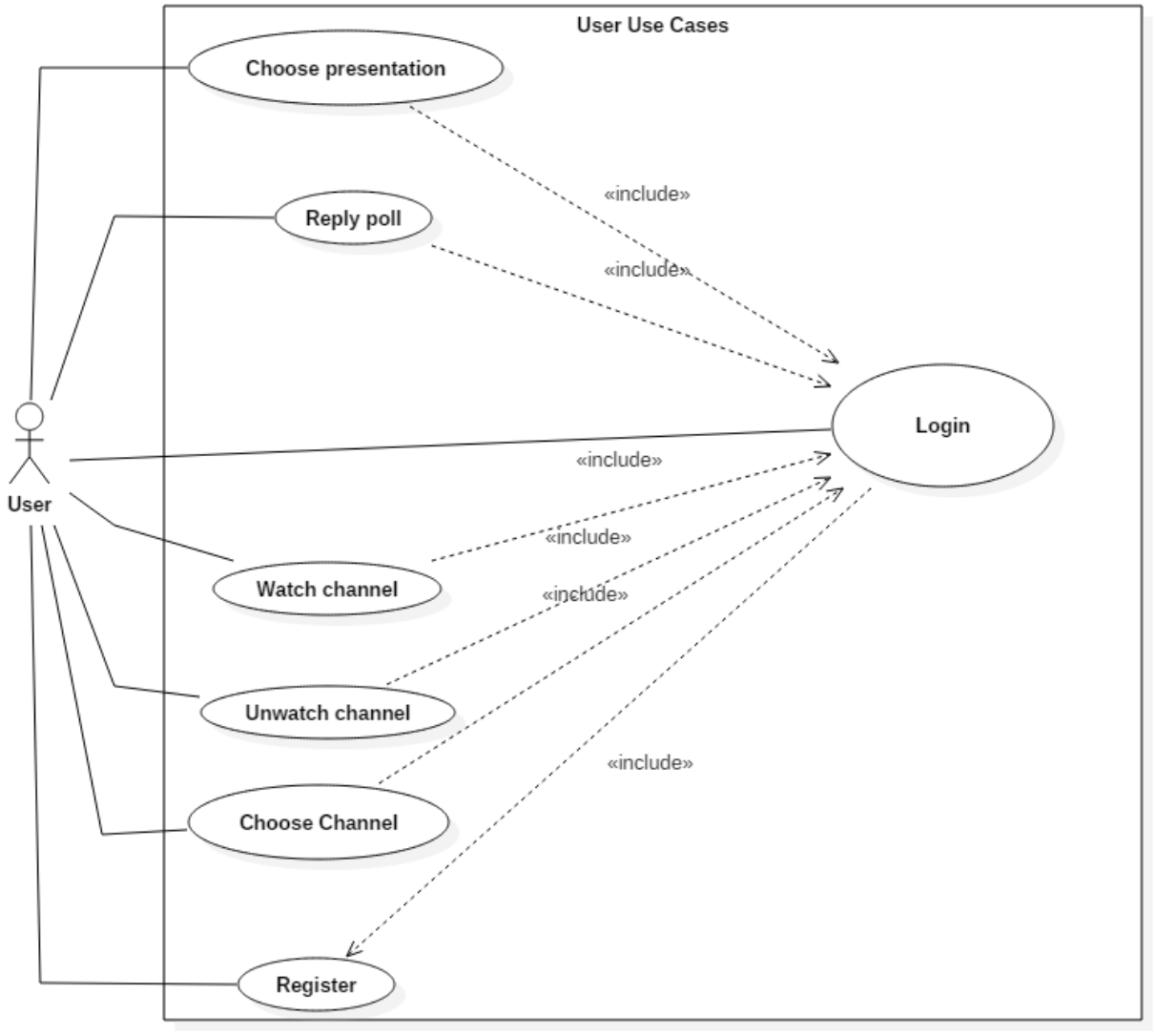

Figure 1: Use Case Diagram for “User”

1.3.2. Admin Use Case Diagram

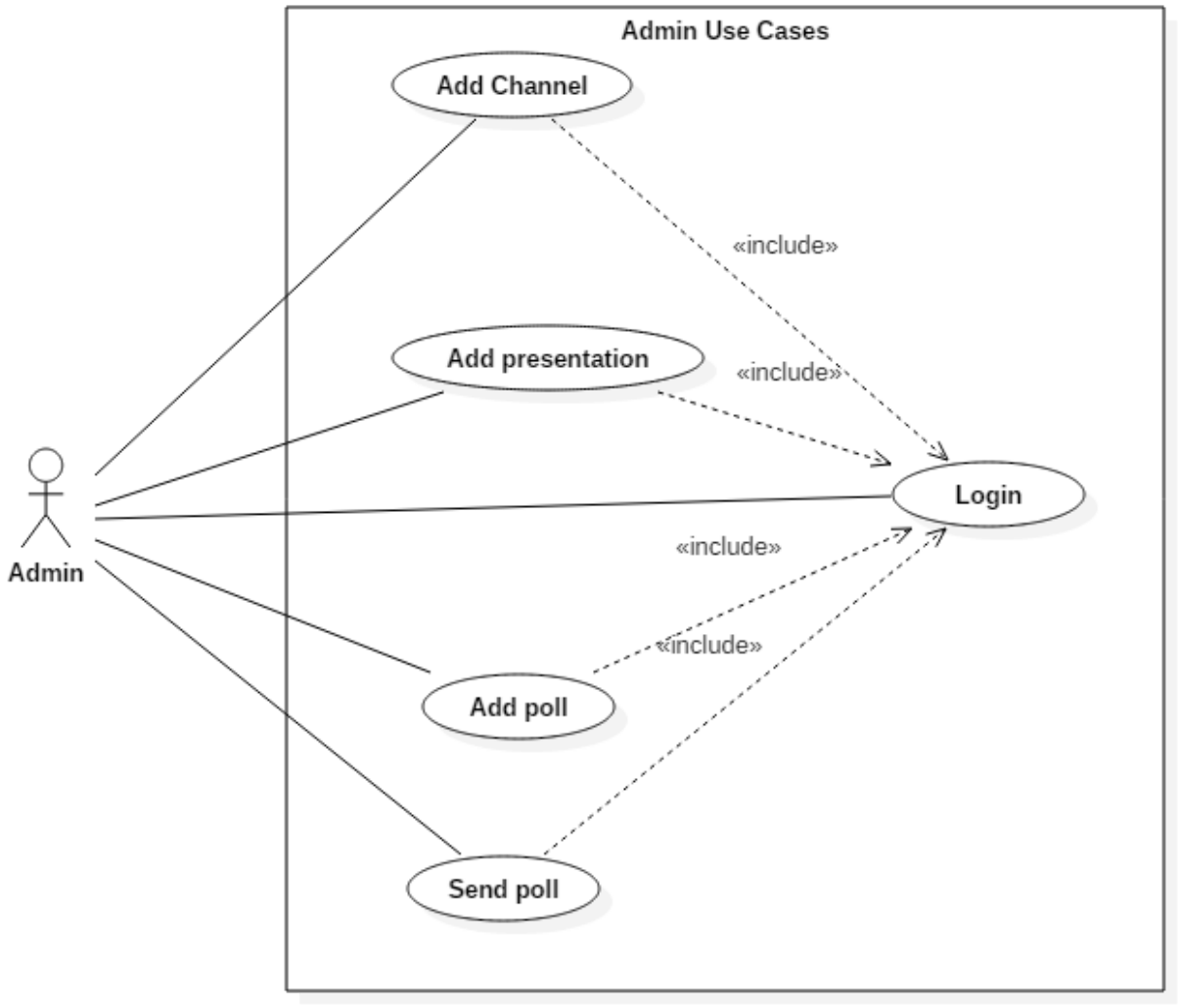

Figure 2: Use Case Diagram for “Admin”

2. SYSTEM DESIGN

2.1. Module Structure

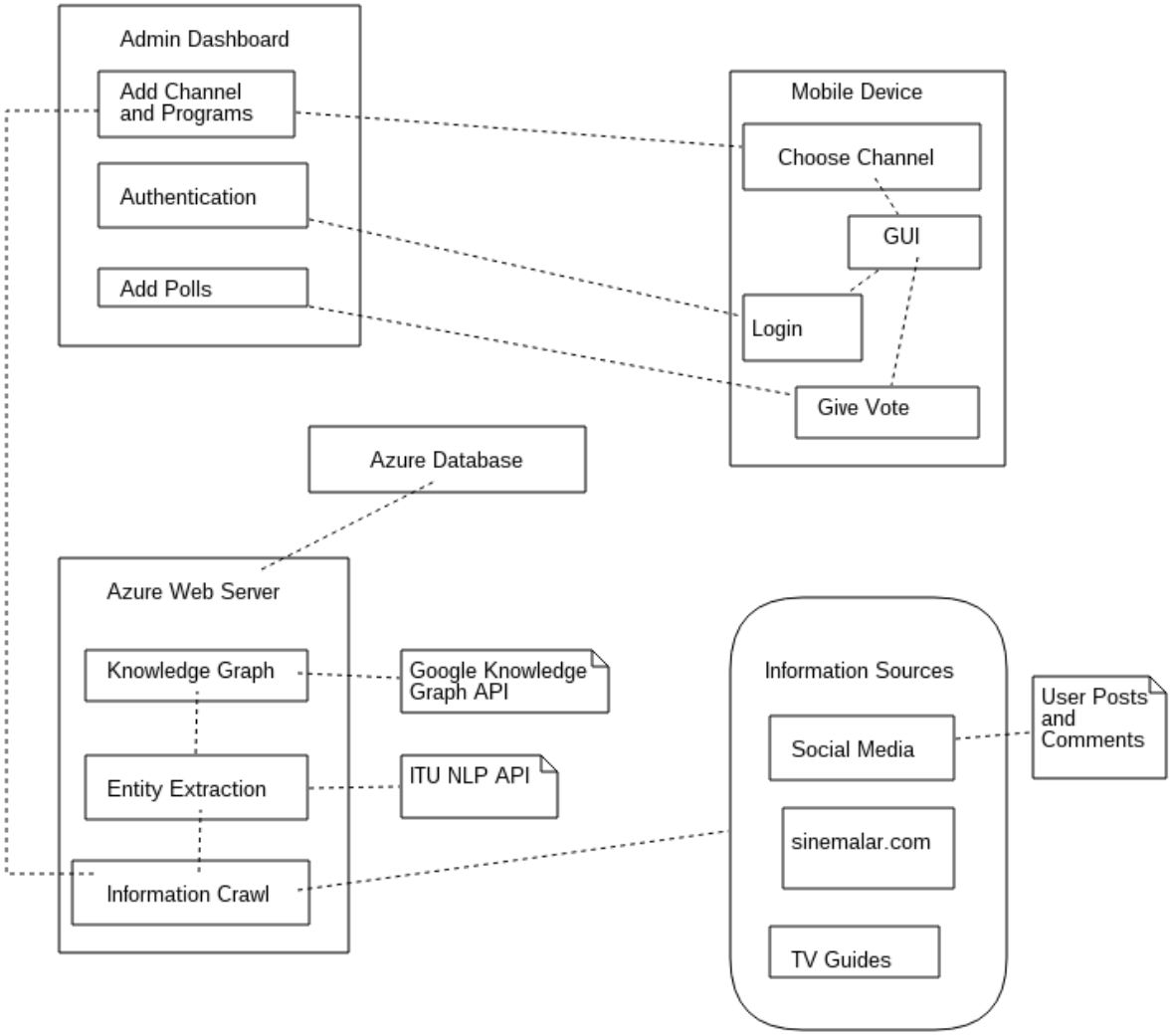

Figure 3: Module Structure

2.2. Source Code Structure

We have one project in our GitLab account; namely, Bitirsoft / intv-android-application. We have one branch within it, which is the master branch. Inside of it, we have four main directories:

1. INTV_Entity_Extraction: This contains source codes of our entity extraction module, which is intended to be run in the server side and it is written independently of our server-side code; i.e. it is written concurrently with it.
2. INTVandroid/inTV: This comprises codes for the client-side of the app; i.e., the GUI of the mobile app.
3. inTVAdmin: This comprises codes for the admin-side. It includes the GUI of the admin dashboard, codes for GUI operations, sending HTTP requests to the database in order to record data.
4. inTVdB: This contains the database-related codes such as handling the HTTP requests which come from the server-side, table creation scripts, and so on.

The reason why we followed such a structure is the fact that we had done the labor dividing within our group accordingly in advance.

2.3. Component Diagram

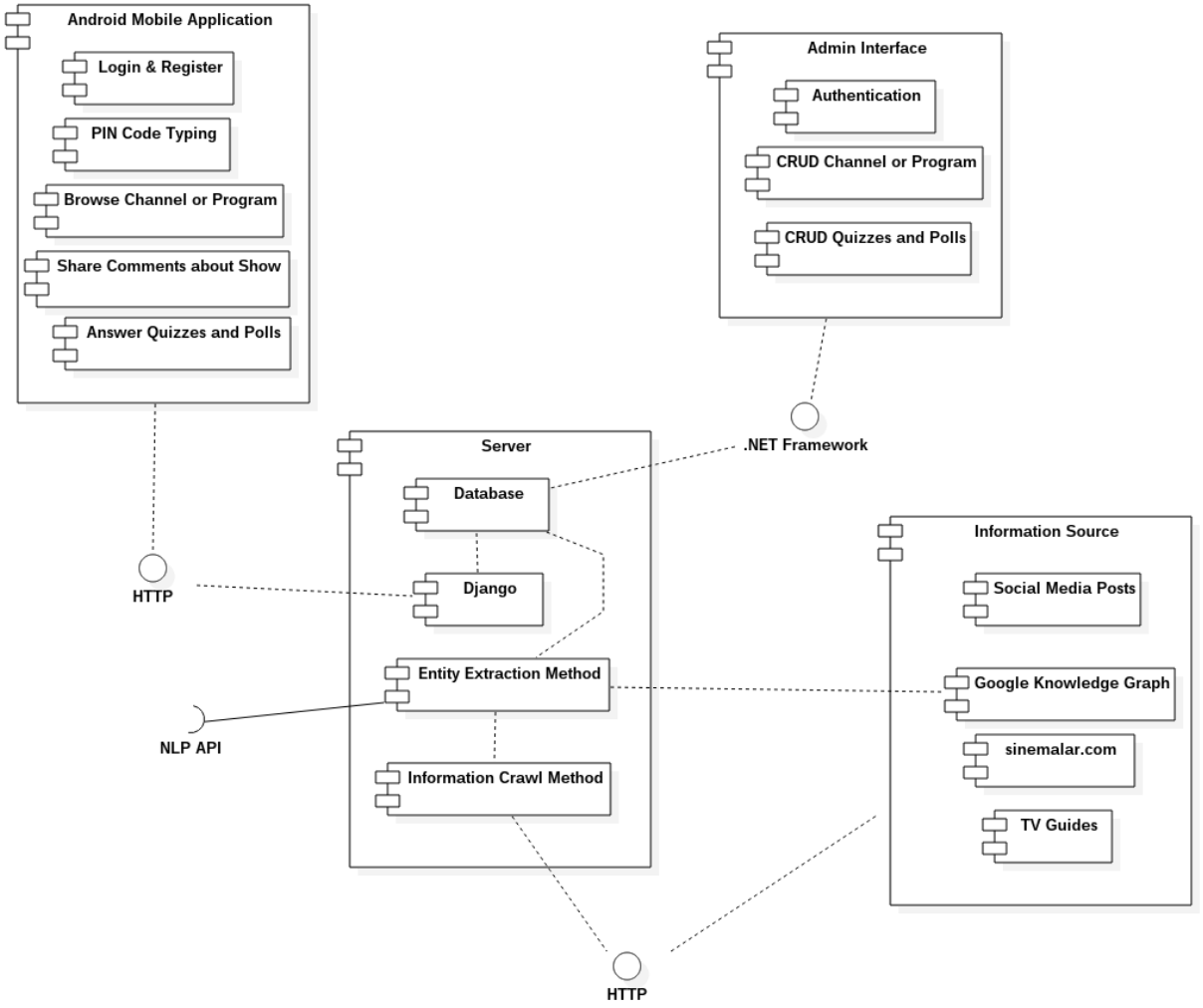

Figure 4: Component Diagram

2.4. Deployment Diagram

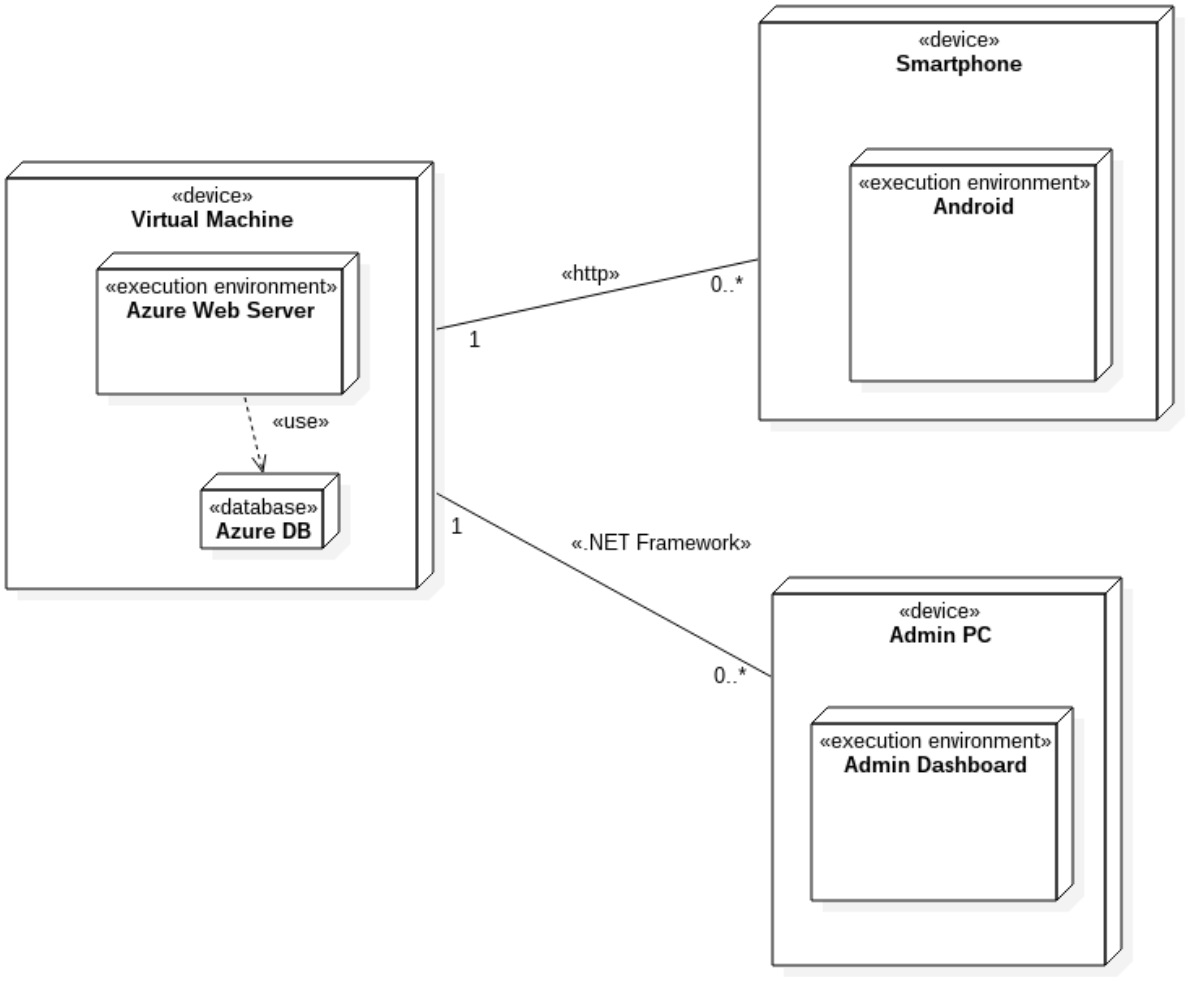

Figure 5: Deployment Diagram

3. TESTING

3.1. Testing Code Structure

Test coding used only for unit tests and NLP API functionality. Hamcrest matcher framework is used for that purpose. Created objects from every class using their all constructors. All methods of classes are tested. Tests are extended to cover possible null values. Unit tested classes are Has, Program, Survey, Airtime, Channel and Component. Located at Android Application Project. As for NLP API we used code from their own website.

3.2. Test Cases and Results

Test Case ID	T-1-Unit
Scenario	Creating object from every class and calling all their constructors. Testing null values with their constructors. Included classes are: Has, Program, Survey, User, Airtime, Channel, Component.
Pass/Fail Criteria	All of them should create an object with properties given at constructors. No result except successful creation of objects with given properties accepted.
Result	No problem detected. Classes works as intended.

Test Case ID	T-2-UI (Mobile)
Scenario	Opening channel selection menu by dragging from left and tapping on button at top left corner.
Pass/Fail Criteria	Channel selection menu should slide in from left and channel list should appear with given list of channel. Other results does not pass.
Result	No problem detected. Channel menu opens as expected.

Test Case ID	T-3-UI (Mobile)
Scenario	Selecting a channel from channel menu and displaying program from that channel while channel selection menu closes.
Pass/Fail Criteria	When channel selected, Menu should slide back to left and disappear, current program on that selected channel should appear on screen with its properties. Missing program's Air time and program's Name are not acceptable. Also picture of either channel or image should display. Any one of them acceptable. An option to set program as Watching should appear on screen. There shouldn't be any other information on screen. Any information extra would fail the test.
Result	No problem detected. Program's Air time and Name displays correctly. Also an image shown (either program or channel image, not both).

Test Case ID	T-4-UI (Mobile)
Scenario	Switching from browsing mode to Watching mode by tapping on "Watching Now" button.
Pass/Fail Criteria	When a program set as Watching, it should start showing it's all Components, along with old information shown, it should also start showing mail icon on bottom right corner if a survey available on that time.
Result	No problem detected. Program's Components, component details, Air time and Name displays correctly. Also an image shown (either program or channel image)

Test Case ID	T-5-UI (Mobile)
Scenario	Opening survey by tapping on mail icon on bottom right corner.
Pass/Fail Criteria	A pop up with a question and survey choices should appear with sending or cancelling or skipping option. One of answers can expect text input from user or not depending on survey type. Multiple answers can be selected or not depending on survey type. Pop up should not cover all the screen. Tapping outside popup should select Cancel automatically.
Result	No problem detected. Pop up appears with question and choices. Depending on survey type answers either multichoice or single choice. Send cancel skip buttons available.

Test Case ID	T-6-UI (Mobile)
Scenario	Ending a survey with selecting Skip or Send.
Pass/Fail Criteria	After Skip or Send selected, survey popup window should disappear, also on main screen mail icon should not be available for that survey again. New mail icon should appear if there is another survey. New mail icon appearance can delay up to a minute.
Result	No problem detected. Different surveys ended with Skip or Send. New mail icon appeared.

Test Case ID	T-7-UI (Mobile)
Scenario	Switching from Login screen to Main Screen by entering account name and password and logging in or receiving error about wrong credentials.
Pass/Fail Criteria	Once application launched it should show Login screen. Once account name and password entered at login screen if they are matching it should switch to main screen where you can select Channel. If account name and password does not much, no action should be taken except giving an error.
Result	No problem detected. Correct credentials could login and wrong ones got error message and couldn’t login.

Test Case ID	T-8-UI (Mobile)
Scenario	Choosing PIN Code option from Channel selection menu and entering PIN Code
Pass/Fail Criteria	Once PIN Code option chosen from Channel menu, menu should disappear and text input popup should appear with Confirm and Cancel buttons. Pop up should not cover all screen.
Result	No problem detected. PIN Code input pop up appeared with Confirm and Cancel buttons.

Test Case ID	T-9-UI (Mobile)
Scenario	Entering PIN Code and getting presentation related information on screen or not accepting PIN Code if it is wrong.
Pass/Fail Criteria	Once PIN Code entered Confirm button should check it. If it is correct popup should disappear and presentation information should be displayed, otherwise it should pop up should stay and wait for new input.
Result	No problem detected. PIN Code entered both wrong and correct. Wrong one waited another input, and correct one returned Presentation information.

Test Case ID	T-10-UI (Admin)
Scenario	On Admin login page when correct password and username enter and login or can’t login.
Pass/Fail Criteria	Once input entered and Login button clicked if credentials are correct it should login, otherwise some error on input field should be shown. It should direct to main page if login is successful.
Result	No problem detected. Admin login is successful when credentials are correct, wrong login credentials raises small error message.

Test Case ID	T-11-UI (Admin)
Scenario	Selecting Add New Channel from menu on left. Selecting Add New Presentation from menu on left. Selecting Add New Poll from menu on left.
Pass/Fail Criteria	It should bring up a new form, that requires user to enter properties of the item that will be added.
Result	No problem detected. Forms brought up as expected with required fields.

Test Case ID	T-12-UI (Admin)
Scenario	Information entered required fields on Channel, Presentation and Poll adding forms. Confirm button clicked.
Pass/Fail Criteria	If form's fields are valid it should return to available Channel, Presentation or Poll list depending on which form it was. Otherwise it should show error on input fields.
Result	No problem detected. When entered valid inputs, page changed according to which item is added and brought up its list. When invalid inputs entered page didn't change and errors on fields shown.

Test Case ID	T-13-UI (Admin)
Scenario	Selecting item from Channel, Presentation or Poll and editing or deleting selected item.
Pass/Fail Criteria	Once an item selected(click to item) from list it should be editable, Deleting is editing as inactive instead of removing, so they shouldn't be removed from list immediately. Also selecting an item should show its fields and allow admin to edit them. Clicking Save button should save the changes.
Result	No problem detected. Information edited on selected item and saved via Save button.

Test Case ID	T-14-UI (Admin)
Scenario	Sending poll to users by clicking Send Poll from a poll's page.
Pass/Fail Criteria	Once a poll selected from opened page, we should be able to click Send Poll and it should give a success failure message depending on result.
Result	No problem detected. Polls sent successfully message received.

Test Case ID	T-15-Integration (mobile+server)
Scenario	Mobile application getting Channel List, Program Details, Component Details, Login Authorization, Surveys from database using http requests sent to Django on server machine. Django fetched required information from database and returned a response to user application on mobile.
Pass/Fail Criteria	Django should be able to respond to fast, consecutive requests. Requests should return 200.
Result	Our tests completed without hitting fail criteria.

Test Case ID	T-16-Integration (admin+server)
Scenario	Admin application sending Add Channel, Login Authorization, Add Surveys, Add Presentation requests as well as their Get counterparts to Django on server machine. Django handles database changes according to those requests.
Pass/Fail Criteria	Django should be able to respond to fast, consecutive requests. Requests should return 200.
Result	Our tests completed without hitting fail criteria.

Test Case ID	T-17-Integration (Entity handling program+server)
Scenario	Entity handler application sending Add Component, Add Component Fields, Add Related requests to Django on server machine. Django makes necessary additions to database.
Pass/Fail Criteria	Django should be able to respond to fast, consecutive requests. Requests should return 200.
Result	Our tests completed without hitting fail criteria.

Test Case ID	T-18-API (NLP API)
Scenario	Test code of NLP API run before using it with example texts.
Pass/Fail Criteria	It should be able to extract entities successfully with their types. Within maximum of 30 seconds.
Result	Our tests completed without hitting fail criteria. Most of them respond within 15 seconds mark. Extracted entity results were satisfying.

Test Case ID	T-19-Validation
Scenario	Once channel selected, daily information crawled from TV guide(s) on the Internet and which program user currently watching is found. Also PIN Code can be used for same purpose.
Pass/Fail Criteria	Program should be able to detect what is being watched. (Identification)
Result	Depending on TV guide we use as source and how well known show is, we can bring desired results. PIN Code is guaranteed result, especially useful for presentations.

Test Case ID	T-20-Validation
Scenario	Social media APIs used in order to get real time information from Twitter and similar sites, also Google knowledge graph used in order to provide further data about show.
Pass/Fail Criteria	Providing real-time data about current show. (Enrichment)
Result	We can get information from Google Knowledge graph, especially detailed information for more known shows.

Test Case ID	T-21-Validation
Scenario	Users can read-respond polls and receive notifications and give feedback via polls
Pass/Fail Criteria	Provide user interaction interface, polls related to show, online quiz. (Interaction)
Result	Polls are used for online quiz as well, also polls can have text input field which allows for feedbacks.