
InnoCalendar

Group 7
Ilgiz Mustafin
Amir Gimaev
Mikhail Sheinberg

Project Introduction & Business Goals

Introduction

- **InnoCalendar:** *“A service extracting elective course information from Google Docs to Google Calendar for individual course selection”*
- **Problem:** *“Elective courses schedule is available in Google Docs, and the schedule changes. It is hard to track changes. In the current situation it is required to open Google Docs every day in order to see updates. However, there are no notifications at all.”*

Proposal Evolution

Before

"A service extracting elective course information from Google Docs to Google Calendar for individual course selection."

After

"A Telegram Bot which provides schedule of Elective Courses."

Reason

"Replacing the non-realizable requirement"

Quality of Interview Transcript

Individual Transcript Analysis

We received one pair of audio-transcript and compared the audiofile with the transcript. Use cases document contains the merged transcript.

- Transcript is not word-by-word representation of the interview
- Transcript has only the main outcomes of the interview
- Some possibly relevant points were missed in the transcript (technical requirements, functional requirements)

Merged Transcript Analysis

We compared the merged interview transcript with the audio of one interview.

- Merged transcript contains some of the points missing in the individual transcript (Technical requirements)
- Merged transcript misses specific numbers which are present in the individual interview

Discussion

- Each of the transcripts on their own are not enough (some information is present only in one of the files)
- Together they better represent the requirements of the customer

Quality of the use cases

Analysis Plan

1. Analyze the initial use cases document
2. Analyze changes in the document

Initial Use Cases Document

- Preconditions and assumptions are too broad
(*“... user must have a Telegram account”,
“The student’s internet connection is on or is within a range of
network connectivity to receive the notification”*)
- Precondition and assumption are the same
- Postcondition tells what should be done, not what should
be the new state
(*“Telegram bot notices the change between the google sheet
and the database and sends notification”*)

Initial Use Cases Document 2

- Alternate flows and exception are too far from the use case
*(“Telegram’s server is down and bots are inactive”,
“The user doesn’t receive the notification because his internet connection is turned off and misses the class”)*
- Postcondition tells what other actors should do
(“User views google sheet of entire elective course”)
- Application as an actor

Evolution

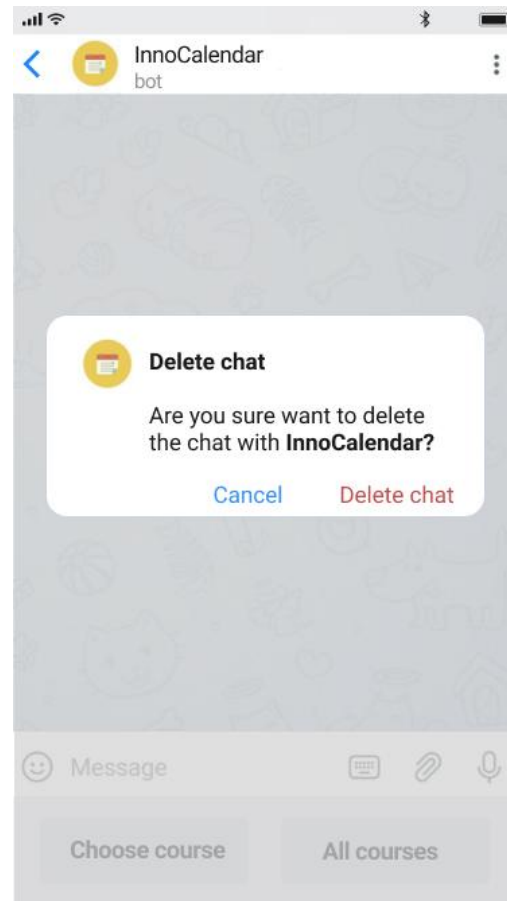
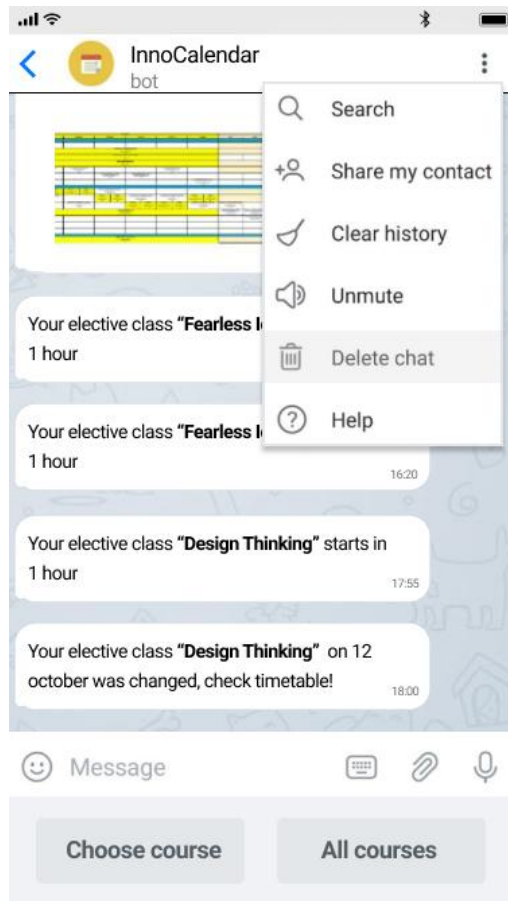
Use case changes have no modification reasons recorded

- “Telegram bot responds by sending a list of buttons UI elements”
- “Telegram bot at every 12 hours 30 min interval scans the elective course google sheet...”
- “A notification with the course details venue and time are sent to the user”

Quality of the MVP and user interfaces

User Interfaces

- UI is traceable with the use case
- Shows the final look of the project
- Wrong implementation of the use case “User wants to unsubscribe from the bot” (next slide)



“User wants to unsubscribe from the bot”

MVP

MVP works and covers all use cases present at the moment of the implementation. The code is correct, but there were some problems:

- Project is runnable only in IntelliJ IDEA with default settings → hard to run on a server/without IDEA
- Common language code style conventions are not followed (snake case names in Java code)
- Strong coupling → hard to add tests

Experience with quality management

Unit Tests Implementation

- Application uses `org.telegram.telegrambots` library for Java for Telegram bot implementation
- Custom bots should override method `void onUpdateReceived(Update)` and send messages from this method
- To add unit tests we used a function *Update* → *Response* to test resulting messages before sending them to user
- Now we can mock Updates and test bot responses

Running Unit Tests

Unit tests can be launched by running `mvn clean test`.

Next slide will show the video of running the tests.

```

[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ innoelectivebot ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 2 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:compile (default-compile) @ innoelectivebot ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 11 source files to /home/ilgiz/uni/re/InnoCalendar/target/classes
[INFO] /home/ilgiz/uni/re/InnoCalendar/src/main/java/dao/UserDao.java: Some input files use unchecked or unsafe
operations.
[INFO] /home/ilgiz/uni/re/InnoCalendar/src/main/java/dao/UserDao.java: Recompile with -Xlint:unchecked for detai
ls.
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ innoelectivebot ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /home/ilgiz/uni/re/InnoCalendar/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ innoelectivebot ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 2 source files to /home/ilgiz/uni/re/InnoCalendar/target/test-classes
[INFO] /home/ilgiz/uni/re/InnoCalendar/src/test/java/BotTest.java: /home/ilgiz/uni/re/InnoCalendar/src/test/java
/BotTest.java uses or overrides a deprecated API.
[INFO] /home/ilgiz/uni/re/InnoCalendar/src/test/java/BotTest.java: Recompile with -Xlint:deprecation for details
.
[INFO]
[INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @ innoelectivebot ---
[INFO]
[INFO] -----
[INFO] T E S T S
[INFO] -----
[INFO] Running service.CourseServiceTest
[INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.029 s - in service.CourseServiceTest
[INFO] Running BotTest
ERROR StatusLogger No Log4j 2 configuration file found. Using default configuration (logging only errors to the
console), or user programmatically provided configurations. Set system property 'log4j2.debug' to show Log4j 2 i
nternal initialization logging. See https://logging.apache.org/log4j/2.x/manual/configuration.html for instructi
ons on how to configure Log4j 2


```


Resulting Software

Final Software


InnoCalendar bot is available on Telegram – **@iellactivebot**

Next slide will show the video of bot usage.





Global search results




InnoElectiveBot


@iellectivebot

No messages found

InnoElectiveBot

bot





Introduction to Public Speaking for IT-specialist (cours...

Personal Efficiency skills of IT-specialist (course in Ru...

Technical Writing and Communication

Venture Capital Hacks in IT industry: From Zero to Ne...

Value Pursuit


Strategic Management of Technology and Innovation


International Trade in IT: Procedures and Regulations


Applied Economics: Introduction to IT Entrepreneurship

Science Ethics

Russian as a foreign language

 Write a message...





Choose course

All courses

Unsubscribe

Dialogues Architecture

- Implemented architecture is just a `switch` on the received update message
 - Bot doesn't take into account the previous dialogue
-
- We propose to implement dialogues as state-machines
 - Easier to add new dialogues
 - No limitation on restricting each message to have one action

Teamwork

Analysis

- We didn't face any difficulties while communicating with the other team
- Their responds were quick and to the point

Questions