


Branch: **implementation** ▾

[InnoCalendar](#) / [Docs](#) / **use\_cases.md**

Find file

Copy path

 **zhenytim** Update use\_cases.md 1c1fb35 3 minutes ago

1 contributor

71 lines (55 sloc) 5.58 KB

RawBlameHistory

### Use case scenarios

Use case #1	User starts the bot
Actors	Student, Bot
Pre-conditions	The student must have a telegram account The elective course schedule should be available on Google sheets
Flow of Events	1. The user sends the initiating command to the Telegram bot 2. Telegram verifies if the user sent the right command 3. Telegram bot responds by sending a list of UI elements to choose course, view entire elective course sheet
Post-conditions	User is registered in InnoCalendat bot
Alternate flows and exceptions	The user doesn't send the right command
Assumption	The user has a Telegram account

Use case #2	View entire elective course schedule
-------------	--------------------------------------

<b>Use case #2</b>	<b>View entire elective course schedule</b>
Actors	Student,Telegram bot
Pre-condition	Students sends the right command to initiate communication with bot
Flow of Events	<ol style="list-style-type: none"> <li>1. The student clicks on the button to view all courses</li> <li>2. The telegram bot sends a link to the google sheet containing elective course</li> </ol>
Post-conditions	User gets link for google sheet of entire elective course
Alternate flows and exceptions	<ol style="list-style-type: none"> <li>1. The user doesn't send the right command to the telegram bot</li> <li>2. The user doesn't have internet connection on his phone and link fails to load</li> </ol>
Assumption	The user has internet connection on his phone as to load the page containing google sheet

<b>Use case #3</b>	<b>User wants to choose an elective courses in order to get notifications</b>
Actors	Student, Telegram bot
Pre-condition	User sends the right command to initiate communication with bot Course to be registered by student exists
Flow of Events	<ol style="list-style-type: none"> <li>1. The student initiates communication with telegram bot</li> <li>2. Telegram bot sends lists of buttons to choose exact course for notifications</li> <li>3. When the user is provided with the</li> </ol>

<b>Use case #3</b>	<b>User wants to choose an elective courses in order to get notifications</b>
	intended course, the user can register for it by clicking on the course button. User can click on course button again to cancel registration for this course.
Post-conditions	Information about user's choice saved in internal database
Alternate flows and exceptions	Telegram bot fails to save user's choice details into database because the phone was out of network coverage
Assumption	The course intended to be registered for is found

<b>Use case #4.1</b>	<b>The bot sends notifications when Admin makes changes in the elective course schedule on the Google Sheet</b>
Actors	Telegram bot, Admin, Student
Pre-condition	Google Sheets to be modified exist Telegram bot must have stored elective course schedule in the database Students must be registered for the course to be notified
Flow of Events	<ol style="list-style-type: none"> <li>1. The Admin modifies record on the Google Sheet to elective course schedule</li> <li>2. The Telegram bot at every 30 min interval scans the elective course google sheet and compares it against the stored information in the database</li> <li>3. If the Telegram notices any change between the database and the google sheets, it sends a notification to user about</li> </ol>

<b>Use case #4.1</b>	<b>The bot sends notifications when Admin makes changes in the elective course schedule on the Google Sheet</b>
	changes
Post-conditions	Telegram bot notices the change between the google sheet and the database and sends notification
Alternate flows and exceptions	<ol style="list-style-type: none"> <li>1. Telegram's server is down and bots are inactive</li> <li>2. The user doesn't receive the notification because his internet connection is turned off</li> </ol>
Assumption	<p>That Google sheet to be modified by the admin exists and this current information are stored in the database to be accessed by the bot</p> <p>The student's internet connection is on or is within a range of network connectivity to receive the notification</p>

<b>Use case #4.2</b>	<b>The bot sends notifications one hour before commencement of elective course</b>
Actors	Telegram bot, Student
Pre-condition	Students must be registered for the course to be notified
Flow of Events	<ol style="list-style-type: none"> <li>1. The telegram bot scans the database every 5 min</li> <li>2. If after scanning the database, the bot notices that a course is scheduled to start in an hour, the telegram bot sends a notification to students that registered for</li> </ol>

<b>Use case #4.2</b>	<b>The bot sends notifications one hour before commencement of elective course</b>
	that course 3. A notification with the course details are sent to the user
Post-conditions	Notification is sent by telegram bot to user's(student) phone
Alternate flows and exceptions	1. The user doesn't receive the notification because his internet connection is turned off and misses the class 2. Telegram's server is down and bots are inactive
Assumption	The user will receive the notification early enough to prepare for class

<b>Use case #5</b>	<b>User wants to unsubscribe from the bot</b>
Actors	Student, Telegram bot
Pre-condition	Student subscribed to the bot to get notification about the elective course schedule
Flow of Events	1. Student initiates deletion 2. Student confirms the unsubscription 3. Bot will unsubscribe the user 4. Bot deletes student's data from the database
Post-conditions	Chat with Bot will be deleted from Student's telegram account
Alternate flows and exceptions	Student doesn't confirm the unsubscription

Use case #5	User wants to unsubscribe from the bot
Assumption	The student has a Telegram account

### Appendix

**The elective course schedule** - the time table for the elective courses managed by Admin on Google sheets.

**Notification** - An alert message sent by bot to the user on Telegram.

**Telegram Bot** - is an embedded application that run inside Telegram.

**Admin** - the person who make changes to the schedule on on google sheet.

**User** - the person who wants to get notified and initiate commands from the bot.