Міністерство освіти і науки України

Національний університет «Львівська політехніка»



**Лабораторна робота №1**

з дисципліни: «Інженерія програмного забезпечення»

на тему: « Система моніторингу рейтингу студентів та пошуку кращих»

Виконав: ст. гр. КІ-32

Бабічок В.М

Прийняв:

Цигилик Л.О.

Львів – 2020

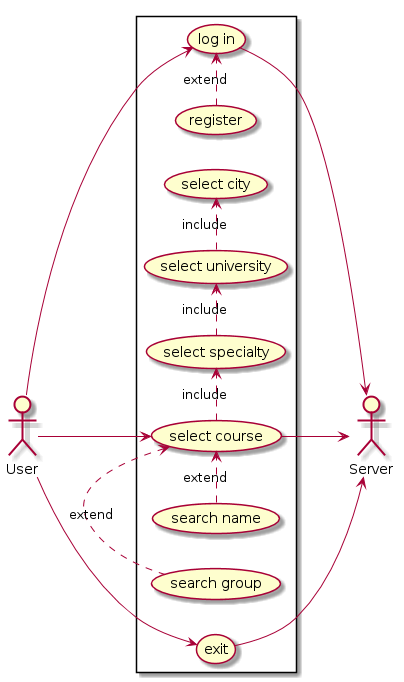
Тема: UML діаграми варіантів використання та UML діаграми активності.

Мета: Освоїти принципи створення UML діаграм варіантів використання що описують сценарій роботи системи (Use case diagram) та діаграм активності для клієнтської та серверної частин.

Постановка задачі: розробити UML діаграми сценаріїв для клієнтської і серверної частини, а також діаграми активності для кожного сценарію.

Користувач має можливість зареєструватись або увійти в систему як зареєстрована особа. Сама система дає можливість переглядати успішність того чи іншого студента. У програмі передбачений пошук міст України, університетів та спеціальності, також користувач за бажанням може шукати конкретного студента або конкретну групу в якій навчаються студенти. Усі ці дані будуть зберігатись на сервері, а клієнтська частина буде лише відображати отримані результати від сервера. Сервер буде виконувати всю роботу, тобто пошук інформації, обробляти та відповідати на запити користувача.

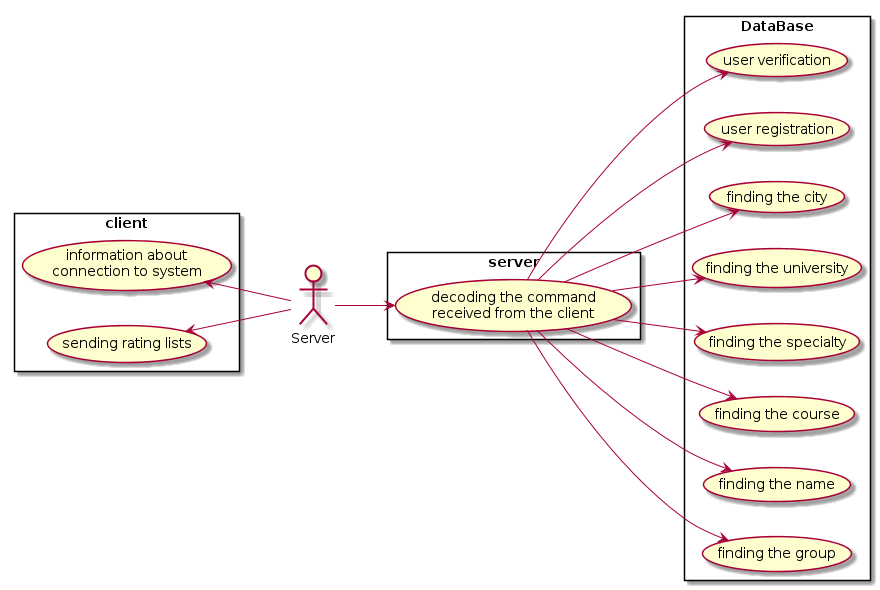
UML діаграма клієнтської частини



Опис діаграми

Користувач при відкритті програми має вибір реєструватись, якщо це новий користувач, або ж авторизуватись як існуючий користувач. Наступне вікно відображає список з переліком міст України, після вибору підходящого міста, висвітлюється список університетів, після вибору університету, користувач має змогу вибрати зі списку спеціальностей, йому цікаво, після чого він вибирає курс на якому навчаються студенти. Після цих маніпуляцій користувачу відображається рейтинговий список, тоді по бажанню користувача, він може у списку шукати конкретного користувача або конкретну групу. Якщо користувач завершив моніторинг рейтингових списків, він може вийти із програми. Всі обрахунки і всі запити будуть відправлятись на сервер.

UML діаграма серверної частини

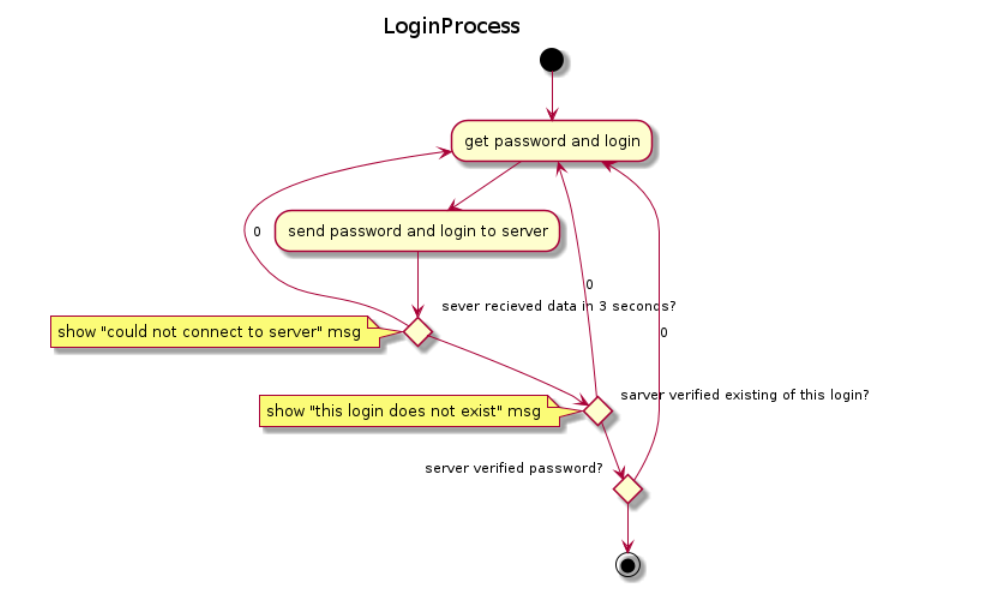


Опис діаграми

При надходженні запитів до сервера від клієнта, сервер декодує інформації щоб дізнатись що саме користувач хоче дізнатись, це може бути як авторизація чи реєстрація користувача, також це може бути вибір міста, університету, спеціальності, курсу, ім’я, групи і всі ці результати серверу потрібно надіслати до користувача.

Клієнтська частина

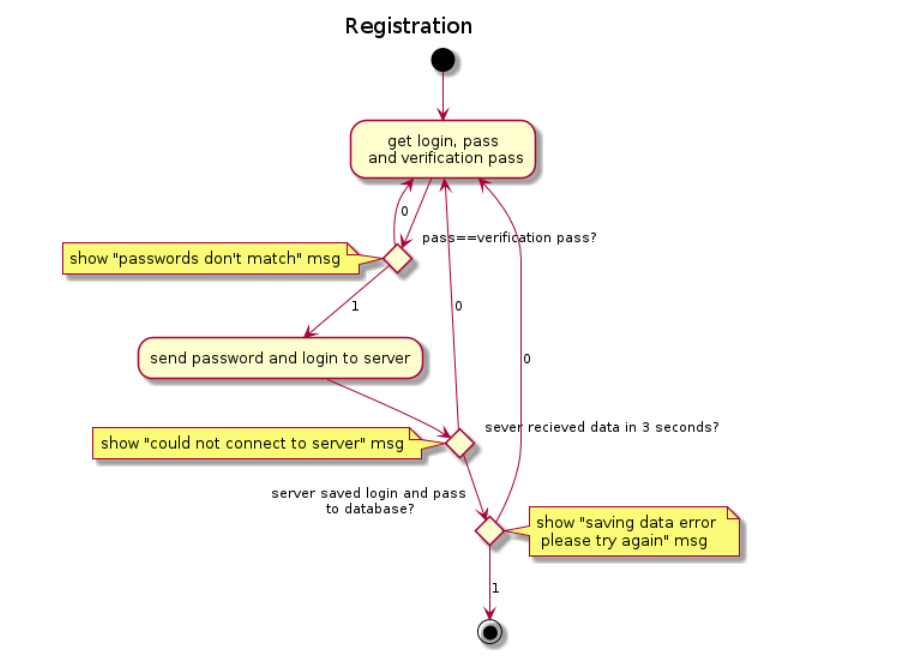
Процес авторизації



Опис

Користувач вводить логін та пароль, далі ці дані відправляються на сервер, якщо сервер не отримав нічого, то виводиться повідомлення про проблеми з під’єднанням до сервера, після чого користувач знову вводить логін та пароль. Якщо ж сервер отримав логін і пароль, то відбувається перевірка логіну чи він ідентичний який зберігається у базі даних, якщо ні то повертаємся до вводу логіну і паролю, наступний крок відбується перевірка паролю, якщо він не вірний який є у базі даних, то повертаємось до вводу логіну і паролю, а якщо пароль вірний то користувач входить в систему.

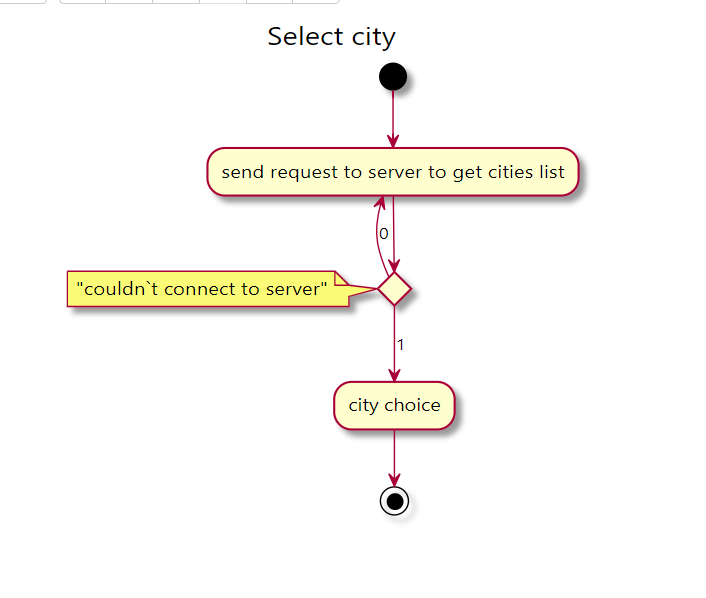
Процес реєстрації



Опис

Користувач вводить логін, пароль та повторно пароль, після чого проходить перевірка паролю чи вони співпадають між собою, якщо не співпадають, то виводиться повідомлення, що паролі не однакові і користувач змінює паролі так, щоб вони співпадали. Якщо ж вони однакові, то вони надсилаються на сервер, якщо сервер їх не отримав, то виводиться повідомлення про проблеми з підключенням до сервера, однак якщо він отримав пароль і логін, то він їх зберігає в базі даних, але якщо не сервер не відовідає, то виводиться повідомлення, що не вийшло зберегти дані і користувач повертається до самого початку процесу реєстрації.

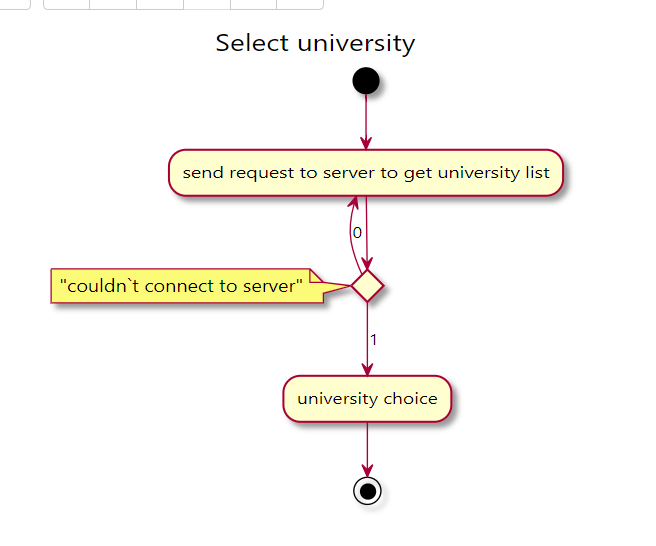
Вибір міста



Опис

Користувач надсилає запит на отримання списку міст, якщо сервер не відповідає, то виводиться повідомлення про проблеми із з’єднанням з сервером, однак якщо вдалось з’єднатись, то користувач може вибрати потрібне йому місто.

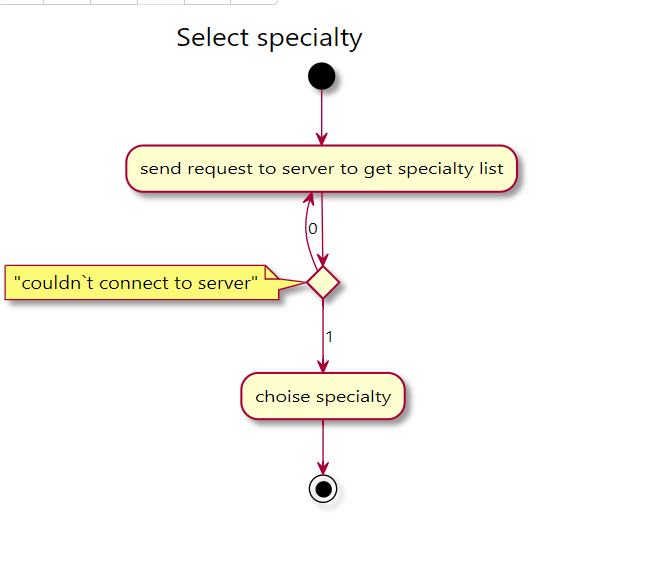
Вибір університету



Опис

Користувач надсилає запит на отримання списку університетів, якщо сервер не відповідає, то виводиться повідомлення про проблеми із з’єднанням з сервером, однак якщо вдалось з’єднатись, то користувач може вибрати потрібний йому університет.

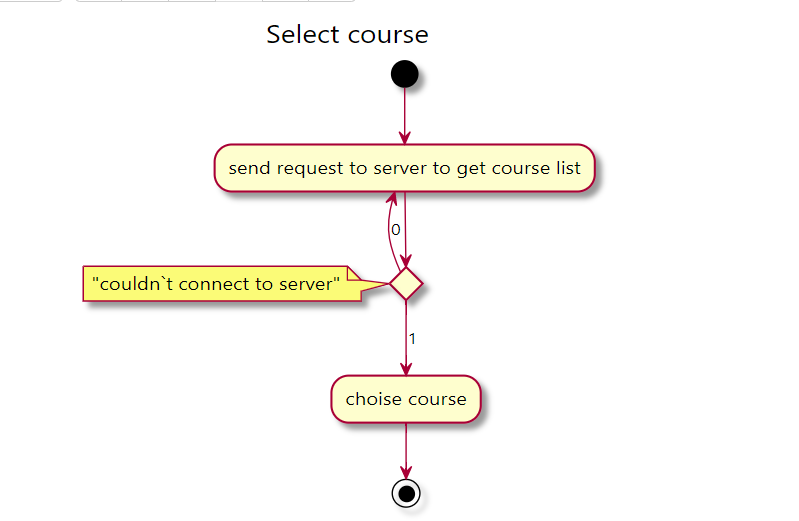
Вибір спеціальності



Опис

Користувач надсилає запит на отримання списку спеціальностей, якщо сервер не відповідає, то виводиться повідомлення про проблеми із з’єднанням з сервером, однак якщо вдалось з’єднатись, то користувач може вибрати потрібну йому спеціальність.

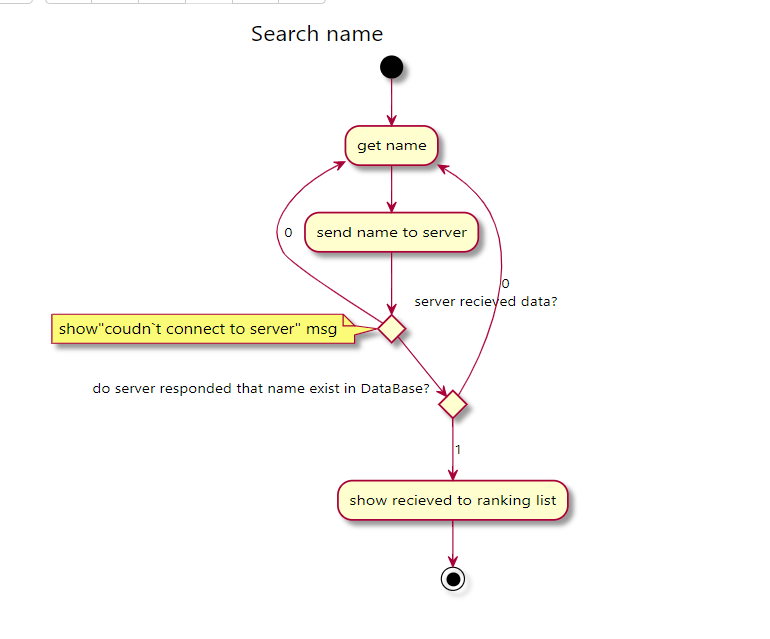
Вибір курсу



Опис

Користувач надсилає запит на отримання списку курсів, якщо сервер не відповідає, то виводиться повідомлення про проблеми із з’єднанням з сервером, однак якщо вдалось з’єднатись, то користувач може вибрати потрібний йому курс.

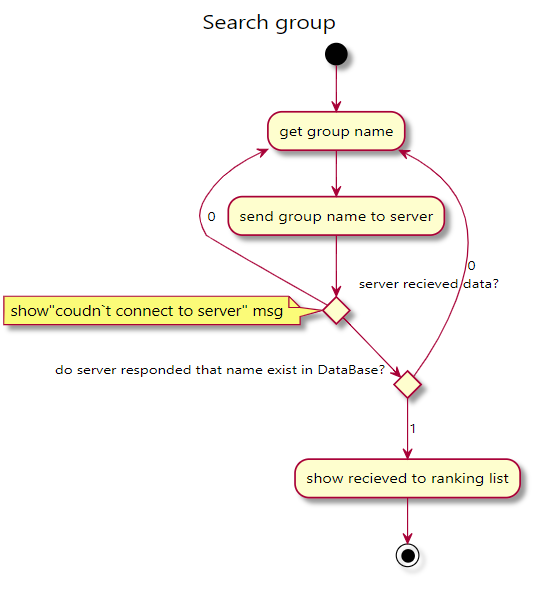
Пошук за ім’ям



Опис

Користувач вводить ім'я, воно надсилається до сервера, якщо сервер не відповідає то виводиться повідомлення, про проблеми із з’єднанням з сервером, якщо ж сервер отримав дані, то він перевіряє чи таке ім’я присутнє в базі даних, якщо ні, то користувач знову вводить ім’я, але якщо таке ім’я присутнє, то користувачу покаже його в рейтинговому списку.

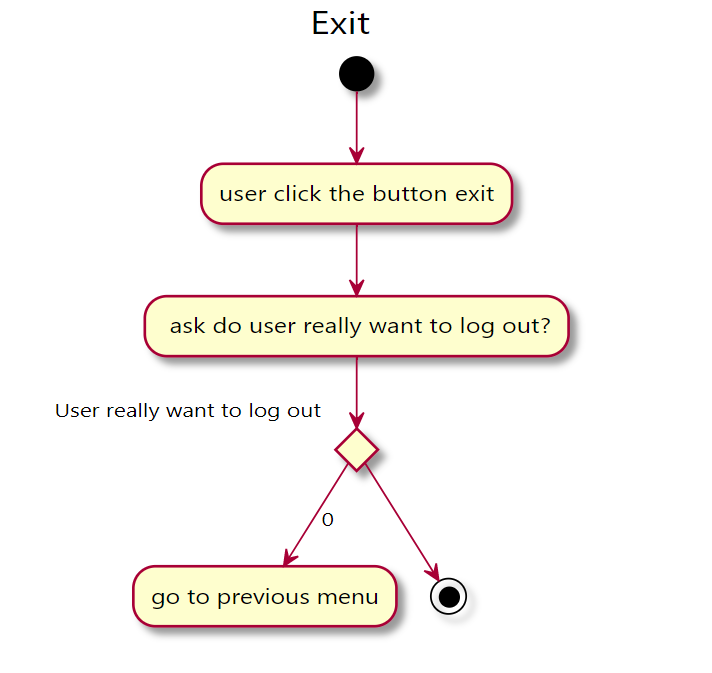
Пошук за групою



Опис

Користувач вводить назву групи, дані надсилається до сервера, якщо сервер не відповідає то виводиться повідомлення, про проблеми із з’єднанням з сервером, якщо ж сервер отримав дані, то він перевіряє чи така назва групи присутня в базі даних, якщо ні, то користувач знову вводить назву групи, але якщо така назва групи присутня, то користувачу покаже її в рейтинговому списку.

Процес виходу

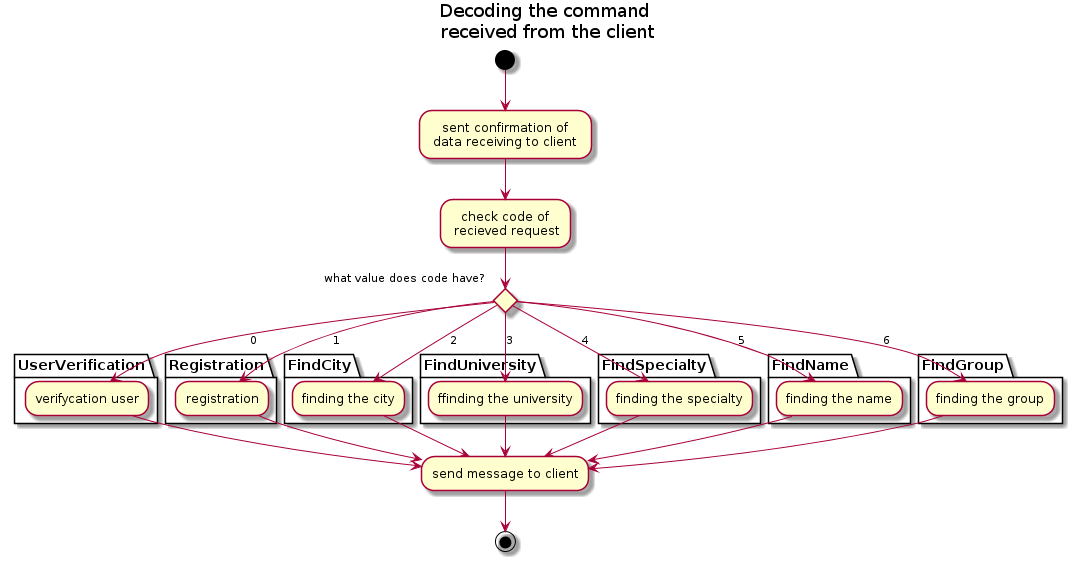


Опис

Користувач натискає кнопку «exit», програма запитує чи він дійсно хоче вийти з програми, якщо ні, то користувача перекидує в попереднє меню, якщо так, то користувач виходить. Ця перевірка зроблена для запобігання необачних натискань.

Серверна частина

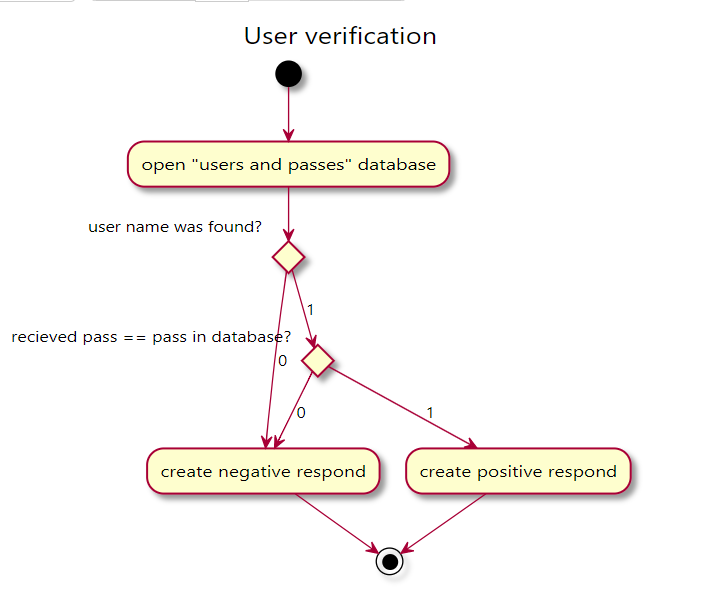
Декодування команд



Опис

Ця діаграма показує послідовність дій які виконуються при декодуванні команди які надходять від користувача до сервера. У кожної команди є свій код, що допомагає серверу визначити дії які потрібно виконати.

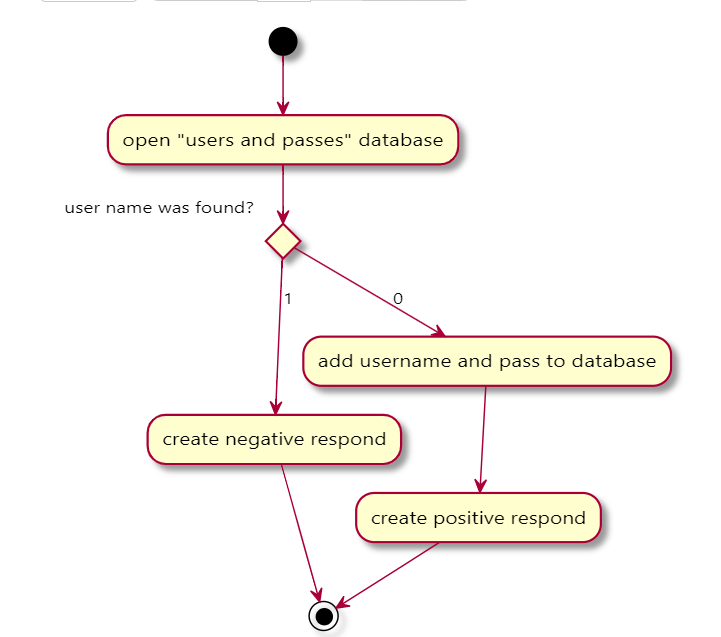
Верифікація клієнта



Опис

Відкривання таблиці «users and passes», перевірка чи присутній логін, якщо ні, то формується негативна відповідь, якщо знайдено такий, то перевіряється пароль, якщо пароль не вірний то формується негативна відповідь, якщо ж пароль присутній, то формується позитивна відповідь.

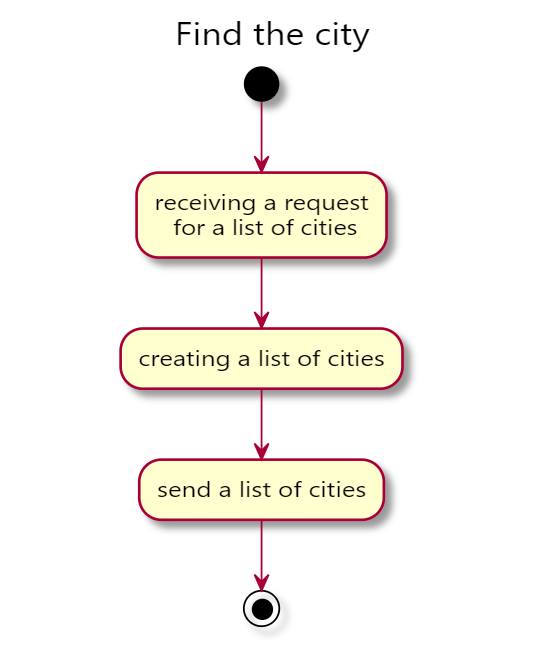
Реєстрація клієнта



Опис

Відкривання таблиці «users and passes», перевірка чи присутній логін, якщо ні, то логін і пароль записуються до таблиці в базі даних і формується позитивна відповідь , якщо такий вже існує, то формується негативна відповідь.

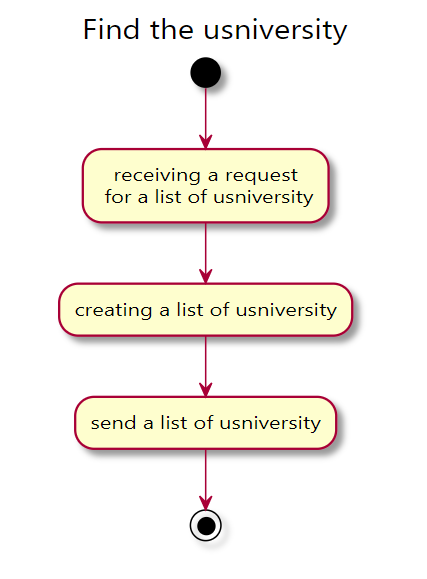
Знаходження міста



Опис

Діаграма показує як сервер формує список міст і надсилає їх користувачу. Сервер отримує запит на список міст, після чого він формує цей список і аж тоді він його надсилає.

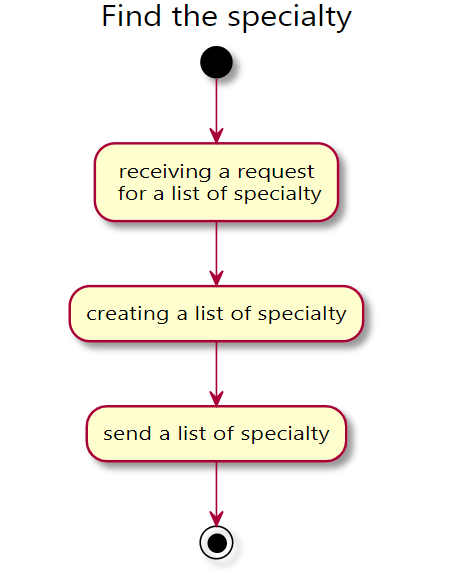
Знаходження університету



Опис

Діаграма показує як сервер формує список університетів і надсилає їх користувачу. Сервер отримує запит на список університетів, після чого він формує цей список і аж тоді він його надсилає.

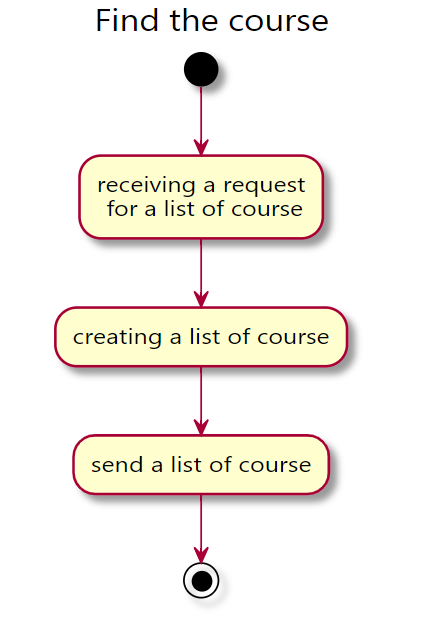
Знаходження спеціальності



Опис

Діаграма показує як сервер формує список спеціальностей і надсилає їх користувачу. Сервер отримує запит на список спеціальностей, після чого він формує цей список і аж тоді він його надсилає.

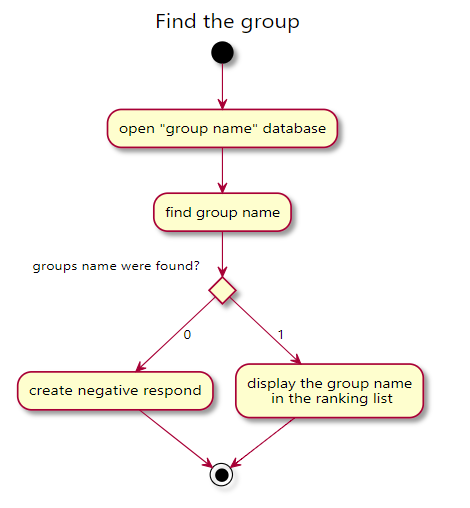
Знаходження курсу



Опис

Діаграма показує як сервер формує список курсів і надсилає їх користувачу. Сервер отримує запит на список курсів, після чого він формує цей список і аж тоді він його надсилає.

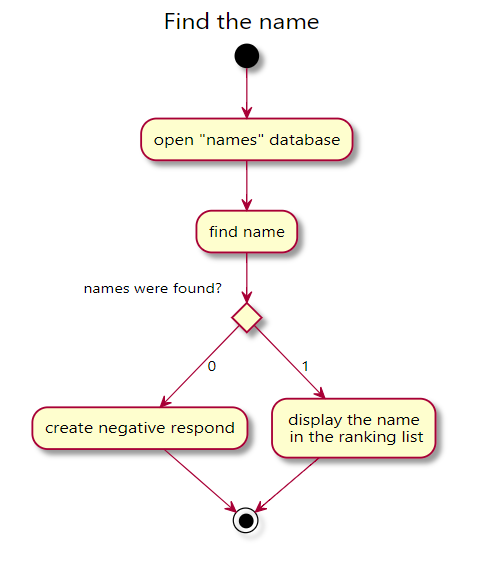
Знаходження групи



Опис

Під час виконання цього процесу сервер відкриває базу даних з назвами груп і шукає у ній потрібної назви. Якщо така назва не була знайдена то до клієнта буде відправлено повідомлення про відсутність такої групи, якщо ж така назва групи була знайдена, то сервер відобразить в рейтингових списках цю групу.

Знаходження ім’я



Опис

Під час виконання цього процесу сервер відкриває базу даних з переліком імен і шукає у ній потрібне ім’я. Якщо таке ім’я не було знайдено, то до клієнта буде відправлено повідомлення про відсутність такого ім’я в рейтинговому списку, якщо ж таке ім’я була знайдено, то сервер відобразить в рейтингових списках це ім’я.

Додотак з кодами до діаграм

UML діаграма сценарію роботи клієнтської частини

@startuml

left to right direction

skinparam packageStyle rectangle

actor User

actor Server

rectangle {

User --> (log in)

(log in) <.left. (register): extend

User --> (select course)

(select course).right.>(select specialty):include

(select specialty).right.> (select university):include

(select university).right.>(select city):include

(select course)<.left.(search name):extend

(select course)<.left.(search group):extend

User --> (exit)

(select course) -->Server

(log in) -down-> Server

(exit) -down-> Server

}

@enduml

UML діаграма сценарію роботи серверної частини

@startuml

left to right direction

skinparam packageStyle rectangle

actor Server

rectangle client {

(information about\n connection to system) <-- Server

(sending rating lists) <-- Server

}

rectangle server {

Server -->(decoding the command\n received from the client)

}

rectangle DataBase {

(decoding the command\n received from the client)-->(user verification)

(decoding the command\n received from the client)-->(user registration)

(decoding the command\n received from the client)-->(finding the city)

(decoding the command\n received from the client)-->(finding the university)

(decoding the command\n received from the client)-->(finding the specialty)

(decoding the command\n received from the client)-->(finding the name)

(decoding the command\n received from the client)-->(finding the group)

}

@enduml

Діаграми активності клієнтської частини та їх опис

Процес авторизації

@startuml

title LoginProcess

(\*) -->"get password and login"

-->"send password and login to server"

if "sever recieved data in 3 seconds?" then

note left: show "could not connect to server" msg

-left->[0]"get password and login"

else

if "sarver verified existing of this login?" then

note left: show "this login does not exist" msg

-->[0]"get password and login"

else

if "server verified password?" then

-->[0]"get password and login"

else

-->(\*)

endif

endif

endif

@enduml

Процес реєстрації

@startuml

title LoginProcess

(\*) -->"get login, pass\n and verification pass"

if "pass==verification pass?" then

note left: show "passwords don't match" msg

-left->[0]"get login, pass\n and verification pass"

else

-->[1]"send password and login to server"

if "sever recieved data in 3 seconds?" then

note left: show "could not connect to server" msg

-->[0]"get login, pass\n and verification pass"

else

if "server saved login and pass\n to database?" then

note right: show "saving data error \n please try again" msg

-->[0]"get login, pass\n and verification pass"

else

-->[1](\*)

endif

endif

endif

@enduml

Процес вибору міста

@startuml

title Select city

(\*)-->"send request to server to get cities list"

if"" then

note left:"couldn`t connect to server"

-left->[0]"send request to server to get cities list"

else

-->[1]city choice

-->(\*)

endif

@enduml

Процес вибору університету

@startuml

title Select university

(\*)-->"send request to server to get university list"

if"" then

note left:"couldn`t connect to server"

-left->[0]"send request to server to get university list"

else

-->[1] university choice

-->(\*)

endif

@enduml

Процес вибору спеціальності

@startuml

title Select specialty

(\*)-->"send request to server to get specialty list"

if"" then

note left:"couldn`t connect to server"

-left->[0]"send request to server to get specialty list"

else

-->[1] choise specialty

-->(\*)

endif

@enduml

Процес вибору курсу

@startuml

title Select course

(\*)-->"send request to server to get course list"

if"" then

note left:"couldn`t connect to server"

-left->[0]"send request to server to get course list"

else

-->[1] choise course

-->(\*)

endif

@enduml

Процес пошуку за ім’ям

@startuml

title Search name

(\*)-->"get name"

-->"send name to server"

if"server recieved data?" then

note left:show"coudn`t connect to server" msg

-up->[0]"get name"

else

if"do server responded that name exist in DataBase?" then

-up->[0]"get name"

else

-->[1]"show recieved to ranking list"

-->(\*)

endif

endif

@enduml

Процес пошуку за назвою групи

@startuml

title Search group

(\*)-->"get group name"

-->"send group name to server"

if"server recieved data?" then

note left:show"coudn`t connect to server" msg

-up->[0]"get group name"

else

if"do server responded that name exist in DataBase?" then

-up->[0]"get group name"

else

-->[1]"show recieved to ranking list"

-->(\*)

endif

endif

@enduml

Процес виходу

@startuml

title Exit

(\*)-->"user click the button exit"

-->" ask do user really want to log out?"

if"User really want to log out" then

-->[0]"go to previous menu"

else

-->(\*)

endif

@enduml

Діаграми активності серверної частини та їх опис

Процес декодування отриманої команди

@startuml

title Decoding the command\n received from the client

(\*) --> "sent confirmation of\n data receiving to client "

-->"check code of\n recieved request"

if "what value does code have?" then

partition UserVerification {

-->[0]"verifycation user"

}

-->"send message to client"

else

partition Registration {

-->[1]"registration"

}

-->"send message to client"

else

partition FindCity {

-->[2]"finding the city"

}

-->"send message to client"

else

partition FindUniversity {

-->[3]"ffinding the university"

}

-->"send message to client"

else

partition FindSpecialty {

-->[4]"finding the specialty"

}

-->"send message to client"

else

partition FindName {

-->[5]"finding the name"

}

-->"send message to client"

else

partition FindGroup {

-->[6]"finding the group"

}

-->"send message to client"

endif

-->(\*)

@enduml

Процес верифікації

@startuml

title User verification

(\*) --> open "users and passes" database

if "user name was found?" then

-->[0] create negative respond

-->(\*)

else

[1]if "recieved pass == pass in database?" then

-->[0] "create negative respond"

else

-->[1] "create positive respond"

-->(\*)

endif

endif

@enduml

Процес реєстрації

@startuml

title Registration

(\*) --> open "users and passes" database

if "user name was found?" then

-->[1] create negative respond

-->(\*)

else

-->[0]"add username and pass to database"

-->"create positive respond"

-->(\*)

endif

@enduml

Процес знаходження міста

@startuml

title Find the city

(\*) -->receiving a request\n for a list of cities

-->creating a list of cities

-->send a list of cities

-->(\*)

@enduml

Процес знаходження університету

@startuml

title Find the usniversity

(\*) -->receiving a request\n for a list of usniversity

-->creating a list of usniversity

-->send a list of usniversity

-->(\*)

@enduml

Процес знаходження спеціальності

@startuml

title Find the specialty

(\*) -->receiving a request\n for a list of specialty

-->creating a list of specialty

-->send a list of specialty

-->(\*)

@enduml

Процес знаходження курсу

@startuml

title Find the course

(\*) -->receiving a request\n for a list of course

-->creating a list of course

-->send a list of course

-->(\*)

@enduml

Процес знаходження ім’я

@startuml

title Find the name

(\*) --> open "names" database

-->"find name"

if "names were found?" then

-->[0]"create negative respond"

-->(\*)

else

-->[1]"display the name\n in the ranking list"

-->(\*)

endif

@enduml

Процес знаходження групи

@startuml

title Find the group

(\*) --> open "group name" database

-->"find group name"

if "groups name were found?" then

-->[0]"create negative respond"

-->(\*)

else

-->[1]"display the group name\n in the ranking list"

-->(\*)

endif

@enduml