California State University, Northridge

Kevin Chaja
Awesome Company Inc ™

Computer Science 490
Senior Design
Project 1

Automated Classes

Developed By: Awesome Company Inc

Professor Chaja
Philip D. Kim
ID: 108508736

September 12, 2020

Idea

Automated Classes:

As students, it's easy to forget important dates like registration which can lead to adding unwanted professors or not graduating on time. With *Automated Classes* by $\mathcal{A}$wesome $\mathcal{C}$ompany $\mathcal{I}$nc ™ promptly register next semester. And most important, help students graduate on time!

Overview

Goals:

1. Simple and user friendly application to schedule classes.
2. Avoid anxiety and stress, while some students are waiting impatiently, most students like myself carelessly forget to add classes. But with *Automated Classes*, it can add classes while enjoying vacation or busy with life.
3. Making student lives better by helping them graduate on time, which returns a better economy and of course profit for $\mathcal{A}$wesome $\mathcal{C}$ompany $\mathcal{I}$nc ™!

Proposed Solution

Pros of Solution:

1. Selenium, a third party with range of automating tools, libraries, support multiple web browsers and multiple platforms (Linux, MacOS, Linux).
2. While Selenium is licensed under the Apache 2.0 license with the Software Freedom Conservancy as the copyright holder, which will result in quicker startup company.
3. Canvas LMS API, a REST API for accessing and modifying data externally from the main application into *Automated Classes*, another open source and secruity shouldn't be too much trouble as long as code is implemented correctly.

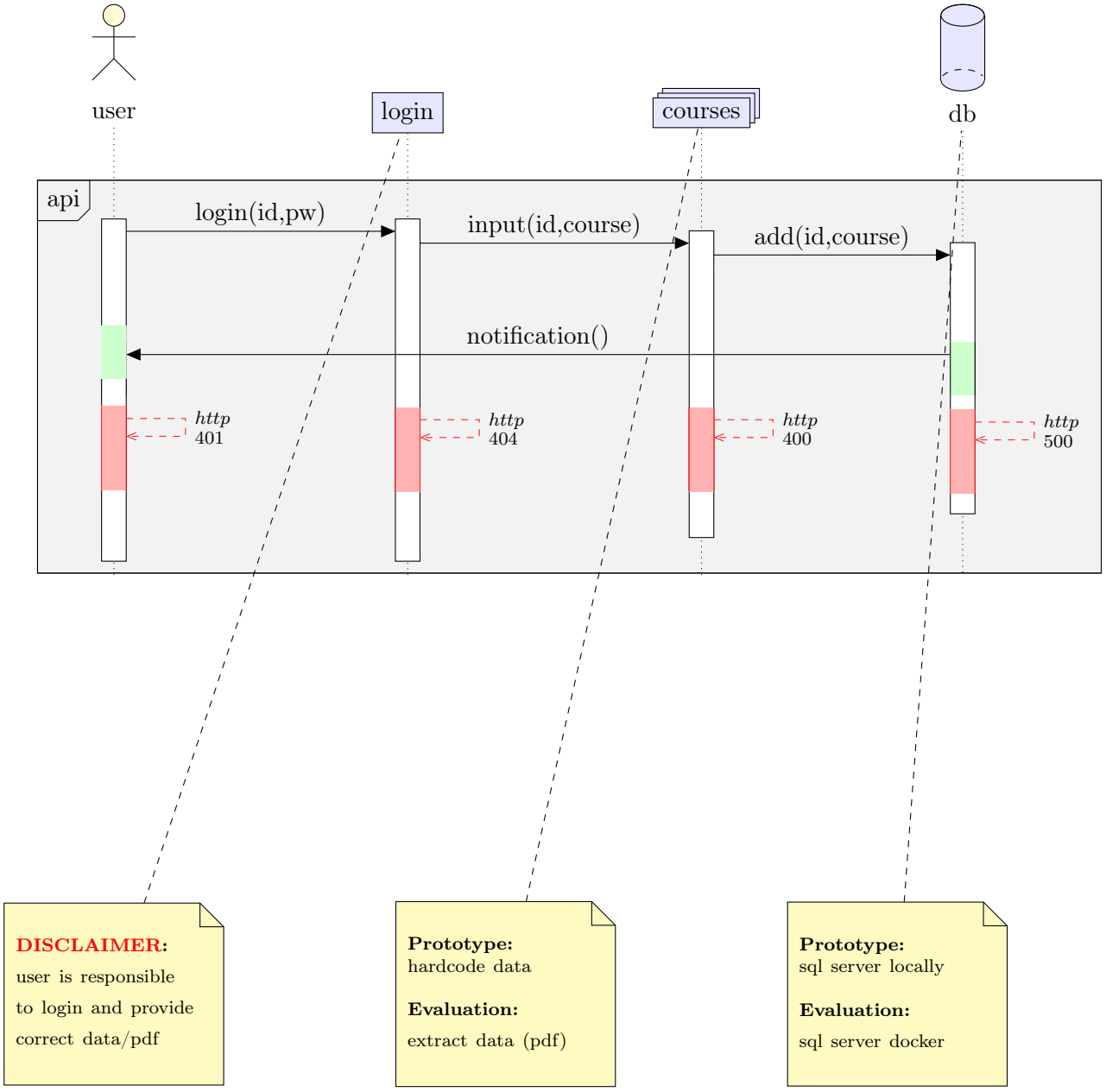
Proposed Solution

Cons of Solution:

1. Selenium, uses other third party software so getting a lawyer might be a good idea, however this will cost a lot of money.
2. Canvas LMS API, requires application submission, therefore cannot infringe the proprietary or intellectual property rights of any third party. And possible termination if code is implemented poorly or not up to their standard.
3. Secruity concerns may slow software development down in the future.

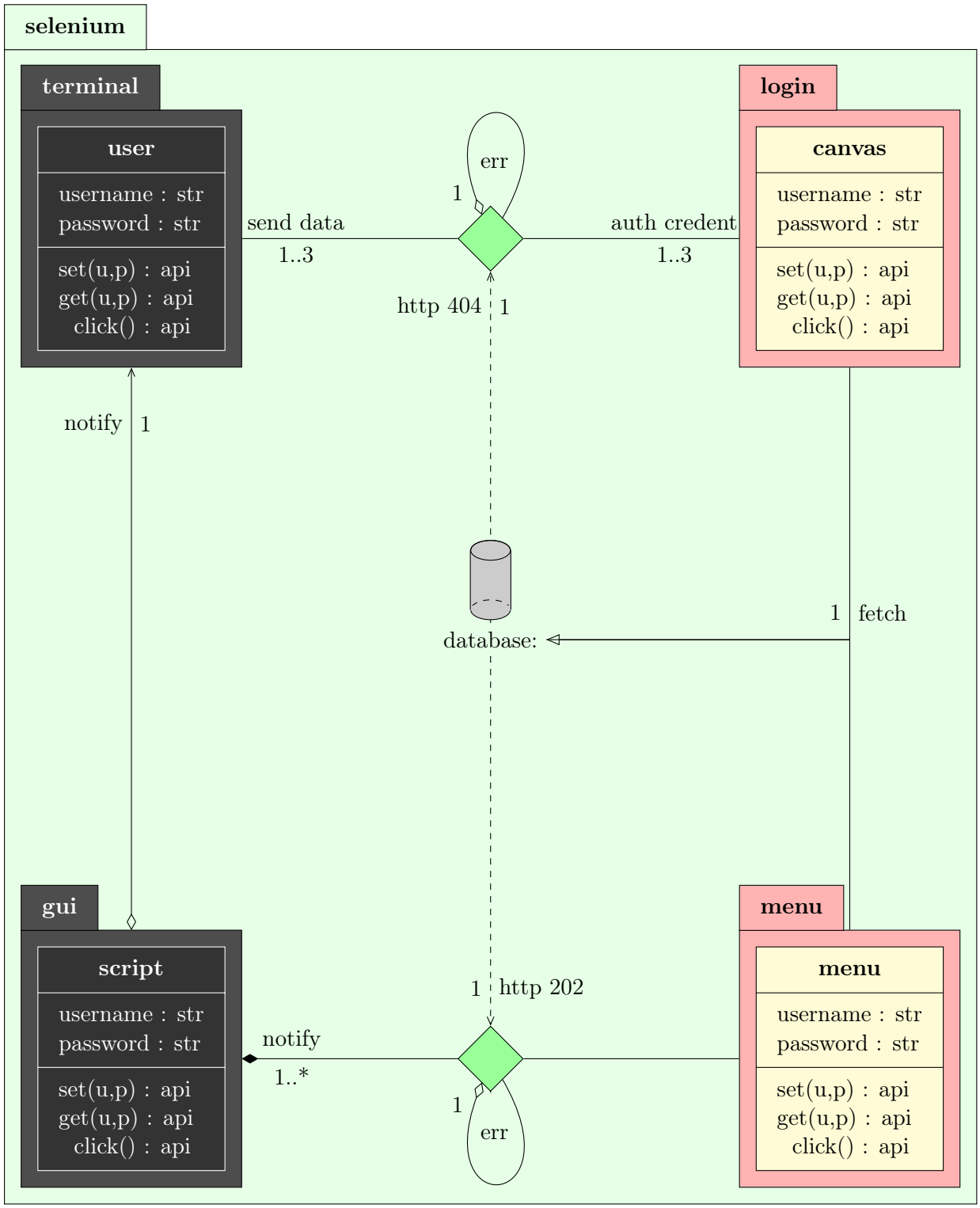
System Architecture

Top Level (User Case):



System Architecture

Low Level (Coding Design):



System Architecture

Technologies Used: `Python 3.7.7`, `Selenium`

```
#!/usr/local/bin/python3
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
import settings # import settings (credentials)
# Following pseudo-code will be similar for other api calls/implementations
#   1. find element name (i.e; <input name='username'>)
#   2. invoke webdriver methods (i.e; browser.find_element_by_name('username'))
#   3. click on a event (i.e; login.click())
canvas_url=settings.LINK # canvas url link in settings.py
username=settings.USER # login username in settings.py
password=settings.PASS # login password in settings.py
html_username="username" # html element <input name='username'>
html_password="password" # html element <input name='password'>
html_submit="submit" # html element <input name='submit'>
browser = webdriver.Safari() # selenium webdriver for safari browser
browser.get((canvas_url)) # 'Allow Remote Automation' in Safari (developer)
# try block to automate login python script
try:
  # find input username field in browser
  input_username = browser.find_element_by_name(html_username)
  # invoke send_keys with username credential to field
  input_username.send_keys(username)
  # find input password field in browser
  input_password = browser.find_element_by_name(html_password)
  # invoke send_keys with password credential to field
  input_password.send_keys(password)
  # find input submit button in browser
  login = browser.find_element_by_name(html_submit)
  # invoke click to submit login credentials
  login.click()
except Exception:
  print("LOGIN FAILED")
# How to run Python script:
# reference: https://www.selenium.dev/documentation/en/getting_started/
# 1. 'pip install selenium' & properly set path
# 2. Include correct credentials (URL,USER,PASS) in settings.py
# 3. Only MacOS, Safari browser -> Develop -> Allow Remote Automation
# 4. In terminal, $ python3 canvas.py
```