Software Engineering Lab
Implementation Report I - Build I

Automated Lab Program Evaluator

Ashwin Joisa : 16CO104
Praveen Raj : 16CO115

March 16, 2018

1. Basic Information

<Write information about programming languages used, APIs, Libraries, IDEs, any other tools and information on how to run your application>

Automated Lab Program Evaluator is a software for automation of evaluation process during lab exams and tests.

1. Basic Information about the Software.

- **Environment :-** Linux or Unix based Operating System.
- **Programming language :-** C++ (specifically C++11)
- **Libraries used :-**
  - bits/stdc++.h
  - unistd
  - signal.h
  - sys/wait.h
- **API used :-**
  - MySQL Connector/C++
- **IDE used :-**
  - Atom C++ IDE
- **Other Tools and Softwares used :-**
  - Sublime Text editor.
  - MySQL server
  - Git version control tool

2. Running the Software.

- **Method 1 :**
  - Go into the directory where the Software is stored.
  - Execute `./ALPE`
- **Method 2 :**
  - Go into the directory where the Software is stored.
  - Compile the file “ALPE.cpp” using the following command.

```
sudo g++ src/ALPE.cpp -o ALPE -Wall -I/usr/include/cppconn -L/usr/lib
-lmysqlcppconn -std=c++11
```

  - Execute `./ALPE`

- For more details, refer https://github.com/Aj163/Automated-Lab-Program-Evaluator/blob/master/README.md

2. Functional Requirements

FRID	NAME	Description
1	Registration	Admin is able to add new Student and Teacher accounts.
2	Login/Logout	Student, Teacher and Admin can Log into and Log out of their respective accounts and carry out their respective functions.
3	View Question	Student is able to view the questions.
4	Submit Solution	Student is able to submit his/her solution for a question as source code.
5	View Submissions	1. Student is able to view his submitted files. 2. Teacher is able to retrieve the submitted files of a particular student.
6	Add question	Teacher is able to add and delete Questions for the test/exam.
7	View Results	1. Teacher is able to view results of the whole class as a spreadsheet. 2. Student can view, if his/her solution satisfies all the test cases or not, after each submission.

3. Functional Requirements (Not Implemented)

FRID	NAME	Description
1	Delete Question	Teacher can delete an existing question

4. Screenshots

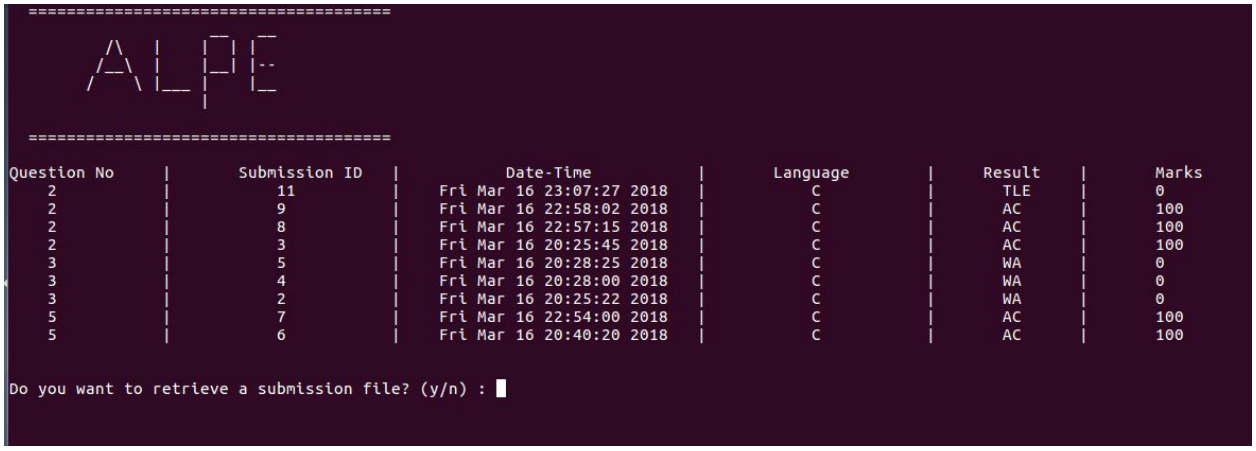
4.1. FRID:FR NAME

1. Login

2. Admin adding new Student Account

3. Admin deleting an existing Student's Account

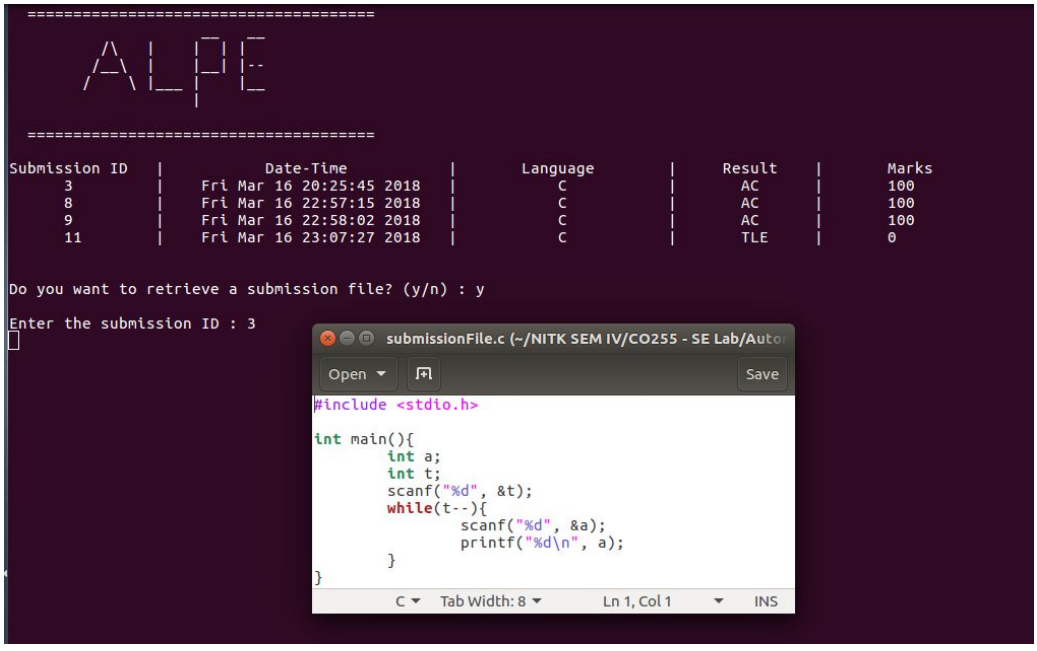
4. Teacher viewing all the submission made by a particular Student



5. Teacher adding a new Question

6. Student submitting his/her solution file

7. Student viewing his/her submissions for a particular a question.



5. Plan for Next Build / Release

- We have implemented the project according to Classical Waterfall Model, hence no improvements could be done on the requirements.
- In the next build, we plan to,
    - Improve the User Interface.
    - Fix the bugs present.
    - Add support for the submissions to be made in languages other than C.
- We will make a build script for building the project with the required dependencies.

6. Summary

This report contains:

1. Basic Information about the software, including the dependencies, and how to run the software.
2. Functional Requirements
3. Non-functional Requirements
4. Screenshots of the software
5. Plans for the next build