**[Nhóm 14]**



**Ước lượng dự án nguồn mở**

***Python Fire***

*[Link GitHub nguồn:* [*https://github.com/google/python-fire.git*](https://github.com/google/python-fire.git)*]*

MỤC LỤC

[1. Giới thiệu dự án 4](#_Toc93415451)

[1.1. Mô tả dự án 4](#_Toc93415452)

[1.2. Công cụ quản lý 4](#_Toc93415453)

[2. Các nhân sự tham gia dự án 4](#_Toc93415454)

[2.1. Thông tin liên hệ phía khách hàng 4](#_Toc93415455)

[2.2. Thông tin thành viên nhóm 4](#_Toc93415456)

[2.3. Phân chia vai trò của thành viên dự án và khách hàng 4](#_Toc93415457)

[3. Khảo sát dự án 5](#_Toc93415458)

[3.1. Tóm tắt về ứng dụng 5](#_Toc93415459)

[3.1.1. Ý nghĩa 5](#_Toc93415460)

[3.1.2. Tính năng cơ bản 5](#_Toc93415461)

[3.1.3. Từ khóa công nghệ 5](#_Toc93415462)

[3.2. Thống kê về mã nguồn 5](#_Toc93415463)

[3.3. Thống kê về hợp tác 7](#_Toc93415464)

[3.4. Kết quả chạy thử nghiệm 8](#_Toc93415465)

[3.4.1. Cài đặt 8](#_Toc93415466)

[3.4.2. Chạy thử nghiệm chương trình HelloWorld 8](#_Toc93415467)

[3.4.3. Lỗi gặp phải khi chạy thử 8](#_Toc93415468)

[3.5. Phạm vi dự án 8](#_Toc93415469)

[4. Ước lượng chung 9](#_Toc93415470)

[4.1. Ước lượng thời gian 9](#_Toc93415471)

[4.2. Ước lượng rủi ro 9](#_Toc93415472)

[4.2.1. Rủi ro #1 9](#_Toc93415473)

[4.2.2. Rủi ro #2 9](#_Toc93415474)

[4.2.3. Rủi ro #3 10](#_Toc93415475)

[4.2.4. Rủi ro #4 10](#_Toc93415476)

[4.2.5. Rủi ro #5 10](#_Toc93415477)

[5. Ước lượng giá thành 11](#_Toc93415478)

[5.1. Chi phí nguyên vật liệu 11](#_Toc93415479)

[5.2. Chi phí cơ sở vật chất 11](#_Toc93415480)

[5.3. Bảng tính lương thành viên 11](#_Toc93415481)

[5.4. Chi phí cho công việc 12](#_Toc93415482)

[5.5. Chi phí phát sinh 13](#_Toc93415483)

[5.6. Tổng chi phí cho dự án 13](#_Toc93415484)

[6. Ước lượng chất lượng 13](#_Toc93415485)

[6.1. Các tiêu chí đánh giá chất lượng 13](#_Toc93415486)

[6.2. Các hình thức kiểm thử có thể dùng (ước lượng testcase) 14](#_Toc93415487)

[7. Đóng dự án 14](#_Toc93415488)

[7.1. Quản lý mã nguồn 14](#_Toc93415489)

[7.2. Quản lý công việc 15](#_Toc93415490)

[8. Các tài liệu tham khảo 16](#_Toc93415491)

Phiên bản tài liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ngày lập | Mô tả thay đổi | Phiên bản | Người lập | Người duyệt |
| 29/12/2021 | Fork project về và thêm collaborator | 0.1 | Hoàng Nguyễn | Hoàng Nguyễn |
| 02/01/2022 | Sửa các lỗi nghiêm trọng | 0.1.1 | Hoàng Trần | Hoàng Nguyễn |
| 08/01/2022 | Thêm tính năng tạo CLI cho class | 0.2 | Khánh Nam | Hoàng Nguyễn |
| 10/01/2022 | Sửa các lỗi liên quan đến CLI cho class | 0.2.1 | Cả team | Hoàng Nguyễn |

# Giới thiệu dự án

## Mô tả dự án

Python Fire là một thư viện có chức năng tự động tạo giao diện dòng lệnh (CLI) từ bất kỳ Python object nào.

## Công cụ quản lý

**Link Quản lý mã nguồn:** GitHub

<https://github.com/nvhoang55/python-fire.git>

# Các nhân sự tham gia dự án

## Thông tin liên hệ phía khách hàng

1. Anh Elon Musk: kỹ sư, nhà tài phiệt, nhà phát minh, doanh nhân công nghệ và nhà từ thiện.
2. Ông Jeff Bezos: doanh nhân, nhà tư bản công nghiệp, chủ sở hữu truyền thông và nhà đầu tư.
3. Anh Mark Zuckerberg: nhà lập trình máy tính người Mỹ kiêm doanh nhân mảng công nghệ Internet.

## Thông tin thành viên nhóm

|  |  |
| --- | --- |
| Họ và tên | MSSV |
| Nguyễn Việt Hoàng | 20176020 |
| Trần Lê Hoàng | 20176023 |
| Phạm Khánh Nam | 20148980 |

## Phân chia vai trò của thành viên dự án và khách hàng

|  |  |  |
| --- | --- | --- |
| Họ và tên | Vai trò | Nhiệm vụ |
| Nguyễn Việt Hoàng | Nhóm trưởng | Phân chia công việc  Khảo sát dự án: tóm tắt về ứng dụng, thống kê mã nguồn, kết quả chạy thử nghiệm, phạm vi dự án |
| Trần Lê Hoàng | Thành viên | Ước lượng chung: thời gian, rủi ro  Đóng dự án |
| Phạm Khánh Nam | Thành viên | Ước lượng: giá thành, chất lượng  Thống kê tài liệu liên quan |

# Khảo sát dự án

## Tóm tắt về ứng dụng

### Ý nghĩa

* Python / Google Fire là một thư viện mã nguồn mở được phát triển và duy trì bởi Google và nó được sử dụng phổ biến nhất để tạo giao diện dòng lệnh từ mã Python.
* Đây đồng thời là một module có thể chuyển đổi mã Python thành giao diện dòng lệnh chỉ với một lệnh gọi. Nó là một thư viện có thể biến bất kỳ thành phần Python nào thành giao diện dòng lệnh một cách dễ dàng.
* Python Fire cũng có thể chuyển đổi một module thành một giao diện dòng lệnh mà không cần biết mã nguồn của nó.
* Nó được gọi là Fire vì khi bạn gọi hàm fire(…), nó sẽ “fire off” (thực hiện) lệnh của bạn.

### Tính năng cơ bản

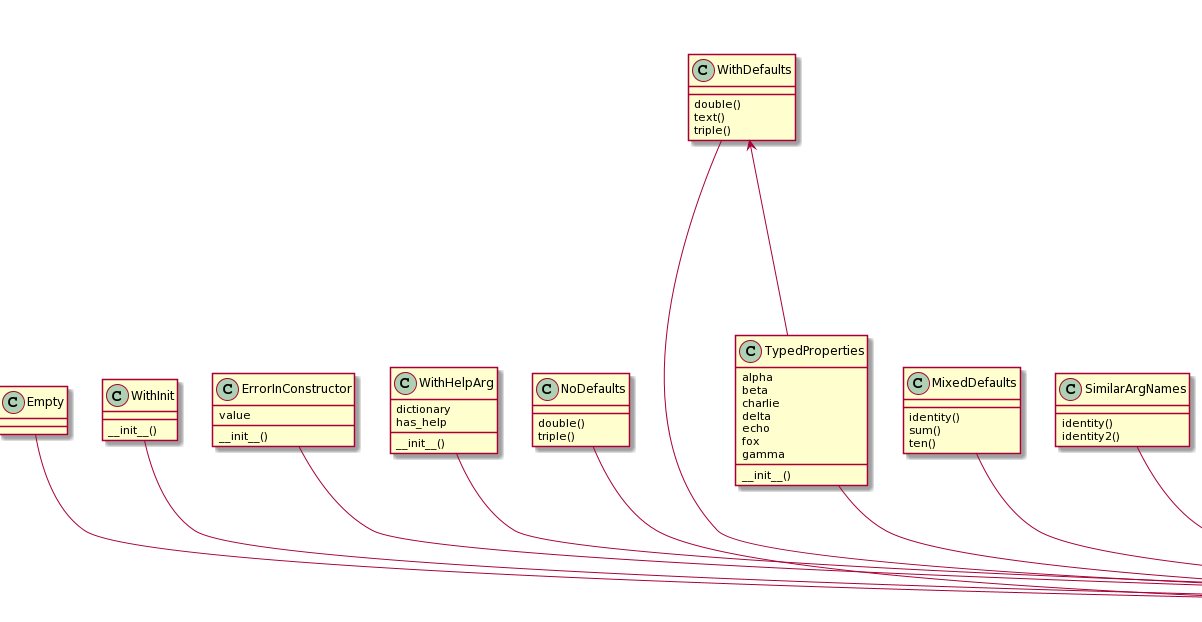
* Python Fire giúp tạo CLI bằng Python một cách đơn giản.
* Python Fire hữu ích trong việc phát triển và debug mã Python.
* Python Fire giúp khám phá mã hiện có hoặc biến mã của người khác thành CLI.
* Python Fire giúp chuyển đổi giữa Bash và Python dễ dàng hơn.
* Python Fire làm cho việc sử dụng Python REPL dễ dàng hơn bằng cách thiết lập REPL với các module và các biến đã import và khởi tạo.

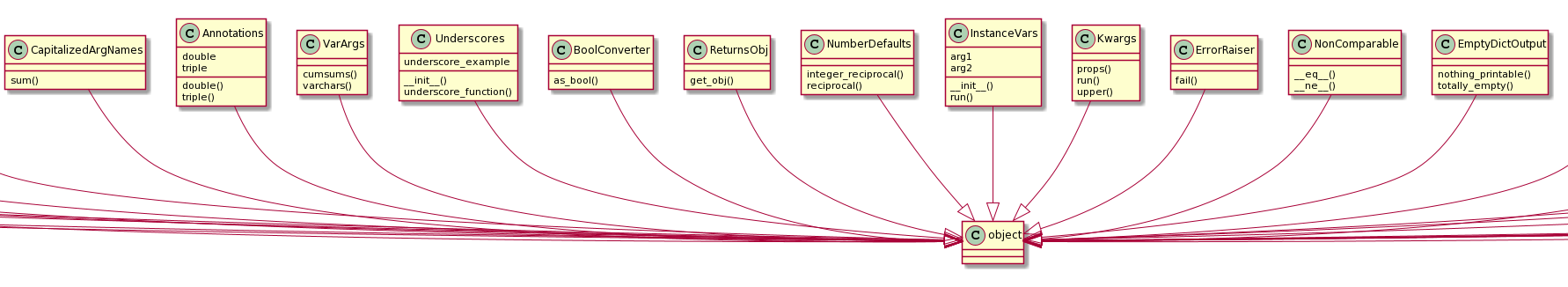
### Từ khóa công nghệ

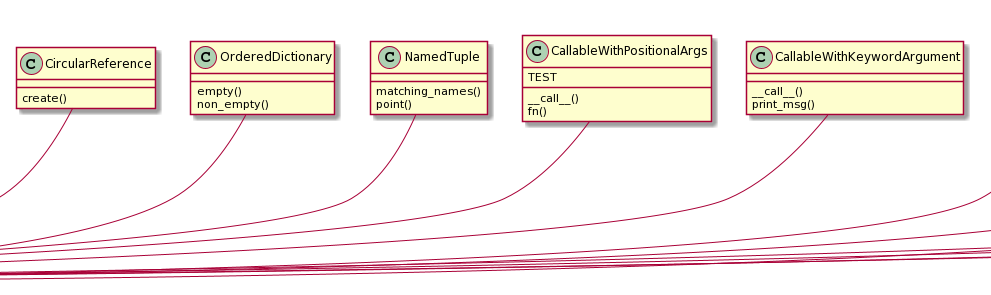
Python, Shell, CLI, Python REPL, Bash, …

## Thống kê về mã nguồn

* Tool hỗ trợ phân tích thông số mã nguồn: [SonarQube](https://docs.sonarqube.org/latest/)
* Số lượng file:
  + Số lượng dòng code: 6,527
  + Số lượng dòng chú thích: 3,413
  + Độ phức tạp của file:
    - [[Cyclomatic Complexity](https://en.wikipedia.org/wiki/Cyclomatic_complexity)](https://docs.codeclimate.com/docs/cyclomatic-complexity): 1,365
    - [Cognitive Complexity](https://docs.codeclimate.com/docs/cognitive-complexity): 1,193
* Số lượng hàm: 681
* Số lượng class: 111,
* Sơ đồ phân cấp class:







Hình 1. Sơ đồ phân cấp Class

Ảnh có chứa bàn

Mô tả được tạo tự động

Hình . Thống kê mã nguồn từ SonarQube

## Thống kê về hợp tác

* Số lượng collaborator tham dự (của dự án gốc): 45
* Số lượt commit: 259
* Số branch: 5
  + Thông tin về 5 người tham gia dự án nguồn mở với số commit nhiều nhất

|  |  |  |  |
| --- | --- | --- | --- |
| Tên đầy đủ | Số commit | Link tài khoản GitHub | Số repository hiện có |
| David Bieber | 165 | [dbieber (David Bieber) (github.com)](https://github.com/dbieber) | 65 |
| Joe Chen | 28 | [joejoevictor (Joe Chen) (github.com)](https://github.com/joejoevictor) | 9 |
| Jacob Austin | 6 | [jacobaustin123 (Jacob Austin) (github.com)](https://github.com/jacobaustin123) | 37 |
| Saurabh Patel | 4 | [saurabhkpatel (Saurabh Patel) (github.com)](https://github.com/saurabhkpatel) | 34 |
| Jeff Tratner | 4 | [jtratner (Jeff Tratner) (github.com)](https://github.com/jtratner) | 85 |

## Kết quả chạy thử nghiệm

### Cài đặt

Cài đặt thông qua pip:

pip install fire

### Chạy thử nghiệm chương trình HelloWorld

Tạo file example.py với nội dung như sau:

import fire

def hello(name):

return 'Hello {name}!'.format(name=name)

if \_\_name\_\_ == '\_\_main\_\_':

fire.Fire()

Sau đó chạy trong terminal:

$ python example.py hello World

Hello World!

### Lỗi gặp phải khi chạy thử

Python Fire gặp lỗi khi pass argument đính kèm dấu #:

*# test.py*

import fire

def test(a): print(a, type(a))

fire.Fire(test)

Kết quả:

> python test.py hi#there

**hi <class 'str'>**

> python test.py "hi#there"

**hi <class 'str'>**

Cách khắc phục (workaround): bọc tham số bởi 1 cặp ngoặc đơn nữa:

> python test.py **'**"hi#there"**'**

## Phạm vi dự án

Chạy được trên nền tảng, OS: Windows, macOS, và Linux

# Ước lượng chung

## Ước lượng thời gian

|  |  |  |
| --- | --- | --- |
| Mục tiêu | Thời gian | Mô tả |
| Cài đặt | ~ 5 phút | Cài thành công mã nguồn mở (trước tiên cần phải cài pip hoặc conda làm môi trường để cài python fire). |
| Hiểu rõ | ~ 2 tuần – 1 tháng | Đọc docs và code dựa vào ví dụ của mã nguồn. |
| Thêm tính năng | ~ 2-3 tháng | Lên ý tưởng cho tính năng mới.  Thiết lập kế hoạch phát triển – kiểm thử - publish). |
| ~ 3-5 ngày | Chỉnh sửa tính năng cần (phân tích lỗi). |

## Ước lượng rủi ro

### Rủi ro #1

* Tên rủi ro: Rủi ro về nhân lực thiếu kinh nghiệm
* Mô tả rủi ro: Thành viên phụ trách thiếu hiểu biết về nghiệp vụ hoặc không có kinh nghiệm về công nghệ đang được sử dụng trong dự án.
* Xác suất xảy ra: 100%
* Mức độ thiệt hại: Rất lớn (>80%) - Tốn kén ngân sách và thời gian
* Giải pháp xử lý: Trong trường hợp không có thành viên đáp ứng được điều kiện để trao đổi task thì đề xuất với khách hàng dời deadline cho thành viên phụ trách công việc có thể vừa tìm hiểu và vừa thực hiện (Tăng khoảng 10-20% thời gian ban đầu)

### Rủi ro #2

* Tên rủi ro: Kiểm định mức đơn vị nghèo nàn
* Mô tả rủi ro: Kiểm định mức đơn vị do lập trình viên thực hiện trước khi bàn giao phần mềm để tích hợp và kiểm định mức hệ thống. Công việc này đòi hỏi thời gian, nếu không giám sát chặt chẽ thường dẫn đến việc bị bỏ qua hoặc làm chiếu lệ.
* Xác suất xảy ra: 90%
* Mức độ thiệt hại: Rất lớn (>80%) – Rủi ro này gây ra những lỗi phần mềm tiềm ẩn rất khó để phát hiện và chỉnh sửa khi phần mềm đi vào hoạt động, hoặc nếu chỉnh sửa sẽ tốn rất nhiều công sức và thời gian
* Giải pháp xử lý:
* Cần có những lập trình viên giỏi kiểm tra phần mềm để xem thử phần mềm có tuân thủ theo đúng những tiêu chuẩn của một phần mềm hay không. Đảm bảo các yêu cầu khác như tính an toàn, tính ổn định.
* Khâu thiết kế phần mềm phải đảm bảo đầy đủ các yêu cầu, tiêu chuẩn mà khách hàng đề ra.
* Quá trình kiểm thử phải được giám sát chặt chẽ, không được làm qua loa, đại khái, trang bị các phương tiện kiểm tra hiện đại để quá trình kiểm tra đạt kết quả cao. Đơn giản nhất thì mỗi hàm cho 1 unit test – giảm được chi phí của quá trình.

### Rủi ro #3

* Tên rủi ro: Thành viên nghỉ phép
* Mô tả rủi ro: Thành viên trong team xin phép nghỉ vì lý do cá nhân (bệnh tật, xử lý chuyện gia đình …) trong khi các phần công việc được giao chưa hoàn thành.
* Xác suất xảy ra: 10%
* Mức độ thiệt hại: Nhỏ (<30%)
* Giải pháp xử lý: Bàn giao các phần công việc cho thành viên khác trong team trước ngày nghỉ. Các thành viên nhận việc rồi đánh giá lại thời gian dự kiến hoàn thành task.

### Rủi ro #4

* Tên rủi ro: Trùng lịch với thời gian thi trên trường
* Mô tả rủi ro: Thời gian thực hiện task trùng với lịch thi trên trường
* Xác suất xảy ra: 50%
* Mức độ thiệt hại: Trung bình (30-50%)
* Giải pháp xử lý: Đề xuất với khách hàng tăng thời gian thực hiện dự án lên bằng số tuần thi trên trường. Trong trường hợp khách hàng không đồng ý thì từ chối dự án.

### Rủi ro #5

* Tên rủi ro: Trùng tiêu đề bug, các tiêu đề bug có chất lượng kém
* Mô tả rủi ro: Các báo cáo bug được diễn đạt kém hoặc bị trùng lặp dẫn đến hiệu suất công việc giảm, mất nhiều thời gian của bên dev và test.
* Xác suất xảy ra: 20%
* Mức độ thiệt hại: Lớn (50-80%)
* Giải pháp xử lý: Đặt tiêu đề bug riêng biệt
  + Đặt tên theo một công thức.
  + Ngôn ngữ dùng để báo cáo.
  + Kết hợp tiêu đề bug.
  + Từ ngữ dùng để diễn tả.

# Ước lượng giá thành

Dự án có quy mô một nhóm nhỏ gồm 3 sinh viên tại Đại Học Bách Khoa Hà Nội thực hiện.

## Chi phí nguyên vật liệu

|  |  |  |  |
| --- | --- | --- | --- |
| Các hạng mục | Số lượng | Đơn giá | Thành tiền |
| Máy chủ CSDL | 1 | 3.000.000đ | 30.000.000đ |
| Chi phí đăng ký bản quyền | 1 | 1.000.000đ | 1.000.000đ |
| Tổng chi phí | | | **31.000.000đ** |

## Chi phí cơ sở vật chất

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Các hạng mục | Số lượng | Đơn giá | Thời gian | Thành tiền |
| Thuê văn phòng | 1 | 3.000.000đ/tháng | 5 | 15.000.000đ |
| Tiền mạng internet | 1 | 500.000đ/tháng | 5 | 2.500.000đ |
| Tiền điện, nước | 1 | 500.00đ/tháng | 5 | 2.500.000đ |
| Dây mạng | 100m | 10.000đ/m |  | 1.000.000đ |
| Switch 8 cổng | 1 | 500.000đ |  | 500.000đ |
| Tổng chi phí: | | | | 21.500.00đ |

## Bảng tính lương thành viên

* Lương thành viên được tính theo đơn vị ngày với thời lượng 8h/ngày
* Năng lực thành viên trong nhóm có thể khác nhau những do quy mô dự án nhỏ và trong quá trình làm việc các thành viên có nghĩa vụ giúp đỡ nhau để hoàn thành công việc nên mức lương các thành viên trong nhóm là như nhau

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ tên | Vị trí | Lương (VND/ngày) |
| 1 | Nguyễn Việt Hoàng | Nhóm trưởng | 500.000đ |
| 2 | Trần Lê Hoàng | Thành viên | 500.000đ |
| 3 | Phạm Khánh Nam | Thành viên | 500.000đ |

## Chi phí cho công việc

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Mã WBS | Giai đoạn | Tên công việc | | EST cuối cùng | Số người tham gia | Tiền lương (Giá trị gần đúng) |
| 1.0 | 1 | **Khảo sát** | | **7.56** |  | **10.260.000đ** |
| 1.1 | 1 | Gặp gỡ khách hàng | 2.16 | 2 | 2.160.000đ |
| 1.2 | 2 | Xác định yêu cầu | 3.24 | 3 | 4.860.000đ |
| 1.3 | 3 | Kiểm tra yêu cầu | 1.08 | 3 | 1.620.000đ |
| 1.4 | 4 | Báo cáo | 1.08 | 3 | 1.620.000đ |
| 2.0 | 2 | **Giai đoạn phân tích** | | **27** |  | **24.375.000đ** |
| 2.1 | 1 | Tổng hợp yêu cầu | 7.64 | 3 | 11.460.000đ |
| 2.2 | 2 | Lập bảng chi tiết công việc | 7.56 | 1 | 3.780.000đ |
| 2.3 | 3 | Xây dựng Use case | 2.16 | 3 | 3.240.000đ |
| 2.4 | 4 | Lập kế hoạch thực hiện | 6.48 | 1 | 3.240.000đ |
| 2.5 | 5 | Phân công việc | 2.07 | 1 | 1.035.000đ |
| 2.6 | 6 | Báo cáo | 1.08 | 3 | 1.620.000đ |
| 3.0 | 3 | **Giai đoạn thiết kế** | | **17.28** |  | **25.920.000đ** |
| 3.1 | 1 | Thiết kế phần mềm | 16.2 | 3 | 24.300.000đ |
| 3.2 | 2 | Báo cáo | 1.08 | 3 | 1.620.000đ |
| 4.0 | 4 | **Giai đoạn xây dựng** | | **55.61** |  | **83.415.000đ** |
| 4.1 | 1 | Xây dựng lớp sơ đồ | 10.98 | 3 | 16.470.000đ |
| 4.2 | 2 | Xây dựng cơ sở dữ liệu | 11.15 | 3 | 16.725.000đ |
| 4.3 | 3 | Xây dựng các chức năng | 32.4 | 3 | 48.600.000đ |
| 4.4 | 4 | Báo cáo | 1.08 | 3 | 1.620.000đ |
| 5.0 | 5 | **Giai đoạn chạy thử** | | **7.83** |  | **11.740.000đ** |
| 5.1 | 1 | Kiểm tra | 2.16 | 3 | 3.240.000đ |
| 5.2 | 2 | Sửa lỗi phát sinh | 3.42 | 3 | 5.130.000đ |
| 5.3 | 3 | Cập nhật lại phần mềm | 1.169 | 3 | 1.750.000đ |
| 5.4 | 4 | Báo cáo | 1.08 | 3 | 1.620.000đ |
| 6.0 | 6 | **Giai đoạn kiểm thử** | | **7.56** |  | **11.340.000đ** |
| 6.1 | 1 | Test module | 2.16 | 3 | 3.240.000đ |
| 6.2 | 2 | Test tích hợp | 3.24 | 3 | 4.860.000đ |
| 6.3 | 3 | Đưa vào hoạt động | 1.08 | 3 | 1.620.000đ |
| 6.4 | 4 | Báo cáo | 1.08 | 3 | 1.620.000đ |
| 7.0 | 7 | **Giai đoạn chuyển giao** | | **7.91** |  | **6.110.000đ** |
| 7.1 | 1 | Viết tài liệu hướng dẫn | 1.08 | 3 | 1.620.000đ |
| 7.2 | 2 | Bàn giao sản phẩm | 2.51 | 1 | 1.255.000đ |
| 7.3 | 3 | Đào tạo sử dụng | 1.97 | 1 | 985.000đ |
| 7.4 | 4 | Xin xác nhận, chữ ký của bên khách hàng | 1.26 | 1 | 630.000đ |
| 7.5 | 5 | Báo cáo kết thúc dự án | 1.08 | 3 | 1.620.000đ |
| Tổng chi phí | | | | | | 173.150.000đ |

## Chi phí phát sinh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Các hạng mục | Số lượng | Đơn giá | Thành tiền |
| 1 | Chi phí đi lại gặp gỡ khách hàng | 2 người x 2 ngày | 200.000đ/người/ngày | 800.000d |
| 2 | Chi phí liên hoan | 2 bữa | 3.000.000đ/bữa | 6.000.000đ |
| 3 | Chi phí đi lại cài đặt dự án | 3 người x 1 buổi | 200.000đ/người/buổi | 600.000đ |
| 4 | Chi phí liên lạc trao đổi với khách hàng |  | 1.000.000đ | 1.000.000đ |
| Tổng chi phí : | | | | 8.400.000đ |

## Tổng chi phí cho dự án

|  |  |  |
| --- | --- | --- |
| STT | Các hạng mục | Thành tiền |
| 1 | Chi phí nguyên vật liệu | 31.000.000đ |
| 2 | Chi phí cơ sở vật chất | 21.500.000đ |
| 3 | Chi phí trả lương cho các thành viên dự án | 173.150.000đ |
| 4 | Chi phí phát sinh khác | 8.400.000đ |
| Tổng chi phí : | | 234.050.000đ |

# Ước lượng chất lượng

Mục đích của ước lượng chất lượng:

* Xác định gần đúng chất lượng sản phẩm thỏa mãn được khách hàng không
* Giám sát đánh giá, kịp thời phát hiện sai sót từ đó có kế hoạch khắc phục
* Được tiến hành ở từng giai đoạn của dự án

## Các tiêu chí đánh giá chất lượng

* Tính ổn định của ứng dụng: Ứng dụng chạy ổn định, ko bị lỗi khi hệ thống và phần mềm xảy ra một số thay đổi. Chương trình gọn nhẹ, ổn định và có khả năng tương thích với nhiều nền tảng hệ điều hành khác nhau. Ứng dụng có khả năng được bảo trì, sửa đổi một cách dễ dàng.
* Tính khả dụng: Đảm bảo sự đầy đủ và dễ dàng cho người dùng. Đơn giản và dễ sử dụng với người dùng. Các chức năng phải đáp ứng đủ được các yêu cầu của khách hàng
* Tính toàn vẹn của dữ liệu: Đảm bảo dữ liệu đồng bộ, dữ liệu không bị thay đổi hay mất mát, đặc biệt trong xử lý các tác vụ. Dữ liệu phải đảm bảo được sự toàn vẹn khi được người dùng tiến hành cập nhật
  + Ước lượng số dòng code: 6.5 KLOC

## Các hình thức kiểm thử có thể dùng (ước lượng testcase)

* Kiểm thử chức năng định kì : Các chức năng sẽ được kiểm tra định kỳ để đảm bảo thư viện trơn tru và không phát sinh lỗi trong quá trình hoạt động.
* Kiểm tra code: Lập trình viên có kinh nghiệm kiểm tra code đột xuất.
* Kiểm thử bởi khách hàng: Đưa mẫu thiết kế cho khách hàng để thăm dò phản ứng và lấy ý kiến của khách hàng về cách sử dụng ứng dụng. Kiểm tra các chức năng của ứng dụng xem đã phù hợp với các yêu cầu của khách hàng hay không dựa trên các giả thiết từ phía khách hàng và tập dữ liệu mẫu.
* Kiểm tra tính toàn vẹn của dữ liệu: Dữ liệu của ứng dụng sẽ được kiểm tra tính toàn vẹn dựa theo tập hợp những dữ liệu mẫu.

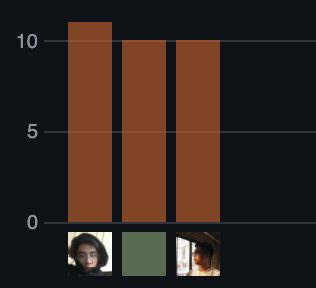
# Đóng dự án

Thực hiện các thống kê

## Quản lý mã nguồn

1. Thống kê về commit

|  |  |
| --- | --- |
| Người commit | Lượng commit |
| Trần Lê Hoàng | 11 |
| Nguyễn Việt Hoàng | 10 |
| Phạm Khánh Nam | 10 |
| Bot | 2 |
| Tổng | 33 |



Hình . Biểu đồ lượng commit của thành viên nhóm

1. Phân bố commit của dự án (sáng chiều đêm…):

Commit sau 9h tối hằng ngày.

Merge code: 10h tối.

1. Số dòng lệnh bị thay đổi: 0
2. Số lượng các branch được tạo ra: 6
3. Số dòng lệnh của dự án: 6600 dòng

## Quản lý công việc

1. Quản lý tasks:

Đã hoàn thành: 10

Chưa hoàn thành: 0

Chậm tiến độ: 0

1. Bố trí task theo Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Ngày giao | Ngày hoàn thành | Người nhận task |
| Fork project | 28.12.2021 | 28.12.2021 | Nguyễn Việt Hoàng |
| Clone project về và đọc qua dự án | 29.12.2021 | 29.12.2021 | Nguyễn Việt Hoàng |
| Phạm Khánh Nam |
| Trần Lê Hoàng |
| Chạy thử project | 30.12.2021 | 30.12.2021 | Nguyễn Việt Hoàng |
| Trần Lê Hoàng |
| Phạm Khánh Nam |
| Thực hiện >= 10 commit | 01.12.2021 | 01.12.2022 | Nguyễn Việt Hoàng |
| 03.12.2022 | Phạm Khánh Nam |
| 04.12.2022 | Trần Lê Hoàng |
| Phân tích và tính các số liệu về mã nguồn | 04.01.2022 | 04.01.2022 | Nguyễn Việt Hoàng |
| Vẽ class diagram cho mã nguồn | 05.01.2022 | 05.01.2022 | Nguyễn Việt Hoàng |
| Tính toán các chi phí của dự án | 07.01.2022 | 10.01.2022 | Phạm Khánh Nam |
| Ước lượng chất lượng dự án | 12.01.2022 | 12.01.2022 | Phạm Khánh Nam |
| Tính toán rủi ro của dự án | 11.01.2022 | 17.01.2022 | Trần Lê Hoàng |
| Thống kê đóng dự án | 14.01.2022 | 15.01.2022 | Trần Lê Hoàng |

# Các tài liệu tham khảo

* Google Python Fire: [Link](https://github.com/google/python-fire)
* Explore Python Fire Libraries : [Link](https://husteduvnhttps:/www.pluralsight.com/guides/explore-python-libraries:-python-fire.sharepoint.com/sites/BiTpLn-2021-1/Shared%20Documents/Quản%20trị%20dự%20án%20CNTT/DS%20nhóm.xlsx)
* Python Fire Module: [Link](https://www.geeksforgeeks.org/python-fire-module/)
* A Simple way for automatically creating CLI form any Python Project: [Link](https://www.geeksforgeeks.org/python-fire-module/)
* Introducing Python Fire: [Link](https://opensource.googleblog.com/2017/03/python-fire-command-line.html)