DevDuo: Pair Programming Matcher Tool

Epic Fellows: Mason Gelletly, Caroline Joseph, Quinn Anderson, Tatiana Monteiro

Problem Statement

It is very difficult to manually pair two engineers to work towards a solution

- Pair programming is **powerful**, and not utilized enough in industry
- There is **no efficient, automated solution** for pairing up teammates

Our Solution

An automated tool to **promote effective collaboration**

- Leverages engineer’s skills, need, areas of expertise, and availability, for efficient peer programming matchups

- Strong emphasis on **community and growth**

Related Work

Formation

Google’s CS Research Mentorship Program

Concepts Used

Understanding **Software Processes** helped us form a structured and effective approach

Top 3 reasons for project failure:

Project Challenged Factors	% of Responses
1. Lack of User Input	12.8%
2. Incomplete Requirements & Specifications	12.3%
3. Changing Requirements & Specifications	11.8%

hurdles and plan accordingly

Concepts Used

Requirements Analysis helped us understand potential user needs

- Assisted us in limiting misunderstandings and confusion later in development process
- Writing use cases helped to clarify what goals a user may have when using DevDuo

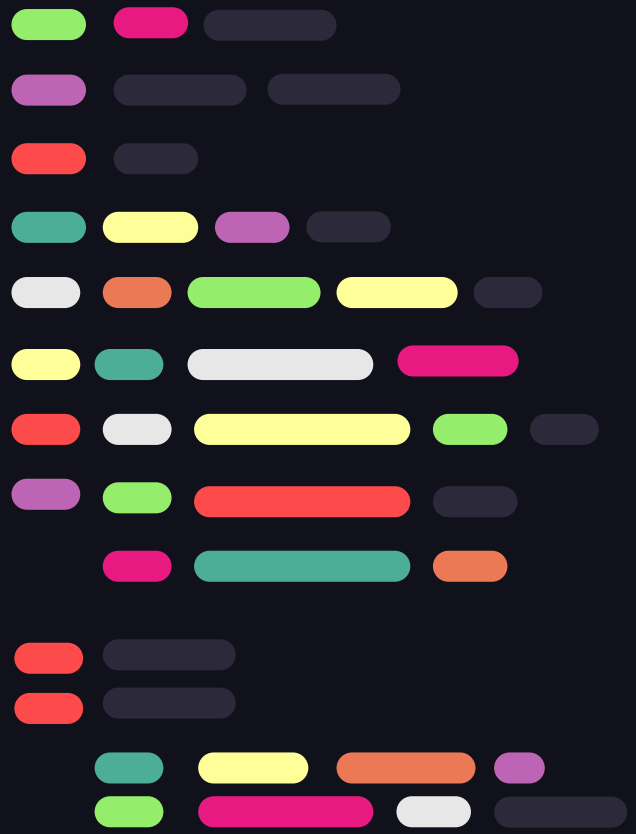
Concepts Used

Low Level Design helped us understand the **how** of DevDuo

- Understanding design patterns allowed us leverage tried and true methods of design
- Helped us understand how individual components work together in software

Future Work

- **Advanced matching algorithm** that leverages several more aspects of user data
- **Reviewing system** that could ward users away from negative interactions
- **Built-in IDE** support



Demo! }

*

Visual Representation

- user.py
- driver.py

n Terminal Help quinnanderson [SSH: rlogin.cs.vt.edu]

user.py driver.py

cs2104 > cs3704 > user.py > User

```python
#user class to store information about the user
class User:
    def __init__(self, name, experience, availability, project, now):
        self._name = name #name of user
        self._experience = experience #years of experience as an int
        self._availability = availability #availaibility as string ie MW or MTuTh etc
        self._project = project #project they are working on / department ie, we will match people in the same places
        self._now = now #true if they need a match now, false if they want to wait for matches in pool

    # getters
    def get_name(self):
        return self._name

    def get_experience(self):
        return self._experience

    def get_availability(self):
        return self._availability

    def get_project(self):
        return self._project

    def get_now(self):
        return self._now

    # setters
    def set_name(self, name):
        self._name = name

    def set_experience(self, experience):
        self._experience = experience

    def set_availability(self, availability):
        self._availability = availability

    def set_project(self, project):
        self._project = project

    def set_now(self, now):
        self._now = now
```

Terminal Help quinnanderson [SSH: rlogin.cs.vt.edu]

user.py driver.py

cs2104 > cs3704 > driver.py > ...

```python
#import our matcher
from user import User

#this is where we will store our pool of workers
pool = []

#in a real application we would have it be empty to start, and users would add to the pool over time
#in this scenario to demonstrate use cases we have filled it with multiple users

#adding users to our pool
user1 = User("Alice", 1, "MW", "Project A", False)
user2 = User("Bob", 8, "MTuTh", "Project B", False)
user3 = User("Charlie", 2, "MWTh", "Project C", False)
user4 = User("David", 5, "TuTh", "Project D", False)
user5 = User("Eva", 4, "MW", "Project A", False)
user6 = User("Jose", 28, "MTuTh", "Project B", False)
user7 = User("Grace", 7, "MWTh", "Project C", False)
user8 = User("Mia", 3, "TuTh", "Project D", False)
user9 = User("Ivy", 10, "MW", "Project A", False)
user10 = User("Jack", 1, "MTuTh", "Project B", False)
#adding them to list
pool = [user1, user2, user3, user4, user5, user6, user7, user8, user9, user10]

#function to create a profile from user input
def create_profile():
    name = input("Name: ")
    experience = int(input("Years of experience: "))
    availability = input("Enter availability (Days ie 'MW', 'TuThF', etc): ")
    project = input("Enter project: ")


    #creating a user profile
    return User(name, experience, availability, project, False)

#main 'runner' of our program, while loop to keep asking for users to input
#this is a prototype, but this would be run with the front end and instead of a while loop it would update when
#a user create a profile etc. it would just be a function to match and we wouldnt need the while

while True:
    add_user = input("Would you like to create a profile to match? (y/n): ").lower()

    if add_user == "n":
        print("Thank you for using the Pair Programming Matcher Tool!")
        break
    elif add_user == "y":
```

Terminal Help quinnanderson [SSH: rlogin.cs.vt.edu]

user.py driver.py

cs2104 > cs3704 > driver.py > ...

```python
        break
    elif add_user == "y":
        new_user = create_profile()
    else:
        print("Invalid response, please enter 'y or 'n")

    #after creating profile we want to see what the user wants out of this matcher
    #notice that availibility is important as we cannot match people who are not available on the same days

    #checking if the user wants to be added to pool, or to get a match immediately (if possible)
    check = input("Do you want to immediately find a match? (y/n): ")
    if check == "n":
        print("You have successfully been added to pool!")
        new_user.set_now(False)
        #add use to pool
        pool.append(new_user)
    elif check == "y":
        new_user.set_now(True)
        print("You are looking for a match: Choose one of the number options below to describe your situation")
        print("(1) Match with a more experienced programmer for mentorship")
        print("(2) Match with a programmer with similar levels of experience")
        print("(3) Match with a programmer on a specific project")
        #these prints will display our use cases, and as we are just a prototype we can add more cases at will
        print("(4) Other match")
        found_match = False
        selected_match = None
        choice = input("Select your option: ")
        if choice == "1":
            for u in pool:
                if u.get_experience() > 5 and u.get_availability() == new_user.get_availability():
                    selected_match = u
                    found_match = True
                    break
            if found_match:
                print("We found a match!")
                print(f"We are matching you with {selected_match.get_name()}, who was {selected_match.get_experience()} years of experience.
                pool.remove(selected_match)
            else:
                print("No match right now... adding to pool")
                #since no match but match now is true they will get priority in "Other match"
                pool.append(new_user)
        #finds people within two years of experience
        elif choice == "2":
            for u in pool:
                if u.get_experience() >= new_user.get_experience() - 2 and u.get_experience() <= new_user.get_experience() + 2 and u.get_av
```

nal Help quinnanderson [SSH: rlogin.cs.vt.edu]

driver.py

cs3704 > driver.py > ...

```python
            if u.get_experience() >= new_user.get_experience() - 2 and u.get_experience() <= new_user.get_experience() + 2 and u
                selected_match = u
                found_match = True
                break
        if found_match:
            print("We found a match!")
            print(f"We are matching you with {selected_match.get_name()}, who was {selected_match.get_experience()} years of exp
            pool.remove(selected_match)
        else:
            print("No match right now... adding to pool")
            #since no match but match now is true they will get priority in "Other match"
            pool.append(new_user)
    elif choice == "3":
        for u in pool:
            if u.get_project() == new_user.get_project() and u.get_availability() == new_user.get_availability():
                selected_match = u
                found_match = True
                break
        if found_match:
            print("We found a match!")
            print(f"We are matching you with {selected_match.get_name()}, who is working on the same project.")
            pool.remove(selected_match)
        else:
            print("No match right now... adding to pool")
            #since no match but match now is true they will get priority in "Other match"
            pool.append(new_user)
    elif choice == "4":
        for u in pool:
            #first check the "priority" matches
            if u.get_now == True and u.get_availability() == new_user.get_availability():
                selected_match = u
                found_match = True
                break
        if found_match:
            print("We found a match!")
            print(f"We are matching you with {selected_match.get_name()}, who is looking for a match.")
            pool.remove(selected_match)
        else:
            if u.get_availability() == new_user.get_availability():
                selected_match = u
                found_match = True
                break
            if found_match:
                print("We found a match!")
```

un Terminal Help quinnanderson [SSH: rlogin.cs.vt.edu]

user.py driver.py

cs2104 > cs3704 > driver.py > ...

```python
            for u in pool:
                #first check the "priority" matches
                if u.get_now == True and u.get_availability() == new_user.get_availability():
                    selected_match = u
                    found_match = True
                    break
            if found_match:
                print("We found a match!")
                print(f"We are matching you with {selected_match.get_name()}, who is looking for a match.")
                pool.remove(selected_match)
            else:
                if u.get_availability() == new_user.get_availability():
                    selected_match = u
                    found_match = True
                    break
                if found_match:
                    print("We found a match!")
                    print(f"We are matching you with {selected_match.get_name()}.")
                    pool.remove(selected_match)
                else:
                    print("No match right now... adding to pool")
                    pool.append(new_user) #still add to pool



    #for extra spacing
    print("")
    print("")
```

Use Case 1

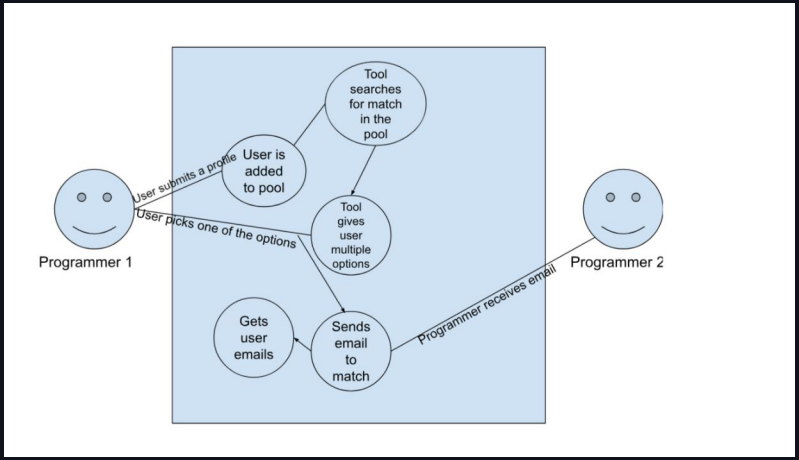
New programmer wants to match with a more experienced programmer for mentorship.

Precondition:

- User must create a profile that includes their preferences for availability, expertise, and more.

Main Flow:

- User will request a match [S1]. The matching tool will generate complimentative possible matches for the user to pick from based on their profile [S2].



Use Case 2

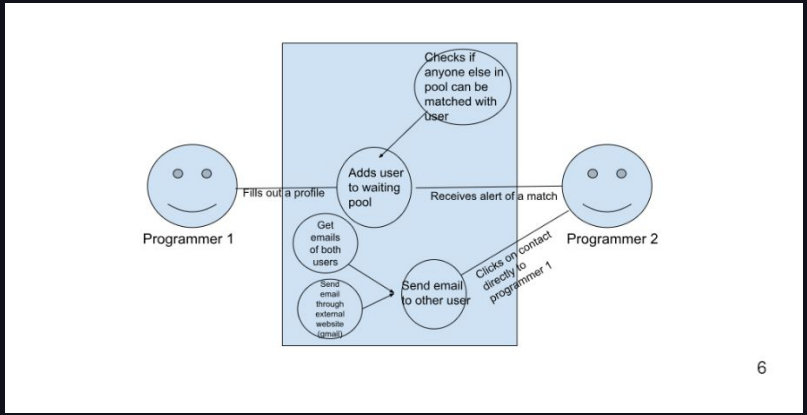
Programmer wants to join the pool to assist coworkers and does not need a match right now.

Precondition:

- User has preferences in general, like availability, expertise, etc.

Main Flow:

- User will open an application and select a button to make a profile [S1]. Once the form is submitted, the tool will update the user's preferences in the system [S2]. The user will be in the pool until another programmer who fits the requirements requests a match through the matching tool [S3]. If a match is made, the user is alerted and can directly contact the match to begin pair programming [S4].



Use Case 3

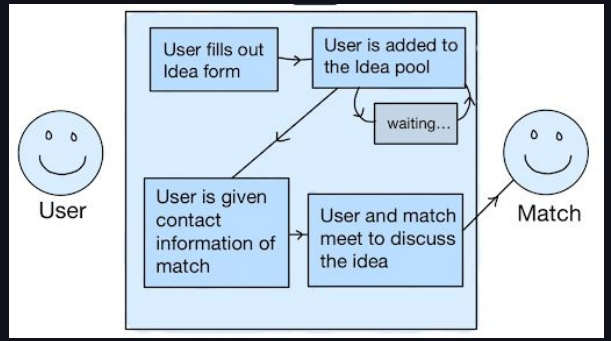
Engineer has an idea that they would like to see implemented, seeking a partner to work with and bounce ideas off of.

Precondition:

- User has an account with their basic information: work expertise, availability, etc...

Main Flow:

- User will open the application and request a match, they will fill out the form they are presented with, complete with a flag denoting they have a project in mind and the basics of said project [S1]. The user will be in the pool until another programmer is available [S2]. If a match is made, the user is alerted and can directly contact the match to set up a call and begin to discuss the potential project [S3].



Demoed Use Case

cs2104 > cs3704 > driver.py > ...

```
5    pool = []
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE PORTS GITLENS

```
[quinnanderson@chestnut cs3704]$ python driver.py
Would you like to create a profile to match? (y/n): y
Name: Quinn Anderson
Years of experience: 2
Enter availability (Days ie 'MW', 'TuThF', etc): MTuTh
Enter project: Project H
Do you want to immediately find a match? (y/n): y
You are looking for a match: Choose one of the number options below to describe your situation
(1) Match with a more experienced programmer for mentorship
(2) Match with a programmer with similar levels of experience
(3) Match with a programmer on a specific project
(4) Other match
Select your option: 1
We found a match!
We are matching you with Bob, who was 8 years of experience.


Would you like to create a profile to match? (y/n): y
Name: Brian Aguilar
Years of experience: 6
Enter availability (Days ie 'MW', 'TuThF', etc): MWF
Enter project: Project C
Do you want to immediately find a match? (y/n): n
You have successfully been added to pool!


Would you like to create a profile to match? (y/n): y
Name: Louie Kim
Years of experience: 3
Enter availability (Days ie 'MW', 'TuThF', etc): TuTh
Enter project: Project D
Do you want to immediately find a match? (y/n): y
You are looking for a match: Choose one of the number options below to describe your situation
(1) Match with a more experienced programmer for mentorship
(2) Match with a programmer with similar levels of experience
(3) Match with a programmer on a specific project
(4) Other match
Select your option: 3
We found a match!
We are matching you with David, who is working on the same project.


Would you like to create a profile to match? (y/n): n
Thank you for using the Pair Programming Matcher Tool!
[quinnanderson@chestnut cs3704]$
```

Thanks!

< Any questions? >

References

Image Citations

- https://formation.dev/
- https://gaoxiangluo.github.io/2021/09/16/I-have-been-accepted-to-Google-s-CS-Research-Mentorship-Program-CSRMP/